

Chapter 12

Virtual Restoration of Archaeological Artifacts

Ilan Shimshoni and Ayellet Tal

Abstract The task of restoring the look of an artifact as it used to be thousands of years ago, is a fascinating task, for a variety of reasons. The first reason is the very nature of the task of generating information that is not in the input. Second, the condition the artifacts are found—eroded, broken and noisy—differ from those of natural images in two dimensions or smooth geometries in three dimensions. Finally, datasets from which algorithms may learn, are rarely available; and even when available they are small. These aspects make state-of-the-art general algorithms prohibitive in the archaeological domain and special methods need to be developed. This chapter focuses on four sub-problems of restoration: reassembly, reconstruction of artifacts from drawings, re-colorization, and ancient document enhancement. It reviews our recent advances in terms of methods and results.

12.1 Introduction

Archaeology provides a unique window into the past and offers valuable insights into the history and culture of humanity. It enables us to delve into and acquire knowledge about societies that have ceased to exist, their cultures, and their accomplishments. Additionally, it has the potential to contribute to scientific advancements by offering insights into various areas like climate change, technology, the development of human societies, and even the downfall of ancient civilizations.

It is an exhilarating era in terms of technology, with transformative changes taking place across the world. However, certain fields have been left lagging behind, and archaeology is one of them. This is regrettable, considering the immense potential technology holds to incorporate archaeology into the ongoing technological

Ilan Shimshoni
The University of Haifa, e-mail: ishimshoni@is.haifa.ac.il

Ayellet Tal
Technion, e-mail: ayellet@ee.technion.ac.il

revolution and enhance the scientific investigation of the past with improved efficiency. Specifically, advancements in computer vision and computer graphics can be leveraged to automate certain aspects of the study of archaeological artifacts.

From the perspective of computer vision and graphics, the tasks performed by archaeologists studying sites with a focus on small artifacts can be categorized into four main areas:

1. **Excavation:** This involves the physical act of digging and uncovering the artifacts.
2. **Documentation:** It entails the creation of textual and visual records of the findings, which are typically documented in books and site reports.
3. **Localization:** This involves determining the temporal and spatial context of the findings through various means such as visual comparisons, chemical tests etc.
4. **Restoration:** This refers to the process of restoring the artifacts to their original state before any damage occurred.

It is worth noting that the latter three tasks are fundamental challenges in computer vision & graphics. In particular, the visual aspect of documentation can be interpreted as edge detection, non-photo realistic rendering or even domain transfer [13, 14, 24, 25, 26, 34, 54]; The visual component of location may be viewed as classification and retrieval [2, 3, 6, 7, 14, 29, 35, 39, 44, 49, 52]; Restoration, which is the focus of this chapter, will be discussed in detail below.

Nonetheless, the application of cutting-edge computer vision techniques to the field of archaeology is prone to significant challenges and may not yield satisfactory results. This can be attributed to two primary factors. First, archaeological artifacts typically lack clean and well-behaved geometries. Rather, they are often broken, noisy, and extremely eroded. Second, the availability of massive datasets, which is crucial for training deep learning algorithms, is limited in the domain of archaeology. This is due to two problems: the number of existing artifacts and that these artifacts usually have to be processed or labelled by experts, such as conservators or archaeologists who specialize in the specific domain. Without an ample supply of labeled data, deep learning approaches that heavily rely on large datasets may not be effectively utilized.

The primary focus of this chapter revolves around the fourth task, namely restoration, which aims to revive the original appearance of damaged artifacts. Within the scope of restoration, this chapter provides an overview of our research efforts in four distinct sub-tasks: re-assembly, reconstruction of artifacts from drawings, re-colorization, and document enhancement. In the subsequent sections, we delve into each of these sub-tasks.

Reassembly. The objective is to reconstruct a broken artifact by accurately assembling its fragmented pieces, similar to solving a puzzle. [9, 10, 45, 48]. Currently, the prevailing practice in archaeology involves manual reassembly, which is a painstaking and time-consuming process, where pieces are meticulously glued together one by one. Instead, our aim is to automate this intricate puzzle-solving task. Unfortunately, even the “easy” variants of puzzle solving, such as rectangular pieces of equal size, are NP hard. Our case is more difficult, as the fragments come in all sizes and shapes, their boundaries are eroded and do not align perfectly, and there may be

missing pieces. Within this general problem, reassembly might be attempted on different input domains, such as frescoes, papyri, ceramic objects, and even statues. In each case a different type of matching transformation between the pieces is required. In Section 12.2 we will discuss the recent advancements in this field, on the first two types of objects, discussing the progress made towards automated reassembly techniques.

Reconstruction of artifacts from drawings. Oftentimes, artifacts are securely stored within museum and storage rooms, with access limited to their drawings found in site reports and books. These drawings, created by archaeological artists during the documentation process, serve as the sole means of visualizing these artifacts. The challenge lies in the reverse engineering process, where the goal is to generate three-dimensional virtual objects based on these drawings [23, 57, 59]. Section 12.3 elaborates on this task.

Re-colorization. A prevailing belief regarding the great statues of ancient Greece and Rome is that they were originally crafted from pure, unpainted stone. However, researchers have been arguing that these statues used to be quite alive, vibrant, and full of color. Unfortunately, due to centuries of deterioration and subsequent cleaning processes, any remnants of pigments that may have existed on the statues have been lost. This poses the question: can we restore the original appearance of these scanned statues? In Section 12.4 we present methods for doing so [27, 28].

Document enhancement. Ancient transcript might be damaged due to stains, ink spills or natural degradation that occur over time. Our objective is to automatically extract the text from these transcripts [22, 50, 53, 55]. Specifically, our aim is to *binarize* the image pixels, distinguishing between background and foreground (text) pixels. Section 12.5 presents some results.

It is worth noting that Some of the methods presented in the following sections may differ in terms of the input representation. Some approaches assume that the input consists of images of artifacts, while others assume that the objects are three-dimensional and captured through scanning processes. The choice between these representations depends on the available data.

12.2 Reassembly

Reassembling fragmented artifacts is a fundamental and essential challenge in archaeology. Frequently, artifacts discovered during archaeological excavations are found in a broken state, with both structural and visual degradation. Additionally, there may be missing fragments, further complicating the reassembly process. Consequently, considerable effort and time are dedicated by expert conservators to manually reconstruct these artifacts with utmost precision and completeness.

In computer vision, reassembly can be seen as a specialized form of puzzle solving. The task becomes particularly challenging due to the complexity of the inputs described earlier. When tackling the reassembly problem, it can be divided into two main stages. The first stage involves matching, where the goal is to find

the relative geometric transformation between two adjacent fragments and assign a confidence score to each potential match. This matching process serves as a fundamental step within the broader placement algorithm, which aims to reassemble the artifact from its fragments. When considering reassembly problems, they can be characterized in different ways. Firstly, the matching process can be based on the shape and/or appearance of the fragments. However, in cases where the fragments have undergone geometric degradation, shape matching may not be straightforward. The second criterion for reassembly characterization is the type of local matching transformation employed. It can be discrete or continuous. In the continuous case, the transformation can be a 2D translation or a general 2D rigid transformation, typically applied to 2D artifacts. For 3D objects, a general 3D rigid transformation is utilized. In certain special cases, such as surfaces of revolution (e.g., jars, jugs, or vases), more limited 3D transformations are applicable.

In this section, we will review two of our works on the subject. The first work focuses on the reassembly of colored 2D fragments, specifically addressing the reconstruction of artifacts like frescoes [9]. The second work deals with matching papyrus fragments [1].

12.2.1 Reassembly of colored 2D fragments

Given N images, each capturing a single fragment of an artifact, our goal is to reassemble the artifact. In particular, each fragment should be assigned a 2D rigid transformation that indicates how to place it within the re-assembly. Our proposed algorithm consists of four steps, as illustrated in Fig. 12.1. We briefly discuss each of the steps below.

Fragment Extrapolation. In the first step, each fragment is extrapolated. This serves two goals. First, it shows how the fragment used to look before it was damaged and eroded. Second, it reduces the problem of continuity from one fragment to another into the problem of matching pieces within the extrapolated region. Formally, given a fragment, the goal is to produce a k -pixel wide band around it. The algorithm extrapolates the image using patches from the original image, utilizing patch-based optimization. This optimization takes into account not only colors and image gradients, but also geometric and photometric transformations, which is vital for our case.

Sampling a set of valid transformations. The objective of this stage is to sample the space of transformations between a given pair of fragments. When sampling these transformations, three conditions need to be satisfied. First, the transformations must be valid. In our context, a valid transformation is one that brings the fragments closer to each other, increasing the likelihood of adjacency, without causing overlap. Thus, we require that the original pixels of the fragments do not overlap, but the extrapolated pixels of the fragments do overlap. Second, it is important to ensure that the number of sampled transformations is feasible. To address this, we demonstrate how the set of valid transformations can be represented as a configuration space

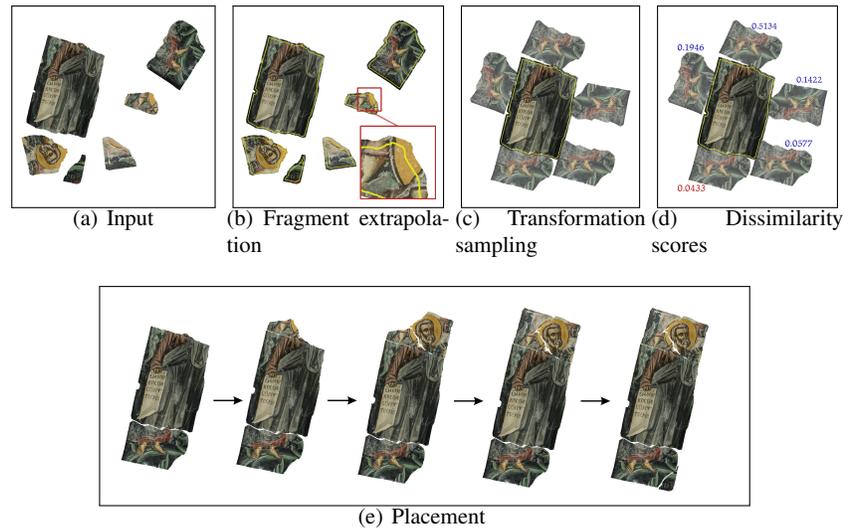


Fig. 12.1 Reassembly algorithm outline

problem, similar to those used in robotics. This allows for easier sampling within this space. Finally, due to the sampling process, some of the initially promising transformations may be imprecise. To refine these transformations, a fine-tuning approach is necessary. In our algorithm, we this fine-tuning is performed during the placement stage.

Finding good matches. Computing a dissimilarity score for a given transformation between a pair of fragments lies at the core of re-assembly. Thanks to Step 1, the continuity problem was reduced to matching the extrapolated region of one fragment and the other original fragment, on the overlapping region. The strongest cues for matching are the colors and the color edges, which should continue from one fragment to its adjacent piece. Matching becomes challenging in our domain due to three additional difficulties: (1) faded colors (2) imprecise transformations (due to sampling) that cause misalignment of the edges and (3) degradation that may yield spurious edges that should be ignored.

Our algorithm handles these problems. However, even when the dissimilarities are computed, our problem is not yet solved. This is so since a low dissimilarity value does not necessarily imply confidence in a match, as other matches with the same fragment may be almost as likely. We propose to compute a confidence score, which predicts the likelihood of the pair of fragments to be neighbors. The confidence in the match takes into account, in addition to the dissimilarity score, also the scores of competing matches in the same region of the fragment.

Placement. In line with previous works, our placement strategy follows a greedy approach, where fragments are added iteratively, one at a time. However, our algorithm incorporates new considerations to address additional complexities arising from the

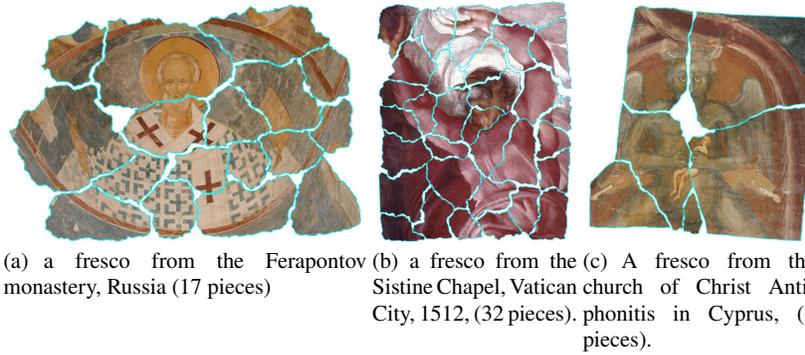


Fig. 12.2 Fragments are re-assembled by our algorithm. The borders of the fragments are marked in cyan. The fragments are faded, stained, scratched, and the boundaries do not accurately match.

continuity of the transformation space and the shape of the fragments. To handle the challenge of imprecise transformations, the selected transformation is refined within a small neighborhood around it. For a visual demonstration of our placement algorithm, we refer the reader to its visualization available on YouTube [8].

Several results of running the algorithm on frescoes are shown in Fig. 12.2.

Limitations. If the gaps between fragments are substantial, the algorithm may encounter difficulty in accurately reconstructing the fresco. This is due to the potential mismatch of neighbors on either side of the gap.

12.2.2 Matching papyrus fragments

In this work we proposed a method for matching and assembling pairs of ancient papyrus fragments containing mostly unknown scriptures. Papyrus paper is manufactured from papyrus plants and therefore portrays typical thread patterns resulting from the plant's stems. The proposed algorithm is founded on the hypothesis that these thread patterns contain unique local attributes, such that nearby fragments show similar patterns reflecting the continuations of the threads. We posit that these patterns can be exploited using image processing and machine learning techniques to identify matching fragments. The algorithm and system support the quick and automated classification of matching pairs of papyrus fragments as well as the geometric alignment of the pairs against each other.

Since we have no ground truth of previously matched fragments, we synthetically generate a suitable training set with label. For this purpose, we utilize three relatively large scroll pieces from a nearly complete papyrus scroll as the basis for a training and validation set. We start by digitally tearing the three pieces into three distinct sets of small fragments. We train the algorithm on two of the sets and attempt to match

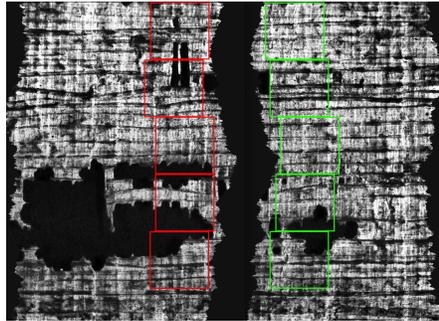


Fig. 12.3 An example of two adjacent artificially torn fragments (from plate 303) with a set of candidate squares to be used in the matching phase

the third set's fragments. We perform the tearing in several randomized iterations while altering the number of fragments and their shape in each iteration.

The algorithm consists of a series of steps and is based on deep-learning and machine-learning methods. The first step deconstructs the problem of matching fragments into a smaller problem of finding thread continuation matches in local edge areas (squares) between pairs of fragments. Due to the local nature of the thread pattern similarity, we focus on seeking matches only within squares that are cropped from the edges of the fragments. In order to produce the squares we slide a "crop window" of 250×250 pixels along the edges of each fragment and crop corresponding square images at regular intervals. Fig. 12.3 shows an example of the traversal of the "crop window" and the resulting squares.

Each pair of squares is given as input to a convolutional neural network, yielding local matching scores. During training we take into account that matching pairs might not be accurately aligned and due to the tearing between the fragments might be at a certain distance from each other. The result of this stage yields very high recall but low precision.

To deal with this problem we combine the results of the first step by verifying them geometrically. For each match a suggested relative translation between the fragments is produced. Since it is not accurate, it votes for a rectangle in the translation space, as can be seen in Fig. 12.4. Correct matches will generate a region in the translation space with a large number of overlapping rectangles.

We enhance this voting with geometric alignment techniques from which we extract additional spatial information. Eventually, we feed all the data collected from these steps into a Random Forest classifier in order to produce a higher order classifier capable of predicting whether a pair of fragments is a match.

Our algorithm was trained on a batch of fragments which was excavated from the Dead Sea caves and is dated circa the 1st century BCE. The algorithm shows excellent results on a validation set, which is of a similar origin and conditions. An example of these results can be seen in Fig. 12.5. However, on a test batch, which is considered extremely challenging due to its poor condition and the small size of its fragments, the performance is sub optimal, returning a relatively large ratio of false

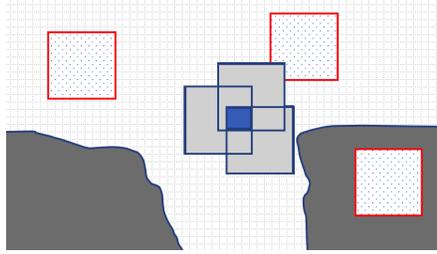


Fig. 12.4 An illustration showing edges of two potentially matched fragments and squares representing the fragments' proposed relative position alignment as voted by their matching square pairs. Three of the squares in this illustration overlap each other, thereby creating a strongly supported alignment area, while the three others are not supported by other votes and are therefore not considered in the alignment process nor as part of the spatial features used for classification.

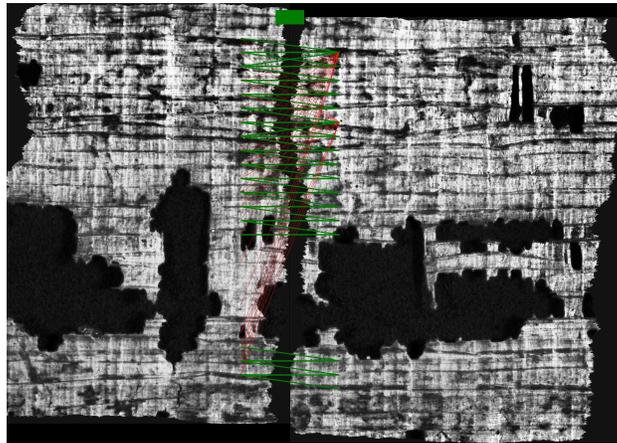


Fig. 12.5 An example of a pair of matched fragments (from plate 303) from the validation set showing the concatenation based on the relative positioning as well as lines connecting between the centers of the supporting matching squares on each fragment. Green lines indicate well supported squares while red lines are not supported. The green square at the top reflects the highly voted area for the relative alignment point.

positives. Nevertheless, the algorithm was quite useful by eliminating 98% of the possible matches, thus reducing the amount of work needed for manual inspection. Indeed, experts that reviewed the results have identified some positive matches as potentially true and referred them for further investigation.

Limitations. The geometric matching component depends on the size of the fragments and for relatively small fragments the algorithm struggles to find the correct matches. In addition, multi-spectral images (which exist) could be used as the input to the algorithm, which might improve its results.

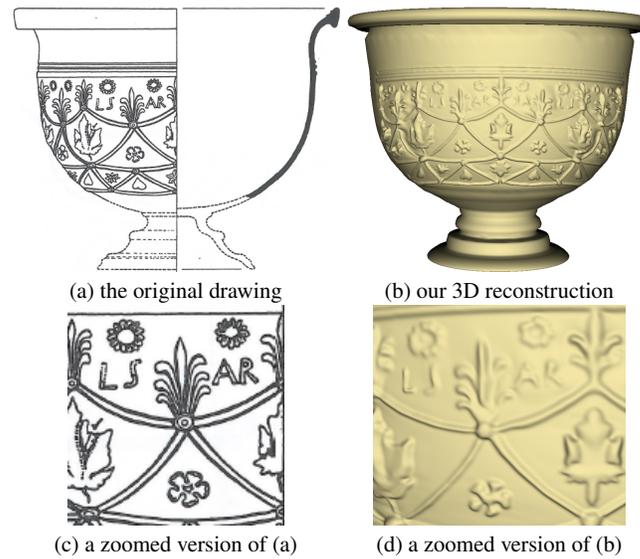


Fig. 12.6 Reconstruction of a Roman vase from a manual drawing consisting of 571 curves.

12.3 Reconstruction of artifacts from drawings

Line drawings have been the standard method of archaeological artifact documentation for many years. While many findings might have been lost or destroyed, their illustrations remain. Therefore, the only existing inputs for reconstruction are the line drawings appearing in an archaeological report (Fig. 12.6(a,c)). Usually, production of such drawings is derived from the analysis of an expert archaeologist. The archaeologist translates into a drawing his interpretation of the object. So our goal is not to recreate a digital replica of the object itself, but to recreate a missing object as it was perceived by the scientist who drew the archaeological report (Fig. 12.6(b,d)). These inputs are usually highly complex. State-of-the-art algorithms for automatic reconstruction were not designed with such drawings in mind. In our work [23, 24], we focus on artifacts containing reliefs. They are important sources of information, since they are fingerprints of periods and cultures. We applied our algorithm to real examples from archaeological reports of two types: Hellenistic and Roman relief pottery [5] and Cosa lamps [11]. Our approach solves two subproblems: reconstruction of the base and reconstruction of the details (the relief) on top of the base.

Base Estimation. Given the drawing of the artifact (Fig. 12.7(a)), its silhouette is first extracted. In some cases more than one view exists and they are also used. The silhouettes are then compared to a large number of 3D objects from a database, as seen in Fig. 12.7(b). The object with the smallest L_2 distance is chosen. It is seldom the case that the database contains a base model that perfectly fits the drawing, as can be seen in Fig. 12.7(c). Therefore, we deform the retrieved model to better match the drawing. The key idea of our algorithm is to move the points on the orthographic

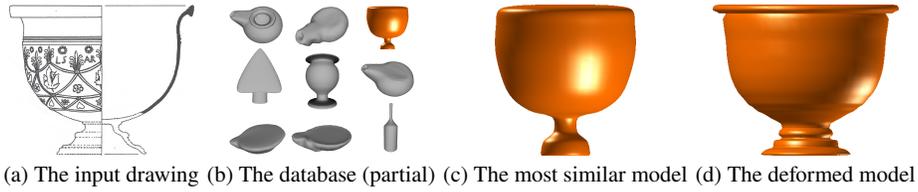


Fig. 12.7 Base estimation algorithm. Given a line drawing (a) and a database (b), our algorithm first retrieves the most similar model from the database (c) and then deforms it to better suit the line drawing (d). In this example, we used a database of 1830 models.

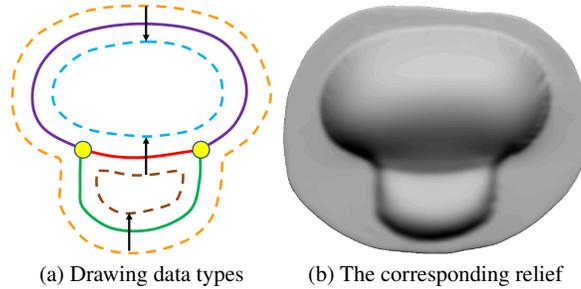


Fig. 12.8 Notations. Curves are solid lines, junctions are yellow circles, margins are dashed lines, and step edge directions are arrows. Each curve and margin is drawn in a different color.

projections to their matched points on the drawing and move the other points of the model so as to preserve its shape. The final result can be seen in Fig. 12.7(d).

Relief Reconstruction. For most relief objects, the details can be described as a height function defined on a surface, termed the base. The lines of drawings of relief objects typically indicate changes of the height function. This height function is usually smooth far from the lines and constant along them. Its gradient is strongest near the lines in the direction perpendicular to the lines. Hence, the value of the height function in the line's neighborhood depends only on the distance from the line. A drawing consists of drawing curves, junctions, and margins (Fig. 12.8). We define the drawing curves, or simply the curves, as the lines of the given drawing. They indicate visually-meaningful locations on the relief. The curves may be connected by junctions, but cannot cross them. Margins are the borders of a curve's neighborhood. A drawing defines the relief (the height function). Outside the margins, the relief is assumed to be smooth. Inside the margins the relief is approximated by a step edge of height h and width w . Height h can be positive or negative, representing the direction of the edge. The shape of the relief defined by step edge $s(x)$ is $p(u)$:

$$p(u) = h \cdot s(u/w), \quad (12.1)$$

where $u \in [-w, w]$ is the width parameter of the step edge. The sought-after surface is a combination of the base B and the relief. See Fig. 12.9 for an illustration.

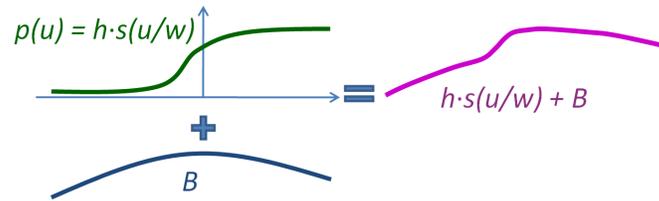


Fig. 12.9 The relief in a curve's neighborhood is a combination of the step edge $p(u)$ & the base B .

Interpreting the line drawing requires setting the height of the step edge of each curve. Doing this manually for a complex drawing composed of dozens, or even hundreds of curves is a cumbersome process. Moreover, the set of assigned heights has to be consistent. Consistency refers to the requirement that two different paths between points should result in the same height difference. Interestingly, the key idea of our method for computing the heights of the step edges is to reduce this problem to the problem of a constrained topological ordering of a graph.

Reconstruction Given a base, a set of curves, and the heights of their step edges that were computed above, the goal is to reconstruct the relief surface. The output of the algorithm is a height function defined for every point on the base, yielding the resulting surface. The main problem the algorithm addresses is the sparsity of the input curves. This is approached by producing a smooth interpolation in regions in which curves do not exist. The interaction challenge requires that the reconstructed surface will not contain undesired artifacts due to interactions between close curves.

To achieve this goal, we require that the reconstructed height function satisfy two constraints. Locally, the relief in the curve's neighborhood is defined up to a constant using only the step edges of the margins. Globally, the relief should be the smoothest function that coincides with the relief obtained locally for every curve. During the reconstruction, the algorithm should compute for each curve its constant. This is done using the Laplacian operator. Let us denote by Δ the Laplacian operator on a scalar function. We are seeking the Laplacian L of the required surface R , which is a scalar function defined on the base surface. L should be equal to the Laplacian of relief in the neighborhoods of the curves and its Laplacian $\Delta(L)$ should be zero elsewhere. Solving this linear set of equations yields the resulting surface. Fig. 12.10 illustrates a couple of results of our algorithm.

Limitations An underlying assumption of the algorithm is that the surface can be accurately portrayed by a step edge in the line's neighborhood. When this assumption is violated, the relief cannot be reconstructed automatically, and some user interaction is needed.



Fig. 12.10 Reconstruction of two artifacts. (a) Hellenistic relief large light-brown krater (Found in V. Mrdakovica-Croatia, [5] catalog number 1.) The drawing consists of 1300 tightly interconnected lines. (b) Roman triangular nozzle lamp of Eros, A.D. 50-ca 100. [11] catalog number 504.

12.4 Re-colorization

Researchers have been arguing that classic statues and buildings of ancient Greece and Rome were not pure unpainted stone or green tarnished bronze, but rather quite alive, vibrant, and full of color [4]. The colors were created using mineral and other organic materials. Unfortunately, after centuries of deterioration only little traces of pigments remained, yet chemical analysis can often estimate the original color.

Since the number of colors and hues used by the artists was limited, re-colorization algorithms may be able to restore the look of the scanned statues. Recolorization is defined as a computer-assisted process for adding color to monochrome visual data. It was originally defined for black & white images [30, 43]. However, in order to handle scanned ancient statues, the problem was extended to three-dimensional objects [27], as described hereafter.

The interactive colorization process begins with an artifact represented by a polygonal mesh. The user initiates the process by applying a few color strokes on the mesh surface. Based on these initial scribbles, the system proceeds to colorize the entire mesh accordingly. The user has the option to further refine the colorization by adding more scribbles, and the system updates the colorization in response to these additional inputs. Notably, the scribbles do not require high precision and can be loosely placed. An example session, as depicted in Fig. 12.11, showcases a typical process that can be completed within a few minutes. The algorithm that realizes this process proceeds in three steps:

1. Initialization: For each face the scribble passes through, the closest vertex is colorized with the color of the scribble.
2. Vertex similarity. A similarity measure between neighboring vertices is computed. This is done by seeking a descriptor that robustly characterizes the local geometry of the vertex. In [27] we propose to use a variation of *spin images* [19] as vertex descriptor. Then, a similarity measure between vertex descriptors, which must be robust to small changes in the mesh, is computed. We use the *diffusion distance* [32] as a similarity measure [32].

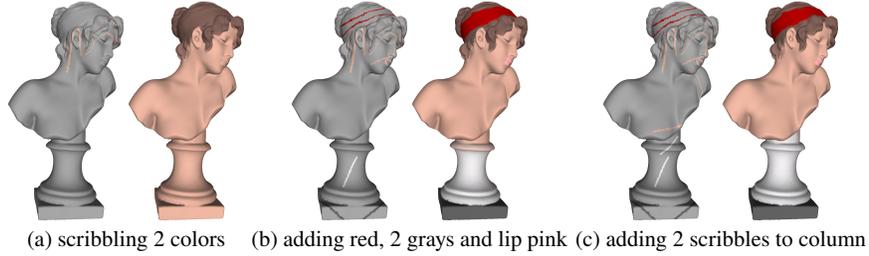


Fig. 12.11 A typical colorization session. (a) The user starts by scribbling two colors. (b) Then, the user adds red to the head band, light gray to the column, dark gray to the base and pink to the lips (middle). (c) Finally, the user fine-tunes the results by adding color scribbles close to the column boundaries, to prevent color bleeding.

3. Propagation: Given these similarities, the algorithm propagates the colors to the remaining vertices of the mesh. The basic assumption is that adjacent vertices that have similar descriptors should get similar colors. This stage is naturally the key of the algorithm. It is done through optimization, in which we impose the constraint that two neighboring vertices should have similar colors if their geometry is similar.

In particular, for all the vertices of mesh M , we minimize the difference between the color at vertex v_i and the weighted average of the colors at neighboring vertices $v_j \in N(v_i)$. This is done for each color channel C in the YUV color space, by minimizing the following loss:

$$\Omega(C) = \sum_{v_i \in M} \left(C(v_i) - \sum_{v_j \in N(v_i)} k_{ij} C(v_j) \right)^2. \quad (12.2)$$

In this equation, k_{ij} is a weight function, which is large when the descriptor of v_i is similar to that of v_j and small otherwise, and for which $\sum_{v_j \in N(v_i)} k_{ij} = 1$. It is defined as

$$k_{ij} \propto e^{s_{ij}^2 / 2\sigma_i^2}, \quad (12.3)$$

where s_{ij} is the similarity of v_i and v_j , as computed in the second stage and σ_i is the standard deviation of the similarity values between each of the vertices in $N(v_i)$ and v_i .

The output is a mesh in which every vertex has a designated color. We note that to prevent color bleeding we define a new vector field on meshes, which is in accordance with the valleys of the object. Fig. 12.12 demonstrates some of the results.

A drawback of the above algorithm is that it does not handle repetitive patterns. As a result, to colorize an objects with patterns, color strokes on each pattern instance are needed, which might be very time consuming. In [28], a method for colorization, which performs also pattern classification, is presented.

The key idea is that the few strokes, which are provided for colorization, can be extremely helpful also for pattern classification. Thus, once the patterns are classified,



Fig. 12.12 Only a few scribbles are needed to colorize the cloth, despite its folded structure. Note how the fingers are correctly separated from the folded cloth or the hand-held objects.

even highly complex surfaces can be automatically colorized using the user's strokes around and within a single instance of each pattern. The interactive session is similar to the one described above. However, the user needs scribble only on and around one instance of the pattern. The approach consists of three steps, which are performed for each pattern in turn.

1. *Pattern-aware filtering.* The surface is filtered in a pattern-preserving manner. In the filtered surface, the pattern boundaries are kept intact, whereas the details in the pattern region are filtered out. This is done by first segmenting the sub-surface into its foreground (a single pattern instance) and its background, by the colorization technique of [27], using the user's scribbles. Then, a uniform Laplace smoothing is applied. The result of this stage facilitates subsequent steps, where sharp pattern boundaries are important for achieving the desirable colorization.
2. *Pattern-driven region descriptors.* Each vertex is associated with a descriptor that characterizes its neighborhood, in accordance with the pattern to be colorized. In particular, for each vertex its surrounding region is first extracted, which is segmented into foreground (pattern) and background. Then, two types of descriptors of the foreground region are considered: region-based and boundary-based. The region-based descriptor captures the geometry of the region, whereas the boundary descriptor relies only on the shape of the region's boundary. Utilizing two different descriptors not only characterizes the region better, but is also essential for the co-training described next.
3. *Pattern classification.* The aim is to determine which vertices of the surface belong to the pattern and which do not. The key idea is to classify a subset of the classified vertices to automatically produce additional colorization strokes in accordance with the pattern. These strokes are the input to [27]. More specifically, for each face the scribble passes through, the closest vertex gets the color of the scribble. The result is a very small number of training examples. To handle shortage in training examples, the set of positive examples is enriched by using the foreground vertices found in Step 1. We face a two-class classification problem, for which

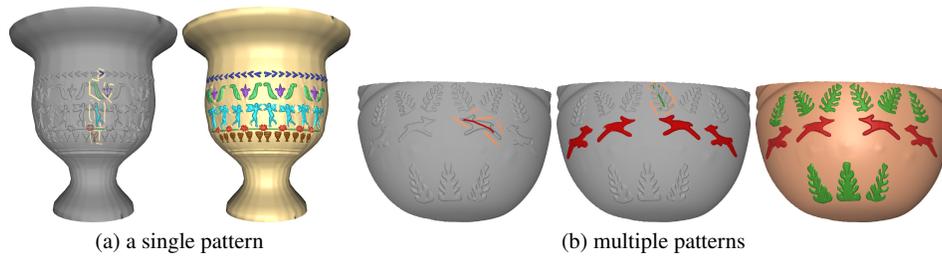


Fig. 12.13 Pattern-driven colorization. Given a 3D surface, the user scribbles a handful of strokes within and around a single pattern instance. The algorithm colorizes the whole surface accordingly.

there is a large number of positive examples and a small number of negative examples. Fortunately, classification of all the vertices can be avoided, as the goal is to classify only a representative subset of vertices with very high confidence (as additional strokes), which will suffice for colorization. In the proposed method semi-supervised learning is utilized.

Fig. 12.13 shows a couple of challenging examples, jointly with their scribbles.

In the time since the publication of these works, progress has been made in the development of 3D re-colorization algorithms, primarily focusing on applications in interior design for scenes and furniture, as well as CAD models. It is anticipated that generative models, previously employed in the aforementioned domains [16, 56], will be expanded to tackle the complex archaeological domain. Additionally, an alternative approach involves considering the problem as interactive segmentation and employing advanced segmentation algorithms tailored to our specific domain.

Limitations. The algorithms necessitate a reasonably dense input mesh. If the input mesh exhibits high irregularity or sparsity, or if the vertices of two valleys are one vertex apart, remeshing may be necessary. Additionally, in the presence of patterns, our assumption is that instances of the pattern are distinguished by a background area that separates them.

12.5 Ancient document enhancement

Binarization is the process of converting a multi-tone image into a binary image. When the image is a document, binarization classifies every pixel as text (foreground) or non-text (background). Binarization facilitates other document processing tasks, such as layout analysis and character recognition. Historical document images might be more difficult to binarize than modern scanned documents, due to their degraded state, as well as to the digitization process. There exist several excellent reviews of the problem and the methods that attempt to address it [18, 33, 36, 37, 46, 53]. These are categorized into the following types [53]:

- *Thresholding*. Setting a threshold that distinguishes between foreground and background, based on either local or global statistics (or both).
- *Edge detection*. These approaches perform edge detection in an attempt to find the boundary between foreground and background.
- *Image transforms*. These approaches extract local features based on image transforms such as Gabor filter banks or Fourier Transform.
- *Mixture of Gaussians*. This is a parametric probability distribution whose function is a weighted sum of Gaussian distributions, which can be fit to any set of unlabeled data points (features). Then, each data point is assigned to its most probable cluster, each of which is either a foreground or a background class.
- *Conditional Random Fields*. A CRF is associated with an energy (cost) function that scores the binarization. Thus, it may incorporate spatial dependencies among pixels into the binarization process. Usually, the energy function is hand-crafted and does not learn parameters.
- *Machine Learning*. Like in many other fields, recently machine learning methods, especially deep learning models, have become the state of the art in historical document binarization. The goal of these networks is to directly classify pixels.

Rather than proposing a new algorithm, we introduce a novel general approach [22], which improved the results of any state-of-the-art algorithm, by providing it with new information (input). Interestingly, this information is based on a solution to a seemingly unrelated problem of visibility detection in \mathbb{R}^3 . The key idea is that if we choose a suitable representation of pixels as points in \mathbb{R}^3 and then detect the subset of the points that are visible from a viewpoint (and/or occluding it), hidden information about the image is revealed. This information is used to create a new image, which when given as input to an existing algorithm, will improve its own results, while leaving the algorithm's general structure as usual.

Visibility has been defined in various dimensions and for different types of objects, however our focus is on a specific variant of it: Given a point set in \mathbb{R}^3 and a viewpoint, the goal is to determine the subset of points visible from the viewpoint. Since points cannot occlude each other (unless they accidentally fall on the same ray from the viewpoint), the more precise problem definition is to determine the subset of points that would be visible, if the surface from which the points were sampled, was known.

Why does visibility information assist us? The representation we propose will assign text pixels, which are dark on white background, to points in deep valleys. Background, which is constant and of high intensity, will turn to elevated plateaus. Stains on text, which have no distinct intensity characteristics, will be represented as structure-less, resembling "volcanic" terrains. Stains on background, which normally cause darker local regions, will outcome in craters or shallow valleys upon the elevated plateaus. It turns out that visibility detection operators manage to distinguish between deep valleys and "volcanic" terrains (text) to elevated plateaus, craters and shallow valleys (background).

Our approach proceeds as follows: Given an image of a document (Fig. 12.14(a)), which might be stained, shadowed or otherwise distorted, our goal is to produce another image (Fig. 12.14(d)), which better distinguishes the foreground from the

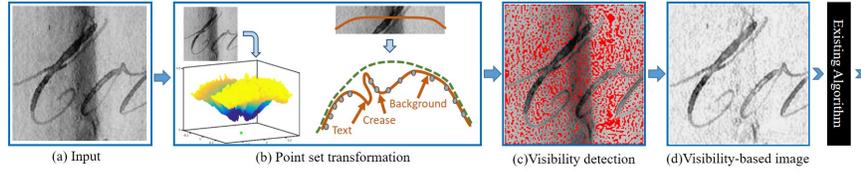


Fig. 12.14 Algorithm outline: The intensity image (a) is transformed to a 3D point set on a “globe”, where the distance of a point from the globe’s center (in green) is determined according to the intensity (b). Visibility detection operators detect which pixels have been transformed to visible/occluding points from/to the viewpoint (the globe’s center). Visible/occluding pixels are marked red in (c). Notice that the letters hardly have any red points, whereas the background and the stain have many. This implies a strong visibility–foreground correlation. Using the information of (c), a visibility-based image is created (d). It maintains the text of (a), while making stains disappear. This image is used to alter existing algorithms.

background. This image will benefit any algorithm that processes the document. Our approach is encapsulated into four stages:

Stage 1: Transform each pixel to a 3D point. Intuitively, one could imagine the image as if it was pasted onto a portion of the world globe, such that (1) valleys and craters on the surface are created from pixels of low intensity relatively to their vicinity; (2) ridges and mountains are created from high intensity pixels relatively to their vicinity; (3) plateaus are created from areas of pixels with low varying intensity.

To realize it, let p_i be a pixel within the image at location (x_i, y_i) and intensity level $I_i \in [0, 1]$. The transformation first spreads the image onto a portion of the unit sphere’s surface and then brings the pixels closer or further away according to their intensity, as demonstrated in Fig. 12.14(b). This figure shows both the full point cloud representing the whole image, as well as the transformation of a single (orange) curve through the image.

In spreading, p_i is first translated to a scaled coordinate system relatively to the image coordinate system, having its origin located in the center of the image s.t.:

$$x'_i = \frac{x_i - \frac{width}{2}}{M}, \quad y'_i = \frac{y_i - \frac{height}{2}}{M}, \quad (12.4)$$

where M is the maximum image dimension. This guarantees that $(x'_i, y'_i) \in [−0.5, 0.5] \times [−0.5, 0.5]$.

To set the height of a point on the globe, an initial height is defined for all pixels and p_i is then represented in polar coordinates (r_i, θ_i, ϕ_i) . Then, each pixel is transformed to the unit sphere at $(1, \theta_i, \phi_i)$. Finally, r_i is set relatively to p_i ’s intensity using a linear transformation to I_i s.t.:

$$r_i = a_1 \cdot I_i + a_0. \quad (12.5)$$

Stage 2: Detect the visible/occluding points from (/ to) the center of the globe. As illustrated in Fig. 12.14(c), red (occluding) dots exist on the background, including the stain, but not on the foreground letters. This is the key to the success of our

approach. This stage is performed by applying the visibility detection algorithm described in [20, 21].

Stage 3: Create a visibility-based image by using the information of Stage 2 above, as shown in Fig. 12.14(d). In particular, a smoothed image is created using only the intensity values of the pixels that generated the occluding points. For the rest of the pixels, we apply the *natural-neighbor interpolation* [47], using only the occluding pixels. Hence, their values are approximated by their neighboring occluding (background) pixels. Next, the smoothed image is subtracted pixel-wise from the input image and subsequently normalized to the range of $[0, 1]$, resulting in a visibility-based image. Since the value of a background pixel in the original image and in the smoothed image are equal, its value in the visibility-based image will be 1, as desired. Even if it is not detected occluding, the fact that it resides on a plateau or a ridge, means that its vicinity contains points that originated from pixels with similar intensity values. Consequently, the pixel will end up with a value approaching 1. On the other hand, if the pixel is a text pixel, then its 3D point will end up within a valley. This point, as well as points around it, are unlikely to be marked occluding. Thus, its interpolated value in the smoothed image will be much greater than its true intensity. Upon creation of the visibility-based image, its value will be close to 0.

Stage 4: Apply any existing binarization algorithm, using the visibility-based image rather than the input image.

The main dataset used in the literature for this application comes from the DIBCO series of competitions, e.g., [12, 40, 41, 42]. Every year, since 2009 (except 2015), a new dataset of 10–20 images of representative degradations, are released. The data and the evaluation software can be downloaded. Fig. 12.15 shows a couple of examples out of [40], where the results of the original algorithms are improved. Note how the vertical crease (top) is detected as background and how the noisy region on the right-hand side of the bottom image is better treated.

Limitation. The algorithm might produce suboptimal outcomes in cases where the fonts are small, and there is minimal color variance.

12.6 Conclusion

In this chapter we reviewed our work on cultural heritage restoration problems. We covered very diverse types of artifacts, from papyrus fragments, to ancient documents, archaeological drawings, and Hellenistic and Roman statues. These represent artifacts, which are 2D by nature and 3D artifacts. The algorithms presented in this chapter are very diverse by nature, showing how challenging are the problems in the field of computational archaeology.

Future work may benefit from the integration of image processing techniques with deep learning models. Furthermore, as is the case with all applications of archaeology, large datasets are a necessity, in particular when deep learning models are used. This is a major challenge since large datasets are very hard to come by in the field of archaeology and therefore the big challenge will be to develop deep

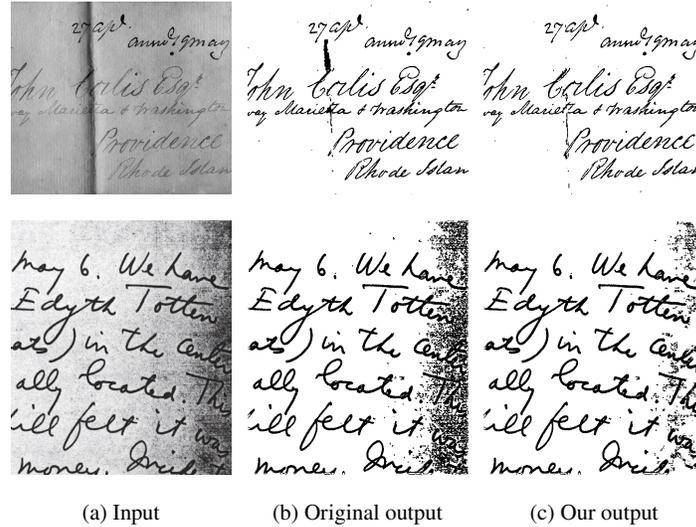


Fig. 12.15 Binarization results: (a): Images out of [40]. An algorithm's output in (b) is improved by using our approach in (c). Top: [17] Alg. 3 is improved from 92.06% to 93.61%. Our algorithm manages to detect the vertical crease as background. Bottom: [51] is improved from 76.18% to 90.18%. Our algorithm enables better treatment of the noisy region in the right side of the image. Scores are in PF-measure.

learning algorithms, which can work on relatively small datasets. Moreover, another challenge is to get the data labeled. This is also difficult since it usually can be done only by experts. Thus, exploiting unlabeled examples with a small set of labeled examples is also of great importance.

As mentioned earlier, there are numerous unresolved challenges in the field of restoration. One such intriguing aspect that was discussed in this chapter pertains to the reassembly of 2D objects. However, an equally fascinating area of study involves the reassembly of 3D objects. This encompasses a broad range of objects, including general 3D shapes[38, 58] as well as specific categories like surfaces of revolution [15] (e.g., pots, vases, etc.). While there has been considerable research dedicated to this problem, it is important to note that it is still an ongoing endeavor and not yet considered solved.

Additionally, we explored research related to document enhancement. This could be enhanced using deep learning techniques[50, 31]. Such algorithms can serve as components of a comprehensive end-to-end pipeline aimed at extracting the text content from documents. Deep learning techniques, specifically OCR (Optical Character Recognition) methods, in conjunction with large language models, can be leveraged to accomplish this task. These models have the capability to not only recover the text from documents but also correct errors that arise during the process.

References

1. Abitbol, R., Shimshoni, I., Ben-Dov, J.: Machine learning based assembly of fragments of ancient papyrus. *Journal on Computing and Cultural Heritage (JOCCH)* **14**(3), 1–21 (2021)
2. Anwar, H., Anwar, S., Zambanini, S., Porikli, F.: Deep ancient roman republican coin classification via feature fusion and attention. *Pattern Recognition* **114**, 107871 (2021)
3. Barucci, A., Cucci, C., Franci, M., Loschiavo, M., Argenti, F.: A deep learning approach to ancient egyptian hieroglyphs classification. *Ieee Access* **9**, 123438–123447 (2021)
4. Bonn-Muller, E.: Carved in living color. *Archaeology* **61**(1), 14–15 (2008)
5. Brusić, Z.: Hellenistic and roman relief pottery in liburnia: North-east adriatic, croatia. *BAR International series* 817 (1999)
6. Canul-Ku, M., Hasimoto-Beltran, R., Jiménez-Badillo, D., Ruiz-Correa, S., Román-Rangel, E.: Classification of 3d archaeological objects using multi-view curvature structure signatures. *IEEE Access* **7**, 3298–3313 (2018)
7. Chetouani, A., Treuillet, S., Exbrayat, M., Jesset, S.: Classification of engraved pottery sherds mixing deep-learning features by compact bilinear pooling. *Patt. Recog. Let.* **131**, 1–7 (2020)
8. Derech, N., Tal, A., Shimshoni, I.: Placement visualization (2019). URL <https://www.youtube.com/watch?v=2-pjbiJ1GQ&feature=youtu.be>
9. Derech, N., Tal, A., Shimshoni, I.: Solving archaeological puzzles. *Pattern Recognition* **119**, 108065 (2021)
10. Dimov, D.T.: Rotation-invariant ncc for 2d color matching of arbitrary shaped fragments of a fresco. *Pattern Recognition Letters* **138**, 431–438 (2020)
11. Fitch, C.R., Goldman, N.W., Goldman, N.: *Cosa, the lamps*, vol. 39. University of Michigan Press (1994)
12. Gatos, B., Ntirogiannis, K., Pratikakis, I.: Icdar 2009 document image binarization contest (dibco 2009). In: 2009 10th International conference on document analysis and recognition, pp. 1375–1382. *IEEE* (2009)
13. Gilboa, A., Tal, A., Shimshoni, I., Kolomenkin, M.: Computer-based, automatic recording and illustration of complex archaeological artifacts. *Journal of Archaeological Science* **40**(2), 1329–1339 (2013)
14. Hayon, O., Münger, S., Shimshoni, I., Tal, A.: *Arcaid: Analysis of archaeological artifacts using drawings* (2022)
15. Hong, J.H., Yoo, S.J., Zeeshan, M.A., Kim, Y.M., Kim, J.: Structure-from-sherds: Incremental 3d reassembly of axially symmetric pots from unordered and mixed fragment collections. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5443–5451 (2021)
16. Hou, J.U., Zhao, B., Ansari, N., Lin, W.: Range image based point cloud colorization using conditional generative model. In: *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 524–528. *IEEE* (2019)
17. Howe, N.R.: Document binarization with automatic parameter tuning. *International Journal on Document Analysis and Recognition* **16**(3), 247–258 (2013)
18. Ismail, S.M., Abdullah, S.N.H.S., Fauzi, F.: Statistical binarization techniques for document image analysis. *J. Comput. Sci.* **14**(1), 23–36 (2018)
19. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence* **21**(5), 433–449 (1999)
20. Katz, S., Tal, A.: On the visibility of point clouds. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1350–1358 (2015)
21. Katz, S., Tal, A., Basri, R.: Direct visibility of point sets. In: *ACM SIGGRAPH 2007 papers*, pp. 24–es. *ACM* (2007)
22. Kligler, N., Katz, S., Tal, A.: Document enhancement using visibility detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2374–2382 (2018)
23. Kolomenkin, M., Leifman, G., Shimshoni, I., Tal, A.: Reconstruction of relief objects from line drawings. In: *CVPR 2011*, pp. 993–1000. *IEEE* (2011)
24. Kolomenkin, M., Leifman, G., Shimshoni, I., Tal, A.: Reconstruction of relief objects from archeological line drawings. *Journal on Computing and Cultural Heritage (JOCCH)* **6**(1), 1–19 (2013)

25. Kolomenkin, M., Shimshoni, I., Tal, A.: On edge detection on surfaces. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2767–2774. Ieee (2009)
26. Lawonn, K., Trostmann, E., Preim, B., Hildebrandt, K.: Visualization and extraction of carvings for heritage conservation. *IEEE transactions on visualization and computer graphics* **23**(1), 801–810 (2016)
27. Leifman, G., Tal, A.: Mesh colorization. *Computer Graphics Forum* **31**(2), 421–430 (2012)
28. Leifman, G., Tal, A.: Pattern-driven colorization of 3d surfaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 241–248 (2013)
29. Lerner, J., Shimshoni, I.: Thin section analysis for ceramic petrography using motion analysis and segmentation techniques. *Machine Vision and Applications* **33**(5), 70 (2022)
30. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. In: ACM SIGGRAPH 2004 Papers, pp. 689–694. ACM (2004)
31. Lin, Y.H., Chen, W.C., Chuang, Y.Y.: Bedsr-net: A deep shadow removal network from a single document image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12905–12914 (2020)
32. Ling, H., Okada, K.: Diffusion distance for histogram comparison. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 1, pp. 246–253. IEEE (2006)
33. Lins, R.D., Bernardino, R., Jesus, D.M.: A quality and time assessment of binarization algorithms. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1444–1450. IEEE (2019)
34. Luo, T., Li, R., Zha, H.: 3d line drawing for archaeological illustration. *International Journal of Computer Vision* **94**, 23–35 (2011)
35. Ma, Y., Arandjelović, O.: Classification of ancient roman coins by denomination using colour, a forgotten feature in automatic ancient coin analysis. *Sci* **2**(2), 37 (2020)
36. More, P.K., Dighe, D.: A review on document image binarization technique for degraded document images. *Int. Res. J. Eng. Technol* **3**(3), 1132–1138 (2016)
37. Mustafa, W.A., Aziz, H., Khairunizam, W., Ibrahim, Z., Shahrman, A., Razlan, Z.M.: Review of different binarization approaches on degraded document images. In: 2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA), pp. 1–8. IEEE (2018)
38. Niu, X., Wang, Q., Liu, B., Zhang, J.: An automatic chinaware fragments reassembly method framework based on linear feature of fracture surface contour. *ACM Journal on Computing and Cultural Heritage* **16**(1), 1–22 (2022)
39. Phillips, S.C., Walland, P.W., Modafferi, S., Dorst, L., Spagnuolo, M., Catalano, C.E., Oldman, D., Tal, A., Shimshoni, I., Hermon, S.: GRAVITATE: Geometric and Semantic Matching for Cultural Heritage Artefacts. In: C.E. Catalano, L.D. Luca (eds.) Eurographics Workshop on Graphics and Cultural Heritage. The Eurographics Association (2016)
40. Pratikakis, I., Gatos, B., Ntirogiannis, K.: DIBCO 2011: Document Image Binarization Contest. In: International Conference on Document Analysis and Recognition (ICDAR) (2011)
41. Pratikakis, I., Gatos, B., Ntirogiannis, K.: DIBCO 2013: Document Image Binarization Contest. In: International Conference on Document Analysis and Recognition (ICDAR) (2013)
42. Pratikakis, I., Zagoris, K., Karagiannis, X., Tsochatzidis, L., Mondal, T., Marthot-Santaniello, I.: Competition on document image binarization (dibco 2019). In: International conference on document analysis and recognition (ICDAR), p. 1547–56 (2019)
43. Qu, Y., Wong, T.T., Heng, P.A.: Manga colorization. *ACM Transactions on Graphics (ToG)* **25**(3), 1214–1220 (2006)
44. Romanengo, C., Biasotti, S., Falcidieno, B.: Recognising decorations in archaeological finds through the analysis of characteristic curves on 3d models. *Pattern Recognition Letters* **131**, 405–412 (2020)
45. Saalfeld, P., Böttcher, C., Klink, F., Preim, B.: Vr system for the restoration of broken cultural artifacts on the example of a funerary monument. In: 2021 IEEE Virtual Reality and 3D User Interfaces (VR), pp. 739–748. IEEE (2021)
46. Sezgin, M., Sankur, B.I.: Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic imaging* **13**(1), 146–168 (2004)

47. Sibson, R., et al.: A brief description of natural neighbour interpolation. *Interpreting multivariate data* **21**, 21–36 (1981)
48. da Silva Teixeira, T., de Andrade, M.L.S.C., Da Luz, M.R.: Reconstruction of frescoes by sequential layers of feature extraction. *Pattern Recognition Letters* **147**, 172–178 (2021)
49. Sipiran, I., Lazo, P., Lopez, C., Jimenez, M., Bagewadi, N., Bustos, B., Dao, H., Gangisetty, S., Hanik, M., Ho-Thi, N.P., et al.: Shrec 2021: Retrieval of cultural heritage objects. *Computers & Graphics* **100**, 1–20 (2021)
50. Souibgui, M.A., Kessentini, Y.: DE-GAN: A conditional generative adversarial network for document enhancement. *IEEE Trans. on Patt. Ana. and Mach. Intel.* **44**(3), 1180–1191 (2020)
51. Su, B., Lu, S., Tan, C.L.: Robust document image binarization technique for degraded document images. *IEEE Transactions on Image Processing* **22**(4), 1408–1417 (2013)
52. Tal, A.: 3d shape analysis for archaeology. *3D Research Challenges in Cultural Heritage: A Roadmap in Digital Heritage Preservation* pp. 50–63 (2014)
53. Tensmeyer, C., Martinez, T.: Historical document image binarization: A review. *SN Computer Science* **1**(3), 173 (2020)
54. Wilczek, J., Monna, F., Jébrane, A., Chazal, C.L., Navarro, N., Couette, S., Smith, C.C.: Computer-assisted orientation and drawing of archaeological pottery. *Journal on Computing and Cultural Heritage (JOCCH)* **11**(4), 1–17 (2018)
55. Xiong, W., Zhou, L., Yue, L., Li, L., Wang, S.: An enhanced binarization framework for degraded historical document images. *EURASIP Journal on Image and Video Processing* **2021**(1), 13 (2021)
56. Yang, Z., Liu, L., Huang, Q.: Learning generative neural networks for 3d colorization. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32, pp. 2580–2587 (2018)
57. Zeng, Q., Martin, R.R., Wang, L., Quinn, J.A., Sun, Y., Tu, C.: Region-based bas-relief generation from a single image. *Graphical models* **76**(3) (2014)
58. Zhang, Y., Li, K., Chen, X., Zhang, S., Geng, G.: A multi feature fusion method for reassembly of 3d cultural heritage artifacts. *Journal of Cultural Heritage* **33**, 191–200 (2018)
59. Zhang, Y.W., Wang, J., Wang, W., Chen, Y., Liu, H., Ji, Z., Zhang, C.: Neural modelling of flower bas-relief from 2d line drawing. *Computer Graphics Forum* **40**(6), 288–303 (2021)