MOSAIC: Multi-object Segmentation for Assisted Image ReConstruction

Sonia Caggiano[1], Maria De Marsico[2](✉), Riccardo Distasi[3], and Daniel Riccio[4]

[1] Master of Architecture and PhD in Digital Painting Restoration, Salerno, Italy
soniacaggiano@yahoo.it
[2] Sapienza University of Rome, Via Salaria 113, 00198 Rome, Italy
demarsico@di.uniroma1.it
[3] University of Salerno, 84084 Fisciano, SA, Italy
ricdis@unisa.it
[4] University of Naples "Federico II", 80121 Campi Flegrei, NA, Italy
daniel.riccio@unina.it

Abstract. This paper presents a tool targeted at archaeologists and cultural heritage operators. The tool assists the process of reconstructing broken pictorial artifacts from their physical fragments. The fragments are organized into a database indexed on features such as color distribution, shape and texture. The system can be queried using any fragment as the key, and the results are displayed from the most similar to the most dissimilar. The system provides the operator with complete workflow from photoacquisition onwards. The performance has been assessed with computer simulations and a real use case. Two of the simulations are discussed, as well as the real use case, based on an actual XV century fresco that needed reconstruction.

Keywords: Image processing · Feature extraction · Feature-based indexing · Jigsaw puzzle · Cultural heritage

1 Introduction

Imagine being an archaeologist at work on site. The remnants of a church are being dug out, and it is known that the wall that used to be right in front of your present standing position was painted with a fresco. The scene depicted on the fresco is not known, however. All you can hope to have access to—given enough patience and careful, slow work—will be a collection of painted fragments that crumble to dust if not handled with the utmost gentleness. Reconstructing at least part of the original design involves much manipulation: the pieces have to be repeatedly rotated, tentatively aligned, moved in batches and so on. Going through all the available fragments to locate a possible candidate piece for a particular spot of your puzzle is no trivial task, either: each time a piece is touched or moved it might break, or at least its edge might be ground away, depending on the materials used to build and coat the old wall. It would be nice to have some digital tool to ease the whole process.

A. Fred et al. (Eds.): ICPRAM 2015, LNCS 9493, pp. 282–299, 2015.
DOI: 10.1007/978-3-319-27677-9_18

Enter MOSAIC—an automated system for computer-aided reconstruction of jigsaw puzzles. Such puzzles are grouped into two broad categories, requiring different approaches: apictorial puzzles, where fragment shape is the only kind of available information, and pictorial puzzles, where texture and color information is available and can be meaningfully used—which, unlike most commercial puzzles, does not imply that the solution image is known a priori. Pictorial or apictorial, it has been shown that providing an exact algorithmic solution is an NP-complete problem: the computing time grows super-polinomially with problem size [8]. However, if a non-exact solution is acceptable, there are techniques that provide approximate solutions in a shorter time [4]. The available literature offers several options for tackling both types of jigsaw puzzles—and several applications as well, mostly in the fields of cultural heritage and ancient document reconstruction.

In recent times, advances in hardware and software design have provided improved performance. For example, the Roman site in Tongeren, Belgium contained a number of artifacts that have been at least partially reconstructed [2]. The fragments have been acquired using an ad hoc 3D scanner, and the extracted shapes have been matched by a custom piece of software. The total cost, however, remained quite high—around $25,000 for the scanner only. Such a massive employ of work and money did improve the number of true matches nearly sixfold, but this result is still far from satisfactory, considering the actual numbers: 3 true matches had been found manually, and the system proposed 6103 tentative matches that became 17 true matches after human screening.

Another semi-automatic system was made by Brown et al. [3]. It was used for recontruction frescoes at the Akrotiri site in Thera, Santorini. Greece. Although the sistem does use 3D data, most of the results are obtained via 2D image feature extraction.

Freeman and Gardner's classic paper was one of the earliest to tackle the problem of apictorial jigsaw puzzles [8]. Five fundamental puzzle properties were pinpointed: orientation (unknown a priori), connectivity (presence or absence of internal "holes"), perimeter shape (known/unknown a priori), uniqueness (does the problem admit only one solution?), radiality (topology of fragment juncture). The contours of the fragments are represented as chain codes, and code length is used as a heuristic for search space dimensionality reduction.

Papaodysseus et al. face this problem in the context of reconstructing wall paintings [12]. This context makes their paper particularly relevant for the present discussion. The focus is on specific real-world issues that arise when dealing with a fresco: lack of information about the original content of the painting, non-uniqueness, and especially non-connectedness arising from the presence of very small fragments that are not available to the problem solver. The technique for correspondence deals with missing information using local curve matching.

Better solutions can be obtained by fully exploiting all available information. Most techniques actually used in the cultural heritage field regard the problem not as an apictorial, but as a pictorial puzzle. For instance, Chung et al. use color [5], while Sagiroglu and Ercil use texture [13]. However, actual testing has been limited to problems involving a relatively small number of fragments.

On the other hand, for an example that disregards fragment shape altogether, consider Nielsen et al. [11]. Their method relies on features of the whole represented pictorial scene without using any information pertaining to single pieces. The reported results for this technique show low error margins: the solution to a 320-fragment problem only had 23 pieces out of place—an error margin of 7.2 % obtained by using only color and texture information.

As we have seen, the virtual reconstruction of pictorial fragments is an intrinsically hard problem, and approximate solutions are often all we can get. For this reason, several sophisticated image processing techniques are being incorporated in most recent systems. The most promising are based on local texture analysis, chrominance analysis and contour analysis on single fragments. Methods based on properties of the whole scene are quite powerful, when the original appearance is known or can be at least partially inferred, and can provide further features to consider. Such techniques can produce multimodal representations that allow users to refine the solution progressively, adding detail and information to the features of the solution search space.

The present paper proposes a system for the segmentation and indexing of pictorial fragments: Multi-Object Segmentation for Assisted Image reConstruction (MOSAIC). MOSAIC supports the rebuilding of a fresco from fragments by a human operator. No information about the original appearance of the whole artwork is assumed to be available. The system has been tested with several computer simulations and on a real case study: the reconstruction of a fresco from fragments found in the St. Trophimena church in Salerno (Italy).

2 MOSAIC Procedures

Mosaic can be classified among techniques for Jigsaw pictorial puzzle solving where texture and color information is available. This information can be meaningfully used together with shape information. As mentioned above, it has been shown that an exact algorithmic solution to puzzle solving is NP-complete [7]. Therefore, a number of models supporting alternative approaches have been investigated—e.g., [4]. Our proposal relies on a human-in-the-loop approach.

In particular, MOSAIC was expressly designed to support archaeologists and restorers facing fresco recomposition from fragments. Its software modules implement a procedure for image acquisition and processing. The aim of this procedure is not to perform a completely automatic reconstruction, but rather to relieve the expert from most of the burden implied by reordering fragments and grouping them in similar clusters. The result is a catalog of the single fragments, which are grouped according to their texture/color and shape characteristics. This grouping allows answering user queries, so that reconstruction is made easier, quicker and more effective. The application provides a virtual workspace; here, among the other actions, the user can perform the actions that would have been performed in a real reconstruction attempt, i.e., virtually rotate, translate and search for similar fragments. Figure 1 shows a schema of the sequence of steps executed by the system. After image acquisition, segmentation and feature

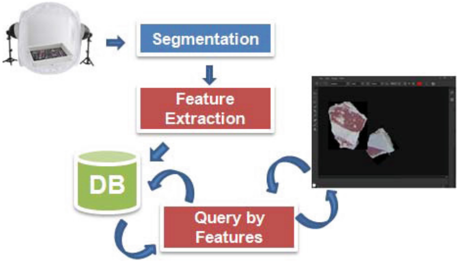


Fig. 1. Steps executed by the MOSAIC system.

extraction modules build the fragment archive. A query engine allows searching the archive for relevant fragments, and a manipulation interface allows the user to manipulate them virtually to attempt recomposing the broken picture.

In the starting phase—image acquisition—the physical fragments are laid in a white tray, whose bottom is covered by a dark foam to reduce reflexions. The tray containing the fragments is placed inside a box for photographic acquisition, which is made by a white curtain and two lateral spotlights. Close to the tray, a colorimeter is used to check for the need for automatic color corrections. The tray is then photographed by a suitable device (in this work, we used an 8-Mpixel Canon camera), orthogonally pointing it from a height of 90 cm.

2.1 Fragment Catalogue

Segmentation. Segmentation accuracy significantly affects the rest of the procedure. The purpose of this operation is to correctly separate each fragment, so that individual features can be extracted from each of them. Segmentation is carried out in two steps. In the first one, the image is binarized and turned into B/W with no shades of gray. Nowadays, binarization of a color photo might appear as a trivial task. However, using naive thresholding on a set of different raw images does not work. In particular, in our case no single threshold value is effective across all trays, unless some pre-processing occurs to enhance the image color separability. Values that are too low fail in separating one piece from another, while values that are too high ones produce "holes" inside pieces. In some simple cases, a morphological fill operation can fill such holes, but in other cases the piece may even come out as two separate fragments—an error that is quite hard to correct later. An example of problematic binarization is shown in Fig. 2.

The process of binarization that we carry out first needs to pre-process the raw image in order to amplify the difference between the brighter pixels (fragments) and the darker ones (the background foam is a dark shade of gray—almost black, but not quite). This pre-processing is described below.

The original image is represented in RGB space, whose single channels are stored in three separate matrices r, g, and b. Two new matrices are then created: M and m. The element $M(i,j)$ of the new matrix M is equal to the maximum

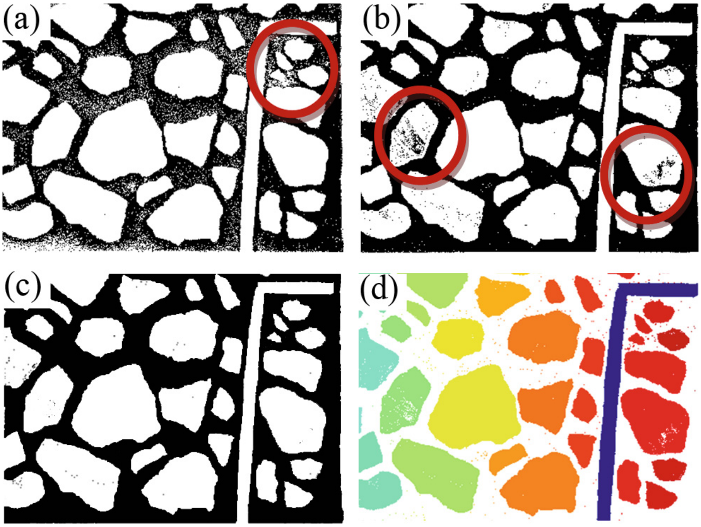


Fig. 2. Effect of threshold parameter t_B choice on the segmentation of a tray image: (a) too low; (b) too high; (c) optimal value $t_B = 0.1$; (d) connected components detected after binarization.

over the three channels in (i, j) position, i.e. the maximum among $r(i, j)$, $g(i, j)$ and $b(i, j)$. In practice, matrix M represents a new image where each pixel contains the largest (brightest) component of the original image. The new matrix m is built so that the image that it represents is a greyscale version of the original image: its element $m(i, j)$ is equal to the mean of $r(i, j)$, $g(i, j)$ and $b(i, j)$. Matrices M and m are used to derive the enhanced image I as follows. First, the transformation described in (1) is applied:

$$I\left(i,j\right) \leftarrow M\left(i,j\right) \cdot \left|m\left(i,j\right) - \delta\right| \,. \tag{1}$$

The value δ in Eq. (1) is a constant offset which is experimentally determined and which is constant over all images. In our specific case, $\delta = 50$. In practice, this value is chosen to be close to the mean luminance value of all pixels in the tray (both fragments and background). Then, the pixel values in I are scaled by dividing them by their mean value $\bar{I}$ as in (2):

$$I\left(i,j\right) \leftarrow I\left(i,j\right)/\bar{I} \,. \tag{2}$$

After these operations that enhance color separability, we can apply a single threshold value for all trays. A value of $t_B = 0.1$ has been found to be suitable for all tray images in our pool. The new I is then turned into a 0–1 binary image by thresholding it according to the t_B binarization threshold value:

$$I\left(i,j\right) \leftarrow \begin{cases} 0, & \text{if}\, I\left(i,j\right) < t_B \\ 1, & \text{if}\, I\left(i,j\right) \geq t_B \end{cases} . \tag{3}$$

We now explain the rationale for the above operations. The grayscale image m and the maximum component image M are pointwise multiplied in order to enhance the pixels where both m and M have larger values. The resulting image is then divided by its mean value to perform a kind of normalization, so that the value of the threshold t_B used for binarization does not depend on the particular image anymore. The final result of binarization can be seen in Fig. 2(c).

An algorithm for detecting connected components takes the binary image just obtained as input. In an ideal situation, each fragment should correspond to a single connected component and vice versa, as in Fig. 2(d). To obtain this, a morphologic fill operator is applied to each fragment shape extracted from the binary image, in order to fill possible existing gaps or holes. Afterwards, specific information about the newfound fragment is computed, namely its area, its perimeter, and its orientation. We finally obtain a binary connected component, that will be used as a mask to retrieve the fragment from the original image by a pixel-wise logical AND operation.

Feature Extraction. The module for feature extraction takes the fragments resulting from segmentation, so that these can be indexed to allow a convenient successive retrieval. The features chosen for indexing/retrieval are the (basic) shape(s) depicted on the fragment and a spatiogram, which describes the spatial distribution of colors on the fragment surface. The user can search the fragment database according to similar color, similar shape, or similar spatial color distribution, in order to retrieve fragments similar to a given "key" fragment.

Color. Information about color is represented here by a spatial histogram, or spatiogram, computed over the fragment [1]. A histogram can be considered as a zeroth-order spatiogram, while second-order spatiograms also contain spatial means and covariances for each histogram bin, i.e., information about the spatial domain. In other words, a spatiogram is a histogram where the count of occurrences of each color or group of colors (bin) is augmented with spatial information related to it, namely the mean vector and covariance matrices, respectively, of the coordinates of the pixels containing that color (bin). Given an image I and a quantization of its colors (or gray levels) in B bins, we denote the histogram of I as h_I, and the number of pixels in bin b as $h_I(b)$. The second order spatiogram is then

$$h_I^{(2)}(b) = \left\langle n_b, \mu_b, \sum_b \right\rangle , \tag{4}$$

where $b = 1, \ldots, B$, n_b is the number of pixels that fall in the b-th bin (as in the histogram), and μ_b and $\sum_b$ are the mean vector and covariance matrices, respectively, of the coordinates of those pixels. The similarity between two spatiograms can be computed as the weighted sum of the similarity between the two corresponding histograms, where the weight depends on the order of the spatiogram, and the histogram similarity can be computed by a number of techniques, as for example histogram intersection or the Bhattacharyya coefficient, as shown in (5) and (6), where n'_b and n''_b are the values of the two histograms for bin b:

$$int(n'_b, n''_b) = \frac{min(n'_b, n''_b)}{\sum_{j=1}^{B} n'_j} \tag{5}$$

$$bhatt(n'_b, n''_b) = \frac{\sqrt{n'_b \cdot n''_b}}{\sqrt{(\sum_{j=1}^{B} n'_j)(\sum_{j=1}^{B} n''_j)}} \,. \tag{6}$$

For more details, see [1]. Similarly to histograms, spatiograms allow simple manipulations and are especially useful for comparisons between image areas, because they do not need a geometric mapping between the areas involved. In addition to histograms, the further spatial information about color distribution provides increased matching accuracy.

Shape. The information extracted describes the predominant shapes of the pictorial elements depicted in a fragment. The pixels first undergo a clustering procedure based on color. Clusters resulting in each fragment are then analyzed to extract shape information.

Color clustering is performed by a mean-shift based method [6]. Such methods have been chosen because they are non-parametric and fairly insensitive to noise or similar low-level distortions. Moreover, mean-shift based clustering does not require to predetermine the number of clusters. However, in most cases the result obtained is over-segmented for our purposes. This is exemplified in Fig. 3.

Over-segmentation is corrected by determining a thresholding "color radius" t_C: the distinct RGB colors that lie at a distance less than t_C are coalesced into a single color by re-labeling them. In our specific case, the value for this radius threshold has been determined experimentally and is set as $t_C = 32$. This correction significantly improves clustering accuracy, as demonstrated by comparing Fig. 3(c) and (d).

Each color cluster is considered independently in the context of a single fragment. The pixels belonging to one cluster (and then appearing to the system as being of the same color) are fed to an algorithm for the detection of connected components. This process is depicted in Fig. 3(e). Each detected connected component is in turn processed independently, in order to determine the shapes represented. The smaller components—those whose area is less than 4 % of the total fragment area—are discarded as not significant (small color holes, noise, as well as processing defects occurred during either binarization, segmentation, color clustering, or detection of connected components). At the end of this process, each fragment F_h is characterized by a variable number s_h of shapes S_{h_i}, $i = 1 \ldots s_h$, whose surface area equals at least 4 % of the total fragment surface. These components undergo contour detection so their shapes can be geometrically described. Since this processing is performed on a per-fragment basis, and since the following analysis is performed on single shapes in the fragment, in order to simplify the notation we will drop from now on the subscript identifying fragment and shape.

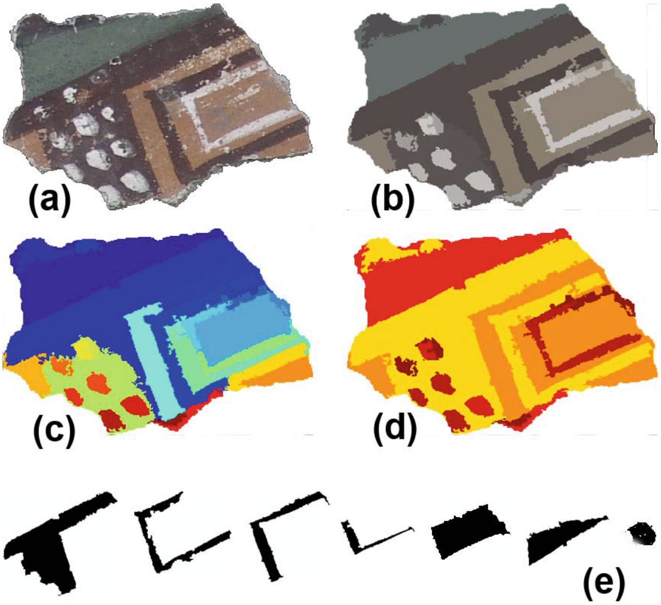


Fig. 3. Shape extraction: (a) the original fragment; (b) the fragment after color clustering, using original colors; (c) false colors reveal oversegmentation; (d) thresholding and re-labeling; (e) extracted shapes.

Shape S is analyzed through its contour C. A further consideration is needed before describing such analysis. Not all components correspond to relevant shapes, even if their area is over the 4 % threshold: some of them are just stains and contribute nothing but noise to the system. Therefore, we investigated possible relevance criteria to assign a higher weight to relevant shapes than to stains. In this context, contour smoothness is a useful relevance criterion. The underlying assumption, which is supported by experts collaborating to MOSAIC design, is that the contour of a pictorial element is usually smoother than the contour of a stain or blemish, which on the contrary is more jagged. Figure 4 shows an example. Since they still contribute to the spatiogram, their possible color information content is not lost in any case. According to this, the shape processing continues as follows.

The contour C of a shape is represented as an ordered sequence of n_C points:

$$C = \{P_1, P_2, \ldots, P_{n_C}\}, \tag{7}$$

where the contour step count n_C differs from shape to shape (and may also be a first cue of its length/regularity). Given a point P_k in C, let us consider another contour point P_{k+l} located l steps further along the same path. In other words, the path from P_k to P_{k+l} has step count l, which also corresponds to the lowest distance between them. Let $\mathrm{d}(\cdot,\cdot)$ be the Euclidean distance between any two

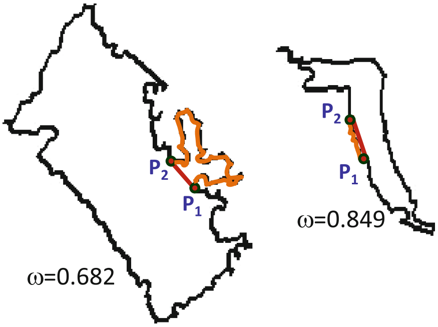


Fig. 4. Smoothness ranks shape contours along the jagged/smooth axis.

points. The smoothness for the subpath starting at P_k and spanning l points $C_{k,l} = \{P_k, P_{k+1}, \ldots, P_{k+l}\}$ is computed as

$$\omega(k,l) = \mathrm{d}(P_k, P_{k+l})/l\,. \tag{8}$$

The actual value used for smoothness calculation depends on n_C and is $l = \lfloor 4\log_2 n_C \rfloor$. The smoothness for the whole contour C is given by

$$\omega(C) = \sum_{k=1}^{n_C} \omega(k,l)\,. \tag{9}$$

It assumes values in $[0,1]$ and is used as a weight in matching operations, as will be shown shortly.

2.2 Fragment Search

After the process of feature extraction, color information is represented by the fragment spatiogram, and comparison between a query and the fragments in the catalogue is performed with related techniques. It is more interesting here to consider search by shape.

After feature extraction carried out on single shapes, each shape is represented as a triple

$$S = \langle \mathbf{v}, \omega, c \rangle\,. \tag{10}$$

In this characterization, $\mathbf{v} = (v_1, \ldots, v_7)$ is the vector of the first 7 central moments of the shape (a thorough discussion of central moments and some of their applications to pattern recognition can be found in [9,10]). The other two elements in the triple, ω and c, are the shape smoothness and mean color value, respectively. A fragment F_h containing s_h shapes is therefore characterized by s_h such triples.

We compare two shapes $S_1 = \langle \mathbf{v_1}, \omega_1, c_1 \rangle$ and $S_2 = \langle \mathbf{v_2}, \omega_2, c_2 \rangle$, by computing their similarity as the normalized dot product of their moment vectors (i.e., the cosine of the angle between them), weighted by the product of their smoothness values:

$$\mathrm{sim}(S_1, S_2) = \omega_1 \omega_2 \frac{\mathbf{v_1} \cdot \mathbf{v_2}^T}{|\mathbf{v_1}|\,|\mathbf{v_2}|}\,. \tag{11}$$

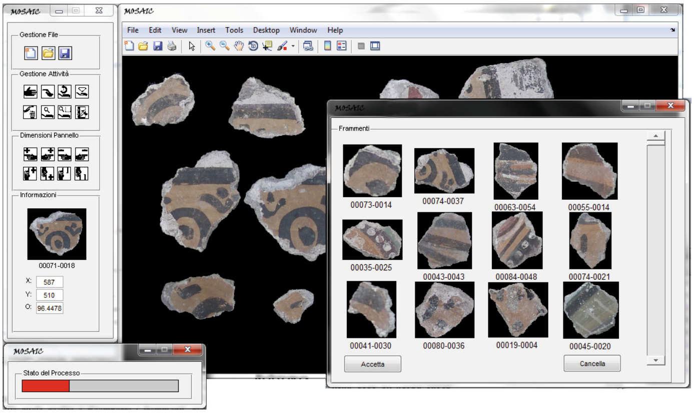
Since each fragment contains more shapes, the similarity between two fragments F_1 and F_2 is given by the maximum shape-to-shape similarity:

$$\mathrm{sim}(F_1, F_2) = \max_{\substack{S \in F_1 \\ T \in F_2}} \mathrm{sim}(S, T)\,.$$

In the most common type of query, the key is represented by a single shape S and the search goes through each fragment indexed in the database, looking for shapes with high values of similarity to S. The similarity score assigned to a fragment is the maximum similarity score achieved by a shape it contains. Smaller shapes are discarded as not relevant. When a query is based on shape, the fragments are returned in decreasing order of similarity (of the included shapes) to S.

3 MOSAIC Interface

MOSAIC offers a graphical user interface (GUI) that allows the operator to create a work session to manipulate the virtual fragments. These can be handled and reconstruction can be performed by retrieving relevant fragments through queries based on shape, color or a combination thereof. The interface is shown in Fig. 5. A work session created from scratch appears as a blank workspace where fragments can be brought in. Here is an outline of the main functionalities offered by the interface.



X: 587
Y: 510
O: 96.4478

Fig. 5. MOSAIC: the graphical user interface and the workspace.

- **Workspace Management.** Fragments can be picked from the database, manually or via queries, and brought into the workspace. Useless fragments can be removed from the workspace.
- **Manipulation.** The operator, possibly an archaeologist or other cultural heritage specialist, can translate or rotate fragments, in a way resembling the real work with physical pieces.
- **Search.** It is possible to query the system by selecting a fragment from the workspace. The query can address a number of features, among which a specific pictorial shape among those represented in the fragment, the whole set of shapes represented, color distribution, texture (both shape and color are relevant), and more.

4 Experimental Results

Given the nature of the MOSAIC system, it is significantly easier to assess its performance qualitatively than quantitatively; therefore, custom experiments have been designed. The main objective was the evaluation of precision/recall achieved by MOSAIC in the retrieval of fragments close to a given one. This can give a measure of the usefulness of the system, since it is related to the intended use of the system by a human operator, who is trying to reconstruct shattered pictorial artworks.

It is reasonable to assume that adjacent pieces often tend to have similar textural features: searching for features similar ti those characterizing the query key fragment should return fragments lying in the same picture area. In a symmetric way, given a key fragment, the user wants to retrieve the fragments that most probably should be put again close to it, as in the original picture: one can reasonably assume that they present a similar texture. Following these considerations, jigsaw puzzle solution can be simplified, by suggesting to the operator the fragments to pick from a smaller returned subset rather than from the full set. In order to verify the effectiveness of the indexing process, each fragment was characterized with the canvas coordinates of its barycenter, computed assuming uniform mass distribution. Obviously, this is not possible in real situations. The main problem to solve is often exactly related to the lack of knowledge of the original position of the fragments. If this is the case, during reconstruction, we might only recover the position of the already identified fragments, but only if we have an image of the original artwork. However this spatial annotation was useful to analyze some issues, and as a kind of ground truth. Given a key fragment A used as a query, let its barycenter be P_A; a fragment B retrieved by the system is deemed to be "close enough"—that is, significant—if its barycenter P_B falls within a circle of radius t_M centered in P_A.

We report here the results of two selected experiments, that were ran on two pictures that were artificially "broken" into fragments. These two experiments have been chosen among the others because the two pictures present different features. The first chosen picture is the *Madonna della Misericordia* or *Madonna delle pietre*, by an unknown author of the late XV century. The original figure is shown in Fig. 6 while its virtually fragmented version is shown in Fig. 7. The original size of the painting is 140 × 90 cm. The acquisition was performed at 300 dpi, 24 bit RGB, yielding 1185 × 1566 pixels. This picture presents a single human figure in the centre, but in some regions of the surface very small details are depicted.

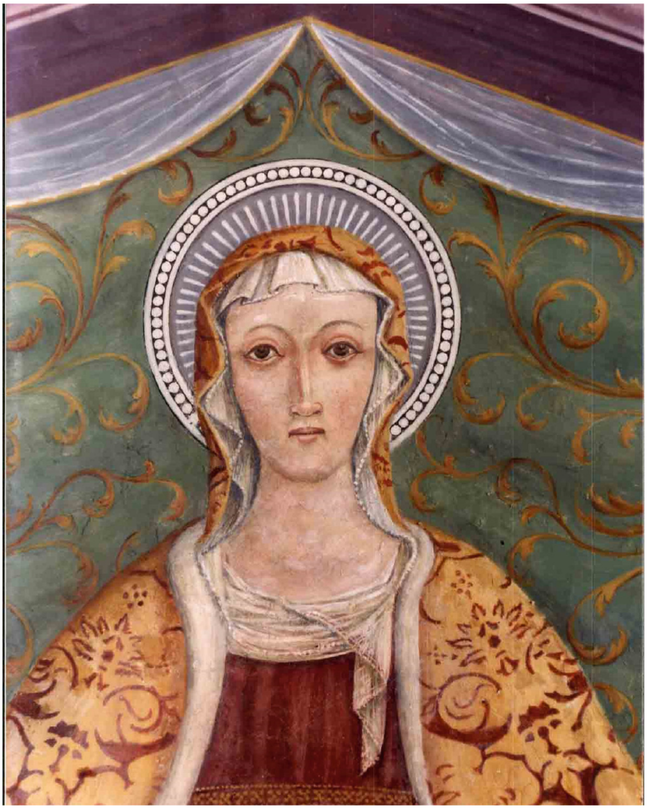


Fig. 6. Madonna della Misericordia: the first image chosen for showing the quantitative performance assessment.

The virtual fragmentation consisted of irregularly shaped small pieces; the smallest ones were discarded to simulate more realistically the true fragmentation and loss that may result from a traumatic event. The 2900 resulting pieces had sizes ranging from roughly 2 × 2 cm to 8 × 8 cm. Such pieces were entered into MOSAIC as they were acquired by a camera and indexed.

The choice of a suitable value for the threshold t_M is strongly dependent on image scaling/resolution. To take this into account, the value of t_M is expressed as the product of a proportionality factor δ in the range $[0, 1]$ times a representative length quantifying the image resolution, namely the diagonal length L_d, in

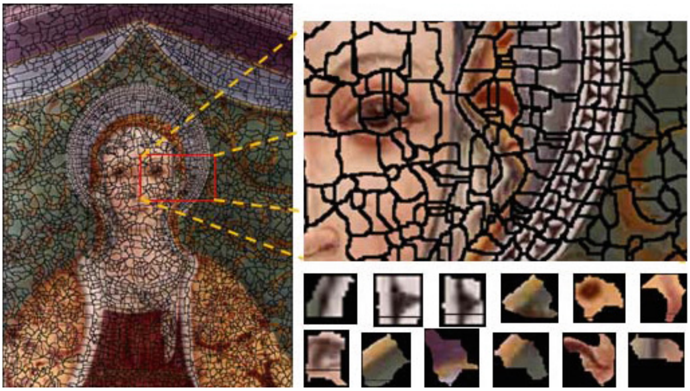


Fig. 7. Madonna della Misericordia: the fragmented version with a detail of some of the fragments.

our case 1964 pixels. Experiments have been performed with $\delta = 0.02$, $\delta = 0.05$ and $\delta = 0.1$, with resulting values of $t_M = 39$, $t_M = 98$ and $t_M = 196$ pixels respectively. Each fragment was used in turn as the key to a MOSAIC query. The average precision and recall curves over all fragments are plotted in Fig. 8.

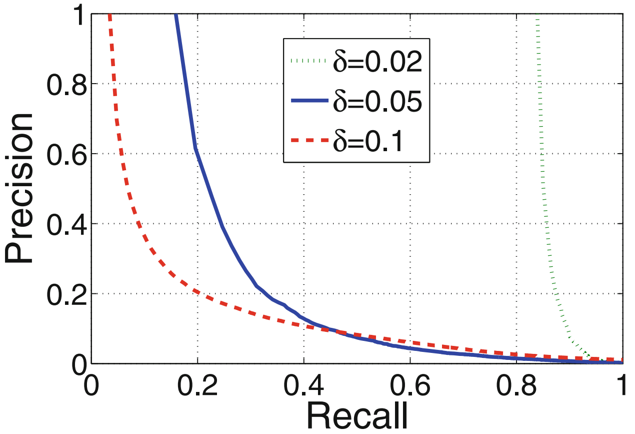


Fig. 8. Madonna della Misericordia: average precision/recall curves for MOSAIC with $\delta = 0.02$, $\delta = 0.05$ and $\delta = 0.1$.

The diagram shows that smaller values of δ generally yield better retrieval: for any given key fragment, the results of interest are usually contained in a relatively small space neighborhood of the key rather than spread around widely. However, looking at Fig. 7, we can notice that there is an intrinsic issue that harms precision and recall: it is represented by the fairly extended regions with

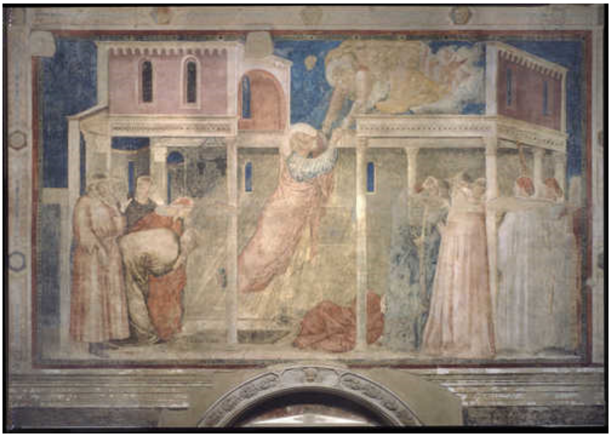


Fig. 9. Giotto's Assunzione di S. Giovanni Evangelista: the second image chosen for quantitative performance assessment.

homogenous features. For example, the whole halo area has nearly homogeneous textural and chromatic features. As a consequence, any two fragments belonging to that area will be "close" in the feature space, while actually lying at a potentially large distance in the reconstructed picture.

The second chosen experiment was carried out on a picture of a fresco by Giotto: "Assunzione di S. Giovanni Evangelista" from Cappella Peruzzi. The processed image is 72 dpi, 24 bit RGB, with size 2560 × 1622 pixels. The same experimental parameters were used, e.g., binarization threshold, color radius, etc., and similar processing considerations were adopted. The original picture is shown in Fig. 9. The virtually fragmented picture is shown in Fig. 10. We can notice the different spatial and chromatic composition with respect to the Madonna: many figures, many architectonic elements, but even wider homogeneous regions. Notwithstanding this, we decided to stress MOSAIC by using the same parameters as above. The results are shown in Fig. 11.

It is interesting to notice that, notwithstanding the higher complexity of the second depicted scene, the general trend of results (i.e., the shape of the curves and their relative positions) remains stable even with the same parameters used for the first picture. This testifies the robustness of the image processing results.

4.1 An Experiment on a Real Life Case

In simulated experiments, it is possible to perform both a qualitative and a quantitative assessment of the retrieval results, since a ground truth is available. This is not always possible in real situations, i.e., in field operations that a system of this kind aims at supporting. One of the main difficulties in many of these cases is just the lack of knowledge of the preexisting appearance of the artwork at hand, i.e., of a reference "model" that can guide the reconstruction. Actually, the

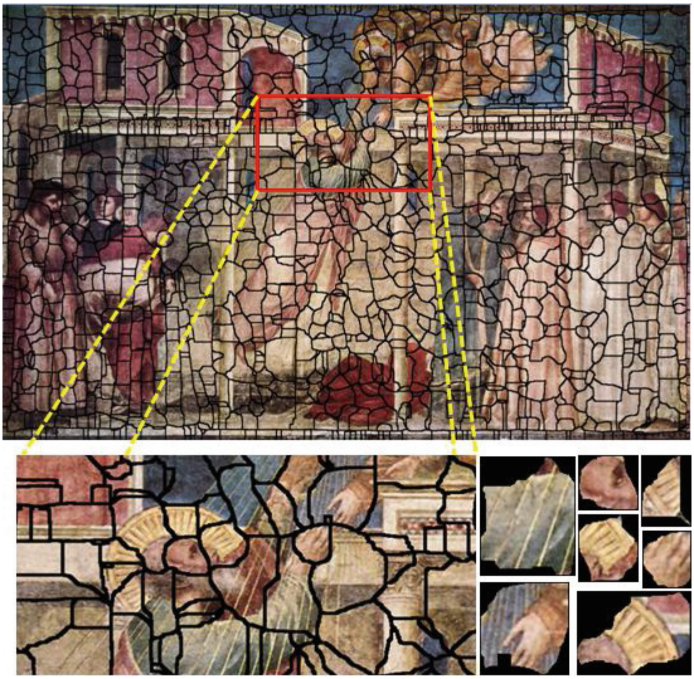


Fig. 10. Giotto's Assunzione di S. Giovanni Evangelista: the fragmented version, with a detail of some of the fragments.

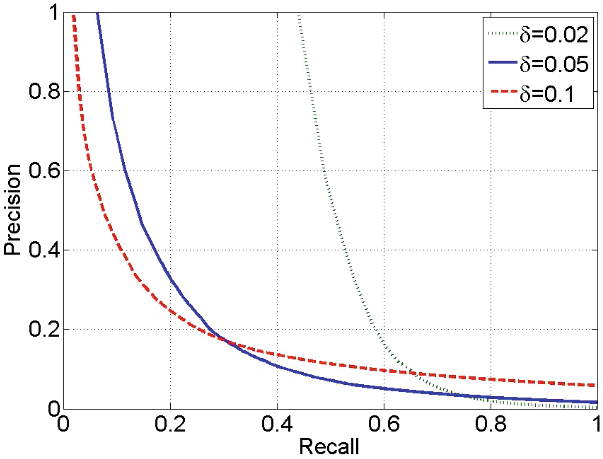


Fig. 11. Assunzione di S. Giovanni Evangelista: average precision/recall curves for MOSAIC with $\delta = 0.02$, $\delta = 0.05$ and $\delta = 0.1$.

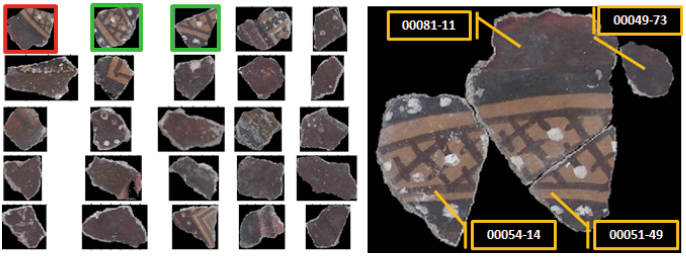


Fig. 12. A query, its result, and a partial reconstruction made from some of the fragments returned.

MOSAIC system has also been tested in such a kind of real case study, when only a qualitative evaluation is possible, since the original appearance of the frescos was unknown. The experiment involved the reconstruction of a set of 6419 fresco fragments that were found during a restoration work at the St. Trophimena Church site in Salerno. Since no information was available about the original appearance of the frescoes, virtual reconstruction was the only option to recover at least parts of the original work without further damaging to the fragments. Examples from a work session are shown in Fig. 12, depicting a reconstruction in progress, and Fig. 13, illustrating a shape-based query.

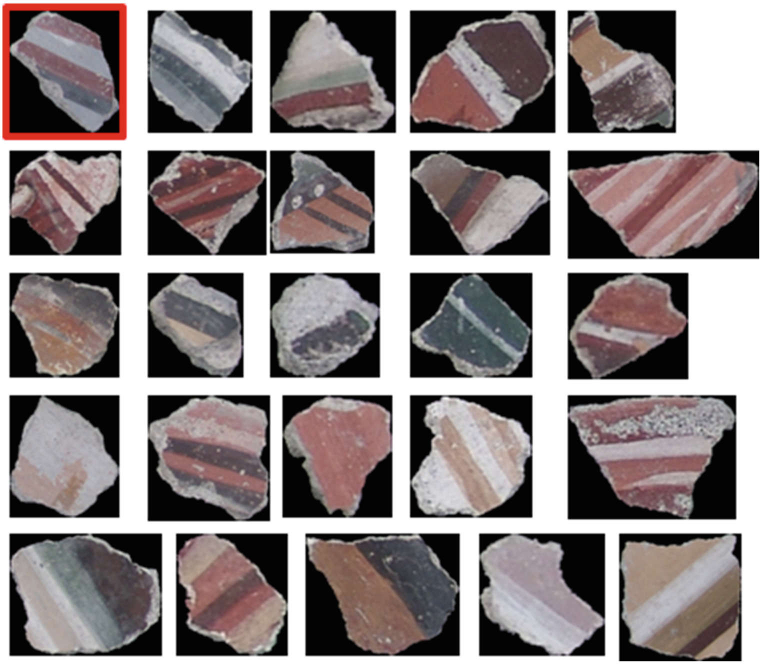


Fig. 13. Query by shape: strips.

As anticipated, the lack of information about the “real” solution to this jigsaw puzzle makes it impossible to obtain an objective/quantitative measure of the quality of the retrieved set of fragments provided by MOSAIC, but a qualitative assessment of its effectiveness is possible. The end users were involved both in the interface design phase and in the experiments, and their detailed feedback confirmed the practical value of MOSAIC system.

In Fig. 12, each label has two values. The first is the tray where the fragment lies, while the second is the serial number inside that tray. As the figure shows, it is quite usual that pieces close to each other in the original picture can end up in quite distant trays when they are picked up from the original site or during cataloging.

5 Conclusions

The reconstruction of fragmented pictorial artworks is a task always very demanding and often unfeasible, depending on the conditions of the fragments and on the experience of the operator. MOSAIC (Multi-Object Segmentation for Assisted Image reConstruction) is a system aiming at supporting this delicate work by allowing the computer aided reconstruction of virtual fragments. The extraction of relevant features related to color and shape allows cataloging and indexing of the fragments. Queries can be formulated through a simple GUI. Different kinds of selection of the query key are available: it is possible to select a whole fragment as example of the searched companions, or even a single shape represented on a fragment. The results of the comparison with the stored virtual fragments are sorted by similarity to the query key, and provide candidates for puzzle solving in the area of the relevant fragment. This can speed up the reconstruction process significantly, and improve the quality of the result. The system has been tested first via computer simulations. We report here two examples of such simulations. In both cases the solutions were known a priori, so to have a ground truth to assess the retrieval results. Later on the system has also been tested in a real world situation, where the solution was unknown and therefore results can be appreciated only qualitatively. It is interesting to notice the consistent results obtained on the two simulations, notwithstanding the different complexity of the pictures and the use of exactly the same parameters. This assesses the robustness and flexibility of MOSAIC procedures. Domain experts participated in both the design and testing phase of MOSAIC, and have provided precious feedback for tuning the system parameters. Future work will entail more detailed interaction with restorers, archaeologists, and other cultural heritage operators to better understand their needs, the problems related to specific situations, and offer them and even better support.

References

1. Birchfield, S.T., Rangarajan, S.: Spatiograms versus histograms for region-based tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1158–1163, Jun 2005

2. Brown, B., Laken, L., Dutré, P., Gool, L.V., Rusinkiewicz, S., Weyrich, T.: Tools for virtual reassembly of fresco fragments. In: Proceedings of the 7th International Conference on Science and Technology in Archaeology and Conservations. pp. 1–10. SCITEPRESS (2010)
3. Brown, B., Toler-Franklin, C., Nehab, D., Burns, M., Dobkin, D., Vlachopoulos, A., Doumas, C., Rusinkiewicz, S., Weyrich, T.: A system for high-volume acquisition and matching of fresco fragments: reassembling theran wall paintings. ACM Trans. Graph. (Proc. SIGGRAPH) **27**(3), 1–10 (2008)
4. Cho, T.S., Avidan, S., Freeman, W. T.: A probabilistic image jigsaw puzzle solver. In: CVPR, pp. 183–190. IEEE (2010). http://dblp.uni-trier.de/db/conf/cvpr/cvpr2010.html#ChoAF10
5. Chung, M.G., Fleck, M., Forsyth, D.: Jigsaw puzzle solver using shape and color. In: Proceedings of the 4th International Conference on Signal Processing (ICSP 1998). vol. 2, pp. 877–880 (1998)
6. Comaniciu, D., Meyer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **24**(5), 603–619 (2002)
7. Demaine, E.D., Demaine, M.L.: Jigsaw puzzles, edge matching, and polyomino packing: connections and complexity. Graphs Combinatorics **23**(Supplement), 195–208 (2007). (special issue on Computational Geometry and Graph Theory: The Akiyama-Chvatal Festschrift)
8. Freeman, H., Garder, L.: Apictorial jigsaw puzzles: the computer solution of a problem in pattern recognition. IEEE Trans. Electron. Comput. **2**(EC–13), 118–127 (1964)
9. Hu, M.: Visual pattern recognition by moment invariants. IRE Trans. Inf. Theor. **IT–8**, 179–187 (1962)
10. Mercimek, M., Mumcu, K.G.T.V.: Real object recognition using moment invariants. Sadhana Acad. Proc. Eng. Sci. **30**(6), 765–775 (2005)
11. Nielsen, T.R., Drewsen, P., Hansen, K.: Solving jigsaw puzzles using image features. Pattern Recogn. Lett. **14**(29), 1924–1933 (2008)
12. Papaodysseus, C., Panagopoulos, T., Exarhos, M.: Contour-shape based reconstruction of fragmented, 1600 bc wall paintings. IEEE Trans. Signal Process. **6**(50), 1277–1288 (2002)
13. Sagiroglu, M., Ercil, A.: A texture based matching approach for automated assembly of puzzles. In: Proceedings of the 18th International Conference on Pattern Recognition (ICPR 2006). pp. 1036–1041 (2006)