mpLBP: A point-based representation for surface pattern description
Elia Moscoso Thompson, Silvia Biasotti, Julie Digne, Raphaëlle Chaine

To cite this version:

HAL Id: hal-02406554
https://hal.archives-ouvertes.fr/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\textsuperscript{a,}\textsuperscript{*}, Silvia Biasotti\textsuperscript{a}, Julie Digne\textsuperscript{b}, Raphaëlle Chaine\textsuperscript{b}

\textsuperscript{a}Istituto di Matematica Applicata e Tecnologie Informatiche ‘E. Magenes’ - CNR
\textsuperscript{b}LIRIS, Université Claude Bernard Lyon 1, CNRS, France

\textbf{A R T I C L E I N F O}

\textbf{Article history:}
Received December 6, 2019

\textbf{Keywords:} Computers and Graphics, Shape analysis, Pattern retrieval

\textbf{A B S T R A C T}

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 \cite{1}, the analysis of artworks styles \cite{2}, the classification of fabric patterns \cite{3} or the categorization of objects \cite{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: \textit{geometric} patterns that represent small variations on the surface geometry, e.g., repeated, small incisions, chiselings, bumps, etc.; and \textit{colorimetric} ones, e.g., elements with small painted decorations on the surface. Figure\cite{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 \cite{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
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 [15]. 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 LBPR descriptor [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 [23] or mesh color if such information is provided). The edgeLBP [10, 25, 11, 8, 15, 4] performs an LBPR 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 LBPR 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 $\pi$, 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 $j = 1 \cdots n_{rad}$, 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 $\theta_k$. 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:

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

where $\theta_k = k \frac{\pi}{P_j}, \ k = 1 \cdot P_j$. Finally, we assign to each sector $(j,k)$ a value $sec(\tilde{p})^k_j$ 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 $1 \cdot \sum P_j$ 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 $\pi$ 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 $R_{max}$, which is the maximum value for the parameter $R$ such that all the points on $M$ have an acceptable description.
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 [1] which we briefly summarize. For each pixel \( p \), the set of pixels \( \bar{p_j} \) 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 \( h(\bar{p_j}) \leq h(p) \) 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 \( \bar{p_j} \). 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 \( \bar{p} \) defined as in Section 5.1. 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 \( \bar{p} \). Thus, each sector can be seen as the evaluation of \( h \) at a sample of the surface. For all the points \( \bar{p} \) in \( S \), we define \( LBP(\bar{p}) \) the feature vector of \( n_{rad} \) elements as follows:

\[
LBP(\bar{p})_j = \sum_k (str(\bar{p}))_k.
\]

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 \( \sum_s (P_s + 1) \) 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 \( mpLBP(S)_j \) is equal to the number of points \( \bar{p} \) in \( S \) such that \( LBP(\bar{p})_j = m \) (with \( j = 1, ..., n_{rad} \) and \( m = 0, ..., P_j \)).

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 Fig-
  ure \( \text{[2]b} \), \( R \) 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 bub-
  ble should contain at least one circle entirely).

- \( n_{\text{rad}} \): 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 \( P_j \) guar-
antee that all the sectors have the same area. For this reason,
we selected \( P_j = \text{multP}(2j - 1) \), \( \text{multP} \in \mathbb{N}_+ \). In this case, \( P_j \)
depends on \( n_{\text{rad}} \). This degree of freedom was tuned by the pa-
parameter \( \text{multP} \) (that replaces the \( P_j \) parameters). For instance,
in Figure \( \text{[2]c} \) the parameters are \( n_{\text{rad}} = 7 \) and \( \text{multP} = 2 \), 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 con-
verts a patch of a surface into a sort of image. Therefore, it
would make sense to straightforwardly adopt a square neigh-
borhood and its variations, following the image literature on
pattern recognition. In general, it is also possible to define var-
ious 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 sam-
pling strategies. In Section \([5,5]\) we quantitatively discuss the
performance of these neighborhood choices over a well-known
benchmark.

- **Scheme 1**: same concept of the mpLBP punctual descrip-
tor, but both the descriptor and the sectors are shaped as
squares. Figure \([2]a\) 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 ra-
dius, 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 eval-
uate its performances. Moreover, with respect to what re-
ported in \([12]\), we run additional tests on the benchmarks con-
sidered in the previous year and two additional datasets are
considered, extracted from a collection of archaeological frag-
ments.

4.1. Datasets

**SHREC17 Benchmark, geometric patterns.** The SHREC’17
benchmark \([3]\) on the retrieval of relief patterns is com-
posed 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 differ-
et 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 sta-
ibility 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 Sci-
cence 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
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 ground truth 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].

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 \textit{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 $\pi$ defined in Section 5.1.
Table 1: Base models and patterns used in the SHREC’18 benchmark. Each pattern 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. 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).

<table>
<thead>
<tr>
<th>Base model</th>
<th>Patterns</th>
<th>Surfaces with patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Base models" /></td>
<td><img src="image2" alt="Patterns" /></td>
<td><img src="image3" alt="Surfaces with patterns" /></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>GRAVITATE(geo)</th>
<th>GRAVITATE(col)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feathered Pattern</td>
<td>Guilloche</td>
</tr>
<tr>
<td>Pattern of Circlets</td>
<td>Six Petals</td>
</tr>
<tr>
<td>Spirals</td>
<td>Chequer</td>
</tr>
<tr>
<td>Line Pattern</td>
<td>Striped Band</td>
</tr>
<tr>
<td>Smooth Fringe</td>
<td>Pattern of circles (P.o.C) (Painted)</td>
</tr>
<tr>
<td>Hatched Fringe</td>
<td>Lotus and Buds</td>
</tr>
<tr>
<td>Scales v1</td>
<td>Scales v2</td>
</tr>
<tr>
<td>Guilloche v2</td>
<td>Pattern of Curves</td>
</tr>
</tbody>
</table>

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 $h_j$ 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 clarity.

5. Experimental results and discussions

This Section summarizes the mpLB 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 $\text{mult}P = 4$. Since most...
Table 1. Results on the SHREC’17 benchmark, both the Original (Top) and the Complete (Bottom) Dataset. Runs marked with * are those where the point neighborhood are rotated by a random angle around the normal of the respective point.

<table>
<thead>
<tr>
<th>Method</th>
<th>NN</th>
<th>1-Tier</th>
<th>2-Tier</th>
<th>mAP</th>
<th>e</th>
<th>nDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Dataset</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMC-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KLBP-FV/TKS</td>
<td>0.633</td>
<td>0.363</td>
<td>0.494</td>
<td>0.590</td>
<td>0.293</td>
<td>0.062</td>
</tr>
<tr>
<td>edgeLBP - run2</td>
<td>0.522</td>
<td>0.295</td>
<td>0.412</td>
<td>0.307</td>
<td>0.247</td>
<td>0.003</td>
</tr>
<tr>
<td>T/mc/SIFT/FV</td>
<td>0.872</td>
<td>0.710</td>
<td>0.849</td>
<td>0.741</td>
<td>0.447</td>
<td>0.088</td>
</tr>
<tr>
<td>mpLBP - Cmax</td>
<td>0.933</td>
<td>0.706</td>
<td>0.845</td>
<td>0.744</td>
<td>0.449</td>
<td>0.087</td>
</tr>
<tr>
<td>mpLBP - Cmean</td>
<td>0.911</td>
<td>0.733</td>
<td>0.863</td>
<td>0.861</td>
<td>0.427</td>
<td>0.075</td>
</tr>
<tr>
<td>mpLBP - Cmean*</td>
<td>0.917</td>
<td>0.733</td>
<td>0.863</td>
<td>0.761</td>
<td>0.427</td>
<td>0.087</td>
</tr>
<tr>
<td>Complete Dataset</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMC-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KLBP-FV/TKS</td>
<td>0.763</td>
<td>0.272</td>
<td>0.389</td>
<td>0.271</td>
<td>0.261</td>
<td>0.068</td>
</tr>
<tr>
<td>edgeLBP - run2</td>
<td>0.966</td>
<td>0.333</td>
<td>0.449</td>
<td>0.339</td>
<td>0.332</td>
<td>0.759</td>
</tr>
<tr>
<td>T/mc/SIFT/FV</td>
<td>0.993</td>
<td>0.712</td>
<td>0.850</td>
<td>0.739</td>
<td>0.667</td>
<td>0.929</td>
</tr>
<tr>
<td>mpLBP - Cmax</td>
<td>0.994</td>
<td>0.678</td>
<td>0.820</td>
<td>0.738</td>
<td>0.653</td>
<td>0.931</td>
</tr>
<tr>
<td>mpLBP - Cmean</td>
<td>0.997</td>
<td>0.677</td>
<td>0.815</td>
<td>0.733</td>
<td>0.653</td>
<td>0.931</td>
</tr>
<tr>
<td>mpLBP - Cmean*</td>
<td>0.994</td>
<td>0.702</td>
<td>0.841</td>
<td>0.759</td>
<td>0.665</td>
<td>0.938</td>
</tr>
<tr>
<td>mpLBP - SI</td>
<td>0.999</td>
<td>0.868</td>
<td>0.813</td>
<td>0.737</td>
<td>0.650</td>
<td>0.930</td>
</tr>
<tr>
<td>mpLBP - HF</td>
<td>0.999</td>
<td>0.383</td>
<td>0.480</td>
<td>0.431</td>
<td>0.419</td>
<td>0.802</td>
</tr>
<tr>
<td>mpLBP - Cmean</td>
<td>0.994</td>
<td>0.733</td>
<td>0.863</td>
<td>0.761</td>
<td>0.427</td>
<td>0.075</td>
</tr>
<tr>
<td>mpLBP - Cmean*</td>
<td>0.997</td>
<td>0.733</td>
<td>0.863</td>
<td>0.761</td>
<td>0.427</td>
<td>0.075</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Run</th>
<th>NN</th>
<th>FT</th>
<th>ST</th>
<th>mAP</th>
<th>e</th>
<th>nDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>TWB3</td>
<td>0.755</td>
<td>0.502</td>
<td>0.688</td>
<td>0.577</td>
<td>0.455</td>
<td>0.795</td>
</tr>
<tr>
<td>TWB4</td>
<td>0.292</td>
<td>0.517</td>
<td>0.731</td>
<td>0.593</td>
<td>0.481</td>
<td>0.808</td>
</tr>
<tr>
<td>edgeLBP-R4</td>
<td>0.915</td>
<td>0.317</td>
<td>0.789</td>
<td>0.566</td>
<td>0.600</td>
<td>0.898</td>
</tr>
<tr>
<td>edgeLBP-R5</td>
<td>0.950</td>
<td>0.740</td>
<td>0.892</td>
<td>0.790</td>
<td>0.606</td>
<td>0.911</td>
</tr>
<tr>
<td>mpLBP - set1</td>
<td>0.965</td>
<td>0.739</td>
<td>0.862</td>
<td>0.781</td>
<td>0.600</td>
<td>0.910</td>
</tr>
<tr>
<td>mpLBP - set2</td>
<td>0.990</td>
<td>0.744</td>
<td>0.864</td>
<td>0.762</td>
<td>0.590</td>
<td>0.900</td>
</tr>
</tbody>
</table>

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.

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 is 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,
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 $\lambda_v$, expressed as a percentage of the diameter of the smallest bounding sphere. The values of $\lambda_v$ 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 $\lambda_c$, 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 geometrical 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 mlBP 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).

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

<table>
<thead>
<tr>
<th>Class considered</th>
<th>NN</th>
<th>FT</th>
<th>ST</th>
<th>e</th>
<th>nDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feathered Pattern</td>
<td>1.000</td>
<td>0.722</td>
<td>0.911</td>
<td>0.434</td>
<td>0.934</td>
</tr>
<tr>
<td>Pattern of Circles</td>
<td>1.000</td>
<td>0.922</td>
<td>0.978</td>
<td>0.439</td>
<td>0.989</td>
</tr>
<tr>
<td>Spirals</td>
<td>1.000</td>
<td>0.644</td>
<td>0.756</td>
<td>0.385</td>
<td>0.911</td>
</tr>
<tr>
<td>Line Pattern</td>
<td>0.900</td>
<td>0.300</td>
<td>0.367</td>
<td>0.263</td>
<td>0.661</td>
</tr>
<tr>
<td>Smooth Fringe</td>
<td>1.000</td>
<td>0.467</td>
<td>0.500</td>
<td>0.229</td>
<td>0.804</td>
</tr>
<tr>
<td>Hatched Fringe</td>
<td>1.000</td>
<td>0.400</td>
<td>0.598</td>
<td>0.371</td>
<td>0.773</td>
</tr>
<tr>
<td>Overall</td>
<td>0.950</td>
<td>0.576</td>
<td>0.683</td>
<td>0.354</td>
<td>0.845</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class considered</th>
<th>NN</th>
<th>FT</th>
<th>ST</th>
<th>e</th>
<th>nDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feathered Pattern</td>
<td>1.000</td>
<td>0.511</td>
<td>0.767</td>
<td>0.410</td>
<td>0.849</td>
</tr>
<tr>
<td>Pattern of Circles</td>
<td>1.000</td>
<td>0.889</td>
<td>0.944</td>
<td>0.439</td>
<td>0.976</td>
</tr>
<tr>
<td>Spirals</td>
<td>1.000</td>
<td>0.544</td>
<td>0.700</td>
<td>0.400</td>
<td>0.870</td>
</tr>
<tr>
<td>Line Pattern</td>
<td>0.800</td>
<td>0.289</td>
<td>0.444</td>
<td>0.302</td>
<td>0.685</td>
</tr>
<tr>
<td>Smooth Fringe</td>
<td>1.000</td>
<td>0.522</td>
<td>0.533</td>
<td>0.239</td>
<td>0.818</td>
</tr>
<tr>
<td>Hatched Fringe</td>
<td>1.000</td>
<td>0.400</td>
<td>0.500</td>
<td>0.366</td>
<td>0.765</td>
</tr>
<tr>
<td>Overall</td>
<td>0.967</td>
<td>0.526</td>
<td>0.648</td>
<td>0.359</td>
<td>0.827</td>
</tr>
</tbody>
</table>

### Table 4. mlBP performance for data corrupted with noise. Top: the Original Dataset of the SHREC’17 benchmark. Bottom: Single Pattern Dataset of the SHREC’18 benchmark.

<table>
<thead>
<tr>
<th>Method</th>
<th>NN</th>
<th>1-Tier</th>
<th>2-Tier</th>
<th>mAP</th>
<th>e</th>
<th>nDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>mlBP - set1 Clean</td>
<td>0.917</td>
<td>0.711</td>
<td>0.839</td>
<td>0.743</td>
<td>0.420</td>
<td>0.861</td>
</tr>
<tr>
<td>mlBP - set1, $\lambda_c = 0.2$</td>
<td>0.911</td>
<td>0.693</td>
<td>0.846</td>
<td>0.733</td>
<td>0.380</td>
<td>0.790</td>
</tr>
<tr>
<td>mlBP - set1, $\lambda_c = 0.4$</td>
<td>0.872</td>
<td>0.618</td>
<td>0.709</td>
<td>0.664</td>
<td>0.350</td>
<td>0.753</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>NN</th>
<th>1-Tier</th>
<th>2-Tier</th>
<th>mAP</th>
<th>e</th>
<th>nDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>mlBP - set1 Clean</td>
<td>0.965</td>
<td>0.739</td>
<td>0.862</td>
<td>0.781</td>
<td>0.600</td>
<td>0.910</td>
</tr>
<tr>
<td>mlBP - set1, $\lambda_c = 5$</td>
<td>0.919</td>
<td>0.514</td>
<td>0.653</td>
<td>0.586</td>
<td>0.440</td>
<td>0.822</td>
</tr>
<tr>
<td>mlBP - set1, $\lambda_c = 7$</td>
<td>0.755</td>
<td>0.332</td>
<td>0.485</td>
<td>0.355</td>
<td>0.355</td>
<td>0.941</td>
</tr>
</tbody>
</table>

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

These timings are those obtained on the 120 vertices mesh, different schemes and that $n_{rad}$ and $P$ do not affect the computation times 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 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
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 objects, height-fields and so on). A further extension is the application of the punctual descriptor used by the mpLBP 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 DIIT.DA04.028.001 and DIIT.DA021.080.001, and the e-Roma project from the Agence Nationale de la Recherche (ANR-16-CE38-0009).

References


[39] Peyrè, G. Graph Toolbox - a toolbox to process graph and triangulated meshes. 2009. URL: http://www.ceremade.dauphine.fr/$\texttt{simpeyre/matlab/graph/content.html}$


