Automatic Color Based Reassembly of Fragmented Images and Paintings

Efthymia Tsamoura and Ioannis Pitas, *Fellow, IEEE*

***Abstract*—The problem of reassembling image fragments arises in many scientific fields, such as forensics and archaeology. In the field of archaeology, the pictorial excavation findings are almost always in the form of painting fragments. The manual execution of this task is very difficult, as it requires great amount of time, skill and effort. Thus, the automation of such a work is very important and can lead to faster, more efficient, painting reassembly and to a significant reduction in the human effort involved. In this paper, an integrated method for automatic color based 2-D image fragment reassembly is presented. The proposed 2-D reassembly technique is divided into four steps. Initially, the image fragments which are probably spatially adjacent, are identified utilizing techniques employed in content based image retrieval systems. The second operation is to identify the matching contour segments for every retained couple of image fragments, via a dynamic programming technique. The next step is to identify the optimal transformation in order to align the matching contour segments. Many registration techniques have been evaluated to this end. Finally, the overall image is reassembled from its properly aligned fragments. This is achieved via a novel algorithm, which exploits the alignment angles found during the previous step. In each stage, the most robust algorithms having the best performance are investigated and their results are fed to the next step. We have experimented with the proposed method using digitally scanned images of actual torn pieces of paper image prints and we produced very satisfactory reassembly results.**

I. Introduction

THE problem of reassembling image fragments arises in many scientific fields, such as forensics and archaeology. In the field of archaeology, the pictorial excavation findings are almost always in the form of painting fragments. For example, they can be fragments of painted pottery, murals, or mosaics, which must be assembled to form the original painting. A related aspect of the problem is the development of a generative model for cracks and fractures as proposed in [1]. More rarely, there are cases where the form of the original object (e.g., mosaic) is known but has to be reassembled because of a destruction. In [2], a pattern matching algorithm for the comparison of digital images is implemented using discrete circular harmonic expansions. The manual execution of the above tasks is very difficult, as it requires great amount of time, skill and effort. Thus, the automation of such a work is very important and can lead to faster, more efficient, painting reassembly and to a significant reduction in the human effort involved.

In our work, the automated reassembly of images from fragments follows a four step model, similar to the one presented in [3] for 3-D object reconstruction. The first step of our approach is the identification of probable adjacent image fragments, in order to reduce the computational burden of the subsequent steps. There, several color-based techniques are employed. However, as stated in a following section, this step can be omitted sacrificing the lower running time of the image reassembly procedure, for higher performance.

The second operation is the identification of the matching contour segments of the image fragments. The corresponding step employs a neural network based color quantization approach for the representation of the image contours, followed by a dynamic programming technique that identifies their matching image contour segments. In [4], an analogous algorithm is presented, for the discovery of the matching contour segments of 2-D fragmented objects. However, the fragment contour comparison is based on the shape of the input 2-D object fragments, while a dynamic programming technique is employed in order to identify their matching segments. Aminogi *et al.* [5], presented an algorithm based on both the shape and the color characteristics of the input 2-D image fragments contours. There, one contour pixel sequence is overlaid on another one and, for each such "placement", the curvature and color differences of the corresponding contour pixels are estimated. If their total sum is less than a user defined threshold, the contour segments are considered to match. In the experimental results section, the latter approach is compared with the one proposed here. The results show that the latter has better performance.

Once the matching contour segments are identified, a third operation takes place. Here, the geometrical transformation, which best aligns two fragment contours along their matching segments, is found. Several such approaches exist, e.g., [6]–[8], however, most of them are either not robust to matching errors and/or have high computational complexity. A very popular registration technique is the *Iterative Closest Point (ICP)* method [9]. New modified versions of the ICP have been proposed, which are robust to noise [10]–[12]. These approaches limit the effects of noise on the registration performance by outlier trimming based on a least squares distance criterion. In our work, the above ICP variants were experimentally evaluated and the best one among them is selected for integration with the proposed four stage image reassembly algorithm.

The last step in solving the fragment reassembly problem is the reassembly of the overall image from its constituent

Manuscript received March 10, 2009; revised September 29, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Pier-Luigi Dragotti.

The authors are with the Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece (e-mail: etsamour@csd.auth.gr; tsamoura@delab.csd.auth.gr; pitas@aiia.csd.auth.gr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2009.2035840

fragments. Here, a novel algorithm is proposed. The novelty of our algorithm lies on the fact that employs both the contour matching results and the alignement angles of the fragments, found during the second and the third step, respectively. The majority of the algorithms in this step, as will be presented in the related work section, utilize only criteria that are based on the contour matchings, e.g., the length of the matching. It is clear that it is essential that each step of the algorithm feeds the next one with correct results, otherwise the image reassembly may contain errors, or may even fail completely. Our goal is to investigate and propose the most robust techniques in order to produce accurate results at each intermediate step. To summarize, the main steps of the proposed method are shown in Fig. 1 and can be described as follows.

1) **Discovery of spatial adjacent image fragments**. The identification of probable spatially adjacent image fragments is done by utilizing techniques that are widely employed in content based image retrieval (CBIR) systems. The purpose of this step is to reduce the computational burden of the steps that follow. Once this step is finished, we select to retain, for every image fragment, a list of the most probable adjacent fragments.
2) **Discovery of matching contour segments of adjacent image fragments**. A novel approach based on the Smith-Waterman algorithm [13] is employed in order to match the colors appearing in the contours of adjacent image fragments. Various color similarity criteria are being evaluated. Based on such similarity criteria, for each image fragment, one matching contour segment with K other image fragments is retained.
3) **Image fragments contour alignment**. The purpose of this step is to find the appropriate geometrical transformation of one fragment relative to its adjacent one, in order to align them along their matching contour segments. Many variants of the ICP algorithm are employed and evaluated to this end.
4) **Overall image assembly**. Once the matching contour segments of couples of input image fragments are identified and properly aligned, the remaining step is the reassembly of the overall image. Since the criteria that are based on the contour matchings do not suffice for the overall image reassembly, a novel feature, namely the alignment angles found during step 3, is introduced.

In this paper, an integrated method for automatic color based 2-D image fragment reassembly is presented. This paper is organized as follows. Section II discusses the related work. Section III describes the discovery of the spatial adjacent fragments. Section IV presents the identification of the matching contour segments of the spatial adjacent image fragments, while Section V describes the derivation of the optimal geometrical transformation that aligns contours along their matching segments. Section VI presents the overall image assembly algorithm. Experimental results are presented in Section VII, while conclusions are drawn in Section VIII.

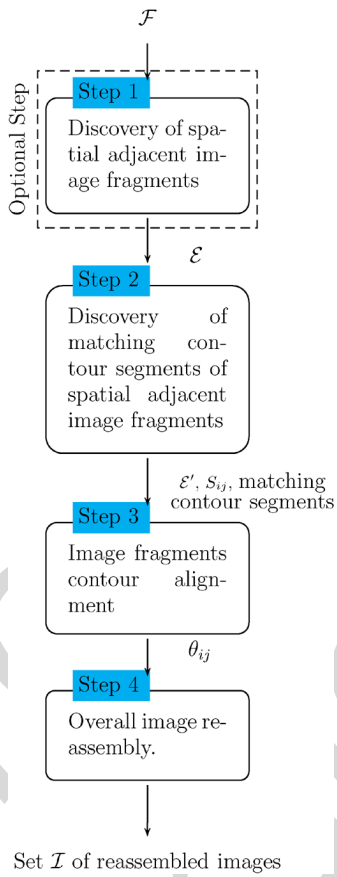

Fig. 1. Overall image reassembly approach.

II. Related Work

A. Two-Dimensional Paper Document Reassembly

Similar to 2-D image fragment reassembly, in paper document reassembly, torn paper fragments must be assembled to form the image of an entire page of a paper document. Work on this area was conducted in [14]–[16]. The above employ shape representations of the paper fragments, in order to reassembly the original documents. In [15], polygonal approximation is initially applied to reduce the complexity of the paper fragment contours and geometrical features are extracted from these polygonal curves. Then, a method based on [4] is used to assemble the entire document from its constituent paper fragments. In [16], shape features, namely turning functions, are estimated from every fracture contour and are utilized to discover matching contour segments. After that, each matching is assigned a confidence score. The alignment transformation of the fragments is simultaneously found during matching. After the discovery of matching contour segments, the final reassembly step performs two actions, namely matching relaxation and fragments merging. During matching relaxation, every pair of aligned input fragments is checked for overlap along their matching contour segments. If they overlap, this matching is discarded. Otherwise, the neighboring fragments of this pair are identified. A score, called support, is assigned to the neighborhood of each pair of non-overlapping fragments. This score increases as the number of neighboring fragments as

well as the matching confidence assigned to pairs of fragments increase. The fragments that have neighborhoods with maximum support are merged into new fragments and the whole procedure starts again, i.e., the matching contour segments are identified for all pairs of fragments, and so on.

B. Two-Dimensional Puzzle Assembly

Many methods were also proposed for the 2-D puzzle reassembly problem. In [17], color and textural features of the puzzle pieces are utilized. The matching and alignment of puzzle pieces is carried out using an FFT-based image registration technique. In [18], the puzzle reassembly process consists of two steps; frame and interior assembly. A traveling salesman problem (TSP) is formulated for frame assembly, while backtracking and branch&bound techniques are employed for interior assembly. The puzzle pieces are matched employing the L_2 distance of their contours curves. An improvement of this method is presented in [19]. In [20], the overall puzzle assembly is done using a Best-First procedure. There, two criteria are utilized to sort matching contour segments. The first one is the residual error of corresponding contour pixels after the discovery of the optimal geometrical transformation, while the second criterion is the arc-length of the matching contour segments. It is clear that the problem of 2-D puzzle assembly does not meet the major difficulty of the image and object reconstruction problems; that is the missing or highly damaged image (or) object fragments. Thus, in general, the algorithms proposed in this field would be inadequate to solve such problems.

C. Three-Dimensional Object Reconstruction

Regarding 3-D object reconstruction, an automatic method for matching and alignment of 3-D, free-form archaeological fragments is proposed in [21]. The input fragments are not pre-processed. The matching is performed utilizing only the 3-D points of the whole surfaces of the objects. The output matching-alignment minimizes the distance between the 3-D surface points of the two fragments. Andrews *et al.* [22] propose an automatic method for the reconstruction of pairs or triplets of 3-D symmetric archaeological fragments. The matching is found through a two-phase method. During the first phase several matchings-alignments are estimated for every pair of fragments. The 3-D points of the fracture curves in the outer and inner surface of the fragments as well as the axis of rotation of the fragments are utilized to this end. In the second phase, these matchings are refined using the quasi-Newton method, and evaluated according to several criteria namely the angle formed by the fragments rotation axes, the perpendicular distance between the rotation axes and the distance of the matched fracture curves points. Eventually, one matching is retained for every pair of fragments. Finally, in the overall object reconstruction step, a greedy merge strategy selects pairs of fragments to form triplets In [23], a human-supervised collaborative reconstruction system is described. The aim of the system is to propose a potential matching between any pair of input fragments. The matching is found by utilizing shape features (curvature and torsion) estimated from all 3-D points in the fracture curves of the fragments. The shape similarity of the fracture curves is ranked with a cyclic distance algorithm. Each matching defines correspondences between 3-D points in the fracture curves. Then the users select to merge or not the proposed fragments. The fragments alignment is performed interactively (in a VRML environment) by the users. The object reconstruction procedure follows the merge-update paradigm. Significant work on this area is also done in the Digital Forma Urbis Romae project [24]. The latter seeks to reconstruct a giant marble map of ancient Rome dating back to 200 AD, while only a small portion of fragments of the original map still exist. The reconstruction process in [24] is not automatic and is guided by expert users. Information and details about other 3-D object reconstruction methods can be found in [3].

Finally, in [25], a method is introduced to measure the average amount of information contained in a 2-D ceramic tile fragment contour, in terms of its curvature. This parameter shows how many false contour matches are expected to be found among a given set of ceramic tile fragments.

III. Discovery of Spatial Adjacent Image Fragments

The purpose of this step is the spatial adjacent image fragments identification by using their probable high color similarity. We have utilized techniques that are widely employed in CBIR systems in order to identify these similarities [26]. In this section, we briefly mention the descriptors and the measures we have experimented with.

Color quantization can be based on a commercial color palette, e.g., the Gretag Macbeth Color Checker [26]. The Macbeth Palette can be used to evaluate color reproduction systems and it consists of 24 different colors that are scientifically chosen to represent a variety of naturally occurring colors. Color quantization is used to find the normalized quantized color image histograms, which can be used for color image retrieval. We have also experimented with the Spatial Chromatic Histogram [27], which provides information both of color presence and color spatial distribution. The Spatial Chromatic Histogram S_I of image I having C quantized colors is given by $S_I(i) = (h(i), \mathbf{b}(i), \sigma(i)),\ i = \{1, \ldots, C\}$. In the above equation h denote the normalized color histogram, i.e., $h(i)$ is defined as the number of pixels having color i divided by the total number of pixels, $\mathbf{b}(i)$ is a 2-D vector expressing the center of mass and $\sigma(i)$ is the standard deviation of the i^{th} color label, respectively [27]. We have used modified versions of the L_1, L_2 and Histogram Intersection measures and scaled them to the range [0, 1], with 1 denoting a perfect similarity. In the following equations h_1 and h_2 denote the normalized color histograms extracted from images I_1 and I_2, respectively. The utilized matching measures are the following.

1) Scaled L_1 norm

$$d_{L_1}(h_1, h_2) = 1 - 0.5 \sum_{i=1}^{C} |h_1(i) - h_2(i)| . \quad (1)$$

2) Scaled L_2 norm

$$d_{L_2}(h_1, h_2) = 1 - \frac{1}{\sqrt{2}} \sum_{i=1}^{C} (h_1(i) - h_2(i))^2 . \quad (2)$$

3) Scaled Histogram Intersection

Inputs
$\mathcal{F}$: set of N image fragments
L: the size of most chromatically similar fragments per input image fragment.
Outputs
$\mathcal{E}$: set of image fragments couples.

1: $\mathcal{S} \leftarrow \emptyset$;{$\mathcal{S}$ is a list of spatial chromatic histograms.}
2: **for all** $f \in \mathcal{F}$ **do**
3: quantize f using Gretag Macbeth Color Checker ;
4: estimate the spatial chromatic histogram of image fragment f, S_f, according to [27];
5: append S_f to $\mathcal{S}$;
6: **end for**
7: $\mathcal{E} \leftarrow \emptyset$;
8: **for** $i = 1$ to $N - 1$ **do**
9: **for** $j = i + 1$ to N **do**
10: $m[j - i - 1] = d(S_{f_i}, S_{f_j})$; {m is a one dimensional real matrix. d is one from (1) to (4).}
11: **end for**
12: sort m in descending order ; $thr = m[L]$;
13: $\mathcal{E} = \mathcal{E} \cup \{(f_i, f_j) | d(S_{f_i}, S_{f_j}) \geq thr\}$;
14: **end for**

Fig. 2. First step of the proposed 2-D image reassembly approach.

$$d_{HI}(h_1, h_2) = \sum_{i=1}^{C} \min\left(h_1(i), h_2(i)\right)\left(1 - |h_1(i) - h_2(i)|\right). \quad (3)$$

4) Spatial Chromatic distance [27]

$$d_{SC}(I_1, I_2) = \sum_{i=1}^{C} \min\left(h_1(i), h_2(i)\right) \times \left(\frac{\sqrt{2} - \|b_1(i) - b_2(i)\|^2}{\sqrt{2}} + \frac{\min\left(\sigma_1(i), \sigma_2(i)\right)}{\max\left(\sigma_1(i), \sigma_2(i)\right)}\right). \quad (4)$$

Once this step is finished, we select to retain, for every image fragment, a list of the L most chromatically similar fragments. The first step is shown in Fig. 2. It must be emphasized that this step is not prerequisite for the correct reassembly of an image and can be easily skipped, if the number of image fragments is rather small. The purpose of this task is to reduce the computational burden of the steps that follow.

IV. Discovery of Matching Contour Segments of Adjacent Image Fragments

Let us suppose that we have a set of image fragments where we have to find their matching contour segments. This set may contain the image fragment pairs identified by the previous step or all the image fragment pairs (if the first step has been skipped). This operation is performed on image fragment pairs. In this section, we present a novel algorithm for identifying matching contour segments of couples of input fragments. Our approach to fragment contour matching is based exclusively on information regarding the color of their contours.

In order to avoid comparing directly contour pixel colors that may contain noise, a color quantization preprocessing step is utilized, which takes pixel samples from the contours of all image fragments. Many color quantization methods exist [28]. For example, quantization can be done using ready made palettes, such as the Gretagh Macbeth palette [26]. The Mean Shift algorithm [29] is a popular solution for color image quantization. Its advantage is that the number of the color clusters is automatically defined. However, the Mean Shift algorithm exhibits a severe local maxima sensitivity. This drawback may result to significant misplacements of color cluster centers, as will be shown in the experimental results section.

In the following, we employ Kohonen neural networks (KNNs) [30] for color quantization purposes. KNNs belong in the class of unsupervised neural networks. They can cluster input vectors without any external information, following an iterative procedure based on competitive learning [30]. KNNs consist of two node layers; the input and the output layer. In the former, the number of nodes equals the dimension of input vectors, while in the latter the number of nodes equals the amount of produced clusters. The nodes in the output layer are organized by means of a lattice [30]. In KNNs, each node in the input layer $\mathbf{s}_i$ has a connection w_{ik} with every node $\mathbf{c}_k$ in the output layer. For a network with n input nodes, the weight vector $\mathbf{w}_j = [w_{1k}, w_{2k}, \ldots, w_{nk}]$ ending at an output node c_k, is the center of a cluster. In KNNs, given an input vector, the output node with the highest response (winning node) for that as well as all "neighboring" nodes that belong to an area around it, update their weight vectors.

In the following, we shall describe the KNN variant employed to perform color quantization of image fragments. First, a random number of N_p pixels is sampled from the input image fragments and mapped to La^*b^* color space. The number of sampled N_p pixels that are required to successfully quantize the color space, is a minimal portion of the total image fragments' pixels, as it will be demonstrated in the experimental results section. After that a $[3 \times C]$ KNN is defined, where 3 corresponds to the dimension of the input La^*b^* space and C to the predefined number of color clusters. Let $\mathbf{x} = [x_1, x_2, x_3]$, be one of the N_p sampled pixels, after mapping to the La^*b^* color space. The following learning procedure is iteratively applied.

1) A winning node c_j is selected, i.e., output node whose weight vector $\mathbf{w}_j$ has the highest similarity with the input vector $\mathbf{x}$, than any other output node c_k

$$\|\mathbf{x} - \mathbf{w}_j\| = \min_{\forall c_k} \{\|\mathbf{x} - \mathbf{w}_k\|\}. \quad (5)$$

Euclidean distance is utilized in our experiments as the similarity criterion.

2) A neighborhood function is utilized to estimate the weight vectors updates. Thus, the weight vector $\mathbf{w}_k$ of an output node c_k is updated under

$$\Delta_{\mathbf{w}_k} = \gamma \Omega_{c_j}(c_k)\left(\|\mathbf{x} - \mathbf{w}_k\|\right) \quad (6)$$

where $\Omega_{c_j}(c_k) = e^{(-\|\mathbf{p}_k - \mathbf{p}_j\|^2)/2\sigma^2}$, γ is the learning parameter $(0 < \gamma < 1)$, σ denotes the spread of the "neighborhood" around the winning node c_k and $\mathbf{p}_k$, $\mathbf{p}_j$ correspond to places inside the lattice of an output node c_k and the winning c_j, respectively.

For better convergence, the learning parameter γ and the standard deviation σ of the neighborhood function gradually decrease after an iteration (epoch in neural networks terminology).

The learning procedure stops, when several criteria are met, e.g., when a predefined number of epochs have finished, or when the changes $\Delta_{\mathbf{w}_k}$ of clusters' centers are very small. After training the network, the weight vector of every output node corresponds to a cluster center. Details about the above parameters, as well as a comparison of the above color quantization techniques can be found in Section VII. Following one of the previously described approaches, the color of every contour pixel is represented by its color cluster label c_j, $j = \{1, \ldots, C\}$.

Let $U = [u_i]_{i=1}^{n}$ and $V = [v_j]_{j=1}^{m}$ be two pixel sequences that follow the entire contours of two different image fragments. Once the color clustering has been achieved for the input image fragments, sequences $[a_i]_{i=1}^{n}$ and $[b_j]_{j=1}^{m}$ are assigned to U and V respectively. a_i and b_j are two color cluster labels, where the former is estimated at pixel u_i and the latter at pixel v_j. Then u_i and v_j have the same color, if $a_i = b_j$. We define a similarity function F of the two contour pixels as follows:

$$F_{u_i,v_j}[a_i, b_j] = \begin{cases} e > 0, & a_i = b_j \\ d < 0, & a_i \neq b_j. \end{cases} \tag{7}$$

Given U and V and their label lists $[a_i]_{i=1}^{n}$ and $[b_j]_{j=1}^{m}$, we search for the contour pixel mapping function $\mathbf{\Phi}$, such that:

- for every $\Phi[u_i] = v_k$ and $\Phi[u_{i+1}] = v_l$, $k \leq l \leq m$;
- $\Phi[u_i] \neq \emptyset$.

The first condition means that more than one contour pixels in U can be mapped to the same contour pixel in V. However, the former contour pixels must be strictly consecutive. This condition guarantees that no "folding" of the U contour pixel sequence, in order to match it with the corresponding matching contour segment in V, will take place. The second condition ensures that every contour pixel in U is mapped to a contour pixel in V. The algorithm that is used to identify the mapping function $\mathbf{\Phi}$ is a variant of the Smith Waterman dynamic programming algorithm [13], which is a local sequence matching algorithm: given two input sequences, it can identify the mapping function between their segments. Each mapping between segments of input sequences is assigned with a score $S > 0$; as the score S increases, the color similarity becomes higher. A similarity $n \times m$ matrix $\mathbf{H}$ is set up, where the i^{th} row of matrix $\mathbf{H}$ corresponds to u_i, while the j^{th} column corresponds to v_j. $H_{i,j}$ is the best mapping score S of the pair of every sub-sequences ending at u_i and v_j pixels. The algorithm gradually fills matrix $\mathbf{H}$ and forms the mapping function $\mathbf{\Phi}$. During filling, each matrix cell is assigned with the highest possible value, as our purpose is to maximize the mapping score S.

As in a typical dynamic programming algorithm, the solution to an instance of the problem is given in terms of solutions to its smaller sub-instances. Thus

$$H_{i,j} = \begin{cases} H_{i-1,j-1} + F_{u_i,v_j}[a_i, b_j] \\ H_{i,j-1} + g \\ H_{i-1,j} + g \end{cases} \tag{8}$$

where $g < 0$. $H_{i,j}$ is assigned the maximum value of the right hand terms in (8). Let $\mathbf{\Phi}$ be the estimated mapping of $[u_k]_{k=1}^{i-1}$ and $[v_l]_{l=1}^{j-1}$ subsequences. If $H_{i,j} = H_{i-1,j-1} + F_{u_i,v_j}[a_i, b_j]$, then the relation $\Phi[u_i] = v_j$ is appended to $\mathbf{\Phi}$. On the other hand, if $H_{i,j} = H_{i,j-1} + g$ or $H_{i,j} = H_{i-1,j} + g$ is selected, then $\Phi[u_i] = v_{j-1}$ or $\Phi[u_{i-1}] = v_j$ is appended to $\mathbf{\Phi}$. A *gap* is formed, when either the second or the third case occurs in (8). Generally, a gap is formed when one or more contour pixels of the first fragment are mapped to the same contour pixel of another fragment. The percentage of mapping gaps is a measure of the dissimilarity between the input contour segments; a high gap percentage reveals contours with many dissimilarities. The parameter g is negative, in order to penalize mappings with many gaps. When this stage is completed, we identify an area in $\mathbf{H}$ with high similarity values H_{ij}. Diagonal patterns are preferred, since they correspond to mappings without many gaps. Let H_{e_1,e_2} and H_{s_1,s_2} be the lowest right and highest left borders of this area. In the implemented experimental modification, H_{e_1,e_2} is the maximum value of $\mathbf{H}$, while $H_{s_1,s_2} = 0$. Then we select the mapping starting from (s_1, s_2) and ending at (e_1, e_2), which is formed during filling $\mathbf{H}$. The Smith Waterman algorithm steps are shown in Fig. 4.

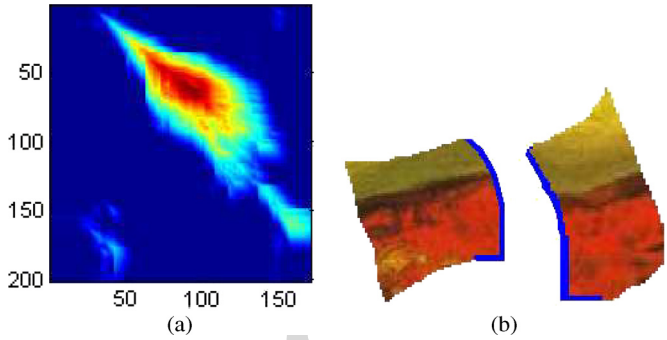

Fig. 3. (a) Color similarity matrix. (b) Matching based on color information. Common contour segments are denoted by blue.

A basic question is how to select e, d and g. These parameters are strongly correlated with the desired characteristics of the contour matching. Specifically, if a matching without many gaps is desirable, then a high absolute value is selected for g. Otherwise, the value of d must be close to g. The complexity of the proposed method for a pair of fragments is $O(nm)$, where n and m are the lengths of the two pixel contour sequences. For an image that is partitioned among into N fragments, the worst case computational complexity scenario (omitting the first step) is $O(N^2\widetilde{n}^2)$, where $\widetilde{n}$ is the average contour length of input image fragments. Such a similarity matrix is shown in Fig. 3(a). The horizontal axis of the matrix corresponds to the quantized contour pixels color sequence of the first image, while the columns correspond to the quantized contour pixels color sequence of the second image. High values correspond to contour segments with a high degree of similarity (deep red colored areas). The correctly matched color contour sequences are shown in Fig. 3(b).

After completing the second step, for each image fragment f, one matching contour segment with $K(0 \leq K \leq L)$ other image fragments is retained, producing the set of *true* adjacent image fragments couples $\{(f, f_i), i = 1, \ldots, K\}$.

V. Image Fragments Contour Alignment

In this section, we investigate methods for finding the best geometrical transformation that aligns fragment contours along

Inputs
$U = [u_i]_{i=1}^{n}$: contour pixel sequences of fragment f_p
$V = [v_j]_{j=1}^{m}$: contour pixel sequences of fragment f_r
$[a_i]_{i=1}^{n}$: color cluster label sequences of U
$[b_j]_{j=1}^{m}$: color cluster label sequences of V
parameters of the Smith-Waterman algorithm e, d, g: $e > 0$, $d < 0$, $g < 0$
Output
$\mathbf{\Phi}$: a mapping function between $[u_i]_{i=1}^{n}$ and $[v_j]_{j=1}^{m}$.
S_{pr}: the mapping score of $\mathbf{\Phi}$.

{Initialize $\mathbf{H}$ }
for $i = 1$ to n **do**
  for $j = 1$ to m **do**
    $H_{i,j} = F_{u_i,v_j}(a_i, b_j)$;
  end for
end for

for $i = 1$ to n **do**
  for $j = 1$ to m **do**
    $H_{i,j} = \max\{H_{i-1,j-1} + F_{u_i,v_j}(a_i, b_j), H_{i-1,j} + g, H_{i,j-1} + g, 0\}$
    {The zero value in is used in order to prevent **H** entries from taking negative values.}
  end for
end for

Select an area in matrix $\mathbf{H}$. Let H_{e_1,e_2} and H_{s_1,s_2} be the lowest right and highest left borders of this area;
$S_{pr} = H_{e_1,e_2}$;
$i = e_1$; $j = e_2$;
while $\{i \geq s_1 \wedge j \geq s_2\}$ **do**
  $index = \max\{H_{i-1,j-1}, H_{i-1,j}, H_{i,j-1}\}$;
  {$index$ corresponds to the place of the maximum element in $\{H_{i-1,j-1}, H_{i-1,j}, H_{i,j-1}\}$, i.e. $index \in \{1, 2, 3\}$}
  if $index = 1$ **then**
    $\Phi[u_i] = v_j$; $i = i - 1$; $j = j - 1$;
  else if $index = 2$ **then**
    $\Phi[u_{i-1}] = v_j$; $i = i - 1$;
  else
    $\Phi[u_i] = v_{j-1}$; $j = j - 1$;
  end if
end while

Fig. 4. Second step of the proposed reassembly approach.

their matching segments. All matching contour segments identified during the previous steps must be properly aligned, before the reassembly of the overall image. The disadvantages of the alignment algorithms found in the literature (as discussed in the introduction) have led us to use variants of the ICP algorithm [9] for contour alignment. The ICP algorithm generally starts with two point sets (contour segments in our case) and an initial guess of their relative rigid body geometrical transformation. It then refines the transformation parameters, by iteratively generating pairs of point correspondences and by minimizing an error metric. Given two curves $\mathcal{P} = \{\mathbf{p_1}, \ldots, \mathbf{p_{N_p}}\}$ and $\mathcal{M} = \{\mathbf{m_1}, \ldots, \mathbf{m_{N_m}}\}$, the following steps are performed.

1) Compute the subset of pairs of closest points

$$\mathcal{Y} = \{(\mathbf{p}_i, \mathbf{m}_j) | \mathbf{p}_i \in \mathcal{P}, \mathbf{m}_j \in \mathcal{M} \; \mathbf{m}_j \text{ is the closest point to } \mathbf{p}_i\}. \quad (9)$$

2) Compute a Least Squares estimate of the geometrical transformation mapping $\mathcal{P}$ onto $\mathcal{M}$

$$(\mathbf{R}, \mathbf{t}) = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{|\mathcal{Y}|} \|\mathbf{m}_i - \mathbf{R}\mathbf{p}_i - \mathbf{t}\|^2 \quad (10)$$

where $(\mathbf{p}_i, \mathbf{m}_i) \in \mathcal{Y}$.

3) Apply the transformation to the $\mathcal{P}$ data points

$$\mathcal{P} = \mathbf{R}\mathcal{P} + t. \quad (11)$$

4) If stopping criterion is satisfied exit; else, go to step 1.

The original form of the ICP algorithm is not robust to outliers, since it does not trim noisy data. Such outliers, coming from errors during the contour segment matching, create serious problems to alignment, if not properly handled. In order to obtain ICP versions that are robust to outliers, many ICP variants have been proposed in the literature. In [31], these variants have been classified into six categories. In order to develop a robust contour alignment procedure, we have studied three popular robust versions of ICP and have evaluated their performance.

A. Robust ICP (RICP)

The first variant that we studied is RICP [10]. It trims outliers in the second step (10) of the ICP and computes an L_2 error norm, which is, hopefully, free from outlier influence. Outliers are rejected, according to the following procedure. Let $\mathbf{p_i}$ be points from the first curve and $\mathbf{m_j}$ be their corresponding closest points on the second curve. Select randomly two points from each of the 2-D input curves and execute their registration, by solving the linear system

$$\begin{pmatrix} \mathbf{m}_{1_x} & \mathbf{m}_{2_x} \\ \mathbf{m}_{1_y} & \mathbf{m}_{2_y} \end{pmatrix} = \mathbf{R} \begin{pmatrix} \mathbf{p}_{1_x} & \mathbf{p}_{2_x} \\ \mathbf{p}_{1_y} & \mathbf{p}_{2_y} \end{pmatrix} \quad (12)$$

where $\mathbf{R}$ is a rotation matrix. Compute the residuals $\mathbf{s} = \mathbf{R}\mathbf{p} - \mathbf{m}$. The above procedure is repeated for a sufficient number of iteration [10]. Once all potential registrations are evaluated, the one that minimizes the median of the residuals is chosen. The correspondences having residual error larger than a fixed threshold are removed. The least squares error norm is computed using only the remaining contour points.

B. Trimmed ICP and Picky ICP

The main steps of both trimmed ICP and picky ICP algorithms are the following [11].

1) For each point of $\mathcal{P}$, find the closest point in $\mathcal{M}$ and compute the individual distances d_i^2.
2) Sort d_i^2 in ascending order, select the N_{po} least values and calculate their sum S'_{LTS}.
3) If any of the stopping conditions is satisfied, exit; otherwise, set $S_{LTS} = S'_{LTS}$ and continue.
4) For the N_{po} selected pairs, compute the optimal geometrical transformation $(\mathbf{R}, \mathbf{t})$ that minimizes S_{LTS}.
5) Transform $\mathcal{P}$ according to $(\mathbf{R}, \mathbf{t})$ and go to step 1.

This algorithm terminates either if (a) the maximum number of iterations is reached, or (b) the *trimmed mean squared error* $e = S_{LTS}/N_{op}$ is less than a user defined threshold, or (c) the relative change of the trimmed mean squared error $|e - e|'$,

where e' is the trimmed mean squared error found in the previous iteration, is less than a user defined threshold. Despite their good performance, the above algorithms, namely RICP, trimmed IC and picky ICP have the disadvantage that the outlier percentage in the input curves, must not exceed 20%. Otherwise, the computed transformation is wrong.

C. ICP Registration Using Invariant Features (ICPIF)

ICPIF is another ICP variant proposed in [32]. New features were introduced in order to improve the correspondence selection, such as the second order moments and the spherical harmonics. After the computation of the above features for every point of $\mathcal{P}$ and $\mathcal{M}$, the traditional L_2 distance, computed during the second step of ICP (10), is replaced by the weighted sum of the old L_2 distance and distances based on the introduced features. The goal is to estimate point correspondences that are not only based on the Euclidean metrics but also incorporate new shape invariant features.

VI. Overall Image Reassembly

Once the true adjacent image fragments couples and their matching contour segments are identified and properly aligned, as described in Sections IV and V, the remaining step is the reassembly of the overall image. However, the mapping scores of the matching contour segments do not suffice for image reassembly, since they are not always well correlated with the true ones. Thus, the approaches that focus only on using the mapping scores, such as [20], are inadequate for producing the true overall image. For this reason, we have developed a novel solution to this problem, which is based on the introduction of new use of the alignment angle, that best aligns the matching contour segments of two image fragments, as described in Section V. Consider three image fragments f_i, f_j and f_k each one having one matching contour segment with the rest ones. We denote by θ_j the rotation angle by which the individual fragment f_j must be rotated, in order to be correctly placed inside the overall reassembled image. The alignment angle, by which we must rotate fragment f_i to align it with the matching contour segment of fragment f_j (before fragment f_j is rotated by θ_j), is denoted by θ_{ij}. In order to align fragments f_i and f_j with respect to each other and place them correctly in the reassembled image, the following steps must be performed:

1) Rotate fragment f_j by θ_j to correctly orient it in the assembled image.
2) Rotate fragment f_i by $\theta_{ij} + \theta_j$ to correctly align its matching contour segment with the corresponding matching contour segment of fragment f_j.

This procedure will simultaneously align fragment f_i with fragment f_j and provide its correct orientation inside the entire image. We can then state that the matching contour segments of pairs (f_i, f_j) and (f_i, f_k) are *compatible*, if and only if

$$\theta_{ij} + \theta_j = \theta_{ik} + \theta_k. \tag{13}$$

Following this logic, in our approach we consider that an image is fully reconstructed, if all its matching contour segments are compatible, according to (13); this image is called *valid*. It should be noted that a valid image is not always the "correct" one, since the contour segments of two image fragments can be mis-matched or mis-aligned. However, the experimental evaluation of the technique shows that such flaws do not significantly affect the reassembly performance of the proposed technique.

Contour segments that match may become incompatible, if the following errors occurred in the previous steps:

1) the fragment color similarities found in Section III;
2) the matching segments of two fragment contours are different from the true ones (see Section IV);
3) wrong alignment transformation was estimated (see Section V).

Based on (13), we can define the so-called *relative alignment angle*, which will direct the image reassembling procedure. The relative alignment angle ϕ_i^j of a fragment f_i regarding a fragment f_j is evaluated by the formula:

$$\phi_i^j = \theta_{ij} + \theta_j \tag{14}$$

where θ_{ij} is the alignment angle by which we must rotate fragment f_i to align it with the matching contour segment of fragment f_j and θ_j is the rotation angle of the individual fragment f_j so that it is correctly placed inside the overall reassembled image. It can be easily deduced from the above analysis, that for a set of fragments placed in the reassembled image and a fragment f_i having matching contour segments with a subset of the above image fragments, i.e., $I = \{f_1, f_2, \ldots, f_n\}$, the above matching contour segments are compatible, if

$$\phi_i^1 = \phi_i^2 = \phi_i^3 = \ldots = \phi_i^n. \tag{15}$$

Consequently, if a new fragment i is to be matched with an image that it consists of image fragments, then the amount of the new valid images is equal to the cardinality r of the relative alignment angle set, $\{\phi_i^l, l = 1, \ldots, r\}$. We have developed a reassembly algorithm that utilizes the aforementioned assumptions to produce a user-defined number of possible reassembled images. The algorithm is initialized by taking M couples of input fragments, where $M \leq N * K$ and N is the number of input image fragments. These M couples have the highest matching scores of the corresponding matching contour segments. These couples are reassembled (correctly aligned) to produce a number of M input images that will be further extended by inserting one fragment each time. The selection of the initial number of M image couples is crucial for the algorithm performance, since the involvement of an erroneous input pair of fragments would inevitably lead to a wrong image reconstruction. The reliability of the proposed initialization is both intuitionally expected and experimentally proven.

The set of M input images is iteratively updated in order to include further image fragments. At every step and for every input image (initially consisting of a couple of image fragments), the fragment having the maximum mapping score with an image fragment that belongs in the reconstructed image, is added. After computing the relative alignment angles of this fragment regarding the input image parts, a number of images, equal to the cardinality of these relative aligned angles, are reassembled for each input image. Those images compose the new set of input reassembled images (replacing the previous ones), that will be included in the next iteration. Consequently,

Inputs
$\mathcal{F}$: set of N image fragments.
$\mathcal{E}'$: set of retained image fragments couples (f_i, f_j), (see section III). The following information must be supplied along with each couple (f_i, f_j):
θ_{ij}: the alignment angle by which fragment f_i must be rotated in order to be aligned with the matching contour segment of f_j.
S_{ij}: the mapping score of the couple.
M: amount of image fragments couples taken from $\mathcal{E}'$ used to initialize this algorithm.
Output
$\mathcal{I}$: set of reassembled images.

$\mathcal{I} \leftarrow \emptyset$;
$\mathcal{T} = \{(f_i, f_j) \in \mathcal{E}' : S_{ij}$ is one of the M highest similarity scores$\}$;
for each (f_i, f_j) *in* $\mathcal{T}$ **do**
  rotate f_i under θ_{ij} and align it with the matching contour segment of f_j ;
  reassemble a new image I from f_i and f_j ;
  $\mathcal{I} = \mathcal{I} \cup \{I\}$;
end for

repeat
  Take one partially reassembled image I from $\mathcal{I}$; {Let $I = \{f_1, f_2, \ldots, f_m\}$}
  Find an image fragment $f_i \notin I$ such that $(f_i, f_j) \in \mathcal{E}'$, and $S_{ij} = \{\max_{i|f_i \notin I}\{\max_{j|f_j \in I}\{S_{ij}\}\}\}$;
  $I' = \{f_j | f_j \in I \wedge (f_i, f_j) \in \mathcal{E}'\}$; {$I'$ is the set of image fragments that belong in the partially reconstructed image I and have a matching segment with fragment f_i. }
  Estimate $\phi_i^j, \forall f_j \in I'$;
  Let r be the cardinality of the estimated relative alignment angles, i.e. $\phi_i^l, l = \{1, \ldots, r\}$;
  Form sets $\mathcal{T}_l = \{f_j \in I' : \phi_i^{j_1} = \phi_i^{j_2} = \ldots = \phi_i^{j_n}\} \cup (I - I')$, where $l = \{1, \ldots, r\}$ and n is the size of each $\mathcal{T}_l$; {Each set $\mathcal{T}_l$ contains image fragments, for which fragment f_i has the same relative alignment angle, plus the rest fragments in I for which f_i has no matching contour segment with. }
  for all $\mathcal{T}_l, l = \{1, \ldots, r\}$ **do**
    Form a new image NI after rotating f_i under ϕ_i^l and aligning it with the corresponding matching contour segments of fragments in $\mathcal{T}_l$;
    $\mathcal{I} = \mathcal{I} \cup \{NI\}$;
  end for
  $\mathcal{I} = \mathcal{I} - \{I\}$; {remove old image I.}
until no fragment can be added in any reassembled image $I \in \mathcal{I}$

Fig. 5. Overall image reassembly step.

in the next step a new fragment will be involved in the update procedure of the input images. This iterative algorithm is finalized when no more image fragments are left to be inserted into the reassembled images. The steps of the algorithm are shown in Fig. 5.

The proposed method differs from other ones found in the literature, as it is based on the alignment angles. Thus, it is sensitive to errors in the relative alignment angle estimations. However, we overcome this drawback, by setting a threshold when comparing the relative alignment angles. We regard two relative alignment angles ϕ_1 and ϕ_2 to be equal if $|\phi_1 - \phi_2| < \epsilon$, where ϵ is a decision threshold. Of course, a backup approach is to add manual intervention to the system, and ask the user to select the correct alignment transformation.

Fig. 6. Scanned paper pieces at 300-dpi resolution.

TABLE I
RECALL PERCENTAGE OF THE CORRECT COUPLES OF ADJACENT IMAGE FRAGMENTS FOR $L = 15$

Histogram distance measure	Recall
d_{L_1}	45.48
d_{L_2}	46.53
d_{HI}	**53.25**
d_{SCH}	48.71

TABLE II
EFFECT OF COLOR QUANTIZATION METHODS IN CONTOUR SEGMENT MATCHING

Color Matching Method	False Negative Error	False Positive Error
KNN [30]	**20.18%**	**22.56%**
Mean Shift Algor. [29]	53.69%	53.68%
Gret. Macbeth Pal. [26]	51.36%	51.35%
Method presented in [5]	67.73%	60.78%

TABLE III
ERFORMANCE OF THE PROPOSED ICP VARIANTS IN CORRECTLY MATCHED CONTOUR SEGMENT ALIGNMENT

ICP variant	Performance %
ICP [9]	45.71
Trimmed ICP [11]	62.68
RICP [10]	64.36
ICPIF [32]	**77.12**

VII. IMAGE REASSEMBLY EXPERIMENTS

The performance of the proposed method was evaluated using 70 paper image prints. Each paper image print had size 25 cm × 20 cm and was torn into $N = 20$ paper pieces that were scanned at 300 dpi resolution, as shown in Fig. 6. We selected such a low scanning resolution in order to create a challenging image fragments dataset. Experiments are conducted in a 4 kernel PC with 4 GB RAM. The algorithm has been implemented in C++. The first step of the proposed method, as described in Section III,

TABLE IV
OVERALL IMAGE REASSEMBLY PERFORMANCE

Overall Assembly	% 10 fragm. perfor.	% 20 fragm. perfor.	10 fragm. comput. time	20 fragm. comput. time
No first stage	79.23	63.49	2.31 min	7.58 min
Including first stage	57.43	42.15	1.07 min	3.42 min
Human	-	-	5.64 min	17.37 min

is the discovery of similarly colored image fragments. We selected to retain for every image fragment the 15 most probable adjacent fragments ($L = 15$). In Table I, the mean recall percentage of the correct couples of adjacent image fragments for $L = 15$ is shown for various histogram similarity measures. These results are averaged over all the image fragments of the entire image fragments dataset. The d_{HI} measure exhibits the highest performance.

Whether executing (or not) the first step of the proposed method, the contour segment matching for every input couple of image fragments must be done. When the first step is executed, the matching contour segments are identified only for retained image couples, otherwise it is executed for all possible input fragment couples. We have experimented by combining the dynamic programming algorithm with several color quantization methods, described in Section IV. We have also evaluated the performance of the matching contour algorithm introduced in [5]. Each time, N_p random pixels were sampled from all fragments that constitute the overall image, in order to perform color quantization either by using the Kohonen neural nets or the Mean Shift algorithm. In both cases N_p was set equal to 0.25% of the total image fragment pixels. We have chosen Kohonen nets having $C = 50$ color clusters, while the learning procedure took 750 epochs. Nodes in the output layer of the neural network were organized under a random lattice. The spread parameter of Mean Shift algorithm [29] was set to 5.5, thus producing $C = 57$ color clusters. Following one from the previously described color quantization techniques, each contour pixel is represented by its color cluster label ($C = 24$ utilizing the Gretag Macbeth Checker). The parameters of the Smith Waterman algorithm were set to $e = 1$, $d = -0.5$ and $g = -0.5$. The contour matching results are shown in the last row of Table II. The second column refers to the percentage of unidentified matching contour pixels (false negative error rate), while the third column refers to the percentage of identified pixels that do not belong in true matching contour segments (false positive rate). The selection of the specific performance metrics is justified by the fact that most of the contour matching approaches, [5], [26], [29], lead to the extraction of "estimate" matching contour segments that overlap with the "correct" matching contour segments. The matching performance depends on the quality of the overlap, which is estimated by the ratio of unidentified and mis-identified contour pixels, or false negative and false positive rate. After this step, for each image fragment, one matching contour segment to $K = 10$ image fragments is retained, as described in Section IV. Color quantization using the Kohonen nets provides the best solution, identifying correctly more than 70% of the contour segment pixels, while minimizing the amount of misidentified contour pixels.

All matching contour segments identified in Section IV must be properly aligned, before the reassembly of the overall image. Such segments were produced by combining the dynamic programming algorithm with color quantization using the Kohonen nets. We have evaluated the performance of the proposed ICP variants by aligning the correct matching contour segments, i.e., the contour segments that really match in the reassembled image. These contour matchings may contain misidentified pixels or erroneously non-identified contour pixels. Each ICP variant was executed 1000 times. Implementation details concerning parameter selection, as well as computational complexity, can be found in [9]–[11] and [32] respectively. A complexity analysis of the ICP algorithm is presented in [33]. According to [33], the worst-case lower bound on the number of iterations performed by the ICP algorithm in order to converge is $\Omega(n/d)^{d+1}$, where n is the size of the input data points set and d is the dimensionality of the input data, while the smoothed [33] upper bound of the ICP algorithm is polynomial, independent of the dimensionality of the data. In Table III, the performance (percentage of visually correctly identified alignments over correctly matched contour segments) is shown. According to Table III, ICPIF outperforms the rest of the ICP variants.

The final step is the overall image reassembly. We evaluated the proposed method with/without employing the first step (the discovery of probably adjacent image fragments) and compared it with the manually (human) reassembly. The performance was estimated as follows. The performance of the method was estimated as follows. Let I be a manually reassembled image and $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$ be the set of automatically reassembled images generated by the algorithm. The reassembly is defined to be correct, when there is a $\widetilde{k} \in \{1, \ldots, n\}$ for which $I_{\widetilde{k}}$ is declared similar to I by a human observer.

In Table IV, the mean performance and computational time characteristics of the presented approach, regarding images fragmented into 10 and 20 pieces respectively, are shown. In both cases the image reassembly is performed employing exactly the same techniques in every step, while L and K parameters in the first case are set to 7 and 5, respectively. The first two columns show the percentage of correctly reassembled images, as described above, for the case of 10 fragments and 20 fragments, respectively. The last two columns show the respective mean computational time. The last row shows mean manually reassembly time efficiency (the performance of manually reassembly is not measured). Eight people of ages between 23 to 31 participated in this experiment. We have averaged the time needed for each person to correctly

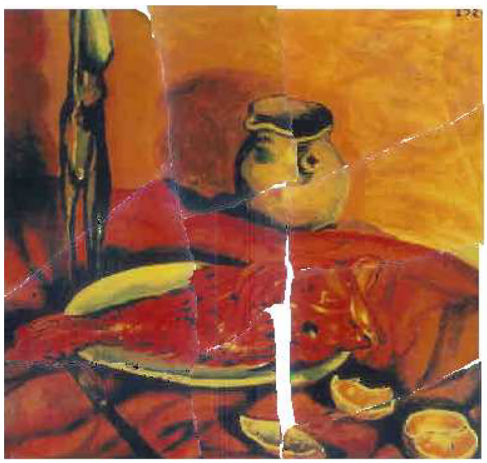

Fig. 7. Automatically reassembled image produced from the fragments of Fig. 6.

Fig. 8. New fragments created by aligning and assembling original fragments along their matching contour segments.

reassemble each fragmented image. As expected, the overall performance will be higher when the first stage is omitted. However, in this case the computation cost is higher, since both the second and the third step are executed for every couple of input image fragments. In the first case, where the the amount of fragments is somewhat small (10 fragments), the reassembly time efficiency is high in every case (with/without first step, manually reassembly). As the amount of fragments per input image increases, the time efficiency deviations become higher. Table IV shows that the choice whether to employ the first step on the reassembly process depends on the computational cost that is associated within the input image fragments. As the amount of image fragments increases, the higher overall performance of the variation that omits the first step gives way to the higher computational cost that is associated with it. Fig. 8 displays aligned couples of image fragments produced during the second and the third stages. The overall reassembled image is shown in Fig. 7. It can be seen that its reconstruction is nearly perfect. The white region in the middle of the reassembled image is due to missing pieces of paper that were not scanned.

VIII. Conclusions and Future Work

In this paper, we have introduced a novel integrated color-based image fragments reassembly method that consists of several distinct novel algorithms, which try to overcome inherent deficiencies, burdens and limitations. The overall assembly performance has proven to be particularly satisfactory. The experiments were conducted with 70 paper image prints. We plan to further improve the performance of the proposed method. Improvements can be done in each step of the proposed method. For example, the step of the discovery of spatial adjacent image fragments may be improved by employing not only color but textual or semantic features as well. Another possible extension is to utilize both the color and the shape of the fragments contours in order to perform matching. Furthermore, more sophisticated algorithms for faster contour matching, such as the one presented in [4], will be investigated. Another interesting idea would be to interchange the steps used in the proposed method, e.g., a shape alignment algorithm could provide a first set of coarse fragments shape matching results that would be further refined by employing color matching. Finally, the evaluation of the proposed method in a realistic fragmented image database, such as archaeological data, is worth exploring.

References

[1] B. Desbenoit, E. Galin, and S. Akkouche, "Modeling cracks and fractures," *Vis. Comput.*, vol. 21, pp. 717–726, 2005.
[2] M. Fornasiera and D. Toniolo, "Fast, robust and efficient 2D pattern recognition for re-assembling fragmented images," *Pattern Recognit.*, vol. 38, pp. 2074–2087, 2005.
[3] A. R. Willis and D. B. Cooper, "Computational reconstruction of ancient artifacts," *IEEE Signal Process. Mag.*, pp. 165–183, Jul. 2008.
[4] H. C. G. Leitao and J. Stolfi, "A multiscale method for the reassembly of two dimensional fragmented objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1239–1251, Sep. 2002.
[5] F. Amigoni, S. Gazzani, and S. Podico, "A method for reassembling fragments in image reconstruction," in *Proc. Int. Conf. Image Processing*, Sep. 2003, pp. 581–584.
[6] H. J. Wolfson, "On curve matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 5, pp. 483–489, May 1990.
[7] J. Salvi, C. Matabosch, D. Fofi, and J. Forest, "A review of recent range image registration methods with accuracy evaluation," *Image Vis. Comput.*, vol. 25, no. 5, pp. 578–596, 2007.
[8] H. Alt, L. Scharf, and S. Scholz, "Probabilistic matching of sets of polygonal curves," in *Proc. 22nd European Workshop on Computational Geometry (EWCG)*, Delphi, Greece, Mar. 2006, pp. 107–110.
[9] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image Vis. Comput.*, vol. 10, no. 3, pp. 145–155, 1992.
[10] E. Trucco, A. Fusiello, and V. Roberto, "Robust motion and correspondences of noisy 3D point sets with missing data," *Pattern Recognit. Lett.*, vol. 20, pp. 889–898, 1999.
[11] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The trimmed iterative closest point algorithm," in *Proc. 16th Int. Conf. Pattern Recognition*, 2002, vol. 3, pp. 545–548.
[12] T. Zinsser, H. Schnidt, and J. Niermann, "A refined icp algorithm for robust 3D correspondences estimation," in *Proc. Int. Conf. Image Processing (ICIP)*, 2003, pp. 695–698.
[13] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *J. Mol. Biol.*, vol. 147, pp. 195–197, 1981.
[14] B. Biswas, P. Bhowmick, and B. B. Bhattacharya, "Reconstruction of torn documents using contour maps," in *Proc. Int. Conf. Image Processing (ICIP)*, Sep. 2005, vol. 3, pp. 517–520.
[15] E. Justino, L. S. Oliveira, and C. Freitas, "Reconstructing shredded documents through feature matching," *Fores. Sci. Int.*, vol. 160, pp. 140–147, 2006.
[16] L. Zhu, Z. Zhou, and D. Hu, "Globally consistent reconstruction of ripped-up documents," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 1, pp. 1–13, Jan. 2008.
[17] M. S. Sagiroglu and A. Ercil, "A texture based matching approach for automated assembly of puzzles," in *Proc. 18th Int. Conf. Pattern Recognition (ICPR)*, 2006, vol. 3, pp. 1036–1041.
[18] A. Kalvin, E. Schonberg, J. Schwartz, and M. Sharir, "Two dimensional model based boundary matching using footprints," *Int. J. Robot. Res.*, vol. 5, no. 4, pp. 38–55, 1986.
[19] D. Goldberg, C. Malon, and M. Bern, "A global approach to automatic solution of jigsaw puzzles," *Computat. Geom.: Theory Appl.*, vol. 28, pp. 165–174, 2004.
[20] W. Kong and B. B. Kimia, "On solving 2D and 3D puzzles using curve matching," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, vol. 2, pp. 583–590.

[21] G. Papaioannou, E. A. Karabassi, and T. Theoharis, "Reconstruction of three-dimensional objects through matching of their parts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 114–124, Jan. 2002.

[22] S. Andrews and D. H. Laidlaw, "Toward a framework for assembling broken pottery vessels," in *Proc. 18th American Conf. Artificial Intelligence*, 2002, pp. 945–946, American Association for Artificial Intelligence (AAAI).

[23] Y. Lu, H. Gardner, H. Jin, N. Liu, R. Hawkins, and I. Farrington, "Interactive reconstruction of archaeological fragments in a collaborative environment," in *Proc. 9th Biennial Conf. Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA '07)*, Washington, DC, 2007, pp. 23–29, IEEE Computer Society.

[24] D. Koller, J. Trimble, T. Najbjerg, N. Gelfand, and M. Levoy, "Fragments of the city: Stanford's digital forma urbis romae project," *J. Roman Arch. Suppl.*, vol. 61, pp. 237–252, March 2006.

[25] H. C. G. Leitao and J. Stolfi, "Measuring the information content of fracture lines," *Int. J. Comput. Vis.*, vol. 65, no. 3, pp. 163–174, 2005.

[26] M. A. Gavrielides, E. Sikudova, and I. Pitas, "Color based descriptors for image fingerprinting," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 740–748, Aug. 2006.

[27] L. Cinque, G. Ciocca, S. Levialdi, A. Pellicano, and R. Schettini, "Color based image retrieval using spatial chromatic histograms," *Image Vis. Comput.*, vol. 19, pp. 786–979, 2001.

[28] R. C. Gonzalez and R. E. Woods*, Digital Image Processing*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 2006.

[29] Y. Cheng, "Mean-shift mode seeking and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, Aug. 1995.

[30] G. A. Carpenter and S. Grossberg*, Pattern Recognition by Self-Organizing Neural Networks*. Cambridge, MA: MIT Press, 1991.

[31] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc . 3rd Int. Conf. 3D Digital Imaging and Modeling (3DIM)*, Jun. 2001, pp. 145–152.

[32] G. C. Sharp, S. W. Lee, and D. K. Wehe, "Icp registration using invariant features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 90–102, Jan. 2002.

[33] D. Arthur and S. Vassilvitskii, "Worst-case and smoothed analysis of the icp algorithm, with an application to the k-means method," in *Proc. 47th Annu. IEEE Symp. Foundations of Computer Science (FOCS)*, 2006, pp. 153–164.

Efthymia Tsamoura received the Diploma degree in computer science with honors (9.15/10) in 2007. She is currently pursuign the Ph.D. degree in the Computer Science Department, Aristotle University of Thessaloniki (AUTH).

During October 2007–June 2008, she was a research assistant in the Multimedia Knowledge Laboratory, Informatics and Telematics Institute (ITI), where she participated in the European project Vidi-Video. Her research interests lie in the field of distributed query optimization and grid computing, as well as image and video analysis processing. She was awarded from the State Scholarship Foundation of Greece for exceptional performance during the academic years 2003-2004, 2004-2005, 2005-2006, and 2006-2007, while this work was conducted during her undergraduate thesis

Ioannis Pitas (S'83–M'84–SM'94–F'07) received the Diploma of Electrical Engineering in 1980 and the Ph.D. degree in electrical engineering in 1985 both from the Aristotle University of Thessaloniki, Greece.

Since 1994, he has been a Professor in the Department of Informatics, Aristotle University of Thessaloniki. From 1980 to 1993, he served as a Scientific Assistant, Lecturer, Assistant Professor, and Associate Professor in the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki. He has served as a Visiting Research Associate or Visiting Assistant Professor at several universities. He has published 153 journal papers, 400 conference papers, and contributed to 22 books in his areas of interest and edited or coauthored another five books. He has also been an invited speaker and/or member of the program committee of several scientific conferences and workshops.

Dr. Pitas has served as Associate Editor or Co-Editor of four international journals and as General or Technical Chair of three international conferences. His current interests are in the areas of digital image and video processing and analysis, multidimensional signal processing, watermarking, and computer vision.

Automatic Color Based Reassembly of Fragmented Images and Paintings

Efthymia Tsamoura and Ioannis Pitas, *Fellow, IEEE*

***Abstract*—The problem of reassembling image fragments arises in many scientific fields, such as forensics and archaeology. In the field of archaeology, the pictorial excavation findings are almost always in the form of painting fragments. The manual execution of this task is very difficult, as it requires great amount of time, skill and effort. Thus, the automation of such a work is very important and can lead to faster, more efficient, painting reassembly and to a significant reduction in the human effort involved. In this paper, an integrated method for automatic color based 2-D image fragment reassembly is presented. The proposed 2-D reassembly technique is divided into four steps. Initially, the image fragments which are probably spatially adjacent, are identified utilizing techniques employed in content based image retrieval systems. The second operation is to identify the matching contour segments for every retained couple of image fragments, via a dynamic programming technique. The next step is to identify the optimal transformation in order to align the matching contour segments. Many registration techniques have been evaluated to this end. Finally, the overall image is reassembled from its properly aligned fragments. This is achieved via a novel algorithm, which exploits the alignment angles found during the previous step. In each stage, the most robust algorithms having the best performance are investigated and their results are fed to the next step. We have experimented with the proposed method using digitally scanned images of actual torn pieces of paper image prints and we produced very satisfactory reassembly results.**

I. Introduction

THE problem of reassembling image fragments arises in many scientific fields, such as forensics and archaeology. In the field of archaeology, the pictorial excavation findings are almost always in the form of painting fragments. For example, they can be fragments of painted pottery, murals, or mosaics, which must be assembled to form the original painting. A related aspect of the problem is the development of a generative model for cracks and fractures as proposed in [1]. More rarely, there are cases where the form of the original object (e.g., mosaic) is known but has to be reassembled because of a destruction. In [2], a pattern matching algorithm for the comparison of digital images is implemented using discrete circular harmonic expansions. The manual execution of the above tasks is very difficult, as it requires great amount of time, skill and effort. Thus, the automation of such a work is very important and can lead to faster, more efficient, painting reassembly and to a significant reduction in the human effort involved.

In our work, the automated reassembly of images from fragments follows a four step model, similar to the one presented in [3] for 3-D object reconstruction. The first step of our approach is the identification of probable adjacent image fragments, in order to reduce the computational burden of the subsequent steps. There, several color-based techniques are employed. However, as stated in a following section, this step can be omitted sacrificing the lower running time of the image reassembly procedure, for higher performance.

The second operation is the identification of the matching contour segments of the image fragments. The corresponding step employs a neural network based color quantization approach for the representation of the image contours, followed by a dynamic programming technique that identifies their matching image contour segments. In [4], an analogous algorithm is presented, for the discovery of the matching contour segments of 2-D fragmented objects. However, the fragment contour comparison is based on the shape of the input 2-D object fragments, while a dynamic programming technique is employed in order to identify their matching segments. Aminogi *et al.* [5], presented an algorithm based on both the shape and the color characteristics of the input 2-D image fragments contours. There, one contour pixel sequence is overlaid on another one and, for each such "placement", the curvature and color differences of the corresponding contour pixels are estimated. If their total sum is less than a user defined threshold, the contour segments are considered to match. In the experimental results section, the latter approach is compared with the one proposed here. The results show that the latter has better performance.

Once the matching contour segments are identified, a third operation takes place. Here, the geometrical transformation, which best aligns two fragment contours along their matching segments, is found. Several such approaches exist, e.g., [6]–[8], however, most of them are either not robust to matching errors and/or have high computational complexity. A very popular registration technique is the *Iterative Closest Point (ICP)* method [9]. New modified versions of the ICP have been proposed, which are robust to noise [10]–[12]. These approaches limit the effects of noise on the registration performance by outlier trimming based on a least squares distance criterion. In our work, the above ICP variants were experimentally evaluated and the best one among them is selected for integration with the proposed four stage image reassembly algorithm.

The last step in solving the fragment reassembly problem is the reassembly of the overall image from its constituent

Manuscript received March 10, 2009; revised September 29, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Pier-Luigi Dragotti.

The authors are with the Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece (e-mail: etsamour@csd.auth.gr; tsamoura@delab.csd.auth.gr; pitas@aiia.csd.auth.gr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2009.2035840

fragments. Here, a novel algorithm is proposed. The novelty of our algorithm lies on the fact that employs both the contour matching results and the alignement angles of the fragments, found during the second and the third step, respectively. The majority of the algorithms in this step, as will be presented in the related work section, utilize only criteria that are based on the contour matchings, e.g., the length of the matching. It is clear that it is essential that each step of the algorithm feeds the next one with correct results, otherwise the image reassembly may contain errors, or may even fail completely. Our goal is to investigate and propose the most robust techniques in order to produce accurate results at each intermediate step. To summarize, the main steps of the proposed method are shown in Fig. 1 and can be described as follows.

1) **Discovery of spatial adjacent image fragments**. The identification of probable spatially adjacent image fragments is done by utilizing techniques that are widely employed in content based image retrieval (CBIR) systems. The purpose of this step is to reduce the computational burden of the steps that follow. Once this step is finished, we select to retain, for every image fragment, a list of the most probable adjacent fragments.
2) **Discovery of matching contour segments of adjacent image fragments**. A novel approach based on the Smith-Waterman algorithm [13] is employed in order to match the colors appearing in the contours of adjacent image fragments. Various color similarity criteria are being evaluated. Based on such similarity criteria, for each image fragment, one matching contour segment with K other image fragments is retained.
3) **Image fragments contour alignment**. The purpose of this step is to find the appropriate geometrical transformation of one fragment relative to its adjacent one, in order to align them along their matching contour segments. Many variants of the ICP algorithm are employed and evaluated to this end.
4) **Overall image assembly**. Once the matching contour segments of couples of input image fragments are identified and properly aligned, the remaining step is the reassembly of the overall image. Since the criteria that are based on the contour matchings do not suffice for the overall image reassembly, a novel feature, namely the alignment angles found during step 3, is introduced.

In this paper, an integrated method for automatic color based 2-D image fragment reassembly is presented. This paper is organized as follows. Section II discusses the related work. Section III describes the discovery of the spatial adjacent fragments. Section IV presents the identification of the matching contour segments of the spatial adjacent image fragments, while Section V describes the derivation of the optimal geometrical transformation that aligns contours along their matching segments. Section VI presents the overall image assembly algorithm. Experimental results are presented in Section VII, while conclusions are drawn in Section VIII.

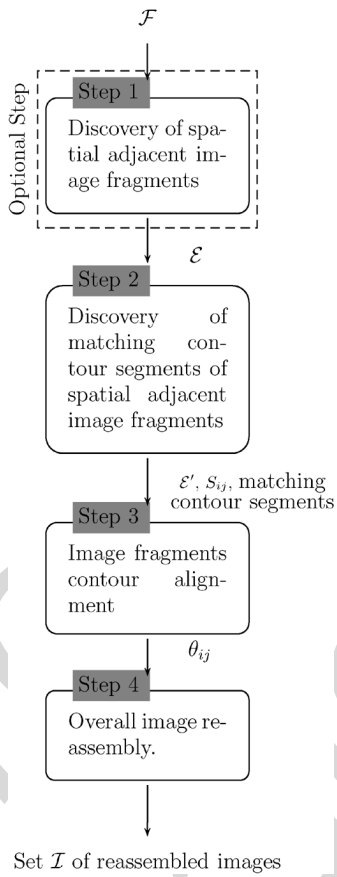

Fig. 1. Overall image reassembly approach.

II. Related Work

A. Two-Dimensional Paper Document Reassembly

Similar to 2-D image fragment reassembly, in paper document reassembly, torn paper fragments must be assembled to form the image of an entire page of a paper document. Work on this area was conducted in [14]–[16]. The above employ shape representations of the paper fragments, in order to reassembly the original documents. In [15], polygonal approximation is initially applied to reduce the complexity of the paper fragment contours and geometrical features are extracted from these polygonal curves. Then, a method based on [4] is used to assemble the entire document from its constituent paper fragments. In [16], shape features, namely turning functions, are estimated from every fracture contour and are utilized to discover matching contour segments. After that, each matching is assigned a confidence score. The alignment transformation of the fragments is simultaneously found during matching. After the discovery of matching contour segments, the final reassembly step performs two actions, namely matching relaxation and fragments merging. During matching relaxation, every pair of aligned input fragments is checked for overlap along their matching contour segments. If they overlap, this matching is discarded. Otherwise, the neighboring fragments of this pair are identified. A score, called support, is assigned to the neighborhood of each pair of non-overlapping fragments. This score increases as the number of neighboring fragments as

well as the matching confidence assigned to pairs of fragments increase. The fragments that have neighborhoods with maximum support are merged into new fragments and the whole procedure starts again, i.e., the matching contour segments are identified for all pairs of fragments, and so on.

B. Two-Dimensional Puzzle Assembly

Many methods were also proposed for the 2-D puzzle reassembly problem. In [17], color and textural features of the puzzle pieces are utilized. The matching and alignment of puzzle pieces is carried out using an FFT-based image registration technique. In [18], the puzzle reassembly process consists of two steps; frame and interior assembly. A traveling salesman problem (TSP) is formulated for frame assembly, while backtracking and branch&bound techniques are employed for interior assembly. The puzzle pieces are matched employing the L_2 distance of their contours curves. An improvement of this method is presented in [19]. In [20], the overall puzzle assembly is done using a Best-First procedure. There, two criteria are utilized to sort matching contour segments. The first one is the residual error of corresponding contour pixels after the discovery of the optimal geometrical transformation, while the second criterion is the arc-length of the matching contour segments. It is clear that the problem of 2-D puzzle assembly does not meet the major difficulty of the image and object reconstruction problems; that is the missing or highly damaged image (or) object fragments. Thus, in general, the algorithms proposed in this field would be inadequate to solve such problems.

C. Three-Dimensional Object Reconstruction

Regarding 3-D object reconstruction, an automatic method for matching and alignment of 3-D, free-form archaeological fragments is proposed in [21]. The input fragments are not pre-processed. The matching is performed utilizing only the 3-D points of the whole surfaces of the objects. The output matching-alignment minimizes the distance between the 3-D surface points of the two fragments. Andrews *et al.* [22] propose an automatic method for the reconstruction of pairs or triplets of 3-D symmetric archaeological fragments. The matching is found through a two-phase method. During the first phase several matchings-alignments are estimated for every pair of fragments. The 3-D points of the fracture curves in the outer and inner surface of the fragments as well as the axis of rotation of the fragments are utilized to this end. In the second phase, these matchings are refined using the quasi-Newton method, and evaluated according to several criteria namely the angle formed by the fragments rotation axes, the perpendicular distance between the rotation axes and the distance of the matched fracture curves points. Eventually, one matching is retained for every pair of fragments. Finally, in the overall object reconstruction step, a greedy merge strategy selects pairs of fragments to form triplets In [23], a human-supervised collaborative reconstruction system is described. The aim of the system is to propose a potential matching between any pair of input fragments. The matching is found by utilizing shape features (curvature and torsion) estimated from all 3-D points in the fracture curves of the fragments. The shape similarity of the fracture curves is ranked with a cyclic distance algorithm. Each matching defines correspondences between 3-D points in the fracture curves. Then the users select to merge or not the proposed fragments. The fragments alignment is performed interactively (in a VRML environment) by the users. The object reconstruction procedure follows the merge-update paradigm. Significant work on this area is also done in the Digital Forma Urbis Romae project [24]. The latter seeks to reconstruct a giant marble map of ancient Rome dating back to 200 AD, while only a small portion of fragments of the original map still exist. The reconstruction process in [24] is not automatic and is guided by expert users. Information and details about other 3-D object reconstruction methods can be found in [3].

Finally, in [25], a method is introduced to measure the average amount of information contained in a 2-D ceramic tile fragment contour, in terms of its curvature. This parameter shows how many false contour matches are expected to be found among a given set of ceramic tile fragments.

III. Discovery of Spatial Adjacent Image Fragments

The purpose of this step is the spatial adjacent image fragments identification by using their probable high color similarity. We have utilized techniques that are widely employed in CBIR systems in order to identify these similarities [26]. In this section, we briefly mention the descriptors and the measures we have experimented with.

Color quantization can be based on a commercial color palette, e.g., the Gretag Macbeth Color Checker [26]. The Macbeth Palette can be used to evaluate color reproduction systems and it consists of 24 different colors that are scientifically chosen to represent a variety of naturally occurring colors. Color quantization is used to find the normalized quantized color image histograms, which can be used for color image retrieval. We have also experimented with the Spatial Chromatic Histogram [27], which provides information both of color presence and color spatial distribution. The Spatial Chromatic Histogram S_I of image I having C quantized colors is given by $S_I(i) = (h(i), \mathbf{b}(i), \sigma(i)), i = \{1, \ldots, C\}$. In the above equation h denote the normalized color histogram, i.e., $h(i)$ is defined as the number of pixels having color i divided by the total number of pixels, $\mathbf{b}(i)$ is a 2-D vector expressing the center of mass and $\sigma(i)$ is the standard deviation of the i^{th} color label, respectively [27]. We have used modified versions of the L_1, L_2 and Histogram Intersection measures and scaled them to the range [0, 1], with 1 denoting a perfect similarity. In the following equations h_1 and h_2 denote the normalized color histograms extracted from images I_1 and I_2, respectively. The utilized matching measures are the following.

1) Scaled L_1 norm

$$d_{L_1}(h_1, h_2) = 1 - 0.5 \sum_{i=1}^{C} |h_1(i) - h_2(i)| . \quad (1)$$

2) Scaled L_2 norm

$$d_{L_2}(h_1, h_2) = 1 - \frac{1}{\sqrt{2}} \sum_{i=1}^{C} (h_1(i) - h_2(i))^2 . \quad (2)$$

3) Scaled Histogram Intersection

Inputs
$\mathcal{F}$: set of N image fragments
L: the size of most chromatically similar fragments per input image fragment.
Outputs
$\mathcal{E}$: set of image fragments couples.

1: $\mathcal{S} \leftarrow \emptyset$;{$\mathcal{S}$ is a list of spatial chromatic histograms.}
2: **for all** $f \in \mathcal{F}$ **do**
3: quantize f using Gretag Macbeth Color Checker ;
4: estimate the spatial chromatic histogram of image fragment f, S_f, according to [27];
5: append S_f to $\mathcal{S}$;
6: **end for**
7: $\mathcal{E} \leftarrow \emptyset$;
8: **for** $i = 1$ to $N - 1$ **do**
9: **for** $j = i + 1$ to N **do**
10: $m[j - i - 1] = d(S_{f_i}, S_{f_j})$; {m is a one dimensional real matrix. d is one from (1) to (4).}
11: **end for**
12: sort m in descending order ; $thr = m[L]$;
13: $\mathcal{E} = \mathcal{E} \cup \{(f_i, f_j) | d(S_{f_i}, S_{f_j}) \geq thr\}$;
14: **end for**

Fig. 2. First step of the proposed 2-D image reassembly approach.

$$d_{HI}(h_1, h_2) = \sum_{i=1}^{C} \min\left(h_1(i), h_2(i)\right)\left(1 - |h_1(i) - h_2(i)|\right). \quad (3)$$

4) Spatial Chromatic distance [27]

$$d_{SC}(I_1, I_2) = \sum_{i=1}^{C} \min\left(h_1(i), h_2(i)\right) \times \left(\frac{\sqrt{2} - \|b_1(i) - b_2(i)\|^2}{\sqrt{2}} + \frac{\min\left(\sigma_1(i), \sigma_2(i)\right)}{\max\left(\sigma_1(i), \sigma_2(i)\right)}\right). \quad (4)$$

Once this step is finished, we select to retain, for every image fragment, a list of the L most chromatically similar fragments. The first step is shown in Fig. 2. It must be emphasized that this step is not prerequisite for the correct reassembly of an image and can be easily skipped, if the number of image fragments is rather small. The purpose of this task is to reduce the computational burden of the steps that follow.

IV. Discovery of Matching Contour Segments of Adjacent Image Fragments

Let us suppose that we have a set of image fragments where we have to find their matching contour segments. This set may contain the image fragment pairs identified by the previous step or all the image fragment pairs (if the first step has been skipped). This operation is performed on image fragment pairs. In this section, we present a novel algorithm for identifying matching contour segments of couples of input fragments. Our approach to fragment contour matching is based exclusively on information regarding the color of their contours.

In order to avoid comparing directly contour pixel colors that may contain noise, a color quantization preprocessing step is utilized, which takes pixel samples from the contours of all image fragments. Many color quantization methods exist [28]. For example, quantization can be done using ready made palettes, such as the Gretagh Macbeth palette [26]. The Mean Shift algorithm [29] is a popular solution for color image quantization. Its advantage is that the number of the color clusters is automatically defined. However, the Mean Shift algorithm exhibits a severe local maxima sensitivity. This drawback may result to significant misplacements of color cluster centers, as will be shown in the experimental results section.

In the following, we employ Kohonen neural networks (KNNs) [30] for color quantization purposes. KNNs belong in the class of unsupervised neural networks. They can cluster input vectors without any external information, following an iterative procedure based on competitive learning [30]. KNNs consist of two node layers; the input and the output layer. In the former, the number of nodes equals the dimension of input vectors, while in the latter the number of nodes equals the amount of produced clusters. The nodes in the output layer are organized by means of a lattice [30]. In KNNs, each node in the input layer $\mathbf{s}_i$ has a connection w_{ik} with every node $\mathbf{c}_k$ in the output layer. For a network with n input nodes, the weight vector $\mathbf{w}_j = [w_{1k}, w_{2k}, \ldots, w_{nk}]$ ending at an output node c_k, is the center of a cluster. In KNNs, given an input vector, the output node with the highest response (winning node) for that as well as all "neighboring" nodes that belong to an area around it, update their weight vectors.

In the following, we shall describe the KNN variant employed to perform color quantization of image fragments. First, a random number of N_p pixels is sampled from the input image fragments and mapped to La^*b^* color space. The number of sampled N_p pixels that are required to successfully quantize the color space, is a minimal portion of the total image fragments' pixels, as it will be demonstrated in the experimental results section. After that a $[3 \times C]$ KNN is defined, where 3 corresponds to the dimension of the input La^*b^* space and C to the predefined number of color clusters. Let $\mathbf{x} = [x_1, x_2, x_3]$, be one of the N_p sampled pixels, after mapping to the La^*b^* color space. The following learning procedure is iteratively applied.

1) A winning node c_j is selected, i.e., output node whose weight vector $\mathbf{w}_j$ has the highest similarity with the input vector $\mathbf{x}$, than any other output node c_k

$$\|\mathbf{x} - \mathbf{w}_j\| = \min_{\forall c_k} \{\|\mathbf{x} - \mathbf{w}_k\|\}. \quad (5)$$

Euclidean distance is utilized in our experiments as the similarity criterion.

2) A neighborhood function is utilized to estimate the weight vectors updates. Thus, the weight vector $\mathbf{w}_k$ of an output node c_k is updated under

$$\Delta_{\mathbf{w}_k} = \gamma \Omega_{c_j}(c_k)\left(\|\mathbf{x} - \mathbf{w}_k\|\right) \quad (6)$$

where $\Omega_{c_j}(c_k) = e^{(-\|\mathbf{p}_k - \mathbf{p}_j\|^2)/2\sigma^2}$, γ is the learning parameter $(0 < \gamma < 1)$, σ denotes the spread of the "neighborhood" around the winning node c_k and $\mathbf{p}_k$, $\mathbf{p}_j$ correspond to places inside the lattice of an output node c_k and the winning c_j, respectively.

For better convergence, the learning parameter γ and the standard deviation σ of the neighborhood function gradually decrease after an iteration (epoch in neural networks terminology).

The learning procedure stops, when several criteria are met, e.g., when a predefined number of epochs have finished, or when the changes $\Delta_{\mathbf{w}_k}$ of clusters' centers are very small. After training the network, the weight vector of every output node corresponds to a cluster center. Details about the above parameters, as well as a comparison of the above color quantization techniques can be found in Section VII. Following one of the previously described approaches, the color of every contour pixel is represented by its color cluster label c_j, $j = \{1, \ldots, C\}$.

Let $U = [u_i]_{i=1}^{n}$ and $V = [v_j]_{j=1}^{m}$ be two pixel sequences that follow the entire contours of two different image fragments. Once the color clustering has been achieved for the input image fragments, sequences $[a_i]_{i=1}^{n}$ and $[b_j]_{j=1}^{m}$ are assigned to U and V respectively. a_i and b_j are two color cluster labels, where the former is estimated at pixel u_i and the latter at pixel v_j. Then u_i and v_j have the same color, if $a_i = b_j$. We define a similarity function F of the two contour pixels as follows:

$$F_{u_i,v_j}[a_i, b_j] = \begin{cases} e > 0, & a_i = b_j \\ d < 0, & a_i \neq b_j. \end{cases} \tag{7}$$

Given U and V and their label lists $[a_i]_{i=1}^{n}$ and $[b_j]_{j=1}^{m}$, we search for the contour pixel mapping function $\mathbf{\Phi}$, such that:

- for every $\Phi[u_i] = v_k$ and $\Phi[u_{i+1}] = v_l$, $k \leq l \leq m$;
- $\Phi[u_i] \neq \emptyset$.

The first condition means that more than one contour pixels in U can be mapped to the same contour pixel in V. However, the former contour pixels must be strictly consecutive. This condition guarantees that no "folding" of the U contour pixel sequence, in order to match it with the corresponding matching contour segment in V, will take place. The second condition ensures that every contour pixel in U is mapped to a contour pixel in V. The algorithm that is used to identify the mapping function $\mathbf{\Phi}$ is a variant of the Smith Waterman dynamic programming algorithm [13], which is a local sequence matching algorithm: given two input sequences, it can identify the mapping function between their segments. Each mapping between segments of input sequences is assigned with a score $S > 0$; as the score S increases, the color similarity becomes higher. A similarity $n \times m$ matrix $\mathbf{H}$ is set up, where the i^{th} row of matrix $\mathbf{H}$ corresponds to u_i, while the j^{th} column corresponds to v_j. $H_{i,j}$ is the best mapping score S of the pair of every sub-sequences ending at u_i and v_j pixels. The algorithm gradually fills matrix $\mathbf{H}$ and forms the mapping function $\mathbf{\Phi}$. During filling, each matrix cell is assigned with the highest possible value, as our purpose is to maximize the mapping score S.

As in a typical dynamic programming algorithm, the solution to an instance of the problem is given in terms of solutions to its smaller sub-instances. Thus

$$H_{i,j} = \begin{cases} H_{i-1,j-1} + F_{u_i,v_j}[a_i, b_j] \\ H_{i,j-1} + g \\ H_{i-1,j} + g \end{cases} \tag{8}$$

where $g < 0$. $H_{i,j}$ is assigned the maximum value of the right hand terms in (8). Let $\mathbf{\Phi}$ be the estimated mapping of $[u_k]_{k=1}^{i-1}$ and $[v_l]_{l=1}^{j-1}$ subsequences. If $H_{i,j} = H_{i-1,j-1} + F_{u_i,v_j}[a_i, b_j]$, then the relation $\Phi[u_i] = v_j$ is appended to $\mathbf{\Phi}$. On the other hand, if $H_{i,j} = H_{i,j-1} + g$ or $H_{i,j} = H_{i-1,j} + g$ is selected, then $\Phi[u_i] = v_{j-1}$ or $\Phi[u_{i-1}] = v_j$ is appended to $\mathbf{\Phi}$. A *gap* is formed, when either the second or the third case occurs in (8). Generally, a gap is formed when one or more contour pixels of the first fragment are mapped to the same contour pixel of another fragment. The percentage of mapping gaps is a measure of the dissimilarity between the input contour segments; a high gap percentage reveals contours with many dissimilarities. The parameter g is negative, in order to penalize mappings with many gaps. When this stage is completed, we identify an area in $\mathbf{H}$ with high similarity values H_{ij}. Diagonal patterns are preferred, since they correspond to mappings without many gaps. Let H_{e_1,e_2} and H_{s_1,s_2} be the lowest right and highest left borders of this area. In the implemented experimental modification, H_{e_1,e_2} is the maximum value of $\mathbf{H}$, while $H_{s_1,s_2} = 0$. Then we select the mapping starting from (s_1, s_2) and ending at (e_1, e_2), which is formed during filling $\mathbf{H}$. The Smith Waterman algorithm steps are shown in Fig. 4.

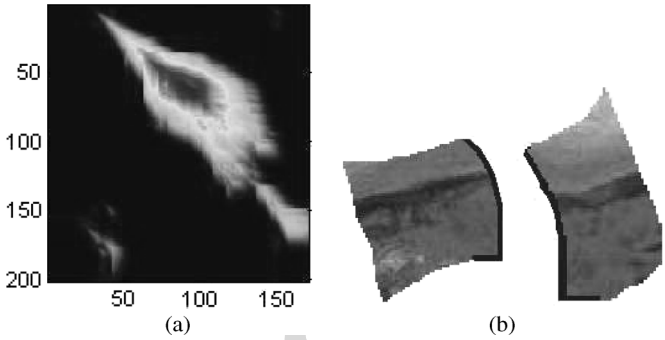

Fig. 3. (a) Color similarity matrix. (b) Matching based on color information. Common contour segments are denoted by blue.

A basic question is how to select e, d and g. These parameters are strongly correlated with the desired characteristics of the contour matching. Specifically, if a matching without many gaps is desirable, then a high absolute value is selected for g. Otherwise, the value of d must be close to g. The complexity of the proposed method for a pair of fragments is $O(nm)$, where n and m are the lengths of the two pixel contour sequences. For an image that is partitioned among into N fragments, the worst case computational complexity scenario (omitting the first step) is $O(N^2\widetilde{n}^2)$, where $\widetilde{n}$ is the average contour length of input image fragments. Such a similarity matrix is shown in Fig. 3(a). The horizontal axis of the matrix corresponds to the quantized contour pixels color sequence of the first image, while the columns correspond to the quantized contour pixels color sequence of the second image. High values correspond to contour segments with a high degree of similarity (deep red colored areas). The correctly matched color contour sequences are shown in Fig. 3(b).

After completing the second step, for each image fragment f, one matching contour segment with $K(0 \leq K \leq L)$ other image fragments is retained, producing the set of *true* adjacent image fragments couples $\{(f, f_i), i = 1, \ldots, K\}$.

V. Image Fragments Contour Alignment

In this section, we investigate methods for finding the best geometrical transformation that aligns fragment contours along

Inputs
$U = [u_i]_{i=1}^{n}$: contour pixel sequences of fragment f_p
$V = [v_j]_{j=1}^{m}$: contour pixel sequences of fragment f_r
$[a_i]_{i=1}^{n}$: color cluster label sequences of U
$[b_j]_{j=1}^{m}$: color cluster label sequences of V
parameters of the Smith-Waterman algorithm e, d, g: $e > 0$, $d < 0$, $g < 0$
Output
$\mathbf{\Phi}$: a mapping function between $[u_i]_{i=1}^{n}$ and $[v_j]_{j=1}^{m}$.
S_{pr}: the mapping score of $\mathbf{\Phi}$.

{Initialize $\mathbf{H}$ }
for $i = 1$ to n **do**
  for $j = 1$ to m **do**
    $H_{i,j} = F_{u_i,v_j}(a_i, b_j)$;
  end for
end for

for $i = 1$ to n **do**
  for $j = 1$ to m **do**
    $H_{i,j} = \max\{H_{i-1,j-1} + F_{u_i,v_j}(a_i, b_j), H_{i-1,j} + g, H_{i,j-1} + g, 0\}$
    {The zero value in is used in order to prevent **H** entries from taking negative values.}
  end for
end for

Select an area in matrix $\mathbf{H}$. Let H_{e_1,e_2} and H_{s_1,s_2} be the lowest right and highest left borders of this area;
$S_{pr} = H_{e_1,e_2}$;
$i = e_1$; $j = e_2$;
while $\{i \geq s_1 \wedge j \geq s_2\}$ **do**
  $index = \max\{H_{i-1,j-1}, H_{i-1,j}, H_{i,j-1}\}$;
  {$index$ corresponds to the place of the maximum element in $\{H_{i-1,j-1}, H_{i-1,j}, H_{i,j-1}\}$, i.e. $index \in \{1, 2, 3\}$}
  if $index = 1$ **then**
    $\Phi[u_i] = v_j$; $i = i - 1$; $j = j - 1$;
  else if $index = 2$ **then**
    $\Phi[u_{i-1}] = v_j$; $i = i - 1$;
  else
    $\Phi[u_i] = v_{j-1}$; $j = j - 1$;
  end if
end while

Fig. 4. Second step of the proposed reassembly approach.

their matching segments. All matching contour segments identified during the previous steps must be properly aligned, before the reassembly of the overall image. The disadvantages of the alignment algorithms found in the literature (as discussed in the introduction) have led us to use variants of the ICP algorithm [9] for contour alignment. The ICP algorithm generally starts with two point sets (contour segments in our case) and an initial guess of their relative rigid body geometrical transformation. It then refines the transformation parameters, by iteratively generating pairs of point correspondences and by minimizing an error metric. Given two curves $\mathcal{P} = \{\mathbf{p_1}, \ldots, \mathbf{p_{N_p}}\}$ and $\mathcal{M} = \{\mathbf{m_1}, \ldots, \mathbf{m_{N_m}}\}$, the following steps are performed.

1) Compute the subset of pairs of closest points

$$\mathcal{Y} = \{(\mathbf{p}_i, \mathbf{m}_j) | \mathbf{p}_i \in \mathcal{P}, \mathbf{m}_j \in \mathcal{M} \quad \mathbf{m}_j \text{ is the closest point to } \mathbf{p}_i\}. \quad (9)$$

2) Compute a Least Squares estimate of the geometrical transformation mapping $\mathcal{P}$ onto $\mathcal{M}$

$$(\mathbf{R}, \mathbf{t}) = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{|\mathcal{Y}|} \|\mathbf{m}_i - \mathbf{R}\mathbf{p}_i - \mathbf{t}\|^2 \quad (10)$$

where $(\mathbf{p}_i, \mathbf{m}_i) \in \mathcal{Y}$.

3) Apply the transformation to the $\mathcal{P}$ data points

$$\mathcal{P} = \mathbf{R}\mathcal{P} + t. \quad (11)$$

4) If stopping criterion is satisfied exit; else, go to step 1.

The original form of the ICP algorithm is not robust to outliers, since it does not trim noisy data. Such outliers, coming from errors during the contour segment matching, create serious problems to alignment, if not properly handled. In order to obtain ICP versions that are robust to outliers, many ICP variants have been proposed in the literature. In [31], these variants have been classified into six categories. In order to develop a robust contour alignment procedure, we have studied three popular robust versions of ICP and have evaluated their performance.

A. Robust ICP (RICP)

The first variant that we studied is RICP [10]. It trims outliers in the second step (10) of the ICP and computes an L_2 error norm, which is, hopefully, free from outlier influence. Outliers are rejected, according to the following procedure. Let $\mathbf{p_i}$ be points from the first curve and $\mathbf{m_j}$ be their corresponding closest points on the second curve. Select randomly two points from each of the 2-D input curves and execute their registration, by solving the linear system

$$\begin{pmatrix} \mathbf{m}_{1_x} & \mathbf{m}_{2_x} \\ \mathbf{m}_{1_y} & \mathbf{m}_{2_y} \end{pmatrix} = \mathbf{R} \begin{pmatrix} \mathbf{p}_{1_x} & \mathbf{p}_{2_x} \\ \mathbf{p}_{1_y} & \mathbf{p}_{2_y} \end{pmatrix} \quad (12)$$

where $\mathbf{R}$ is a rotation matrix. Compute the residuals $\mathbf{s} = \mathbf{R}\mathbf{p} - \mathbf{m}$. The above procedure is repeated for a sufficient number of iteration [10]. Once all potential registrations are evaluated, the one that minimizes the median of the residuals is chosen. The correspondences having residual error larger than a fixed threshold are removed. The least squares error norm is computed using only the remaining contour points.

B. Trimmed ICP and Picky ICP

The main steps of both trimmed ICP and picky ICP algorithms are the following [11].

1) For each point of $\mathcal{P}$, find the closest point in $\mathcal{M}$ and compute the individual distances d_i^2.
2) Sort d_i^2 in ascending order, select the N_{po} least values and calculate their sum S'_{LTS}.
3) If any of the stopping conditions is satisfied, exit; otherwise, set $S_{LTS} = S'_{LTS}$ and continue.
4) For the N_{po} selected pairs, compute the optimal geometrical transformation $(\mathbf{R}, \mathbf{t})$ that minimizes S_{LTS}.
5) Transform $\mathcal{P}$ according to $(\mathbf{R}, \mathbf{t})$ and go to step 1.

This algorithm terminates either if (a) the maximum number of iterations is reached, or (b) the *trimmed mean squared error* $e = S_{LTS}/N_{op}$ is less than a user defined threshold, or (c) the relative change of the trimmed mean squared error $|e - e|'$,

where e' is the trimmed mean squared error found in the previous iteration, is less than a user defined threshold. Despite their good performance, the above algorithms, namely RICP, trimmed IC and picky ICP have the disadvantage that the outlier percentage in the input curves, must not exceed 20%. Otherwise, the computed transformation is wrong.

C. ICP Registration Using Invariant Features (ICPIF)

ICPIF is another ICP variant proposed in [32]. New features were introduced in order to improve the correspondence selection, such as the second order moments and the spherical harmonics. After the computation of the above features for every point of $\mathcal{P}$ and $\mathcal{M}$, the traditional L_2 distance, computed during the second step of ICP (10), is replaced by the weighted sum of the old L_2 distance and distances based on the introduced features. The goal is to estimate point correspondences that are not only based on the Euclidean metrics but also incorporate new shape invariant features.

VI. Overall Image Reassembly

Once the true adjacent image fragments couples and their matching contour segments are identified and properly aligned, as described in Sections IV and V, the remaining step is the reassembly of the overall image. However, the mapping scores of the matching contour segments do not suffice for image reassembly, since they are not always well correlated with the true ones. Thus, the approaches that focus only on using the mapping scores, such as [20], are inadequate for producing the true overall image. For this reason, we have developed a novel solution to this problem, which is based on the introduction of new use of the alignment angle, that best aligns the matching contour segments of two image fragments, as described in Section V. Consider three image fragments f_i, f_j and f_k each one having one matching contour segment with the rest ones. We denote by θ_j the rotation angle by which the individual fragment f_j must be rotated, in order to be correctly placed inside the overall reassembled image. The alignment angle, by which we must rotate fragment f_i to align it with the matching contour segment of fragment f_j (before fragment f_j is rotated by θ_j), is denoted by θ_{ij}. In order to align fragments f_i and f_j with respect to each other and place them correctly in the reassembled image, the following steps must be performed:

1) Rotate fragment f_j by θ_j to correctly orient it in the assembled image.
2) Rotate fragment f_i by $\theta_{ij} + \theta_j$ to correctly align its matching contour segment with the corresponding matching contour segment of fragment f_j.

This procedure will simultaneously align fragment f_i with fragment f_j and provide its correct orientation inside the entire image. We can then state that the matching contour segments of pairs (f_i, f_j) and (f_i, f_k) are *compatible*, if and only if

$$\theta_{ij} + \theta_j = \theta_{ik} + \theta_k. \tag{13}$$

Following this logic, in our approach we consider that an image is fully reconstructed, if all its matching contour segments are compatible, according to (13); this image is called *valid*. It should be noted that a valid image is not always the "correct" one, since the contour segments of two image fragments can be mis-matched or mis-aligned. However, the experimental evaluation of the technique shows that such flaws do not significantly affect the reassembly performance of the proposed technique.

Contour segments that match may become incompatible, if the following errors occurred in the previous steps:

1) the fragment color similarities found in Section III;
2) the matching segments of two fragment contours are different from the true ones (see Section IV);
3) wrong alignment transformation was estimated (see Section V).

Based on (13), we can define the so-called *relative alignment angle*, which will direct the image reassembling procedure. The relative alignment angle ϕ_i^j of a fragment f_i regarding a fragment f_j is evaluated by the formula:

$$\phi_i^j = \theta_{ij} + \theta_j \tag{14}$$

where θ_{ij} is the alignment angle by which we must rotate fragment f_i to align it with the matching contour segment of fragment f_j and θ_j is the rotation angle of the individual fragment f_j so that it is correctly placed inside the overall reassembled image. It can be easily deduced from the above analysis, that for a set of fragments placed in the reassembled image and a fragment f_i having matching contour segments with a subset of the above image fragments, i.e., $I = \{f_1, f_2, \ldots, f_n\}$, the above matching contour segments are compatible, if

$$\phi_i^1 = \phi_i^2 = \phi_i^3 = \ldots = \phi_i^n. \tag{15}$$

Consequently, if a new fragment i is to be matched with an image that it consists of image fragments, then the amount of the new valid images is equal to the cardinality r of the relative alignment angle set, $\{\phi_i^l, l = 1, \ldots, r\}$. We have developed a reassembly algorithm that utilizes the aforementioned assumptions to produce a user-defined number of possible reassembled images. The algorithm is initialized by taking M couples of input fragments, where $M \leq N * K$ and N is the number of input image fragments. These M couples have the highest matching scores of the corresponding matching contour segments. These couples are reassembled (correctly aligned) to produce a number of M input images that will be further extended by inserting one fragment each time. The selection of the initial number of M image couples is crucial for the algorithm performance, since the involvement of an erroneous input pair of fragments would inevitably lead to a wrong image reconstruction. The reliability of the proposed initialization is both intuitionally expected and experimentally proven.

The set of M input images is iteratively updated in order to include further image fragments. At every step and for every input image (initially consisting of a couple of image fragments), the fragment having the maximum mapping score with an image fragment that belongs in the reconstructed image, is added. After computing the relative alignment angles of this fragment regarding the input image parts, a number of images, equal to the cardinality of these relative aligned angles, are reassembled for each input image. Those images compose the new set of input reassembled images (replacing the previous ones), that will be included in the next iteration. Consequently,

Inputs
$\mathcal{F}$: set of N image fragments.
$\mathcal{E}'$: set of retained image fragments couples (f_i, f_j), (see section III). The following information must be supplied along with each couple (f_i, f_j):
θ_{ij}: the alignment angle by which fragment f_i must be rotated in order to be aligned with the matching contour segment of f_j.
S_{ij}: the mapping score of the couple.
M: amount of image fragments couples taken from $\mathcal{E}'$ used to initialize this algorithm.
Output
$\mathcal{I}$: set of reassembled images.

$\mathcal{I} \leftarrow \emptyset$;
$\mathcal{T} = \{(f_i, f_j) \in \mathcal{E}' : S_{ij}$ is one of the M highest similarity scores$\}$;
for each (f_i, f_j) *in* $\mathcal{T}$ **do**
  rotate f_i under θ_{ij} and align it with the matching contour segment of f_j ;
  reassemble a new image I from f_i and f_j ;
  $\mathcal{I} = \mathcal{I} \cup \{I\}$;
end for

repeat
  Take one partially reassembled image I from $\mathcal{I}$; {Let $I = \{f_1, f_2, \ldots, f_m\}$}
  Find an image fragment $f_i \notin I$ such that $(f_i, f_j) \in \mathcal{E}'$, and $S_{ij} = \{\max_{i|f_i \notin I}\{\max_{j|f_j \in I}\{S_{ij}\}\}\}$;
  $I' = \{f_j | f_j \in I \wedge (f_i, f_j) \in \mathcal{E}'\}$; {$I'$ is the set of image fragments that belong in the partially reconstructed image I and have a matching segment with fragment f_i. }
  Estimate $\phi_i^j, \forall f_j \in I'$;
  Let r be the cardinality of the estimated relative alignment angles, i.e. $\phi_i^l, l = \{1, \ldots, r\}$;
  Form sets $\mathcal{T}_l = \{f_j \in I' : \phi_i^{j_1} = \phi_i^{j_2} = \ldots = \phi_i^{j_n}\} \cup (I - I')$, where $l = \{1, \ldots, r\}$ and n is the size of each $\mathcal{T}_l$; {Each set $\mathcal{T}_l$ contains image fragments, for which fragment f_i has the same relative alignment angle, plus the rest fragments in I for which f_i has no matching contour segment with. }
  for all $\mathcal{T}_l, l = \{1, \ldots, r\}$ **do**
    Form a new image NI after rotating f_i under ϕ_i^l and aligning it with the corresponding matching contour segments of fragments in $\mathcal{T}_l$;
    $\mathcal{I} = \mathcal{I} \cup \{NI\}$;
  end for
  $\mathcal{I} = \mathcal{I} - \{I\}$; {remove old image I.}
until no fragment can be added in any reassembled image $I \in \mathcal{I}$

Fig. 5. Overall image reassembly step.

in the next step a new fragment will be involved in the update procedure of the input images. This iterative algorithm is finalized when no more image fragments are left to be inserted into the reassembled images. The steps of the algorithm are shown in Fig. 5.

The proposed method differs from other ones found in the literature, as it is based on the alignment angles. Thus, it is sensitive to errors in the relative alignment angle estimations. However, we overcome this drawback, by setting a threshold when comparing the relative alignment angles. We regard two relative alignment angles ϕ_1 and ϕ_2 to be equal if $|\phi_1 - \phi_2| < \epsilon$, where ϵ is a decision threshold. Of course, a backup approach is to add manual intervention to the system, and ask the user to select the correct alignment transformation.

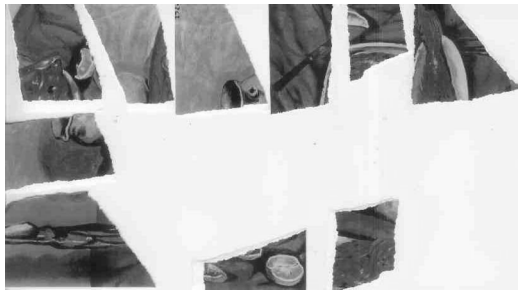

Fig. 6. Scanned paper pieces at 300-dpi resolution.

TABLE I
RECALL PERCENTAGE OF THE CORRECT COUPLES OF ADJACENT IMAGE FRAGMENTS FOR $L = 15$

Histogram distance measure	Recall
d_{L_1}	45.48
d_{L_2}	46.53
d_{HI}	**53.25**
d_{SCH}	48.71

TABLE II
EFFECT OF COLOR QUANTIZATION METHODS IN CONTOUR SEGMENT MATCHING

Color Matching Method	False Negative Error	False Positive Error
KNN [30]	**20.18%**	**22.56%**
Mean Shift Algor. [29]	53.69%	53.68%
Gret. Macbeth Pal. [26]	51.36%	51.35%
Method presented in [5]	67.73%	60.78%

TABLE III
ERFORMANCE OF THE PROPOSED ICP VARIANTS IN CORRECTLY MATCHED CONTOUR SEGMENT ALIGNMENT

ICP variant	Performance %
ICP [9]	45.71
Trimmed ICP [11]	62.68
RICP [10]	64.36
ICPIF [32]	**77.12**

VII. IMAGE REASSEMBLY EXPERIMENTS

The performance of the proposed method was evaluated using 70 paper image prints. Each paper image print had size 25 cm × 20 cm and was torn into $N = 20$ paper pieces that were scanned at 300 dpi resolution, as shown in Fig. 6. We selected such a low scanning resolution in order to create a challenging image fragments dataset. Experiments are conducted in a 4 kernel PC with 4 GB RAM. The algorithm has been implemented in C++. The first step of the proposed method, as described in Section III,

TABLE IV
OVERALL IMAGE REASSEMBLY PERFORMANCE

Overall Assembly	% 10 fragm. perfor.	% 20 fragm. perfor.	10 fragm. comput. time	20 fragm. comput. time
No first stage	79.23	63.49	2.31 min	7.58 min
Including first stage	57.43	42.15	1.07 min	3.42 min
Human	-	-	5.64 min	17.37 min

is the discovery of similarly colored image fragments. We selected to retain for every image fragment the 15 most probable adjacent fragments ($L = 15$). In Table I, the mean recall percentage of the correct couples of adjacent image fragments for $L = 15$ is shown for various histogram similarity measures. These results are averaged over all the image fragments of the entire image fragments dataset. The d_{HI} measure exhibits the highest performance.

Whether executing (or not) the first step of the proposed method, the contour segment matching for every input couple of image fragments must be done. When the first step is executed, the matching contour segments are identified only for retained image couples, otherwise it is executed for all possible input fragment couples. We have experimented by combining the dynamic programming algorithm with several color quantization methods, described in Section IV. We have also evaluated the performance of the matching contour algorithm introduced in [5]. Each time, N_p random pixels were sampled from all fragments that constitute the overall image, in order to perform color quantization either by using the Kohonen neural nets or the Mean Shift algorithm. In both cases N_p was set equal to 0.25% of the total image fragment pixels. We have chosen Kohonen nets having $C = 50$ color clusters, while the learning procedure took 750 epochs. Nodes in the output layer of the neural network were organized under a random lattice. The spread parameter of Mean Shift algorithm [29] was set to 5.5, thus producing $C = 57$ color clusters. Following one from the previously described color quantization techniques, each contour pixel is represented by its color cluster label ($C = 24$ utilizing the Gretag Macbeth Checker). The parameters of the Smith Waterman algorithm were set to $e = 1$, $d = -0.5$ and $g = -0.5$. The contour matching results are shown in the last row of Table II. The second column refers to the percentage of unidentified matching contour pixels (false negative error rate), while the third column refers to the percentage of identified pixels that do not belong in true matching contour segments (false positive rate). The selection of the specific performance metrics is justified by the fact that most of the contour matching approaches, [5], [26], [29], lead to the extraction of "estimate" matching contour segments that overlap with the "correct" matching contour segments. The matching performance depends on the quality of the overlap, which is estimated by the ratio of unidentified and mis-identified contour pixels, or false negative and false positive rate. After this step, for each image fragment, one matching contour segment to $K = 10$ image fragments is retained, as described in Section IV. Color quantization using the Kohonen nets provides the best solution, identifying correctly more than 70% of the contour segment pixels, while minimizing the amount of misidentified contour pixels.

All matching contour segments identified in Section IV must be properly aligned, before the reassembly of the overall image. Such segments were produced by combining the dynamic programming algorithm with color quantization using the Kohonen nets. We have evaluated the performance of the proposed ICP variants by aligning the correct matching contour segments, i.e., the contour segments that really match in the reassembled image. These contour matchings may contain misidentified pixels or erroneously non-identified contour pixels. Each ICP variant was executed 1000 times. Implementation details concerning parameter selection, as well as computational complexity, can be found in [9]–[11] and [32] respectively. A complexity analysis of the ICP algorithm is presented in [33]. According to [33], the worst-case lower bound on the number of iterations performed by the ICP algorithm in order to converge is $\Omega(n/d)^{d+1}$, where n is the size of the input data points set and d is the dimensionality of the input data, while the smoothed [33] upper bound of the ICP algorithm is polynomial, independent of the dimensionality of the data. In Table III, the performance (percentage of visually correctly identified alignments over correctly matched contour segments) is shown. According to Table III, ICPIF outperforms the rest of the ICP variants.

The final step is the overall image reassembly. We evaluated the proposed method with/without employing the first step (the discovery of probably adjacent image fragments) and compared it with the manually (human) reassembly. The performance was estimated as follows. The performance of the method was estimated as follows. Let I be a manually reassembled image and $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$ be the set of automatically reassembled images generated by the algorithm. The reassembly is defined to be correct, when there is a $\widetilde{k} \in \{1, \ldots, n\}$ for which $I_{\widetilde{k}}$ is declared similar to I by a human observer.

In Table IV, the mean performance and computational time characteristics of the presented approach, regarding images fragmented into 10 and 20 pieces respectively, are shown. In both cases the image reassembly is performed employing exactly the same techniques in every step, while L and K parameters in the first case are set to 7 and 5, respectively. The first two columns show the percentage of correctly reassembled images, as described above, for the case of 10 fragments and 20 fragments, respectively. The last two columns show the respective mean computational time. The last row shows mean manually reassembly time efficiency (the performance of manually reassembly is not measured). Eight people of ages between 23 to 31 participated in this experiment. We have averaged the time needed for each person to correctly

Fig. 7. Automatically reassembled image produced from the fragments of Fig. 6.

(1) (2) (3)

Fig. 8. New fragments created by aligning and assembling original fragments along their matching contour segments.

reassemble each fragmented image. As expected, the overall performance will be higher when the first stage is omitted. However, in this case the computation cost is higher, since both the second and the third step are executed for every couple of input image fragments. In the first case, where the the amount of fragments is somewhat small (10 fragments), the reassembly time efficiency is high in every case (with/without first step, manually reassembly). As the amount of fragments per input image increases, the time efficiency deviations become higher. Table IV shows that the choice whether to employ the first step on the reassembly process depends on the computational cost that is associated within the input image fragments. As the amount of image fragments increases, the higher overall performance of the variation that omits the first step gives way to the higher computational cost that is associated with it. Fig. 8 displays aligned couples of image fragments produced during the second and the third stages. The overall reassembled image is shown in Fig. 7. It can be seen that its reconstruction is nearly perfect. The white region in the middle of the reassembled image is due to missing pieces of paper that were not scanned.

VIII. Conclusions and Future Work

In this paper, we have introduced a novel integrated color-based image fragments reassembly method that consists of several distinct novel algorithms, which try to overcome inherent deficiencies, burdens and limitations. The overall assembly performance has proven to be particularly satisfactory. The experiments were conducted with 70 paper image prints. We plan to further improve the performance of the proposed method. Improvements can be done in each step of the proposed method. For example, the step of the discovery of spatial adjacent image fragments may be improved by employing not only color but textual or semantic features as well. Another possible extension is to utilize both the color and the shape of the fragments contours in order to perform matching. Furthermore, more sophisticated algorithms for faster contour matching, such as the one presented in [4], will be investigated. Another interesting idea would be to interchange the steps used in the proposed method, e.g., a shape alignment algorithm could provide a first set of coarse fragments shape matching results that would be further refined by employing color matching. Finally, the evaluation of the proposed method in a realistic fragmented image database, such as archaeological data, is worth exploring.

References

[1] B. Desbenoit, E. Galin, and S. Akkouche, "Modeling cracks and fractures," *Vis. Comput.*, vol. 21, pp. 717–726, 2005.

[2] M. Fornasiera and D. Toniolo, "Fast, robust and efficient 2D pattern recognition for re-assembling fragmented images," *Pattern Recognit.*, vol. 38, pp. 2074–2087, 2005.

[3] A. R. Willis and D. B. Cooper, "Computational reconstruction of ancient artifacts," *IEEE Signal Process. Mag.*, pp. 165–183, Jul. 2008.

[4] H. C. G. Leitao and J. Stolfi, "A multiscale method for the reassembly of two dimensional fragmented objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1239–1251, Sep. 2002.

[5] F. Amigoni, S. Gazzani, and S. Podico, "A method for reassembling fragments in image reconstruction," in *Proc. Int. Conf. Image Processing*, Sep. 2003, pp. 581–584.

[6] H. J. Wolfson, "On curve matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 5, pp. 483–489, May 1990.

[7] J. Salvi, C. Matabosch, D. Fofi, and J. Forest, "A review of recent range image registration methods with accuracy evaluation," *Image Vis. Comput.*, vol. 25, no. 5, pp. 578–596, 2007.

[8] H. Alt, L. Scharf, and S. Scholz, "Probabilistic matching of sets of polygonal curves," in *Proc. 22nd European Workshop on Computational Geometry (EWCG)*, Delphi, Greece, Mar. 2006, pp. 107–110.

[9] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image Vis. Comput.*, vol. 10, no. 3, pp. 145–155, 1992.

[10] E. Trucco, A. Fusiello, and V. Roberto, "Robust motion and correspondences of noisy 3D point sets with missing data," *Pattern Recognit. Lett.*, vol. 20, pp. 889–898, 1999.

[11] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The trimmed iterative closest point algorithm," in *Proc. 16th Int. Conf. Pattern Recognition*, 2002, vol. 3, pp. 545–548.

[12] T. Zinsser, H. Schnidt, and J. Niermann, "A refined icp algorithm for robust 3D correspondences estimation," in *Proc. Int. Conf. Image Processing (ICIP)*, 2003, pp. 695–698.

[13] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *J. Mol. Biol.*, vol. 147, pp. 195–197, 1981.

[14] B. Biswas, P. Bhowmick, and B. B. Bhattacharya, "Reconstruction of torn documents using contour maps," in *Proc. Int. Conf. Image Processing (ICIP)*, Sep. 2005, vol. 3, pp. 517–520.

[15] E. Justino, L. S. Oliveira, and C. Freitas, "Reconstructing shredded documents through feature matching," *Fores. Sci. Int.*, vol. 160, pp. 140–147, 2006.

[16] L. Zhu, Z. Zhou, and D. Hu, "Globally consistent reconstruction of ripped-up documents," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 1, pp. 1–13, Jan. 2008.

[17] M. S. Sagiroglu and A. Ercil, "A texture based matching approach for automated assembly of puzzles," in *Proc. 18th Int. Conf. Pattern Recognition (ICPR)*, 2006, vol. 3, pp. 1036–1041.

[18] A. Kalvin, E. Schonberg, J. Schwartz, and M. Sharir, "Two dimensional model based boundary matching using footprints," *Int. J. Robot. Res.*, vol. 5, no. 4, pp. 38–55, 1986.

[19] D. Goldberg, C. Malon, and M. Bern, "A global approach to automatic solution of jigsaw puzzles," *Computat. Geom.: Theory Appl.*, vol. 28, pp. 165–174, 2004.

[20] W. Kong and B. B. Kimia, "On solving 2D and 3D puzzles using curve matching," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, vol. 2, pp. 583–590.

[21] G. Papaioannou, E. A. Karabassi, and T. Theoharis, "Reconstruction of three-dimensional objects through matching of their parts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 114–124, Jan. 2002.

[22] S. Andrews and D. H. Laidlaw, "Toward a framework for assembling broken pottery vessels," in *Proc. 18th American Conf. Artificial Intelligence*, 2002, pp. 945–946, American Association for Artificial Intelligence (AAAI).

[23] Y. Lu, H. Gardner, H. Jin, N. Liu, R. Hawkins, and I. Farrington, "Interactive reconstruction of archaeological fragments in a collaborative environment," in *Proc. 9th Biennial Conf. Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA '07)*, Washington, DC, 2007, pp. 23–29, IEEE Computer Society.

[24] D. Koller, J. Trimble, T. Najbjerg, N. Gelfand, and M. Levoy, "Fragments of the city: Stanford's digital forma urbis romae project," *J. Roman Arch. Suppl.*, vol. 61, pp. 237–252, March 2006.

[25] H. C. G. Leitao and J. Stolfi, "Measuring the information content of fracture lines," *Int. J. Comput. Vis.*, vol. 65, no. 3, pp. 163–174, 2005.

[26] M. A. Gavrielides, E. Sikudova, and I. Pitas, "Color based descriptors for image fingerprinting," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 740–748, Aug. 2006.

[27] L. Cinque, G. Ciocca, S. Levialdi, A. Pellicano, and R. Schettini, "Color based image retrieval using spatial chromatic histograms," *Image Vis. Comput.*, vol. 19, pp. 786–979, 2001.

[28] R. C. Gonzalez and R. E. Woods*, Digital Image Processing*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 2006.

[29] Y. Cheng, "Mean-shift mode seeking and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, Aug. 1995.

[30] G. A. Carpenter and S. Grossberg*, Pattern Recognition by Self-Organizing Neural Networks*. Cambridge, MA: MIT Press, 1991.

[31] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc . 3rd Int. Conf. 3D Digital Imaging and Modeling (3DIM)*, Jun. 2001, pp. 145–152.

[32] G. C. Sharp, S. W. Lee, and D. K. Wehe, "Icp registration using invariant features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 90–102, Jan. 2002.

[33] D. Arthur and S. Vassilvitskii, "Worst-case and smoothed analysis of the icp algorithm, with an application to the k-means method," in *Proc. 47th Annu. IEEE Symp. Foundations of Computer Science (FOCS)*, 2006, pp. 153–164.

Efthymia Tsamoura received the Diploma degree in computer science with honors (9.15/10) in 2007. She is currently pursuign the Ph.D. degree in the Computer Science Department, Aristotle University of Thessaloniki (AUTH).

During October 2007–June 2008, she was a research assistant in the Multimedia Knowledge Laboratory, Informatics and Telematics Institute (ITI), where she participated in the European project Vidi-Video. Her research interests lie in the field of distributed query optimization and grid computing, as well as image and video analysis processing. She was awarded from the State Scholarship Foundation of Greece for exceptional performance during the academic years 2003-2004, 2004-2005, 2005-2006, and 2006-2007, while this work was conducted during her undergraduate thesis

Ioannis Pitas (S'83–M'84–SM'94–F'07) received the Diploma of Electrical Engineering in 1980 and the Ph.D. degree in electrical engineering in 1985 both from the Aristotle University of Thessaloniki, Greece.

Since 1994, he has been a Professor in the Department of Informatics, Aristotle University of Thessaloniki. From 1980 to 1993, he served as a Scientific Assistant, Lecturer, Assistant Professor, and Associate Professor in the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki. He has served as a Visiting Research Associate or Visiting Assistant Professor at several universities. He has published 153 journal papers, 400 conference papers, and contributed to 22 books in his areas of interest and edited or coauthored another five books. He has also been an invited speaker and/or member of the program committee of several scientific conferences and workshops.

Dr. Pitas has served as Associate Editor or Co-Editor of four international journals and as General or Technical Chair of three international conferences. His current interests are in the areas of digital image and video processing and analysis, multidimensional signal processing, watermarking, and computer vision.