Interactive 3D Segmentation of Rock-Art by Enhanced Depth Maps and Gradient Preserving Regularization*

Matthias Zeppelzauer1, Georg Poier2, Markus Seidl1, Christian Reinbacher2, Samuel Schuler2, Christian Breiteneder3, and Horst Bischof2

1St. Pölten University of Applied Sciences
2Graz University of Technology
3Vienna University of Technology

Abstract

Petroglyphs (rock engravings) have been pecked and engraved by humans into natural rock surfaces thousands of years ago and are among the oldest artifacts that document early human life and culture. Some of these rock engravings have survived until the present and serve today as a unique document of ancient human life. Since petroglyphs are pecked into the surface of natural rocks they are threatened by environmental factors as weather and erosion. To document and preserve these valuable artifacts of human history the 3D digitization of rock surfaces has become a suitable approach due to the development of powerful 3D reconstruction techniques in recent years. The results of 3D reconstruction are huge 3D point clouds which represent the local surface geometry in high resolution. In this paper, we present an automatic 3D segmentation approach that is able to extract rock engravings from reconstructed 3D surfaces. To solve this computationally complex problem, we transfer the task of segmentation to the image-space in order to efficiently perform segmentation. Adaptive learning is applied to realize interactive segmentation and a gradient preserving energy minimization assures smooth boundaries for the segmented figures. Our experiments demonstrate the efficiency and the strong segmentation capabilities of the approach. The precise segmentation of petroglyphs from 3D surfaces provides the foundation for compiling large petroglyph databases which can then be indexed and searched automatically.

1 Introduction

Petroglyphs have been scratched, carved and pecked into rock panels all over the world resulting in a vast number of engraved figures on natural rock surfaces. These figures represent an important artifact for the documentation and study of

*©ACM, Journal on Computing and Cultural Heritage (JOCCH). This is the author’s version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in JOCCH, 9, 4, Article 19 (2016), https://doi.org/10.1145/2950962
human history and cultural development. Traditionally, petroglyph shapes were documented (together with their locations on a rock panel) by manually tracing the shapes as illustrated in Fig. 1. This is an extremely time-consuming task considering the fact that hundreds of thousands of petroglyphs exist that have so far not been documented in detail [1]. Furthermore, pure tracings capture only the 2D shape and are not able to represent the full 3D characteristics (depth) of petroglyphs.

In recent years continuous progress in the development of 3D acquisition techniques has been made. This resulted in increasing activity in the 3D scanning of rock surfaces, especially in archeology. Although questions regarding long-term storage and handling of the vast amounts of digitized 3D data still have to be answered, we observe that 3D reconstruction gains increasing importance for the documentation and conservation of rock-art. The detailed 3D reconstruction of petroglyphs does not only enable a precise documentation but further facilitates the development of (semi-)automated tools for analysis, indexing, and retrieval of petroglyph shapes to support rock-art researchers in their daily work and to make petroglyphs easier accessible to the public.

Existing work on the automated analysis of rock-art deals mostly with the recognition and classification of pre-segmented petroglyph shapes [3, 4, 5, 6, 7, 8, 9]. The automatic segmentation of petroglyph shapes is an important preprocessing step which has hardly been approached and has even been considered infeasible by some researchers [3]. Only a few works on the segmentation of petroglyph shapes exist. For instance, Deufemia et al. [7] use manual tracings as shown in Fig. 1 as input for segmentation. Since manual tracings already
represent a precise segmentation of the shapes, the task reduces to the labelling of the tracings in a binary image. For many sites manual tracings do not exist. To automate the documentation and to reduce the efforts for manual tracing, we [10] proposed a method that is able to segment petroglyphs in high-resolution photographs of a rock surface. The approach performs pixel-based classification where each pixel is represented by its local neighborhood (a fixed-size patch). Results of the study show that segmentation can in many cases be solved by this 2D approach. For the description of the pixel neighborhood, texture features such as LBP (Local Binary Patterns) and variants of GLCM (Gray Level Co-Occurrence Matrix) have proven to be most effective. However, in practice there is one central obstacle of working with 2D photographs that strongly influences the quality of the results: The dependence on the lighting situation. The incoming light direction strongly influences how the three-dimensional micro-structure of a surface is finally depicted in a photograph. On an overcast day with ambient lighting the surface structure becomes less visible than on a sunny day in the morning or in the afternoon, when the sun rays are close to parallel to the surface. Near parallel illumination is beneficial for image-based 2D approaches as it generates stronger shadows and thus better reflects the surface geometry.

An alternative way to capture the surface structure is to directly apply 3D scanning and recognition, which is more and more becoming a standard tool in archaeological documentation. To our knowledge, our work is the first to present a method for segmentation of rock-art from high-resolution 3D surface reconstructions. The method is able to automatically segment petroglyphs from a given 3D reconstruction and further allows the user to interactively refine the segmentation. To accelerate processing, we map the 3D input data (usually a point cloud) to image space (a depth map) and extract the surface topography by enhancing the geometric micro-structure captured by the depth map. The resulting enhanced depth map is input to a classifier which estimates the probability that a given pixel belongs to a pecked region. Next, we optimize the contour of the segmented shapes to reduce noise and improve the smoothness. Finally, we incorporate user-input in terms of scribbles to iteratively refine the segmentation. Figure 2 illustrates the segmentation process from the input.

Figure 2: The proposed semi-automatic segmentation process: The left image shows the input point cloud viewed from projection direction. The middle image shows the result of the initial fully automated segmentation. The image on the right shows the final result after manual refinement by using scribbles. The green scribbles represent foreground, i.e., pecked areas and the red scribbles background, i.e., natural rock surface.
point cloud (left), to the fully automatic segmentation (middle), to the interactively refined segmentation (right). Our quantitative and qualitative evaluations show that the proposed method provides fully automatic segmentations with high-quality and that interactive refinement is useful to improve results. Moreover, an implementation of the method using modern graphics processing units (GPUs) enables immediate feedback in less than one second for a one megapixel image. We further show that segmentation on 3D data strongly outperforms pure 2D segmentation. This underlines the necessity of 3D representations for the documentation and representation of rock-art.

The proposed interactive segmentation approach has originally been presented in [11]. In this article, we describe our approach in more detail, add an extended motivation for our approach for image-space topography extraction and put the entire work in a broader context. Furthermore, we provide a more comprehensive evaluation of the results both in a qualitative and in a quantitative manner.

The paper is organized as follows. In Section 2 we review related work on the automatic 3D acquisition, segmentation, and higher-level processing of rock-art. Section 3 describes our 3D segmentation approach from image-space topography extraction to interactive segmentation and boundary regularization. We introduce the material used in our evaluation and the experimental setup in Section 4. Section 5 presents quantitative and qualitative results. Finally, we draw conclusions in Section 6.

2 Related Work

This work mainly focuses on the segmentation of petroglyphs from 3D meshes and pointclouds. Additionally to reviewing related segmentation techniques, we briefly review related work on the 3D acquisition of rock-art as well as methods for the recognition and classification of already segmented rock-art to put this work into a broader context.

3D Data Acquisition Petroglyphs have been scratched and pecked into rock surfaces and thus represent three-dimensional artifacts. Thus, it is a natural approach to capture and analyze these artifacts in their original domain. The first step in a fully 3D processing pipeline for ancient rock-art is the 3D reconstruction of the rock surfaces. Suitable 3D reconstruction algorithms are based for example on structure-from-motion (SfM) (e.g., [12, 13]) and structured light scanning (SLS) (e.g., [14]). They enable the accurate reconstruction of surfaces at sub-millimeter resolution. Thus, 3D reconstruction has become a promising technique for the digitization of rock-art. An early example in the rock-art domain is presented by [15] who propose a system for 3D acquisition and presentation of Puerto Rican petroglyphs. A more recent approach that considers multiple scales to acquire 3D data of petroglyphs together with their surrounding rock panel has been proposed by [16]. In [17, 18] 3D laser scanning is used to capture stone inscriptions that have been carved into rock. In our work, we employ high-resolution SfM and SLS scans with resolutions of up to 0.1 mm to capture the geometric micro-structures related to individual peck-marks and scratches.
3D Surface Representation  The results of 3D reconstruction are either large point clouds or already triangulated meshes. In both cases we are interested in the extraction of the geometric details that make up the irregularities (e.g. peck-marks) across a surface, the so called surface texture. Surface texture refers to the geometric micro-structure of a surface in terms of roughness, waviness, lay, coarseness, smoothness, polish, burnish, and bumpiness [19, 20]. It is defined as the repetitive random deviation from the “ideal” surface. This deviation forms the three dimensional topography of a surface [21].

Parameters for the description of surface topography have been defined in different scientific domains. Geomorphometry (a branch of geology) introduces a broad range of parameters for the description of land surfaces and terrains in terms of peaks, valleys, ridges, passes, etc. [22]. Material science provides a rich vocabulary for the description of surface topography (e.g. in terms of roughness, waviness, and lay) and specifies a broad range of measures for its parametrization [23, 24, 19, 25]. A categorization of surface parameters employed in archeology is presented in [20].

Additional to the surface parameters defined in the domains mentioned above, powerful higher-dimensional surface descriptors have been introduced in the computer science domain in recent years. Numerous 3D descriptors exist, which enable a precise description of the local geometry around a point or mesh vertex. [26] propose spin images as local descriptors for the dense description of meshes for the task of object recognition in 3D scenes. Darom and Keller introduce a scale detection scheme for mesh points and propose scale-invariant spin images [27]. Furthermore they extract SIFT features from local depth images of a points’ neighborhood to model the surface topography around it. [28] extend the well-known 2D shape context descriptor [29] to 3D point clouds (3DSC) and show that it outperforms spin images. [30] introduce a local surface descriptor for meshes (MeshHOG) that can be computed from an arbitrary scalar function (e.g. curvature) defined over the surface. [31] propose the normal-aligned radial feature (NARF) which captures local depth variations in range data. [32] propose two local 3D point cloud descriptors, namely persistent point feature histogram (PFH) and an accelerated version fast PFH (FPFH) in [33]. Both build upon the relations between surflets, i.e. the combination of a surface point and its normal [34]. [35] propose a 3D descriptor (SHOT) based on the point normals that is defined in a robust local reference frame.

The local descriptors above are in general well-suited for the description of attributes related to surface topography although most of them were originally developed for different purposes (e.g. salient feature point description). The major challenge lies in the dense extraction of the descriptors from a point cloud (or mesh) which is necessary to capture the topography continuously across a surface. Even if the extraction is restricted to a regular grid of points with a larger spacing the task becomes computationally demanding and not applicable to large-scale data.

An alternative approach is to map the 3D data to two dimensions and to perform topography analysis much more efficiently in the image domain. This allows to use powerful image features, such as local binary patterns (LBP) [36] and histograms of oriented gradients (HOG) [37]. The mapping of an arbitrary 3D surface to a plane is however non-trivial and usually introduces distortions.

An approach for the mapping of topographic information from curved surfaces into 2D has recently been proposed by [38]. The idea of the approach is
to extract local geometric variations in 3D and to map them to a 2D deviation map that represents the surface topography. The authors first smooth the 3D surface to obtain a representation of the global geometry. Next, they compute the local geometric deviations by subtracting the smoothed surface from the original one. The smoothed surface is next mapped to 2D by non-linear dimensionality reduction using the Isomap algorithm [39]. Isomap is a generalization of multidimensional scaling (MDS) where the Euclidean distances are replaced by geodesic distances which enables the modeling of non-linear manifolds. After dimensionality reduction the previously computed deviations are assigned to the mapped 2D points. The resulting 2D deviation map is rasterized and can be efficiently analyzed further using image features. The major shortcoming of the approach in [38] is that the non-linear dimensionality reduction is computationally complex and becomes highly demanding for large-scale data. This is problematic as high-resolution 3D reconstructions usually exhibit several millions of points.

We follow the general idea of [38] but compute the deviation map in the image space without the need for non-linear dimensionality reduction. This reduces the computational complexity significantly, as most operations can be performed efficiently in image space.

Petroglyph Segmentation  After 3D reconstruction and the extraction and parametrization of surface topography a major challenge is the robust segmentation of petroglyphs from the digitized surfaces. The automatic segmentation of rock surfaces containing petroglyphs has rarely been addressed so far. Deufemia and Paolino [7] apply segmentation to digitized manual tracings. This is a rather straight-forward task since manual tracings are binary images that already represent a manual pre-segmentation of the figures. Previously, we proposed an approach for the segmentation of petroglyphs directly from images of natural rock surfaces [10]. We employed high-resolution digital photos of rock surfaces as input and extracted a set of texture features from the images. Next, we trained an ensemble of SVM classifiers on the features and proposed a fusion of the classifier outputs that enabled the interactive fine-tuning of the segmentation by the user.

In contrast to our previous approach in [10], we use 3D point clouds of the rock surface instead of photographs. In photos the geometric surface structure is reflected by the image texture. Image texture is, however, highly dependent on illumination. Light (almost) parallel to the surface, for example, creates shadows which reflect certain properties of the surface geometry, while orthogonal light makes the surface structure practically invisible. Our approach solely uses 3D data as input. Thus, once the 3D reconstruction is accomplished, we can operate independently from lighting on the surface. Additionally, the full 3D information that makes up the petroglyphs is preserved.

Petroglyph Classification  Segmentation is an important pre-processing step to enable further higher-level analysis tasks as the classification of petroglyph shapes and the search for similar shapes in a database. [3, 4], for example, use a modified Generalized Hough Transform (GHT) for the mining of large petroglyph shape datasets. Deufemia et al. [5] apply the Radon transform as a shape descriptor for unsupervised shape recognition via self-organizing maps (SOM).
In a second step, they use a fuzzy visual language parser to solve ambiguous interpretations by incorporating archaeological knowledge. Additionally they propose a two-stage classification of petroglyphs \[6\]. The authors employ shape context descriptors to generate an initial clustering with SOMs. In the second step, they use an image deformation model to classify petroglyph shapes. Subsequently, Deufemia et al. \[7\] use Fourier descriptors to detect and classify petroglyphs from full scenes which stem from manual tracings. Seidl et al. \[8, 9\] combine skeletal descriptors with shape descriptors for petroglyph classification. In absence of a robust segmentation approach the above mentioned methods build upon pre-segmented or manually segmented figures. The segmentation approach proposed in this paper aims at closing this gap in the processing pipeline.

3 3D Segmentation Approach

In this section we present our 3D segmentation approach that builds upon a novel method for the extraction of surface topography. To motivate our approach, we first demonstrate the extraction of surface topography according to \[38\] in Section 3.1 and then introduce our image-space method for topography extraction in Section 3.2. For a description of the approach by Othmani et al. see Section 2. The result of topography extraction is a 2D deviation map that represents the geometric micro-structure of the surface. In Section 3.3 we describe how to extract and emphasize the geometric patterns related to the peck-marks in the map that make up petroglyph shapes. On the resulting map - the enhanced deviation map (EDM) - a classifier is trained that generates probabilities for each pixel that express the likelihood that a pixel belongs to a pecked region. Akin to Santner et al. \[40\], the resulting probability map is input to an energy minimization that generates a binary segmentation of the input with a smooth contour. To leave the final decision of the segmentation to the user, we incorporate a mechanism for iterative refinement of the segmentation result. The additional user input enables our method to learn incrementally and to improve future segmentations.

3.1 3D topography extraction

To demonstrate topography extraction in the following we employ a 3D surface reconstruction with \(P = 1 \cdot 10^6\) points, see Figure 3(a). The surface represents a curved rock with different topographic patterns. We first extract surface topography in 3D according to \[38\]. Then we smooth the surface (see Figure 3(b)) and compute the deviations between the smoothed surface and the original one. For visualization purposes we map the deviations onto the original surface in Figure 3(c). Next, we apply nonlinear dimensionality reduction to the smoothed point cloud (by Isomap \[39\]) and map the deviations onto the obtained 2D projection. The resulting 2D deviation map captures the geometric micro-structure of the surface, which can be observed from Figure 3(d).

The computation of the deviation map consumes considerable computation time. The most demanding processing step is dimensionality reduction since Isomap has a runtime of \(O(P^2 + P^3) = O(P^3)\) and space requirements of \(O(P^2)\) \[11, 42\] which makes the processing of point clouds with several mil-
Figure 3: Extraction of surface topography in 3D according to [38]. The point cloud (a) is smoothed (b) and the deviations from the smoothed surface (c) are mapped to 2D by nonlinear dimensionality reduction. The result is a 2D representation of surface topography (d). Red color indicates large deviations and blue color small deviations.

ions of points intractable on standard hardware. To work around this problem, we employ Isomap with sparse MDS by using only a subset of landmark points for modeling [42]. Thereby, the runtime reduces to $O(nP + knP + n^2) = O(P)$ and the memory consumption decreases theoretically to $O(nP) = O(P)$, where $n$ is the number of landmarks with $n \ll P$ and $k = 2$ the embedding dimension. This makes the computation feasible but still the processing of the 1M point cloud from Figure 3(a) requires approximately 10h on a standard PC (Intel Xeon E5 with 8 cores). For another point cloud with 4M points the algorithm does not terminate in reasonable time.

3.2 Image-space topography extraction

As pointed out in the previous section, the extraction of surface topography in 3D is computationally demanding and not feasible for larger point clouds. We propose an alternative approach that first projects the point cloud to a 2D depth map and then extracts topographic information from the projected point cloud resulting in a much more efficient way to compute the deviation map [43].

As there are usually no self-occlusions in pecked rock surfaces, this mapping can be done without loss of information. In this case, the depth map is able to represent the full geometric information of the surface.

The input to our approach is a high-resolution point cloud with $P$ points. In a first step, we estimate a support plane for the input point cloud that minimizes the least squares distances to the plane. Next, we estimate the location of each 3D point on the support plane by an orthographic projection with projection direction orthogonal to the support plane. We map the signed distances
Figure 4: Extraction of surface topography in 2D: (a) the point cloud viewed from the support plane’s normal direction; (b) the depth projection on the support plane; (c) the reconstructed deviation map according to our approach; (d) an annotation that labels surface areas with different topography, such as the human-shaped figure in the center whose head is marked with an arrow.

between the 3D points and the support plane (along the normal direction) to the respective projected location on the support plane. The result is a 2D depth map of the 3D surface that captures both, the global geometry of the surface as well as the local topography. Figures 4(a) and 4(b) show the point cloud from Figure 3 viewed from the projection direction together with its projected depth map.

The depth map clearly shows the global curvature of the surface which is largest along the x-direction. The global variations strongly dominate the local variations and, hence, make them hardly recognizable. Note that the topography is also difficult to recognize from the visual appearance of the point cloud (see Figure 4(a)) which shows that surface topography is not easily visible from color texture.

The extraction of the surface topography from the depth map requires the compensation of the global geometric variations. We first smooth the depth map to obtain the global depth distribution of the underlying 3D surface by convolution with a two-dimensional Gaussian-shaped filter:

\[ G(x, y) = e^{-\frac{x^2 + y^2}{\sigma^2}}, \]

with standard deviation \( \sigma \). The filter is computed for a local window of \( W^2 \) pixels in x and y direction and is normalized to \( \sum_x \sum_y G(x, y) = 1 \). The filter is applied to all possible locations \( (x, y) \) of the original depth map \( D \) by:

\[ \hat{D}(x, y) = G * D = \sum_{i=-W/2}^{W/2} \sum_{j=-W/2}^{W/2} G(i, j)D(x + i, y + j). \]
The result is an estimation of the local average for all surface locations. Next, the local average is subtracted from the depth map at every location of the surface: \( \hat{D}(x, y) = D(x, y) - \bar{D}(x, y) \). The result is a deviation map \( \hat{D} \) similarly to the approach in [38]. For both maps we observe similar patterns (compare Figures 4(c) and 3(d)), which can for example be observed from the local maxima in the maps (red areas) which in both cases represent rock crannies. Note, for a better comparison with the deviation map from [38], which contains only absolute values, Figure 4(c) shows the absolute deviation map \( |\bar{D}| \).

Our deviation map \( \hat{D} \) is free of global geometric variations and captures local micro-structures of a surface that are not recognizable from the depth map, such as rock crannies and surface irregularities. Additional topographic structures on the surface in Figure 4 originate from human-made engravings in the rock surface that are different in topography than the surrounding rock surface. We provide a manual labeling of all human-made engravings in Figure 4(d) to facilitate the localization of the related patterns in the deviation map. The deviation map shows patterns that correlate well with the annotations, which indicates that the deviation map well represents the different topographies. This can for example be observed for the human-shaped figure in the center of the surface whose head is marked by an arrow in Figure 4.

From the comparison of our deviation map in Figure 4(c) with that of [38] in Figure 3(d) we observe that both maps have different shapes. As the investigated surface is not developable (non-zero Gaussian curvature) it cannot be flattened onto a plane without distortions. The different shapes originate from the different types of projections used in both approaches. While Isomap overestimates the curvature along the y-axis, the proposed approach underestimates the curvature along the x-axis for the given point cloud.

With increasing global curvature the distortions increase as well (independent of the employed flattening approach). To keep the distortions small locally, we propose to partition the surface into regions of limited curvature and to compute several deviation maps. This concept can be further generalized by computing a local deviation map by local depth projections for each pixel to minimize distortions.

Apart from unavoidable distortions, the computation of our deviation map is much more efficient. The orthogonal projection can be performed in linear time \( O(P) \) with respect to the number of points \( P \) in the input point cloud. The smoothing takes \( O(Nk) \), where \( N \) is the number of pixels in the projected image and \( k \) the number of pixels in the convolution kernel. The remaining image processing operations (subtraction and optionally taking the absolute) are in \( O(N) \). Thus, the method is linear in \( P \) and \( N \).

In contrast to the map generated by the 3D approach, our deviation map \( \hat{D} \) is signed. The sign encodes whether a projected point is below or above the smoothed average surface. The sign enables the distinction between peaks and valleys in the surface (given that the orientation of the exterior surface is known). So far we have taken the absolute value of \( \hat{D} \) for comparison with [38]. In the following, we make use of the additional sign information for the enhancement of the deviation map.
3.3 Topography enhancement

From Figure 4(c) we observe that the visual recognition and interpretation of surface topography is difficult from the deviation map. Further processing is necessary to facilitate the extraction of meaningful features from the maps. Peaks and valleys on the surface are important building blocks of the surface topography. We enhance patterns related to peaks and valleys to emphasize the topographic structure. Positive values in our deviation map indicate surface points below the smoothed surface (valleys) while negative values represent local peaks. For the enhancement of topographic structures we process patterns related to peaks and valleys separately which gives us a more comprehensive and flexible representation of surface topography. We first split the deviation map $D$ into two images $D_v$ and $D_p$ by:

$$D_v(x,y) = \max(D(x,y), 0)$$ (3)

$$D_p(x,y) = |\min(D(x,y), 0)|$$ (4)

$D_v$ contains the positive portion of $D$ which represents the map of the local valleys and $D_p$ captures the negative portion and provides the map of the local peaks.

The operations $\min(.)$ and $\max(.)$ for truncating the value range may introduce discontinuities in the maps that were not present in the original deviation map. To compensate for these introduced discontinuities, we locally smooth the image with a Gaussian filter $G$ as defined in Section 3.2 and obtain $D_v^s = D_v * G$ as well as $D_p^s = D_p * G$.

In areas where the surface has strong irregularities in depth (e.g. along cracks and crannies in the surface), we obtain comparatively high deviation values that lead to outliers in the value distribution of the deviation map (as well as in $D_v$ and $D_p$). The outliers bias the value range in the deviation maps and skew the distribution to the left (see Figure 5(b)). To obtain a more balanced distribution, we take the logarithm of the values in both maps: $E_v = \log(D_v^s)$ and $E_p = \log(D_p^s)$. Note that this operation can be safely performed in $\mathbb{R}$, since both maps contain only positive values due to the truncation and inversion in Equations 3 and 4. After logging the maps, the influence of the outliers is compensated and the distribution becomes approximately Gaussian (see Figure 5(d)). As a result the geometric micro-structures become better recognizable in the enhanced deviation maps $E_v$ and $E_p$. Figures 5(a) and 5(c) demonstrate the effect of logging. The enhanced deviation map (EDM) in Figure 5(c) better accentuates irregularities of the surface and improves the visibility of topographic details.

Figure 6 illustrates the topography extraction and enhancement on a different surface from our dataset. The point cloud viewed from projection direction together with its depth map is shown in Figures 6(a) and 6(b). The deviation map in Figure 6(c) compensates the global depth variations and reveals the micro-structure of the surface. The corresponding enhanced deviation map $E_v$ is shown in Figure 6(d). The map $E_v$ enhances patterns related to local valleys and thus highlights individual peck-marks. This makes the map $E_v$ a suitable basis representation for the segmentation of pecked surface areas.
Figure 5: Enhancement of topographic structures. The smoothed deviation map $D_v$ (a) shows a skewed value distribution (b). After enhancement the deviation map $E_v$ in (c) has a Gaussian-shaped distribution (d) and better reveals the micro-structures related to surface topography.

Figure 6: A 3D reconstruction of a rock surface in which the shape of a human figure has been engraved. (a) The colored point cloud from Fig. 7(a) viewed from projection direction; (b) The projected depth map; (c) The deviation map; (d) The deviation map with enhanced topographic structure emphasizes peck-marks that make up the engraving.
3.4 Classification

In order to provide a pixel-wise segmentation of the EDM $E^w$ we estimate the probability for each pixel as belonging to either foreground (pecked rock surface) or background (natural rock surface). The final image segmentation algorithm, which is described in Section 3.5, integrates this information to find an optimal segmentation which stays close to the classifier output and minimizes the boundary length of the segmentation, leading to smoother contours.

To obtain an estimate for each pixel we train a classification model that takes a local image patch around the pixel as input and outputs the desired foreground probability with respect to the learned model. In our basic configuration (referred to as “Plain” in Section 5.1) the pixel values of the input image (EDM) are taken directly as features. This means that for a given pixel, its pixel value together with the pixel values of its spatial neighbors are concatenated into a feature vector. Alternatively, other feature representations, such as Local Binary Patterns (LBP) [44] can be used as features (see our experiments in Section 5.1).

We rely on Random Forests [45, 46, 47] as a classification model. Random Forests have been employed for various tasks in different computer science communities, in particular in computer vision [48, 49]. Random Forests are fast to train and allow for iterative refinement which makes them well-suited for our task, where we first want to train a classifier offline on existing labeled data and then adapt the pre-trained model in an online fashion by data obtained from user interactions.

Training Random Forests: A Random Forest consists of a number of decision trees. Each decision tree is a hierarchical classifier that separates the input feature space into non-overlapping cells and assigns each cell a probability distribution over the class labels, foreground and background in our case. Given a set of training examples (image features describing the spatial neighborhood of locations in the input images, $E^w$ in our case, with the corresponding class labels), a decision tree is trained by recursively splitting the data into smaller parts such that one class becomes more and more dominant in the distribution. This objective can be formulated as a maximization of the information gain:

$$G(\mathcal{X}, \Theta) = H(\mathcal{X}) - \sum_{c \in \{l, r\}} |\mathcal{X}_c| H(\mathcal{X}_c). \quad (5)$$

This objective expresses the change in information entropy $H(.)$ when splitting the data into two sets, $\mathcal{X}_l$ and $\mathcal{X}_r$, which are assigned to two child nodes in the tree hierarchy. The splitting function $\Theta$, which typically involves a simple feature value comparison, is chosen to maximize the information gain. This optimization is typically done via a randomized grid search, where $F$ splitting tests are randomly sampled and the best is selected. Such a procedure has been proven effective to enforce the decorrelation between the individual trees in the forest [46]. The training/splitting procedure continues recursively until some stopping criteria are met. These criteria typically include a maximum tree depth or a minimum number of data samples per node. If one of the criteria is met, a leaf node is created and the class distribution is calculated given the training data falling into this leaf node.

Prediction with Random Forests: Given a set of $T$ fully trained decision trees, inference with Random Forests is highly efficient. For our task of semantic
segmentation, image features are extracted densely at each pixel and are routed from the root node to the corresponding leaf node by evaluating the learned splitting functions $\Theta$ along this path, which is done for all trees. Thus, an image feature ends up with $T$ probability distributions from the $T$ trees. These distributions are simply averaged over the differently trained decision trees to obtain the final prediction of the Random Forest for a single pixel in the input image.

3.5 Regularization of Segmentation

The output of the classifier described in the previous section is a pixel-wise probability of belonging to either foreground or background. The simplest way to turn that information into a binary segmentation would be to threshold the probability at some user-defined threshold. However, this usually yields a deficient result since the decision is made for each pixel individually and, thus, neglects the structure which is generally contained in natural images at this point. To make use of the underlying structures of natural images and to improve the segmentation result we cast the problem of segmenting an image into two disjoint regions as an energy minimization problem as defined in e.g. [40] of the form:

$$\min_{u \in [0,1]} \left\{ \frac{1}{2} \int_{\Omega} g(x) |\nabla u(x)| \, dx + \lambda \int_{\Omega} u(x) f(x) \, dx \right\},$$  

(6)

where $\Omega \subseteq \mathbb{R}^2$ is the image domain, $g(x)$ is an edge detector function and $f(x)$ represents the output of the classifier described above. The first term is called regularizer (or smoothness term) and ensures that the solution is piecewise constant, which means that small errors caused by the data term are corrected while sharp edges in the segmentation are retained. The weighting term $g(x)$ enforces the segmentation result to be aligned with dominant edges in the input image. The second term is called data term and ensures that the solution stays close to the output of the classifier. Factor $\lambda$ allows for steering the relative influence of the regularizer.

To minimize the objective we rely on a convex relaxation approach [50]. For efficiency reasons this is implemented on the graphics processing unit (GPU). The result of the minimization is an indicator function $u$, which is 1 for foreground regions and 0 for background. Fig. 13 shows an example result and compares the maximum-a-posteriori (MAP) result of the classifier with the result after regularization.

3.6 Iterative Refinement

The manual segmentation of rock-art is often subjective and, thus, may vary between users. Additionally, segmentation errors may remain in the automatic segmentation. Thus, a user probably wants to add corrections and modifications to the automated segmentation procedure. To take this into account our method provides the opportunity to refine the automatically generated results.

We provide two schemes of user interaction. On the one hand the user is able select additional training data for the classifier described in Section 3.4. This will affect the results on a more global level, i.e., user interaction in one
part of the image will likely affect the results on completely different locations of the image. This type of refinement enables, for example, the adaptation of the classifier to a novel pecking style. On the other hand, the user is also able to correct the results by making hard decisions. These hard decisions are integrated in the data term and will thus only affect the final regularization step. In both cases the user draws scribbles on the image and assigns a label, i.e., foreground or background to each of them.

In the first case these scribbles are used as new training data in order to adapt the classifier model online. To adapt the classifier we choose a rather straightforward approach. The training samples are traversed down the pre-trained trees in the forest and the class statistics at the leafs are updated, respectively. We opted for this simple procedure because it is extremely fast and offers a lot of flexibility.

In the second case the user’s scribbles are integrated in the data term by setting the probability for the respective pixels to zero or infinite and, thus, forcing the regularization to make specific decisions at these pixels.

4 Dataset and Experimental Setup

The Camonica valley in Brescia, Italy is a world heritage site since 1979 that provides one of the largest collections of rock-art in the world [1]. According to UNESCO more than 140.000 engravings can be found in the Camonica valley [51]. The rock-art employed in our work stems from different sites in the Camonica valley namely Serudina, Foppe di Nadro and Naquane.

4.1 Dataset

For the accurate representation of rock-art, high resolution 3D reconstructions are necessary that provide sufficient detail to capture the individual peck-marks that make up a petroglyph shape. We employed either structured light scanning (SLS) or structure from motion (SfM) and generated high resolution scans of rock surfaces in the three locations mentioned above. The selected areas show individual motifs (such as anthropomorphs, animals, abstract symbols) as well as entire scenes (e.g. hunting scenes). The shapes of the depicted figures are quite complex, consisting of fully pecked areas (inside of body) as well as thin structures (legs and antlers) as well as fine structures that were rather scratched than pecked (called “filiform”). Across different locations, we observe different pecking styles with different roughness and depth. Some areas are partly covered by moss which hides the actual rock surface and thus modifies the original surface topography. Hence, the moss impedes the subsequent surface analysis. The figures are partly incomplete, due to damages of the rock surface (e.g. erosion) and due to occlusions by other figures which were pecked on top of others at a later time. These factors impede interpretation and thus also segmentation.

As a result of 3D reconstruction we obtain large unstructured point clouds of several millions of points. An example for a point cloud with 4.4M points together with a close-up zoom into the cloud is shown in Fig. 7. For each 3D point a RGB color value is available. Point coordinates are converted to metric units (meters). The obtained resolution is in the majority of the reconstructions better than 0.1mm and thus sufficient to represent individual peck-marks. To
account for reconstruction artifacts, we apply smoothing and removal of outlier points prior to automated processing.

A detailed annotation of all reconstructions has been performed by domain experts who labeled all pecked areas in the surfaces. Figure 8 shows an example with an annotated hunter and a deer. Both figures are annotated, as well as individual peck marks in the surrounding and the bow represented as filiform. The detailed annotations enable both, the training and the objective quantitative evaluation of our segmentation method. For the annotation orthophotos have been generated from the point clouds by projecting the 3D point colors to a support plane (similarly to the depth map generation in Section 3.2).

The selected dataset for our evaluation contains of $N = 26$ 3D scans of rock surfaces from *Seradina, Foppe di Nadro* and *Naquane*. Each site represents a different rock from which several detail scans were acquired. All scans together consist of 115.3M 3D points. The number of points per scan ranges from 800K to 10.6M with an average of 4.4M. All surface scans in the dataset were completely labeled as either being an engraved area (class 1) or the natural rock surface (class 2). All individual peck-marks in the scans were labeled, even if they do not spatially coincide with an engraved figure. Class 1 represents 20.9% of the data and is thus underrepresented. Table 1 provides an overview of the captured data.
Table 1: Overview of the captured 3D data.

<table>
<thead>
<tr>
<th>Location</th>
<th>Rock</th>
<th>Area</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foppe di Nadro</td>
<td>24</td>
<td>7</td>
<td>Rosa Camuna</td>
<td>SLS</td>
</tr>
<tr>
<td>Foppe di Nadro</td>
<td>24</td>
<td>7</td>
<td>Warrior</td>
<td>SLS</td>
</tr>
<tr>
<td>Foppe di Nadro</td>
<td>24</td>
<td>9</td>
<td>Deer</td>
<td>SLS</td>
</tr>
<tr>
<td>Naquane Ossimo</td>
<td>8</td>
<td>11</td>
<td>Stele (part 1)</td>
<td>SLS</td>
</tr>
<tr>
<td>Naquane Ossimo</td>
<td>8</td>
<td>11</td>
<td>Stele (part 2)</td>
<td>SLS</td>
</tr>
<tr>
<td>Naquane Ossimo</td>
<td>8</td>
<td>11</td>
<td>Stele (part 3)</td>
<td>SLS</td>
</tr>
<tr>
<td>Naquane Ossimo</td>
<td>8</td>
<td>11</td>
<td>Stele (part 4)</td>
<td>SLS</td>
</tr>
<tr>
<td>Naquane</td>
<td>50</td>
<td>14</td>
<td>Standing Rider (part 1)</td>
<td>SFM</td>
</tr>
<tr>
<td>Naquane</td>
<td>50</td>
<td>14</td>
<td>Standing Rider (part 2)</td>
<td>SFM</td>
</tr>
<tr>
<td>Naquane</td>
<td>50</td>
<td>15</td>
<td>Sun-Shape Superimposition</td>
<td>SFM</td>
</tr>
<tr>
<td>Seradina 12C</td>
<td>1</td>
<td>Warrior Scene (part 1-1)</td>
<td>SLS</td>
<td></td>
</tr>
<tr>
<td>Seradina 12C</td>
<td>1</td>
<td>Warrior Scene (part 1-2)</td>
<td>SLS</td>
<td></td>
</tr>
<tr>
<td>Seradina 12C</td>
<td>1</td>
<td>Warrior Scene (part 1-3)</td>
<td>SLS</td>
<td></td>
</tr>
<tr>
<td>Seradina 12C</td>
<td>1</td>
<td>Warrior Scene (part 1-4)</td>
<td>SLS</td>
<td></td>
</tr>
<tr>
<td>Seradina 12C</td>
<td>1</td>
<td>Warrior Scene (part 2-2)</td>
<td>SLS</td>
<td></td>
</tr>
<tr>
<td>Seradina 12C</td>
<td>1</td>
<td>Warrior Scene (part 2-3)</td>
<td>SLS</td>
<td></td>
</tr>
<tr>
<td>Seradina 12C</td>
<td>1</td>
<td>Warrior Scene (part 2-4)</td>
<td>SLS</td>
<td></td>
</tr>
<tr>
<td>Seradina 12C</td>
<td>1</td>
<td>Warrior Scene (part 3-1)</td>
<td>SLS</td>
<td></td>
</tr>
<tr>
<td>Seradina 12C</td>
<td>1</td>
<td>Warrior Scene (part 3-2)</td>
<td>SLS</td>
<td></td>
</tr>
<tr>
<td>Seradina 12C</td>
<td>1</td>
<td>Warrior Scene (part 3-3)</td>
<td>SLS</td>
<td></td>
</tr>
<tr>
<td>Seradina 12C</td>
<td>4</td>
<td>Hunter With Bow</td>
<td>SLS</td>
<td></td>
</tr>
<tr>
<td>Seradina 12C</td>
<td>5</td>
<td>Hunter With Speer (part 1)</td>
<td>SLS</td>
<td></td>
</tr>
<tr>
<td>Seradina 12C</td>
<td>5</td>
<td>Hunter With Speer (part 2)</td>
<td>SLS</td>
<td></td>
</tr>
<tr>
<td>Seradina 12C</td>
<td>10</td>
<td>Hunting Scene (part 1)</td>
<td>SLS</td>
<td></td>
</tr>
<tr>
<td>Seradina 12C</td>
<td>10</td>
<td>Hunting Scene (part 2)</td>
<td>SLS</td>
<td></td>
</tr>
<tr>
<td>Seradina 12C</td>
<td>10</td>
<td>Hunting Scene (part 3)</td>
<td>SLS</td>
<td></td>
</tr>
</tbody>
</table>

4.2 Experimental Setup

Our evaluation covers different aspects: (i) we evaluate the performance differences between a full 3D approach (based on depth information) and the traditional 2D approach based on color photos as in Seidl & Breiteneder [10]; (ii) we investigate the influence of the amount of training data on segmentation performance (generalization ability). For both investigations we use results obtained by the fully automatic segmentation to obtain comparable and objective results. (iii) We also simulate interactive refinements and measure the impact on the segmentation performance, (iv) we investigate the benefit of regularization of the segmentation output compared to pure pixel-based classification, and (v) we evaluate the runtime of our approach. Ultimately, (vi) we present qualitative results to demonstrate the capabilities of our method.

If not mentioned otherwise, we used the following settings throughout all experiments. We employ $k$-fold cross-validation where $k$, the number of folds, is 4. This means that the dataset of $N = 26$ scans is divided randomly into $k$ approximately equally sized subsets. Next, we run $k$ evaluations, where each subset is used once as test set while the remaining subsets serve as training set. Finally, the results of all $k$ evaluations are averaged to obtain an estimate of the overall accuracy. This procedure is also used to fix model parameters like the size of the spatial neighborhood, which is considered to classify individual pixels, or the maximum tree depth for the Random Forest.

We use a standard patch-based approach for learning the Random Forest
model [52, 53], where each training sample (pixel) is represented by a square patch (pixel neighborhood). In our case the patches are approximately of size 10 × 10 mm and sampled from the plain enhanced deviation map (EDM) $E^v$ in the standard configuration of our approach. For comparison we employ other representations than the EDM, such as the depth map or a color image (orthophoto) as baselines.

For training, we randomly sample 4000 patches per class from each scan in the training set. For the entire training set (according to our cross-validation protocol) we obtain approx. 152K training patches to learn our Random Forest model. We train ensembles with 10 trees and stop training if less than 5 samples arrive in a node of a tree or a maximum depth of 18 is reached. The number of sampled tests $F_s$ for optimization of the splitting functions depends on the overall number of possible tests $F$. We set $F_s = \sqrt{F}$. For the final segmentation we set the parameter $\lambda$, which trades off the data term and the smoothness term, to 0.6.

To measure the segmentation quality we employ the Dice Similarity Coefficient (DSC) which is a standard measure for the evaluation of segmentation approaches. DSC measures the mutual overlap between the automatic segmentation $X$ and a manual labeling $Y$:

$$DSC(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|}.$$  \hfill (7)

DSC is between 0 and 1 and 1 means a perfect segmentation.

To provide some more detailed insights we also compute the hit rate (HR) and the false acceptance rate (FAR). The hit rate measures the number of correctly classified foreground pixels:

$$HR(X, Y) = \frac{|X \cap Y|}{|X \cap Y| + |Y \setminus X|}.$$  \hfill (8)

The false acceptance rate, on the other hand, measures the amount of pixels incorrectly classified as foreground:

$$FAR(X, Y) = \frac{|X \setminus Y|}{|X \cap Y| + |X \setminus Y|}.$$  \hfill (9)

Hit rate and false acceptance rate both range between 0 and 1. However, note that for the hit rate a higher value is better, whereas for the false acceptance rate a lower value is preferred. In practice hit rate and false acceptance rate are correlated. An improvement of the hit rate is often accompanied by an increased number of false detections and vice versa.

5 Results

To assess the performance of our method we conducted quantitative and qualitative evaluations on the material from Section 4.

5.1 Quantitative Results

In this section we present results of all evaluated aspects by utilizing the dataset described in Section 4.2. We start by showing the influence of 3D information
Table 2: Quantitative results for different setups, comparing the capabilities of color (2D) and depth (3D) information. The results are depicted for the individual sites from which the scans originate as well as over all scans in our dataset.

<table>
<thead>
<tr>
<th>Representation</th>
<th>Seradina DSC</th>
<th>Serpe di Nado DSC</th>
<th>Naquane DSC</th>
<th>Average (over scans)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color (Plain)</td>
<td>0.394</td>
<td>0.363</td>
<td>0.388</td>
<td>0.546 0.665 0.389</td>
</tr>
<tr>
<td>Depth Map (Plain)</td>
<td>0.581</td>
<td>0.747</td>
<td>0.543</td>
<td>0.817 0.497 0.590</td>
</tr>
<tr>
<td>EDM (Plain)</td>
<td>0.567</td>
<td>0.758</td>
<td>0.619</td>
<td>0.854 0.510 0.603</td>
</tr>
<tr>
<td>Color (LBP)</td>
<td>0.346</td>
<td>0.373</td>
<td>0.372</td>
<td>0.516 0.702 0.356</td>
</tr>
<tr>
<td>EDM (LBP)</td>
<td>0.556</td>
<td>0.753</td>
<td>0.611</td>
<td>0.819 0.509 0.594</td>
</tr>
</tbody>
</table>

on the segmentation result. After that we look at the generalization ability to new, unseen scenes and show the influence of interactivity and regularization of the segmentation.

5.1.1 2D vs. 3D Information

First, we investigate whether the acquired 3D information proves useful for segmentation. We compare the segmentation results when using color information (orthophotos) with that of a full 3D approach incorporating depth information. To start with, we learned models solely based on either color or depth information. The results are shown in rows 1 and 2 of Table 2.

We observe that using color information the results are relatively poor, which is reflected by an average Dice coefficient of about 0.39. However, by employing depth information (from the depth map) a relative improvement of about 50% is achieved, yielding an overall Dice coefficient of 0.59. For all investigated locations, the use of depth information strongly improves the average performance. The performance gain is reflected in particular in the hit rate which improves from 0.55 for 2D information to 0.82 for 3D information. This shows that much more peaked areas can be labeled correctly when using 3D information. At the same time the number of false detections is reduced (FAR reduces from 0.67 to 0.5), which demonstrates that the overall segmentation performance is increased.

The average performance is further improved when applying the enhanced deviation map (EDM) as input representation for segmentation as proposed by our approach (see row 3 in Table 2). The HR improves by 3.7% and the FAR improves by 1.3%. The result is an overall improvement of segmentation performance, which is reflected by the increase of DSC of 1.3%. The results of these experiments show (i) that 3D outperforms 2D information and (ii) that the enhancement of the depth map is beneficial for segmentation.

Additionally, to taking color image, depth map, and EDM directly as input, we extract texture features (Local Binary Patterns, LBP as proposed by Ojala & Pietikäinen [36]) from the representations and use them as input for segmentation instead. Rows 4 and 5 in Table 2 show the corresponding results. We observe that although the use of texture features improves performance on some of our scans, overall performance cannot increase from using LBP features. At first sight this might seem surprising since texture represents an important cue for discriminating areas with different surface geometries. From
Figure 9: Segmentation performance as a function of training set size for two input representations: the depth map and the enhanced deviation map. Using a large number of training samples our method performs similar for both input representations. When the number of training samples is small, the enhancement of the depth map strongly improves results. Shaded regions represent the standard deviation of the results.

our experiments, however, we observe that the Random Forest together with the patch-based approach sufficiently captures texture information. This is indeed a well known property of this approach and was also exploited in prominent prior work by Shotton et al. [53].

Note that, since we were focusing on different representations which mainly affect the classifier, we skipped the regularization of the segmentation for these experiments. Instead, we directly used the maximum-a-posteriori (MAP) estimate of the Random Forest classifier to label each pixel as either belonging to a pecked or non-pecked area, i.e. the classifier itself decided which label to apply to each pixel.

5.1.2 Generalization Ability

Next, we investigate the generalization ability of our method. The goal is to gain further insights about the amount of necessary manual labeling effort needed for the classifier to learn a reasonable model of foreground and background. For this purpose we split our dataset into differently sized training and test sets, i.e. we randomly sample a number of $m$ surface scans from our dataset of $N$ scans, train a model on randomly selected samples of the $m$ scans and evaluate it on the remaining $N - m$ scans. We iterate this procedure for different sample sizes $m$, focusing on small $m \ll N$. Figure 9 shows the results as a function of $m$, where $m \in \{1, 2, \ldots, \frac{N}{2}\}$. To avoid a bias coming from a particular sample selection, we run these experiments 15 times by sampling different random subsets of the scans for each distinct $m$. The final results shown in Fig. 9 are averaged over all runs together with the standard deviation across all 15 runs.

We observe that the topography enhancement described in Section 3.3 strongly improves generalization ability to unseen images when the training set is small. The average segmentation performance is almost independent of the size...
Table 3: Comparison of Dice coefficients for cross-validation over different locations. Quantitative results obtained for scans from Seradina when the classifier is trained on scans of only Foppe di Nadro and Naquane, as well as results for scans from Foppe di Nadro and Naquane when the classifier is trained only from scans of Seradina.

<table>
<thead>
<tr>
<th>Training Set:</th>
<th>Test Set:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seradina</td>
<td>Seradina</td>
</tr>
<tr>
<td>Foppe di N. &amp; Naquane</td>
<td>Foppe di N.</td>
</tr>
<tr>
<td>Seradina</td>
<td>Naquane</td>
</tr>
<tr>
<td>Average (over scans)</td>
<td>HR</td>
</tr>
<tr>
<td>Depth Map</td>
<td>0.547</td>
</tr>
<tr>
<td>EDM</td>
<td>0.558</td>
</tr>
</tbody>
</table>

of the training set as shown by the progression of the solid curve in Fig. 9. Without enhancement (dashed line) the method requires a much higher number of training samples to achieve similar results. Over all experiments, the average performance of our method with EDM always outperforms that of the depth map (independent from the size of the training set). Even with only one training image the EDM-based approach outperforms all results obtained with the depth map. These experiments represent a strong argument in favor of our method. Especially, since one of the main rationales behind designing and implementing a system for automatic segmentation of petroglyphs is to reduce tedious manual labeling effort, the ability to generalize from a small amount of training data is crucial.

Another question in this context is whether models learned from scans of one site can generalize to other sites. This is especially interesting since rock-art at different sites may exhibit different peck styles, e.g. because different tools were used for engraving. To answer this question we separated the dataset into two sets, one containing the scans from Seradina and the other one the scans from Foppe di Nadro and Naquane. The two latter locations were joined because they are neighboring locations and their petroglyphs are of similar style. Next, we trained our model solely on the scans from one of the two sets and evaluated it on the other, and vice-versa, resulting in three experiments:

- Training on data from Foppe di Nadro plus Naquane and testing on Seradina: column 2 in Table 3
- Training on data from Seradina and testing on Foppe di Nadro: column 3 in Table 3
- Training on data from Seradina and testing on Naquane: column 4 in Table 3

We observe that automatic segmentation on the basis of depth maps does not generalize well. The average performance drops from 59% (see Table 2) when data from all sites is used for training to only 56.5%. The performance difference for the EDM, however, is negligible (DCE of 59.5% vs. 60.3% when data from all sites are used). HR decreases by 1.7%, while FAR stays at nearly the same level (+0.4%). These results confirm that the enhancement of the depth map is important for the generalization ability of rock-art segmentation and that our approach has the potential to generalize well to new and previously unseen locations.
Table 4: Results for simulation of interactive segmentation. We provide the Dice coefficients for the individual sites as well as over the entire dataset. The simulated user input improves segmentation performance.

<table>
<thead>
<tr>
<th>Setup</th>
<th>Seradina DSC</th>
<th>Foppe di Nadro DSC</th>
<th>Naquane DSC</th>
<th>Average (over scans)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No interaction</td>
<td>0.567</td>
<td>0.758</td>
<td>0.619</td>
<td>0.854 0.510 0.603</td>
</tr>
<tr>
<td>Simulated interaction</td>
<td>0.604</td>
<td>0.810</td>
<td>0.636</td>
<td>0.878 0.481 0.637</td>
</tr>
</tbody>
</table>

5.1.3 Interactive Segmentation

Objective evaluations of interactive methods are difficult. The two main issues are that (i) a large user study with a significant number of participants needs to be performed to obtain meaningful results, and (ii) the metrics to measure the outcome of such a study often leaves room for interpretation. For this work a user study was out of scope. To get an impression of the potential of our interactive segmentation method, we approach to simulate user interactions. We randomly select a small number of samples from the manual labeling of a given scan and provide it as additional “user input” to our method. Using this information as input we can re-train the classifier, apply it to the same scan and measure the obtained performance difference. Please note, however, that such an evaluation is limited in expressiveness and does not replace a user study, which is considered future work.

For this experiment we sampled 500 pixels for each class per scan, which is a relatively small number considering that the average number of pixels in the projected images is about 29 million. Table 4 shows that our method yields better results with additional clues for segmentation. The averaged DCE improves by +2.4%, HR improves by +2.4%, and the false acceptance rate decreases by 2.9%.

5.1.4 Regularization

In a final experiment we evaluate the effect of regularization (see Section 3.5) on the segmentation performance. As a baseline we evaluate the plain output of the classifier (without interaction), where each pixel is labeled with the class exhibiting the maximum likelihood. Next, we run the regularization to find a coherent segmentation, which indeed minimizes the boundary length of the detected foreground regions. Results in Table 5 show that the regularization consistently improves the results by a small margin (DCE +1.5%).

The regularization favors larger coherent parts and tends to remove fine-grained details. As a consequence the portion of correctly classified foreground pixels (HR) may decline (as in our experiments where the HR decreases by 1.6%). At the same time, the number of falsely classified background pixels decreases as well (-2.2% in our experiments). As the background class is usually much larger than the foreground class, the overall benefit is positive. This is reflected by the improvement of the average performance in terms of DCE.

The regularization of the segmentation has, however, further implications on the practical work. The removal of fine-grained details may induce additional refinement work by a user because the refinement of the fine details is usually more tedious and time-consuming than adding or removing larger parts. To give
Table 5: Comparison of Dice coefficients with and without regularization of the segmentation. Regularization slightly but consistently improves the segmentation for all sites and in average.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Seradina DSC</th>
<th>Foppe di Nadro DSC</th>
<th>Naquane DSC</th>
<th>Average (over scans) HR</th>
<th>FAR</th>
<th>DSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Regularized</td>
<td>0.567</td>
<td>0.758</td>
<td>0.619</td>
<td>0.854</td>
<td>0.510</td>
<td>0.603</td>
</tr>
<tr>
<td>Regularized</td>
<td>0.580</td>
<td>0.781</td>
<td>0.635</td>
<td>0.838</td>
<td>0.489</td>
<td>0.618</td>
</tr>
</tbody>
</table>

the user more control over the segmentation result, we allow the user to steer the regularization by adapting the $\lambda$ parameter, which trades-off data fidelity and cohesiveness. See Fig. 15 in Section 5.2 for visual examples that demonstrate the effect of varying parameter $\lambda$.

5.1.5 Runtime

Semi-automatic segmentation methods like the one proposed in this work require interaction between user and algorithm. Therefore, the response time to new user input should be minimal. All components of our method are implemented using modern parallel computing architectures provided by graphics cards. The runtime of our method is linear with the number of pixels in the input image. For a 1 MP image, user feedback can be provided within 600 ms, with feature extraction and classification contributing 100 ms and regularization 500 ms. This runtime analysis excludes the projection of the point cloud to a depth map and the creation of the EDM which has to be done only once.

5.2 Qualitative Results and Iterative Refinement

In this section we present qualitative results to complement our quantitative evaluation and provide further insights into the proposed method. We show the scans and the corresponding results as they were captured, i.e. without changing their orientation. Thus, the petroglyphs might appear in arbitrary orientations. Although for some scans the intended orientation is relatively easy to recognize for humans, for most pecked regions a unique orientation does not exist, e.g., in the case of superimposition of several figures and for pecked symbols. Our intention is show the scans without adding an additional interpretation by rotation.

In Fig. 10 we show results obtained using purely automatic segmentation without any user interaction. We compare the binarized results when using either the raw depth map (DM) or the enhanced deviation map (EDM). Note, that here we show the unregularized maximum-a-posteriori (MAP) result from the classifier to put focus exclusively on the comparison between the DM and EDM, i.e. to remove possible influences from regularization. The results are shown together with the corresponding input depth map and the manual labeling from an individual expert. Additionally, we give the corresponding scores (DSC,HR,FAR) for each of the segmentations in the figure captions. Comparing these scores for different results on the same image shall provide a better un-

---

1 All timings have been measured on a recent Linux computer using a Nvidia GeForce GTX780 GPU.
derstanding of these scores. We observe that using the EDM does not improve the false acceptance rate (FAR), but rather the method is able to find much more of the regions we are interested in. Therefore, the hit rate (HR) increases by nearly 30%, yielding an improvement of more than 15% in terms of the Dice Similarity Coefficient (DSC). From Figures 10(c) and 10(d) we further observe that the EDM facilitates the dense segmentation of larger pecked areas (e.g., the interior of the horse). The falsely detected areas in both images, however, seem to correspond to a large extent.

As pointed out above our method tries to align the segmentation boundaries with gradients in the depth map. This is accomplished by considering the gradient magnitudes of the depth map in the regularization term of the energy minimization problem in Eq. 6. To inspect the results in detail Fig. 11 shows close-up views of the segmentation result as well as the corresponding gradient magnitudes and classifier output which are considered in the regularization. Note that – despite the rough and noisy information delivered by the classifier – the method produces a coherent segmentation, while aligning the segmentation border with gradients in the depth map. Since experts usually decide in a similar manner where to place the borders of the tracing, this way the method reduces the necessary manual interaction.

The results demonstrate that the proposed automatic approach alone is already able to aid archaeological experts. The method is able to largely reduce the manual tracing effort but can also point the experts to regions, which might have been missed otherwise. Nevertheless, due to the small amount of labeled training data compared to the large variety of targeted rocks and peckings, the method still fails to provide a clean result for some scans. We show one such example in Fig. 12. The method is not able to correctly classify large parts of the un-pecked rock surface. In such cases the possibility to interactively refine the segmentation is vital.

Figs. 13 and 14 illustrate an example interactive segmentation workflow on a single input scan, where the user refines the result until she is satisfied. The pointcloud is shown together with its depth map in Figs. 13(a) and 13(b). The fully automatically generated result from the pixel-wise classifier is shown in 13(c). The final output of the automatic segmentation is the regularized segmentation result in Fig. 13(d).

After the fully automatic segmentation the user can either keep this segmentation or apply an arbitrary number of additional refinement steps. For refinement the user draws scribbles onto the segmentation result (red scribbles for falsely classified background regions and green scribbles for falsely classified foreground regions). Example scribbles are shown in Figs. 14(a) and 14(c) for two rounds of iteration. The respective segmentation results are depicted in Figs. 14(b) and 14(d).

Finally, Fig. 15 shows some more results for different rock surfaces. Here, we also show the probabilistic map from the classifier, an intermediate output of our method. For illustration we used different settings for $\lambda$ for the different results. The impact (higher $\lambda$ yields more coherent and smoother segmentations) is clearly visible. See the figure captions for details. A comparison of the final segmentations with manual labeling shows that the proposed method achieves good segmentation results.
Figure 10: Comparison between the binarized segmentation result based on the raw depth map and the enhanced deviation map. Given only the raw depth measurements, the model learned from training data cannot generalize to this test image. However, using the enhanced deviation map the pecked region is segmented much better. Here, we show the unregularized maximum-a-posteriori (MAP) output from the classifier in order not to distort the comparison. In the depth maps brighter pixel values mean higher distance from the camera. Images are cropped for better visibility.
Figure 11: Illustration of segmentation along depth discontinuities. We show close up views of the segmentation result and intermediate processing steps for the rendered orthophoto (a) and the corresponding segmentation (b). The blue rectangles mark the region shown as close-ups in (c-f). For the close-up view of (a) shown in (c) we depict the corresponding gradient magnitudes (d) and the classifier output (e). Darker values in (d) represent stronger gradients and brighter values in (e) represent stronger foreground likelihoods. Both data sources are used for the regularization ($g(x)$ and $f(x)$, resp., in Eq. 6). (f) shows a close-up of the segmentation result (b). The method tries to deliver a coherent segmentation based on the classifier output, while aligning the segmentation borders with the gradients in the depth map.
Figure 12: Failure case of the automatic segmentation. For this example the specific appearance of the raw rock surface, together with the specific kind of pecked structures and weathering influences (erosion) make it extremely difficult for the automatic segmentation approach to provide a clean result. While the regularization helps to get rid of some false positives in the background, still significant user input is required to correct segmentation errors. Note that from the orthophoto (a) and the enhanced deviation map (b) it becomes evident that this reconstruction represents a particularly difficult case as the pecked surface areas are hardly visible to the unaided eye.
Figure 13: Orthophoto of the point cloud (a), the corresponding depth map (b) and results obtained from the raw output of the classifier (c), i.e., the maximum-a-posteriori (MAP) estimate, and after regularization (d) with $\lambda = 0.7$. See Fig. 14 for the effect of interaction based on this result.
Figure 14: Example results for interactive refinement. Based on the results shown in Fig. 13, the user draws scribbles to mark incorrectly labeled regions (a). We use red scribbles to mark regions which should be labeled as background but are labeled as foreground (i.e., false positives) and green scribbles to mark false negatives, respectively. The refined result after incorporating the user input is shown in (b). Such kind of user interaction can be repeated until the user is satisfied. For instance, (c) and (d) show another round of user interaction.
Figure 15: Two segmentation examples with intermediate processing steps: (a)-(f) and (g)-(l). For the refined segmentations, we superimposed the user input (scribbles). For comparison we set $\lambda = 0.6$ for the first example (e) and (f), and $\lambda = 0.1$ for the second example in (k) and (l). Decreasing $\lambda$ yields smoother segmentations with a higher likelihood of merging larger regions.
6 Conclusion

In this article we have presented a novel method for the segmentation of petro-
glyph shapes from 3D reconstructions. The approach first extracts and enhances
the geometric micro-structure of the reconstructed surface efficiently in image-
space. The enhanced representation is used as input to a pixel-wise classification
scheme, which employs an automatically learned model to assign a probability
to each pixel that specifies whether it belongs to a pecked region or not. Next,
we optimize the contour of the segmented shapes by a gradient preserving regular-
ization which provides contours that better match the expectation of human
experts. To enable refinements and the correction of errors, we provide an in-
tuitive method for interactive refinement based on scribbles which further lets
the method learn and improve future segmentations. Our evaluation shows that
the proposed method yields accurate segmentations over a large dataset of 3D
surfaces. The method clearly outperforms 2D color-based segmentation and
yields a high generalization ability even with a very limited amount of training
data. Due to the fact that our method can iteratively be re-trained with novel
data it can easily be adapted to new sites with peckings of different topogra-
phy. A major challenge that remains (that is inherent to the task) is the strong
heterogeneity of manual interpretations for rock-art segmentation. This means
that there is usually no single unambiguous solution for the segmentation of a
figure or scene. To account for this fact we have integrated the possibility to
interactively refine the segmentations. Thus each segmentation can be fitted to
the interpretation and the perception of the current user.

From our experiments we conclude that the proposed segmentation method
has the potential to bridge the gap that exists between 3D reconstruction of
rock surfaces and higher-level shape analysis of petroglyphs. Thus, we expect
the proposed method to facilitate the documentation of rock-art in the future.

Acknowledgment The work for this paper has been carried out in the project
3D-Pitoti which is funded from the European Community’s Seventh Framework
Programme (FP7/2007-2013) under grant agreement no 600545; 2013-2016. We
thank Alberto Marretta for providing image material and Erich Kobler for the
initial prototype of the graphical user interface.
References


