Eurographics Workshop on 3D Object Retrieval (2019)
S. Biasotti, G. Lavoué, B. Falcidieno, and I. Pratikakis (Editors)

mpLBP: An Extension of the Local Binary Pattern to Surfaces based on an Efficient Coding of the Point Neighbours

E. Moscoso Thompson[1], S. Biasotti[1], J. Digne[2], R. Chaine[2],

[1]Istituto di Matematica Applicata e Tecnologie Informatiche 'E. Magenes' - CNR, Italy
[2]LIRIS, Université C. Bernard Lyon 1, CNRS, France

Abstract

The description of surface textures in terms of repeated colorimetric and geometric local surface variations is a crucial task for several applications, such as object interpretation or style identification. Recently, methods based on extensions to the surface meshes of the Local Binary Pattern (LBP) or the Scale-Invariant Feature Transform (SIFT) descriptors have been proposed for geometric and colorimetric pattern retrieval and classification. With respect to the previous works, we consider a novel LBP-based descriptor based on the assignment of the point neighbours into sectors of equal area and a non-uniform, multiple ring sampling. Our method is able to deal with surfaces represented as point clouds. Experiments on different benchmarks confirm the competitiveness of the method within the existing literature, in terms of accuracy and computational complexity.

CCS Concepts

• ***Information systems*** → ***Information retrieval;*** • ***Computing methodologies*** → ***Shape modeling; Shape analysis;***

1. Introduction

The characterization of relief and color patterns over surfaces is now capturing a larger attention in the research community because these characteristics are key aspects for interpreting and indexing the informative content of 3D models. The analysis of surface patterns has a suite of potential application domains such as the recognition of natural structures, like trees [OVSP13], the analysis of artworks styles [ZPS*16], the classification of fabric patterns [BMTA*17] or the categorization of objects [MTBS*18].

Many natural surfaces and decorations possess repeating elements that strongly characterize their type, material, and style [WLKT09]. We refer to these decorative elements as *patterns*. In this context, a single or a couple of decorative elements (i.e. an eye, a rosette, etc.) do not represent a pattern. We group the patterns in two categories: *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. When dealing with geometric patterns, we assume that the geometric variations can locally be interpreted as a height field over the surface. Figure 1 shows examples of artworks and design objects characterized by geometric and colorimetric patterns. The fourth and fifth models, in particular, share a common pattern despite their different shapes and functionalities.

Recently, methods based on extensions to surfaces of the Local Binary Pattern descriptor (LBP), namely the meshLBP [WTBB15, WBB15] and the edgeLBP [MTB18a, MTB18a] or based on the Scale Invariant Feature (SIFT) [Gia18] have shown that the retrieval and classification of patterns of surfaces is feasible.

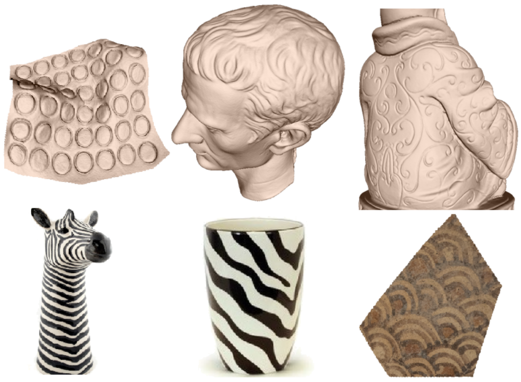

Figure 1: *Examples of three surfaces with geometric patterns (Top) and three surfaces with colorimetric ones(Bottom).*

Originally, the LBP [OPH96, OPM02] has been introduced to characterize the binary distribution of the intensities on a ring around one pixel of an image. The intensities on the ring are thresholded with respect of the value of the current pixel. Furthermore, the scale of the LBP descriptor directly depends on the radius chosen to construct the ring.

In this work, we introduce a new extension of the LBP to sur-

DOI: 10.2312/3dor.20191056

faces. This descriptor is able to deal with surfaces described as a set of points. If the surface is given as a tessellation, this set of points can be the set of vertices, supplemented by additional points sampled on the faces if the number of vertices is low (see Section 3). Those points are organized in a kd-tree structure that permits an efficient search of the neighbors [FBF77] to extract concentric rings needed by the LPB descriptor. Rings are adaptively sampled so that an "equal sector" area is preserved along the neighbour rings without changing the width of the rings. In the experiments we will show how the new descriptor, named Mean Point Local Binary Patterns (*mpLBP* for short), considerably reduces the computational cost with respect to its direct competitor, the *edgeLBP*, while preserving competitive retrieval and classification performances.

The remainder of this paper is organized as follows. Section 2 overviews previous research for the retrieval and classification of patterns over surfaces. Section 3 introduces the new punctual operator used to build the mpLBP descriptor. Section 4 presents the retrieval and classification performance of the method on two benchmarks [BMTA*17, MTTW*18] and discusses its robustness. Concluding remarks are provided in Section 5.

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 key aspect for the detection of specific texture patterns is the recognition of the texture properties robustly to the possible variations [CMK*14]. 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) [OPH96, OPM02], the Scale Invariant Feature Transform (SIFT) [Low04] and the Histogram of Oriented Gradients (HOG) [DT05]. LBP-based methods are very popular and a large number of LBP variants has been proposed [PHZA11]. An extended taxonomy of 32 LBP variations and their performance evaluation for texture classification has been proposed in [LFG*17] where the LBP variations and 8 convolutional network based features are evaluated over 13 datasets of 2D images. Among the LPB 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 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 [CMK*14]. 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% [CMKV16] at the cost of a higher computational complexity.

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 opportune 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 the treatment of three-dimensional data. As an example of reduction strategy, the method in [OVSP13] 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, [ZPS*16] adopts a height map to project the reliefs and engraves of rock artifacts into an image and classify them. The LBPI and CMC approaches proposed in the SHREC'17 contest [BMTA*17] 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, [Gia18] has proposed to use an opportune parameterization around a patch centroid to project the mean curvature values into an image and then, to adopt the SIFT + Fisher Vector [CMK*14] strategy to compare the parametric images. The Mesh Local Binary Pattern (meshLBP) approach [WTBB16, WTBB15, WBB15, TWB18] proposed the first extension of the LBP description [OPH96] to triangle meshes. The main idea behind the meshLBP is that triangles play the role of pixels; there, the 8-neighbor connectivity of images is ideally substituted by a 6-neighbor 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 (in the examples, mainly Gaussian and mean curvatures, and the shape index [KvD92] or a colorimetric property like the gray-scale values). The edgeLBP [MTB18a, MTB18b, MTB19, BMTA*17, MTTW*18] perform an LBP evaluation that is based on the rings build over the mesh edges.

For point clouds, local surface patches can also be constructed by regression using the neighborhood around one point [ABCO*01, CP03, OGG09, BDC18] 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 or a quadric (see [HCDC17] for an application to super-resolution).

3. mpLBP description

The main idea behind the mpLBP is to evaluate the Local Binary Pattern by estimating the variation of a surface property on a set of neighborhood points. The surface is represented either by a triangulation or a point cloud. We need to build a proper local descriptor which allows an LBP evaluation, that is to say a ringed structure. The mpLBP algorithm can be divided in two main parts: the creation of the punctual descriptor (Section 3.1) and the LBP evaluation (Section 3.2). Finally in Section 3.3 we discuss parameters definition and tuning.

3.1. Punctual descriptor

Let us have a point set S in the 3D Euclidean space and a surface property defined on the elements of S, $h : S \to \mathbb{R}$, a function able to capture the local pattern variations. (e.g.: curvature-based values in case of geometric patterns, a color channel in case of depicted

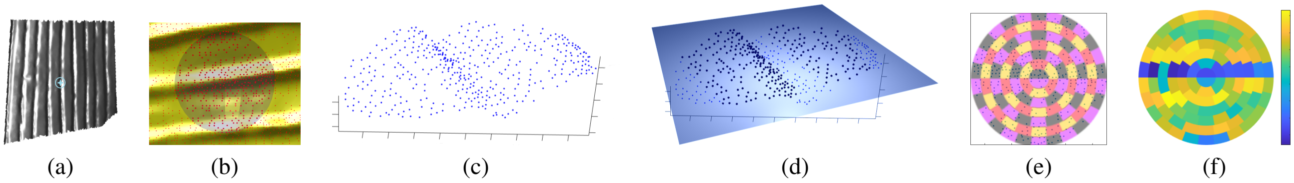

Figure 2: *Computation of a mpLBP descriptor at a point $\tilde{p}$ (marked with a light-blue star in (a)). The neighborhood $S[\tilde{p}]$ of $\tilde{p}$ is shown with the dark bubble in (b). The point density in $S[\tilde{p}]$ is depicted in (c). The plane π is represented in (d), while the clustering into sectors is shown in (e). Finally, the punctual descriptor, represented as a 'circular' feature vector, is shown in (f).*

decorations, etc.). Let us consider a point $\tilde{p} \in S$ and the set $S[\tilde{p}]$ of all the other points p_i in S with 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. R is the parameter which mostly influences the computational costs. The kd-tree is computed once per model (its computational cost is $n\log(n)$). After that, the extraction of the neighbours can be performed in a nearly constant times.

Points in $S[\tilde{p}]$ are projected on a plane π obtained using linear regression. The projected points are sorted in n_{rad} concentric rings based on their distances from $\tilde{p}$. Within each ring, no sorting is necessary in this context (as the evaluation we are aiming at is rotational invariant), still we decided to use the maximum curvature direction, which serves as a reference direction for sorting the points into sectors, adding robustness to the descriptor. The sorting of each p_i is given by the angle θ_i equal to angle between $\overrightarrow{Pr_\pi(p_i) - \tilde{p}}$ and the maximum curvature direction, where $Pr_\pi(p_i)$ is the projection of the point p_i on π. The number of rings is determined by the parameter n_{rad}, that we call *radial resolution*. More formally, each ring is defined as follows:

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

Each $S[\tilde{p}]_j$ is divided in P_j sectors, based on the θ_i values. Note that P_j may vary along the rings. We call P_j the *spatial resolution*. A formal definition for the sector k of the ring j (sector (j,k) for short) of the point $\tilde{p}$ is the following:

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

where $\theta_k = k\frac{2\pi}{P_j}$. 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.

$sec(\tilde{p})_j^k$ is defined as the weighted mean of the values $h(p_i)$ of the points in the sector (j,k). The weights are computed using a Gaussian filter centered in the middle of sector (j,k) (with coordinates $((j-\frac{1}{2})\frac{R}{n_{rad}}, (k-\frac{1}{2})\frac{2\pi}{P_j})$) and $\sigma = (j-\frac{1}{2})\frac{R}{n_{rad}}\sin(\frac{2\pi}{2P_j})$ to account for the whole sector in a robust manner, see Figure 3(Left). We define the punctual descriptor at $\tilde{p}$ as the vector of the values $sec(\tilde{p})_j^k$, which size is $\sum_j P_j$. Figure 2 represents the pipeline to build the feature vector.

In our implementation, we excluded the computation of the punctual descriptor at points that are close to the boundary of the model (if any), as they generates point descriptors with many empty sectors. As a general rule, if a point descriptor has more than $\frac{1}{4}\sum_j P_j$ empty sectors, it is not considered. If the boundary of the model is known, it is enough to consider only the points that are at least far R from the boundary. When the intersection of the sphere of radius R with the point cloud generates multiple parts like those in Figure 3(Right), the encoding of the punctual descriptor is not correct. Thus, for a given model M we assume that the projection onto π is injective and that the surface locally captured by the sphere should locally be 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 on M.

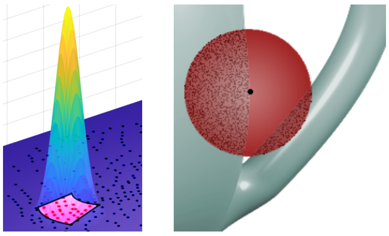

Figure 3: *Left: the Gaussian filter to weight the points (in red) in a given sector (in pink). The colors range from blue (0) to yellow (1). Right: a surface sampling that can occur if a the radius R is greater than $\tilde{R}_{max}$: points from two parts of the model are extracted.*

3.2. Local Binary Pattern evaluation

Our idea is to apply the Local Binary Pattern (LBP) evaluation to the punctual descriptor introduced in Section 3.1.

The LBP paradigm is very popular for images and many versions are available [PHZA11]. In that case, the function h corresponds to a color channel (often a grey-scale value). We extend the LBP version in [OPH96]. For each pixel p, the set of pixels $\tilde{p}_j$ with 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). The histogram H of the LBP values for all the image pixels is the LBP descriptor of the image. Multiple rings can be considered, increasing the size and descriptive capability of the descriptor. Figure 4 shows this process for a single pixel (Left) using no riegnand the final descriptor (Right).

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

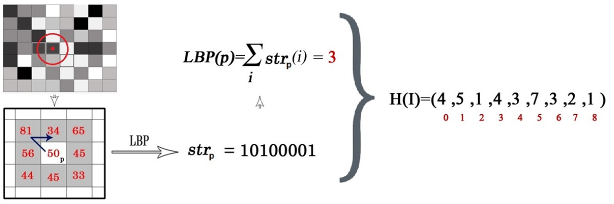

Figure 4: *The LBP evaluation for an image. Top-left: in red, the pixel p is highlighted with a dot and the circle of radius R. The values of h around p are reported in the bottom-left image.*

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:

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

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

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 generate 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 $j = 1, ..., n_{rad}$ and $m = 0, ..., P_j$).

3.3. Parameter settings

Similarly to the edgeLBP [MTB18a], also the mpLBP depends on the radius R, the radial resolution n_{rad} and the spatial resolution P (even if, for the mpLBP, P_j may vary over the rings). In particular, the way R is supposed to be tuned is the same as that of the edgeLBP. The main difference is the fact that points on edges, considered in the edgeLBP, are replaced by areas (sectors) that contain usually more than one point. In this new scenario, the size of the sectors became crucial in order to keep the quantity of information carried by each sector uniform. Thus, we opted for a set of parameters which keeps the areas equal to each other. The parameter tuning should be done according to the following instructions.

- R: the radius of the dark bubble in Figure 2(b) should 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. Setting $P_j = multP(2j-1)$, $multP \in \mathbb{N}_+$, all the sectors have the same area. P_j is defined by n_{rad}. This degree of freedom is tuned by the parameter $multP$ (that replaces P_* as a parameter). 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.

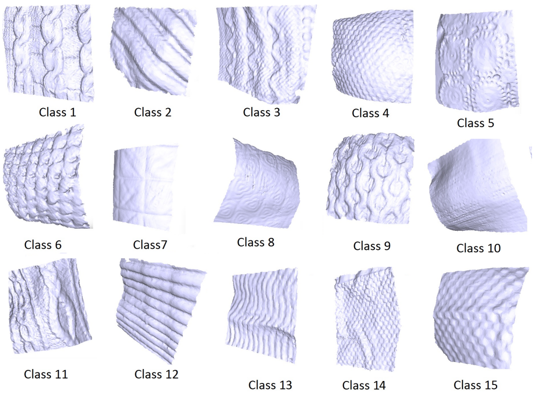

Figure 5: *The knitted patterns of the SHREC17 contest.*

4. Benchmarks, evaluation measures and mpLBP performances

We test the retrieval and classification capability of the mpLBP by matching its performances to those obtained by other methods suited for the same task on two very recent benchmarks.

4.1. Design of the experiments

In the following, we present the dataset, the performance measures and the criteria for the parameter selection.

4.1.1. Dataset

SHREC17 Benchmark, geometric patterns The SHREC'17 benchmark dataset [BMTA*17] on the retrieval of relief patterns is composed by 720 triangle meshes derived from knitted objects. Models are grouped into 15 classes, each one made of 48 models characterized by one of the textile pattern in Figure 5. Each class was created from the acquisition of the same surface with 12 different embeddings; then, each model was re-sampled four times. Two datasets were derived: the first one is related to the complete set of 720 models and aims at evaluating the overall robustness and stability of methods with respect to different mesh representations. The second one groups the 180 original meshes according to their textile pattern and it is better suited to analyze the capability of a method to effectively recognize a pattern independently of the overall surface embedding.

SHREC18 Benchmark, colorimetric patterns The SHREC'18 benchmark [MTTW*18] originated from 20 base models without any texture or colorimetric information to which were applied 15 gray and white texture each. Then, 300 models were derived from the combination of 20 base models and 15 textures with a semi-automatic algorithm [MTTW*18]. In addition, the luminosity of

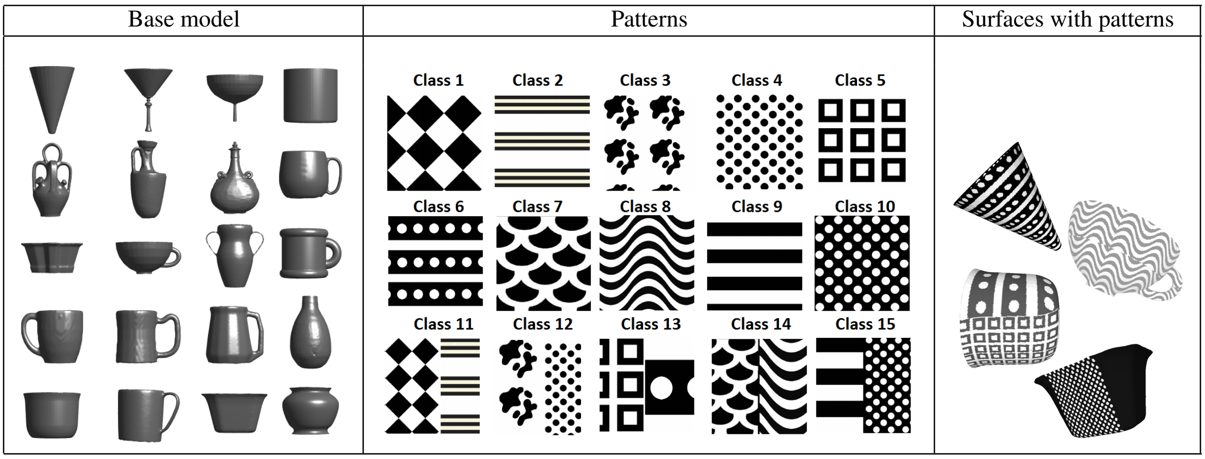

Figure 6: *The 20 base models used in the SHREC'18 benchmark are shown on the left. Each pattern (middle) was applied to all of the base models and changed in terms of their luminosity (examples of the final models are shown on the right).*

the textures was modified by using a random value to obtain the same pattern with 20 different shades. At least 30% of the model surfaces are covered by one of the 15 patterns, whereas the remainder of the surface is only black or only white. Five patterns are mixed versions of the other 10 patterns (see Figure 6). Two different classifications are provided: one comprises only the models with one single pattern (*Single pattern dataset*), the other includes all the models (*Complete dataset*).

4.1.2. Evaluation Measures

The results of our tests are evaluated using a number of classical information retrieval measures.

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. For a class with $|C|$ members, $k = 1$ for the Nearest Neighbor (NN), $k = |C| - 1$ for the first tier (FT), and $k = 2(|C| - 1)$ for the second tier (ST). These values range from 0 to 1.

Normalized Discounted Cumulative Gains The Discounted Cumulative Gain (DCG) derives from the concept of the Cumulative Gain. The cumulative gain sums the graded relevance values of all results in the list of retrieved objects of a given query. The DCG is based on the assumption that relevant items are more useful if they appear earlier in a query list and therefore 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 [SMKF04].

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 [Sal65]. We consider the mean Average Precision (mAP), which is the area under a precision-recall curve [BYRN99]. The e-measure e [Rij79] was also introduced as a quality measure of the first models retrieved for every query. Formally, $e = \frac{2}{Precision^{-1}+Recall^{-1}}$, where *Precision* and *Recall* are those defined in the previous evaluation measure.

4.1.3. Experimental settings

The choice of the function h depends on the type of patterns to be characterized. In case of geometric patterns, we adopted the maximum curvature, as implemented in Matlab in [Pey], as it provided the best performance for the edgeLBP description and the model in the benchmarks were all triangle meshes [MTB18a]. On point clouds, the PCL [RC11] or CGAL [The18] libraries provide valid estimation of such quantities. For the colorimetric patterns, we used the L-channel of the CIELab color-space [AKK00, HP11]: the L-channel encodes the color luminosity, that is descriptive enough for the patterns used in the SHREC'18 benchmark.

Initially, we adopted the same parameters of the edgeLBP (R proportional to the pattern size, $n_{rad} = 5$ and $P = 15$ for all the rings). Then, we observed that keeping the area of the sectors constant is very beneficial for the punctual descriptor. The larger the R value, the better the overall results, despite the increase of the computational costs. Then n_{rad} and $multP$ are tuned so that the sectors are big enough to contain at least a few points. Once this condition is verified, very small differences between the results obtained with different parameters settings were very small (±3.5%). Moreover, too many empty sectors jeopardize the mpLBP performances. When it is the case, we over-sampled the surfaces with the ReMesh tool [AF06].

4.2. Results

For each benchmark, we extensively tested the mpLBP retrieval and classification performances. For sake of conciseness, in this Section we report only the best runs.

Performances on the SHREC'17 benchmark In addition to the methods of [BMTA*17] that obtained the best performances, we compare the mpLBP with the edgeLBP [MTB18a] and the SIFT-based method in [Gia18]. Among all the settings tested, the best

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 - *run2*	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 - *set1*	0.917	0.711	0.859	0.743	0.420	0.861
mpLBP - *set2*	0.917	0.707	0.861	0.745	0.421	0.865

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 - *run2*	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 - *set1*	0.993	0.676	0.820	0.732	0.630	0.921
mpLBP - *set2*	0.997	0.667	0.818	0.733	0.635	0.925

Table 1: *Results on the SHREC'17 benchmark, both the Original (Top) and the Complete (Bottom) Dataset.*

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 - *set3*	0.965	0.739	0.862	0.781	0.600	0.910
mpLBP - *set4*	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 - *set3*	0.903	0.739	0.862	0.668	0.520	0.850
mpLBP - *set4*	0.907	0.573	0.735	0.639	0.510	0.840

Table 2: *Performance scores over the Single pattern data set and Complete data set of the SHREC'18 benchmark.*

performing ones are $R = 14$, $n_{rad} = 7$ and $multP = 4$ (*set1*) and $R = 15$, $n_{rad} = 7$ and $multP = 4$ (*set2*). Since most of the models in the dataset have few vertices, we re-sampled all the models so they have at least 40000 vertices. Table 1 reports the mpLBP scores together with the other methods, with respect to NN, FT, ST, e-measure, mAP and nDCG.

Performances on the SHREC'18 benchmark The performance of the mpLBP on this benchmark is compared against those obtained in [MTTW*18] and [MTB19]. The parameter settings with the best evaluations are $R = 0.10$ $n_{rad} = 7$ $multP = 1$ (*set3*) and $R = 0.14$ $n_{rad} = 7$ $multP = 1$ (*set4*). Table 2 summarizes the best scores obtained (more runs and methods are available in [MTTW*18] and [MTB19]).

Discussions The mpLBP scores equivalently to the edgeLBP and T/mC/SIFT/FV over the benchmark on geometric patterns (SHREC'17), while it generally obtains the best performance rate for the colorimetric patterns (SHREC'18 benchmark). As discussed in Section 3.3, a key issue for the success of the mpLBP description is that the point cloud is dense enough, i.e., most of the sectors of the descriptors are not empty. Not surprisingly, the mpLBP performs better on the colorimetric benchmark were, to guarantee the decorations were intelligible, the original surfaces were already quite densely sampled ($100K$ vertices, each).

Overall, the mpLBP performance is in par or superior (by a thin margin) with the current state of the art, independently these methods are based on engineered and/or learned descriptors. If compared to the edgeLBP, the winning aspect is its lower computational costs (see Section 4.2.2). Regarding the T/mC/SIFT/FV method in [Gia18], it 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 with models with handles and protrusions, like part of those in the SHREC'18 contest. Indeed, the T/mC/SIFT/FV translates the problem into a texture image comparison and requires a resampling with 20K vertices, while the mpLBP works directly on the 3D model (mesh or point cloud). From these considerations, we think that the T/mC/SIFT/FV is a global descriptor that downsamples the model vertices as a pre-processing step. On the contrary, the mpLBP is local and it computation depends on the number of vertices, therefore their time complexity is not directly comparable, while also scoring similar performances.

4.2.1. Robustness

We tested the mpLBP robustness to different types of noise, depending on the pattern nature. For the geometric patterns, we adopted a Gaussian noise on the vertex coordinates based on a parameter λ_g, which is the percentage of the diameter of the smallest sphere that bounds the surface. The values of λ_g considered are 0.2 and 0.4. See Figure 7(Top) for an example of the mesh degradation.

For the colorimetric patterns, the RGB values associated to the vertices were randomly perturbed. In particular, bits of noise, based on an integer parameter λ_c, were added to each RGB channel (we assumed 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 7(Bottom) for examples of the pattern degradation.

Table 3 lists the mpLBP score. Overall, the descriptors keep good performances under the smallest noise intensity ($\lambda_g = 0.2$ and $\lambda_c = 5$). When the noise increases, the performance of the mpLBP drops significantly, especially in the case of colorimetric patterns. This is probably due to our evaluation strategy (since we adopt the mean over a sector) together with the fact that patterns are defined by small variations of the function h, which are easily corrupted by noise.

4.2.2. Computational costs

The prototype of the mpLBP algorithm is implemented in MATLAB. The strength of the mpLBP method is the low computational complexity rather than the impressiveness of the performances (generally in line with those of the edgeLBP). Using the kd-tree structure instead of the much more computational demanding navigation of the mesh elements, the mpLBP is by far faster than methods implemented on meshes and, in particular, the edgeLBP. In Table 4 we report (in seconds) the computational times we observed

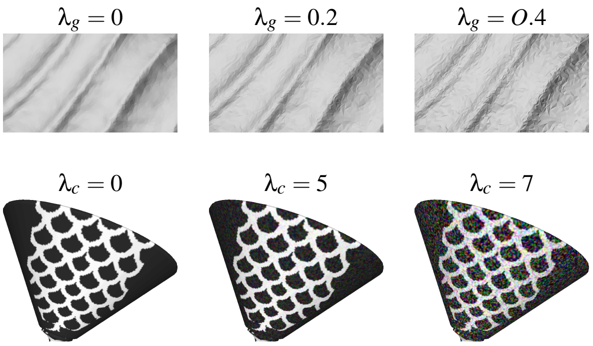

Figure 7: *Pattern distortion when noise is randomly added. Top row: a geometric pattern is corrupted using increasing Gaussian noise. Bottom row, an increasing random noise is added to each RGB color channel.*

SHREC’17: Original Dataset, geometric noise						
Method	NN	1-Tier	2-Tier	mAP	e	nDCG
mpLBP - *set1* Clean	0.917	0.711	0.859	0.743	0.420	0.861
mpLBP - *set1*, $\lambda_g = 0.2$	0.911	0.693	0.846	0.733	0.380	0.790
mpLBP - *set1*, $\lambda_g = 0.4$	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 - *set3* Clean	0.965	0.739	0.862	0.781	0.600	0.910
mpLBP - *set3*, $\lambda_c = 5$	0.915	0.514	0.653	0.586	0.440	0.822
mpLBP - *set3*, $\lambda_c = 7$	0.75	0.332	0.445	0.457	0.355	0.741

Table 3: *mpLBP performance for data with noise. Top: the Original Dataset of the SHREC’17 benchmark, Bottom: Single Pattern Dataset of the SHREC’18 benchmark.*

running both the edgeLBP and the mpLBP on meshes with different number of vertices (from 5000 to 120000 vertices) and different parameter settings. Tests are run on a personal computer Intel Core i7 processor (at 4.2 GHz) with 32Mb RAM. For a fair comparison, in this test the number of sectors for the mpLBP is the same of the edgeLBP. While n_{rad} and P does not affect the computation times that much, the radius size and the number of vertices are the biggest bottlenecks (as expected).

5. Conclusions

We extended the LBP concept to surfaces represented as point clouds and defined a novel description, called mpLBP, whose core strength is its computational efficiency. When the performances and the computation times are observed together, the mpLBP shines as one of the best methods in the current literature for pattern classification/retrieval. Due to the way the sectors evaluation is done (the mean of the values of a function on a set of points), the mpLBP remains sensible to noise. Still, in presence of light noise, the performances are competitive with the current state of art methods.

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 customized and it is not tied to a specific surface property (curvatures, colors, height-fields and so on) it 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 or of the entire pipeline to the problem of pattern recognition over surfaces. This last is still an open problem, as observed in [BMTB*18], and a quick and well performing technique such as the mpLBP is for sure a meaningful contribution towards a possible solution.

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 4: *Computational times for edgeLBP/mpLBP (in seconds). The top-left cell of each Table indicates the number x of vertices.*

6. Acknowledgments

This study was partially supported by the CNR-IMATI projects DIT.AD004.028.001 and DIT.AD021.080.001, and the ANR PAPS project ANR-14-CE27-0003.

References

[ABCO*01] Alexa M., Behr J., Cohen-Or D., Fleishman S., Levin D., Silva C. T.: Point set surfaces. In *Proceedings of the Conference on Visualization ’01* (Washington, DC, USA, 2001), VIS ’01, IEEE Computer Society, pp. 21–28. 2

[AF06] Attene M., Falcidieno B.: Remesh: An interactive environment to edit and repair triangle meshes. In *Proc. SMI’06* (2006), IEEE Computer Society, p. 41. 5

[AKK00] Albuz E., Kocalar E. D., Khokhar A. A.: Quantized CIELab* space and encoded spatial structure for scalable indexing of large color image archives. In *ICASSP ’00* (2000), vol. 6, pp. 1995–1998. 5

[BDC18] BÃĽarzi Y., Digne J., Chaine R.: Wavejets: A local frequency framework for shape details amplification. *Computer Graphics Forum* (2018). 2

[BMTA*17] Biasotti S., Moscoso Thompson E., Aono M., Hamza A. B., Bustos B., Dong S., Du B., Fehri A., Li H., Limberger F. A., Masoumi M., Rezaei M., Sipiran I., Sun L., Tatsuma A., Forero S. V., Wilson R. C., Wu Y., Zhang J., Zhao T., Fornasa F., Giachetti A.: Retrieval of surfaces with similar relief patterns. In *Eurographics Workshop on 3D Object Retrieval* (2017), Pratikakis I., Dupont F., Ovsjanikov M., (Eds.), The Eurographics Association. 1, 2, 4, 5

[BMTB*18] Biasotti S., Moscoso Thompson E., Barthe L., Berretti S., Giachetti A., Lejemble T., Mellado N., Moustakas K., Manolas I., Dimou D., Tortorici C., Velasco-Forero S., Werghi N., Polig M., Sorrentino G., Hermon S.: Recognition of Geometric Patterns Over 3D Models. In *Eurographics Workshop on 3D Object Retrieval* (2018), Telea A., Theoharis T., Veltkamp R., (Eds.), The Eurographics Association. 7

[BYRN99] Baeza-Yates R. A., Ribeiro-Neto B.: *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999. 5

[CMK*14] 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* (Washington, DC, USA, 2014), CVPR '14, IEEE Computer Society, pp. 3606–3613. 2

[CMKV16] Cimpoi M., Maji S., Kokkinos I., Vedaldi A.: Deep filter banks for texture recognition, description, and segmentation. *International Journal of Computer Vision 118*, 1 (May 2016), 65–94. 2

[CP03] Cazals F., Pouget M.: Estimating differential quantities using polynomial fitting of osculating jets. In *SGP '03* (2003), pp. 177–187. 2

[DT05] Dalal N., Triggs B.: Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Conference on* (2005), vol. 1, pp. 886–893. 2

[FBF77] Friedman J. H., Bentley J. L., Finkel R. A.: An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw. 3*, 3 (Sept. 1977), 209–226. 2

[Gia18] Giachetti A.: Effective characterization of relief patterns. *Computer Graphics Forum 37*, 5 (2018), 83–92. 1, 2, 5, 6

[HCDC17] Hamdi-Cherif A., Digne J., Chaine R.: Super-resolution of point set surfaces using local similarities. *Computer Graphics Forum* (2017). 2

[HP11] Hunt R. W. G., Pointer M. R.: *Measuring Colour, Fourth Edition*. Wiley, 2011. 5

[KvD92] Koenderink J. J., van Doorn A. J.: Surface shape and curvature scales. *Image & Vision Computing 10*, 8 (1992), 557 – 564. 2

[LFG*17] Liu L., Fieguth P., Guo Y., Wang X., PietikÃĎinen M.: Local binary features for texture classification: Taxonomy and experimental study. *Pattern Recognition 62* (2017), 135 – 160. 2

[Low04] Lowe D. G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision 60*, 2 (2004), 91–110. 2

[MTB18a] Moscoso Thompson E., Biasotti S.: Description and retrieval of geometric patterns on surface meshes using an edge-based lbp approach. *Pattern Recognition 82* (2018), 1 – 15. 1, 2, 4, 5

[MTB18b] Moscoso Thompson E., Biasotti S.: Edge-based LBP Description of Surfaces with Colorimetric Patterns. In *Eurographics Workshop on 3D Object Retrieval* (2018), Telea A., Theoharis T., Veltkamp R., (Eds.), The Eurographics Association. 2

[MTB19] Moscoso Thompson E., Biasotti S.: Retrieving color patterns on surface meshes using edgelbp descriptors. *Computers & Graphics 79* (2019), 46 – 57. 2, 6

[MTBS*18] Moscoso Thompson E., Biasotti S., Sorrentino G., Polig M., Hermon S.: Towards an Automatic 3D Patterns Classification: the GRAVITATE Use Case. In *Eurographics Workshop on Graphics and Cultural Heritage* (2018), Sablatnig R., Wimmer M., (Eds.), The Eurographics Association. 1

[MTTW*18] Moscoso Thompson E., Tortorici C., Werghi N., Berretti S., Velasco-Forero S., Biasotti S.: Retrieval of Gray Patterns Depicted on 3D Models. In *Eurographics Workshop on 3D Object Retrieval* (2018), Telea A., Theoharis T., Veltkamp R., (Eds.), The Eurographics Association, pp. 63–69. 2, 4, 6

[OGG09] Oztireli A. C., Guennebaud G., Gross M.: Feature preserving point set surfaces based on non-linear kernel regression. *Computer Graphics Forum 28* (2009), 493–501(9). 2

[OPH96] Ojala T., Pietikäinen M., Harwood D.: A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition 29*, 1 (1996), 51–59. 1, 2, 3

[OPM02] Ojala T., Pietikainen M., Maenpaa T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE TPAMI 24*, 7 (2002), 971–987. 1, 2

[OVSP13] Othmani A., Voon L. F. L. Y., Stolz C., Piboule A.: Single tree species classification from terrestrial laser scanning data for forest inventory. *Pattern Recognition Letters 34*, 16 (2013), 2144–2150. 1, 2

[Pey] Peyre G.: Toolbox graph - A toolbox to process graph and triangulated meshes. http://www.ceremade.dauphine.fr/∼peyre/matlab/graph/content.html. 5

[PHZA11] Pietikäinen M., Hadid A., Zhao G., Ahonen T.: *Computer Vision Using Local Binary Patterns*, vol. 40 of *Computational Imaging and Vision*. Springer, 2011. 2, 3

[RC11] Rusu R. B., Cousins S.: 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)* (Shanghai, China, May 9-13 2011). 5

[Rij79] Rijsbergen C. J. V.: *Information Retrieval*, 2nd ed. Butterworth-Heinemann, Newton, MA, USA, 1979. 5

[Sal65] Salton G.: The evaluation of automatic retrieval procedures - selected test results using the smart system. *American Documentation 16*, 3 (1965), 209–222. 5

[SMKF04] Shilane P., Min P., Kazhdan M., Funkhouser T.: The princeton shape benchmark. In *Shape modeling applications, 2004. Proceedings* (2004), IEEE, pp. 167–178. 5

[The18] The CGAL Project: *CGAL User and Reference Manual*, 4.13 ed. CGAL Editorial Board, 2018. URL: `https://doc.cgal.org/4.13/Manual/packages.html`. 5

[TWB18] Tortorici C., Werghi N., Berretti S.: Performing Image-like Convolution on Triangular Meshes. In *Eurographics Workshop on 3D Object Retrieval* (2018), Telea A., Theoharis T., Veltkamp R., (Eds.), The Eurographics Association. 2

[WBB15] Werghi N., Berretti S., Bimbo A. D.: The mesh-LBP: A framework for extracting local binary patterns from discrete manifolds. *IEEE Trans. Image Processing 24*, 1 (2015), 220–235. 1, 2

[WLKT09] Wei L.-Y., Lefebvre S., Kwatra V., Turk G.: State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR* (2009), pp. 93–117. 1

[WTBB15] Werghi N., Tortorici C., Berretti S., Bimbo A. D.: Local binary patterns on triangular meshes: Concept and applications. *Computer Vision and Image Understanding 139* (2015), 161–177. 1, 2

[WTBB16] Werghi N., Tortorici C., Berretti S., Bimbo A. D.: Boosting 3D LBP-based face recognition by fusing shape and texture descriptors on the mesh. *IEEE Trans. Information Forensics and Security 11*, 5 (2016), 964–979. 2

[ZPS*16] Zeppelzauer M., Poier G., Seidl M., Reinbacher C., Schulter S., Breiteneder C., Bischof H.: Interactive 3D segmentation of rock-art by enhanced depth maps and gradient preserving regularization. *J. Comput. Cult. Herit. 9*, 4 (Sept. 2016), 19:1–19:30. 1, 2