International Journal of Computer Vision (2022) 130:2707–2732
https://doi.org/10.1007/s11263-022-01669-7

Snowvision: Segmenting, Identifying, and Discovering Stamped Curve Patterns from Fragments of Pottery

Yuhang Lu[1] · Jun Zhou[1] · Sam T. McDorman[2] · Canyu Zhang[1] · Deja Scott[1] · Jake Bukuts[1] · Colin Wilder[3] · Karen Y. Smith[4] · Song Wang[1]

Received: 2 April 2021 / Accepted: 27 July 2022 / Published online: 27 August 2022

Abstract
In southeastern North America, Indigenous potters and woodworkers carved complex, primarily abstract, designs into wooden pottery paddles, which were subsequently used to thin the walls of hand-built, clay vessels. Original paddle designs carry rich historical and cultural information, but pottery paddles from ancient times have not survived. Archaeologists have studied design fragments stamped on sherds to reconstruct complete or nearly complete designs, which is extremely laborious and time-consuming. In Snowvision, we aim to develop computer vision methods to assist archaeologists to accomplish this goal more efficiently and effectively. For this purpose, we identify and study three computer vision tasks: (1) extracting curve structures stamped on pottery sherds; (2) matching sherds to known designs; (3) clustering sherds with unknown designs. Due to the noisy, highly fragmented, composite-curve patterns, each task poses unique challenges to existing methods. To solve them, we propose (1) a weakly-supervised CNN-based curve structure segmentation method that takes only curve skeleton labels to predict full curve masks; (2) a patch-based curve pattern matching method to address the problem of partial matching in terms of noisy binary images; (3) a curve pattern clustering method consisting of pairwise curve matching, graph partitioning and sherd stitching. We evaluate the proposed methods on a set of collected sherds and extensive experimental results show the effectiveness of the proposed algorithms.

Keywords Curve-pattern segmentation · Design identification · Curve-pattern matching · Curve-pattern clustering · Swift creek complicated stamped pottery

1 Introduction

Hand-made pottery was widely produced throughout the pre-industrial world (Fu, 2019; Smith & Knight, 2012; Talcott, 1935; Snow, 1998). Almost universally in late stages of coiled vessel manufacture and occasionally in vessels that

Communicated by Takeshi Oishi.

Y. Lu and J. Zhou contributed equally to this work.

✉ Karen Y. Smith
SmithKY@dnr.sc.gov

✉ Song Wang
songwang@cec.sc.edu

Yuhang Lu
yuhang@email.sc.edu

Jun Zhou
zhouj@mailbox.sc.edu

Sam T. McDorman
mcdorman@email.sc.edu

Canyu Zhang
canyu@email.sc.edu

Deja Scott
ds17@email.sc.edu

Jake Bukuts
jbukuts@email.sc.edu

Colin Wilder
wildercf@mailbox.sc.edu

[1] Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA

[2] Department of Anthropology and School of Information Science, University of South Carolina, Columbia, SC, USA

[3] Department of History, University of South Carolina, Columbia, SC, USA

[4] South Carolina Department of Natural Resources, Columbia, SC, USA

were thrown on a potter's wheel, wooden paddles were used by potters to thin pottery vessel walls and finalize vessel form Roux (2019). In a subset of regions that used paddles in this way, particularly in southeastern Native North America, potters and woodworkers carved linear and curvilinear geometric designs into wooden paddles in stylistically coherent forms and transferred those designs to pottery vessels during the paddle stamping stage. Although these wooden paddles have not survived in the archaeological record, careful study of the pottery sherds bearing these stamps can lead to the reconstruction of unique paddle designs. These designs provide rich historical and cultural information on communities of learning and practice, migrations and smaller scale movements of people, and stylistic norms and creative conventions. In this paper, we present the Snowvision project and our efforts to digitally preserve and analyze paddle-stamped pottery produced throughout southeastern North America. We focus on the Swift Creek Complicated Stamped pottery type, created during the Middle and Late Woodland periods between roughly AD 100 and 800 (Smith et al., 2010). Figure 1(b) shows four such pottery fragments from the period 350–650 AD, when the stylistic tradition of carving wooden paddles was at its most complex (i.e., multiple different design elements arranged and combined in unique ways on the paddle).

Carved paddle designs were primarily composed of connected and intertwined curved lines. The same paddle was usually applied to many different locations on the pottery vessel's exterior surface to achieve the desired decorative effect. Also, the same paddle may be applied to many different vessels, and sometimes applied to pottery found at different sites, indicating the movement of paddles and/or people. Current existing designs were composed manually by experienced archaeologists (Broyles, 1968; Snow, 1975). With access to extensive collections from key excavation sites, the archaeologists assembled sherds belonging to the same paddles, traced or measured sherd curves on individual sherds, and looked for regions with overlap but also with new design information. Slowly the sherd information is complied until a paddle design is complete. An example of design reconstruction is illustrated in Fig. 1(b). The archaeologists first visually identify a number of pottery sherds from the same paddle (top left), and then match the curve patterns on the sherd surfaces to construct larger design pieces (top right) until the final full design is constructed (bottom left). More fragmented pottery sherds can then be identified as belonging to their complete designs by matching their curve patterns to a set of composed designs (Gamble, 2004), as its procedure illustrated in Fig. 1(c). The association between the pottery sherds and their underlying designs provides unique evidence of interactions among potters in Native Societies of southeastern North America. An important problem in our project is to facilitate this process using computer-vision techniques, for manually assessing a large number of objects can be extremely time-consuming, especially when these objects are highly fragmented.

Snowvision, a project named after the archaeologist Frankie Snow (Snow, 1975), is an interdisciplinary effort for leveraging state-of-the-art computer vision techniques to explore and study paddle designs from large collections of pottery sherds. It provides standards and guidelines for digitizing sherds and developing rich and accurate metadata, a frontend website from which archaeologists at different institutions can submit and share their sherds, and a set of backend advanced computer vision algorithms to assist archaeologists to more efficiently explore the underlying designs of the sherds. An acquisition procedure including sherd digitization and preprocessing is developed, and a collection of pottery fragments and reconstructed designs is available to the public for study. At the core of the backend processing are three important computer-vision tasks with unique challenges:

- Task I: Accurately segmenting sherd images for the stamped curve pattern;
- Task II: Identifying whether the curve pattern on a sherd matches a known design;
- Task III: Identifying and grouping sherds with the same but unknown design.

The curve patterns produced in Task I are fed to Tasks II and III and archaeologists can take the identification results produced by Tasks II and III for design study, including discovering and reconstructing new designs.

These three computer-vision tasks are challenging for several reasons. First, sherds are highly fragmented and often only show a small portion of the underlying full design. While multiple sherds may have the same design, i.e., stamped by the same paddle, they are usually from different vessels and cannot be assembled by fitting their sherd boundaries as in many other studies. Second, a paddle is usually applied to the vessel multiple times with partial overlap and varying orientations. As a result, segmented curves from a sherd can be a *composite pattern*, which can not be directly matched as a portion of a full design, as shown by an example in Fig. 2. This way, even between a sherd and its underlying full design, the curve-pattern matching can be partial-to-partial instead of partial-to-global. Finally, the sherds can be eroded, deformed and/or faintly stamped. They usually produce many false positives and negatives in Task I, which further increases the difficulty of matching and identification in Tasks II and III.

By considering these challenges, in this paper we develop new computer vision algorithms to address the three tasks, respectively. In Task I, we try to segment weakly stamped curve structures from the depth images of pottery sherds. The blurry boundary and various noises make the segmen-

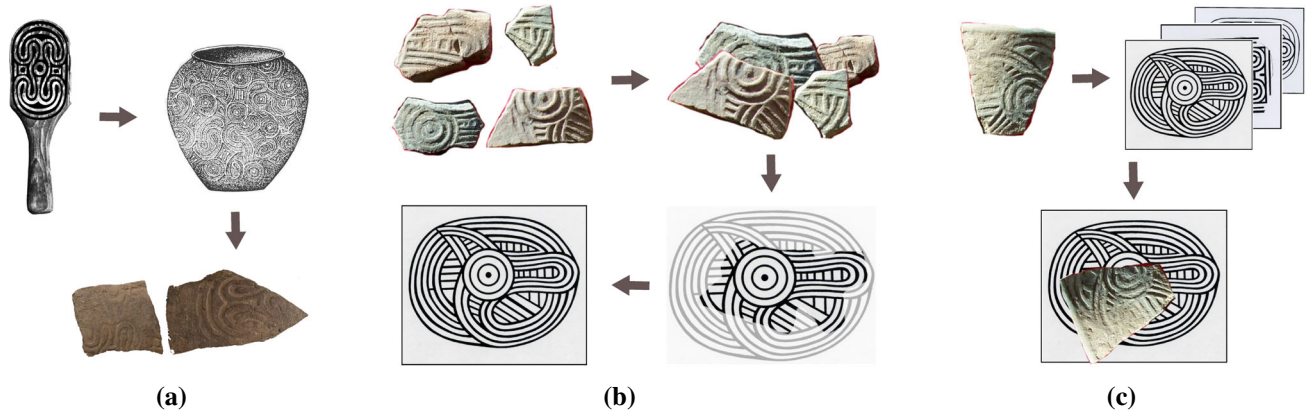

Fig. 1 An illustration of **a** the lifecycle of a paddle design; **b** the reconstruction of a paddle design from sherds; and **c** the identification of a design on a sherd. (Vessel in (**a**) and original designs in (**b**) and (**c**) reproduced with permission, courtesy of Frankie Snow, South Georgia State College.)

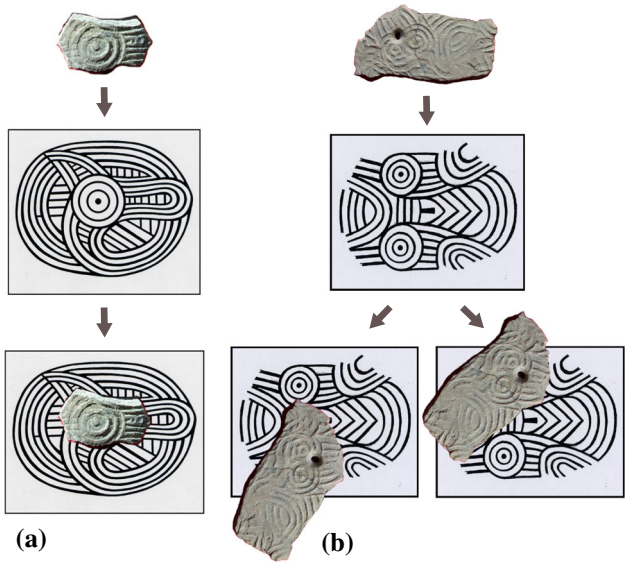

Fig. 2 An illustration of matching a sherd with **a** single pattern and **b** composite pattern to their ground-truth designs. The sherd with composite pattern contains two single patterns that each matches to the design at its own location. (Original designs reproduced with permission, courtesy of Frankie Snow, South Georgia State College.)

tation difficult for traditional non-learning-based methods. Recent CNN-based methods are robust to noise, but they require a large number of pixel-wise ground-truth masks for training, which are very difficult to obtain. By learning and incorporating the curve geometry implied in the underlying designs, we propose a weakly supervised CNN-based method which could learn curve structure segmentation from skeleton labels. It can achieve much better sherd segmentation with significantly reduced annotation costs. More specifically, we train a Fully Connected Network (FCN) to detect the skeletons of the curve patterns in the *depth images* of sherds. Then we train a dense prediction convolutional network to identify and prune false positive skeleton pixels. Finally, we recover the curve width by a scale-adaptive thresholding algorithm to get the final segmentation, in the form of a *binary curve-pattern image*.

In Task II, we match the curve pattern segmented from a sherd to every known paddle design and then select a small set of designs with the highest matching scores for archaeologists to make the final assessment. Since binary images of noisy curve patterns could not provide consistent and discriminative image features, it is very difficult for traditional keypoint-based methods to achieve good performance. Other matching methods cannot handle the challenges caused by partial matching and composite patterns. To address this problem, we propose a patch-based matching method by decomposing curve patterns into patches and establishing dense patch correspondence in the deep feature space. It combines the advantages of template matching and keypoint matching by finding local matching without detecting keypoints. Specifically, we apply uniform sampling for constructing patches and employ a learning-based curve feature descriptor to derive a heatmap for local similarity between the sherd and the design. We consider both the heatmap value and the overlap area in deciding the matching between sherd and designs.

In Task III, given a collection of sherds with unknown designs, we cluster them in terms of their underlying (unknown) designs. With the goal of identifying and recreating new designs, we expect the sherds in the same cluster to show only partial/local overlap or even no overlap but local similarity to other common sherds in the cluster. Many existing clustering algorithms rely on global image features and therefore cannot handle our problem well. In this paper, we formulate pottery sherd clustering as a graph partitioning problem and propose a new iterative clustering pipeline consisting of pairwise matching, graph partitioning and sherd stitching. The proposed algorithm integrates local feature matching into the image clustering pipeline. In specific,

given the segmented curve-pattern images of a collection of sherds, we first conduct patch-based pairwise matching between each pair of sherds to construct a fully connected graph, and then partition it into subgraphs using an adaptive thresholding algorithm. We last propose an iterative cluster refining strategy, with curve-pattern stitching in the iteration, for identifying and refining the sherd clustering.

In the experiments, we quantitatively evaluate our proposed algorithms for the three tasks on a collection of pottery sherds from key excavation sites. We also report comparison experiments with other state-of-the-art methods to justify the effectiveness of the proposed algorithms. Preliminary result of Task I has been published in a conference (Lu et al., 2018).

While our paper is focused on a specific application in archaeology, which is important to the archaeology community, we also make new contributions to the computer vision community as below:

(1) We consider the unique challenges of the three tasks and make new contributions in algorithm/model development, e.g., weakly-supervised skeleton-aware segmentation model in Task I, deep-learning-based patch matching in Task II, and graph-based curve-pattern clustering in Task III, as described above. The studied problems extend the research scope of segmentation, matching, and clustering in computer vision. The proposed algorithms and methods may also benefit the research on other applications involving curve-structure images, such as pavement crack detection (Li et al., 2018), vessel/nerve segmentation (Fu et al., 2020), and footwear impression matching (Kong et al., 2019).
(2) We collect high-quality point clouds of thousands of real pottery sherds, with manual annotations as ground truth. This is highly laborious and time-consuming, and we will release this dataset, together with the manual annotations and the code of evaluation, to public. This will help promote the research in the computer vision community to interesting problems in archaeology.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 briefly introduces the Snowvision system and its components. Section 4 elaborates on the developed algorithms for addressing the above-mentioned three computer-vision tasks. Section 5 introduces the evaluation dataset and the experimental results, followed by a brief conclusion in Sect. 6.

2 Related Work

In this section, we briefly review prior works related to the Snowvision project and the proposed three computer-vision tasks.

Snowvision aims to leverage computer-vision techniques to explore pottery sherds that are important to archaeologists. Many prior projects, such as CATA (Martínez-Carrillo, 2008) and ArchAIDE (Gualandi et al., 2021; Anichini et al., 2020), employ the profile, color, texture, rotational axis, and geometric shape information to classify and determine whether different sherds are fragments of the same vessel (Lucena et al., 2016; Banterle et al., 2017; Han & Hahn, 2014; Makridis & Daras, 2012; Ostertag & Beurton-Aimar, 2020; Smith et al., 2010; Kampel & Sablatnig, 2003; Rasheed & Nordin, 2018) and then develop matching algorithms to reconstruct the whole vessel by assembling the involved sherds (Kampel & Sablatnig, 2003; Son et al., 2013; Stamatopoulos & Anagnostopoulos, 2016; Willis et al., 2003; Willis & Cooper, 2004). One may think of these prior works as solving a jigsaw problem. However, Snowvision handles sherds that are usually not from the same vessel. Even if sherds were from the same vessel, it is not productive from a design perspective to use them to reconstruct a partial or whole vessel because the design would still be incomplete. In Snowvision, we focus on the curve patterns on sherds and try to identify/reconstruct the full paddle design underlying each sherd. Most jigsaw algorithms based on the fragment-shape fitting and color/texture consistencies are incompatible with our problem.

Our Task I is focused on curve-pattern segmentation, which has been studied in many specific applications, such as crack detection from pavement images (Zou et al., 2012), and the segmentation of blood vessels (Lorigo et al., 2001) and outer-cortex sulcals (Tao et al., 2002) from medical images. The images in these applications are mainly RGB or gray-scale ones, and the corresponding segmentation algorithms usually rely on specific domain knowledge not applicable to Snowvision, where the inputs are scanned depth images of pottery sherds. General-purpose image segmentation such as low-level methods based on edge detection (Arbelaez et al., 2011; Wang et al., 2004), region growing/splitting (Tremeau & Borel, 1997), pixel clustering (Li & Chen, 2015), or graph cuts (Shi & Malik, 2000; Wang & Siskind, 2003); and mid-level methods, such as active contours (Chan & Vese, 2001; Vese & Chan, 2002), LevelSet (Vese & Chan, 2002; Li et al., 2010), or GrabCut (Rother et al., 2004) can be adapted to handle our problem of segmenting the depth images by treating depth value as image intensity. However, their segmentation performances are usually poor given weak stamps and noise on the sherd.

Deep-learning based segmentation algorithms have been developed by learning high-level features of the desired segments in a supervised way (Badrinarayanan et al., 2015; Zheng et al., 2015), such as Fully Convolutional Network (FCN) (Long et al., 2015), DeepLab (Chen et al., 2016, 2018) and HRNet (W. et al., 2020). However, the performance when directly applying these deep-learning algorithms to our sherd depth image is also not satisfactory without considering the

the curve-geometry and curve-pattern features. By considering more geometry information, Deep-Skeleton (Shen et al., 2016) uses DNNs for skeleton extraction, which, however, is not specifically developed for curve patterns and may produce many false positive skeletons when used for segmenting our sherd depth images. In this paper, we propose a new CNN-based network to better segment curve patterns by considering more curve geometric features.

At the core of our Task II is actually a curve-pattern matching problem that has been studied for a long time in computer vision. For example, Frenkel and Basri (2003) proposed to match two curves by deforming them into one identical curve and then a graph-based fast-marching algorithm is used to find the optimal deformation as the curve matching cost. However, this method requires all the curves to be represented by a set of sampled points with a consistent order, which is infeasible in our problem given various image-segmentation errors and the nature of partial matching. The low-level matching algorithms, such as template matching (Brunelli, 2009), Chamfer matching (Barrow et al., 1977), and shape context (Belongie et al., 2001) can be applied to match the binary images of the curve patterns on sherd and design. However, these methods are sensitive to noise and not able to handle the partial-to-partial correspondence in composite patterns matching. The pipeline of keypoint detection, description and matching has been widely adopted in natural image matching. Another typical approach is keypoint matching. It includes handcrafted-feature-based methods, such as SIFT (Brown & Lowe, 2007), SURF (Bay et al., 2006) and ORB (Rublee et al., 2011); and learning-based methods, such as LIFT (Yi et al., 2016), LF-Net (Ono et al., 2018) and RF-Net (Shen et al., 2019). These methods could find partial-to-partial correspondence by matching keypoints such as corners and edges. However, such keypoints are rare in texture-less binary curve pattern images. There are also feature descriptors for texture-less objects, such as BOLD (Tombari et al., 2013) and BIND (Chan et al., 2017). But these methods are proposed for partial-to-global object detection, and not applicable to our Task II.

Curve-pattern matching is also related to fingerprint matching (Jain et al., 2001), in which minutiae features are extracted at ridge bifurcations or ridge endings. However, the curve patterns on sherds usually carry much less minutia than that in fingerprints and therefore, algorithms developed for fingerprint matching usually perform poorly on this task. Curve-pattern matching can also be regarded as a region-based image retrieval (RBIR) problem (Carson et al., 1999), of which the goal is to find regions in the target image that share similar features with a specified region on the query image. For example, Hinami et al. (2017) proposed a region-based network whose multi-task CNN features can jointly handle multi-aspect object specification. However, these methods assume the region of interest is known on the query image. In our study, without knowing the curve-pattern composition in advance on a sherd, directly applying these methods may fail in handling composite patterns. In this paper, we develop a new patch-based matching method based on deeply learned features to better address the proposed curve-pattern matching problem.

In the ArchAIDE project (Gualandi et al., 2021; Anichini et al., 2020), researchers study a shape-based recognition problem of matching sherd profiles to vessel profiles, where each profile consists of a single curve that is represented by a sequence of sampled points along the profile in vector graphics. Differently, in our proposed matching tasks, each curve structure consists of multiple disjoint curves and curve width is also an important cue in matching. Besides, profile matching in ArchAIDE is a partial-to-global matching problem, while the proposed curve-pattern matching is partial-to-partial.

Also related to our work is Pirrone et al. (2019), in which a patch-based two-branch Siamese network, named Papy-S-Net, was developed to check whether two papyrus fragments are from the same papyrus piece. It tackles a problem more related to general-purpose image classification - matched fragments (patches) show no spatial overlap and the learned deep features reflect the general appearance and text style present in image patches. Differently, the sherd/design matching in our work indicates their (partial) structural overlap, embodied by the matching location and orientation.

Our Task III of identifying sherds from the same unknown design can be formulated as a curve-pattern clustering problem. Cluster analysis is a long-studied topic, resulting in many classical algorithms, such as K-means (Lloyd, 1982), K-Means++ (Arthur & Vassilvitskii, 2006), Affinity Propagation (Frey Dueck, 2007), MeanShift (Wu & Yang, 2007), Hierarchical Clustering (Jain & Dubes, 1988), Spectral Clustering (Shi & Malik, 2000), and more recent deep-learning based algorithms, e.g., DeepCluster (Caron et al., 2018), DAC (Chang et al., 2017), and DCCM (Wu et al., 2019), ARO (Otto et al., 2017) and GCN Clustering (Wang et al., 2019). Unlike conventional image clustering problems, the curve-pattern clustering aims at finding sherds with common partial curve patterns. Existing algorithms mostly take the input image as a whole, thus they are not directly applicable to our problem. In this paper, we combine the divide-and-conquer strategy and deep metric learning, and proposed a partial-feature-aware graph-based curve-pattern clustering method to solve the problem of sherd identification.

3 Snowvision

Some 1,820 archaeological sites with Swift Creek pottery have been recorded across the states of Alabama, Florida, Georgia, and South Carolina (Smith & Stephenson, 2018).

Hundreds of thousands of pottery sherds from these sites reside in curation facilities and archaeological laboratories across the Southeast. Making sense of these large and scattered collections is a daunting task for researchers. The Snowvision project takes on this challenge with state-of-the-art research in computer vision and archaeology. At its core, Snowvision is a backend study of these heritage objects using identification and segmentation of curve patterns on the pottery fragments. This involves the digital acquisition of pottery fragments and reconstructed paddle designs. These cultural heritage objects are stored in the free and public World Engraved digital archive to centralize and share collective knowledge about the diversity and geographical reach of the designs over time.

World Engraved will act as an online repository to collect, organize, and disseminate sherds and design reconstructions from the many institutions that hold Swift Creek pottery. To gain an understanding of the expected user group and gather feedback on the World Engraved website, a two-part user needs study was conducted between November 2019 and March 2020. All respondents were professional archaeologists interested in contributing sherd and design data, but the user population is expected to include students, artisans, curators, and collectors who may be members of the general public or descendent Native Nations. Feedback from the user needs study has helped refine the collected metadata, increase usability of the World Engraved website, and has assisted in the development of data policies. Data policies include the protection of sensitive data from public view, data sharing that meets the needs of users through a Creative Commons license, and metadata collection that credits all contributors and workers involved in a scanning project.

This section introduces the digitization pipeline of Snowvision and the architecture of World Engraved, the Snowvision system where scientists can interactively share computing resources, data, and expertise in an academic cloud system. To date, the project has digitized more than 8,000 sherds using this Snowvision digitization pipeline and is expected to increase this number to at least 25,000 as the project continues to grow. More than 800 paddle design reconstructions have been digitized, and 220 have been scaled by calibrating them to the associated pottery sherd depth scans. Detailed metadata about the sherds and design reconstructions is collected and published with the objects. Subsets of the scanned sherds and scaled designs were used in the later experiments.

3.1 Snowvision Digitization Pipeline

Common methods of digitizing cultural heritage objects include RGB photography, depth imaging, point cloud, 3D mesh, normal map, and many others. All known paddle design reconstructions were manually drawn by archaeologists, primarily Bettye J. Broyles and Frankie Snow. The Snowvision digitization pipeline uses a digital camera and a high-precision 3D scanner to collect high-quality RGB images and depth scans of the surface of pottery sherds. The RGB images are easier for the human eye to read than the depth scans, but color differences in the RGB images due to lighting, pottery manufacture, and depositional conditions make them unsuitable for curve extraction. Therefore, RGB photographs of the pottery sherds are collected for display in World Engraved but not used in the proposed three computer-vision tasks. Since the carved paddles impress ridges and valleys on the surface of a vessel and depth imaging records 3-dimensional data from the pottery sherds, a depth scan captures all the information required for analyzing curve patterns.

The depth difference between the ridge and valley on the surface of sherds is quite small, typically in the range of $1mm$ to $3mm$. The NextEngine Ultra HD, a linear array laser scanner with an accuracy of $\pm 0.127mm$, is used to capture this low-level depth difference. A dense point cloud of the sherd surface is captured and then converted into a depth scan. The scanner is placed on a four-leg stand face-down to capture sherds placed below on a flat surface, with the impressed exterior sherd surface facing up towards the scanner. Multiple sherds can be scanned at one time for efficiency, as long as they are not too close to appear whole in the scanning area. An example of three sherds scanned together is shown in Fig. 3.

The 3D scanning process captures a point cloud with density of 6, 200 points per cm^2, and each scan takes approximately 5 minutes. The points corresponding to the surface below the sherds are removed using the 3D coordinates, and the KD-tree is employed to obtain the point cloud of each sherd when multiple sherds are scanned at one time. In KD-tree, points are grouped for each sherd using a threshold of $1mm$ for the spatial distance. From the point cloud of each sherd, a depth image is generated by grid sampling with a resolution of $10,000$ pixels per cm^2 and the values of the depth image are normalized to $[0, 255]$. A mask image is also constructed to identify the irregular shape of the sherd. Finally, as a fragment of a rounded vessel, the sherd surface is curved instead of planar. An adaptive histogram equalization method CLAHE (Pizer et al., 1990) is applied to enhance the contrast between sherd regions so that the vessel curvature can be removed from the depth scan.

As an alternative to depth scan, normal maps have been used to digitize other cultural heritage objects (Yeh et al., 2016; MacDonald, 2015). Comparing with depth scans, normal maps are good at preserving tiny bumps by focusing on higher-order normal directions instead of depth values. However, on the excavated sherds in our work, normal direction is more sensitive to local geometry change/noise on the surface and normal directions along the curve structures may not be consistent, which increases the difficulty of the segmenta-

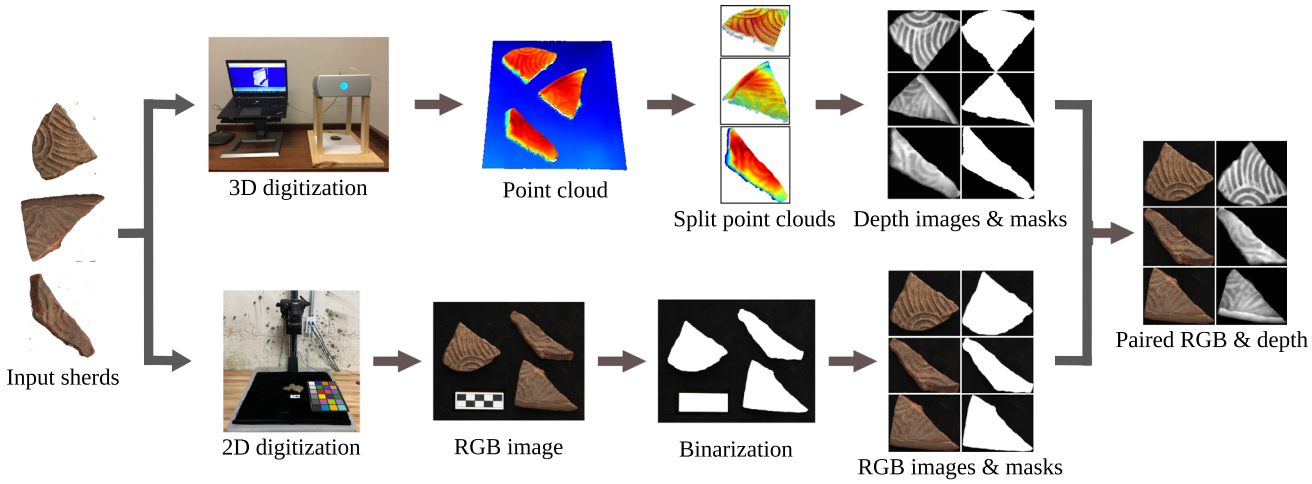

Fig. 3 The Snowvision digitization pipeline

tion task. Therefore we choose depth scan over normal map in Snowvision.

RGB photographs taken with controlled lighting and scale cards are standard in archaeological data collection and dissemination so that information about the color and size of an artifact can be communicated through digital imagery. A digital SLR camera with a 60 mm lens is used to capture RGB images. Like the 3D scanning process, multiple sherds are placed on a flat surface with the camera above and face-down and captured in one RGB image to expedite the digitization process. A sherd splitting algorithm was developed to divide a RGB image with multiple sherds into multiple RGB images each with a single sherd, and then to match such single-sherd RGB images to their corresponding depth scans. Specifically, as illustrated in Fig. 3 (bottom), given a multi-sherd RGB image, we binarize it and retrieve the boundaries of sherds by intensity thresholding. We then conduct shape matching between the sherd boundaries extracted from RGB images and the sherd masks extracted from depth scans using Hu-Moment values (Žunić et al., 2010). This pre-processing procedure can pair the RGB and depth images of the same sherds in the Snowvision database.

3.2 Snowvision System Architecture

Our Snowvision system, called World Engraved, consists of four major modules: a UI frontend service, a UI backend service, a storage service and a backend service. Its architecture is illustrated in Fig. 4. The UI frontend service allows the researchers to submit sherds for Snowvision processing and provides an interface for users to view and search for designs and sherds, as well as the segmentation and identification results. The UI backend service serves as a hub to transfer data between the UI frontend, the storage and the backend services. The storage service stores all digital objects and their metadata. The backend service allows users to submit unidentified sherds in RGB images and 3D scans with their metadata. The point clouds will be first converted to depth maps and separated into multiple individual sherds. RGB images are separated as discussed above. Users can interact with individual sherds to edit, publish, process sherds for sherd segmentation (Task I), design identification (Task II), sherd identification (Task III), result reviewing, or entry deletion. When sherds are published, the images and metadata are added to the World Engraved public archive.

4 Proposed Computer-Vision Methods

Given a set of query sherds, we take their scanned depth images and carry out Task I of *curve-pattern segmentation* by producing *binary curve-pattern images*, as shown in Fig. 4. Then we carry out Task II of *design identification* by comparing the curve patterns segmented from each sherd with each known design in a pairwise way. A high matching score between a sherd and a design may indicate that this sherd is probably stamped with the design, and we send them to archaeologists for final confirmation. For the sherds that cannot be matched to a known design, we carry out Task III of *sherd identification* by clustering them into different groups, expecting that each group consists of sherds with the same (unknown) design. Clustering results are then sent to archaeologists for confirmation and possible discovery and reconstruction of new designs. In this section, we follow the pipeline in Fig. 4 to introduce the computer-vision algorithms developed for the above three tasks.

4.1 Task I: Curve-Pattern Segmentation

The proposed curve-pattern segmentation method consists of three parts: (1) a fully convolutional network (FCN) for

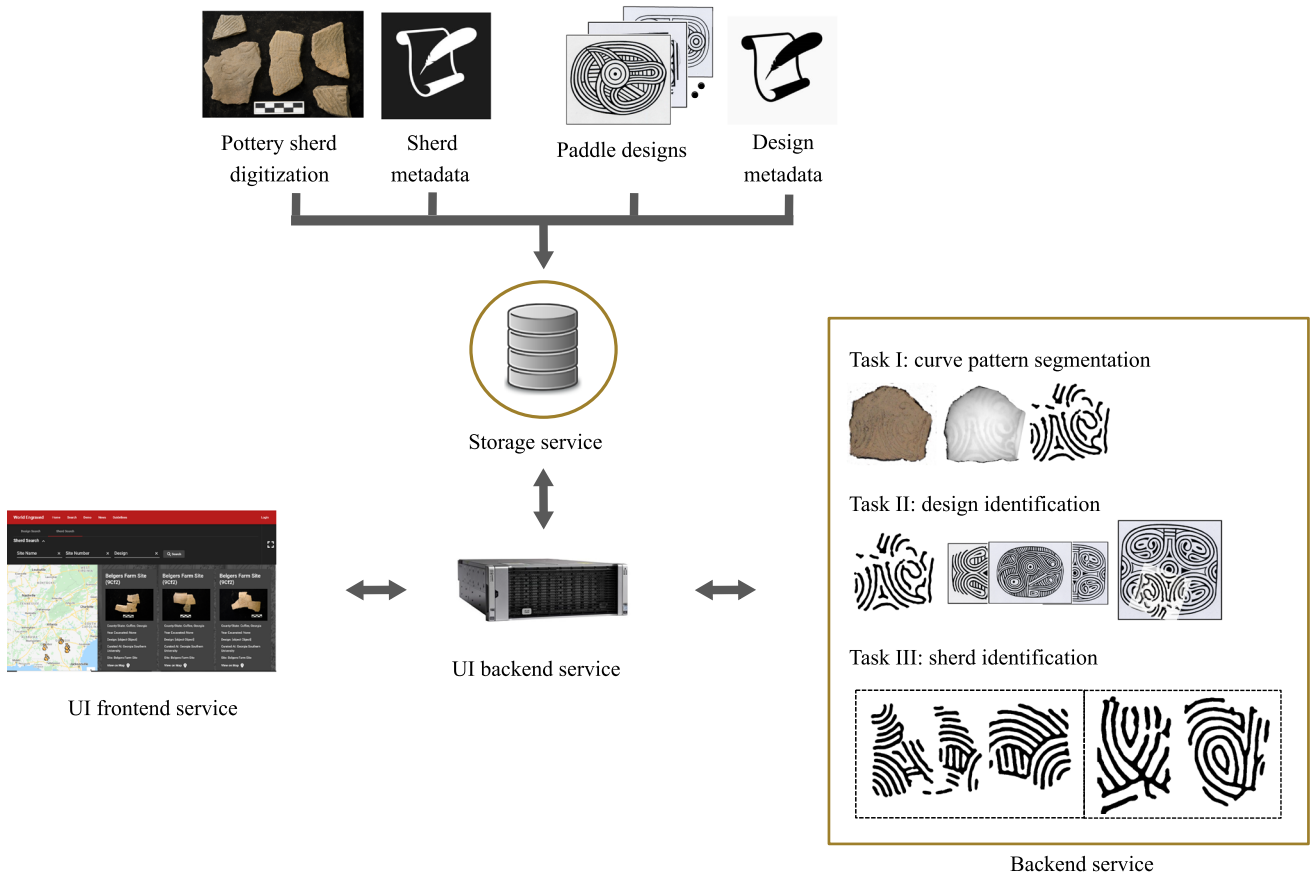

Fig. 4 The architecture of Snowvision (Original designs reproduced with permission, courtesy of Frankie Snow, South Georgia State College)

simultaneously predicting the curve skeleton map and estimating the curve width at each skeleton pixel, (2) a dense prediction ConvNet for refining the curve skeletons, and (3) an adaptive thresholding algorithm for the final segmentation by restoring the curve width. We expect the use of deep neural networks can improve the robustness to noise and errors in the input depth images and the extraction and processing of curve skeletons can better leverage the geometry and shape information underlying the curve patterns.

4.1.1 Step 1: Detecting Curve Skeletons using FCN

We train an FCN to detect the skeleton of the curve patterns from an input depth image, which is formulated as a pixel-labeling problem: skeleton pixel has label 1 and non-skeleton pixel has label 0. The FCN network architecture is illustrated in Fig. 5, by following the encoder-decoder architecture developed in Long et al. (2015). Encoders 1 and 2 are small ConvNets made up of two 3×3 convolutional layers, two ReLu layers and one 2×2 max-pooling layer. Encoder 3 is a small ConvNet made up of three 3×3 convolutional layers, three ReLu layers and one 2×2 max-pooling layer. After an encoder, the image size will be reduced to 1/4. Therefore,

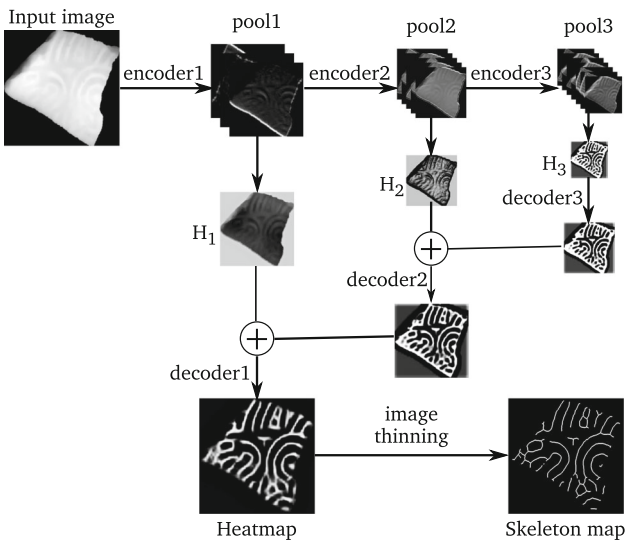

Fig. 5 Network architecture of the FCN used for skeleton detection

the receptive field sizes of feature maps generated by the three encoders are 2×2, 4×4, and 8×8, respectively. After each encoder, a fully connected layer is employed to match the number of feature maps with the number of labels. In

order to generate pixel-wise prediction result, the fully connected layers are implemented by 1×1 convolutional layers. These results are denoted as H_1, H_2 and H_3, respectively, as shown in Fig. 5. Note that the size of H_1, H_2 and H_3 are successively downsampled by factors of 2, 4, and 8 from the original image size. The decoders are three deconvolution layers with a kernel size of 4×4 and a stride of 2. The kernels are fixed to perform bilinear interpolation (Xie & Tu, 2015).

The use of multiple encoders/decoders can extract image features in different levels of details. To make full use of all the extracted features, the decoders are organized in a way of stepwise accumulation when fusing them together. The output skeleton heat map H can be computed by:

$$H = softmax(\Psi^{(2)}(H_1 + \Psi^{(2)}(H_2 + \Psi^{(2)}(H_3)))), \quad (1)$$

where Ψ indicates the upsampling operation performed by the decoders and its associated superscript is the upsampling factor, e.g., $\Psi^{(2)}$ indicates an upsampling of map by a factor of 2. With the skeleton heat map H, we apply a common image thinning algorithm (Lam et al., 1992) to generate the single-pixel width skeleton map $\hat{P}$.

Inspired by Shen et al. (2016), we can compare the three score maps H_1, H_2 and H_3 to estimate the scale at each detected skeleton pixel. The scale value at a skeleton pixel reflects the local curve width at this pixel. More specifically, since different encoders correspond to different receptive field sizes, at each pixel the receptive field size of the encoder with the largest score reflects the scale at this pixel. Before we compare the score of different maps, we need to first upsample them to the original image size. This way, the scale $s(x, y)$ at the skeleton pixel (x, y) can be computed by:

$$s(x, y) = \arg \min_{k \in \{1,2,3\}} \hat{H}_k(x, y), \quad (2)$$

where $\hat{H}_k = \Psi^{(2^k)}(H_k)$ is the upsampled score map of H_k. Later we will use the estimated scale values to help recover the curve width.

4.1.2 Step 2: Refining Skeletons using Dense Prediction ConvNet

While we expect the trained FCN can learn curve geometry and pattern features, the detected skeletons may still contain many false positives, which are difficult to remove by tuning the FCN itself, as shown in Step 1 of Fig. 6. In Step 2, we train a supervised classifier to identify and prune such false positives by learning more curve features. Specifically, centered at each skeleton pixel (x, y) detected by FCN, we take its 45×45 neighboring region in the original depth image as the input and train a dense prediction ConvNet to determine whether (x, y) is a true skeleton pixel or a false positive.

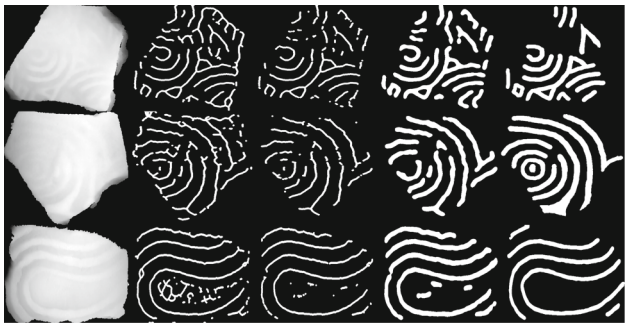

Fig. 6 Example results after each step of the proposed curve-pattern segmentation method

In practice, it is very difficult to perfectly localize the ground-truth skeletons on the sherds at a pixel-level accuracy. Instead of training a hard classifier, we train a soft classifier to predict the probability of being skeleton pixels. Therefore, we convert the ground-truth binary skeleton map to a smoothed skeleton probability map by:

$$Q(x, y) = \frac{1}{1 + \min_{(x', y') \in G} \sqrt{(x - x')^2 + (y - y')^2}}, \quad (3)$$

where G is the set of skeleton pixels in the ground-truth skeleton map. Using Q to guide the training of the network, the binary classification problem is converted to a regression problem. Accordingly, we use sigmoid instead of softmax in the last layer of the proposed dense prediction ConvNet.

In this paper, we propose to use a ConvNet consisting of three convolutional layers, three max-pooling layers and two fully connected layers. Its specific configuration is summarized in Table 1. For a testing image, let the set of the skeleton pixels detected by FCN be $\hat{P}$ and the skeleton probability map generated by ConvNet in this step be $\hat{Q}$, we prune the low-probability (< 0.5) skeleton pixels in $\hat{P}$ to achieve a refined set of skeleton pixels as:

$$P = \{(x, y) | (x, y) \in \hat{P} \wedge \hat{Q}(x, y) \geq 0.5\}. \quad (4)$$

4.1.3 Step 3: Final Segmentation by Recovering Curve Width

In the final step, we recover full curves from the refined skeleton map using estimated curve widths. Denote the original depth image by I and let P be the set of skeleton pixels refined in Step 2. For each skeleton pixel $(x, y) \in P$, we have a scale value $s(x, y)$ derived in Step 1. According the responses of three encoders in Step 1, the scale values are roughly esti-

Table 1 The configuration of the dense prediction ConvNet, where n, k, st, pd are the number of output channels, kernel size, stride and padding size, respectively. ReLu and BatchNorm layers are not included

Type	Configuration
Sigmoid	–
Fully connected	n:2
Dropout	ratio:0.5
Fully connected	n:512
MaxPooling	k:2 × 2, st:2
Convolution	n:128, k:3 × 3, st:1, pd:1
MaxPooling	k:2 × 2, st:2
Convolution	n:64, k:3 × 3, st:1, pd:1
MaxPooling	k:2 × 2, st:2
Convolution	n:32, k:3 × 3, st:1, pd:1
Input	45 × 45 gray-scale image

mated to be 1, 2 or 3, corresponding to the maximum curve widths of 4, 8 and 16, respectively. Simply dilating skeletons with these rough widths will not generate accurate curve masks. Instead, we construct the curve-pattern segmentation, in the form of a binary map S of the same size as I, using the following Algorithm 1.

The core idea of Algorithm 1 is to search curve pixels in the neighborhood of skeletons using an adaptive thresholding strategy. Around each skeleton pixel, we search in a circular neighborhood with a radius of the predicted width, in which curve pixels are supposed to have greater intensity than non-curve pixels. Considering that the depth of different curves may vary, we set an adaptive intensity threshold around each skeleton pixel. Traditionally, the threshold can be set as the average intensity of a neighborhood. But in our circular neighborhood, we usually have more curve pixels than non-curve pixels, thus the average intensity can be higher than the desired one. To address this issue, we use only the intensity of the skeleton pixel and the smallest intensity in the circular neighborhood to determine the threshold, as formulated in the line 5 of Algorithm 1. This algorithm does not require the detected skeleton to be exactly aligned with the center line of the curves – a small dislocation between them may not change the final segmentation – it only requires the detected skeletons to be located inside the underlying curve regions. Sample results after this step are shown in Fig. 6.

4.2 Task II: Design Identification by Curve-Pattern Matching

In the past decades, archaeologists have manually reconstructed hundreds of complete designs from fragmented pottery sherds. In this section, we address the problem of identifying whether the curve pattern on a query sherd comes from one of these known designs. For this we match the

Algorithm 1 Curve-pattern segmentation by recovering curve width

Input: Depth image I; Refined skeleton P; Scale values s
Output: Binary segmentation map S

1: Initialize all the elements in S to zero.
2: **for** each skeleton pixel $(x, y) \in P$ **do**
3: Compute neighborhood:
$\mathbf{N} = \left\{(x', y') | \sqrt{(x - x')^2 + (y - y')^2} \leq 2^{s(x,y)}\right\}$.
4: **for** each pixel $(x', y') \in \mathbf{N}$ **do**
5: **if** $I(x', y') \geq \frac{I(x,y) + \min_{(x'',y'') \in \mathbb{N}} I(x'', y'')}{2}$ **then**
6: $S(x', y') = 1$
7: **end if**
8: **end for**
9: **end for**

curve pattern segmented from the query sherd to each known design and the best matched designs are identified and sent to the archaeologists for confirmation. Since the query sherd may exhibit composite patterns, this is not a partial-to-global matching. Instead, it is a partial-to-partial matching, i.e., only a portion of the curve pattern segmented from the query sherd is matched to a portion of the underlying design, as illustrated in Fig. 2. In this paper, we mainly consider possible translation and rotation transforms when matching a sherd to a design, since the scales of them have been calibrated in their digitization, as discussed in Sect. 3.1.

To handle the partial-to-partial matching between sherd and design, we proposed in this section a new patch-based curve-pattern matching algorithm that consists of three steps: (1) dividing and sampling the binary curve-pattern images of the query sherd and the design for small image patches, (2) employing a triplet CNN network to evaluate the pairwise matching score between sherd and design, and (3) identifying the matching portions by region growing. The whole sherd-to-design matching process is shown in Fig. 7.

4.2.1 Step 1: Patch Sampling from Sherd and Design

For the query sherd, the proposed segmentation algorithm in Sect. 4.1 converts it to a 2D binary image of curve pattern. The considered design also takes form of a binary image by digitization. We first perform a thinning operation to convert both binary images into skeleton images, which are also binary. We then uniformly divide each of the skeleton images to non-overlapping grids of size $g \times g$. If a grid contains one or more skeleton pixels, we take the skeleton pixel closest to the grid center as the patch center, and then take its surrounding $p \times p$ region as a patch, but on the curve-pattern image before thinning. This process is briefly illustrated in Fig. 8. With this patch sampling strategy, we obtain two sets of sherd and design patches for matching. We only need to consider one patch for each grid instead of for each pixel, and so that the searching space can be significantly reduced.

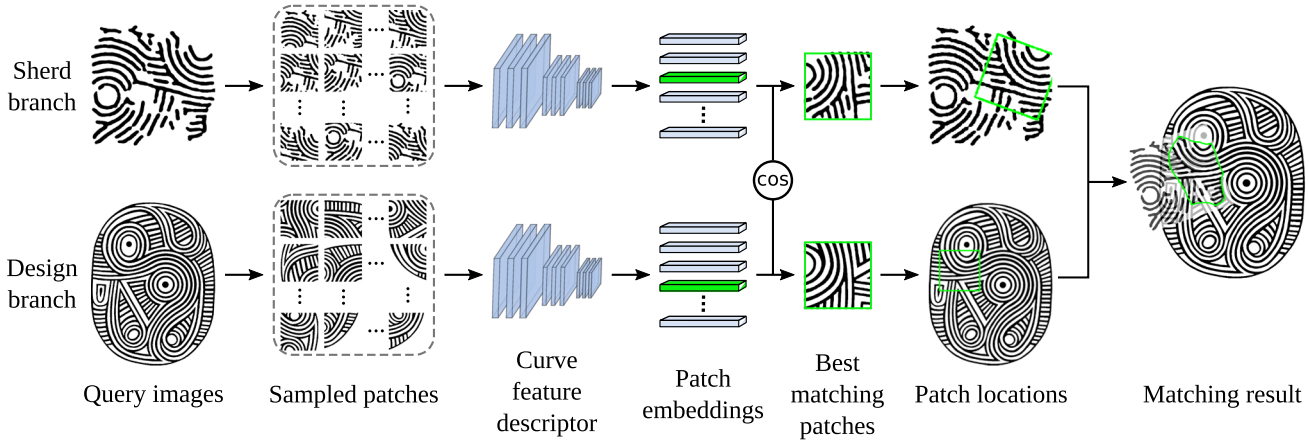

Fig. 7 An illustration of the proposed matching between sherd and design (Original designs reproduced with permission, courtesy of Frankie Snow, South Georgia State College)

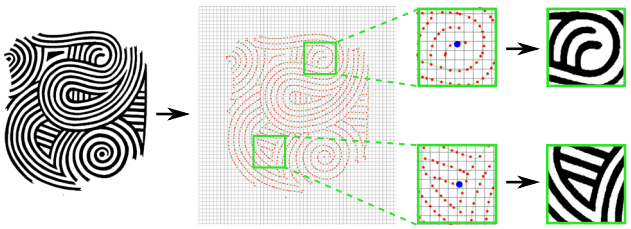

Fig. 8 Illustration of our patch sampling strategy. We first divide the image into small grids. In each grid, we find the skeleton pixel (red points) that is closest to the grid center. Around each selected skeleton pixel (blue points), we crop a patch of size $p \times p$ on the curve-pattern image. (Original designs reproduced with permission, courtesy of Frankie Snow, South Georgia State College) (Color figure online)

Finally, considering the rotation transform, we augment each patch constructed for the query sherd by rotating it 36 times by $0°, 10°, \cdots, 350°$, respectively.

The patch size p is an important parameter that needs to be pre-set. The smaller the value of p is, the more patches we sample and compare, leading to higher computational time. In addition, overly small patches may lack discriminative curve pattern features for design identification. On the contrary, an overly large patch may still cover composite patterns, which prevents the correct matching between a sherd and its underlying design. In our experiment, we empirically set $p = 300$ pixels such that each patch covers a $3cm \times 3cm$ region, on either the surface of query sherd or the design.

4.2.2 Step 2: Matching Score Estimation by Learning

After Step 1, both the query sherd and the consider design are represented by a set of sampled and augmented patches, each of which is a binary image of curve patterns. We match every pair of patches sampled from the sherd and the design, respectively, and evaluate their similarity by a matching score. The proposed patch matching score shall be robust to not only the noise and deformation present in the images of sherd, but also a certain level of displacement, i.e., translation and rotation, because our patch sampling in Step 1 could not cover all possible locations and rotations. For this purpose, we train a CNN-based curve feature descriptor to compute pairwise patch matching scores via deep embedding learning.

Table 2 Network architecture of the curve feature descriptor, where n, k, st, pd are the number of output channels, kernel size, stride and padding size, respectively. ReLu and BatchNorm layers are not included

Type	Configuration
Convolution	n:128, k:8 × 8, st:1, pd:0
Dropout	ratio:0.5
Convolution	n:128, k:3 × 3, st:1, pd:1
Convolution	n:128, k:3 × 3, st:2, pd:1
Convolution	n:64, k:3 × 3, st:2, pd:1
Convolution	n:64, k:3 × 3, st:2, pd:1
Convolution	n:32, k:3 × 3, st:2, pd:1
Convolution	n:32, k:3 × 3, st:1, pd:1
Input	300 × 300 curve-pattern image

The network architecture of the curve feature descriptor is illustrated in Table 2. In training, we optimize it in the form of triplet network (Hoffer & Ailon, 2015), which consists of three branches with shared weights and the three inputs are an anchor sherd patch I_S, a positive design patch I_D^+ and a negative design patch I_D^-, respectively. The positive I_D^+ refers to the ground-truth patch on the design D that matches the sherd patch I_S and the negative I_D^- can be any design patch that does not match I_S. The three branches of the network then produce three deep feature vectors F_S, F_D^+ and F_D^-, respectively. We propose the following triplet loss for network training:

$$L_{tri} = \max\left(\left\|F_S - F_D^+\right\|_2 - \left\|F_S - F_D^-\right\|_2 + \alpha, 0\right), \quad (5)$$

where α is a constant margin value. This loss function aims at minimizing the distance between F_S and F_D^+, and maximizing the distance between F_S and F_D^-.

The selections of I_D^+ and I_D^- are crucial to the training of triplet network. If we only use easy-to-discriminate I_D^+ and I_D^- in training, the triplet loss is likely to be 0 and the trained network may not be able to distinguish hard-to-discriminate cases in testing. To address this issue, we adopt two strategies to construct the training samples I_D^+ and I_D^-. First, to tolerate certain-level of displacement in matching, we spatially transform the window of each positive patch I_D^+ and use it to crop the design I_D for more augmented positive design patches. More specifically, we construct 81 positive design patches by taking the combination of rotation of $-10°$, $0°$, or $10°$, scaling of factor 0.9, 1, or 1.1, horizontal translation of -10, 0, or 10 pixels, and vertical translation of -10, 0, or 10 pixels. Second, for each anchor sherd patch I_S, we use template matching (Brunelli, 2009) to search from the patches sampled from all the known designs, except for the ground-truth design D, a set of 81 patches with the highest template-matching scores. We take them as hard negative design patches I_D^- for enhancing the triplet network training. Using these two strategies, we also achieve a data balance between positive and negative samples. In testing, for each sherd patch I_S and each design patch I_D, we can use the trained network to compute their deep feature vectors and calculate their cosine similarity as their matching score:

$$\phi(I_S, I_D) = \frac{F_S \cdot F_D}{\|F_S\|_2 \, \|F_D\|_2}, \quad (6)$$

where F_S and F_D are the extracted curve feature vectors of I_S and I_D, respectively.

4.2.3 Step 3: Matched-Region Identification

To help the visual examination and confirmation by archaeologists, following the results in Step 2 we further identify the best the matched region between the query sherd S and the design D. Let I_S^* and I_D^* be the best matched patches between S and D. By corresponding the coordinate of the four corners of these two patches, we can estimate an isometric transformation T from sherd S to design D. We then overlay the transformed sherd image $T(S)$ (the binary curve-pattern segmentation) to the design D for identifying their maximal matched region by following a region-growing strategy.

More specifically, at each location, we can crop a $p \times p$ patch on both $T(S)$ and D and then apply the algorithm in Step 2 to estimate their matching score. This way, we can obtain a heat map for the area shared by $T(S)$ and D, with matching scores as heat values. We then take the location with the largest heat value as seed of the matched region and check its neighboring locations – if their matching scores at a neighboring location is no less than 90% of the matching score at the seed location, they are included into the matched region. We then take each of the newly included locations as seed and repeat the above operation, until no more locations can be added.

Denote the matched region between $T(S)$ and D by R. We finally take the average heat values in R as the matching score $\phi(S, D)$ between S and D:

$$\phi(S, D) = \frac{1}{|R|} \sum_{I_S, I_D \in R} \phi(I_S, I_D). \quad (7)$$

Given a set of M query sherds and a set of N known designs, we can compute this matching score between each pair of them and construct a $M \times N$ similarity matrix. In Snowvision, for each query sherd S, we only return a small set of designs D with $\phi(S, D) \geq t$, with t being a pre-set threshold, together with their best matched regions, to archaeologists for confirmation, i.e., examining whether any of the provided designs is the true one stamped on the query sherd.

If we have $\phi(S, D) < t$ for all known designs D, we mark the query S as a sherd with unknown design, which means it may lead to new designs that have never been discovered. For such sherds, we cluster them based on their underlying curve patterns to facilitate the reconstruction of new designs, which will be studied in Task III.

4.3 Task III: Sherd Identification by Clustering

In this task, we take a group of sherds, actually their segmented binary curve-pattern images, without known designs and perform clustering, expecting that each cluster only contains sherds with the same underlying design. The goal is to provide clustering results to archaeologists for facilitating the discovery and reconstruction of new designs.

Sherd clustering differs from conventional image clustering in two aspects. On one hand, two sherds should be assigned to the same cluster as long as part of them share the same curve pattern, and their remaining parts can be different. On the other hand, we do not require every pair of sherds in each cluster to have the same curve pattern, but allow them to be linked through other sherds. For example, if sherd A and sherd B do not show any partial overlap, but both of them can be matched to sherd C with certain overlap, then A and B are still considered to have the same underlying design.

To address this problem, we formulate sherd clustering as a graph partitioning process. The key idea is to take each input sherd as a node of a fully connected graph, and then partition the graph based on the linkage between each pair of sherds. In general, our method can be divided into three steps: (1) constructing a sherd graph weighted by sherd-to-sherd matching

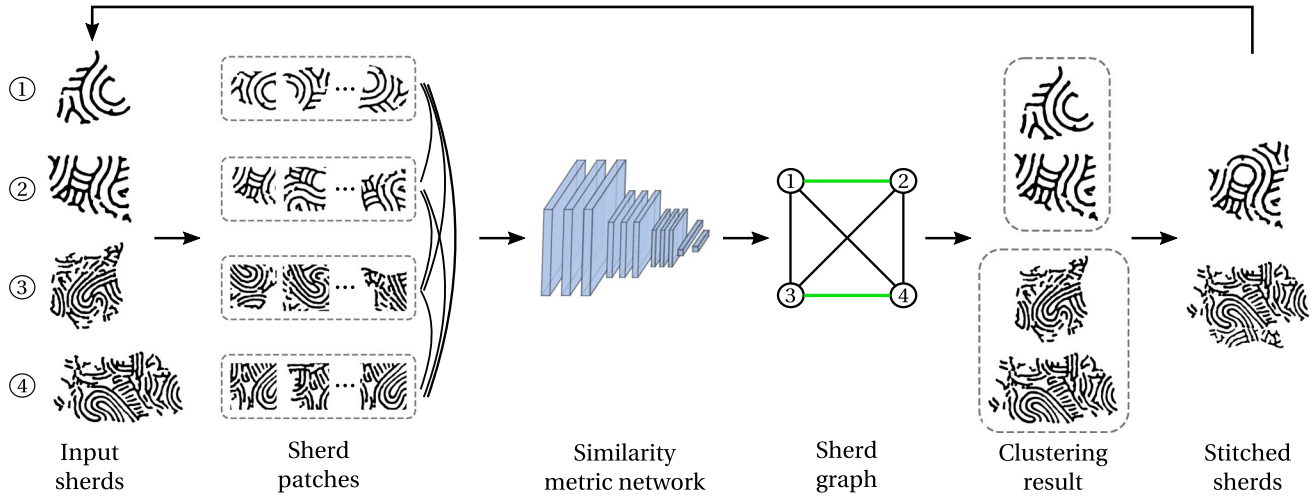

Fig. 9 The pipeline of the proposed sherd clustering method. Here we use an input of 4 sherds as an example

scores, (2) partitioning the graph into subgraphs/clusters by adaptive thresholding, and (3) sherd stitching for iterative clustering. An illustration using the proposed method to cluster four input sherds is shown in Fig. 9.

4.3.1 Step 1: Constructing Sherd Graph

Given a set of N input sherds, in the form of their segmented binary curve pattern images, $\mathbf{S} = \{S_1, S_2, \cdots, S_N\}$, we first construct a fully-connected undirected graph $G = \langle \mathbf{S}, W \rangle$, where W is the edge weight matrix that measures the linkage of each pair of sherds. For each pair of sherds, their edge weight is determined by the similarity of the best matched parts between them. To find out their best matched parts, we follow the patch sampling strategy proposed in Sect. 4.2.1 to sample a set of patches of size $p \times p$ from each sherd.

The cosine similarity of the extracted features was employed in Task II as the similarity metric to rank design patches. However, this metric is not sufficiently discriminative for Task III because the cosine similarities of positive and negative patch pairs can be very close, which is not desired in threshold-based graph partitioning. Therefore, we train a curve similarity metric network $\mathcal{M}$ to directly predict the similarity of two sherd patches. Specifically, on top of the curve feature descriptor described in Table 2, we append a global average pooling (GAP) layer, two fully-connected (FC) layers, and a softmax layer. The output channels of two FC layers are 128 and 2, respectively. The training of $\mathcal{M}$ takes a pair of sherd patches and a 0/1 label that indicates whether they are matched or not as input. We minimize the cross-entropy loss to optimize the parameters of two FC layers, while the curve feature descriptor is fixed. In testing, the edge weight $W_{i,j}$ of two sherds S_i and S_j is the highest similarity among all patch pairs sampled in these two sherds:

$$W_{i,j} = \max_{I_{S_i} \in S_i} \max_{I_{S_j} \in S_j} \mathcal{M}(I_{S_i}, I_{S_j}). \tag{8}$$

Algorithm 2 Sherd graph partitioning by adaptive thresholding

Input: Sherd set $\mathbf{S}$; Edge weight matrix W; Maximum cluster size t_c
Output: Sherd clusters $\mathbf{C}$

Initialize the weight threshold $t_w = \min(W)$, the queue of undivided clusters $\mathbf{C}' = \{\mathbf{S}\}$.
while $\mathbf{C}'$ is not empty **do**
  Let C be the first element of $\mathbf{C}'$ and pop out it
  Update the weight threshold by $t_w \leftarrow t_w + 0.1 * (1 - t_w)$
  Partition C to a set of clusters $\hat{\mathbf{C}}$ using the current weight threshold t_w
  for each cluster $C_i \in \hat{\mathbf{C}}$ **do**
    if $|C_i| \leq t_c$ **then**
      Append C_i to $\mathbf{C}$
    else
      Append C_i to $\mathbf{C}'$
    end if
  end for
end while

4.3.2 Step 2: Sherd Graph Partitioning

After obtaining the weight matrix W, we partition the fully-connected graph G into sherd clusters. Intuitively, we can cut all edges below a pre-set similarity threshold and take the remaining subgraphs as clustering results. However, such clustering result is very sensitive to the pre-set threshold. To solve this problem, we adopt an adaptive thresholding strategy (Zhan et al., 2018) to partition the graph, as shown in Algorithm 2.

We first set the weight threshold to the lowest value of W, and maintain a cluster queue initialized by $\mathbf{S}$. Next, we iteratively process each cluster in the queue in the order of

Breadth First Search (BFS). The main idea is to adjust the weight threshold to limit the size of output clusters to a size threshold of t_c. In each iteration, we gradually increase the weight threshold, and partition the current cluster into sub-clusters. The cluster partitioning is implemented by cutting low-weight edges and grouping remaining connected sherds by Union-Find. This process repeats until the cluster queue is empty.

4.3.3 Step 3: Sherd Stitching for Iterative Clustering

We finally propose a strategy to iteratively refine the clustering result by introducing the stitched curve patterns. As a side benefit of the patch-based matching in Step 1, we are able to find the transformation between the two sherds, by aligning their best matched patches. This way, we can transform the curve patterns of all sherds in the same cluster to a common coordinate system and stitch them together to obtain a larger curve-pattern image, as shown in Fig. 9, by following a similar process described in Sect. 4.2.3. Specifically, in each cluster produced by the previous step, we stitch sherd pairs in the order of their edge weights – from high to low. When stitching two sherds, we overlay the larger one on the top of the smaller one. Compared to curve patterns on individual sherds, the stitched curve patterns bear more geometry information of the underlying design, and we can put them back as new graph nodes for clustering and they can help recall sherds that were not clustered or not correctly clustered in the initial clustering. This iterative clustering can be repeated until there is no more change to the clustering result.

5 Experiments

In this section, we conduct experiments to evaluate the performance of the proposed methods on three tasks. For each task, we first compare our method with representative existing methods, and then conduct in-depth ablation study on the proposed methods.

5.1 Datasets

For this study, we use the Snowvision digitization pipeline in Sect. 3.1 to digitize 1604 pottery sherds from various archaeological sites located in southeastern North America. These 1604 sherds correspond to 133 unique paddle designs. Each sherd has a curve pattern, either single or composite, from only one design. Their ground-truth designs are manually identified by archaeologists.

For Task I, we manually labeled the ground-truth curve pattern mask for a subset of 621 sherds, in which 300 sherds are randomly selected for training, and the rest 321 sherds are used for evaluation. For Task II, we use all 1604 sherds and 133 designs in the design identification experiment. We randomly split all sherds into 604 training sherds and 1000 test sherds. To generate meaningful ground-truth clusters for Task III, we select 1091 sherds with sufficient curve pixels (in the binary curve pattern images) from the whole dataset. These 1091 sherds come from 129 different designs, corresponding to 129 ground-truth clusters. Subsequently, all 129 clusters are randomly split into 60 training clusters and 69 test clusters. The 60 training clusters contain 510 sherds, and the 69 test clusters contain 581 sherds.

Note that for all the sherd data used for the experiments of Tasks II and III, we first apply the network model trained in Task I for curve-pattern segmentation. We then use these automatically segmented, imperfect curve-pattern images for matching and clustering in Tasks II and III, respectively. This way, the resulting performance can well reflect the practical usefulness of the Snowvision system, for which we provide more discussions in later Sect. 5.5.

5.2 Task I Experiments on Curve-Pattern Segmentation

As mentioned above, we select 621 depth images of sherds with manually labeled ground-truth curve-pattern segmentation for Task I. We randomly divide them into 300 training images and 321 testing images. To train the FCN in Step 1, we thin all the ground-truth curve patterns to one-pixel width skeletons, using a standard image thinning algorithm (Lam et al., 1992). Data augmentation is employed here to generate sufficient training data. Specifically, we first split the whole image into small blocks with a size of 100×100. Then these blocks are rotated, scaled and flipped with the same scheme as in Shen et al. (2016). Finally, 141,696 blocks are used in FCN training in Step 1. For Step 2, we randomly take 44,906 window images with a size of 45×45 around the skeleton pixels identified in Step 1 for network training.

For the purpose of better training, the parameters of encoders in the skeleton extraction network are initialized with the FCN-8 s model (Long et al., 2015) pretrained on PASCAL VOC (Everingham et al., 2010). We then train it using the SGD optimizer. The maximum number of training iterations is set to 20,000, with a batch size of 10. The base learning rate is 1×10^{-7} and decays to 1×10^{-8} after 10,000 iterations. Momentum and weight decay are set to 0.9 and 5×10^{-4} respectively.

Considering the dense prediction ConvNet in Step 2 is relatively lightweight, we choose to train it from scratch. It is also trained using the SGD optimizer. The maximum number of training iterations is set to 100,000, with a batch size of 10. The base learning rate is 1×10^{-3}, and it decays exponentially in training with the parameters of $\gamma = 10^{-3}$ and $power = 0.75$. Momentum and weight decay are set to be the same as the FCN in Step 1.

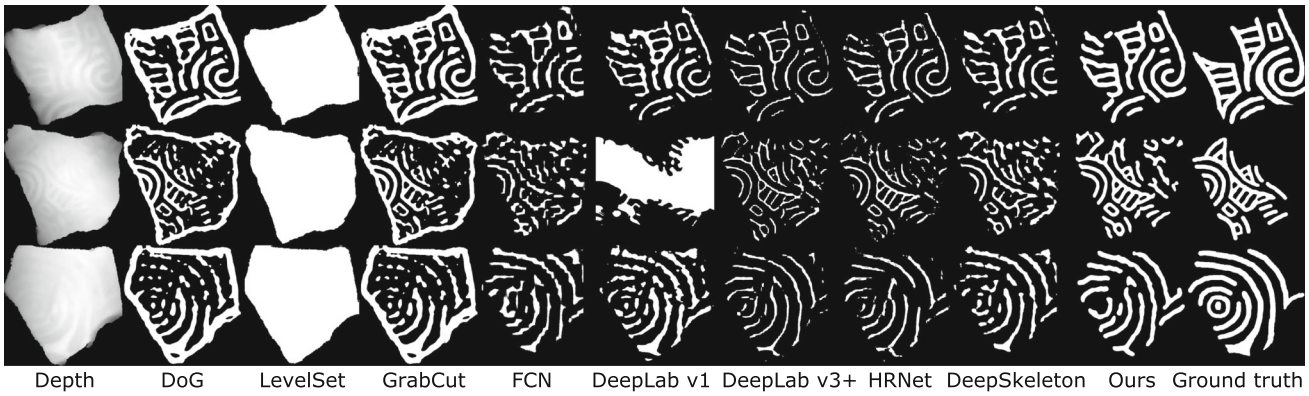

Fig. 10 Examples of the curve-pattern segmentation result from the proposed method and all comparison methods

5.2.1 Comparison with Existing Methods

To justify the effectiveness of our method of curve-pattern segmentation, we select eight widely-used segmentation methods for comparison – Difference of Gaussian (DoG), Level Set (Vese & Chan, 2002), GrabCut (Rother et al., 2004), FCN (Long et al., 2015), DeepSkeleton (Shen et al., 2016), DeepLab v1 (Chen et al., 2016), DeepLab v3+ (Chen et al., 2018) and HRNet (W. et al., 2020). The experiment is conducted on the 321 testing images as described above, and the evaluation criteria is the traditional F-measure of $\frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$.

For fair comparison, we tune the parameters of existing methods on the same training set as ours, and then evaluate on the test set. The detailed settings are introduced as follows. (1) For DoG, we manually tune the kernel size k and the standard deviation σ of two Gaussian filters on 300 training images, and find the best parameter setting to be: $k_1 = k_2 = 45$, $\sigma_1 = 11$, $\sigma_2 = 5$. (2) For LevelSet, we adopt the implementation proposed in Li et al. (2010). We tune two main parameters in the energy function, the weight λ of the length term and the weight α of the area term, and finally choose $\lambda = 5$ and $\alpha = 3$. Other parameters use their default values and the initial contour uses the square boundary of image. (3) For GrabCut, we adopt the implementation in OpenCV. It requires a foreground mask as initialization, and we use the result of DoG, which can produce reasonable curve masks. The number of iterations is set to be 10. (4) For all learning-based methods FCN, DeepLab v1, DeepLab v3+, HRNet and DeepSkeleton, we use their models pretrained on PASCAL VOC as initialization, and then fine-tune on our training sets with respect to their respective hyperparameters. We randomly sample $\frac{1}{10}$ training images as the validation set to select the best model for each of them. Since the input of these models is a 3-channel image, we simply duplicate the input depth image to 3 channels and then normalize it using our mean and standard deviation values. Besides, since DeepSkeleton requires the ground-truth scale values of skeleton pixels as input, we obtain it by applying the distance transformation on ground-truth segmentation masks.

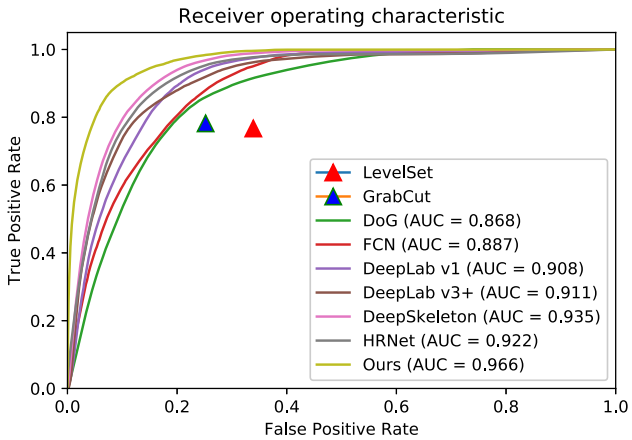

Fig. 11 ROC curves and AUC values of all the methods in Task I

The performance of all methods, averaged over all 321 testing images, are reported in Table 3. We can see that the proposed method achieves the best F-measure, and outperforms the second best result (DeepSkeleton) by 9.2%. Figure 10 shows the segmentation results on three sample images, using all methods. In these images, we can observe that DoG actually enhances the difference between adjacent pixels. As a purely low-level method, it may not capture deep and shallow curves simultaneously. GrabCut was initialized by DoG, but its performance becomes even worse. One major reason might be that the data and smoothness energy defined in GrabCut are not sufficiently discriminative to separate the curve and non-curve regions in such a low-contrast image. This is probably the same reason that makes Level Set fail. As expected, the CNN-based comparison methods normally achieve better performances than the low-level methods. However, their segmentation results usually contain many false positives and the boundaries of the segmented curve patterns are quite rough. Moreover, they require the full curve mask for training, while our method needs only the skeleton

Table 3 Precision, recall and F-measure of all the methods on Task I

Method	Precision	Recall	F-measure
DoG (Bundy & Wallen, 1984)	0.376	0.778	0.498
LevelSet (Li et al., 2010)	0.265	**0.939**	0.403
GrabCut (Rother et al., 2004)	0.362	0.671	0.450
FCN (Long et al., 2015)	0.593	0.494	0.527
DeepLab v1 (Chen et al., 2016)	0.594	0.666	0.590
DeepLab v3+ (Chen et al., 2018)	0.622	0.638	0.627
HRNet (W. et al., 2020)	0.629	0.644	0.632
DeepSkeleton (Shen et al., 2016)	0.637	0.698	0.661
Ours	**0.676**	0.850	**0.753**

Bold font indicates the best performance

labels. Although the proposed method does not achieve the first place in both precision and recall, it achieves the best performance in the final F-measure.

For more comprehensive evaluation, we provide the Receiver Operating Characteristic (ROC) curves and the corresponding Area under Curve (AUC) values of all the methods in Fig. 11. We obtain the ROC curve of our method by changing the threshold of selecting skeleton pixels in Step 2 and keeping the Step 3 unchanged. The ROC curves of LevelSet and GrabCut are absent, because they directly output binary masks instead of probability maps. Both the ROC curves and AUC values indicate the superiority of our method in curve-pattern segmentation.

Fig. 12 Sample segmentation results of the proposed method with modifications to each step. **a** Input depth image. **b** Segmentation result after modifying Step 1. **c** Segmentation result after removing Step 2. **d** Segmentation result after modifying Step 3. **e** Segmentation result of the proposed method without any modification. **f** Ground-truth segmentation

5.2.2 Ablation Study

Intuitively, the three steps of our segmentation method can be replaced by other alternatives or simply ignored. To justify the usefulness of each step, we design three additional experiments, in each of which, we modify or remove one step of the proposed method, and then check its influence to the final segmentation performance.

Modifying Step 1: Step 1 of the proposed method is skeleton extraction. Actually, the FCN we used in this step can be trained to produce curve-pattern segmentation directly. However, we choose to extract skeletons first, and then take additional steps to recover the curve width. In this experiment, we make several adjustments in the FCN in Step 1 to let it output curve-pattern segmentation with width directly. For this purpose, we directly use the ground-truth segmentation as supervision, and remove the scale estimation and image thinning operations. Sample results of this modified method are shown in Fig. 12(b). We can see that these results contain more false positives and rougher segmentation boundaries. Quantitatively, F-measure of the proposed method decreases from 0.753 to 0.681 if we make this modification to Step 1.

Using different encoders in Step 1: In Step 1, three lightweight basic ConvNets are employed as our encoders. In this experiment, we change them to residual blocks to see if our method can benefit from more sophisticated encoders. Specifically, we use the ResNet-50 (He et al., 2016) as the backbone, and three decoders are appended to the layer 2, 3 and 4 of it, respectively. Thus three encoders corresponding to layer1+layer2, layer3 and layer4 of ResNet-50, respectively. With the ResNet-50 model pretrained on ImageNet (Russakovsky et al., 2015) as initialization, we fine-tune it on our training set using the same hyperparameters. The precision, recall and F-measure of the new encoders are 0.672, 0.847 and 0.751, respectively. In natural images, ResNet backbones usually show better feature extraction capability than the regular ConvNets (Minaee et al., 2021). However, in the proposed curve-pattern segmentation, we find that the two kinds of encoders have similar performances in terms of F-measure. We speculate that there are two possible reasons: (1) The ResNet-50 backbone has much more trainable parameters than the original FCN (23 M *vs* 7.6 M), while we

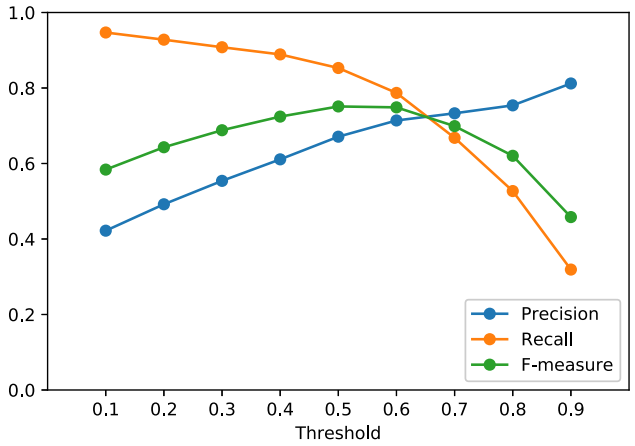

Fig. 13 The performance of the proposed curve-pattern segmentation method by using different thresholds in Eq.(4)

have only 300 training images. The larger ResNet-50 model may be overfitted on the small training set; (2) Our depth images have much less variations than natural images and thus may not require complicated feature descriptors.

Removing Step 2: Step 2 of the proposed method employs a dense prediction ConvNet as a pixel-wise classifier to refine skeletons extracted by FCN in Step 1. To justify its usefulness, we remove this step and recover curve width directly from the skeletons generated in Step 1. Sample results are shown in Fig. 12(c). We can see that the removal of Step 2 leads to more false positives. Quantitatively, F-measure of the proposed method decreases from 0.753 to 0.676 if we remove Step 2. The reason that the ConvNet could better identify skeleton pixels than FCN is because it focuses on local patch features instead of the whole image features, and is more suitable for skeleton extraction. But it would be too slow if we use ConvNet to classify every pixel, thus we only use it to positive pixels selected by FCN.

Changing the threshold in Step 2: In Eq.(4) of Step 2, we use a threshold of 0.5 to select skeleton pixels. In this experiment, we test the sensitivity of our method to this threshold by changing it from 0.1 to 0.9. The precision, recall and F-measure of our method with different thresholds are illustrated in Fig. 13. We can see that the F-measure is reasonably stable when the threshold varies from 0.4 to 0.6. It indicates that our curve recovering method in Step 3 is able to tolerate minor changes in the output of Step 2.

Modifying Step 3: Simple morphological dilation seems to be a very intuitive approach to recover curve width in Step 3. In this experiment, we modify Step 3 by replacing it with a dilation operation with a radius of 15 pixels, which is the best parameter after we try and test all different values. Sample results are shown in Fig. 12(d). While the dilation produces very smooth curve patterns, they are not well aligned with the ground truth. Quantitatively, F-measure of the proposed method decreases by 3.5% if we make this modification to Step 3.

5.3 Task II Experiments on Design Identification

As mentioned above, we take all 1604 sherds for Task II experiments—604 for training and 1000 for testing. We sample every sherd and design image in the training set with a patch size set to 300×300 pixels. In total, 600k triplets were sampled from the training set in the form of image patches. The proposed curve feature descriptor is trained by Adam optimizer with a base learning rate of 10^{-3} and a momentum of 0.9. We train it for 50k iterations with a mini-batch size of 64. The margin value α in the triplet loss is set to 1. The sherd and design images in testing were sampled with the same patch size as in training.

In this task, we use the Cumulative Matching Characteristics (CMC) ranking metric to evaluate the design identification performance. In testing, for each sherd, we match it against all 133 designs and obtain 133 matching scores. We sort these 133 matching scores and pick the top L designs with the highest scores. If the ground-truth design of the sherd is among the identified top L designs, we treat it as a correct design identification under rank L. We repeat this identification for all 1,000 test sherds and calculate the accuracy, i.e., the percentage of the sherds with correctly identified designs, under each rank $L = 1, 2, \cdots, 133$. This way, we can obtain a CMC curve in terms of rank L, as shown in Fig. 14, to evaluate the performance of the overall matching result.

5.3.1 Comparison with Existing Methods

To demonstrate the effectiveness of the proposed method on design identification, we compare with nine representative existing image matching methods, including three low-level matching methods template matching (Brunelli, 2009), Chamfer matching (Barrow et al., 1977), and Shape Context (Belongie et al., 2001); one fingerprint matching method (Tico & Kuosmanen, 2003); three handcrafted-feature-based methods SIFT (Lowe, 1999), SURF (Bay et al., 2006) and ORB (Rublee et al., 2011); and two learning-based methods LF-Net (Ono et al., 2018) and RF-Net (Shen et al., 2019).

The detailed settings of comparison methods are as follows: (1) For template matching, we take sherd images as the source and design images as the target. Besides searching along x, y directions, we also search for the angle of rotation from 0° to 359°. The strides along these three dimensions are set as 10. The sum-of-square-difference is employed as the matching cost. (2) For Chamfer matching, we perform image thinning on all sherd and design curve-pattern images to get one-pixel-wide skeleton images, and compute the min-

imum Chamfer matching cost using the distance transform between each pair of sherd and design. The searching strategy is similar to the one used for template matching. (3) For fingerprint matching, we adopt the minutia-feature-based algorithm implemented in Alilou (2020) and treat the sherd images as query fingerprints and the design images as the fingerprint database. We perform image thinning to all sherd and design curves to make them 3-pixel-wide to reduce redundant minutia on curve ridges. Euclidean distance is computed as the matching cost by every minutia feature pair. By setting a threshold cost of 15 on each feature pair, we compute the number of matched features normalized over the number of features as the matching score of each sherd-design pair. (4) For Shape Context, we thin the curve structures of the sherd and design to one-pixel wide. To deal with partial matching, we use a sliding window technique to match a sherd curve pattern to each window-cropped design curve pattern and then choose the one with the lowest matching distance. The sliding-window size is the same as the sherd image. The shape context algorithm implementation directly comes from the OpenCV package. The number of sampled points per sherd is set to be 100. (5) For SIFT, SURF and ORB, we follow the same identification method by taking the best matched design in a design database for a sherd. Specifically, we use their implementations in OpenCV. Keypoints are matched using Brute-Force Matcher for SIFT and SURF, and FLANN-based Matcher for ORB. Every feature in a sherd is compared to all the features in a design, and we compute the L2-norm distance of the feature pair for SIFT and SURF, and normalized Hamming distance for ORB. Then the matching score is computed as the number of best matched keypoint pairs for a sherd-design pair. In searching for the best matched keypoint pairs, the ratio of the best matched keypoint pair and their second best matched keypoint pair is set to 0.75. (6) Both LF-Net and RF-Net are learning-based methods. We use their released pretrained models as initialization and fine-tune them on the training set of Task II. LF-Net is pretrained on the indoor dataset ScanNet (Dai et al., 2017), while RF-Net is pretrained on HPatches (Balntas et al., 2017). Both contain two networks, a detector and a descriptor. Training the detector requires the ground-truth camera pose and the calibration matrix, which are not available in our datasets. Therefore, we fix the pretrained detectors, and fine-tune the descriptors using their default hyperparameters but reduce the learning rate by 10 times. In their source code, they convert RGB images to grayscale images as input, so we do not need to change the number of channels of our images.

CMC curves of the proposed method and all comparison methods are shown in Fig. 14. The detailed CMC Rank-1/5/10/20 values are reported in Table. 4. From Fig. 14 and Table. 4, we can see that: (1) The proposed method achieves the best performance among all, and outperforms the second best method by 13.7% in terms of CMC Rank-1 accuracy. (2) All comparison methods perform poorly in this task, and several of them even have nearly diagonal CMC curves, which is equivalent to random guessing. It demonstrates the challenge of the curve-pattern matching problem. (3) Template matching has the second best Rank-1 accuracy among all. After checking the matching results, we find that template matching works well on sherds with single patterns, but performs poorly on composite pattern matching; other low-level matching methods, such as Shape Context and Chamfer Matching, are sensitive to noises and perform poorly in both cases. (4) Keypoint-based matching methods including handcrafted-feature-based and learning-based methods do not produce satisfactory results, because they cannot find proper keypoints in our low-texture binary curve-pattern images. By explicitly considering the possible composite patterns, as well as noise, errors, and deformation in curve-patterns images, the proposed method achieves much better CMC performance than all other methods.

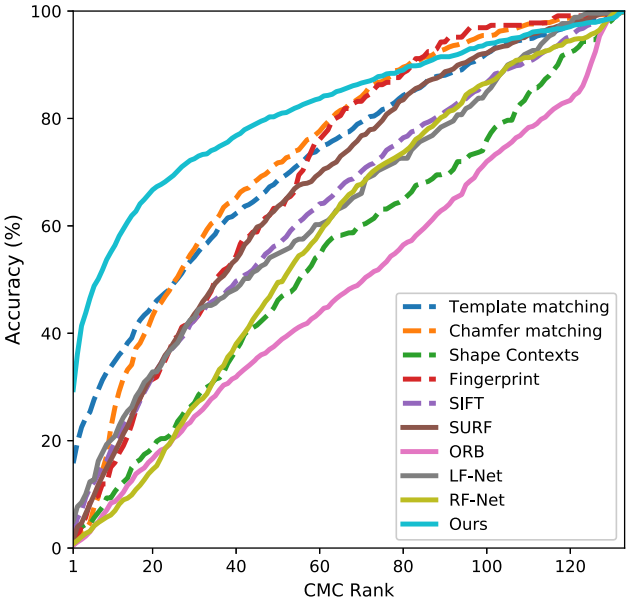

Fig. 14 CMC curves of all the methods in Task II

5.3.2 Ablation Study

Without patch sampling: Step 1 of the proposed method is patch sampling on both sherd and design. As sherds may contain composite patterns, the major purpose of Step 1 is to address this challenge by cropping a patch with only a single pattern that can be matched to a portion of its underlying design. Here we try to directly match whole sherds to whole designs without patch sampling. Specifically, for each pair of sherd and design, we directly use the trained curve feature descriptor to obtain their global image features, and their matching score is computed based on the global features. The CMC accuracy of our method with and without patch sam-

Table 4 CMC accuracy (%) of all the methods in Task II

Method	Rank-1	Rank-5	Rank-10	Rank-20
Template matching (Brunelli, 2009)	15.8	25.3	33.7	45.5
Chamfer matching (Barrow et al., 1977)	0.5	4.7	17.1	43.2
Shape Context (Belongie et al., 2001)	1.7	4.4	9.3	18.7
Fingerprint matching (Tico & Kuosmanen, 2003)	0.8	8.4	15.0	31.4
SIFT (Lowe, 1999)	2.9	11.2	18.2	31.9
SURF (Bay et al., 2006)	1.6	6.4	16.5	32.2
ORB (Rublee et al., 2011)	0.5	3.0	8.4	16.6
LF-Net (Ono et al., 2018)	8.5	16.0	22.6	33.8
RF-Net (Shen et al., 2019)	2.5	5.1	7.6	14.7
Ours	29.5	46.5	55.1	66.7

Table 5 CMC accuracy (%) of the proposed curve-pattern matching method with and without patch sampling

	Rank-1	Rank-5	Rank-10	Rank-20
w/o patch	15.8	25.3	33.7	45.5
w/ patch	29.5	46.5	55.1	66.7

pling are compared in Table 5. The performance degrades significantly after removing patch sampling. It indicates that curve-pattern matching is a partial-to-partial matching problem and cannot be solved using global features.

Changing the patch size: In Step 1, we use a fixed patch size $p = 300$ to sample patches from sherd and design images. In this experiment, we try different patch sizes to see its effect to the matching performance. Specifically, we vary the patch size from 200 to 400 with a stride of 20, and the detailed CMC accuracy is reported in Table 6. We can see that our method achieves the best performance at $p = 300$, and the matching performance is quite stable for $p \in [260, 340]$—CMC Rank-1 values vary from 27.9% to 29.5%. In practice, overly small patches could not carry sufficiently discriminative information for matching, while overly large patches may contain composite patterns or non-curve regions. Both cases may prevent the correct matching between a sherd and its underlying design.

Using different feature descriptors: Although the sherd and design patches are binary images with sparse texture, measuring their curve pattern similarity is non-trivial. To validate the discriminability of the curve feature descriptor trained in Step 2, we try to replace it with other commonly used similarity metrics/feature descriptors to evaluate curve pattern similarity. In specific, we compare with seven representative methods, including two pixel-level similarity metrics Sum-of-Square-Difference (SSD) and Normalized Cross Correlation (NCC); three handcrafted feature descriptors SIFT (Lowe, 1999), HOG (Dalal & Triggs, 2005), and LBP (Ojala et al., 2002); and two learning-based feature descriptors HardNet (Chao et al., 2019) and SKAR (Markuš et al., 2018). For SIFT, we adopt the implementation in OpenCV, but only compute the feature at the patch center with a neighborhood size of 300 to get the patch features.

Table 6 CMC accuracy (%) of the proposed curve-pattern matching method with different patch sizes

Patch size	Rank-1	Rank-5	Rank-10	Rank-20
200	11.7	29.9	40.3	55.1
220	18.5	37.4	48.6	61.6
240	25.2	42.9	55.3	66.6
260	27.9	44.1	54.0	67.3
280	29.1	45.8	54.8	**70.0**
300	**29.5**	**46.5**	55.1	66.7
320	28.6	46.4	**57.1**	66.2
340	28.4	45.1	53.5	65.3
360	27.1	43.3	51.9	63.2
380	25.4	41.1	49.2	59.3
400	24.2	39.1	47.5	56.5

Bold font indicates the best performance

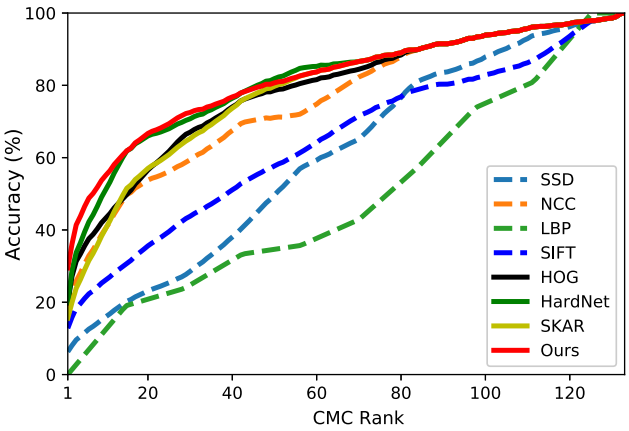

Fig. 15 CMC curves of using different feature descriptors in our curve-pattern matching method for Task II

Table 7 CMC accuracy (%) of the proposed curve-pattern matching method with different feature descriptors

Descriptor	Rank-1	Rank-5	Rank-10	Rank-20
SSD	6.3	11.5	16.0	23.1
NCC	17.4	30.6	40.5	53.7
LBP (Ojala et al., 2002)	0.0	5.5	12.4	20.8
SIFT (Lowe, 1999)	12.8	21.0	26.1	35.6
HOG (Dalal & Triggs, 2005)	19.8	35.5	43.2	56.5
HardNet (Chao et al., 2019)	20.0	39.4	51.1	66.0
SKAR (Markuš et al., 2018)	15.6	28.9	33.7	57.0
Ours	29.5	46.5	55.1	66.7

For HOG, we divide each patch into 8×8 cells and use 16 bins. For LBP, we adopt the implementation of scikit-image (Van der Walt et al., 2014). The radius of circle is 150, and the number of points is set to be 8 times of the radius. For HardNet and SKAR, we use their released models trained on HPatches (Balntas et al., 2017). The CMC curves of using all the methods are shown in Fig. 15, and the detailed CMC Rank-1/5/10/20 values are reported in Table 7.

From Fig. 15 and Table 7, we can see that our curve feature descriptor significantly outperforms other alternative methods, especially on the CMC Rank-1 accuracy. Pixel-level similarity metrics and handcrafted features are not robust to noises and the curve deformations between sherd and design images, so they have inferior performances on this task. Learning-based descriptors have better performance, but they are not specifically trained for curve-pattern matching. With the carefully designed training strategy, our curve feature descriptor is more sensitive to curve structure difference and more robust to minor translation and rotation perturbation.

Changing the similarity metric in Step 2: In Task II and Task III, we use two different metrics to measure the similarity of curve patterns – cosine similarity for the former and the metric network $\mathcal{M}$ for the latter. In this experiment, we replace the cosine similarity in Task II with the metric network $\mathcal{M}$. After changing the metric, the performance of our method drops slightly. The CMC Rank-1/5/10/20 values decrease from 29.5%, 46.5%, 55.1% and 66.7% to 28.9%, 46.0%, 54.7% and 66.6%, respectively. It indicates that, although simple, the cosine similarity is more suitable for Task II.

We use different metrics in Task II and Task III as they have different goals. In Task II, we match each sherd to different designs, and the similarity scores incurred by different sherds can be non-comparable. In Task III, the similarity scores incurred by different sherds must be comparable – they are embedded into a single fully-connected graph for clustering. Therefore, we simply use the cosine similarity for ranking in Task II, but train a more comprehensive metric network for thresholding in Task III.

5.4 Task III Experiments on Sherd Identification

As introduced in Sect. 5.1, a dataset of 1,091 sherds from 129 clusters (designs) is used in the experiment of Task III. The training set includes 510 sherds from 60 clusters. To train the similarity metric network, we sample 25k positive patch pairs of size 300×300 from sherds with overlapped regions in the same cluster. We also sample 25k negative patch pairs by performing template matching on sherds randomly selected from different clusters to mine hard negative pairs. The similarity metric network is trained by Adam optimizer for 20k iterations with a base learning rate of 10^{-3}, a momentum of 0.9, and a batch size of 32. The threshold on cluster size t_c is set to 20.

In testing, 591 sherds from 69 clusters are used to evaluate the performance of the proposed sherd clustering method and existing image clustering methods. Three commonly used metrics are employed to evaluate the clustering results: (1) Purity: The fraction of samples that belong to the same cluster; (2) Adjusted Rand Index (ARI): The fraction of sample pairs that are correctly assigned in the same or different clusters; and (3) Normalized Mutual Information (NMI): The mutual information between the ground-truth clusters and the predicted clusters normalized by the sum of their entropy. For all three metrics, higher value indicates better performance.

5.4.1 Comparison with Existing Methods

We compare our method with eight representative existing clustering methods, including six traditional methods K-Means (Lloyd, 1982), K-Means++ (Arthur & Vassilvitskii, 2006), Spectral Clustering (Shi & Malik, 2000), Mean-Shift (Comaniciu & Meer, 2002), Agglomerative Hierarchical Clustering (AHC) (Jain & Dubes, 1988), Affinity Propagation (Frey Dueck, 2007); and two deep-learning-based methods Approximate Rank-Order Clustering (ARO) (Otto et al., 2017) and GCN Clustering (Wang et al., 2019). For the comparison methods in Task III, we use the curve-pattern descriptor trained in Task II to extract a 128-dimension vector for each sherd and then perform clustering of these feature

Table 8 Purity, ARI and NMI of all the methods in Task III

Method	Purity	ARI	NMI
K-Means (Lloyd, 1982)	0.313	0.016	0.174
K-Means++ (Arthur & Vassilvitskii, 2006)	0.344	0.017	0.196
Spectral Clustering (Shi & Malik, 2000)	0.23	0.013	0.143
MeanShift (Comaniciu & Meer, 2002)	0.165	0.001	0.041
AHC (Jain & Dubes, 1988)	0.27	0.028	0.18
Affinity Propagation (Frey Dueck, 2007)	0.3	0.011	0.158
ARO (Otto et al., 2017)	0.26	0.021	0.108
GCN Clustering (Wang et al., 2019)	0.353	0.017	0.198
Ours	1	0.148	0.644

vectors. We tune their parameters and settings on the training set to maximize the Normalized Mutual Information (NMI). Except for ARO and GCN clustering, all other methods adopt the implementations in scikit-learn.

The detailed experimental settings of comparison methods are as follows: (1) For K-Means and K-Means++, we provide the ground-truth number of clusters in the test images as a favor for them, and set the maximum number of iterations to be 300. (2) For spectral clustering, we adopt the discretized label assignment strategy, and construct the affinity matrix by computing a graph of nearest neighbors. The ground-truth number of clusters is also provided as a favor. (3) For MeanShift, the bandwidth used in the RBF kernel is estimated by a built-in function with a quantile of 0.3, and the maximum number of iterations is set to be 300. (4) For AHC, we construct a connectivity matrix for initialization using kNN, and set the linkage criterion to minimize the variance of the clusters being merged. The linkage distance threshold is set to be 0.7. (5) For affinity propagation, the damping factor is set to be 0.5 and the number of iterations is set to be 200. (6) For ARO, we adopt the implementation in Zhan et al. (2018). It directly takes the feature vectors as input, and does not require fine-tuning. We set the number of considered nearest neighbors to be 30, and the clustering distance threshold as 2. (7) For GCN clustering, we use our own feature vectors and train the clustering model from scratch using the default hyperparameters, except that the numbers of 1-hop and 2-hop nodes are changed to 20 and 5, respectively.

From Table 8, we can see that our proposed clustering method achieves the best performance on all three metrics. All the comparison methods produce poor performance on this task, even if the ground-truth cluster number is given to them. This demonstrates both the challenge of this task and the effectiveness of the proposed method. We hypothesize that the main cause of their failure is that they only leverage the global image features of the sherds (curve pattern images) in clustering, but cannot handle the special partial-to-partial matching in this particular problem. We also notice that our method achieves 100% purity while producing many singleton clusters with only one sherd. This is because we choose high precision in the trade-off between precision and recall. Archaeologists prefer to work on small and pure clusters, rather than large clusters with sherds from many different designs. In practice, we return only clusters with more than 2 sherds for design reconstruction.

We further examine our clustering results by mapping each ground-truth cluster to a predicted cluster using the Hungarian algorithm. After removing predicted clusters with only one sherd, we find 61 predicted clusters correspond to 41 unique designs. By comparing each ground-truth cluster with the corresponding predicted cluster, we can count the number of found sherds (true positive), missed sherds (false negative), and wrong sherds (false positive) for each design. Averaged over all 41 designs, our method can find 41.3% sherds, miss 58.7% sherds, and introduce 0 wrong sherd. In general, our method works better on sherds with adequate and clear curve patterns. Although many sherds are missed, we return the most informative sherds to archaeologists to reconstruct the full design. In Fig. 16, we display the sample clustering results of three designs.

5.4.2 Ablation Study

Using different similarity metrics: In the first step of our method, we train a curve similarity metric network $\mathcal{M}$ to obtain edge weights of the sherd graph. To justify its effectiveness, we replace it by directly computing the cosine similarity of feature vectors extracted by the curve feature descriptor. With this modification, we find the edge weights of both positive and negative sherd pair mostly ranges from 0.8 to 1, which is too close for the adaptive thresholding in Step 2. Accordingly, the purity, ARI and NMI of our method decrease to 0.535, 0.125, and 0.499, respectively.

We also try to directly use the angle difference to measure curve-pattern similarities by applying the arccos function to the cosine similarity – arccos similarity varies in a larger range than cosine similarity. With this metric, the resulting purity, ARI and NMI turn to 0.543, 0.128, and 0.502,

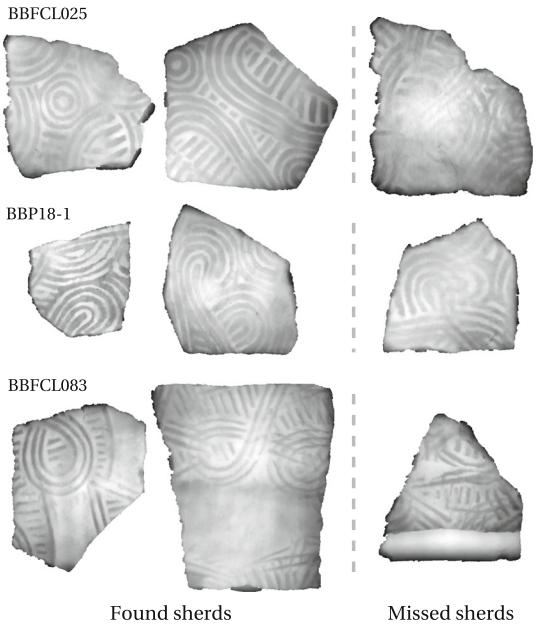

Fig. 16 Examples of the clustering result of our method. For each design, we show two found sherds and one missed sherd. (BBFCL025, BBP18-1, and BBFCL083 are names assigned by archaeologists to identify three unique designs drawn by Bettye Broyles)

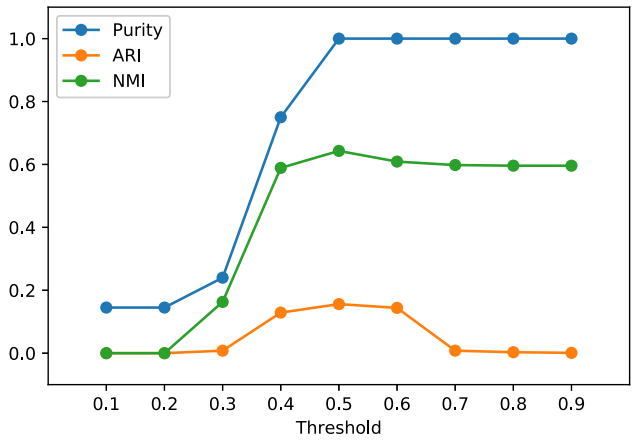

Fig. 17 Purity, ARI and NMI by using different thresholds in the naive thresholding strategy

respectively. We can see that, although it performs slightly better than the cosine similarity, it is still not comparable with the metric network, of which the resulting purity, ARI and NMI are 1, 0.148 and 0,644, respectively. Therefore, simply enlarging the varying range of similarity cannot solve this problem, and it is beneficial to use the similarity metric network to improve the discriminability.

Changing the graph partitioning strategy: In Step 2, we partition the fully-connected sherd graph using an adaptive thresholding strategy. A simple alternative approach is to use a pre-set threshold to cut low-weight edges to partition the graph. In this experiment, we compare the performance of using two different thresholding strategies in our method. Specifically, we evaluate the performance by using a naive threshold, which varies from 0.1 to 0.9 with a step length of 0.1 and the resulting curves of purity, ARI and NMI are shown in Fig. 17. We can see that, when the threshold is 0.5, the proposed naive thresholding leads to the best performance, resulting in the purity, ARI and NMI of 1, 0.156, and 0.643, respectively, which are close to the performance of adaptive thresholding. But the finding of this optimal threshold requires careful tuning on the test set, while the adaptive thresholding does not require it. Therefore we choose to use the adaptive thresholding strategy in our work.

Removing Step 3: As discussed above, after obtaining the initial clustering result, an iterative strategy was adopted in the proposed method to refine it by stitching curve patterns in the same cluster. To justify its usefulness, we skip it and report the performance of the initial clustering result, of which the purity, ARI and NMI are 1.0, 0.12, and 0.631, respectively. Comparing the results before and after the iterative refinement, we find that 11 isolated sherds in the initial result are assigned to their correct clusters in the second iteration, without introducing additional false positives.

5.5 Usefulness of Snowvision

As described earlier, our problem and dataset come from the practice of archaeology and contain many complexities and unique challenges, e.g., partial overlapping curves, composite patterns, and incomplete/noisy data. We design Snowvision to assist the archaeologists to process the data and explore/discover the sherd patterns more efficiently, instead of replacing the archaeologists with full automation, which is unachievable. From the above experiments, we can also see that, while our proposed algorithms outperform many existing state-of-the-art algorithms, their performances in three tasks are still far from the level for achieving full automation, e.g., fully automatic matching usually requires very high CMC Rank-1 accuracy. However, even with the current performance, our system is actually very useful to archaeologists. In this section, we briefly discuss the usefulness of our system by comparing the time an archaeologist might spend performing similar tasks to the time that those tasks take with the current Snowvision system. We run our system on an HP Proliant Server with dual Intel(R) Xeon(R) Platinum 8260 CPU, 192GB memory and an Nvidia Tesla P100 GPU card.

In Task I, we segment the depth images of sherds for stamped curve patterns. In this paper, Task I is a data preprocessing step for Tasks II and III on sherd curve matching. Therefore, time on Task I is simply built into the estimates provided below for Tasks II and III. The training processes of skeleton extraction network and dense prediction ConvNet

in our method take about 1.5 and 3 h, respectively. Once the models are trained, our segmentation method takes on average only 3.7 s to process one image in inference. On the smallest sherd of size 218×272, the processing time is 1.3 s, while on the largest sherd of size 794×903, the processing time is 15.6 s.

In Task II of sherd-to-design matching, the archaeologist has a sherd and wishes to see if the sherd curves match to a known design. If the archaeologist is very familiar with the design, e.g., it is a common design at a site, then manually matching a sherd to a design may go quickly. More often, however, the design is not very common, or the archaeologist may not have a lot of experience comparing sherds to designs. In this case, manual comparison of one sherd to each design may take as much as half a minute per design. With 133 designs to examine, manually comparing just one sherd to each design could take as many as 33.3 min to determine if a match is present, assuming we find its true design at the 67th design on average. With Snowvision, the archaeologists is presented with the top 20 matching results, with 66.7% accuracy, and must decide whether a match is present in the list. Such a streamlined sherd-to-design assessment will only take approximately 1min by visualizing and examining the patch-based matching results on average, to complete, which is far more efficient than the fully manual approach. In Task II, it takes us about 2 h to train the curve feature descriptor on the augmented training dataset. In inference, each sherd-to-design comparison with our trained models takes 0.823 s, averaged over 1000 testing sherds and 133 designs.

If the design on the sherd is not in the database, and the sherd has high quality curve data, then the curves can form the basis for novel design reconstruction, as in Task III of sherd-to-sherd matching. Traditionally the archaeologist will attempt to identify or group sherds by individual paddle design even though the paddle design is not known at the time of grouping. For comparison with Snowvision, we assume to identify and group 581 testing sherds of Task III, or 336,980 unique sherd-to-sherd comparisons. If half a minute is spent on each comparison, it would take an archaeologist as much as 2808 h to process 581 sherds with unknown designs. In Task III, we spend 1 h to train the similarity metric network. In inference, each sherd-to-sherd comparison with our model takes 0.536 s averaged over these 581 sherds. Clustering 581 sherds takes 50.2 h in total. Certainly, our clustering results is not perfect, and not all sherds from a new design are present in one cluster. However, such non-perfect clustering is still very useful for archaeologists–archaeologists only need to compare sherds in each cluster and a subset of sherds from a new design in a cluster may be sufficient for an archaeologist to infer/recreate the new design. A singleton cluster may indicate a standalone sherd, which is also useful information to archaeologists.

The performance of all three tasks in Snowvision likely can be improved with further research. Given the challenges of the problem and dataset, however, we believe Snowvision is unlikely to achieve full automation. Our final goal is to make human effort to find matches more efficient.

6 Conclusion

In this paper, we introduced Snowvision, a multidisciplinary project to digitize, segment, identify, and disseminate stamped curve patterns on unearthed pottery sherds. A research pipeline was developed for different users to study and share pottery sherds. At the core of Snowvision are three important computer-vision tasks of curve-pattern segmentation, design identification and sherd identification. Considering the characteristics of Snowvision data and the special needs of archaeologists, we developed a series of new algorithms by leveraging both the recent deep neural networks and traditional computer-vision techniques to tackle these three tasks. We evaluated the performance of the developed algorithms on a collection of pottery sherds with known paddle designs. Extensive experimental results, including comparison results with many existing methods and carefully-designed ablation studies, verified the effectiveness of the proposed algorithms.

Our project illustrates how important questions in one discipline can drive technical developments in another, and vice versa. Among the questions of import to archaeologists are how people organized themselves into groups (e.g., households, communities of households, communities of learning and practice), how those groups moved around on the landscape (e.g., through aggregation, mobility, and migration), and how and why social organization and other aspects of individual and group dynamics changed over time. Through designs impressed on to pottery, Swift Creek material culture offers us the chance to study the movement of people over the lifespan of a paddle. Tracking the movement of paddles and by extension people has significant potential to enhance our understanding of hunter-gatherer-forager micro- and macro-scale mobility in remote times. Snowvision facilitates this research in new and exciting ways, and we are eager to continue to explore the possibilities that computer vision affords the study of cultural heritage.

Acknowledgements This research is supported by the National Endowment for the Humanities (NEH) Digital Humanities Advancement Grant Program (HAA-266472-19), the National Science Foundation Archaeology and Archaeometry Grant Program (1658987), the National Center for Preservation Technology and Training Grant Program (P16AP00373), the Extreme Science and Engineering Discovery Environment (XSEDE) Science Gateway Program (DBS180011), University of South Carolina ASPIRE II Program, and University of South Carolina Social Science Provost Grant Program. We would like to show our gratitude to Frankie Snow at South Georgia State College for shar-

ing his pearls of wisdom and design images with us during the course of this research. We are very grateful to Dr. Matthew Compton, Curator of the R. M. Bogan Repository at Georgia Southern University, and Dr. Amanda Roberts Thompson, Operations Director of the Laboratory of Archaeology at the University of Georgia, for generously providing access to collections, and Scot Keith at New South Associates for his enthusiasm and encouragement of this research. We would like to thank Research Computing at University of South Carolina for high-performance computing support, and the South Carolina Department of Natural Resources for institutional support. The Muscogee (Creek) Nation has approved the scanning of Broyles Collection sherds shown in Figure 16 and requests consultation on any future use of the scans and images including but not limited to presentations, publications, manuscripts, social media posts, and news stories.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Alilou, V.K. (2020). Fingerprint matching: A simple approach. https://github.com/alilou63/fingerprint.

Anichini, F., Banterle, F., Garrigós, J., Callieri, M., Dershowitz, N., Dubbini, N., Diaz, D. L., Evans, T., Gattiglia, G., Green, K., et al. (2020). Developing the archaide application: A digital workflow for identifying, organising and sharing archaeological pottery using automated image recognition. *Internet Archaeol, 52*, 1–48.

Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 33*(5), 898–916.

Arthur, D., & Vassilvitskii, S. (2006). *k-means++: The advantages of careful seeding*. Tech. rep.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561.

Balntas, V., Lenc, K., Vedaldi, A., & Mikolajczyk, K. (2017). Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: IEEE conference on computer vision and pattern recognition, pp. 5173–5182.

Banterle, F., Itkin, B., Dellepiane, M., Wolf, L., Callieri, M., Dershowitz, N., & Scopigno, R. (2017). Vasesketch: Automatic 3d representation of pottery from paper catalog drawings. In: International conference on document analysis and recognition, IEEE, vol. 1, pp. 683–690.

Barrow, H., Tenenbaum, J., Bolles, R., & Wolf, H. (1977). Parametric correspondence and Chamfer matching: Two new techniques for image matching. In: International joint conference on artificial intelligence, vol. 2, pp. 659–663.

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In: European conference on computer vision, Springer, pp. 404–417.

Belongie, S., Malik, J., & Puzicha, J. (2001). Shape Context: A new descriptor for shape matching and object recognition. In: Advances in Neural Information Processing Systems, pp. 831–837.

Brown, M., & Lowe, D. (2007). Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision, 74*(1), 59–73.

Broyles, B. J. (1968). Reconstructed designs from swift creek complicated stamped sherds. *Southeastern Archaeological Conference Bulletin, 8*, 49–55.

Brunelli, R. (2009). *Template matching techniques in computer vision: Theory and practice*. John Wiley & Sons.

Bundy, A., & Wallen, L. (1984). Difference of gaussians. In: Catalogue of Artificial Intelligence Tools, Springer, p. 30.

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In: European conference on computer vision, pp. 132–149.

Carson, C., Thomas, M., Belongie, S., Hellerstein, J.M., & Malik, J. (1999). Blobworld: A system for region-based image indexing and retrieval. In: International conference on advances in visual information systems, Springer, pp. 509–517.

Chan, J., Addison Lee, J., & Kemao, Q. (2017). Bind: Binary integrated net descriptors for texture-less object recognition. In: IEEE conference on computer vision and pattern recognition, pp. 2068–2076.

Chan, T. F., & Vese, L. A. (2001). Active contours without edges. *IEEE Transactions on Image Processing, 10*(2), 266–277.

Chang, J., Wang, L., Meng, G., Xiang, S., & Pan, C. (2017). Deep adaptive image clustering. In: Proceedings of the IEEE international conference on computer vision, pp. 5879–5887.

Chao, P., Kao, C.Y., Ruan, Y.S., Huang, C.H., & Lin, Y.L. (2019). Hardnet: A low memory traffic network. In: IEEE International conference on computer vision, pp. 3552–3561.

Chen, L.C., Papandreou, G., Kokkinos, I., & Murphy, K., Yuille, A.L. (2016). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv preprint arXiv:1606.00915.

Chen, L.C., Zhu, K., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision, pp. 801–818.

Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(5), 603–619.

Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., & Nießner, M. (2017) Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5828–5839.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In: IEEE conference on computer vision and pattern recognition, vol. 1, pp. 886–893.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision, 88*(2), 303–338.

Frenkel, M., & Basri, R. (2003). Curve matching using the fast marching method. In: International workshop on energy minimization methods in computer vision and pattern recognition, Springer, pp. 35–51.

Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. *Science, 315*(5814), 972–976.

Fu, F., Wei, J., Zhang, M., Yu, F., Xiao, Y., Rong, D., Shan, Y., Li, Y., Zhao, C., Liao, F., et al. (2020). Rapid vessel segmentation and reconstruction of head and neck angiograms using 3d convolutional neural network. *Nature Communications, 11*(1), 1–12.

Fu, L. (2019). A study of geometric stamped pattern pottery and early maritime cultural interactions between mainland china and taiwan. In: Prehistoric Maritime Cultures and Seafaring in East Asia, Springer, pp. 235–249.

Gamble, C. (2004). *Archaeology: The basics*. Rutledge.

Gualandi, M. L., Gattiglia, G., & Anichini, F. (2021). An open system for collection and automatic recognition of pottery through neural network algorithms. *Heritage, 4*(1), 140–159.

Han, D., & Hahn, H. S. (2014). Axis estimation and grouping of rotationally symmetric object segments. *Pattern Recognition, 47*(1), 296–312.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition, pp. 770–778.

Hinami, R., Matsui, Y., & Satoh, S. (2017). Region-based image retrieval revisited. In: ACM international conference on multimedia, pp. 528–536.

Hoffer, E., & Ailon, N. (2015). Deep metric learning using triplet network. In: International workshop on similarity-based pattern recognition, Springer, pp. 84–92.

Jain, A., Ross, A., & Prabhakar, S. (2001). Fingerprint matching using minutiae and texture features. In: International conference on image processing, IEEE, vol. 3, pp. 282–285.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall Inc.

Kampel, M., & Sablatnig, R. (2003). Profile-based pottery reconstruction. In: IEEE conference on computer vision and pattern recognition—Workshop, IEEE, vol. 1, p. 4.

Kong, B., Supancic, J., Ramanan, D., & Fowlkes, C. C. (2019). Cross-domain image matching with deep feature maps. *International Journal of Computer Vision, 127*(11), 1738–1750.

Lam, L., Lee, S. W., & Suen, C. Y. (1992). Thinning methodologies-a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 14*(9), 869–885.

Li, C., Xu, C., Gui, C., & Fox, M. D. (2010). Distance regularized level set evolution and its application to image segmentation. *IEEE Transactions on Image Processing, 19*(12), 3243–3254.

Li, H., Song, D., Liu, Y., & Li, B. (2018). Automatic pavement crack detection by multi-scale image fusion. *IEEE Transactions on Intelligent Transportation Systems, 20*(6), 2025–2036.

Li, Z., & Chen, J. (2015). Superpixel segmentation using linear spectral clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1356–1363.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory, 28*(2), 129–137.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In: IEEE conference on computer vision and pattern recognition, pp. 3431–3440.

Lorigo, L. M., Faugeras, O. D., Grimson, W. E. L., Keriven, R., Kikinis, R., Nabavi, A., & Westin, C. F. (2001). Curves: Curve evolution for vessel segmentation. *Medical Image Analysis, 5*(3), 195–206.

Lowe, D.G. (1999). Object recognition from local scale-invariant features. In: IEEE international conference on computer vision, IEEE, vol. 2, pp. 1150–1157.

Lu, Y., Zhou, J., Chen, J., Wang, J., Smith, K., Colin, W., & Wang, S. (2018). Curve-structure segmentation from depth maps: A CNN-based approach and its application to exploring cultural heritage objects. AAAI Conference on Artificial Intelligence.

Lucena, M., Martínez-Carrillo, A., Fuertes, J. M., Carrascosa, F., & Ruiz, A. (2016). Decision support system for classifying archaeological pottery profiles based on mathematical morphology. *Multimedia Tools and Applications, 75*(7), 3677–3691.

MacDonald, L.W. (2015). Realistic visualisation of cultural heritage objects. Ph.D. thesis, UCL (University College London).

Makridis, M., & Daras, P. (2012). Automatic classification of archaeological pottery sherds. *Journal on Computing and Cultural Heritage, 5*(4), 15.

Markuš, N., Pandžić, I., & Ahlberg, J. (2018). Learning local descriptors by optimizing the keypoint-correspondence criterion: Applications to face matching, learning from unlabeled videos and 3d-shape retrieval. *IEEE Transactions on Image Processing, 28*(1), 279–290.

Martínez-Carrillo, A.L. (2008). Computer applications in archaeological pottery: A review and new perspectives. In: Conference on computer applications and quantitative methods in archaeology, vol. 2, p. 6.

Minaee, S., Boykov, Y.Y., Porikli, F., Plaza, A.J., Kehtarnavaz, N., & Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(7), 971–987.

Ono, Y., Trulls, E., Fua, P., & Yi, K.M. (2018). Lf-net: Learning local features from images. In: Advances in Neural Information Processing Systems, pp. 6234–6244.

Ostertag, C., & Beurton-Aimar, M. (2020). Matching ostraca fragments using a siamese neural network. *Pattern Recognition Letters, 131*, 336–340.

Otto, C., Wang, D., & Jain, A. K. (2017). Clustering millions of faces by identity. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*(2), 289–303.

Pirrone, A., Aimar, M.B., & Journet, N. (2019). Papy-s-net: A siamese network to match papyrus fragments. In: International workshop on historical document imaging and processing, pp. 78–83.

Pizer, S.M., Johnston, R.E., Ericksen, J.P., Yankaskas, B.C., & Muller, K.E. (1990). Contrast-limited adaptive histogram equalization: Speed and effectiveness. In: Conference on visualization in biomedical computing, IEEE Computer Society, pp. 337–338.

Rasheed, N.A., & Nordin, M.J. (2018). Classification and reconstruction algorithms for the archaeological fragments. Journal of King Saud University-Computer and Information Sciences.

Rother, C., Kolmogorov, V., & Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics, 23*(3), 309–314.

Roux, V. (2019). *Ceramics and society: A technological approach to archaeological assemblages*. Springer.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In: International conference on computer vision, IEEE, pp. 2564–2571.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision, 115*(3), 211–252.

Shen, W., Zhao, K., Jiang, Y., Wang, Y., Zhang, Z., & Bai, X. (2016). Object skeleton extraction in natural images by fusing scale-associated deep side outputs. In: IEEE conference on computer vision and pattern recognition, pp. 222–230.

Shen, X., Wang, C., Li, X., Yu, Z., Li, J., Wen, C., Cheng, M., & He, Z. (2019). Rf-net: An end-to-end image matching network based on receptive field. In: IEEE conference on computer vision and pattern recognition, pp. 8132–8140.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(8), 888–905.

Smith, K. Y., & Knight, V. J. (2012). Style in swift creek paddle art. *Southeastern Archaeology, 31*(2), 143–156.

Smith, K. Y., & Stephenson, K. (2018). The spatial dimension of the woodland period. *Southeastern Archaeology, 37*(2), 112–128.

Smith, P., Bespalov, D., Shokoufandeh, A., & Jeppson, P. (2010). Classification of archaeological ceramic fragments using texture and color descriptors. In: IEEE conference on computer vision and pattern recognition—Workshops, pp. 49–54.

Snow, F. (1975). Swift creek designs and distributions: A south Georgia study. *Early Georgia, 3*(2), 38–59.

Snow, F. (1998). Swift creek design investigations. A World Engraved: Archaeology of the Swift Creek Culture pp. 61–98.

Son, K., Almeida, E.B., Cooper, D.B. (2013). Axially symmetric 3d pots configuration system using axis of symmetry and break curve. In:

IEEE conference on computer vision and pattern recognition, pp. 257–264.
Stamatopoulos, M.I., & Anagnostopoulos, C.N. (2016). 3d digital reassembling of archaeological ceramic pottery fragments based on their thickness profile. arXiv preprint arXiv:1601.05824.
Talcott, L. (1935). Attic black-glazed stamped ware and other pottery from a fifth century well. *Hesperia: The Journal of the American School of Classical Studies at Athens, 4*(3), 477–523.
Tao, X., Prince, J. L., & Davatzikos, C. (2002). Using a statistical shape model to extract sulcal curves on the outer cortex of the human brain. *IEEE Transactions on Medical Imaging, 21*(5), 513–524.
W, J. *et al* (2020). Deep high-resolution representation learning for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence.
Tico, M., & Kuosmanen, P. (2003). Fingerprint matching using an orientation-based minutia descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 25*(8), 1009–1014.
Tombari, F., Franchi, A., & Di Stefano, L. (2013). Bold features to detect texture-less objects. In: IEEE International conference on computer vision, pp. 1265–1272.
Tremeau, A., & Borel, N. (1997). A region growing and merging algorithm to color segmentation. *Pattern Recognition, 30*(7), 1191–1203.
Vese, L. A., & Chan, T. F. (2002). A multiphase level set framework for image segmentation using the mumford and shah model. *International Journal of Computer Vision, 50*(3), 271–293.
Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., & Yu, T. (2014). scikit-image: Image processing in python. *PeerJ, 2*, e453.
Wang, S., Kubota, T., & Siskind, J.M. (2004). Salient boundary detection using ratio contour. In: Advances in Neural Information Processing Systems, pp. 1571–1578.
Wang, S., & Siskind, J. M. (2003). Image segmentation with ratio cut. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 25*(6), 675–690.
Wang, Z., Zheng, L., Li, Y., & Wang, S. (2019). Linkage based face clustering via graph convolution network. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 1117–1125.
Willis, A., Orriols, X., & Cooper, D.B. (2003). Accurately estimating sherd 3d surface geometry with application to pot reconstruction. In: IEEE conference on computer vision and pattern recognition—Workshops, IEEE, vol. 1, p. 5.
Willis, A.R., & Cooper, D.B. (2004). Bayesian assembly of 3d axially symmetric shapes from fragments. In: IEEE conference on computer vision and pattern recognition, IEEE, vol. 1, p. I.
Wu, J., Long, K., Wang, F., Qian, C., Li, C., Lin, Z., & Zha, H. (2019). Deep comprehensive correlation mining for image clustering. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 8150–8159.
Wu, K. L., & Yang, M. S. (2007). Mean shift-based clustering. *Pattern Recognition, 40*(11), 3035–3052.
Xie, S., & Tu, Z. (2015). Holistically-nested edge detection. In: IEEE international conference on computer vision, pp. 1395–1403.
Yeh, C.K., Matsuda, N., Huang, X., Li, F., Walton, M., & Cossairt, O. (2016). A streamlined photometric stereo framework for cultural heritage. In: European conference on computer vision, Springer, pp. 738–752.
Yi, K.M., Trulls, E., Lepetit, V., & Fua, P. (2016). Lift: Learned invariant feature transform. In: European conference on computer vision, Springer, pp. 467–483.
Zhan, X., Liu, Z., Yan, J., Lin, D., & Loy, C.C. (2018). Consensus-driven propagation in massive unlabeled data for face recognition. In: Proceedings of the European conference on computer vision (ECCV), pp. 568–583.
Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., & Torr, P.H. (2015). Conditional random fields as recurrent neural networks. In: IEEE international conference on computer vision, pp. 1529–1537.
Zou, Q., Cao, Y., Li, Q., Mao, Q., & Wang, S. (2012). Cracktree: Automatic crack detection from pavement images. *Pattern Recognition Letters, 33*(3), 227–238.
Žunić, J., Hirota, K., & Rosin, P. L. (2010). A hu moment invariant as a shape circularity measure. *Pattern Recognition, 43*(1), 47–57.