



mpLBP: A point-based representation for surface pattern description

Elia Moscoso Thompson, Silvia Biasotti, Julie Digne, Raphaëlle Chaine

► To cite this version:

Elia Moscoso Thompson, Silvia Biasotti, Julie Digne, Raphaëlle Chaine. mpLBP: A point-based representation for surface pattern description. Computers and Graphics, 2020, 86, pp.81-92. 10.1016/j.cag.2019.12.001 . hal-02406554

HAL Id: hal-02406554

<https://hal.science/hal-02406554>

Submitted on 12 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



mpLBP: A point-based representation for surface pattern description

Elia Moscoso Thompson^{a,*}, Silvia Biasotti^a, Julie Digne^b, Raphaëlle Chaine^b

^aIstituto di Matematica Applicata e Tecnologie Informatiche 'E. Magenes' - CNR

^bLIRIS, Université Claude Bernard Lyon 1, CNRS, France

ARTICLE INFO

Article history:

Received December 6, 2019

Keywords: Computers and Graphics,
Shape analysis, Pattern retrieval

ABSTRACT

The Local Binary Pattern (LBP) is a very popular pattern descriptor for images that is widely used to classify repeated pixel arrangements in a query image. Several extensions of the LBP to surfaces exist, for both geometric and colorimetric patterns. These methods mainly differ on the way they code the neighborhood of a point, balancing the quality of the neighborhood approximation with the computational complexity. For instance, using mesh topological neighborhoods as a surrogate for the LBP pixel neighborhood simplifies the computation, but this approach is sensitive to irregular vertex distributions and/or might require an accurate surface re-sampling. On the contrary, building an adaptive neighborhood representation based on geodesic disks is accurate and insensitive to surface bendings but it considerably increases the computational complexity. Our idea is to adopt the kd-tree structure to directly store a surface described by a set of points and to build the LBP directly on the point cloud, without considering any support mesh. Following the LBP paradigm, we define a local descriptor at each point that is further used to define a global statistical Mean Point LBP (mpLBP) descriptor. When used to compare shapes, this descriptor reaches state of the art performances, while keeping a low computational cost. Experiments on benchmarks and datasets from real world objects are provided altogether with the analysis of the algorithm parameters, property and descriptor robustness.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

In the plethora of distinctive elements of a 3D model, relief and color patterns are crucial aspects for many applications that require a local object characterization. Examples of these applications are the recognition of natural structures, like trees [1], the analysis of artworks styles [2], the classification of fabric patterns [3] or the categorization of objects [4].

Patterns, as meant in this work, are decorative elements that are defined by small corrugations of the surface or simple color

arrangements repeated on the surface. We distinguish two types of patterns: *geometric* patterns that represent small variations on the surface geometry, e.g., *repeated*, *small* incisions, chiselings, bumps, etc.; and *colorimetric* ones, e.g., elements with small painted decorations on the surface. Figure 1 shows examples of artworks and design objects characterized by geometric and/or colorimetric patterns. Note that a single element or an element repeated only twice (like the nose or the eyes of a statue) do not represent a pattern. Patterns are among the main factors when characterizing the type, material and style of natural surfaces and many kinds of object decorations, such as archaeological findings [4].

The analysis of patterns on surfaces is challenging for many reasons, mainly because of the structure of the 3D representation of the object and the difficulty in keeping the computational

*Corresponding author: Tel.: +39-010-6475-697; fax: +39-010-6475-660;
e-mail: silvia.biasotti@ge.imati.cnr.it (Silvia Biasotti),
julie.digne@liris.cnrs.fr (Julie Digne),
raphaelle.chaine@liris.cnrs.fr (Raphaëlle Chaine),
elia.moscoso@ge.imati.cnr.it (Elia Moscoso Thompson)

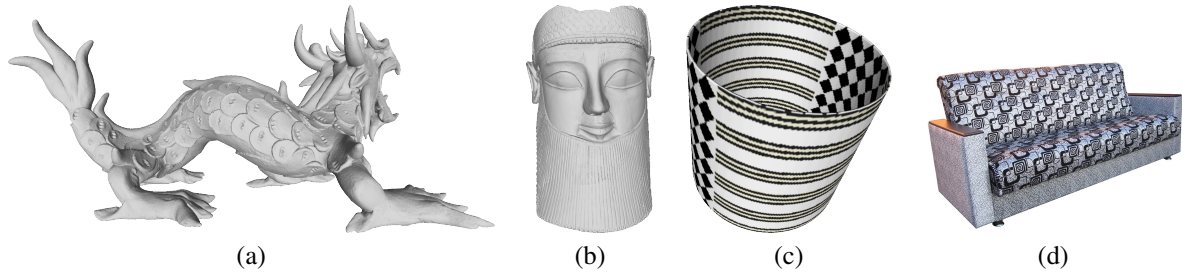


Fig. 1. Examples of two surfaces with geometric patterns (a-b) and two surfaces with colorimetric ones (c-d).

cost of the analysis low enough, to be usable in real world cases. The problem of pattern *recognition* on surfaces is still open [5]. In this paper, we tackle the simpler and related problem of *Pattern Retrieval*: we consider models fully covered by a single pattern and our goal is to identify this pattern. Taking inspiration from the LBP [6, 7], which was introduced to characterize the binary distribution of the intensities on a ring around one pixel of an image, it is possible to extend it to tackle the problem of surface pattern retrieval.

To the best of our knowledge, this has been done in recent years by different authors, culminating in three different operators: the meshLBP [8, 9], the edgeLBP [10, 11] and the Mean Point Local Binary Patterns (*mpLBP* for short) [12]. All these methods show that it is possible to tackle the pattern retrieval and classification problem using the LBP, with different degrees of success. Even considering other approaches (e.g., the SIFT + Fisher Vector on circular patches adopted in [13]), these methods perform well, marking what is currently the state of the art in this research field. The mpLBP, in particular, yields excellent performance scores while keeping low the computational cost. The mpLBP operator defines a LBP-based descriptor able to deal with surfaces represented by sets of points. If the surface is given as a tessellation, this set of points can be the set of vertices, possibly supplemented by additional points sampled on the faces if the number of vertices is low (see Section 3). These points are organized in a kd-tree structure, which makes the navigation in the model representation easier and quicker [14].

This paper extends [12], providing more discussions on the robustness of the mpLBP descriptor, also considering different surface bendings and presenting the mpLBP performance on models obtained from scans of archaeological fragments. Moreover, we analyze the efficiency of the descriptor and its characteristics when different neighborhood shapes and sampling rules are chosen.

The remainder of this paper is organized as follows. Section 2 gives an overview of previous research for the retrieval and classification of patterns over surfaces. Section 3 introduces the punctual operator at the basis of the description, the mpLBP descriptor and presents four possible variants for the ring sampling. Section 4 presents the mpLBP experimental settings, introducing both the datasets and performance measures and the shape properties adopted for the mpLBP computation. Section 5 presents the results of all the tests performed on the mpLBP. In particular, Section 5.1 shows the retrieval and classification performance of the method on two benchmarks [3, 15]

and over a set of scans of archaeological fragments. Section 5.2 focuses on the method robustness with respect to noise and different surface bendings, while Section 5.3 analyzes on the mpLBP performances with different punctual descriptor sampling schemes. Discussions, concluding remarks and feature works are provided in Section 6.

2. State of the Art

The retrieval and classification of reliefs and textures on surfaces can be seen as an extension to surfaces of the texture image retrieval problem. A large variety of methods for texture image analysis has been proposed in the literature. The main challenge for the detection of specific texture patterns is the recognition of the texture properties robustly to the possible variations [16]. A typical strategy to detect patterns on images is to consider local patches that describe the behavior of the texture around pixels. Examples of statistical descriptions are the Local Binary Patterns (LBP) [6, 7], the Scale Invariant Feature Transform (SIFT) [17] and the Histogram of Oriented Gradients (HOG) [18]. LBP-based methods are very popular and a large number of LBP variants has been proposed [19]. An extended taxonomy of 32 LBP variations and their performance evaluation for texture classification has been proposed in [20] where the LBP variations and 8 convolutional network based features are evaluated over 13 datasets of 2D images. Among the LBP variations considered, the overall best performances are obtained by the so-called Median Robust Extended LBP (MRELBP) that evaluates the descriptor over representative regions instead of single pixels. In terms of absolute performances, the method based on CNN and Fisher Vectors [21] obtains the best results but has a considerably higher computational complexity. In parallel, the aggregation of significant feature points obtained by pooling the point descriptors, e.g. SIFT+Fisher Vectors, was evaluated and obtained significant texture classification performances [16]. Similarly to LBP, the combination of a SIFT-based feature description with Convolutional Neural Networks outperforms the feature-based descriptions on classic benchmarks approximately by 10% at the cost of a higher computational complexity, [21].

For the characterization of patterns over surfaces, two strategies have been adopted so far: (i) a reduction of the problem to an image pattern one, for instance with the projection of the data onto an well chosen plane (image) and the application of an image pattern recognition algorithm to the projected data; (ii) the definition of the pattern description directly on the surface,

fact which is not straightforward because it involves handling of three-dimensional data.

As an example of reduction strategy, the method in [1] for tree species classification represents the geometric variations of the tree trunk models with a 3D deviation map over a best fitting cylinder obtained with the Principal Component Analysis (PCA) technique. Then, the cylinder is flattened on a plane and the geometric textures are compared using variations of the complex wavelet transform. Similarly, [2] adopts a height map to project the reliefs and engravings of rock artifacts into an image and classify them. The LBPI and CMC approaches proposed in the SHREC'17 contest [3] adopt, respectively, an image pattern method over a depth-buffer projection of the surface (LBPI) and the comparison of the principal curvatures in the mesh vertices using morphological image analysis techniques (CMC). Recently, [13] has proposed to use an opportune parametrization around a patch centroid to project the mean curvature values into an image and then, to adopt the SIFT + Fisher Vector [16] strategy to compare the parametric images.

The Mesh Local Binary Pattern (meshLBP) approach [22, 8, 9, 23] proposed the first extension of the LBP description [6] to triangle meshes. The main idea behind the meshLBP is that triangles play the role of pixels; there, the 8-neighborhood connectivity of images is ideally substituted by a 6-neighborhood connectivity of the vertices. The role of the gray-scale color is replaced by a function that is meant to capture the main pattern characteristics (usually Gaussian or mean curvatures, shape index [24] or mesh color if such information is provided). The edgeLBP [10, 25, 11, 3, 15, 4] performs an LBP evaluation that is based on the rings built over the mesh edges. Adopting a surface-aware characterization the edgeLBP is able to outperform the meshLBP in terms of quality of the query results but it pays a higher computational cost. To overcome the computational limitation of the edgeLBP, the mpLBP has been introduced in [12]. In this case, to speed the extraction of the descriptor, the mpLBP takes advantage of the kd-tree structure and therefore it deals directly with point clouds.

For point clouds, local surface patches can also be constructed by regression using the neighborhood around one point [26, 27, 28, 29] and those patches can be compared in the parameter space. In most recent approaches, the surface was locally characterized as a digitized height field over the regression surface which may be a plane [30] or a quadric (see [31] for an application to super-resolution).

3. mpLBP descriptor

This section introduces a statistical descriptor that is global to the shape, that aggregates values of local descriptors computed at a set of positions on the surface. The input surface model can be a point cloud or a triangulation (in the last case, the vertices are input points for the descriptor).

The mpLBP procedure can be described by two main steps: the creation of the punctual descriptor (Section 3.1) and LBP evaluations that are further combined to create the mpLBP descriptor (Section 3.2). In Section 3.3 we discuss the definition and tuning of the parameters of the method.

3.1. Punctual descriptor

Let S be a point set embedded in the 3D Euclidean space and a surface property defined on S , $h : S \rightarrow \mathbb{R}$, a function defined on S whose values depends on the pattern we want to describe (e.g.: curvature-based values in case of geometric patterns, a color-based property in case of depicted decorations, etc.). Let us consider the point $\tilde{p} \in S$ and the set $S[\tilde{p}]$ of the points $p_i \in S$ at a distance from \tilde{p} at most equal to R , i.e., $S[\tilde{p}] = \{p_i \in S | d(\tilde{p}, p_i) \leq R\}$. We will discuss the choice of the radius R in Section 3.3. Gathering the sets $S[\tilde{p}]$ means visiting the points of S several times. Since 3D models with patterns must be at high resolution (thus described by a high number of points), the way the distance relations between points are computed must be efficient. In our implementation, these relations are computed using a kd-tree. This structure is computed once per model (with a computational cost of $n \log(n)$).

Points in $S[\tilde{p}]$ are projected on a plane π , obtained using linear regression on $S[\tilde{p}]$. When the density is high enough and if the radius is chosen carefully, that plane may be interpreted as an approximation of the tangent plane. The projected points are sorted in n_{rad} concentric rings based on their distances from \tilde{p} .

The number of rings is given by the parameter n_{rad} , that we call *radial resolution*. Each ring is defined, for $j = 1 \dots n_{rad}$, as follows:

$$S[\tilde{p}]_j = \{p_i \in S[\tilde{p}] | d(\tilde{p}, p_i) \in [R_{j-1}, R_j]\}, \quad R_j = j \frac{R}{n_{rad}}$$

Each $S[\tilde{p}]_j$ is divided in P_j sectors, delimited by some regularly spaced angle values θ_k . Note that P_j may vary along the rings, in order to obtain sectors with similar areas. We call P_j the *spatial resolution*. More formally, we define the sector k of the ring j (sector (j, k) for short) of the point \tilde{p} as:

$$S[\tilde{p}]_j^k = \{p_i \in S[\tilde{p}] | d(\tilde{p}, p_i) \in (R_{j-1}, R_j], \theta_i \in (\theta_{k-1}, \theta_k]\},$$

where $\theta_k = k \frac{2\pi}{P_j}$, $k = 1 \dots P_j$. Finally, we assign to each sector (j, k) a value $sec(\tilde{p})_j^k$ as the representative of the function h in that sector. Figure 2 represents the pipeline to build the punctual descriptor. Note that the punctual descriptor can be seen as a feature vector by simply stacking the values of the descriptor on each ring.

As it usually happens in the LBP implementations, we excluded the computation of the punctual descriptor at points that are close to the boundary of the model (if any). If the boundary of the model is known, it is enough to consider only the points that are at least at distance R from the boundary. In addition, if a point punctual descriptor has more than $\frac{1}{4} \sum_j P_j$ empty sectors, we considered it *invalid* and discard that point. When the intersection of the sphere of radius R with the point cloud generates multiple surface components like those in Figure 3 (Right), we consider such a configuration non acceptable and refine the point neighborhood by selecting a smaller value for R . Indeed, for a given model M we assume that the projection onto π is injective and that the surface locally captured by the sphere is locally homeomorphic to a topological disk. Moreover, we assume the existence of a radius \tilde{R}_{max} , which is the maximum value for the parameter R such that all the points on M have an acceptable description.

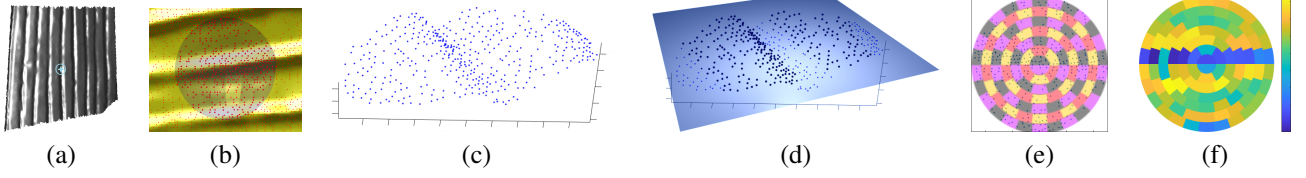


Fig. 2. A mpLBP descriptor at point \tilde{p} (marked with a light-blue star in (a)). (b): neighborhood $S[\tilde{p}]$ of \tilde{p} is shown with a dark sphere. (c): point density in $S[\tilde{p}]$. (d): regression plane π , (e): clustering into sectors. (f): resulting punctual descriptor, represented as a 'circular' feature vector.

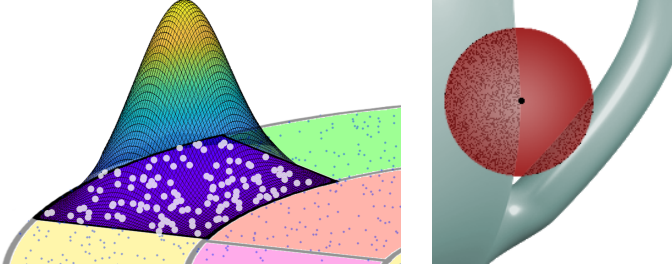


Fig. 3. Left: illustration of the Gaussian filter adopted to weight the points (in white) in a given sector (in purple). The colors of the Gaussian range from blue (0) to yellow (1). Right: example of a neighborhood that could occur if the radius R is larger than \tilde{R}_{max} : points are samples on two disconnected parts.

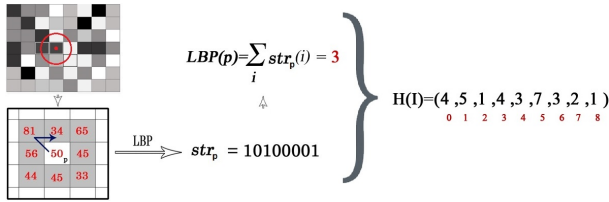


Fig. 4. LBP evaluation for an image. Top-left: in red, the pixel p is highlighted with a dot together with the circle of radius R centered at p . The values of h around p are reported in the bottom-left image.

3.2. Local Binary Pattern evaluation

The punctual descriptor introduced in Section 3.1 is the point neighborhood representation to which we apply the LBP encoding technique.

The LBP paradigm is very popular for images and many versions are available [19]. We extend the LBP description to surfaces following closely the image approach [7] which we briefly summarize. For each pixel p , the set of pixels \tilde{p}_j within distance R from p is called a *ring* of pixels. Visiting each ring from the top-left pixel in counterclockwise order, a binary array with as many elements as the pixels in the ring is created, adding 0 if $h(\tilde{p}_j) \leq h(p)$ and 1 otherwise. Then the LBP value of p is the sum of the numbers in the binary array (it varies from 0 to the number of pixels in the ring). Note that, in this context, it does not matter the order of the comparisons p and all the \tilde{p}_j . The histogram H of the LBP values for all the image pixels is the LBP descriptor of the image. Figure 4 shows this process for a single pixel (Left) and a possible final descriptor (Right). Multiple rings can be considered, increasing the size and descriptive capability of the descriptor.

In our case we consider \tilde{p} defined as in Section 3.1. If the radius R is small enough with respect to the curvature and the

thickness of the object, we can suppose that the rings of the punctual descriptor are locally close to concentric rings using geodesic distance to \tilde{p} . Thus, each sector can be seen as the evaluation of h at a sample of the surface. For all the points \tilde{p} in S , we define $LBP(\tilde{p})$ the feature vector of n_{rad} elements as follows:

$$LBP(\tilde{p})_j = \sum_k (str[\tilde{p}]_j)_k,$$

$$(str[\tilde{p}]_j)_k = \begin{cases} 0 & \text{if } sec(\tilde{p})_j^k < h(\tilde{p}) \\ 1 & \text{otherwise} \end{cases}$$

Then, the mpLBP descriptor of S ($mpLBP(S)$) is the histogram of the LBP values of the points of S . As a final step, the mpLBP is normalized, i.e., all the entries of $mpLBP(S)$ are divided by the number of points considered in the histogram, enhancing the stability of the descriptor.

The $mpLBP(S)$ is a $\sum_j (P_j + 1)$ sized feature vector. Intuitively, we can visualize it as a horizontal concatenation of the rings of the multiple feature vectors in Figure 2(f). In particular, the j -th ring generates a feature vector of $P_j + 1$ entries, where $mpLBP(S)_{(j,m)}$ is equal to the number of points \tilde{p} in S such that $LBP(\tilde{p})_j = m$ (with $j = 1, \dots, n_{rad}$ and $m = 0, \dots, P_j$).

It is worth mentioning that if the neighborhood of a point is rotated significantly around the normal of the point, its punctual descriptor changes. On the contrary, if the rotation is small, the punctual descriptor is stable; indeed, the Gaussian filter adopted to weight the points is stable under rotations smaller than a fraction of the angular sector. In other words, if the grid of sectors (Figure 2(e)) is slightly rotated, the punctual descriptor does not vary significantly. Moreover, we recall that the LBP value per ring (as intended in this paper) is rotation invariant (because it is a sum of 0 and 1 values on the whole ring), we can conclude that the pattern descriptor of the mpLBP is robust to rotations of the surface. This fact has been verified by applying a random rotation to each point neighborhood: the results indicate almost a perfect stability in this sense. More detail on this fact are provided in Section 5.1.

3.3. Parameter settings

The three parameters of the mpLBP are the radius R (used to set the neighborhood size around each point of S), the radial resolution n_{rad} and the spatial resolution P_j . This is similar to the parameter set of the edgeLBP as described in [10], the main difference being that for the edgeLBP the parameter P is fixed across all the rings. In the following, we present some hints on how these parameters should be tuned. The intuition suggests that the mpLBP ability of detecting a pattern depends on the size of the neighborhood of each point, i.e., the size of the disk

must be related to the pattern size. Moreover, the denser the ring sampling, the more complete information is stored, at the cost of a larger storage size.

- R : neighborhood radius shown as a dark bubble in Figure 2(b). R should be set so that neighborhoods contain at least one part of the pattern that we want to describe (e.g.: if the pattern is defined by chiseled circles, the bubble should contain at least one circle entirely).
- n_{rad} : it defines the radial resolution and should be fixed together with P_j (see below).
- P_j : it represents the spatial resolution and varies over the different rings.

Choice of the rings and sampling scheme. The original mpLBP description proposed in [12] adopts as the point description a set of circular rings such that the spatial resolutions P_j guarantee that all the sectors have the same area. For this reason, we selected $P_j = multP(2j - 1)$, $multP \in \mathbb{N}_+$. In this case, P_j depends on n_{rad} . This degree of freedom was tuned by the parameter $multP$ (that replaces the P_j parameters). For instance, in Figure 2(c) the parameters are $n_{rad} = 7$ and $multP = 2$, which means that $S[\tilde{p}]$ has 7 rings, where $S[\tilde{p}]_1$ has 2 sectors, $S[\tilde{p}]_2$ has 6 sectors and $S[\tilde{p}]_3$ has 10 sectors, etc.

However, similarly to the LBP for images, different types of rings may be used. Both the shape (like square, elliptical...) and the sampling scheme (or rather considering only part of the neighborhood) may be changed in order to better suit a given dataset. In general, anisotropic sampling schemes are a valid option in this context. Indeed, the punctual descriptor converts a patch of a surface into a sort of image. Therefore, it would make sense to straightforwardly adopt a square neighborhood and its variations, following the image literature on pattern recognition. In general, it is also possible to define various scheme variations of the standard punctual descriptor. We focus on different sampling areas of the point neighborhood. In the following, we depict some possible neighborhood and sampling strategies. In Section 5.3 we quantitatively discuss the performance of these neighborhood choices over a well-known benchmark.

- *Scheme 1*: same concept of the mpLBP punctual descriptor, but both the descriptor and the sectors are shaped as squares. Figure 5(a) shows how the square neighborhood of the points are divided in sectors. The parameters of this descriptor are half the diagonal of the square (a sort of radius, thus we still refer to it as R) and the square root of the number of sectors (or rather, the number of sectors along the sides of the square) labelled with $pxres$.
- *Scheme 2*: same as the mpLBP punctual descriptor, but only alternate sectors are kept on each ring (e.g., sectors 1, 3, etc.). In our tests we selected only odd sectors; anyway also even indices could be equally considered. In practice, adopting such a strategy we half the number of samplings.
- *Scheme 3*: similar to Scheme 1, but the first rings are kept in their entirety. The implicit assumption behind this scheme is that smaller rings need a denser sampling.

- *Scheme 4*: same as the mpLBP punctual descriptor, only the sectors with an index j such that $j = 1 + 4n$ with $n \in \mathbb{N}$, are considered. Similarly, to the Scheme 1, this strategy aims at decreasing the number of samples.

A representation of these variations are showed in Figure 5. The results of these tests are described in Section 4, see Table 5.

4. Experimental settings

The mpLBP was tested on recent benchmarks in order to evaluate its performances. Moreover, with respect to what reported in [12], we run additional tests on the benchmarks considered in the previous word and two additional datasets are considered, extracted from a collection of archaeological fragments.

4.1. Datasets

SHREC17 Benchmark, geometric patterns. The SHREC'17 benchmark dataset [3] on the retrieval of relief patterns is composed by 720 triangle meshes derived from knitted objects. There are 15 classes of models. Each class was created from the acquisition of the same textile pattern digitized in 12 different embeddings (Figure 6) yielding 180 different models. Then, each model was re-sampled four times (reaching 720 models). Two datasets were derived: the first one is composed by the original 180 models (called *Original Dataset*), while the other is made by all the 720 models (called *Complete Dataset*). The latter mainly aims at evaluating the overall robustness and stability of methods with respect to different mesh representations.

SHREC18 Benchmark, colorimetric patterns. The SHREC'18 benchmark [15] originated from 20 base models without any texture or colorimetric information to which were applied 15 gray and white texture each. By combining these, 300 models are obtained [15]. In addition, the luminosity of the textures was modified by using a random value to obtain the same pattern with 20 different shades. Around 30% of the model surfaces are covered by one of the 15 patterns and the remaining part of the surface is only black or only white. The last five patterns are mixed versions of the initial 10 patterns (see Figure 7). Two different datasets are provided: one containing only the models with a single pattern (the set of 200 models characterized by one of the first 10 patterns, called *Single pattern dataset*) and the other including all the models (all the 300 models, called *Complete dataset*).

GRAVITATE use-case. The GRAVITATE dataset was derived from laser scans of cultural heritage artifacts stored in the Science and Technology in Archaeology Research Center (STARC for short) repository [32]. The fragments came from different collections. They were selected as test beds for the Gravitate EU project [33, 34]. The models are stored as triangulations. Each model is available in several versions, each at a different level of detail, ranging from 50K to millions of vertices. The models are characterized by both colorimetric and geometric

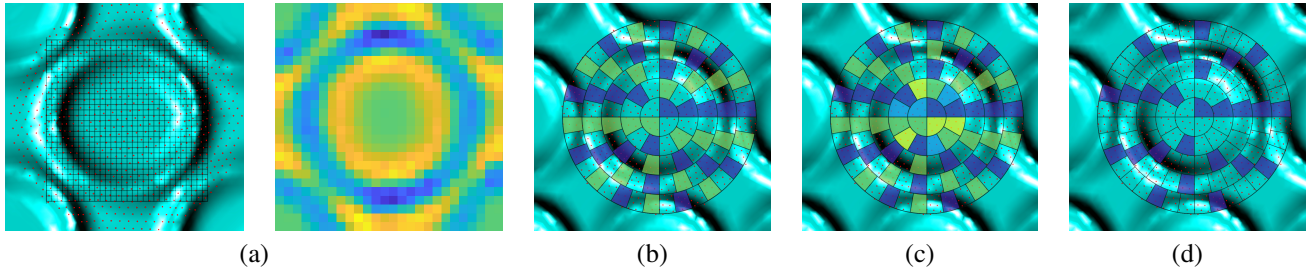


Fig. 5. Various sampling schemes. In (a) we show the grid used to create the the square punctual descriptor (Left) and the final punctual descriptor (Right). In (b,c,d) we highlight the sectors that are considered in Schemes 2,3, and 4.

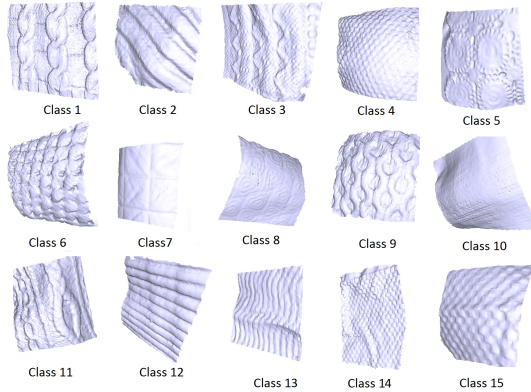


Fig. 6. The knitted patterns of the SHREC17 contest.

patterns. In case of colorimetric patterns, the RGB values are stored at each mesh vertex. In addition to the archaeological documentation, the fragments were grouped by experts and curators according to the types of patterns present on their surface, thus we use such a classification as a groundtruth to better evaluate our results [4]. By manually cutting out the patterns from the models that were considered to be significant by the experts, we created two datasets of patterns on surface [4].

- The *GRAVITATE(geo)* dataset contains 6 classes of geometric patterns, represented in Figure 8. There are 10 models per class, for a total of 60 models. Each patch has around 20K vertices.
- The *GRAVITATE(col)* dataset contains 10 classes of colorimetric patterns, represented in Figure 8. The number of elements per class varied from 4 to 7, for a total of 49 models. Every patch was made of approximately 40K vertices.

4.2. Evaluation Measures

The following evaluation measures are considered to assess our results.

Nearest Neighbor, First Tier, Second Tier. These measures check the fraction of models in the query's class that appears within the top k retrievals. By changing the value of k to 1, $|C| - 1$ and $2(|C| - 1)$, we consider, respectively, 3 measures: Nearest Neighbor (NN), First Tier (FT) and Second Tier (ST). These values range from 0 (worst result, it means that none of

the retrieved elements belong to the class of the query) to 1 (best result, it means that all the retrieved models belonging to the class of the query)

Normalized Discounted Cumulative Gains. The Discounted Cumulative Gain (DCG) derives from the Cumulative Gain, which sums the graded relevance values of all results in the list of retrieved objects of a given query. The DCG assumes that relevant items are more useful if they appear earlier in a query list. Thus, it weights the distances with respect to a relevance value. In the experiments we adopt the nDCG, which is a normalized mean of the DCG computed on each model. We used the implementation proposed in [35].

Average Precision and e-measure. The *Precision* and *Recall* are two common measures for evaluating search strategies. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database, while precision is the ratio of the number of relevant records retrieved to the size of the return vector [36]. We consider the mean Average Precision (mAP), which is the area under a precision-recall curve [37]. The e-measure e [38] was also introduced as a quality measure of the first models retrieved for every query. Formally,

$$e = \frac{2}{Precision^{-1} + Recall^{-1}}.$$

4.3. mpLBP settings

The choice of the function h used to build the punctual descriptor is driven by the kind of pattern we want to describe. In the case of geometric patterns, we may consider one of the many existing curvature-based properties, while for colorimetric information we can consider different color embeddings. While the original mpLBP relied on maximum curvature, we test other options: Minimum curvature, Mean curvature, Gaussian curvature and Shape Index. These properties are computed using the Matlab implementation in [39]. This choice was driven by the curvature measure comparisons in [40]. While this implementation works on triangulations, and estimates the curvature on vertices, it is possible to rely on [28, 41, 27] for estimating these quantities for point clouds.

Moreover, we consider the *height-field* (HF for short) as an additional approximation of the surface properties. Indeed, the HF represents a first-order local approximation of the surface. When $h = hf$, a value is assigned to each point P_i in the neighborhood of P , which is equal to the point-plane distance between the point and the plane π defined in Section 3.1. The

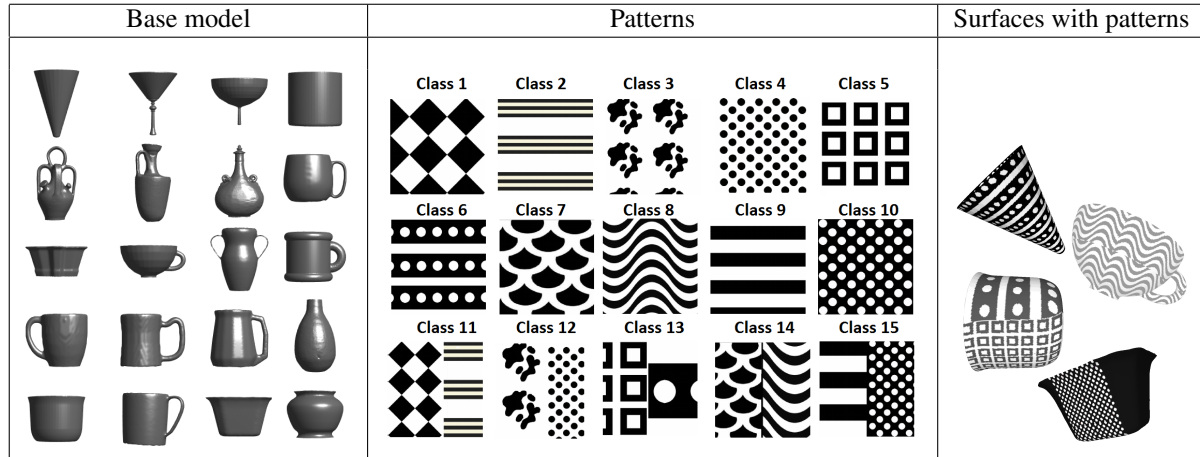


Fig. 7. The 20 base models used in the SHREC'18 benchmark are shown on the left. Each pattern (middle) was applied to all of the base models and changed in terms of their luminosity (examples of the final models are shown on the right).

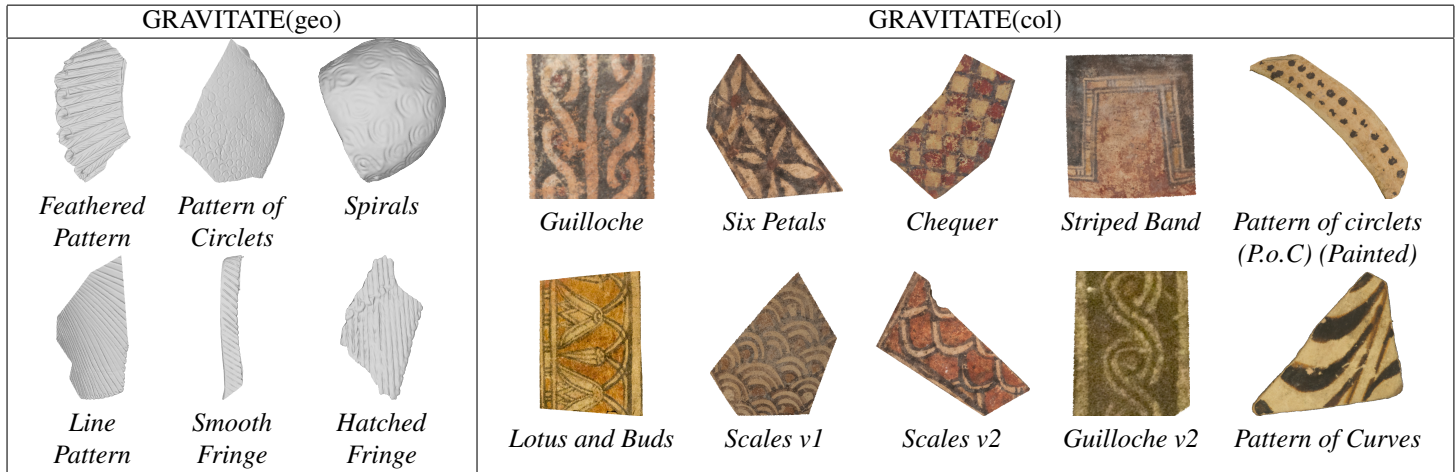


Fig. 8. One sample per each class of the GRAVITATE datasets. Class labels are given by experts of the field [4].

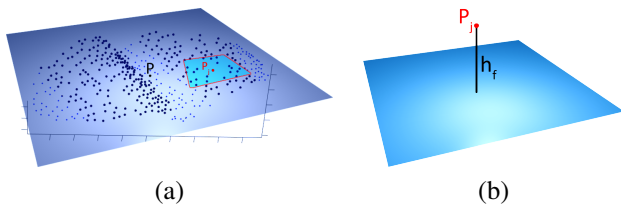


Fig. 9. Visual representation of how the height-field is computed. Starting from a point P , the plane of linear regression is computed (a) and a scalar value h_f is assigned each point P_j in the neighborhood of P (b). The area represented in (b) is the same as the light blue one on (a), with only P_j represented here for the sake of clearness.

5. Experimental results and discussions

This Section summarizes the mpLBP performances over the different datasets with the different settings described in Section 5.1. Then, in Section 5.2 we analyze the robustness of the mpLBP descriptor to surface noise, different bendings a pattern may be embedded in and different neighborhood sampling schema. Finally, Section 5.4 shows the timing comparisons between the mpLBP and the edgeLBP.

5.1. Experimental results

Before discussing the results, it is worth noticing that the evaluation scores of the mpLBP reported in this paper are slightly different than those in [12]: a regression plane alignment bug was found in our previous implementation and it is now corrected.

Performances on the SHREC'17 benchmark. In addition to the methods of [3] that obtained the best performances, we compare the mpLBP with the edgeLBP [10] and the SIFT-based method in [13]. Among all the settings tested, the best performing ones are $R = 14$, $n_{rad} = 7$ and $multP = 4$. Since most

value is negative if $\vec{n}_P \cdot \vec{P_iP} < 0$, where \vec{n}_P is the normal of the point P . A point common to the neighborhoods of two different points may have different height values with respect to the two neighborhood regression planes. Figure 9 shows how the HF is computed.

For colorimetric patterns, only the CIELab color-space [42, 43] is considered (the L-channel in particular, which encodes the luminosity of the colors).

Original Dataset						
Method	NN	1-Tier	2-Tier	mAP	e	nDCG
CMC-2	0.633	0.363	0.494	0.390	0.293	0.662
KLBO-FV-IWKS	0.522	0.295	0.412	0.307	0.247	0.603
edgeLBP - <i>run2</i>	0.911	0.689	0.844	0.725	0.590	0.865
T/mC/SIFT/FV	0.872	0.710	0.849	0.741	0.457	0.883
mpLBP - Cmax	0.933	0.706	0.845	0.744	0.449	0.871
mpLBP - Cmin	0.922	0.732	0.861	0.738	0.442	0.862
mpLBP - Cmean	0.911	0.733	0.861	0.763	0.447	0.888
mpLBP - Cmean*	0.917	0.733	0.863	0.761	0.427	0.875
mpLBP - Cmean*	0.922	0.728	0.862	0.763	0.426	0.877
mpLBP - Cgauss	0.939	0.707	0.845	0.744	0.483	0.872
mpLBP - SI	0.911	0.729	0.835	0.749	0.440	0.876
mpLBP - HF	0.672	0.353	0.451	0.431	0.281	0.658

Complete Dataset						
Method	NN	1-Tier	2-Tier	mAP	e	nDCG
CMC-2	0.763	0.272	0.389	0.271	0.261	0.686
KLBO-FV-IWKS	0.986	0.333	0.449	0.339	0.332	0.759
edgeLBP - <i>run2</i>	0.986	0.634	0.780	0.669	0.421	0.902
T/mC/SIFT/FV	0.993	0.712	0.850	0.739	0.647	0.929
mpLBP - Cmax	0.994	0.678	0.820	0.738	0.653	0.931
mpLBP - Cmin	0.997	0.677	0.815	0.733	0.653	0.931
mpLBP - Cmean	0.994	0.702	0.841	0.759	0.665	0.938
mpLBP - Cgauss	0.994	0.678	0.820	0.738	0.653	0.931
mpLBP - SI	0.997	0.688	0.813	0.737	0.650	0.930
mpLBP - HF	0.999	0.383	0.480	0.431	0.419	0.802

Table 1. Results on the SHREC'17 benchmark, both the Original (Top) and the Complete (Bottom) Dataset. Runs marked with * are those which point neighborhood are rotated by a random angle around the normal of the respective point.

of the models in the dataset have a low number of vertices, we resampled them to 40000 vertices using the Remesh tool [44]. Table 1 reports the mpLBP scores together and compares them with the other methods, with respect to NN, FT, ST, e-measure, mAP and nDCG. In this table we also include the results obtained when testing the independence from the cut of the grid on the neighborhood of the points. We mark with * the runs to which we added to each point neighborhood a small rotation of the tangent parametrization around the point normal (from 0 to 2π). The results show that adding this rotation does not influence critically the results of our method.

The mpLBP scores equivalently or slightly better than the edgeLBP and T/mC/SIFT/FV over the SHREC'17 benchmark on geometric patterns, with small variations depending on the surface property chosen. Overall, over this benchmark, the mpLBP performs well with all the curvature-based properties; in particular, the mean curvature provides slightly better retrieval performances. However, the NN performance over the complete dataset of mpLBP with the height field highlights how this property is able to characterize a model and its re-samplings but it less robust to different surface bendings.

Performances on the SHREC'18 benchmark. The performance of mpLBP on this benchmark is compared against those obtained in [15] and [11]. The parameters settings with the best evaluations are $R = 0.10$ $n_{rad} = 7$ $multP = 1$ (*set1*) and $R = 0.14$ $n_{rad} = 7$ $multP = 1$ (*set2*). Table 2 summarizes the best scores obtained (more runs and methods are available in [15] and [11]).

Over this benchmark, mpLBP and edgeLBP perform equivalently, even if the time for evaluating the edgeLBP on this dataset is approximately twenty times higher than the mpLBP (details on the computation costs are provided in Section 5.4).

Single Pattern Dataset						
Run	NN	FT	ST	mAP	e	nDCG
TWB3	0.755	0.502	0.688	0.577	0.455	0.795
V2	0.82	0.51	0.731	0.593	0.481	0.808
edgeLBP-R4	0.915	0.717	0.879	0.766	0.60	0.898
edgeLBP-R5	0.950	0.740	0.892	0.790	0.606	0.911
mpLBP - <i>set1</i>	0.965	0.739	0.862	0.781	0.600	0.910
mpLBP - <i>set2</i>	0.960	0.744	0.864	0.762	0.590	0.900

Complete Dataset						
Run	NN	FT	ST	mAP	e	nDCG
TWB3	0.593	0.417	0.564	0.460	0.376	0.711
V2	0.79	0.433	0.594	0.493	0.39	0.753
edgeLBP-R4	0.903	0.673	0.827	0.722	0.557	0.878
edgeLBP-R5	0.923	0.667	0.805	0.727	0.546	0.878
mpLBP - <i>set1</i>	0.903	0.739	0.862	0.668	0.520	0.850
mpLBP - <i>set2</i>	0.907	0.573	0.735	0.639	0.510	0.840

Table 2. Performance scores over the Single pattern dataset and Complete dataset of the SHREC'18 benchmark.

Indeed, to guarantee the decorations were intelligible, the original surfaces were densely sampled (100K vertices, each) and this corresponds to a demanding task for the edgeLBP. On the contrary, this is a good basis for mpLBP because the key issue for its success is that the point cloud is dense enough, i.e., most of the sectors of the descriptors should not be empty.

Performances on the GRAVITATE dataset. We run the mpLBP on both datasets with 4 different settings. The results are compared with those reported in [4] for edgeLBP on the GRAVITATE datasets. In particular, on GRAVITATE(geo) we used the Shape Index as h function, as it is done in [4]. In Table 3, the best mpLBP run we had is compared to the edgeLBP results.

As a general premise, since the dataset classes contain few models (less than 10), even a variation of 0.1 in the scores means that only one model is miss-classified. The performances of the edgeLBP and mpLBP are comparable, with differences that depend on the type of pattern and the quality of the model. This can be observed by the overall performances of the runs. Looking at the single class performances, the NN measures are almost identical, with the edgeLBP being slightly more efficient on the geometric patterns. The opposite is true on the colorimetric ones, where the patterns are more degraded. This may be explained by the fact that since the mpLBP scouts an entire area for each sampling (sector) rather than a single point to code the evolution of the ring properties, it slightly averages the local surface properties simulating a slight smoothing effect. This can be positive in case of noisy colorimetric patterns but it could lead to a confusion between geometric noise and actual small surface variations.

5.2. Robustness of the descriptor

In [12] we already explored the robustness of the mpLBP descriptor to noise. In this work, we also analyzed the robustness of the pattern descriptor of a fixed pattern when the latter lies on surfaces with different bendings. All this robustness analysis are reported in this Section.

Robustness to noise. The popularity of scanning devices that can digitize objects increased the number of acquired 3D models available, but not all of them have perfect sensors. Thus,

GRAVITATE(geo) results

edgeLBP settings: $n_{rad} = 7, N = 15, R = 0.6\text{cm}$						mpLBP settings: $n_{rad} = 5, multP = 4, R = 1\text{cm}$					
Class considered	NN	FT	ST	e	nDCG	Class Label	NN	FT	ST	e	nDCG
Feathered Pattern	1.000	0.722	0.911	0.434	0.934	Feathered Pattern	1.000	0.511	0.767	0.410	0.849
Pattern of Circles	1.000	0.922	0.978	0.439	0.989	Pattern of Circles	1.000	0.889	0.944	0.439	0.976
Spirals	1.000	0.644	0.756	0.385	0.911	Spirals	1.000	0.544	0.700	0.400	0.870
Line Pattern	0.700	0.300	0.367	0.263	0.661	Line Pattern	0.800	0.289	0.444	0.302	0.685
Smooth Fringe	1.000	0.467	0.500	0.229	0.804	Smooth Fringe	1.000	0.522	0.533	0.239	0.818
Hatched Fringe	1.000	0.400	0.589	0.371	0.773	Hatched Fringe	1.000	0.400	0.500	0.366	0.765
Overall	0.950	0.576	0.683	0.354	0.845	Overall	0.967	0.526	0.648	0.359	0.827

GRAVITATE(col) results

edgeLBP settings: $n_{rad} = 5, P = 15, R = 0.5\text{cm}$						mpLBP settings: $n_{rad} = 7, P = 15, R = 0.5\text{cm}$					
Class Label	NN	FT	ST	e	nDCG	Class Label	NN	FT	ST	e	nDCG
Guilloche v1	1.000	1.000	1.000	0.171	1.000	Guilloche v1	1.000	1.000	1.000	0.171	1.000
Six Petals	0.500	0.500	0.767	0.270	0.720	Six Petals	0.500	0.400	0.767	0.270	0.709
Chequer	0.857	0.571	0.738	0.271	0.811	Chequer	1.000	0.714	0.976	0.316	0.924
Striped band	0.800	0.200	0.400	0.222	0.544	Striped band	0.600	0.250	0.500	0.189	0.567
P.o.C. (Painted)	1.000	1.000	1.000	0.171	1.000	P.o.C. (Painted)	1.000	1.000	1.000	0.171	1.000
Lotus and Bud	0.500	0.333	0.583	0.171	0.666	Lotus and Bud	0.500	0.417	0.583	0.143	0.566
Scales v1	0.429	0.310	0.619	0.316	0.579	Scales v1	0.571	0.357	0.643	0.263	0.680
Scales v2	1.000	1.000	1.000	0.171	1.000	Scales v2	0.750	0.333	0.667	0.171	0.673
Guilloche v2	1.000	1.000	1.000	0.171	1.000	Guilloche v2	1.000	1.000	1.000	0.171	1.000
Pattern of Curves	0.500	0.250	0.250	0.086	0.424	Pattern of Curves	0.500	0.167	0.250	0.100	0.427
Overall	0.735	0.582	0.723	0.217	0.758	Overall	0.742	0.693	0.918	0.211	0.755

Table 3. Results on the GRAVITATE datasets, and comparison with those of the edgeLBP.

acquired data can be corrupted by acquisition noise. Furthermore, objects can be degraded by time or other factors, which can lead to corruption of the patterns that lie on the surface of the object.

Noise in general is an unwanted variation of a signal (the surface, in this case) usually of small scale. Patterns also are usually small, thus the noise is a problem that is worth addressing. We added noise with various amplitudes to some of our test models. Such variations change the h function and as a consequence the whole punctual descriptor. Depending on the pattern nature, we considered different noise addition. The geometrical patterns are corrupted with a Gaussian noise on the vertices, based on a parameter λ_g , expressed as a percentage of the diameter of the smallest bounding sphere. The values of λ_g considered are 0.2 and 0.4. See Figure 10 (Top) for an example of mesh degradation.

Colorimetric patterns are instead corrupted by adding small variations to the RGB values stored on the model vertices. Such variation is bounded to the parameter λ_c , an integer value added to each RGB channel (we assume the three channels to range from 0 to 255). For example, $\lambda_c = 5$ added three random offsets in the interval $[-5, +5]$ to each color channel. In our tests, we used $\lambda_c \in \{5, 7\}$ (see Figure 10 (Bottom)).

The noise tests are run on the SHREC'17 Original dataset for the geometric patterns and on the SHREC'18 Single pattern dataset for the colorimetric one. Results are reported in Table 4. We observe that the performances significantly decrease in presence of heavy noises, while the mpLBP is robust in case of lighter ones. We think that this behaviour derives from the strategy we adopt to evaluate a property in the sectors. Indeed, the use of the weighted mean for each sector balances the small variations of the h function, while it starts being less efficient in case of higher variations (in this case, all the sectors become similar).

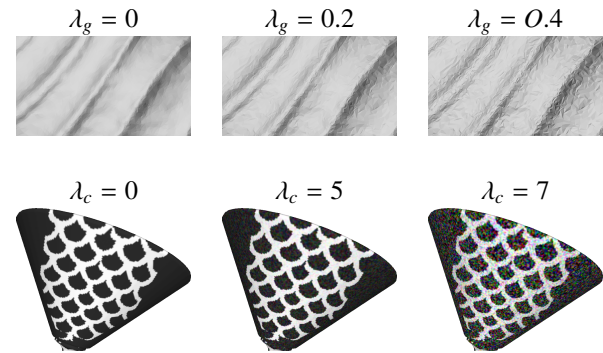


Fig. 10. Pattern distortion when noise is randomly added. Top row: a geometric pattern is corrupted using increasing Gaussian noise. Bottom row, an increasing random noise is added to each RGB color channel.

SHREC'17: Original Dataset, geometric noise						
Method	NN	1-Tier	2-Tier	mAP	e	nDCG
mpLBP - <i>set1</i> Clean	0.917	0.711	0.859	0.743	0.420	0.861
mpLBP - <i>set1</i> , $\lambda_g = 0.2$	0.911	0.693	0.846	0.733	0.380	0.790
mpLBP - <i>set1</i> , $\lambda_g = 0.4$	0.872	0.618	0.769	0.664	0.350	0.753

SHREC'18: Single Pattern Dataset, colorimetric noise						
Method	NN	1-Tier	2-Tier	mAP	e	nDCG
mpLBP - <i>set1</i> Clean	0.965	0.739	0.862	0.781	0.600	0.910
mpLBP - <i>set1</i> , $\lambda_c = 5$	0.915	0.514	0.653	0.586	0.440	0.822
mpLBP - <i>set1</i> , $\lambda_c = 7$	0.75	0.332	0.445	0.457	0.355	0.741

Table 4. mpLBP performance for data corrupted with noise. Top: the Original Dataset of the SHREC'17 benchmark, Bottom: Single Pattern Dataset of the SHREC'18 benchmark.

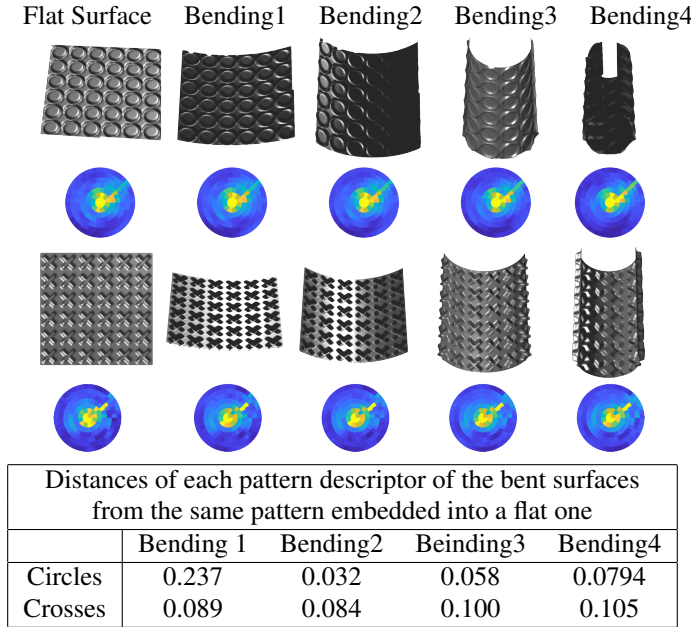


Fig. 11. Examples of two patterns embedded on surfaces with gradually stronger bendings. The table reports the distances between the pattern descriptor of the pattern embedded on the flat surface and on the bent surfaces. The L^2 norm is used to compute such distances.

Robustness to surface bendings. A very relevant feature of a pattern descriptor on surfaces is its robustness to different bendings of the underlying surface. To test this, we created a pattern of circlets and applied it to surfaces with more and more severe bendings. These models are reported in Figure 11 (First row). Since they are generated from a synthetic surface and adopting an isotropic bending we are guaranteed that the models possess the same pattern. In the second row of Figure 11, the corresponding descriptors are showed. A similar test is done using a pattern made up by small crosses (see Figure 11 - Third and Fourth rows). From the results, one can observe that the pattern descriptor changes depending on the embedded pattern. Moreover, it remains quite close, although not perfectly identical, for the same pattern across the bending changes. This show that the descriptor is more sensible to changes in the patterns it describes and far less sensitive to the surface bending.

Robustness to different model samplings. In the experiments shown in Section 5.1, the models with a low resolution were re-sampled, increasing the vertex density to reach a reasonably dense representation. Therefore, it is worth exploring the behaviour of the pattern descriptor when it is computed for different sampling density of the same model. We used the SHREC17 Original Dataset for this experiment. We selected two of the models of this dataset (one from class 8 and one from class 12) and down-sampled it from 40000 vertices to 3000 vertices, with various steps, using the default sampling scheme implemented in [44]. Figure 12 shows the pattern descriptor computed. It is easy to notice that the pattern descriptor holds its shape and changes when the number of vertices considerably decreases (although in the examples 7000 is still sufficiently stable with respect to the higher samplings we noticed that around 10000 vertices is a good quality compromise). This test shows that it

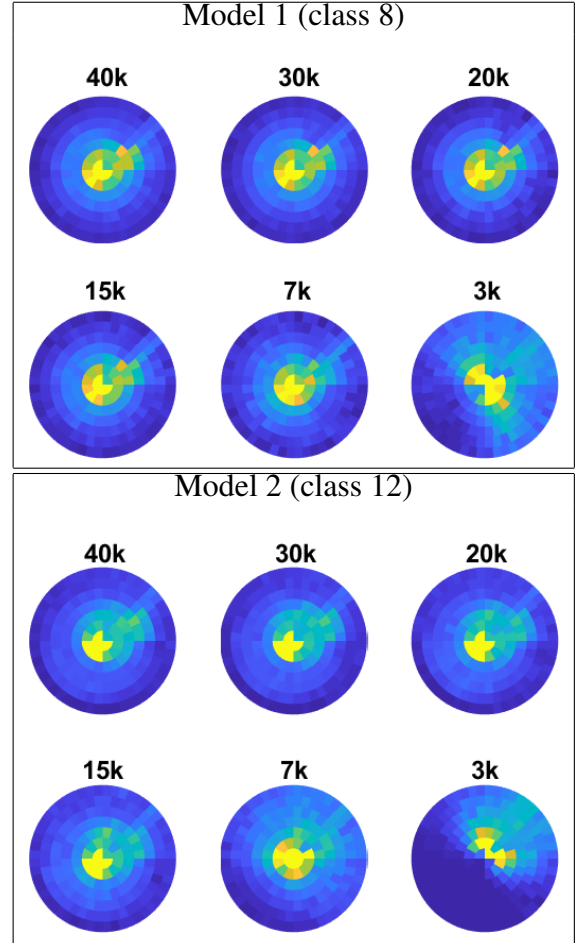


Fig. 12. Pattern descriptor robustness to different samplings. The number of vertices of the model is on top of the respective pattern descriptor.

is possible to have similar performances when re-sampling with different vertex resolutions, but it is necessary to have a minimum vertex density in order to have more stable pattern descriptors (in this case, approximately 10000 vertices or more).

Robustness to different choices of the parameters. The choice of the mpLBP parameters (R , n_{rad} and $multP$) is crucial for the performance of the method because they are related strictly related at the resolution a pattern is analysed. Anyway, we noticed that the performance of the mpLBP is quite stable for small variations of the parameters. In other words, slightly changing the parameters (all three of them) will not jeopardise the performances of the method. This fact was experimentally confirmed by selecting 27 variations of the best mpLBP run over the SHREC17 original dataset (see Table 1). The performance in terms of NN, FT and ST scores of the mpLBP are reported in Figure 13 for all these 27 settings. On the horizontal axis we report the parameter setting, while the vertical axis represents the performance score (different colours are used for the NN, FT and ST, respectively). The performances are very similar for all the settings. The maximum discrepancy observed across all the evaluation measures is around 0.05, which is very tiny, especially considering that it occurs only once in 27 runs.

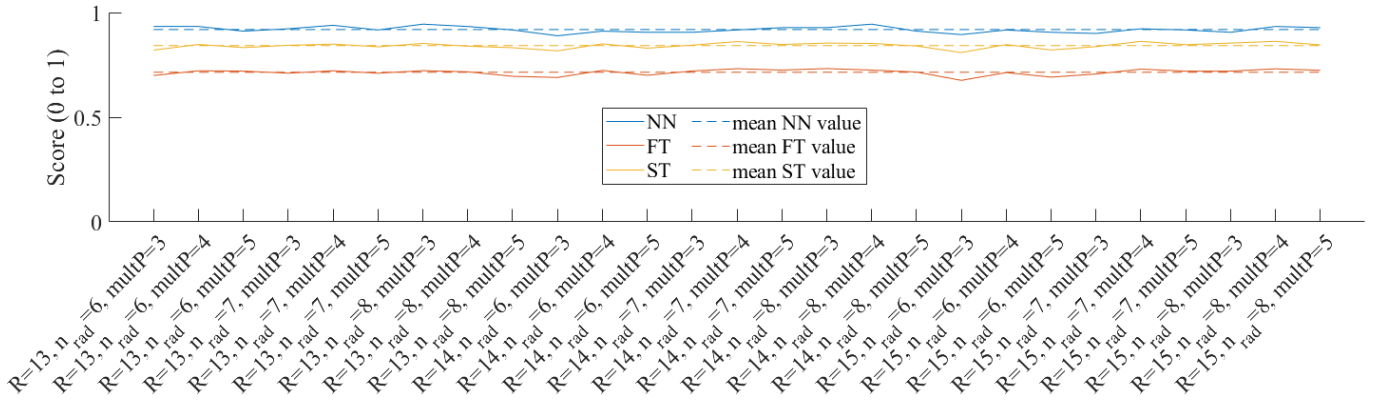


Fig. 13. The performances of 27 different mpLBP runs on the SHREC17 dataset, with different parameter settings. The dashed lines represent the mean value of the respective evaluation measure.

5.3. Different choices of the ring sampling scheme

We took the four possible variants of the point neighborhood sampling schemes described in Section 3.3 and we evaluated them on the SHREC'17 Original dataset. As shape properties, we considered the height-field as defined in Section 4.3 because of its simplicity and rough shape description and the Mean curvature because it generally performs well over geometric patterns. With reference to the labels defined in Section 3.3, Scheme 1 is computed using R ranging from 7 to 15 and $pxres$ ranging from 10 to 16. To evaluate Schemes 2, 3 and 4 we extracted the standard punctual descriptor with parameters $R = 14$, $n_{rad} = 7$ and $multP = 4$ and we considered only the sectors highlighted in the schemes. In this Section, we report only the most significant runs. Table 5 summarizes the results of the tests of mpLBP with these settings. In particular, when using sampling scheme 1 with h equal to the Mean Curvature, we use $R = 10$ and $pxres = 16$, while when using sampling scheme 1 with $h = HF$, we set $R = 7$ and $pxres = 12$. Note that the size of the mpLBP descriptor varies according to the different neighborhood sampling schemes. Looking at the retrieval performances, we notice that mpLBP with scheme 1 (with a square-based point neighborhood) performs poorly compared to the other schemes. This is not really surprising because a square-like point neighborhood inserts an orientation and an anisotropic sampling of the model. Unlike images where a square-like neighborhood is compliant with the intrinsic grid structure, for a surface, a circle or a geodesic neighborhood better reflect the intrinsic surface metric. Schemes 2 and 3 highlight that other sampling strategies over a circular neighborhood are possible and lead to good performances. On the contrary a too sparse sampling like the one proposed the Scheme 4 jeopardises the mpLBP performance. These results highlight that the mpLBP technique can be adapted to different schemes and that it is possible to keep limited the size of the descriptor compared to the original mpLBP settings [12] that probably generated a descriptor with redundant information.

5.4. Computational time

The mpLBP algorithm is implemented in around 200 MATLAB lines of code. The most expensive part, in terms of computational cost is the creation of the punctual descriptor, which

Parameters	NN	1-Tier	2-Tier	mAP	e	nDCG
Scheme 1 - Mean C.	0.739	0.395	0.534	0.481	0.324	0.705
Scheme 1 - HF	0.794	0.504	0.622	0.560	0.360	0.755
Scheme 2 - Mean C.	0.928	0.707	0.835	0.746	0.447	0.878
Scheme 3 - Mean C.	0.928	0.667	0.804	0.715	0.436	0.861
Scheme 4 - Mean C.	0.856	0.596	0.745	0.658	0.418	0.813

Table 5. Retrieval results of the various sampling schemes on the SHREC'17 Original Dataset.

is based on a kd-tree. This characteristic allows the mpLBP algorithm to run in a much shorter time if compared, for example, with the edgeLBP, while keeping similarly high evaluation scores. By running both edgeLBP and mpLBP on meshes with different number of vertices (from 5000 gradually to 120000 vertices) and different parameter settings, we can see the huge gap between the timings of the two methods (see Table 6). Tests are run on a personal computer Intel Core i7 processor (at 4.2 GHz) with 32Gb RAM. The edgeLBP (as currently implemented) has the number of sectors per ring constant across all the rings. In order to have a fair comparison, we also set the number of sectors to be constant for mpLBP (i.e., $P_j = P$ with $P \in \mathbb{N}$ fixed). We observed that n_{rad} and P do not affect the computation times that much. Indeed, the radius size and the number of vertices are the biggest bottlenecks. Figure 14 provides another computational time comparison between edgeLBP and mpLBP showing the much more severe increase of the edgeLBP computational cost compared to the cost increase of the mpLBP. These timings are those obtained on the 120k vertices mesh, with $R = 4.5$, $n_{rad} = 4$ $P = 15$. We do not report the same trend representation for other mesh and parameters for brevity reasons, but those trends are almost identical to the ones reported (only the time scale (y-axis scale) changes based on the radius).

6. Discussions and concluding remarks

We extended the LBP concept to surfaces represented as point clouds and defined a novel description, called mpLBP. Such a descriptor is able to keep state of the art performance and run more efficiently than its analogous edgeLBP which is based on a surface tessellation, see Section 5.4. Overall, the mpLBP performance is in par or superior (by a thin margin) with the

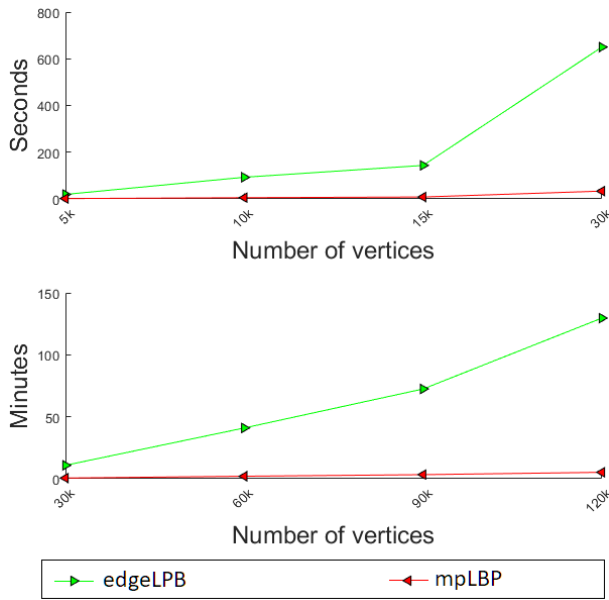


Fig. 14. Computational time trends for the mpLBP and edgeLBP.

5K	R=2,5	R=3,5	R=4,5
$n_{rad} = 4, P = 12$	22.04/2.88	16.89/1.38	19.06/1.35
$n_{rad} = 7, P = 12$	15.74/1.59	19.91/1.55	24.54/1.60
$n_{rad} = 4, P = 18$	11.40/1.27	15.89/1.48	17.12/1.38
$n_{rad} = 7, P = 18$	16.14/1.95	20.39/1.88	30.18/2.55

10K	R=2,5	R=3,5	R=4,5
$n_{rad} = 4, P = 12$	59.33/4.23	79.09/4.62	92.31/5.09
$n_{rad} = 7, P = 12$	71.69/4.35	95.58/4.93	116.51/5.46
$n_{rad} = 4, P = 18$	52.92/3.95	76.55/4.77	83.54/4.95
$n_{rad} = 7, P = 18$	72.43/5.01	95.86/5.53	140.23/6.25

15K	R=2,5	R=3,5	R=4,5
$n_{rad} = 4, P = 12$	81.13/5.31	118.42/7.48	143.29/8.00
$n_{rad} = 7, P = 12$	107.26/6.63	143.08/7.52	178.01/8.40
$n_{rad} = 4, P = 18$	81.92/5.96	115.85/7.38	128.10/7.49
$n_{rad} = 7, P = 18$	107.83/7.53	143.77/8.19	188.56/9.32

30K	R=2,5	R=3,5	R=4,5
$n_{rad} = 4, P = 12$	341.81/19.90	516.53/28.52	651.99/33.08
$n_{rad} = 7, P = 12$	454.23/23.30	618.36/28.36	805.07/33.72
$n_{rad} = 4, P = 18$	348.93/20.43	507.31/28.21	583.39/30.31
$n_{rad} = 7, P = 18$	456.26/25.10	621.50/29.75	811.99/35.25

90K	R=2,5	R=3,5	R=4,5
$n_{rad} = 4, P = 12$	2378.79/109.32	3661.28/158.43	4344.93/196.08
$n_{rad} = 7, P = 12$	3024.61/122.58	4142.54/157.74	5200.46/194.75
$n_{rad} = 4, P = 18$	2344.02/110.05	3481.22/160.97	3989.87/179.15
$n_{rad} = 7, P = 18$	3034.85/128.34	4145.79/163.19	5704.31/201.03

120K	R=2,5	R=3,5	R=4,5
$n_{rad} = 4, P = 12$	4314.18/165.65	6612.18/260.30	8341.62/335.82
$n_{rad} = 7, P = 12$	5583.24/189.33	7812.26/260.18	9954.04/332.90
$n_{rad} = 4, P = 18$	4236.92/170.22	6586.75/262.25	7626.82/309.25
$n_{rad} = 7, P = 18$	5596.74/198.12	7806.80/266.27	10438.45/348.40

Table 6. Computational times for edgeLBP and mpLBP (in seconds). The top-left cell of each table indicates the number of vertices.

current state of the art, whether these methods are based on engineered and/or learned descriptors. Due to the way the descriptor is evaluated over the rings (the mean of the values of a function on a set of points), the mpLBP remains stable to noise. Indeed, in presence of a small noise intensity that simulates the possible perturbation of a common scanning device, the performances are competitive with the current state of art methods. Moreover, it is robust to different surface bendings and is able to support different sampling schemes, as discussed in Section 5.3.

Other competing methods such as the T/mC/SIFT/FV method in [13] implicitly assumes that the same geodesic ‘sphere’ centered in every patch is able to parameterize all the models. The sphere radius is unique and can be obtained easily for the SHREC’17 dataset because the patches have comparable size but it is hard to obtain on datasets with models of different size. Moreover, such a single patch parameterization approach is not suitable to deal with datasets containing models with handles and protrusions, like some of SHREC’18 dataset. Indeed, T/mC/SIFT/FV translates the problem into a texture image comparison and requires a resampling with 20K vertices, while mpLBP works directly on the 3D model (mesh or point cloud). From these considerations, T/mC/SIFT/FV can be understood as a global descriptor that down-samples the model vertices as a pre-processing step. On the contrary, the mpLBP descriptor is local and its computation depends on the number of vertices, therefore the time complexities are not directly comparable, while also scoring similar performances.

While most patterns considered in this work are well described by a single scalar function for each point of the model, the possibility of describing patterns based on two or more properties (e.g.: curvature plus color, multiple color channel and so on) is of interest and one of the future research paths. Future reasoning will be devoted to the punctual descriptor used by the mpLBP. Since its resolution can easily be customized and it is not tied to a specific surface property (curvatures, colors, height-fields and so on), the punctual descriptor itself could be used as a feature vector to encode different surface details and/or as the starting point for more advanced local descriptions. A further extension is the application of the punctual descriptor to the problem of pattern recognition over surfaces. This last is still an open problem, as observed in [5], and a quick and well performing technique such as the mpLBP is a promising contribution towards a possible solution.

7. Acknowledgments

This study was partially supported by the CNR-IMATI projects DIT.AD004.028.001 and DIT.AD021.080.001, and the e-Roma project from the Agence Nationale de la Recherche (ANR-16-CE38-0009).

References

- [1] Othmani, A, Voon, LFLY, Stolz, C, Piboule, A. Single tree species classification from terrestrial laser scanning data for forest inventory. Pattern Recognition Letters 2013;34(16):2144–2150.

- [2] Zeppelzauer, M, Poier, G, Seidl, M, Reinbacher, C, Schuster, S, Breiteneder, C, et al. Interactive 3D segmentation of rock-art by enhanced depth maps and gradient preserving regularization. *J Comput Cult Herit* 2016;9(4):19:1–19:30.
- [3] Biasotti, S, Moscoso Thompson, E, Aono, M, Hamza, AB, Bustos, B, Dong, S, et al. Retrieval of surfaces with similar relief patterns. In: Pratikakis, I, Dupont, F, Ovsjanikov, M, editors. *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association; 2017, p. 95 – 103.
- [4] Moscoso Thompson, E, Biasotti, S, Sorrentino, G, Polig, M, Hermon, S. Towards an Automatic 3D Patterns Classification: the GRAVITATE Use Case. In: Sablatnig, R, Wimmer, M, editors. *Eurographics Workshop on Graphics and Cultural Heritage*. The Eurographics Association; 2018, p. 251 – 254.
- [5] Biasotti, S, Moscoso Thompson, E, Barthe, L, Berretti, S, Giachetti, A, Lejemble, T, et al. Recognition of Geometric Patterns Over 3D Models. In: Telea, A, Theoharis, T, Veltkamp, R, editors. *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association; 2018, p. 71 – 77.
- [6] Ojala, T, Pietikäinen, M, Harwood, D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition* 1996;29(1):51–59.
- [7] Ojala, T, Pietikäinen, M, Maenpää, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE TPAMI* 2002;24(7):971–987.
- [8] Werghi, N, Tortorici, C, Berretti, S, Bimbo, AD. Local binary patterns on triangular meshes: Concept and applications. *Computer Vision and Image Understanding* 2015;139:161–177.
- [9] Werghi, N, Berretti, S, Bimbo, AD. The mesh-LBP: A framework for extracting local binary patterns from discrete manifolds. *IEEE Trans Image Processing* 2015;24(1):220–235.
- [10] Moscoso Thompson, E, Biasotti, S. Description and retrieval of geometric patterns on surface meshes using an edge-based lbp approach. *Pattern Recognition* 2018;82:1 – 15.
- [11] Moscoso Thompson, E, Biasotti, S. Retrieving color patterns on surface meshes using edgelbp descriptors. *Computers & Graphics* 2019;79:46 – 57.
- [12] Moscoso Thompson, E, Biasotti, S, Digne, J, Chaine, R. mpLBP: An Extension of the Local Binary Pattern to Surfaces based on an Efficient Coding of the Point Neighbours. In: Biasotti, S, Lavoué, G, Veltkamp, R, editors. *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association. ISBN 978-3-03868-077-2; 2019, p. 9–16. doi:10.2312/3dor.20191056.
- [13] Giachetti, A. Effective characterization of relief patterns. *Computer Graphics Forum* 2018;37(5):83–92.
- [14] Friedman, JH, Bentley, JL, Finkel, RA. An algorithm for finding best matches in logarithmic expected time. *ACM Trans Math Softw* 1977;3(3):209–226.
- [15] Moscoso Thompson, E, Tortorici, C, Werghi, N, Berretti, S, Velasco-Forero, S, Biasotti, S. Retrieval of Gray Patterns Depicted on 3D Models. In: Telea, A, Theoharis, T, Veltkamp, R, editors. *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association; 2018, p. 63–69.
- [16] Cimpoi, M, Maji, S, Kokkinos, I, Mohamed, S, Vedaldi, A. Describing textures in the wild. In: *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. CVPR '14*; Washington, DC, USA: IEEE Computer Society. ISBN 978-1-4799-5118-5; 2014, p. 3606–3613.
- [17] Lowe, DG. Distinctive image features from scale-invariant keypoints. *Int J Comput Vision* 2004;60(2):91–110.
- [18] Dalal, N, Triggs, B. Histograms of oriented gradients for human detection. In: *Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Conference on*; vol. 1. 2005, p. 886–893.
- [19] Pietikäinen, M, Hadid, A, Zhao, G, Ahonen, T. *Computer Vision Using Local Binary Patterns*; vol. 40 of *Computational Imaging and Vision*. Springer; 2011.
- [20] Liu, L, Fieguth, P, Guo, Y, Wang, X, Pietikäinen, M. Local binary features for texture classification: Taxonomy and experimental study. *Pattern Recognition* 2017;62:135 – 160.
- [21] Cimpoi, M, Maji, S, Kokkinos, I, Vedaldi, A. Deep filter banks for texture recognition, description, and segmentation. *International Journal of Computer Vision* 2016;118(1):65–94.
- [22] Werghi, N, Tortorici, C, Berretti, S, Bimbo, AD. Boosting 3D LBP-based face recognition by fusing shape and texture descriptors on the mesh. *IEEE Trans Information Forensics and Security* 2016;11(5):964–979.
- [23] Tortorici, C, Werghi, N, Berretti, S. Performing Image-like Convolution on Triangular Meshes. In: Telea, A, Theoharis, T, Veltkamp, R, editors. *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association; 2018, p. 111 – 114.
- [24] Koenderink, JJ, van Doorn, AJ. Surface shape and curvature scales. *Image & Vision Computing* 1992;10(8):557 – 564.
- [25] Moscoso Thompson, E, Biasotti, S. Edge-based LBP Description of Surfaces with Colorimetric Patterns. In: Telea, A, Theoharis, T, Veltkamp, R, editors. *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association; 2018, p. 1 – 8.
- [26] Alexa, M, Behr, J, Cohen-Or, D, Fleishman, S, Levin, D, Silva, CT. Point set surfaces. In: *Proceedings of the Conference on Visualization '01. VIS '01*; Washington, DC, USA: IEEE Computer Society. ISBN 0-7803-7200-X; 2001, p. 21–28.
- [27] Cazals, F, Pouget, M. Estimating differential quantities using polynomial fitting of osculating jets. In: *SGP '03*. ISBN 1-58113-687-0; 2003, p. 177–187.
- [28] Oztireli, AC, Guennebaud, G, Gross, M. Feature preserving point set surfaces based on non-linear kernel regression. *Computer Graphics Forum* 2009;28:493–501(9).
- [29] Béarzi, Y, Digne, J, Chaine, R. Wavejets: A local frequency framework for shape details amplification. *Computer Graphics Forum* 2018;37(2):13–24.
- [30] Digne, J, Chaine, R, Valette, S. Self-similarity for accurate compression of point sampled surfaces. *Computer Graphics Forum* 2014;33(2):155–164.
- [31] Hamdi-Cherif, A, Digne, J, Chaine, R. Super-resolution of point set surfaces using local similarities. *Computer Graphics Forum* 2018;37(1):60–70.
- [32] Starck, J, Hilton, A. Correspondence labelling for wide-timeframe free-form surface matching. In: *ICCV*. 2007, p. 1–8.
- [33] GRAVITATE: Discovering relationships between artefacts using 3D and semantic data. 2015-2018. EU H2020 REFLECTIVE project.
- [34] Karageorghis, V, Karageorghis, J, Foundation, AL. The coroplastic art of ancient Cyprus. Nicosia : A.G. Leventis Foundation; 1991. At head of title: A.G. Leventis Foundation.
- [35] Shilane, P, Min, P, Kazhdan, M, Funkhouser, T. The princeton shape benchmark. In: *Shape modeling applications, 2004. Proceedings. IEEE*; 2004, p. 167–178.
- [36] Salton, G. The evaluation of automatic retrieval procedures - selected test results using the smart system. *American Documentation* 1965;16(3):209–222.
- [37] Baeza-Yates, RA, Ribeiro-Neto, B. *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.; 1999. ISBN 020139829X.
- [38] Rijsbergen, CJV. *Information Retrieval*. 2nd ed.; Newton, MA, USA: Butterworth-Heinemann; 1979.
- [39] Peyré, G. Graph Toolbox - A toolbox to process graph and triangulated meshes. 2009. URL: [http://www.ceremade.dauphine.fr/\\$\sim\\$peyre/matlab/graph/content.html](http://www.ceremade.dauphine.fr/\simpeyre/matlab/graph/content.html).
- [40] Moscoso Thompson, E, Biasotti, S. A Preliminary Analysis of Methods for Curvature Estimation on Surfaces With Local Reliefs. In: Cignoni, P, Miguel, E, editors. *Eurographics 2019 - Short Papers*. The Eurographics Association; 2019, p. 25–28.
- [41] Guennebaud, G, Gross, M. Algebraic point set surfaces. *ACM Trans Graph* 2007;26:23.
- [42] Albuz, E, Kocalar, ED, Khokhar, AA. Quantized CIELab* space and encoded spatial structure for scalable indexing of large color image archives. In: *ICASSP '00*; vol. 6. 2000, p. 1995–1998.
- [43] Hunt, RWG, Pointer, MR. *Measuring Colour*, Fourth Edition. Wiley; 2011.
- [44] Attene, M, Falcidieno, B. Remesh: An interactive environment to edit and repair triangle meshes. In: *Proc. SMI'06*. IEEE Computer Society; 2006, p. 41.