

VASESKETCH: Automatic 3D Representation of Pottery from Paper Catalog Drawings

Francesco Banterle, Matteo Dellepiane,
Marco Callieri, Roberto Scopigno
Visual Computing Laboratory
ISTI-CNR, Pisa, Italy
{banterle,dellepiane,callieri,scopigno}@isti.cnr.it

Barak Itkin, Lior Wolf, Nachum Dershowitz
Blavatnik School of Computer Science
Tel Aviv University
Tel Aviv, Israel
{barakitk,wolf,nachum}@cs.tau.ac.il

Abstract—We describe an automated pipeline for digitization of catalog drawings of pottery types. This work is aimed at extracting a structured description of the main geometric features and a 3D representation of each class. The pipeline includes methods for understanding a 2D drawing and using it for constructing a 3D model of the pottery. These will be used to populate a reference database for classification of potsherds.

Furthermore, we extend the pipeline with methods for breaking the 3D model to obtain synthetic sherds and methods for capturing images of these sherds in a way that matches the imaging methodology of archaeologists. These will serve to build a massive set of synthetic sherd images that will help train and test future automated classification systems.

I. INTRODUCTION

One of the most active research areas in AI is Sim2Real, where classifiers and agents are trained on simulated data and then deployed in the real world. Examples include automatic driving based on hundreds of hours of captured driving-simulators' data and visual recognition of objects based on their CAD models. We study the problem of Doc2Sim, where simulated data are obtained from document images. This task is a major stepping stone toward our end goal of Doc2Sim2Real for the classification of archaeological pottery sherds.

Pottery classification in the context of excavation sites is a crucial operation, because pottery sherds are the “carbon dating” of archeology in the absence of organic material. The classification of a sherd provides valuable information about the historical period, commercial routes, eating habits, industrial production, etc.

To obtain a coherent description, the different pottery typologies are subdivided into classes and subclasses, which are described in a reasonably well-structured way.

The main references for pottery specialists are published catalogs that contain the collection of the classes of each typology. Each class has both a textual and a pictorial description, which represents the profile of the main elements (viz. the body and the handle) of a class. Additional

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 693548 (project ArchAIDE).

information (e.g. the decoration style) may be present. Fig. 1 shows two examples of drawings, taken from two different catalogs. Typically, the classification of a sherd is obtained by visually comparing it with the drawings in the relevant catalog, possibly taking into account the most descriptive features such as mouth shape, base, and handles.

One of the goals of the ArchAIDE project¹ is to develop an automatic classification system that supports the work of archaeologists on-site by extracting geometric features from a single image of the sherd taken on-site (or later), such as the one in Fig. 2. This geometric information will then be used by a machine-learning classifier trained on all the classes in a reference database, in order to select a set of candidates for classification.

Starting from the idea of replicating the process used by archaeologists on the field who compare the sherds with the catalog drawings, this paper presents an automatic pipeline for the digitization of pottery profile drawings. The aim is to extract a structured description of the main geometric features and a 3D representation of each class. These data will be used to populate the reference database for classification and to build a massive set of *synthetic sherds*. The synthetic sherds will then be employed to train the classification system.

¹<http://www.archaide.eu>

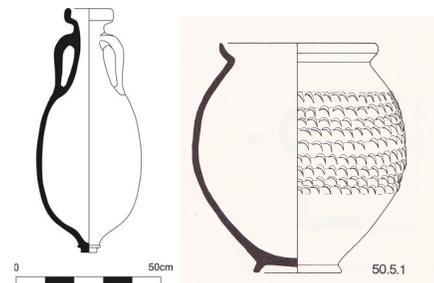


Figure 1: Two drawings of pottery types taken from catalogs.

II. RELATED WORK

A. Pottery Classification and Computer Science

The study of pottery has been a topic of interest in computer science for several years. This may be due to the fact that, historically, several typologies of pottery were manufactured on an industrial scale, so that one can exploit standardized shapes in their analysis.

The advent of 3D acquisition technologies led to several projects in which a 3D representation of a sherd was obtained via 3D scanning: the 3D model was used for analysis [1] and reconstruction of the full vessel [2], [3]. Unfortunately, the 3D acquisition of sherds is not an easy task, especially outside of the laboratory setting (the same holds for recent efforts in using multi-view stereo matching approaches [4]). Hence, the proposed automated systems [5], [6] have never enjoyed widespread use.

Moreover, any classification or reconstruction operation was essentially based on the extraction of the descriptive profile from the 3D model. This operation can be automated only in specific cases: it usually requires intervention by an experienced user to position the sherd before profile extraction.

Profiles are the basic feature that is used for classification and analysis. Once extracted from 3D data or drawings, they were analyzed and compared using several measures, including Hough transform [7], morphological measures [8], [9], and curvature functions [10].

Other works aim at extracting the features described by archaeologists in paper catalogs [11] and use them directly for partial matching and classification [12]. However, in those works the extracted profiles are matched without taking the different components of the profiles into account, whereas our solution uses the component information.

Finally, some work has been proposed regarding the use of appearance [13] or the combination of appearance and shape [14].



Figure 2: An example of an amphora sherd.

B. 3D Reconstruction from Drawings

The creation of 3D shapes from 2D lines is an important issue in computer graphics, since most of the interaction for modeling has to pass through two-dimensional input metaphors. Sketching interfaces have been studied in depth; we refer the reader to a recent survey [15] for an overview.

However, the aim of this paper is to extract information from a more *structured* type of information, where a set of strict rules guides the creation of the drawing. The interpretation of line drawing has been studied right from the beginning of computer graphics [16]. Even recently, in the general case, the interpretation of simple sketches [17] or only the extraction of curved lines [18] still pose severe challenges.

More interesting results can be obtained when the drawing style follow pre-defined rules: orthographic views [19] or drawing scaffolds [20] can guide the reconstruction of complex objects.

Drawings coming from certain communities (e.g. character design [21] or architecture [22]) follow structured rules that can help the extraction of geometric features for reconstruction. Our system works in a similar context, i.e. the guiding features for tracing a drawing can be used for 3D reconstruction. Furthermore, our aim is also to extract a well structured set of features that could be used in different contexts such as classification and comparison.

III. THE AUTOMATIC PIPELINE

Our proposed system comprises three main components, which are described in this section. The first component extracts a set of pre-defined geometric features from the initial drawing: the features will be part of the reference dataset for the classification system. The second one uses the geometric features to generate a 3D representation of the drawing and to “break” the 3D model into a set of sherds. Finally, the third component extracts the training set of features for the classification system from the synthetic sherds.

A. Geometric Features and their Extraction

Pottery drawings in the catalogs are already a *technical* representation of a geometry, but all the semantic information is flattened in a single *raster* layer, encoded following specific representation rules (e.g. line thickness/filling/dashing, and axis orientation). This is due to the nature of the media used and to when this standard was defined. While perfectly fine for human interpretation, this representation is limited. The use of a digital *vectorial* representation allows us to separate the different semantic elements and to add location-based metadata. This new representation is more suitable for the use on digital media, both from human and automatic algorithms. Therefore, our first step is to create such representation. The set of geometric features to be extracted from drawings should

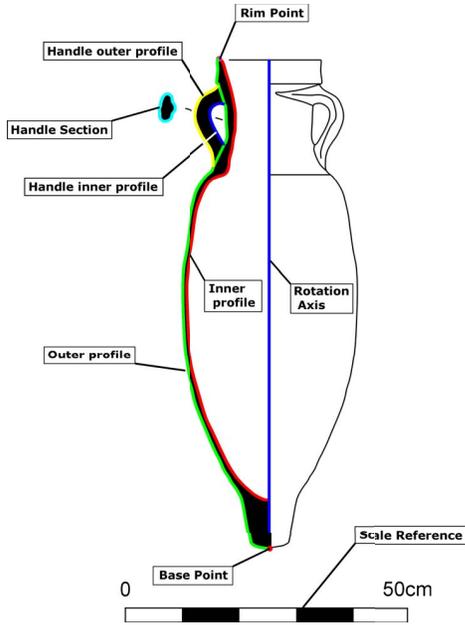


Figure 3: The selected set of geometric features that defines a class.

fulfill three main requisites: they have to (i) describe the features that archaeologists use when analyzing a sherd; (ii) contain features that could be extracted in a semi-automatic way from an image of the sherd taken on-site (and from the synthetic sherds generated for training); and (iii) be complete enough to allow an automatic 3D reconstruction of its class.

Based on these assumptions, our discussion with pottery experts led to the definition of the following features, as shown in Fig. 3:

- *Outer profile* – green outline in figure.
- *Inner profile* – red outline in figure.
- *Handle outer profile* (if present) – yellow outline in figure.
- *Handle inner profile* (if present) – blue outline in figure.
- *Handle section* (if present) – cyan outline in figure.
- *Rim point*: the top point in the profile.
- *Base point*: the bottom point in the profile.
- *Scale factor*: the scaling value to bring all features to real scale.

In addition to the above features, the *rotation axis* has to be extracted because it will be used for the automatic 3D reconstruction. Given the above set of features, the extraction procedure is implemented using the following steps:

1) *Image De-noising and Binarization*: To simplify tracing of profiles, first we have to create a binary version of input image I . To achieve this, we scale the pixel values of I to the range $[0..1]$, and we threshold it with a 0.5 threshold. Then, we apply dilate and erode operators (5×5 for 5 times) to remove possible outliers coming from noise. The result

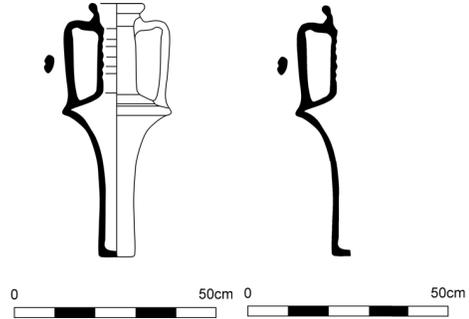


Figure 4: Removal of surface sketches.

of these operations is a de-noised binary image B .

2) *Finding the Rotation Axis*: Once B is computed, we need to find the *rotation axis* of the vessel's drawing, which is usually present. Hough transform [23] is employed to detect the longest vertical line in a drawing, which is assumed to be its rotation axis. We now need to detect all features from the section of the drawing. To avoid outliers, the part of the drawing illustrating the surface of the vessel (i.e. the part on the right) is removed, see Fig. 4.

3) *Extracting the Inner Profile*: To compute the *inner profile*, first we compute all edges in B , and store them in a list of piece-wise linear curves, E . Then, we find the closest point \mathbf{p} to the *rotation axis* from a curve $e_i \in E$. From \mathbf{p} , we march to the top point of e_i , i.e. the *rim point*, and to the bottom point of e_i , i.e. the *base point*. The sequence of points of e_i between these two points and containing \mathbf{p} , belongs to the *inner profile*. We define this curve as e_i^* .

4) *Extracting the Outer Profile and Handles*: All remaining points of e_i , viz. e_i' , contain both the *outer profile* and the *handle outer profile*. Furthermore, the *handle inner profile* and a part of the *outer profile* belong to a curve e_j (see Fig. 5). To properly cut both e_i' and e_j to obtain the *outer profile*, we define an energy function that ensures C_0 and G_0 smoothness when trimming e_i' and e_j and joining the parts belonging to the *outer profile*. Note that all remaining points in e_i' define the *handle outer profile*, and all remaining points in e_j define the *handle inner profile*. The process is shown in Fig. 5. The *handle outer profile*, if it exists, can be detected in a straightforward way by looking at a curve $e_k \in E$ that is smaller than e_j .

5) *Computing the Scale Factor*: The *scale factor*, s_f , is computed by processing B using OCR software. In our case, we used the command-line version of the Tesseract OCR engine.² To improve Tesseract OCR accuracy, we detect, where the scale is, by computing the longest horizontal line using again the Hough transform. At this point, we crop a 50% larger area of B around the scale to include scale values. The cropped area is processed by the OCR engine, which outputs two values: s_{\min} the minimum scale value,

²<https://github.com/tesseract-ocr>

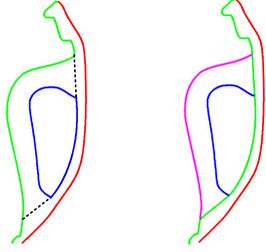


Figure 5: The process of cutting different curves around the handle of the vessel: On the left are the extracted curves before cutting and joining. The e_i' curve is green; e_i^* is red; and e_j is blue. On the right is the result after cutting.

and s_{\max} the maximum scale value. Finally, the scale factor is computed as

$$s_f = \frac{s_{\max} - s_{\min}}{sp},$$

where sp is the length of the scale, in pixels, by using the result of the Hough transform.

B. Generating the 3D Model and Sherds

In order to train a machine-based classifier to identify potsherds, the first step is to actually obtain data. These data are needed both for the training and the testing processes. However, while we have many paper catalogs describing the various types of pottery, very few of them include information on specific sherds. While it is possible to obtain some data by taking photos of sherds that were already classified and stored, this process is tedious and might not be enough, since training machine-learning classifiers, especially deep neural networks, require large amounts of samples.

Therefore, we seek to employ synthetic training data obtained through a specialized 3D algorithm to virtually “break” the 3D models produced in the previous step into many small sherds. When generating data this way, we can obtain a large quantity of samples, with class labels for each sherd, in a way that does not require a large amount of manual work. In addition, we can obtain the same amount of data even for classes for which only few samples were found in the field.

1) *Simplifying the Profile*: The first stage is to simplify the model. The outlines that were obtained previously come from identifying connected components in binary images. This may lead to aliasing artifacts on diagonal lines and curves. By applying line simplification algorithms, we aim to reduce the number of points (to make further computations faster) and eliminate those small aliasing features.

To simplify the outline, we use a modified version of Visvalingam’s algorithm [24]. The original algorithm simplifies an outline by maintaining the list of triangles formed on the line by every three consecutive points. The “importance” of a point is then measured by the area of the triangle of

which it’s the middle point – because removing this point will flatten the triangle to a line, thus making a change of this size. Intuitively, we should always strive to make the smallest changes when removing a point, thus at each step we remove the point that would eliminate the triangle with the smallest area.

The problem with using many out-of-the-box line simplification algorithms, including [24], is that they do not protect against the arise of self-intersections in the outline during the simplification process. To solve this, we use the newer approach suggested in [25], which modifies Visvalingam’s algorithm to avoid self intersections. Our experiments showed that, using [25], most outlines can be reduced from a few thousand points, to typically fewer than 200, with no visible differences. Furthermore, by capping the area of the removed triangles, the level of simplification can be fine-tuned as desired.

2) *Generating the 3D Model*: Many types of pottery are crafted on a potter’s wheel; the body is produced by rotating the clay around a central axis while deforming the profile to form the base of the vessel. The handles (and sometimes other external elements) are created with a manual process. Nevertheless the handle section and attachment to the body are usually common among the exemplars of the same class. In this paper, we suggest mimicking the same process to reconstruct the 3D model out of the profile, treating the body and the handles in different ways.

Generating the body: First, we extract the profile of the body (made by joining the inner and outer profiles) and the rotation axis. Then, we simplify the profile and scale it to the real measures using the scale factor. Afterwards, we align the profile on the xz plane in 3D, so that the rotation axis is located on the z -axis. Finally, we generate the 3D body by rotating the result around the z -axis.

Generating the handles (if present): First, we extract the handle profiles and section. Then, we create the handle model by extruding the handle sections following the handle profiles, while scaling the section according to the distance between the profiles. Afterwards, we align the handles to the xz plane. Finally, the handles and the body are connected by finding the intersection between the models and creating a 2-manifold surface.

Fine-tuning rotational resolution: When generating the body by rotating the simplified profile, one parameter we still have to determine is the number of “vertical rings”, i.e. how many times to duplicate the profile around the axis, to form the body. While it is possible to set this to a fixed number (200 seems to yield smooth results), we can also determine this dynamically using the same simplification logic from Visvalingam’s algorithm, to yield consistent level of detail. Let k be the number of vertical rings, and let r_{\max} be the maximal (horizontal) radius from the rotation axis. Then, the area of a triangle formed by the same point at radius r_{\max}

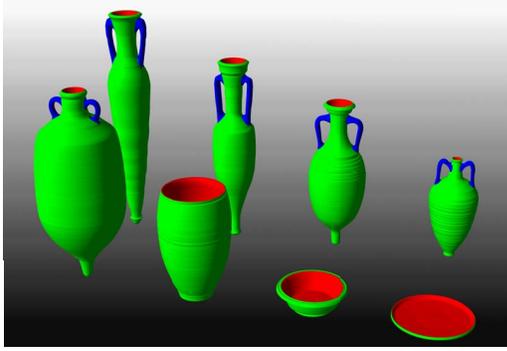


Figure 6: A snapshot of some reconstructed 3D models.

on 3 consecutive vertical rings, is given by the formula:

$$\Delta_{area} = r_{max}^2 \cdot \sin\left(\frac{2\pi}{k}\right) \cdot \left(1 - \cos\left(\frac{2\pi}{k}\right)\right)$$

As can be seen from the formula, the area increases as k decreases. Therefore, to achieve the same level of detail that was obtained in the simplification process, we want to find the minimal k so that the area is still less than the area of triangles we eliminated in the simplification process. In practice, using this formula allows to decrease the number of vertical rings by a factor of 3–5 times compared to the fixed value of $k = 200$.

Results: Fig. 6 shows 8 examples of reconstructed 3D models: the different parts of the object (external and internal body surface, handles) are represented with different colors, visualizing the different geometric features used in the generation process.

3) *Generating Synthetic Sherds:* To break the 3D model into sherds, we use the “Cell Fracture” plug-in of Blender³ open-source tool. The plug-in breaks apart 3D models by generating a 3D Voronoi diagram, and then computing the intersection of each 3D Voronoi cell with the original model. Therefore, the resulting fractures in the model correspond to the separating planes between the Voronoi cells.

Before breaking the model, we annotate faces as exterior/interior (as in Fig. 6). The fracturing process preserves these notations, while also adding a fracture notation for faces that represent fractures. This process can be repeated again to generate many sherds for each model. Furthermore, by controlling the number of Voronoi cells, we can influence the size of the sherds (fewer cells imply larger cells and larger sherds); thus we can obtain sherds of all sizes.

C. Capturing 2D Sherd Images

Since our goal is to generate sherd images that match those that are taken by field archaeologists, we discussed with the archaeologists involved in the ArchAIDE project the semi-canonical views that are often being used to capture

³<https://www.blender.org>



(a) A reconstructed amphora model without handles. This model has 61 vertical rings and 142 profile points. (b) Sherd with a vertical fracture. The inner and outer profiles are on the left and right of the fracture, resp. (c) Sherd without a vertical fracture. Its right side is aligned to outline the outer profile.

Figure 7: Amphora with two types of sherds.

a sherd. From their answers, it seems that the most indicative picture would be the one looking at a fracture face, trying to align the rotation axis to the vertical direction of the image. For such a fracture, we can extract the outlines of the inner and outer profiles with a semi-automatic method (see Fig. 3). Since it is not always possible to find a point of view where both the internal and external profiles are visible, the alternative is to find a point of view where only one profile (usually the external one) is visible, and use only this feature for classification.

1) *Righting Sherds:* For real potsherds, rotational symmetry can be identified in most cases. While some ambiguity may arise in areas that come from handles or other “extra” features, archaeologists are usually able to guess the axis of rotation, also by taking into account other details (such as the geometric features in the internal side of the body). For synthetic data, since the process of breaking the model into sherds did not move sherds from their place, we already have all sherds oriented upwards as desired.

2) *Finding Vertical Fractures and Aligning the Camera:* Our generated data has smooth fracture faces, as they are generated from the bounding planes between 3D convex cells. Therefore, faces that correspond to a vertical fracture can be characterized as faces that form a small angle with the rotation axis. When taking the picture of the sherd, we align the camera so that its “up” direction is the true up direction of the rotation axis, and so there’s only one degree of freedom on its angle – choosing the rotation angle around the rotation axis (the z -axis).

Our goal is to take the picture of the largest vertical fracture. This can be formalized by maximizing the area occupied by faces of vertical fractures in the 2D image. To solve this optimization problem, we approximate the solution by considering only orthographic projection (i.e. no perspective deformation based on distance from the camera) and no occlusion by other faces. We then can characterize



(a) Maximizing left vertical fracture area. (b) Maximizing overall fracture area. (c) Maximizing vertical right fracture area.

Figure 8: Handling multiple vertical fractures. As can be seen in (b), aligning multiple vertical fractures together results in an unintuitive pose. Archaeologists would typically align one fracture at a time.

the target function as

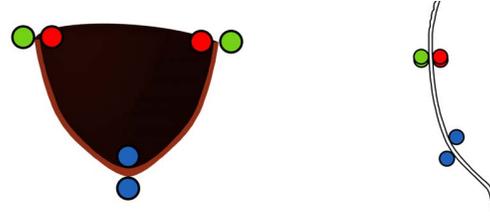
$$\arg \max_{\vec{d}} \sum_{f \in F_{\text{vert}}} \max \left(0, \text{normal}(f) \cdot \vec{d} \right) \cdot \text{area}(f) \quad (1)$$

Instead of solving this optimization problem directly (which is hard due to the non-linearity of max), we quantize the infinite solution space into a finite set of a few hundred possible direction vectors, and pick the best one (see Fig. 7b). As detailed below, this optimization will be solved once for every vertical fracture. Specifically, the non-trivial cases we need to handle are as follows:

No vertical fractures: As mentioned before, some sherds have no vertical fractures (or only have small ones), and for those we must align the side with the outer profile as shown in Fig. 7c. To do this, we find the vertical ring with the longest line along the outer profile, and align the normal of that plane to point to the camera.

Multiple vertical fractures: Each sherd can be the result of multiple fractures in varying directions, with multiple disconnected vertical fractures. In that case, the optimization of Eq. (1) may yield an angle observing two separate vertical fractures instead of being optimally aligned with one such (see Fig. 8b). To solve this, instead of globally maximizing vertical fracture areas in the 2D image, we apply the process separately for each vertical fracture. Let P be the 3D polyhedron representing the sherd, and let G_f be the face graph of P . The vertices of G_f correspond to faces of P , and there is an edge in G_f iff the corresponding faces in P share a common edge. Let G_{vert} be a subgraph of G containing only faces in P that correspond to vertical fractures. Then each connected component in G_{vert} corresponds to a distinct vertical fracture. For each such, we solve the above optimization problem and pick the fracture and angle yielding the highest global score computed over all fractures.

V-fractures: Another issue to consider is “V-fractures” – vertical fractures that form a “V” shape in the sherd, as we have in Fig. 9. These fractures can be separated into two smaller instances of vertical fractures. Let $P(t) : [0..1] \rightarrow \mathbb{R}^2$ be the function representing the 2D curve of the profile outline, progressing clockwise around the outline (the closed



(a) A sherd with annotated t values. (b) Corresponding t values on the profile.

Figure 9: Mapping points on a V-shaped cut to the t time on the profile. By identifying local maxima (the blue points) of t on the fracture, one can decide where the boundary between two separate vertical fractures is.

shape formed by the inner and outer profiles) as t increases. Since P is a one-to-one mapping, we can define the inverse function, P^{-1} from a 2D point on the outline, to its t value along the curve. The inverse mapping is extended to 3D resulting in P_3^{-1} , which maps a point on the 3D rotation of the profile, back to its t along the curve. We then break the V-fractures by finding the P_3^{-1} value of each point on the contour of the fracture faces (i.e. points shared with an inner/outer face), and splitting vertical fractures by finding local maxima/minima in the P_3^{-1} values.

3) *Capturing 2D Sherd Images:* Once we have the direction vector of the camera, we need to create the 2D image. While this can be done via traditional ray-tracing, it is both inefficient (running time depends on the image resolution) and inaccurate – as we’ll again have to trace an outline in a 2D image and simplify it to get rid of aliasing artifacts. Instead, we propose generating the image by using *2D envelopes* of the 3D sherd model. The *upper envelope* of a 3D model is a projection of its faces into 2D, so that when multiple faces share the same (x, y) coordinate, the visible face will be the face with highest z value.

By using the implementation of [26] in CGAL,⁴ we can efficiently obtain the 2D envelope of our model. This is essentially the same 2D projection we would have achieved using classic ray-tracing, but obtained with points and lines instead of discrete pixel values. By finding the correspondence between 2D faces in the planar map of the envelope and 3D faces of sherd, we can compute the 2D outline of the fracture area, thus completing our generation process.

IV. RESULTS AND DISCUSSION

The ArchAIDE project selected three typologies of pottery for the first testing of all its components: *Roman amphorae*, *terra sigillata* and *medieval pottery*. Especially for the first two, archaeologists rely heavily on shape for classification.

The total number of reference classes for the three typologies is on the order of a few hundred. In addition to

⁴<http://www.cgal.org>

previous figures, Fig. 10 shows the results of the proposed pipeline on four examples of the *Roman amphorae* and *Terra sigillata* typologies. The whole pipeline is completed within one minute of computation, varying on account of the complexity of the geometric features.

While the focus of this work is on the use of 3D models for training a classifier, both the vectorial representation of the profile and the 3D model are useful on their own. Following the process, the reference database now contains multiple representations of the shape, each suitable for different kinds of uses. The 3D models can be used directly for printing 3D replicas, CG images or animations, or in real-time immersive applications. The vectorial representation can be helpful for interactive visualization and annotation on digital media, preserving all its metadata and semantic information.

The proposed pipeline is completely automatic and relatively efficient. All the same, some limitations have to be taken into account.

- *Variability of drawings*: The style of drawings coming from different catalogs may vary a lot (i.e. black-filled vs. hatch-filled profile, incomplete object). As more and more catalogs are added, the process should be validated and modified as needed. Alternatively, a conversion step to a canonical drawing style can be employed.
- *Decoration and other features*: The drawings usually contain more information about the class, including for example, the decoration. Unfortunately, the depiction style found in different catalogs is too variable to allow for automatic extraction of this type of information.
- *Asymmetry*: Certain classes of pottery are not fully symmetric (e.g. beakers). This is taken into account in drawings, and some additional rules when analyzing them is needed.
- *Fragmentation of objects*: While the fracture plug-in provides “plausible” sherds, no physical rule is applied. Unfortunately, only high-level indications are available regarding the way a pottery object breaks. Among them: the thickness, the distance with respect to rim and base, and the presence of handles.

V. CONCLUSIONS AND FUTURE WORK

The process of generating virtual vessels from drawings is expected to play a major role in the automation of archaeologists’ work. We have proposed a complete pipeline: from drawing to the generation of virtual sherds. Our models, therefore, are “built to be broken” and simulate the ravages of time. We expect such data to become instrumental in training deep neural networks to matching recovered sherds and catalog drawings, since the collection of real-world datasets for this task is infeasible.

While most traditional work in document analysis results in either low-level (e.g. binarization) or high-level (e.g. a

reading of the text) information, our work is different. We consider documents that describe real objects (that might no longer exist in their complete form) and create, not only a 3D model, but also aligned virtual views of broken pieces of these objects, which might help classify future sherds that would be recovered in some future excavation. In this regard, our document analysis work starts from a description of the real world and completes a full circle to derived objects that might exist. Doing so is an elaborate process in which the information extracted from the document image undergoes a multitude of steps.

Regarding future directions of work, the first effort will be in the creation and testing of the classification system, using both the synthetic sherds, and a number of real images provided by the project consortium. Additionally, the pipeline will be made more robust by implementing tools to handle different types of drawings and asymmetric objects, and by testing a more physically-related fragmentation of 3D models, possibly using data extracted from drawings (e.g. thickness).

REFERENCES

- [1] A. Karasik and U. Smilansky, “3D scanning technology as a standard archaeological tool for pottery analysis: Practice and theory,” *Journal of Archaeological Science*, vol. 35, no. 5, pp. 1148–1168, 2008.
- [2] M. Kappel, R. Sablatnig, and H. Mara, “Robust 3D reconstruction of archaeological pottery based on concentric circular rills,” in *WIAMIS05*, N. Magnenat-Thalmann and J. H. Rindel, Eds., Montreux, Switzerland, 2005, pp. 14–20.
- [3] N. Calin, P. Sorin, P. Daniela, and M. Razvan, “Using reverse engineering in archaeology: Ceramic pottery reconstruction,” *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 6, no. 2, pp. 55–59, 2012.
- [4] J.-B. Barreau, T. Nicolas, G. Bruniaux, E. Petit, Q. Petit, R. Gagne, and V. Gouranton, “Ceramics fragments digitization by photogrammetry, reconstructions and applications,” in *International Conference on Cultural Heritage (EuroMed)*, vol. abs/1412.1330, Nov. 2014.
- [5] M. Kappel and R. Sablatnig, *3D Data Retrieval of Archaeological Pottery*. Berlin: Springer, 2006, pp. 387–395.
- [6] A. Karasik, “A complete, automatic procedure for pottery documentation and analysis,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops (San Francisco, CA)*, Jun. 2010, pp. 29–34.
- [7] P. Durham, P. H. Lewis, and S. J. Shennan, “Artefact matching and retrieval using the generalised Hough transform,” in *Proceedings of Computer Applications in Archaeology*, J. Wilcock and K. Lockyear, Eds., 1995, pp. 25–30.
- [8] A. Karasik and U. Smilansky, “Computerized morphological classification of ceramics,” *Journal of Archaeological Science*, vol. 38, no. 10, pp. 2644–2657, 2011.
- [9] M. Lucena, A. L. Martínez-Carrillo, J. M. Fuertes, F. Carrascosa, and A. Ruiz, *Applying Mathematical Morphology for the Classification of Iberian Ceramics from the Upper Valley of Guadalquivir River*. Cham: Springer International Publishing, 2014, pp. 341–350.

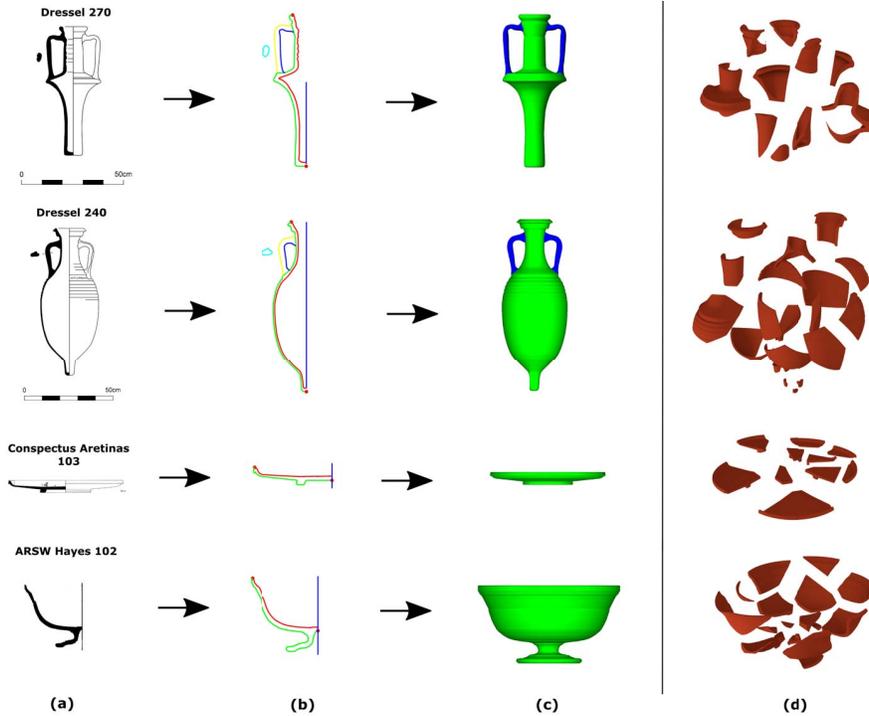


Figure 10: Four examples of the 3D model generation pipeline. (a) Original drawing. (b) Geometric features. (c) 3D model. (d) 3D sherds.

- [10] A. Gilboa, A. Karasik, I. Sharon, and U. Smilansky, "Towards computerized typology and classification of ceramics," *Journal of Archaeological Science*, vol. 31, no. 10, pp. 681–694, 2004.
- [11] M. Kampel and R. Sablatnig, "Automated segmentation of archaeological profiles for classification," in *Object recognition supported by user interaction for service robots*, vol. 1, 2002, pp. 57–60.
- [12] C. Maiza and V. Gaildrat, "Automatic classification of archaeological potsherds," in *The 8th International Conference on Computer Graphics and Artificial Intelligence (3IA'2005)*, May 2005, pp. 135–147.
- [13] M. Makridis and P. Daras, "Automatic classification of archaeological pottery sherds," *J. Comput. Cult. Herit.*, vol. 5, no. 4, pp. 15:1–15:21, Jan. 2013.
- [14] J. Poblome, P. Bes, and C. Piccoli, "Towards the automatic classification of pottery sherds: Two complementary approaches," in *CAA Conference Proceedings*, May 2013, pp. 463–474.
- [15] C. Ding and L. Liu, "A survey of sketch based modeling systems," *Frontiers of Computer Science*, vol. 10, no. 6, pp. 985–999, 2016.
- [16] J. Malik, "Interpreting line drawings of curved objects," *International Journal of Computer Vision*, vol. 1, no. 1, pp. 73–103, 1987.
- [17] C. Tian, M. A. Masry, and H. Lipson, "Physical sketching: Reconstruction and analysis of 3D objects from freehand sketches," *Computer-Aided Design*, vol. 41, no. 3, pp. 147–158, 2009. [Online]. Available: <http://dx.doi.org/10.1016/j.cad.2009.02.002>
- [18] Y. Wang, J. Liu, Y. Chen, and X. Tang, "3D reconstruction of curved objects from single 2D line drawings," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Los Alamitos, CA, USA: IEEE Computer Society, 2009, pp. 1834–1841.
- [19] H. Lee and S. Han, "Reconstruction of 3D interacting solids of revolution from 2D orthographic views," *Comput. Aided Des.*, vol. 37, no. 13, pp. 1388–1398, Nov. 2005.
- [20] B. Xu, W. Chang, A. Sheffer, A. Bousseau, J. McCrae, and K. Singh, "True2Form: 3D curve networks from 2D sketches via selective regularization," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 131:1–131:13, Jul. 2014.
- [21] M. Bessmeltsev, W. Chang, N. Vining, A. Sheffer, and K. Singh, "Modeling character canvases from cartoon drawings," *ACM Trans. Graph.*, vol. 34, no. 5, pp. 162:1–162:16, Nov. 2015.
- [22] X. Yin, P. Wonka, and A. Razdan, "Generating 3D building models from architectural drawings: A survey," *IEEE Computer Graphics and Applications*, vol. 29, no. 1, pp. 20–30, Jan. 2009.
- [23] R. C. Gonzales and P. Wintz, *Digital Image Processing (2Nd Ed.)*. Boston, MA, USA: Addison-Wesley Longman, 1987.
- [24] M. Visvalingam and J. Whyatt, "Line generalisation by repeated elimination of the smallest area," Cartographic Information Systems Research Group, University of Hull, Tech. Rep., 1992.
- [25] J. Davies. (2014) Visvalingam's algorithm: Avoiding intersections. [Online]. Available: <https://www.jasondavies.com/simplify/>
- [26] M. Meyerovitch, "Robust, generic and efficient construction of envelopes of surfaces in three-dimensional spaces," Master's thesis, Tel Aviv University, 2006.