

A study on topological descriptors for the analysis of 3D surface texture

Matthias Zeppelzauer^a, Bartosz Zieliński^b, Mateusz Juda^b, Markus Seidl^a

^aMedia Computing Group, Institute of Creative Media Technologies,
St. Pölten University of Applied Sciences,
Matthias-Corvinus Strasse 15, 3100 St. Pölten, Austria

^bThe Institute of Computer Science and Computer Mathematics,
Faculty of Mathematics and Computer Science, Jagiellonian University,
ul. Łojasiewicza 6, 30-348 Kraków, Poland

Abstract

Methods from computational topology are becoming more and more popular in computer vision and have shown to improve the state-of-the-art in several tasks. In this paper, we investigate the applicability of topological descriptors in the context of 3D surface analysis for the classification of different surface textures. We present a comprehensive study on topological descriptors, investigate their robustness and expressiveness and compare them with state-of-the-art methods including Convolutional Neural Networks (CNNs). Results show that class-specific information is reflected well in topological descriptors. The investigated descriptors can directly compete with non-topological descriptors and capture complementary information. As a consequence they improve the state-of-the-art when combined with non-topological descriptors.

Keywords: Surface Texture Analysis, 3D Surface Classification, Surface Topology Analysis, Surface Representation, Persistent Homology, Persistence Diagram, Persistence Image

1. Introduction

In recent years, methods for sparse and dense reconstruction of 3D scenes have progressed strongly due to the availability of inexpensive off-the-shelf hardware like Microsoft Kinect and the development of robust 3D reconstruction algorithms (e.g. structure from motion techniques, SfM) [1, 2]. Thus, the amount of available 3D data is rising constantly. Furthermore, the reconstruction accuracy is increasing strongly, which enables 3D reconstructions with sub-millimeter resolution [3]. The high resolution enables the accurate description of the geometric micro-structure of a surface, which opens up new opportunities for search and retrieval in 3D scenes, such as the recognition of objects by their specific surface properties as well as the distinction of different types of materials for improved scene understanding.

The geometric micro-structure determines the haptic appearance of a surface (e.g. in terms of roughness, waviness, and lay) and is also referred to as *surface texture*¹ [5, 6, 7]. The geometric structure of a surface is closely related to its topology and thus, topological descriptors are promising candidates for its representation. In this article, we investigate the capabilities of topological analysis of describing and classifying 3D surfaces according to their geometric micro-structure and present a first extensive study on topological descriptors in this context.

Email addresses: m.zeppelzauer@fhstp.ac.at (Matthias Zeppelzauer), bartosz.zielinski@uj.edu.pl (Bartosz Zieliński)

¹Note that surface texture is different from the *apparent texture* or *visual texture* of a surface which refers to the visual (radiometric) appearance of the surface in terms of color variations, brightness, and reflectivity [4].

For our study we employ high-resolution 3D reconstructions of natural rock surfaces from the archaeological domain, see Fig. 1a which exhibit different surface textures. We map the surfaces to depth maps (Fig. 1b) [8] and analyze the maps in a patch-wise manner. For each surface patch we extract topological descriptors and traditional non-topological descriptors. Next, we train a classifier and try to predict the class of texture from each patch. The result is a class label for each surface patch. We employ precise maps with ground-truth labels (see Fig. 1c) to evaluate the classification accuracy.

This article is an extended version of the work in [9] where we reported first preliminary results for topological surface texture analysis. This article significantly extends this work to a comprehensive and in-depth study on topological descriptors for 3D surface analysis. We evaluate different extensions and variants to the proposed topological descriptors and study in detail their robustness and discriminativity. In particular the contribution of this paper over the previous one is an investigation of: (i) the sensitivity of the topological descriptors to their computational parameters; (ii) the influence of different normalization variants; (iii) the role of outliers in persistence diagrams [10]; (iv) the extension of topological descriptors with additional pre-filtering operations (v) the robustness and the explanatory power of the descriptors and (vi) strategies to reduce the dimension of the descriptors' feature vectors. Moreover, we verify the validity of our results on an extended large-scale dataset [11]. To provide the complete picture of our analysis we include the results from [9] in this paper.

Our study reveals that topological descriptors are robust 3D surface descriptors which can compete and partly outperform

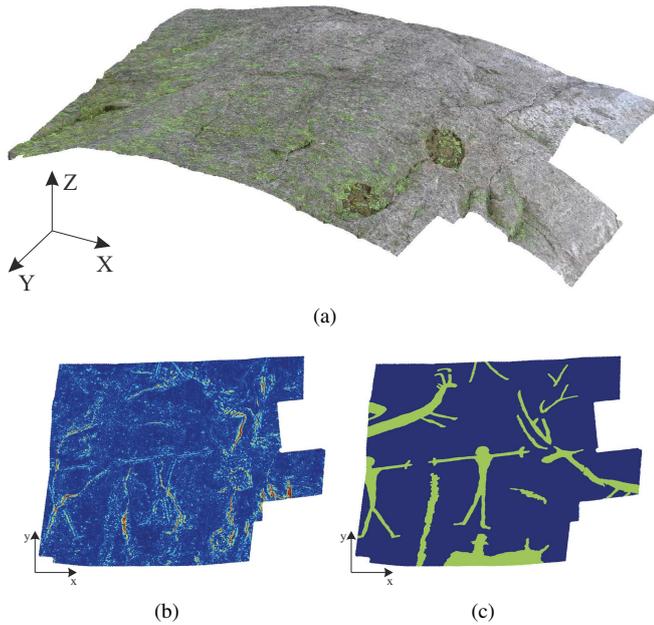


Figure 1: 3D surface data: (a) the 3D point cloud of the surface; (b) the depth projection of the surface; (c) ground truth labeling that specifies areas with different surface texture (blue and green).

traditional (non-topological) descriptors. Moreover our investigations clearly show that topological descriptors capture complementary information to non-topological descriptors and that this information is *necessary* to obtain peak performance in classification. Remarkably, our results on the extended dataset show that our combinations of topological and non-topological descriptors yield a similar performance level (and partly even higher performance) as deep learning in [11].

The remaining article is organized as follows. In Section 2 we review related work on 3D surface analysis and relevant topological approaches in this context. Section 3 introduces the investigated topological descriptors together with their properties and outlines our topological approach for surface texture analysis. Section 4 summarizes the experimental setup and Section 5 presents the study on topological descriptors together with the obtained results. We summarize the major conclusions of our study in Section 6.

2. Related work

Approaches for 3D surface analysis can be partitioned into two classes: methods that operate in 3D space directly and methods that operate on derived image-space representations. The following sections review related work on both approaches.

2.1. Surface analysis in 3D space

The natural domain for the analysis of surfaces is the 3D domain. A popular approach is to analyze the mesh or point cloud of a given surface reconstruction with local 3D surface descriptors that describe the local geometry around a 3D point. Numerous descriptors have been proposed for this purpose in recent

years. Johnson and Hebert propose *spin images* as local descriptors for the dense description of meshes for object recognition in 3D scenes [12]. Darom and Keller extract SIFT features from local depth images of a points' neighborhood to model the surface geometry around it [13]. Zaharescu et al. introduce a local surface descriptor for meshes (MeshHOG) that can be computed from an arbitrary scalar function (e.g. curvature) defined over the surface [14]. Steder et al. propose the *normal-aligned radial feature* (NARF) which captures local depth variations in range data [15]. Frome et al. extend the well-known 2D *shape context* descriptor [16] to 3D point clouds (3DSC) and show that it outperforms spin images [17]. Rusu et al. propose two local 3D point cloud descriptors, namely *persistent point feature histogram* (PFH) [18] and an accelerated version *fast PFH* (FPFH) [19]. Both build upon the relations between *surfels*, i.e. the combination of a point and its surface normal to describe the surface geometry [20]. Tombari et al. propose a 3D descriptor (SHOT) based on the point normals that is defined in a robust local reference frame [21].

Local descriptors are particularly well-suited for the *sparse* description of a surface, i.e. descriptors are computed at salient feature points only. The analysis of 3D surfaces by their surface texture - as investigated in this paper - requires the *dense* analysis and representation of a surface. Previous experiments have shown that local 3D descriptors are impractical for this purpose, especially due to computational reasons, because descriptors have to be computed at each surface point to obtain a dense representation, which becomes highly demanding when a point cloud has several millions of points [8].

2.2. Surface analysis in image-space

To circumvent the problems raised by 3D descriptors in the context of dense surface analysis, Othmani et al. propose to map 3D surfaces to image-space and then apply efficient image analysis techniques there to process the projected surfaces [22]. Usually a 2D depth map is derived from the 3D surface (piecewise or globally, depending on the amount of curvature of the surface) to obtain an image-space representation. If the surface does not contain self-occlusions, the depth map completely represents the 3D surface, except for inaccuracies or losses in resolution due to rasterization [8].

The analysis of 3D surface texture in image-space can be considered as texture analysis task on the depth map. Thus, approaches of image texture analysis become applicable. Popular methods for image texture analysis include histograms of vector quantized filter responses [23] and later generalizations such as the bag-of-visual-words model for textures [24] and the Fisher vector [25]. Recently, deep learning-based approaches for texture analysis have been introduced for the problem of image texture analysis [26], which outperform many existing methods.

An alternative approach to texture analysis (both for image texture and surface texture) is to consider the underlying data (depth map in our case) as a geometric object and to analyze its topological properties. For this purpose *topological data analysis* (TDA) is a promising approach which currently gains increasing importance in data science. TDA provides methods

to study qualitative properties of objects based on the theory of size functions introduced by Frosini et al. [27, 28] and persistent homology by Edelsbrunner et al. [29]. TDA methods have recently been successfully applied to computer vision problems, such as shape and texture analysis. Li et al. combine the bag-of-features approach with persistence diagrams (PDs) for shape retrieval [30]. Reininghaus et al. proposes an SVM kernel for persistence diagrams to make them compatible with statistical machine learning methods [31]. The authors apply topological descriptors together with the novel kernel to shape retrieval and texture classification in images. Another approach to leverage topological information for machine learning tasks is the persistent image (PI) proposed by Adams et al. [32]. The PI is a vector-based representation of PD which is built upon earlier work on size functions [33, 34]. The PI can be employed directly as a feature vector in conventional machine learning techniques and can easily be combined with other descriptors.

Our work mainly builds upon the approach of Adams et al. In our study we investigate different topological descriptors, including PI, for the domain of surface texture analysis. The difference to the investigations in [31] and [32] is that we try to enrich the PI representation by pre-filtering, normalization, and feature selection and that we analyze in-depth the information captured by the PI descriptor, its redundancy and discriminativity. Furthermore, we investigate the sensitivity of the representation to its computational parameters. Beyond this we combine them with non-topological state-of-the-art descriptors to investigate synergy effects.

3. Topological approach

3.1. Overview

Fig. 2 provides an overview of the approach that we propose for the analysis of 3D surfaces by persistence homology features. The input 3D surfaces are first projected to image-space to obtain depth maps as proposed in [8]. Next, depth maps are z-standardized (see Section 3.5) and split into square patches. Patches are normalized (optional step, see Section 3.5) and pre-filtered by Schmid or MR filters or CLBP (optional step, see Section 3.4). Next, topological descriptors are extracted as described in Sections 3.2 and 3.3. Finally feature selection is performed (optionally) before patch-wise classification to decrease the dimensionality of the feature vector and to reduce redundancies. The result is a class label for each surface patch, which were mapped back to the surface in Fig. 2 for illustration.

In the following, we describe the individual components of the approach and start with its core component, namely the persistent homology.

3.2. Persistent homology

By mathematical standards, topology with its 120 years of history, is a relatively young discipline. It grew out of Poincares seminal work on the stability of the solar system as a qualitative tool to study the dynamics of differential equations without explicit formulas for solutions [35, 36, 37]. Due to the lack of

useful analytic methods, topology soon became a purely theoretical discipline. However, in the last few years we observe a rapid development of topological data analysis tools, which open new applications for topology.

Topological spaces appearing in data analysis are typically constructed from small pieces called cells. A natural tool in the study of multidimensional images with topological methods are hypercubes (points, edges, squares, cubes etc.), e.g. a pixel in a 2-dimensional image is equivalent to a square, a voxel in a 3 dimensional volume is equivalent to a cube. Hypercubes are building blocks for structures called cubical complexes. Such representations give topology a combinatorial flavour and make it a natural tool in the study of multi-dimensional data sets.

Intuitively, the rank of the n th homology group, the so called n th Betti number denoted by β_n , counts the number of n -dimensional holes in the topological space. In particular, β_0 counts the number of connected components. As an example consider the image of the digit 8. In this image there is one connected component and two holes, hence $\beta_0 = 1$ and $\beta_1 = 2$. For a hollow sphere we have $\beta_0 = 1, \beta_1 = 0, \beta_2 = 1$. For a tube in a tire (torus) we have $\beta_0 = 1, \beta_1 = 2, \beta_2 = 1$.

Betti numbers do not differentiate between small and large holes. In consequence, the holes resulting from the noise in the data cannot be distinguished from the holes indicative for the nature of the data. For instance, in a noisy image of the digit 8 one can get easily $\beta_0 > 1$. A remedy for this drawback is persistent homology, a tool invented at the beginning of the 21st century [29]. Persistent homology tracks the holes from birth to death when the topological space is gradually built by adding cubes in some prescribed order.

If X is a cubical complex, one can add cubes step by step. Typically, the construction goes through different scales, starting from the smallest pieces. However, in general an arbitrary function $f : X \rightarrow \mathbb{R}$, called the Morse function or the measurement function, may be used to control the order in which the complex is built, starting from low values of f and increasing subsequently. This way we obtain a sequence of topological spaces, called a filtration,

$$\emptyset = X_{r_0} \subset X_{r_1} \subset X_{r_2} \subset \dots \subset X_{r_n} = X,$$

where $X_r := f^{-1}((-\infty, r])$ and r_i is a growing sequence of values of f at which the complex changes. As the space is gradually constructed, holes are born, persist for some time and eventually may die. The length of the associated birth-death intervals (persistence intervals) indicates if the holes are relevant or merely noise. The lifetime of holes is usually visualized by the *persistence diagram* (PD). Persistence diagrams constitute the main tool of topological data analysis. They visualize geometric properties of a multidimensional object X in a simple two-dimensional diagram.

Fig. 3a shows the depth map of a 3D surface where colors correspond to different depths (blue refers to low depth, yellow to high depth). In this case pixels are represented as 2-dimensional cells of a cubical complex. For the complex we can obtain a filtration X_r using a measuring function which has a value for a 2-dimensional cube equal to depth (pixel color).

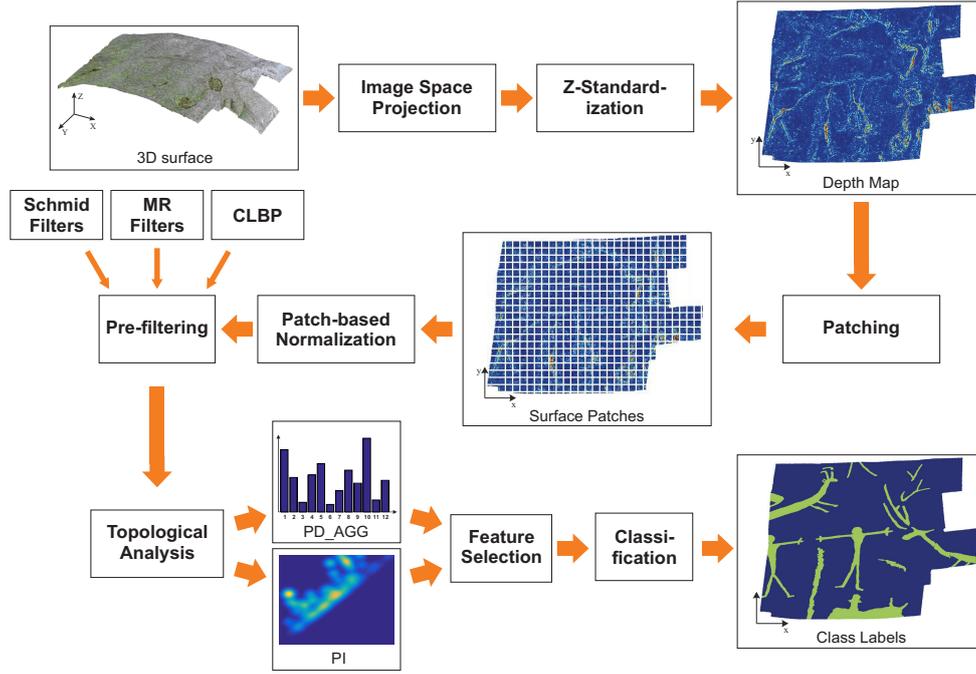


Figure 2: Overview of the proposed approach.

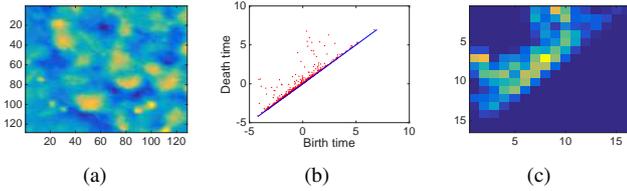


Figure 3: Example surface patch with derived topological descriptors: (a) the depth map of the original 3D surface; (b) the corresponding persistent diagram; (c) and the persistence image (resolution 16×16 pixels, $\sigma_x = \sigma_y = 0.001$ and exponential weighting function g).

For a lower dimensional cell (a vertex or an edge) we can set the function value as a minimum from the higher-dimensional neighborhoods of the cell. Fig. 3b shows the persistence diagram for X_7 .

The PD is an expressive representation of the surface topography. Unfortunately, the data structure of the PD is complex and cannot easily be combined with traditional machine learning methods which require fixed-length input vectors. This hinders the application of computational topology to computer vision and machine learning tasks. There is still no specific answer on how and when the tools of computational topology and machine learning should be used together.

A first attempt to obtain a fixed-length descriptor of a topological space filtration is to compute elementary statistics of persistence intervals (or equivalently persistence diagrams). Let

$$I := \{[b_1, e_1], [b_2, e_2], \dots, [b_n, e_n]\}$$

be a set of persistence intervals. Let $D := \{d_i := (e_i - b_i)\}_{i=1}^n$ be a set of the interval lengths. We build an aggregated descriptor of D , denoted by PD_AGG, using following measures: number of

elements, minimum, maximum, mean, standard deviation, variance, 1-quartile, median, 3-quartile, and norms $\sum \sqrt{d_i}$, $\sum d_i$, and $\sum (d_i)^2$. The result is a 12-dimensional feature vector.

3.3. Persistence image

Aside from the PD_AGG descriptor described above, which can be used with standard classification methods, there are also attempts to use PD directly with appropriately modified classifiers. Reininghaus et al. [31] proposed a multi-scale kernel for PDs, which can be used with a support vector machine (SVM). While this kernel is well-defined in theory, in practice it becomes highly inefficient when the number of training vectors becomes large, as the entire kernel matrix must be computed explicitly (note that this applies to all kernel-based methods). There are alternative methods that aim at representing the PD as a vectorial representation. An example is a persistence landscape (PL) [38] which is a functional representation of a PD. The representation lies in a Banach space and is stable. Another approach to obtain a vectorial representation from PD is persistence image introduced by Adams et al. [32], which according to Makarenko et al. [39] is especially well-suited for texture data. Persistence image (PI) is based on concepts known from the theory of size functions [33, 34]. The method scales well to large training data. In contrast to PL, PI lives in Euclidean space, which makes it directly compatible to a broad set of machine learning techniques.

The PI is derived by mapping a PD to an integrable function $G_p : \mathbb{R}^2 \rightarrow \mathbb{R}$, which is a sum of Gaussian functions centered at each point of the PD. Taking a discretization of a subdomain of G_p defines a grid. An image can be created by computing the integral of G_p on each grid box, thus defining a matrix of pixel values. Formally, the value of each pixel $p = (x, y)$ within a PI

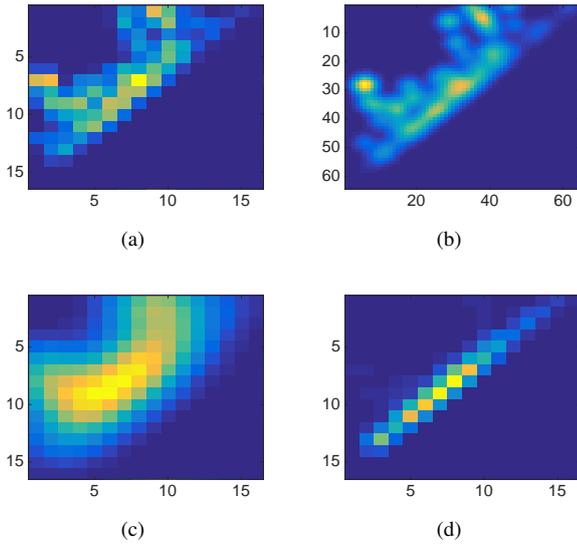


Figure 4: PIs for patch from Fig. 3a computed with different parameter values: (a) original PI from Fig. 3b; (b) resolution 64×64 pixels instead of 16×16 ; (c) $\sigma_x = \sigma_y = 0.01$ instead of 0.001; (d) no weighting function.

is defined by the following equation:

$$PI(p) = \iint_p \sum_{[b_i, e_i] \in I} g(b_i, e_i) \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\frac{(x-b_i)^2}{\sigma_x^2} + \frac{(y-e_i)^2}{\sigma_y^2}\right)} dy dx, \quad (1)$$

where $g(b_i, e_i)$ is a weighting function, which depends on the distance from the diagonal. Points close to the diagonal are usually considered as noise and therefore get lower weights by the function g . Variables σ_x and σ_y are the standard deviations of the Gaussians in x and y direction. The resulting image (see Fig. 3c) is vectorized to achieve a standardized vectorial representation which is compatible to a broad range of machine learning techniques. The pixels under the diagonal are not taken into account in case of PI (they all are zeros).

The advantage of PIs compared to aggregated PD descriptors is that the spatial information from the PD are preserved, which may lead to higher classification accuracies [32]. However, the computation of PI requires numerous parameters like the weighting function g , σ_x and σ_y of the employed Gaussians and the final image resolution. Changes in values of the parameters may result in strongly different PIs, which is illustrated in Fig. 4. It is a priori not clear which parameters yield the best representation for a specific task and how robust PI is to these parameters in practice.

3.4. Pre-filtering

Different types of pre-processing can be applied to the input data before computing topological descriptors, in order to extract more complex topological information. We take into consideration the Schmid filter bank [40] and the Maximum Response (MR) filter bank [41], as they are very popular and rotation invariant. Moreover, robust texture descriptors such as Local Binary Patterns (LBP) or Complete LBP (CLBP)[42]

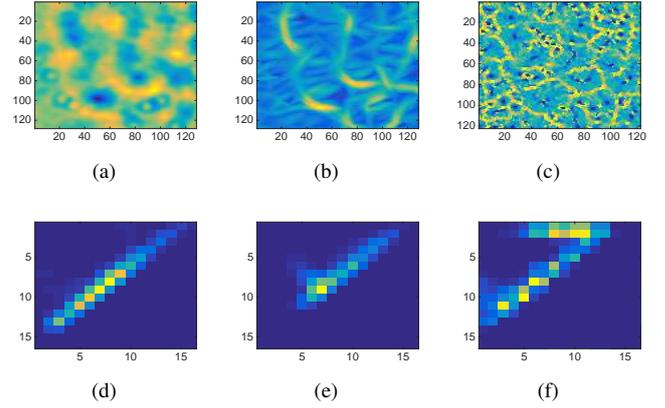


Figure 5: The patch from Fig. 3a preprocessed with: (a) Schmid filter of $size = 49$, $\sigma = 6$ and $\tau = 1$; (b) MR filter of $size = 49$, $\sigma_x = 2$ and $\sigma_y = 6$; (c) CLBP of type $riu2$, $n = 8$ and $r = 3$; together with their PIs (d-f).

may be applied before topological analysis as frequently observed in related work [31].

The patch from Fig. 3a processed with Schmid filters, MR filters, and with CLBP is presented in Fig. 5, together with the respective PIs. It can be observed that different pre-processing leads to strongly different representations. Again, it is not clear which representation is most robust and beneficial with respect to classification accuracy.

3.5. Normalization

Especially when working with depth maps normalization plays an important role for persistent homology analysis. The input depth maps have different depth value ranges depending on the characteristics of the 3D surfaces (e.g. deep valleys and high hills versus flat surfaces). Normalization can be performed at a global or local level.

Global normalization: To compute comparable topological descriptors the depth maps need to be mapped to a similar value range first, i.e. the distribution of the depth values needs to be standardized. We apply z-standardization to the depth values of each depth map separately. As a result all depth maps have zero mean and unit variance and thus cover a similar value range (except for outliers).

Local normalization: Aside from standardizing the value distribution of the depth maps additional normalization can be applied locally *across* the individual patches of the surface. Different types of normalization add different invariants to the resulting feature representations which may result in more robust classification. By performing a patch-based normalization we remove the global reference between the patches, i.e. each patch's value distribution is normalized individually, independently from the neighboring patches. Thus, after normalization the absolute depth reference is removed. What remains is pure topological surface information. This means that two patches with similar topological structure at different depth levels of the surface become similar and are represented by similar PDs.

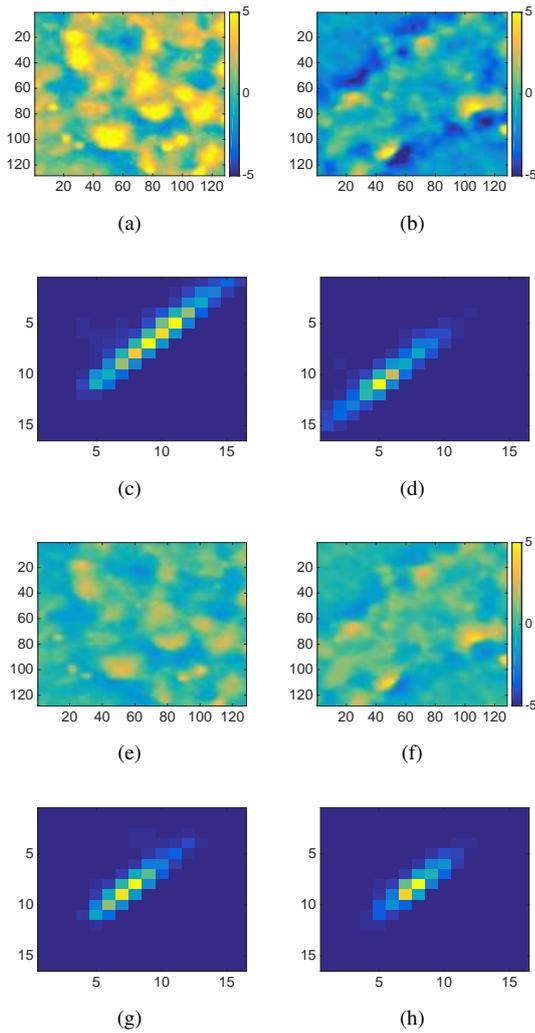


Figure 6: The effect of local normalization: two patches (a) and (b) which are located at different depth levels of the surface together with their PIs (c) and (d). The two patches become more similar after applying local min-max normalization (e) and (f). A similar effect is observed for the corresponding PIs (g) and (h).

Without patch-based normalization the absolute depth information is retained and the two patches yield different (shifted) PDs, see Fig. 6 for an example.

Given a patch P , different types of local normalization can be applied, such as:

- z-standardization: $\bar{P} = (P - \text{mean}(P)) / \text{std}(P)$,
- min-max normalization: $\bar{P} = (P - \text{min}(P)) / (\text{max}(P) - \text{min}(P))$ and
- positional standardization: $\bar{P} = (P - \text{median}(P)) / \text{mad}(P)$,

where std refers to the standard deviation and mad to the median absolute deviation.

3.6. Stability

A crucial property of persistent homology is the stability of persistence diagrams. Briefly speaking, we expect that when

a measurement function (in our case represented by a patch) changes slightly, then its persistence diagram do not change too much as well. There are two major metrics in PDs space: bottleneck distance (d_B) and Wasserstein distance (d_W). Cohen-Steiner et al. prove in [43, 44] results about stability of PD under d_B and d_W metrics for classes of the measurement functions. Especially, by [44, Wasserstein Stability Theorem] we know that the computation of persistence diagrams for Lipschitz functions is stable under d_W .

In the case of depth map it is easy to observe that the measurement function is Lipschitz (the size and pixel values are bounded). In [32] it is proven that PIs are stable under d_W , as well. Hence, we consider the process of obtaining PIs from surface patches being stable in theory. In our experiments we analyze this the stability of the resulting descriptors in practice.

4. Datasets and evaluation protocol

For our experiments, we employ a recently released dataset of high-resolution 3D reconstructions from the archaeological domain with a resolution of approximately 0.1 mm [11]. The dimensions of the scanned surfaces ranges from approx. 20×30 cm to 30×50 cm. The reconstructions represent natural rock surfaces that exhibit human-made engravings (so-called rock art). The engravings represent animals, symbols and figures that have been engraved by humans in ancient times. See Fig. 1 for an example surface. The engraved regions in the surface exhibit a different surface texture than the surrounding natural rock surface. In our experiments we aim at automatically separating the engraved areas from the natural rock surface.

For each surface a precise ground truth has been generated by domain experts that labels all engravings on the surface. The dataset contains two classes of surface topographies: class 1 represents engraved areas and class 2 represents the natural rock surface. Class priors are imbalanced. Class 1 represents approximately 19% of the data and is thus underrepresented. A corresponding ground truth labeling is depicted in Fig. 1c.

In total 26 high-resolution surface reconstructions have been acquired as described above. To accelerate experiments, we form two datasets of different sizes:

Small-scale dataset: . The small scale dataset contains depth maps from 4 surface reconstructions with a total number of 12.3 millions of points. This dataset has also been employed for the experiments in [8] and thus represents a baseline dataset. We employ this dataset for our initial experiments to reduce computation time and to enable a broader set of experiments in our evaluation. Furthermore, by using this dataset our results become comparable to previous work. In our evaluation the depth maps of the first two surface reconstructions are employed for training and the remaining two surfaces represent the independent evaluation set.

Large-scale dataset: . This dataset contains depth maps of all the 26 surface reconstructions that have been acquired. This

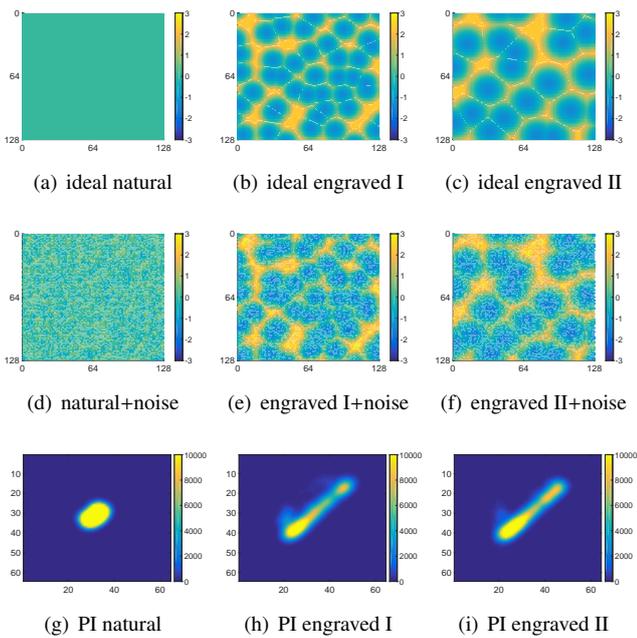


Figure 7: Synthetic dataset: “natural” and “engraved” surfaces without noise (a-c); the same surfaces with noise (d-f); and their PIs (g-i). PI generated for the “natural” surface is unimodal, while PIs generated for “engraved” surfaces are bimodal and less coherent.

dataset is employed in our evaluation to demonstrate the generalization ability of the proposed approach and to verify if conclusions drawn from the small-scale dataset can also be transferred for the larger and more complex one. For classification experiments we randomly select 13 surfaces as a training set and employ the remaining ones for evaluation.

We further perform experiments on synthetic data to provide an experimental proof-of-concept for our topological approach.

Synthetic dataset: . The synthetic dataset serves as a proof-of-concept to simulate the process of engraving a natural rock surface and the effect of the resulting surface topographies on the topological descriptors. We start with an ideally flat surface that represents a “natural” rock surface without any irregularities, see Fig. 7a. All values of this surface are constant and equal to zero. In Figs. 7b and c we simulate differently dense and coarse engravings by adding negative Gaussian distributions to the flat surface. The spacing between Gaussians equals 20 ± 5 and 30 ± 5 pixels. In a next step, we add noise to all surfaces to simulate natural irregularities and effects of long-time exposition of the surfaces to weather. Noise is generated using the surface generator of [45]². Fig. 7d shows the resulting noise, i.e. the simulated flat natural surface. We further add the same noise to the simulated engraved surfaces from Fig. 7b and c, see Figs. 7e and f. The depth values of all surfaces are z-standardized.

²implementation available from: http://www.mysimlabs.com/surface_generation.html, last visited June 2017

Evaluation Protocol. Our evaluation protocol is defined as follows: from the input depth maps, we extract overlapping square image patches which are input to feature extraction (either for topological features or non-topological features). The resulting feature vectors for each patch are then input to classification. The classification performance is measured by the Dice Similarity Coefficient (DSC). DSC measures the matching between the ground truth and the predicted class labels, i.e. the mutual overlap between an automatic labeling X of a depth map and a manual (ground truth) labeling Y :

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

DSC is between 0 and 1, where 1 means a perfect classification.

For each classification experiment we first select a subset S_i of patches from the training set T . Next, we estimate suitable classifier parameters by 5-fold cross-validation on the subset S_i . With the optimized parameters we train a classifier on the entire subset S_i . This classifier is finally tested on the evaluation set. There is no parameter tuning on the evaluation set.

Each classification experiment is repeated 10 times for different randomly selected subsets S_i from the training set T . In each repetition other patches from the training set T are selected and thus the actual training data varies over the 10 repetitions. The result are 10 classification results (DSC values) for each experiment. We report the mean and the standard deviation over all 10 repetitions. By conducting each classification experiment with different subsets of the training set we reduce the dependency of our results on the training data.

As mentioned above, the class priors in our dataset are imbalanced. Skewed datasets pose problems to most classification techniques and often yield suboptimal models as one class dominates the other classes. A classifier especially designed for imbalanced datasets is Random Undersampling Boosting (RUSBoost) [46]. RusBoost builds upon AdaBoost [47] which is an ensemble method that combines the weighted decisions of weak classifiers to obtain a final decision for a given input patch. RUSBoost extends this concept by a data sampling strategy that enforces similar class priors. During each training iteration the majority class in the training set is undersampled in a random fashion to balance the resulting class priors. In this manner, the weak classifiers can be learned from balanced datasets without being biased from the skewed class distribution. In previous experiments RUSBoost has already achieved robust results and high classification performance and has outperformed among others cost-sensitive SVMs [48] with linear and RBF kernels which are also suitable for imbalanced data.

The code used in our study was implemented in Matlab. Most of the descriptors were extracted with VLFeat library [49], except for PD_AGG and PI which were computed with the CAPD::RedHom library [50, 51] with the PHAT [52, 53] algorithm for persistence homology.

5. Experiments and results

We first investigate the expressive power of topological descriptors for our synthetic data set and then perform a compre-

hensive study on the robustness and expressiveness of the topological descriptors for the analysis of real-world 3D surfaces. We investigate different variants of their computation and compare them with traditional non-topological descriptors. Finally, we investigate the effect of combining topological with non-topological descriptors. For each set of experiments we first describe the experimental setup and the investigated aspects in the context of the experiment and then report on the obtained results. Unless otherwise stated the experiments on the 3D reconstructions were performed on the small-scale dataset.

5.1. Experiments on synthetic data

We generated PI descriptors for the synthetic surfaces using the baseline configuration (see Tab. 1). The PI generated for the natural surface in Fig. 7g shows a bright cluster in the center of PI. This is plausible, because most of the holes are born and die at the same time. The PIs of the two engraved surfaces in Figs. 7h and i exhibit a completely different spatial distribution. The reason is that there are two types of holes on the surfaces. One group of holes corresponds to the natural irregularities (noise) and the other one to the engravings. Holes from both groups are born and die at different times. Thus the PIs become bimodal. This is strongly different from the natural surface for which PI is unimodal and more coherent.

These experiments demonstrate that the topological descriptors capture different surface topographies well. In the following sections we analyze real-world 3D surfaces and perform a detailed analysis of topological and non-topological descriptors.

5.2. Topological baseline descriptors

In an initial experiment we extract topological descriptors which serve as a baselines for later comparison. From the depth maps in the training and evaluation sets we extract topological descriptors in a patch-based manner. The patch size is 128×128 pixels (i.e. 10.8×10.8 mm) and the step size between two patches is 16 pixels (1.35 mm). The baseline descriptors include: (i) aggregated descriptors from the persistence diagram (PD_AGG), see Section 3.2 and (ii) descriptors obtained from the persistence image (PI), see Section 3.3.

For both descriptors we select those parameters for which they have yielded best results in our previous investigation [9]. See Tab. 1 for the parameter selections for the baseline configurations and an overview of all possible parameter values.

PD_AGG results in an 12-dimensional feature vector (see Section 3.2), while the feature vector derived from PI has a dimension D of 136 which originates from the fact that all values under the diagonal are zero and are skipped. Thus, $D = (R^2 + R)/2$ where R refers to the resolution of the PI ($R = 16$ for the baseline configuration).

As already mentioned in Section 3.5 the input depth maps have different depth value ranges and are z-standardized to make different surfaces comparable. After z-standardization the depth maps have zero mean and unit variance and thus cover a similar value range (except for outliers). Nevertheless we unify the PDs by defining a parameter “diagram limits” as an interval

Descriptor	Parameter	Possible values	Baseline configuration
PD_AGG	Local normalization	{none, z-std, p-std, minmax}	none
	Diagram limits	[min, max]	[-5, 5]
	Outlier removal	{yes, no}	no
	Mapping	{none, CLBP}	none
	Pre-filtering	{none, Schmid, MR}	none
PI	Local normalization	{none, z-std, p-std, minmax}	none
	Diagram limits	[min, max]	[-5, 5]
	Sigma	\mathbb{R}_+	0.001
	Resolution	\mathbb{N}_+	16
	Weighting	{none, linear, exponential}	none
	Outlier removal	{yes, no}	no
	Mapping	{none, CLBP}	none
	Pre-filtering	{none, Schmid, MR}	none

Table 1: Topological descriptors, their computation parameters and the specification of the baseline configuration. The abbreviations “z-std” stands for z-standardization, “p-std” for positional standardization, and “minmax” for min-max normalization.

Descriptor	DSC				
PD_AGG	0.655 ± 0.012				
PI	$R = 8$	$R = 16$	$R = 32$	$R = 64$	
	$\sigma = 0.00025$	0.701 ± 0.003	0.729 ± 0.003	0.732 ± 0.002	0.725 ± 0.007
	$\sigma = 0.0005$	0.712 ± 0.005	0.730 ± 0.002	0.731 ± 0.004	0.730 ± 0.003
	$\sigma = 0.001$	0.722 ± 0.006	0.733 ± 0.003	0.734 ± 0.004	0.735 ± 0.004
	$\sigma = 0.002$	0.723 ± 0.002	0.728 ± 0.003	0.730 ± 0.003	0.730 ± 0.003

Table 2: DSC obtained for PD_AGG and PI with various resolutions and sigmas. Numbers in bold refer to the baseline descriptors.

$[min, max]$. Both values $[min, max]$ define the axis limits of the underlying PD. We set these values to $[-5, 5]$ which can safely be done as after z-standardization the depth values most likely lie in the interval $[-5, 5]$ except for individual outliers. Note that if patch-based normalization is used (see Section 3.5), the diagram limits need to be adapted according to the type of normalization which may change the depth value range, e.g. in the case of min-max normalization depth values are moved to the interval $[0, 1]$.

The baseline configuration of PD_AGG yields a DSC of 0.655 ± 0.012 . The baseline for PI outperforms PD_AGG with a DSC of 0.733 ± 0.003 . A reason for the better performance of PI is that PI preserves spatial information from the PD as already mentioned in Section 3.3 while PD_AGG captures only overall statistics. We repeat each experiment 10 times with different training data (see Section 4 for details). The standard deviation across all the 10 experiments is low (0.012 and 0.003) which shows that the dependency on the training data is low as well for both descriptors.

5.3. PI with different resolutions and sigmas

To investigate the sensitivity of resolution and sigma for the computation of PI, we extract PIs for different resolutions (8, 16, 32, 64 pixels) and standard deviations (0.00025, 0.0005, 0.001, 0.002). The remaining parameters of the baseline configuration remain constant.

The results are presented in Tab. 2. The performance of the baseline descriptors is highlighted bold. Results show that the sensitivity of PIs to both parameters is low. With increasing resolution the performance slightly improves (especially by going from 8×8 to 16×16 pixels). Higher resolutions do not provide significant improvements. Since the computation time grows

quadratically with resolution it must be stressed that it is desirable to keep the resolution as small as possible.

Experiments for different sigmas show that the best results are obtained with $\sigma = 0.001$ independently from the resolution. Thus both parameters can be optimized independently from each other which saves computation time in parameter tuning and model selection in practice.

5.4. Different normalization variants

We perform experiments with different normalization variants with PI and PD_AGG and compare them to the baseline feature configurations. Variants include:

1. *No normalization at all*: we do not provide diagram limits for the computation of the PD. Thus each PD adapts to the value range of the underlying image patch and thus PD_AGG and PI adapt completely to the value range.
2. *Globally defined diagram limits*: for each input patch the same diagram limits for PD are applied. This corresponds to the baseline features.
3. *Patch-based normalization with z-standardization*: each input patch is z-standardized individually before feature extraction.
4. *Patch-based normalization with positional standardization*: same as above with positional standardization.
5. *Patch-based normalization with min-max normalization*: same as above with minmax normalization.

The first variant demonstrates what happens if we apply the topological features directly on the data without any adaptations and is expected to represent a lower bound of performance. In variant 2 the depth relations (i.e. the absolute depth differences) between patches are retained. In variants 3-5 depth relations between patches are removed and the only the pure topological information inside each patch is retained.

Results for the different normalization variants are presented in Fig. 8. The best accuracy is obtained with globally defined diagram limits, which preserve the depth relations between the patches. Applying no normalization at all is suboptimal for PI. The three patch-based normalization variants all yield lower results. This shows that the depth relations between patches are important for the classification task.

Interestingly, positional standardization which is robust to outliers in the depth distribution leads to the weakest results. This indicates that the outliers in the distribution are of high importance for the task.

Due to the different normalizations the above descriptors (especially variant 2 and the patch-based normalization variants) may represent complementary information that may be beneficial in combination. To investigate this further, we evaluate different combinations of PD_AGG and PI with different normalization types. We select patch-based z-standardization in the following experiments. Finally, we investigate the effect of combining PI with PD_AGG (with similar and different normalizations). For the combination of features we concatenate the respective feature vectors before feeding them into the classifier (early fusion). Results are summarized in Tab. 3.

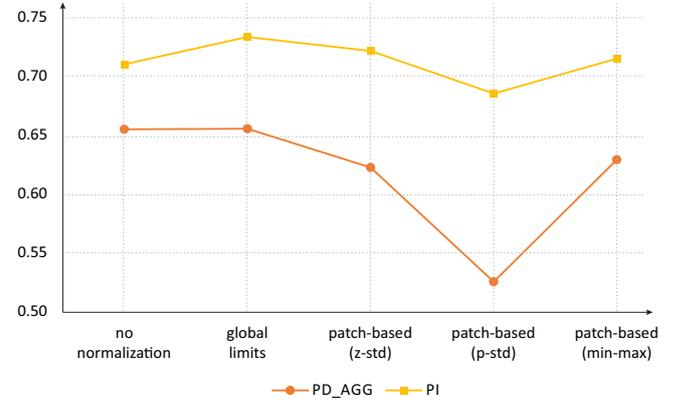


Figure 8: The effect of different normalization methods for PI (yellow line, square markers) and PD_AGG (orange line with circular markers). Patch-based normalizations (z-std, p-std, and min-max) are outperformed by global normalization. Applying no normalization at all (first experiment) is not beneficial for 3D surface analysis.

Descriptor(s)	Normalization	DSC
PD_AGG	Global limits (baseline)	0.655 ± 0.012
PD_AGG	Global limits + z-std	0.671 ± 0.008*
PI	Global limits (baseline)	0.733 ± 0.002
PI	Global limits + z-std	0.735 ± 0.004
PI+PD_AGG	Global limits (baselines)	0.730 ± 0.003
PI+PD_AGG	z-std	0.723 ± 0.005
PI+PD_AGG	Global limits + z-std	0.735 ± 0.002*

Table 3: DSC for combining PI with PD_AGG (with the same and different normalization types). Asterisks (*) correspond to results significantly better than corresponding baseline topological descriptors with $p < 0.05$.

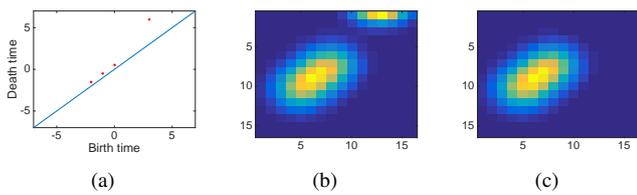


Figure 9: The effect of outliers on the PI. A PD containing 4 sample points $[-2, -1.5]$, $[-1, -0.5]$, $[0, 0.5]$ and $[3, 6]$ (a) and the corresponding PIs generated without (b) and with (c) outlier removal, with diagram limits $[-5, 5]$. Although the point $[3, 6]$ lies outside of the domain of the PI (and is thus considered an outlier), the Gaussian centered around it still extends into the domain of the PI when outliers are not explicitly removed.

The only result in Tab. 3 where the combination of different normalized descriptors improves results is for PD_AGG (DSC increases to 0.671 ± 0.008 with $p < 0.05^3$). In case of PI only a small improvement by 0.002 is observed which is not significant. Interestingly, the combination of PI with PD_AGG does not improve results over the baseline of PI alone. A reason for this may be that PI implicitly captures the information that is represented in PD_AGG.

5.5. Outlier removal

Outliers are usually considered as unwanted artifacts. In persistence homology, however the role of outliers is different. Outliers in the PD, i.e. points with a large distance to the diagonal, are usually considered as the most meaningful and strong topological components of the underlying data.

In our approach outliers are those points in the PD which extend the pre-defined diagram limits. There are two alternatives: to keep them or to remove these outliers completely from the further computation. For the two investigated topological descriptors this has different consequences. In the case of PD_AGG the outliers are simply included in the derived statistics, which is the same as extending the diagram limits to the farthest outliers. For PI this means that although the outliers lie outside of the domain of the PI the Gaussian centered around an outlier during PI construction may still extend into the domain of the PI (see Fig. 9c).

We compare the effect of outlier removal by computing respective variants of our baseline features. The results show that removing outliers decreases results for PI from 0.733 ± 0.003 to 0.731 ± 0.003 and for PD_AGG from 0.655 ± 0.012 to 0.639 ± 0.011 . In case of PD_AGG the decrease is significant with $p < 0.05$). This indicates that outliers play an important role for the topological descriptors.

5.6. Weighting in PI computation

The original formulation of PI includes a weighting function for the individual Gaussian components which grows exponentially with distance to the diagonal of the PD (function g in Equation 1 in Section 3.3). The intuition behind this is to

emphasize stronger topological components in the PD (further away from the diagonal) and to attenuate the influence of noise which is often located close to the diagonal. Previous experiments have shown that the weighting is in practice not always beneficial [9]. To further investigate the role of weighting in the PI computation, we compare the performance of PI with different weighting functions namely: constant weighting, linear weighting and exponential weighting.

Experiments reveal that weighting significantly decreases performance with $p < 0.005$: from 0.733 ± 0.003 to 0.714 ± 0.008 in case of linear weighting and to 0.715 ± 0.007 in case of exponential weighting. Best results are obtained when all Gaussians are equally weighted. This indicates that short intervals close to the diagonal play an important role in the context of surface texture description. This contradicts with the original assumptions made in the design of PI. We investigate this aspect further in Section 5.13.

5.7. Pre-filtering

We investigate different ways to pre-filter the surface patches before extracting topological features. This has recently been proposed to obtain more robust and expressive representations [30, 31]. We employ three different methods for pre-filtering (as described in Section 3.4): (i) Schmid filter bank [40], (ii) MR filter bank [41] and (iii) CLBP encoding [42].

For MR and Schmid filter banks we employ the default parameters proposed by the respective authors. The only exception is that in case of Schmid filter banks we use two times smaller τ , as in preliminary experiments there is hardly any difference in the filter responses for $\tau \geq 2$. Each patch is filtered with all filters from the respective filter bank. For each filtered patch topological features are extracted separately and are then concatenated into one high-dimensional feature vector.

Additionally, we extract CLBP features [42] from the raw patches as proposed in [30] and [31] and then extract topological descriptors from the CLBP_S and CLBP_M maps. For CLBP computation we employ the implementation of [54] with different encodings: rotation invariant LBP (ri) and rotation invariant uniform LBP (riu2) as well as different numbers of neighborhood samples $n = \{8, 16\}$ and different radii $r = \{3, 5\}$.

The results of experiments are presented in Tab. 4. It can be observed that prefiltering with Schmid and MR filter banks does not improve performance, both for PD_AGG and PI. CLBP, however, improves DSC significantly in case of PD_AGG (from 0.655 ± 0.012 to 0.708 ± 0.005 , $p < 0.01$ and CLBP with “riu2” encoding, $r = 5$ and $n = 8$). The effect of pre-filtering is further analyzed in Section 5.12. From the experiments so far it seems that for PI the best strategy is to apply it on the raw data directly, while for PD_AGG which are much simpler and coarser descriptors the additional pre-filtering can be beneficial.

Finally, we combine our baseline features with the respective Schmid, MR and CLBP filtered variants by early fusion (concatenation) to investigate if the descriptors can benefit from each other. Experiments show that most of the combinations decrease performance (see Fig. 10), which is consistent with the pre-filtering results presented above. The only positive effect in

³Statistical significance is computed with the Wilcoxon signed rank test, as most of the samples do not pass the Shapiro-Wilk normality test.

Descriptor	Pre-filtering			DSC	
PD_AGG	(baseline)			0.655 ± 0.012	
PD_AGG	Schmid			0.643 ± 0.023	
PD_AGG	MR			0.607 ± 0.044	
				$n = 8$	$n = 16$
PD_AGG	CLBP	riu2	$r = 3$	0.671 ± 0.004**	0.674 ± 0.009**
			$r = 5$	0.708 ± 0.005**	0.678 ± 0.008**
	CLBP	ri	$r = 3$	0.697 ± 0.006**	0.670 ± 0.019*
			$r = 5$	0.700 ± 0.006**	0.687 ± 0.006**
PI	(baseline)			0.733 ± 0.002	
PI	Schmid			0.702 ± 0.007	
PI	MR			0.698 ± 0.004	
				$n = 8$	$n = 16$
PI	CLBP	riu2	$r = 3$	0.712 ± 0.014	0.725 ± 0.004
			$r = 5$	0.707 ± 0.014	0.719 ± 0.006
	CLBP	ri	$r = 3$	0.709 ± 0.009	0.722 ± 0.007
			$r = 5$	0.706 ± 0.006	0.706 ± 0.005

Table 4: Results for pre-filtering depth maps before extraction of topological features. Asterisks (*) and (**) correspond to results significantly better than corresponding baseline topological descriptor with $p < 0.05$ and $p < 0.01$, respectively. Bold represent the best results for PD_AGG and PI.

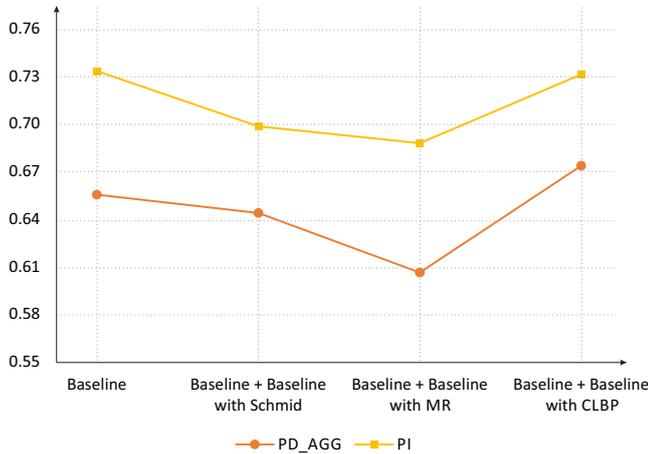


Figure 10: DSC for combining baseline topological descriptors with the respective Schmid, MR and CLBP filtered variants. For PI the combination is not able to further improve results. PD_AGG benefits from adding the CLBP-filtered variant but not significantly.

performance is achieved by the combination of PD_AGG with the respective CLBP filter variant. This is also consistent with the previous results where CLBP improved PD_AGG. The result of the combined descriptors is, however, weaker than that of PD_AGG based on CLBP alone (0.674 vs 0.708). This shows that there is no synergy attained by the combination.

5.8. Feature selection

By applying pre-filtering (as in the previous Section 5.7) or by using PIs with high resolutions the dimension of the resulting feature vectors grows linearly and quadratically, respectively. Thus, strategies to reduce the dimension may be useful in practice. We apply different feature selection strategies to the PI descriptors to investigate to which degree dimensionality reduction affects classification performance and how much redundancy the descriptors exhibit. We apply the following three

strategies:

1. *Gini importance*: we select Gini importance [55] as a strategy because it is a by-product of the employed classifier (RUSBOOST, see Section 4) and it can be obtained without additional cost. It directly reflects the importance of each descriptor entry for distinguishing between the classes.
 Computation: we compute the Gini importance for each entry of the PI descriptor during training of the classifier and keep only the N most important (highest values) entries.
2. *Fisher criterion*: While Gini takes dependencies between descriptor entries into account (through the classifier tree hierarchy), the Fisher criterion [56] is completely independent from the classifier and evaluates the importance of every entry independently. For this reason we decided to use it as a complementary strategy.
 Computation: we compute the Fisher criterion for each descriptor entry during training of the classifier and keep only the N most discriminative entries (highest Fisher criterion)
3. *Combined Gini and Fisher criterion*: we combine both strategies to see if there are synergy effects between them.
 Computation: for each descriptor entry we compute Gini importance and Fisher criterion and average the rankings obtained by both measures. The N entries with the highest combined ranking are retained.

Additionally, to provide a baseline, we select features randomly to investigate the actual benefit of feature selection strategies.

For all three feature selection approaches we vary N from 2, 4, 8, 16, 32, 64 to 100% of the PI descriptor (136 dimensions). Results are illustrated in Fig. 11. Note, that the x-axis in Fig. 11 is not linear. By selecting only 2% of the components, Gini yields a DSC of 0.6015 ± 0.0197 . With the Fisher criterion even

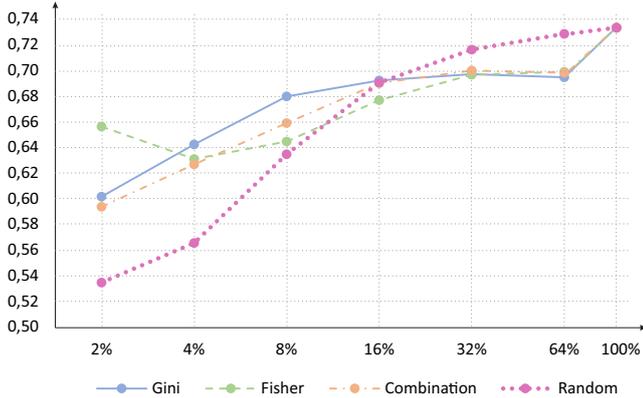


Figure 11: Classification performance (DSC) with different feature selection strategies for differently large subsets of the baseline PI descriptor.

a DSC of 0.6560 ± 0.0213 is achieved. Both values strongly outperform the random baseline (0.535 ± 0.06). With increasing number of features the performance increases continuously for Gini. The Fisher criterion is less stable and is outperformed in most cases by Gini as can be seen in Fig. 11. A reason for the better performance of Gini is that Gini is computed during classifier training and expresses the importance of a given entry for the classifier training. Fisher, in contrast, is computed independently from the classifier from the training data directly. The combined scheme is located approximately between the two approaches which may stem from averaging the respective rankings of the individual measures. Interestingly, feature selection is only reasonable up to a certain number of features. By selection of more than 16% of features the random baseline outperforms the selection strategies. The reason for this behavior is that the feature selection strategies do not take the inter-correlations between the entries into account and thus redundant components are easily selected when the number of selected features increases. We further investigate the importance and discriminativity of individual entries in Section 5.13.

5.9. Comparison with non-topological features

From the patches of the depth map we extract a number of non-topological image descriptors in the same patch-wise manner as the topological descriptors. Non-topological descriptors include: MPEG-7 Edge Histogram (EH) [57], Dense SIFT (DSIFT) [58], Local Binary Patterns (LBP) [59], Histogram of Oriented Gradients (HOG) [60], Gray-Level Co-occurrence Matrix (GLCM) [61], Global Histogram Shape (GHS) [8], Spatial Depth Distribution (SDD) [8], as well as enhanced versions of GHS and SDD (short EGHS and ESDD) that apply additional enhancements to the depth map described in [8].

For the extraction of non-topological features, we employ publicly available implementations if available, such as VLFEAT for SIFT and HOG[62], the LBP implementation of [63], and the GLCM implementation of [64]. GHS, SDD, EGHS, and ESDD are implemented as in [8].

Results for the individual descriptors are shown in Fig. 12 (blue line). The best individual non-topological descriptor is ESDD (DSC of 0.743 ± 0.002) followed by the combination

of EGHS+ESDD (DSC of 0.728 ± 0.005). The performance of our two baseline descriptors are shown with a dotted line (PD_AGG) and a dashed line (PI) in black for comparison. The PD_AGG descriptor achieves similar results to LBP, EH, GLCM, and EGHS. Remarkably, PI alone already outperforms all non-topological features except ESDD. This attests a strong explanatory power of the PI for the description of surface texture while PD_AGG rather lacks in expressiveness.

5.10. Combination with non-topological features

In previous investigations the addition of topological descriptors to other (non-topological) descriptors has improved classification performance [9]. We investigate the effect of combining both types of descriptors by adding the best configurations of PI and PD_AGG to the evaluated non-topological descriptors from Section 5.9. The results are shown as additional curves in Fig. 12. The orange curve (square markers) shows the performance when non-topological descriptors are combined with PD_AGG and the yellow curve (diamond-shaped markers) shows the performance in combination with PI. The combination with PD_AGG improves all descriptors except for EH. PI improves all descriptors and some in fact by a large margin. Even the performance of the best non-topological descriptor (ESDD) can be further improved when combined with PI (from 0.743 ± 0.002 to 0.788 ± 0.005 with $p < 0.01$). A corresponding example result is shown in Figure 13. The surface shows a horse with a rider that holds a long spear in its hand. The first row shows the original rock-surface, the depth map, and the ground-truth labeling (engraved areas in red, remaining areas uncolored). The second row shows the results of ESDD, PI and ESDD+PI. Classification results are color coded: red refers to true positive detections of class 1 (engraved area), green to false positive detections of class 1 and blue to false negative detections of class 1. Uncolored areas are true positive detections of class 2. The results for ESDD and PI are comparable in overall quality but show different strengths and weaknesses of the descriptors. ESDD better captures fine details than PI (e.g. the rear legs) but produces more false detections (green) at natural surface irregularities like the vertical and diagonal cracks in the rock surface. PI is more robust to such irregularities and captures the shape of the engraved areas well but generates more false detections along the boundary of the engraved areas (e.g. along the spear and the horse's tail). In combination the mutual weaknesses compensate each other (e.g. the shaft of the spear and the arm of the rider are better detected) and yield a more accurate overall segmentation. These results illustrate well that topological descriptors represent complementary information about the surface texture compared to traditional texture descriptors and that their combination is beneficial.

5.11. Large-scale experiments

Experiments so far were performed on the small-scale dataset which was chosen to save computation time and in turn gave us the opportunity to broaden our evaluation to many different aspects, parameter values, configurations etc. A central question in our evaluation is how topological descriptors compete with

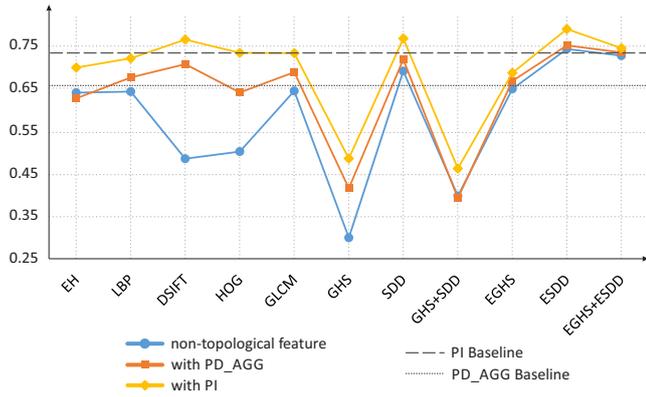


Figure 12: Classification results (DSC) for different non-topological descriptors (blue line with circular markers). The orange (square marker) and yellow (diamond marker) lines represent the performance when the features are combined with PD_AGG and PI, respectively. The baselines for PD_AGG and PI are plotted as dotted and dashed lines, respectively.

non-topological descriptors which currently represent the state-of-the-art for 3D surface and texture analysis [11]. To answer this question more profoundly, we run the experiments from Sections 5.9 and 5.10 on the large-scale dataset from Section 4. We evaluate all descriptors without further parameter tuning to obtain objective and unbiased results, i.e. no fine tuning is performed on the large-scale dataset at all. The topological descriptors are then compared to a selection of the most promising non-topological descriptors. Furthermore, we evaluate the performance of the best descriptor combinations evaluated so far.

Results for the large-scale dataset (together with the corresponding results for the small-scale dataset) are presented in Fig. 14. The overall performance of all descriptors drops for the large-scale dataset. This is expected and results from the higher complexity of the dataset (more heterogeneous data). The absolute performance difference between the two datasets is, however, quite consistent between all descriptors ($\Delta = 0.082$ with a quite low standard deviation of ± 0.014), which shows that the classifier can generalize well from all descriptors. The best topological descriptor is again PI with 0.654 ± 0.002 which is similar to the best non-topological descriptor (ESDD) with 0.661 ± 0.001 (dashed line). The smaller difference in peak performance between topological and non-topological descriptors compared to the small-scale dataset shows that the topological descriptors better generalize to the more complex data in the large dataset.

The combination of topological and non-topological descriptors yields a performance gain in most cases. Peak performance is obtained when all evaluated descriptors are combined: 0.681 ± 0.002 . This improvement is significant with $p < 0.01$ compared to the best combination of non-topological descriptors. The two horizontal lines in Fig. 14 represent the peak performance without and with topological information. The vertical spacing between the lines represents the gain obtained by adding topological information in classification (+0.02). Note that the gain for the small-scale dataset is even larger (+0.045).

We further compare our results with those obtained in [11]

on the same dataset. In contrast to the investigation in [11] which uses 4-fold cross-validation on the entire dataset, we employ 5-fold cross-validation on the training set and additionally use half of the dataset as a completely independent test set. The use of an independent test set makes our evaluation protocol is more demanding than that of [11]. Reference [11] reports results for two approaches: Random Forests (RF) and Convolutional Neural Networks (CNNs). For the CNN the fully convolutional net by [65] has been fine-tuned on randomly selected patches from the training data, see [11] for details. RF yields a DSC of 0.568. CNN outperforms RF with a DSC of 0.667. With a DSC of 0.654 our best topological feature (PI) yields a similar performance level as CNNs in [11]. PI combined with non-topological descriptors even outperforms the results of the CNN with a peak performance of 0.681. Thus, the performance of the investigated descriptors can be considered state-of-the-art.

5.12. Robustness and sensitivity

A crucial property of persistent homology is stability. In the following experiments we evaluate if the theoretical conclusions about stability from Section 3.6 hold in practice. For this purpose, we randomly select number of patches and add different levels of random Gaussian noise (see Fig. 15a-c for an example). Next, we compute the differences between the raw patches and their noisy variants as well as the differences between the PIs computed from the raw and noisy patches. The normalized difference d between two patches is computed as:

$$d(P, N) = \frac{\sum_{p \in P} |P(p) - N(p)|}{\sum_{p \in P} |P(p)| + |N(p)|},$$

where N and P is the original patch with and without random Gaussian noise. The normalized difference between two PIs is computed analogously to that of patches by replacing N and P with the corresponding PIs. In Fig. 16 we plot the differences between the patches (x-axis) vs. the differences between the PIs (y-axis). Color encodes the signal-to-noise ratio (SNR).

With decreasing SNR (increasing noise level) the differences between patches and PIs increases, see Fig. 16a. Interestingly, the differences between the PIs exceeds the differences between the raw patches (i.e. points are above the diagonal) which shows that there is a strong sensitivity to the introduced noise.

Different types of pre-filtering (described in Section 3.4) can also influence the stability of the resulting representations. To evaluate this further, we perform a similar analysis for patches filtered with Schmid filter, MR filter and CLBP (see Fig. 16b-d). Experiments reveal that the difference between PIs grows much slower in case of Schmid filter and MR filter (nearly all points are below the diagonal). This is due to the fact, that both filters are defined based on Gaussian functions and have a low-pass characteristic. Thus, noise is removed to a wide extent in both cases. For CLBP the behavior is similar to the raw patches. However, the absolute value of difference d is much lower ($d < 0.6$ instead of $d < 0.9$) than for the raw patches

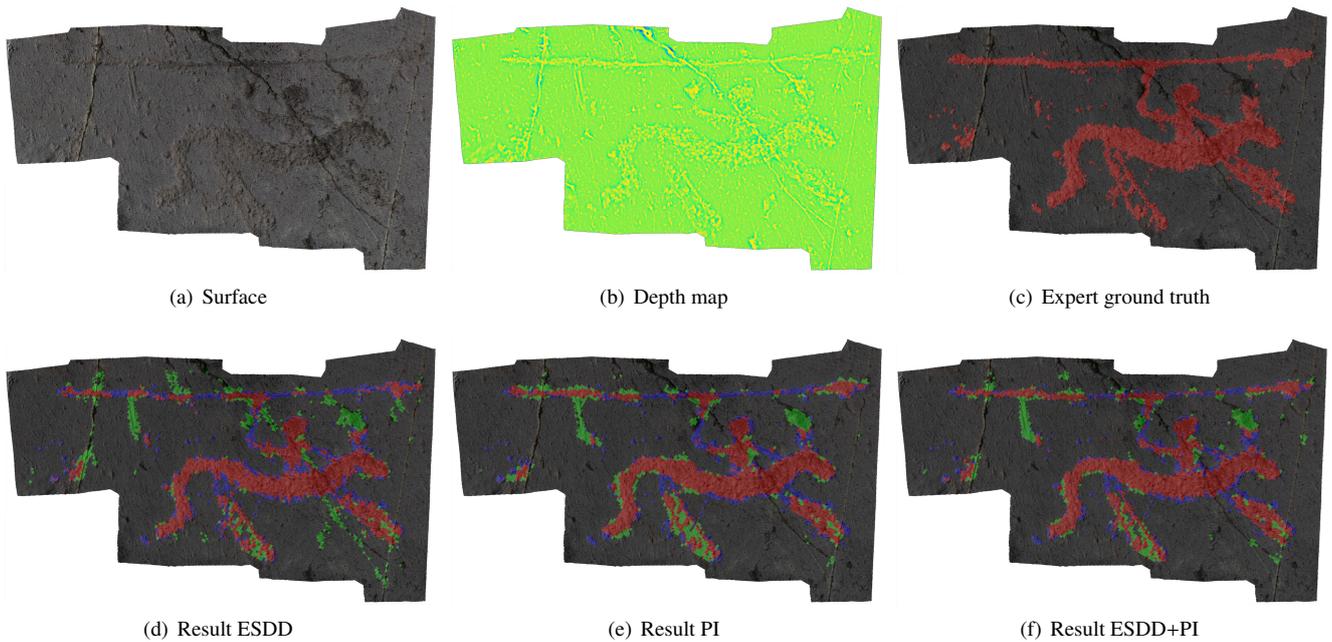


Figure 13: Classification results for a surface from the test set: (a) the original rock surface; (b) the depth map of the surface (in false color and contrast enhanced); (c) the ground truth labeling obtained by domain experts (engraved areas are highlighted red); (d,e,f) show the classification results of ESDD, PI and the combination of both features. Red areas correspond to correctly classified engraved areas (class 1), green areas are false positives detections of class 1 and blue refers to false negatives (non-detected engraved areas). All remaining (uncolored) pixels are true positives for class 2.

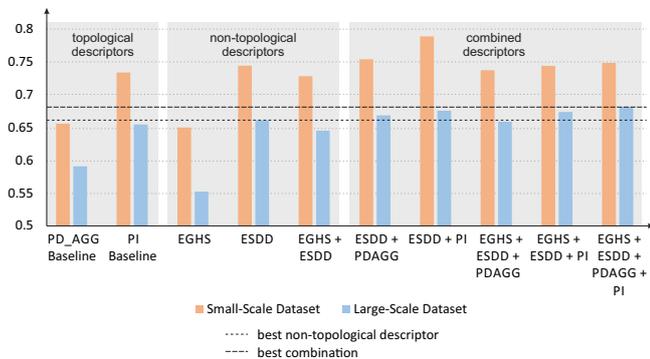


Figure 14: Performance comparison between small-scale and large-scale dataset in terms of DSC. Also for the large-scale dataset the topological descriptors add additional information and are necessary to obtain peak performance.

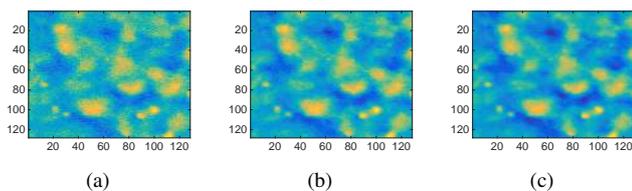


Figure 15: The patch from Fig. 3a with random Gaussian noise. SNR equals: (a) 5; (b) 10; (c) and 15.

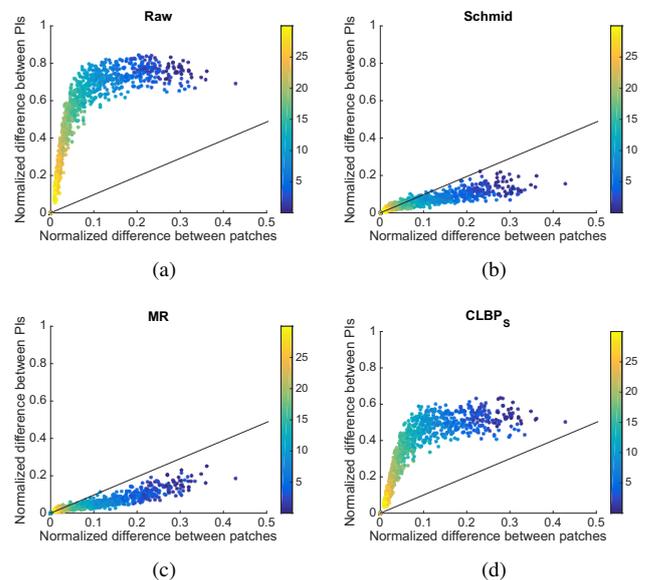


Figure 16: The noise analysis: abscissa represents difference between the original patch and the same patch with random Gaussian noise; ordinate represents normalized difference between their PIs; color corresponds to value of SNR. The dashed line is the diagonal ($f(x) = x$) of the plot. Plots correspond to: (a) original patches; as well as patches filtered with (b) Schmid filter; (c) MR filter; (d) and CLBP.

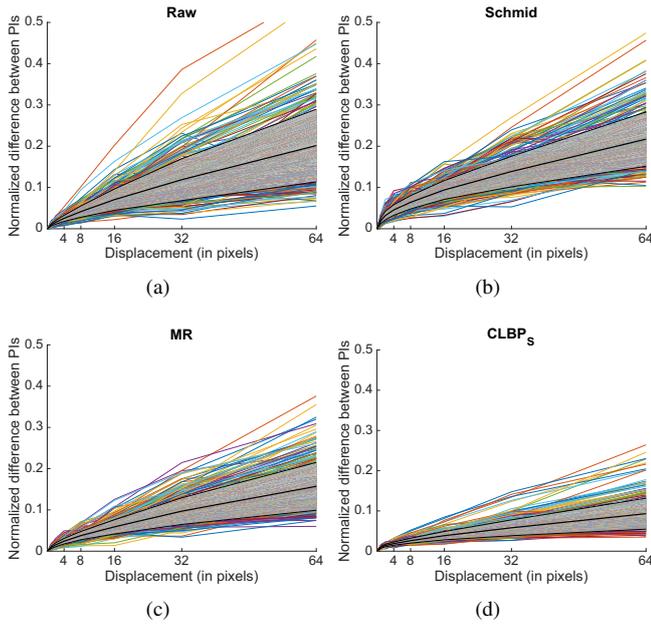


Figure 17: Displacement analysis: abscissa represents displacement between the original patch and its neighbor; ordinate represents normalized difference between their PIs. Each curve corresponds to a single patch. Middle black curve represents mean value, while the upper and lower black lines presents the standard deviation, i.e. the grayish area represents the interval of mean \pm standard deviation. Plots correspond to: (a) original patches; as well as patches filtered with (b) Schmid filter; (c) MR filter; (d) and CLBP.

which shows that CLBP contributes to the robustness of the resulting PI.

Furthermore, we investigate how sensitive PIs are towards displacement of the underlying patches. For this purpose we compute the differences in the PIs stemming from a pair of neighboring and overlapping patches. Again, we randomly select number of patches. Next we shift the patches by 4, 8, 16, 32, and 64 pixels. From the resulting 5 pairs of patches we compute PIs and compute their normalized difference as above. In Fig. 17a-d we plot the differences in PIs for all displacements as separate lines for each patch. The trend of the curves shows that the difference between the PIs grows with the amount of displacement in all four cases. Interestingly, the differences are noticeably smaller in the case of CLBP, which indicates that it introduces an additional robustness to displacements to the PI which is not the case for Schmid and MR filters.

We observe, that our experiments confirm theoretical results from Section 3.6. Furthermore, the computational results show that the constants in the stability theorems have only little influence on the results.

5.13. Discriminativity

An important question in the context of PI is if all entries in the PI (pixels) are of equal importance or if some pixels are more important than others. Experiments of feature selection in Section 5.8 have indicated that a small set of pixels exist that are more important than the other pixels. To determine the importance of the individual pixels we compute for each pixel

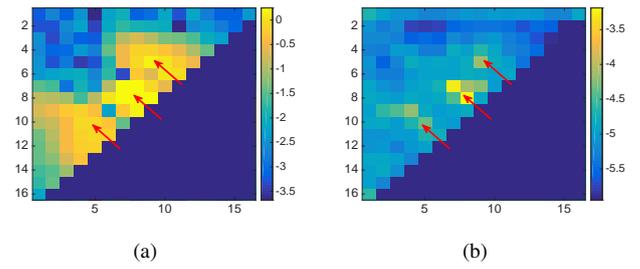


Figure 18: Importance of PI pixels in the classification process, measured with: (a) Fisher discriminant; (b) and Gini importance. The scale in both cases is logarithmic (base 10). Brighter pixels are more important (discriminative) in classification. Red arrows correspond to three groups of important pixels.

(a) the Fisher discriminant and (b) the Gini importance for the entire training set. Fig 18 shows the resulting images where each pixel in the PI is colored according to its importance.

It turns out that the most important pixels are those located in the center of the PI quite near to the diagonal. According to Fisher criterion there are also two smaller groups of important pixels, above and below the center. This is less visible but also present in case of Gini, see red arrows in Fig. 18b.

In order to further understand those results, we generated average PIs for the investigated classes of surface texture separately, see Fig. 19. The averaged PIs clearly show that the two classes exhibit highly different spatial patterns in the PIs which strongly correlate with the three groups of pixels identified with Fisher discriminant and Gini importance. This shows that PI captures the different classes of surface textures well.

The results are further consistent with those obtained for the synthetic dataset (see Section 5.1). Class 1 representing engraved areas has bimodal distribution, while class 2 representing the natural rock surface is unimodal.

The analysis on discriminativity further provides insights into the question of weighting in PI computation (see Section 5.6). In general, PI requires a proper weighting function for stability reasons (see [32]). Our analysis on discriminativity shows, however, that the components near the diagonal are most important and thus most probably do not represent noise. The reason for this is that the 3D surfaces have been smoothed and filtered for outliers prior to our processing. Thus noise (which usually appears along the diagonal of PD) has already been removed and all the remaining points actually represent important topological information. In our case, points near to the diagonal represent small surface deviations that correspond to the fine-structure of the surface and that are of high importance for surface texture classification (see Fig. 18). Thus, using weighting would be counterproductive in our case. In other situations, it is possible to avoid instability issues with PI by using another approach for constructing PI, such as the one described by Eqn. (9) in [31], which reduces the influence of the low-persistence points close to the diagonal when the scale σ increases.

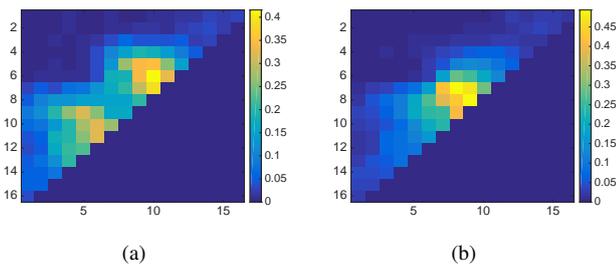


Figure 19: Averaged PIs of the training patches from (a) class 1 and (b) class 2.

6. Conclusions

In this article we have presented an extensive study on topological descriptors for 3D surface texture analysis. From patches of the surfaces' depth maps we extract different topological descriptors which build upon the persistence diagram. The descriptors have fixed dimensions and can thus easily be combined with conventional classifiers as well as with traditional non-topological descriptors. We perform experiments on surface texture classification and investigate different aspects of the topological descriptors, such as sensitivity to parameters, robustness to noise, their discriminative power and the information density. Moreover, we propose and evaluate different normalization strategies, pre-filtering, and alternative weighting functions and investigate the complementary nature of topological and non-topological descriptors. The major conclusions drawn from our study are:

- PD_AGG and PI capture relevant and discriminative information on surface textures, whereby our study clearly shows that the spatial information from the PD preserved in PI leads to a more powerful descriptor.
- PIs exhibit a low sensitivity to changes in resolution and sigma whereas a certain minimum resolution must be ensured to obtain an expressive descriptor. Sigma shows to be independent from resolution to a large extent.
- For 3D surfaces with varying depth ranges, depth normalization is essential to obtain comparable topological descriptors. In our experiments global normalization leads to the best results.
- While related works propose LBP-based representations our study shows that such a pre-filtering does not necessarily improve results in practice.
- Contrary to our expectations, it turned out that the most important information in PI is located near to the diagonal of the PD. A deeper analysis reveals that the most expressive information is spatially clustered in the PI and that these clusters are characteristic for the investigated classes.

A number of open questions and future research directions were identified in our study. Major questions include: (i) how

can weighting in PI construction be improved in a way that characteristics of the underlying data are taken into account? (ii) how can the resolution of PI be improved in areas where the most important information is concentrated to increase the expressiveness of PI? (iii) do more powerful pre-filter functions that build upon topological attributes such as [66] improve the descriptors and can we learn them from the data?

Overall our investigation shows that topological descriptors can compete with non-topological descriptors for surface texture classification. A comparison with [11] shows that our approach can compete with (and even slightly outperform) the performance of a CNN on the same dataset. Topological and non-topological descriptors achieve a significant improvement of performance when they are combined with each other. This confirms that topological descriptors capture relevant information that is not captured by traditional descriptors. Thus, they are a promising approach for further research.

7. Acknowledgments

Parts of the work for this article have 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. Other parts of the work from this article were supported by the National Science Centre, Poland under grant agreement no 2015/19/D/ST6/01215; 2016-2019.

References

References

- [1] D. Crandall, A. Owens, N. Snavely, D. Huttenlocher, Discrete-continuous optimization for large-scale structure from motion, in: *Computer Vision and Pattern Recognition (CVPR)*, 2011 IEEE Conference on, IEEE, pp. 3001-3008.
- [2] C. Wu, Towards linear-time incremental structure from motion, in: *3DTV-Conference*, 2013 International Conference on, IEEE, pp. 127-134.
- [3] J. Wohlfeil, B. Strackenbrock, I. Kosyik, Automated high resolution 3d reconstruction of cultural heritage using multi-scale sensor systems and semi-global matching, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-4 W 4 (2013) 37-43.
- [4] M. Tuceryan, A. Jain, Texture analysis, *The handbook of pattern recognition and computer vision 2* (1998) 207-248.
- [5] ASME, ASME B46.1-1995 Surface texture (surface roughness, waviness and lay): An American National Standard, The American Society of Mechanical Engineers (ASME), New York, USA, 1996.
- [6] L. Blunt, X. Jiang, Advanced techniques for assessment surface topography: development of a basis for 3D surface texture standards "surfstand", Elsevier, 2003.
- [7] J. Barcelo, V. Moitinho de Almeida, Functional analysis from visual and non-visual data. an artificial intelligence approach, *Mediterranean Archaeology and Archaeometry* 12 (2012) 273-321.
- [8] M. Zeppelzauer, M. Seidl, Efficient image-space extraction and representation of 3d surface topography, in: *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, IEEE, Quebec, Canada, 2015.
- [9] M. Zeppelzauer, B. Zieliński, M. Juda, M. Seidl, Topological descriptors for 3d surface analysis, in: *International Workshop on Computational Topology in Image Context*, Springer, 2016, pp. 77-87.
- [10] B. T. Fasy, F. Lecci, A. Rinaldo, L. Wasserman, S. Balakrishnan, A. Singh, et al., Confidence sets for persistence diagrams, *The Annals of Statistics* 42 (2014) 2301-2339.

- [11] G. Poier, M. Seidl, M. Zeppelzauer, C. Reinbacher, M. Schaich, G. Bellando, A. Marretta, H. Bischof, The 3d-pitoti dataset: A dataset for high-resolution 3d surface segmentation, in: Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing, ACM, Firenze, Italy, 2017.
- [12] A. E. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3d scenes, *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 21 (1999) 433–449.
- [13] T. Darom, Y. Keller, Scale-invariant features for 3-d mesh models, *Image Processing*, IEEE Transactions on 21 (2012) 2758–2769.
- [14] A. Zaharescu, E. Boyer, K. Varanasi, R. Horaud, Surface feature detection and description with applications to mesh matching, in: *Computer Vision and Pattern Recognition*, 2009. CVPR 2009. IEEE Conference on, IEEE, pp. 373–380.
- [15] B. Steder, R. B. Rusu, K. Konolige, W. Burgard, Point feature extraction on 3d range scans taking into account object boundaries, in: *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2011, pp. 2601–2608.
- [16] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 24 (2002) 509–522.
- [17] A. Frome, D. Huber, R. Kolluri, T. Bülow, J. Malik, Recognizing objects in range data using regional point descriptors, in: *Computer Vision-ECCV 2004*, Springer, 2004, pp. 224–237. 00401.
- [18] R. B. Rusu, Z. C. Marton, N. Blodow, M. Beetz, Persistent point feature histograms for 3d point clouds, in: *Proc 10th Int Conf Intel Autonomous Syst (IAS-10)*, Baden-Baden, Germany, IOS Press, 2008, pp. 119–128.
- [19] R. B. Rusu, N. Blodow, M. Beetz, Fast point feature histograms (FPFH) for 3d registration, in: *Robotics and Automation*, 2009. ICRA'09. IEEE International Conference on, IEEE, 2009, pp. 3212–3217. 00320.
- [20] E. Wahl, U. Hillenbrand, G. Hirzinger, Surflet-pair-relation histograms: a statistical 3d-shape representation for rapid classification, in: *3-D Digital Imaging and Modeling*, 2003. 3DIM 2003. Proceedings. Fourth International Conference on, IEEE, 2003, pp. 474–481.
- [21] F. Tombari, S. Salti, L. Di Stefano, Unique signatures of histograms for local surface description, in: *Computer Vision-ECCV 2010*, Springer, 2010, pp. 356–369. 00160.
- [22] A. Othmani, L. Lew Yan Voon, C. Stolz, A. Piboule, Single tree species classification from terrestrial laser scanning data for forest inventory, *Pattern Recognition Letters* 34 (2013) 2144–2150.
- [23] T. Leung, J. Malik, Detecting, localizing and grouping repeated scene elements from an image, in: *European Conference on Computer Vision*, Springer, 1996, pp. 546–555.
- [24] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: *Workshop on statistical learning in computer vision*, ECCV, 1-22, Prague, 2004, pp. 1–2.
- [25] F. Perronnin, C. Dance, Fisher kernels on visual vocabularies for image categorization, in: *2007 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 1–8.
- [26] M. Cimpoi, S. Maji, I. Kokkinos, A. Vedaldi, Deep filter banks for texture recognition, description, and segmentation, *International Journal of Computer Vision* 118 (2016) 65–94.
- [27] P. Frosini, Measuring shapes by size functions, in: *Intelligent Robots and Computer Vision X: Algorithms and Techniques*, International Society for Optics and Photonics, 1992, pp. 122–133.
- [28] A. Verri, C. Uras, P. Frosini, M. Ferri, On the use of size functions for shape analysis, *Biological Cybernetics* 70 (1993) 99–107.
- [29] H. Edelsbrunner, D. Letscher, A. Zomorodian, Topological persistence and simplification, *Discrete and Computational Geometry* 28 (2002) 511–533.
- [30] C. Li, M. Ovsjanikov, F. Chazal, Persistence-based structural recognition, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2014, pp. 2003–2010.
- [31] J. Reininghaus, S. Huber, U. Bauer, R. Kwitt, A stable multi-scale kernel for topological machine learning, in: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4741–4748.
- [32] H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, L. Ziegelmeier, Persistence images: A stable vector representation of persistent homology, *Journal of Machine Learning Research* 18 (2017) 1–35.
- [33] M. Ferri, P. Frosini, A. Lovato, C. Zambelli, Point selection: A new comparison scheme for size functions (with an application to monogram recognition), in: *Computer Vision — ACCV'98: Third Asian Conference on Computer Vision Hong Kong, China, January 8–10, 1998 Proceedings*, Volume 1, Springer, 1998, pp. 329–337.
- [34] P. Donatini, P. Frosini, A. Lovato, Size functions for signature recognition, in: *Proc. SPIE*, volume 3454, pp. 178–183.
- [35] H. J. Poincaré, Sur le probleme des trois corps et les équations de la dynamique, *Acta Mathematica* 13 (1890) 1–270.
- [36] H. J. Poincaré, Les méthodes nouvelles de la mécanique céleste, Gauthiers-Villars, Paris (1892, 1893, 1899).
- [37] H. J. Poincaré, *Analysis situs*, J. Éc. Polytech., ser. 2 1 (1895) 1–123.
- [38] P. Bubenik, Statistical topological data analysis using persistence landscapes., *Journal of Machine Learning Research* 16 (2015) 77–102.
- [39] N. Makarenko, M. Kalimoldayev, I. Pak, A. Yessenaliyeva, Texture recognition by the methods of topological data analysis, *Open Engineering* 6 (2016).
- [40] C. Schmid, Constructing models for content-based image retrieval, in: *Computer Vision and Pattern Recognition*, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, volume 2, IEEE, pp. II–39.
- [41] J.-M. Geusebroek, A. W. Smeulders, J. Van de Weijer, Fast anisotropic gauss filtering, *IEEE Transactions on Image Processing* 12 (2003) 938–943.
- [42] Z. Guo, L. Zhang, D. Zhang, A completed modeling of local binary pattern operator for texture classification, *IEEE Transactions on Image Processing* 19 (2010) 1657–1663.
- [43] D. Cohen-Steiner, H. Edelsbrunner, J. Harer, Stability of persistence diagrams, *Discrete & Computational Geometry* 37 (2007) 103–120.
- [44] D. Cohen-Steiner, H. Edelsbrunner, J. Harer, Y. Mileyko, Lipschitz functions have lp-stable persistence, *Foundations of Computational Mathematics* 10 (2010) 127–139.
- [45] N. Garcia, E. Stoll, Monte carlo calculation for electromagnetic-wave scattering from random rough surfaces, *Physical review letters* 52 (1984) 1798.
- [46] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, A. Napolitano, Rusboost: A hybrid approach to alleviating class imbalance, *Systems, Man and Cybernetics, Part A: Systems and Humans*, IEEE Transactions on 40 (2010) 185–197.
- [47] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of computer and system sciences* 55 (1997) 119–139.
- [48] G. Fumera, F. Roli, Cost-sensitive learning in support vector machines, *VIII Convegno Associazione Italiana per LIntelligenza Artificiale* (2002).
- [49] A. Vedaldi, B. Fulkerson, Vlfeat: An open and portable library of computer vision algorithms, in: *Proceedings of the International Conference on Multimedia*, ACM, 2010, pp. 1469–1472.
- [50] M. Juda, M. Mrozek, P. Brendel, H. Wagner, et al., Capd::redhom, <http://redhom.ii.uj.edu.pl>, 2010-2015.
- [51] M. Juda, M. Mrozek, Capd:: Redhom v2-homology software based on reduction algorithms, in: *Mathematical Software-ICMS 2014*, Springer, 2014, pp. 160–166.
- [52] U. Bauer, M. Kerber, J. Reininghaus, Phat - persistent homology algorithms toolbox, <https://bitbucket.org/phat-code/phat>, 2013.
- [53] U. Bauer, M. Kerber, J. Reininghaus, H. Wagner, Phat - persistent homology algorithms toolbox, in: H. Hong, C. Yap (Eds.), *Mathematical Software - ICMS 2014*, volume 8592 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2014, pp. 137–143.
- [54] G. Zhenhua, L. Zhang, D. Zhang, Clbp implementation for matlab, <http://www.comp.polyu.edu.hk/~cslzhang/code/CLBP.rar>, 2010.
- [55] L. Breiman, Random forests, *Machine learning* 45 (2001) 5–32.
- [56] R. O. Duda, P. E. Hart, et al., *Pattern classification and scene analysis*, volume 3, Wiley New York, 1973.
- [57] ISO-IEC, Information Technology - Multimedia Content Description Interface, 15938, ISO/IEC, Moving Pictures Expert Group, 1st edition, 2002.
- [58] D. G. Lowe, Distinctive image features from scale-invariant keypoints, *International journal of computer vision* 60 (2004) 91–110.
- [59] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, *Pattern recognition* 29 (1996) 51–59.
- [60] N. Dalal, B. Triggs, Histograms of oriented gradients for human detec-

- tion, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, IEEE, pp. 886–893.
- [61] R. M. Haralick, K. Shanmugam, I. H. Dinstein, Textural features for image classification, *Systems, Man and Cybernetics, IEEE Transactions on* (1973) 610–621.
- [62] A. Vedaldi, B. Fulkerson, VLFeat: An open and portable library of computer vision algorithms, <http://www.vlfeat.org/>, 2008.
- [63] M. Heikkilae, T. Ahonen, A general local binary pattern (lbp) implementation for matlab, <http://www.cse.oulu.fi/CMV/Downloads/LBPmatlab>, 2014.
- [64] A. Uppuluri, Glcm texture features, <http://de.mathworks.com/matlabcentral/fileexchange/22187-glcm-texture-features>, 2008.
- [65] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440.
- [66] T. P. Nguyen, A. Manzanera, W. G. Kropatsch, Impact of topology-related attributes from local binary patterns on texture classification, in: *Computer Vision - ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part II*, Springer International Publishing, 2015, pp. 80–93.