

See discussions, stats, and author profiles for this publication at: <http://www.researchgate.net/publication/282931520>

Local binary patterns on triangular meshes: Concept and applications

ARTICLE *in* COMPUTER VISION AND IMAGE UNDERSTANDING · AUGUST 2015

Impact Factor: 1.54 · DOI: 10.1016/j.cviu.2015.03.016

READS

24

4 AUTHORS, INCLUDING:

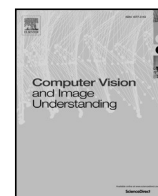


[Naoufel Werghi](#)

Khalifa University of Science Technology & R...

94 PUBLICATIONS 638 CITATIONS

SEE PROFILE



Local binary patterns on triangular meshes: Concept and applications



Naoufel Werghi^a, Claudio Tortorici^a, Stefano Berretti^{b,*}, Alberto del Bimbo^b

^a Khalifa University of Science Technology and Research, Abu Dhabi, United Arab Emirates

^b Information Engineering, University of Florence, 50139, Florence, Italy

ARTICLE INFO

Article history:

Received 17 June 2014

Accepted 29 March 2015

Keywords:

3D local binary patterns

Mesh manifold

Ordered ring facets

3D texture classification

3D texture retrieval

3D face matching

ABSTRACT

In this paper, we introduce an original framework for computing local binary like-patterns on 2D mesh manifolds (i.e., surfaces in the 3D space). This framework, dubbed mesh-LBP, preserves the simplicity and the adaptability of the 2D LBP and has the capacity of handling both open and close mesh surfaces without requiring normalization as compared to its 2D counterpart. We describe the foundations and the construction of mesh-LBP and showcase the different LBP patterns that can be generated on the mesh. In the experimentation, we provide evidence of the uniform patterns in the mesh-LBP, the repeatability of its descriptors, and its robustness to moderate shape deformations. Then, we show how the mesh-LBP descriptors can be adapted to a number of surface local and global analysis including 3D texture classification and retrieval, and 3D face matching. We also compare the performance of the mesh-LBP descriptors with a bunch of state of the art surface descriptors.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

The rapid advancement of the 3D imaging technologies resulted in a new generation of acquisition devices capable of capturing the 3D objects geometry in the physical three-dimensional space. High-resolution 3D static scanners as well as devices with 3D dynamic acquisition capabilities that provide a continuous flow of the 3D geometry of a scene are now available. In addition to this, the geometric and RGB information are often captured in a synchronized way. In this respect, the interest and the widespread use of Kinect-like cameras in consumer and research applications is one of the most evident advancement in the field.

The geometric information captured by such 3D acquisition devices is typically in the form of a cloud of points, which represent the three-dimensional coordinates of a set of samples of the 3D object surface. However, the direct processing of these point clouds is not convenient or even possible, so that other representation formats have been established. Depth images are one of the most commonly used imaging modality, since they permitted a direct extension to the depth dimension of many computer vision and pattern recognition solutions developed for analyzing the photometric information in 2D images. Though the possibility of a straightforward extension of 2D techniques is attractive, this modality loses the full 3D geometry, by reducing it to a 2.5D projection. The full 3D shape information

is instead preserved and encoded in a simple, compact and flexible format by the triangular mesh manifold modality. This is widely used in many fields, such as animation, medical imaging, computer-aided design and many others. The recent advances in shape scanning and modeling have also allowed the integration of both photometric and geometric information into a single support defined over a 2D mesh-manifold. However, despite the abundance and the richness of the mesh manifold modality, the number of solutions for shape representation that exploit the full 3D geometry of the objects is still limited, and not comparable with the large variety of methods available in 2D. Indeed, many effective solutions capable of capturing discriminative information have been developed for 2D still images and videos, but their 3D counterpart is often not available.

An example of such successful 2D descriptors is represented by the local binary pattern (LBP). Since its first formal definition by Ojala et al. [1,2], the LBP has established itself as one of the most effective local shape descriptors for image representation. It has been originally introduced for representing 2D textures in still images, but its computational simplicity and discriminative power attracted the attention of the image processing and pattern recognition community for other different tasks. Rapidly, LBP has found applications in visual inspection [3,4], remote sensing [5–7], face recognition [8–11], facial expression recognition [12], and motion analysis [13,14]. However, the LBP-based methods developed so far operate either on photometric information provided by 2D color images or on geometric information in 2D depth images. The few solutions that extract surface features directly in 3D (typically in the form of surface normals), resort to the 2D case by converting the 3D extracted

* Corresponding author.

E-mail address: stefano.berretti@unifi.it, berretti@dsi.unifi.it (S. Berretti).

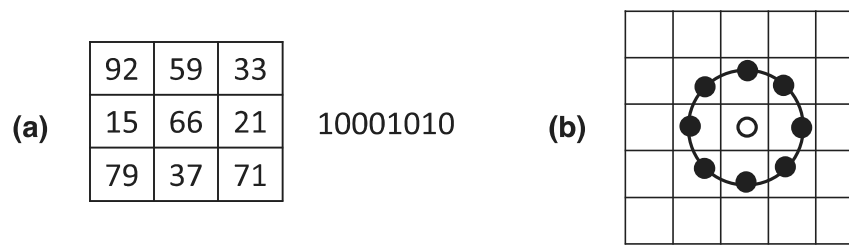


Fig. 1. (a) Computation of the basic LBP code from the 3×3 neighborhood of a central pixel. Each pixel, starting from the upper-left corner is compared with the central pixel to produce 1 if its value is greater or equal, 0 otherwise. The result is an 8-bit binary code. (b) Example of a central pixel with a circular neighborhood of given radius.

features to depth values, and then use ordinary LBP processing on 2D images [15–17]. However, to the best of our knowledge, there is no a framework that allows the computation of LBP on a mesh manifold. Since LBP requires an ordered support for its computation, the major factor that contributed in this lack is the absence of an intrinsic order in the triangular mesh manifold. On the contrary, computation of LBP on 2D images, either photometric or depth, benefits from the implicit ordering of the pixels in the 2D image array.

Motivated by these facts, in this paper we address the challenge of computing LBP on a mesh manifold by proposing an original framework that we call mesh-LBP, which allows the extraction of LBP-like patterns directly from a triangular mesh manifold, without the need of any intermediate representation in the form of depth images. With this framework, we can therefore build on the current 2D-LBP analysis methods, extending them to mesh manifolds as well as to the modality that also embeds photometric information into mesh models. To motivate our solution and to relate it to the state of the art approaches, next we provide an overview of the LBP literature.

1.1. Related work

According to its original definition [1], the LBP operator assigns labels to image pixels performing the following steps: given a pixel (central pixel), first its 3×3 neighborhood is considered, and the gray value of each pixel in the neighborhood is thresholded with the value of the central pixel (i.e., each pixel in the neighborhood is regarded as 1 if its value is greater or equal to the central value, 0 otherwise); then, the sequence of 0/1 in the neighborhood of the central pixel is regarded as a binary number according to a positional coding convention and considered as the LBP value of the central pixel. This is shown in Fig. 1(a), where the upper left pixel in the neighborhood is regarded as the most significant bit in the final code. This eight bits number encodes the mutual relationship between the gray levels of the central pixel and of its neighboring pixels. The histogram of the numbers obtained in such a way can then be used as a texture descriptor of a region or of an entire image. This operator is distinguished by its simplicity, efficient computation, and invariance to monotonic gray-level transformations. Later, the same authors of the basic LBP proposed an extended version that can operate on circular neighborhood of different radii [2], also allowing sub-pixel alterations (see Fig. 1(b)).

These initial formulations led subsequently to the definition of other neighborhood variants, like the oriented elliptic neighborhood LBP (elongated LBP) proposed by Liao and Chung [18], which accounts for anisotropic information, and the multi-block LBP (MB-LBP) that compares the averages of the gray level intensity of neighboring pixels rather than the value of individual pixels, in order to capture macrostructural features in the image [19]. Other versions have been proposed to improve the discriminative power of the descriptor, such as the improved LBP (ILBP) [20], in which pixel values are compared with the average of the neighborhood, and the extended LBP (ELBP) [21] which encodes, in addition to the binary comparison between pixels values, the amplitude of their difference using additive binary

digits. To improve the robustness of LBP, Tan and Triggs [22] introduced the so-called local ternary pattern (LTP), which substitutes the original binary code by a three-values code (1, 0 and -1) by means of a user-defined threshold. This new operator addressed the sensitivity to noise, though at the cost of losing the invariance to monotonic gray-level transformations. A fuzzy-logic version of the LTP was proposed later in [23], where a fuzzy membership function substituted the crisp three-states association used in [22]. A more complete list and discussion on the many LBP variants appeared in the literature can be found in [24].

Considering the case of 3D shape analysis, most if not all the LBP-based approaches have been developed for face recognition applications. Many of the techniques developed in this context operate on standard depth images, where the z-coordinate is mapped to a gray-level value. This format allowed a straightforward application of the 2D-LBP operator as it was demonstrated in the pioneering work of Li et al. [25]. Later, Huang et al. [26] proposed a 3D-LBP operator that also encodes depth differences of neighboring pixels. More recently, Huang et al. [27] extended the 3D-LBP to a multiscale extended LBP (eLBP), which consists of several LBP codes in multiple layers accounting for the exact gray value differences between the central pixel and its neighbors. Sandbach et al. [15] proposed a local normal binary pattern (LNBP), which used the angle between normals at two points, rather than the depth value to obtain the local binary code. Similar to this, in [16] the surface normals are extracted in 3D, then the values of the normal components along the direction of the three coordinate axes are interpreted as depth values, and LBP is computed on these depth maps reporting the values of the normal components. The idea of exploiting surface normals is further extended in [17], where azimuthal projection distance images are constructed. The azimuthal equidistant projection is able to project normals onto points in an Euclidean space according to the direction. Though the projected information is not the depth, depending on the normals of the 3D surface, 2D LBP are still computed on the projection images. Fehr and Burkhardt [28] attempted a 3D-LBP definition specifically tailored for volumetric data by sampling a sphere of a given radius around a central voxel. The approach is computationally expensive in that the rotation-invariance had to be addressed with complex techniques involving spherical correlation in the frequency domain.

1.2. Paper contribution and organization

The analysis above evidences that the LBP descriptor has attracted great interest for the analysis of 2D images, mainly for its simple and efficient computation and for the effective results that can be achieved relying on the LBP theory. Recently, various attempts have been made for extending the LBP framework to the case of 3D meshes, but none of them succeeded in addressing all the issues posed by the need for a simple and effective processing directly performed on a mesh-manifold. Indeed, existing solutions address the LBP extraction on 3D meshes by resorting to the easier 2D case, through the projection of 3D meshes on 2D depth images or by computing normals or

normal-related quantities in 3D and mapping them to a 2D map of values, which is then processed following the conventional LBP approach.

In this paper, we propose a framework that we call mesh-LBP, for designing and extracting local binary patterns directly from a mesh-manifold. In addition to its originality, the proposed framework is characterized by the following features:

- *Effectiveness*—The mesh-LBP operates directly on 3D triangular meshes, thus avoiding any expensive pre-processing, such as registration and normalization, required to obtain depth images.
- *Generalization*—The ability of handling mesh data, permits the mesh-LBP to deal with a larger spectrum of surfaces (e.g., closed, open, self-occluded) as compared to its counterpart defined on depth images. In addition, geometric and photometric information can be managed in a unified framework.
- *Adaptability*—This framework can be adapted to hold most if not all the LBP variants proposed in the literature for 2D and depth images.
- *Simplicity*—The mesh-LBP preserves the simplicity of the original LBP, not requiring any surface parameterization, apart the standard mesh arrangement into facets and vertex arrays, while keeping linear computational complexity.

In addition to present the framework for computing mesh-LBP and show its properties, in this work we propose and experiment the mesh-LBP as a 3D shape descriptor in several visual recognition tasks. In particular, we evidence the capability of mesh-LBP to be used as 3D texture descriptor for classification and retrieval of 3D texture patches. Furthermore, we successfully applied the mesh-LBP to the problem of 3D face recognition, showing the potential of the descriptor for this task also in the presence of marked facial expressions.

The rest of the paper is organized as follows: In Section 2, we introduce the foundations of the mesh-LBP and present its multi-resolution extension. In Section 3, the properties of mesh-LBP are investigated, evidencing the robustness of the descriptor to rotations and irregular tessellations of the mesh, and showing the existence of uniform and repeatable patterns. Experimental evidence of the potential of the mesh-LBP in different application scenarios that involve local and global representations, and in comparison to state of the art solutions is reported in Section 4. Finally, concluding remarks and future research directions are drawn in Section 5.

2. The mesh-LBP framework

In defining the 2D-LBP framework, a central role is played by the gridded structure of images, which permits a natural ordering and traversing of pixels. This intrinsic simplicity in computing LBP on the image domain has inspired direct extension of LBP to depth images capturing 3D information, while keeping the image structure. Our approach is completely different from these previous extensions of LBP to the 3D domain, since it aims to compute LBP-like patterns directly on a triangular mesh, rather than from its depth image counterpart. To this end, the first requirement for extracting mesh-LBP is a scheme for constructing rings of facets around a central facet and for traversing them in an ordered fashion.

Let $S = \langle V, F \rangle$ be the triangular mesh representation of an open or closed surface, where V and F are, respectively, the set of vertices and facets of the mesh. In the following, we assume the mesh regular, i.e., each vertex has a valence of six. This property requires each vertex is shared between six facets of the mesh, but we will show later that our framework can also cope with meshes that do not comply with this ideal case. Let us consider a convex contour on the mesh, given by the edges of a set of facets. We start considering the particular case where the convex contour is constituted by the three edges of an individual facet (central facet). In Fig. 2(a), this case is shown by the white facet (central facet). In Fig. 2(a), this case is shown by the white facet (central facet), indicated with Fin since it is inside the convex contour given by

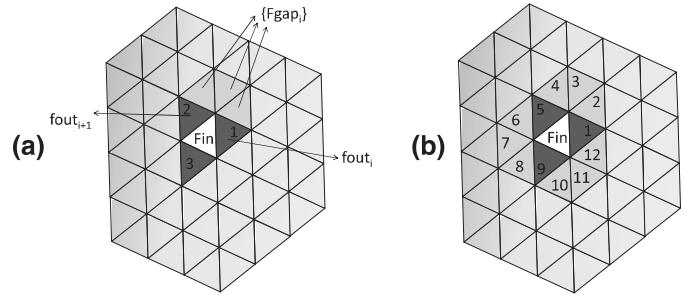


Fig. 2. Construction of an ordered ring of facets. (a) Initial $Fout$ facets (colored in blue) on a convex contour given by the three edges of a central facet (Fin), and the $Fgap$ facets (colored in yellow) bridging the gap between a pair of consecutive $Fout$ facets; (b) the 1st ring around the central facet given by the union of the $Fout$ and $Fgap$ facets. Facets of the ring are ordered counter clockwise. (For the interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

its three edges. Then, we consider the facets that share an edge with the Fin facet (the blue facets surrounding the Fin facet in Fig. 2(a)). We call these facets $Fout$ facets, as they seem pointing outside the convex contour defined by Fin .

Let us assume that the $Fout$ facets are initially ordered in a circular fashion across the contour given by Fin (in Fig. 2(a) the $Fout$ facets are numbered from 1 to 3 in an anti-clockwise fashion). Given that initial arrangement, we bridge the gap between each pair of consecutive $Fout$ facets, that is we extract the sequence of adjacent facets, located between the two consecutive $Fout$ facets and which share their common vertex (the vertex on the Fin contour) and are different from Fin . We call these facets $Fgap$ (see the yellow facets in Fig. 2(a), bridging the gap between the two consecutive facets $fout_1$ and $fout_2$). The “Bridge” procedure reported in pseudo-code in Algorithm 1 is used

Algorithm 1 Bridge.

Input: $fout_i, fout_{i+1}$ two consecutive $Fout$ facets sharing a vertex; fin_i facet which shares an edge with $fout_i$

Output: $Fgap_i$ set of consecutive $fgap$ facets bridging the gap between $fout_i$ and $fout_{i+1}$

```

procedure BRIDGE( $fout_i, fout_{i+1}, fin_i$ )
     $Fgap_i = [ ]$ 
     $v \leftarrow$  vertex shared by  $\{fout_i, fout_{i+1}\}$ 
     $gf \leftarrow$  facet adjacent to  $fout_i$ , different from  $fin_i$  and containing  $v$ 
     $prev \leftarrow fout_i$ 
    while  $gf \neq fout_{i+1}$  do
        append  $gf$  to  $Fgap_i$ 
         $new\_gf \leftarrow$  facet adjacent to  $gf$ , different from  $prev$  and containing  $v$ 
         $prev \leftarrow gf$ 
         $gf \leftarrow new\_gf$ 
    end while
    return  $Fgap_i$ 
end procedure

```

to compute the $Fgap$ facets. By iterating the process of bridging the gap between two consecutive $Fout$ facets with the $Fgap$ facets, results in a ring of facets that are ordered in a circular fashion (see the ring in Fig. 2(b)). The resulting arrangement of the ring facets inherits the same direction (clockwise or anti-clockwise) of the initial sequence of $Fout$ facets. The “GetRing” procedure of Algorithm 2 describes the ring construction, which is obtained by iterative calls to the “Bridge” procedure, resulting in a linear time complexity. We dubbed such obtained ordered ring as ordered ring facets (ORF). It can be observed that for the particular case of a contour formed by the three edges of an individual facet (and for a regular mesh), the obtained ring is

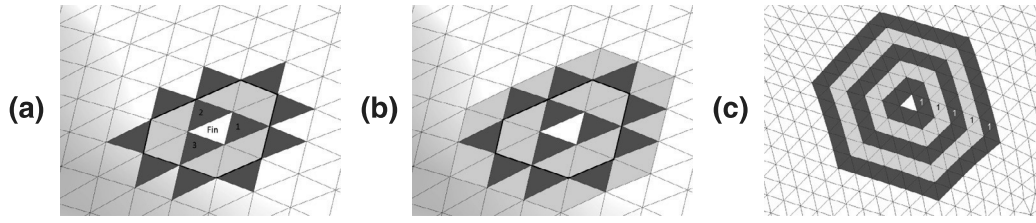


Fig. 3. Construction of multi-resolution mesh-LBP: (a) Extraction of the next set of *Fout* facets, as the facets adjacent to *Fgap*, which are not part of the current ring; (b) Extracting the *Fgap* facets of the second ordered ring; (c) Five concentric ordered rings. Notice that the first facet of each ring (marked by 1) is located at the same relative position.

Algorithm 2 GetRing.

Input: *Fout*, set of n ordered facets, $f_{out_1}, f_{out_2}, \dots, f_{out_n}$, lying on a convex contour; *Fin*, set of n ordered facets, $fin_1, fin_2, \dots, fin_n$, one-to-one adjacent to the *Fout* facets and located inside the region delimited by the convex contour (depending on the contour, *Fin* might include duplicates)

Output: *Ring*, set of facets in a ring-wise order

```

procedure GETRING(Fout, Fin)
  Ring = [ ]
  for all  $\langle f_{out_i}, f_{out_{i \% n + 1}} \rangle, i \leftarrow 1, \dots, n$  do
    append  $f_{out_i}$  to Ring
     $F_{gap_i} \leftarrow \text{BRIDGE}(f_{out_i}, f_{out_{i \% n + 1}}, fin_i)$ 
    append  $F_{gap_i}$  to Ring
  end for
  return Ring
end procedure

```

composed of 12 ordered facets union of the *Fout* and *Fgap* facets, as shown in Fig. 2(b).

Let $h(f) : S \rightarrow \mathcal{R}$, be a scalar function defined on the mesh S (e.g., photometric data or curvature of the surface). The circular ordering of the facets obtained with ORF allows us to compute a local binary operator in the same way as in the standard LBP. In fact, by comparing the scalar value of the facets in the ordered ring with the scalar value computed for the central facet results into a binary pattern (i.e., sequence of 0 and 1 digits). According to this, we define the basic mesh-LBP operator at a facet f_c by thresholding its ordered ring neighborhood f_k :

$$\text{meshLBP}(f_c) = \sum_{k=0}^{11} s(h(f_k) - h(f_c)) \cdot \alpha(k), \quad s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}, \quad (1)$$

where $\alpha(k)$ is a weighting function applied to the bits of the binary pattern. Different definitions of the $\alpha(k)$ permit us to obtain different mesh-LBP values from the binary pattern derived from the central facet and its ring neighborhood: for example, $\alpha(k) = 2^k$ results into the mesh counterpart of the basic LBP operator firstly suggested by Ojala et al. [1]; for $\alpha(k) = 1$, the sum of the digits of the pattern is computed (i.e., the sum is equal to the number of bits set to 1). In the experiments, we will refer to these two functions as the α_1 and α_2 operators, respectively. Finally, we remark here that for the present discussion it is not necessary to detail the particular scalar function $h(f)$ computed on the mesh surface. The effect of several different choices of this function will be given in Sections 3 and 4.

2.1. Multi-resolution mesh-LBP

The potential of the mesh-LBP descriptor can be fully exploited when the surface analysis is extended to neighborhood of the central facet larger than that constituted by the first ORF. This requires the construction of subsequent concentric rings, still using the concept of ORF. This extends our framework to a multi-resolution mesh-LBP.

From Fig. 3(a), we can observe that the first ORF is indeed a convex contour determined by the edges of the *Fgap* facets. This provides an immediate extension of the ORF at subsequent rings. First, from the first ring, the sequence of facets that are one-to-one adjacent to the *Fgap* facets are extracted (Fig. 3(a)). This sequence, which inherits the order property of the *Fgap* facets, constitutes the set of *Fout* facets for the subsequent ring. So, by filling the gap between each two consecutive facets of this sequence, a new ring, which exhibits the same ordered structure of its predecessor is obtained (Fig. 3(b)). By iterating this procedure, we build a sequence of concentric ordered rings, which represent the primitive entity for computing multi-resolution mesh-LBP (Fig. 3(c)). Details of the procedure used for computing the multi-ring structure are reported in Algorithm 3. In this case, the “GetRing” procedure of Algorithm 2 is slightly modified, so that it also returns the set of *Fgap* facets of the current ring and the set of *Fout* facets of the subsequent ring (indicated as *NewFout*).

Algorithm 3 MultiRing.

Input: *Fout_root*, initial set of ordered *Fout* facets; *Fin_root*, initial set of ordered *Fin* facets one-to-one adjacent to the *Fout* facets; *Nr*, number of rings to be constructed around *Fin_root*

Output: *Rings*, set of *Nr* rings of ordered facets constructed around *Fin_root*

```

procedure MULTIRING(Fout_root, Fin_root, Nr)
  Rings  $\leftarrow$  [ ]
  Fout  $\leftarrow$  Fout_root
  Fin  $\leftarrow$  Fin_root
  for  $i \leftarrow 1, Nr$  do
    (Ring, NewFout, Fgap)  $\leftarrow$  GETRING(Fout, Fin)
    append Ring to Rings
    Fout  $\leftarrow$  NewFout
    Fin  $\leftarrow$  Fgap
  end for
  return Rings
end procedure

```

In a real mesh, because of mesh tessellation irregularities, it might happen that the “GetRing” procedure gets trapped into a closed loop resulting in *NewFout* facets being located on the current ring or on duplicated instances. We fix such potential anomalies by simply checking the consistency of the obtained *NewFout* facets after each iteration.

Given a multi-ring constructed around a central facet f_c , the multi-resolution mesh-LBP operator is derived as follows:

$$\text{meshLBP}_m^r(f_c) = \sum_{k=0}^{m-1} s(h(f_k^r) - h(f_c)) \cdot \alpha(k), \quad (2)$$

where r is the ring number, and m is the number of facets uniformly spaced on the ring. The parameters r and m control, respectively, the radial resolution and the azimuthal quantization of the operator.

Finally, we observe that the topology of the neighborhood from which the mesh-LBP features are computed can be changed to accommodate the specificities of a given shape analysis application. In

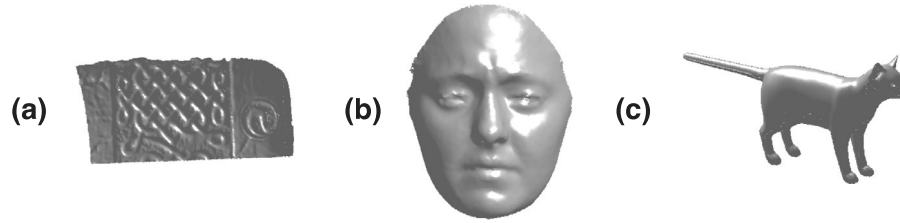


Fig. 4. Mesh models used in the experiment for detecting uniform patterns: (a) Portion of a pot (MIT CSAIL textured 3D models). (b) Face surface (BU-3DFE). (c) Cat model (TOSCA high-resolution).

fact, it can be shown that most, if not all, the different LBP neighborhood and operator variants proposed in the literature can be easily derived from the ordered rings structure of the mesh-LBP [29].

3. Mesh-LBP properties

The mesh-LBP framework reveals some interesting properties that are evidenced in the following: in Sections 3.1 and 3.2, we show two methods that endow the mesh-LBP representation of the required robustness to rotation and non-uniform tessellation of the mesh; in Sections 3.3 and 3.4, the existence of uniform patterns and the repeatability of mesh-LBP are evidenced.

3.1. Achieving invariance to facets ordering

The ORF approach provides an easy and efficient algorithm to cast the facets of a mesh to an ordered ring structure. However, bringing ORF to the mesh-LBP framework requires some invariant properties be fulfilled, so as to permit repeatable computation of the representation. In fact, in order to make the mesh-LBP invariant to the ordering of the facets in the ring and its traversal, two aspects should be addressed: The position of the first facet (i.e., the first *Fout* facet) in the ring, that is from which of the facets the ring starts from; the direction of the ring traversal (clockwise or anti-clockwise). The last aspect can be easily fixed by orienting the normals of the mesh-manifold. Some common solutions used to address the first aspect are:

- Performing a circular bit-wise shift of the binary pattern, as was suggested in the standard LBP [2], and selecting as initial facet that resulting in the minimum LBP value. However, this method reduces the range of the LBP values, and might seriously affect the discriminative power of the operator [30].
- Adopting intrinsically rotation invariant descriptors only. This set includes the number of transitions, the number of 1-valued bits, and the number of 1-valued runs of a given length in the binary patterns. This method preserves the range of the LBP values, yet might still compromise the discrimination power, though to a less extent than the first method.
- Considering all the binary pattern values that originate by moving the initial facet along the ring, but this solution creates redundancy and further burden the computation.

In this work, we have adopted a simpler yet practical solution. Given a sequence of ordered rings, we select as first facet f_1^r in each ring r , the facet f_i which satisfies the following condition:

$$\min_i \text{dist}(c_o, c_i^r), \quad f_i \in \text{ring-}r, \quad (3)$$

where $\text{dist}(\cdot)$ is the Euclidean distance, c_i^r is the center of facet f_i in the ring- r (union of the *Fout* and *Fgap* facets), and c_o is the centroid of the centers of the facets in the ring weighted by their area. Starting from f_1^r , the other facets in the ring are traversed in a clockwise manner. Using this approach throughout all the experiments of Sections 3 and 4, we can show its effective applicability in various contexts.

3.2. Mesh irregular tessellation

In Section 2, we assumed the meshes to be regular. When this condition holds, the number of facets v across the rings evolves according to the following arithmetic progression from ring i to ring $i + 1$:

$$v_{i+1} = v_i + 12. \quad (4)$$

Actually, due to local irregularities in the tessellation of real meshes, the assumption of vertex valence of six cannot hold. Consequently, the regular progression of Eq. (4) is not satisfied, thus hampering the repeatability of the mesh-LBP. This issue can be addressed in different ways:

- Adding a pre-processing stage that regularizes the density of the mesh triangulation.
- Deriving iso-geodesic contours from the ordered rings that act as a support region for computing mesh-LBP operators.
- Applying the local density invariant smoothing proposed by Darom and Keller [31] to the ring vertices around the central facet.

In our experiments, we rather defined a simpler technique that interpolates the scalar function used to compute the mesh-LBP across each ring, so as to obtain a sequence of samples that matches the ideal progression. In the experiments, we found that the mesh-LBP computed using this technique can actually cope to a large extent with mesh irregularities.

3.3. Uniform patterns

In the 2D-LBP, Ojala et al. [2] noticed that the majority of the patterns in textured 2D images have a number of bitwise 0–1 transitions equal at most to 2. These patterns were called “uniform”. Based on this result, in our work we investigated the statistics of mesh-LBP values in order to verify if patterns with a preponderant recurrence can be identified. To this end, we considered a representative set of three surface meshes collected from different sources, each used as representative of different classes of 3D objects. The first surface is a portion of a pot object from the “MIT CSAIL textured 3D models database” [32]. This object exhibits textured shape patterns on the surface. The second surface represents a face scan from the “Binghamton University 3D facial expression database” (BU-3DFE) [33], and shows the case of an open surface, with regions of different 3D textures. The third one is a closed surface of a cat model from the “TOSCA high-resolution database” [34]. These models are shown in Fig. 4(a), (b) and (c), respectively.

Four scalar functions on the mesh manifold have been studied, namely, the *mean curvature* (H), the *Gaussian curvature* (K), the *curvedness* (C), and the *angle between facets normal* (D). For each function, we counted the number of transitions U in the binary patterns computed by using the mesh-LBP operator of Eq. (2) with $\alpha_2(k) = 2^k$, across six levels of spatial resolution (r from 1 to 6), and using 12 samples for the azimuthal quantization ($m = 12$ at each r). The results are depicted in Fig. 5, considering as uniform the patterns of 12 bits with a number of 0–1 transitions less than or equal to four (i.e., $U \leq 4$). It can be observed that the number of uniform patterns exceeds 90% up

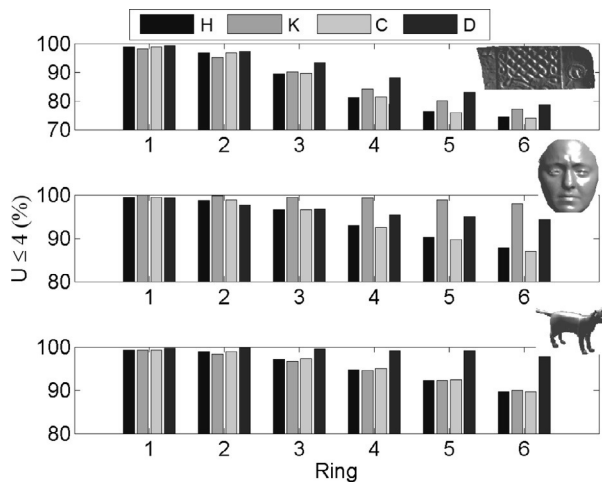


Fig. 5. Percentage of facets whose mesh-LBP computed for the three meshes have a number of transitions U less than or equal to 4 (legend: H —mean curvature; K —Gaussian curvature; C —curvedness; D —angle between facets normals).

to the third ring, across the four scalar functions, for all the three surfaces. The angle between facets normal (D) is the function exhibiting the best uniformity with an overall percentage above 80%. The mean curvature and the curvedness show virtually the same rates. Overall, all the scalar functions show a percentage of uniform patterns above 70%. These observations provide an evidence on the existence of a “uniformity” aspect of the mesh-LBP computed on triangular mesh manifolds. As a consequence, mesh-LBP has the potential of adapting to the uniformity-driven description suggested by Ojala et al. [2].

The uniformity aspect is particularly relevant when the $\alpha_2(k) = 2^k$ weighting function is used for deriving the mesh-LBP patterns. In fact, using 12 samples per ring (i.e., azimuthal quantization $m = 12$), the mesh-LBP takes values in $[0, 4095]$. Using a histogram representation to account for the frequency of the patterns in a local support, this results in a large histogram with 4096 bins. Considering a uniform/non-uniform mesh-LBP partition, the number of histogram bins can be reduced by using distinct bins for uniform patterns, and a single bin for all the non-uniform one. For example, considering as uniform the patterns with a number of transitions equal at most to four ($U \leq 4$), 1123 bins are sufficient to account for the different uniform patterns, while one bin is used for all the remaining patterns (the 2973 non-uniform ones). In this way, a histogram of 1124 bins is sufficient to capture the variability of mesh-LBP. For the $\alpha_1(k) = 1$ weighting function, instead, the number of possible different patterns is 13, so that the uniform/non-uniform distinction is not useful in practice. Based on the above, in the remaining of the paper, if not stated differently, when the mesh-LBP is used in combination with $\alpha_2(k)$, we exploit the uniformity aspect to reduce the dimensionality of the descriptor, hence reducing the space and computational cost of histogram matching, and improve its descriptiveness.

3.4. Repeatability

Repeatability is a fundamental property for 2D or 3D descriptors, measuring the capability of the extracted representation to assume comparable values when computed from corresponding positions in different acquisitions of a same scene or object. According to this, the mesh-LBP repeatability measures the capability of the descriptor to assume comparable values when extracted from corresponding facets of different instances (i.e., scans) of a same 3D object. To prove this property, we acquired 32 facial scans of a same subject with neutral or moderate facial expressions. The four scalar surface functions reported in Section 3.3, namely, *mean curvature*, *Gaussian curvature*, *curvedness* and *angle between facets normal* have been used for

computing mesh-LBP for each facet of the meshes. For each of these functions, we considered two different mesh-LBP operators, that is, $\alpha_1(k) = 1$ and $\alpha_2(k) = 2^k$ (see Eqs. (1) and (2)). In addition, for $\alpha_2(k)$ we considered the variant in which the operator is applied just to the patterns with a *number of transitions* U not greater than 4 (i.e., uniform patterns). Different spatial resolutions corresponding to eight rings $r = 1, \dots, 8$ have been also accounted, whereas the number of samples is kept constant to $m = 12$ at each r . To compute the repeatability of mesh-LBP, we followed an approach similar to that proposed for 3D keypoints by Mian et al. [35]. With this solution, first a scan is selected as *reference*, and each of the other scan (*probe*) is aligned to the reference one using ICP registration. Then, for each facet in the probe, the nearest neighbor facet in the reference is found, whose mesh-LBP value is equal to the mesh-LBP value of the probe facet (the nearest neighbor distance between facets is computed between the 3D coordinates of their centroid). This operation is repeated for each facet in the probe and the distances computed as above are recorded. Varying a proximity radius around the facets it is possible to count the percentage of repeated mesh-LBP values between probe and reference scans for each value of the radius. The overall repeatability is finally obtained by iteratively using one of the scan as reference and all the remaining as probes.

Fig. 6(a)–(c) show the obtained average repeatability as a function of increasing values of the proximity radius, respectively, for the three used mesh-LBP operator functions. The plots reported in the figure concern the mesh-LBP computed on the 1st-ring (i.e., $r = 1$), but a similar behavior resulted for the rings at increasing values of r . In general, we observe that the *Gaussian curvature* and the *angle between facets normal* show a close behavior, obtaining the highest repeatability in all the cases. The *mean curvature* and *curvedness*, instead, score similar results each other, showing a lower performance especially for the α_1 and α_2 operators. Interestingly, for all the scalar surface functions the best repeatability is obtained for the α_2 operator when applied to uniform patterns (case (c) in Fig. 6).

4. Applications

From the formal definition and properties given in the previous sections, it emerges that the mesh-LBP is based upon a notion of local description, which captures differential information of the surface. This makes it not meant to handle general 3D shapes, but rather appropriate for the class of 2D-manifold objects exhibiting geometric texture of the surface or local shape variation. We believe that mesh-LBP is best suited for addressing intra-class classification/retrieval problems, in which objects have a similar global structure and different local shape characteristics, rather than 3D objects retrieval applications where objects exhibit large global shape variability. According to this, mesh-LBP would not be the appropriate tool for describing 3D synthetic objects created by modelling software and characterized by smooth surfaces (like, for example, those included in the Princeton Shape Benchmark dataset [36]), or for evaluating the similarity between 3D shapes characterized by different classes of transformations (example of 3D shapes transformed in different classes by isometry, topology, sampling, scale, holes, noise, are included in the SHREC10 robust correspondence 3D shape benchmark [37]).

Based on these considerations, we investigated the exploitation of the mesh-LBP in three different surface analysis applications, which involve both local and global representations of geometric texture of the surface, namely, 3D texture classification (Section 4.1), 3D texture retrieval (Section 4.2), and 3D face matching (Section 4.3).

4.1. Discriminating 3D texture patterns

In this experiment, we investigated the potential of the mesh-LBP for discriminating texture patterns on 2D mesh-manifolds. We remark here that in this study textures are intended as 3D repeatable

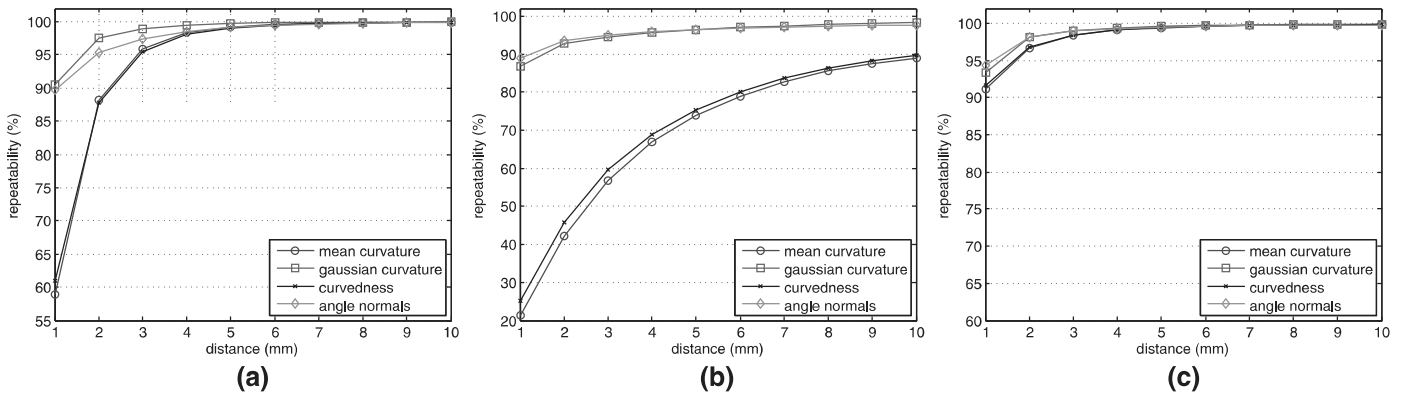


Fig. 6. Repeatability of mesh-LBP: (a) $\alpha_1(k) = 1$; (b) $\alpha_2(k) = 2^k$; (c) α_2 applied to uniform patterns only (i.e., number of transitions $U \leq 4$).

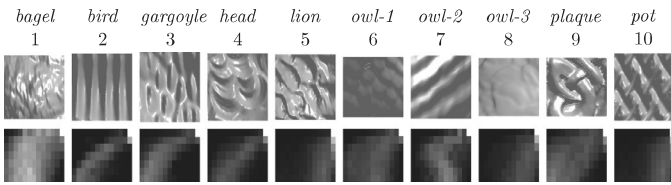


Fig. 7. Top: 3D texture samples from the 10 classes. Bottom: The corresponding histograms obtained with the scalar surface function *angle between facets normal*, and the α_1 operator, using 7 rings and 12 samples per ring (histograms with 7 rows and 13 columns).

patterns which corrugate the surface geometry of a 3D object mesh. This concept is completely different and separated from the 2D texture, which is related to the photometric appearance of the model and, if present, is coded by a 2D image. In so doing, our goal is to probe the capability of mesh-LBP as a framework for 3D texture classification, rather than to elaborate a proper method for such task. As an example, the task of classifying 3D texture patterns is relevant for medical applications, where 3D scans or reconstructions of biological organs and tissues can exhibit different geometric patterns of the surface that can be used to distinguish between diseased and normal response.

In the following, we propose an experimentation on surface samples exhibiting a variety of 3D shape textures, collected from eight different object models of the “MIT CSAIL textured 3D models database” [32]. These objects are *bagel*, *bird*, *gargoyles*, *head*, *lion*, *owl*, *plaque* and *pot*. All these models are characterized by a reasonably uniform mesh, and we were able to identify 10 distinct 3D texture patterns from them, as reported in the first row of Fig. 7 (in particular, three texture patterns were derived from the *owl* object). For each sample, we computed a 1D-histogram of the mesh-LBP (Eq. (2)) using the operator functions $\alpha_1(k) = 1$ and $\alpha_2(k) = 2^k$, a varying spatial resolution $r = 1, \dots, 7$, and an azimuthal quantization $m = 12$. For the α_1 operator, the resulting mesh-LBP take values in $[0,12]$, and these values are accumulated in a 1D histogram with 13 bins for each ring. For the α_2 operator, for which the range of mesh-LBP is $[0,4095]$, we adopted a uniform/non-uniform mesh-LBP partition, that is 1123 bins are used for the different uniform patterns having a number of transitions equal at most to four, and one bin for all the remaining patterns (the 2973 non-uniform ones). Based on this setting, two 2D histograms of size $(7,13)$ and $(7,1124)$ are computed for each texture (i.e., each row of the histogram corresponds to the mesh-LBP computed for a ring), which are associated, respectively, with the α_1 and α_2 operators.

To compute the distance between two histograms H_1 and H_2 , the complement of the Bhattacharyya coefficient $B(\cdot)$, i.e., $\sqrt{1 - B(H_1, H_2)}$ was used. We repeated the histogram computation for each model

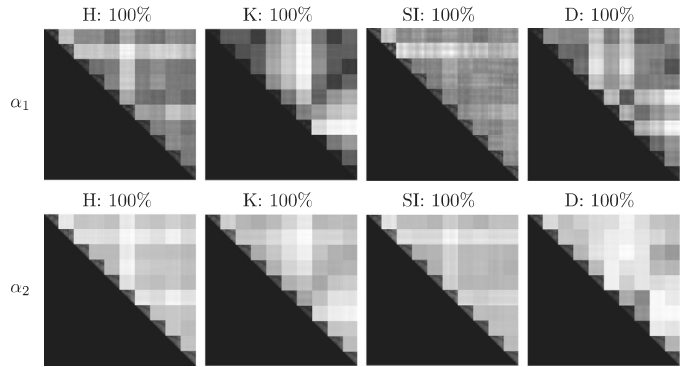


Fig. 8. Matrices reporting the distances between all the instances of the texture classes (30 instances per class). Distances are computed for the mesh-LBP obtained using H , K , SI , and D scalar surface descriptors (top row for α_1 and bottom row for α_2 operator, respectively). The classification accuracy, estimated as the percentage of occurrences where the inter-class distance is greater than the intra-class distance across all the classes is also reported for each descriptor.

using four scalar functions, namely, the *mean curvature* (H), the *Gaussian curvature* (K), the *shape index* (SI) (instead of the *curvedness*) and the *angle between facets normal* (D). As an example, Fig. 7 (second row) depicts histogram instances of the first type (i.e., α_1 operator) obtained with the *angle between facets normal*.

The assessment of the discriminative power of the different descriptors has been performed by considering 30 different instances for each texture class. For each of them, the different descriptors have been computed. Fig. 8 reports the distance matrices between all the classes' instances (i.e., 30 instances for each of the 10 classes) mapped in ascending order to a gray-level scale from black to white. The instances have been arranged and ordered class-wise, that is, the first 30 instances correspond to the first class, the second 30 instances to the second class, etc. Likewise the distance matrix is arranged in 100 blocks, where the diagonal ones correspond to the 30×30 intra-class distance matrix and the others blocks to the 30×30 inter-class distance matrix. Results for the mesh-LBP computed with the scalar functions H , K , SI and D , for the operator α_1 and α_2 are depicted in the top and bottom row of the figure, respectively. In the mesh-LBP distance matrices, we can easily distinguish the diagonal blocks representing the intra-class distances blocks by their dark colors when compared to the other blocks. The classification accuracy, estimated as the percentage of occurrences where the inter-class distance is greater than the intra-class distance across all the classes is also reported for each descriptor on top of the distance matrices in Fig. 8. A perfect classification of 100% is obtained in all the cases. These results confirm the discriminant capability of the mesh-LBP descriptors.

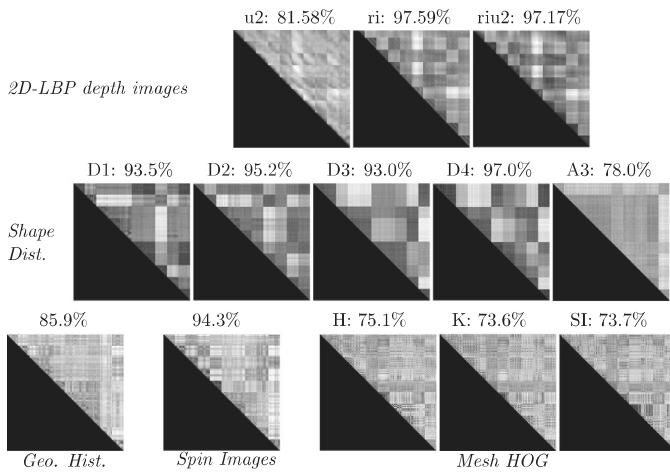


Fig. 9. Matrices reporting the distances between all the instances of the texture classes (30 instances per class). Distances are computed for: three variants of 2D-LBP on depth images (top); shape distribution variants (middle); geometric histogram (bottom left); spin images (bottom center); mesh-HOG (bottom right) computed for different surface scalar functions, namely, H , K and SI . For each descriptor, the overall classification accuracy is also reported in percentage.

4.1.1. Comparative evaluation

We compared the mesh-LBP with other standard 3D surface descriptors including the 2D-LBP on depth images [25], geometric histograms (GH) [38], and the shape distribution variants [39], namely, the distance between a fixed point and one random point on the surface ($D1$), the distance between two random points on the surface ($D2$), the square root of the area of the triangle between three random points on the surface ($D3$), the cube root of the volume of the tetrahedron between four random points on the surface ($D4$), and the angle between three random points on the surface ($A3$). In addition, we also considered the spin-images [40] and the mesh-HOG [41] descriptors.

Using these descriptors, we performed the same experiments discussed above for the mesh-LBP. Experiments with 2D-LBP descriptors on depth images differ by the fact these have been derived from depth images of the textured surfaces. For these experiments, for each texture class, samples were constructed at different rotation angles, varying from 0 to $2\pi/3$, around the surfaces principal orientation, to avoid self-occlusion effects. For each sample, we computed multi-resolution 2D-LBP patterns with nearly the setting than their mesh-LBP counterparts. That is, a radial resolution varying from 1 to 7, and an azimuthal resolution of 8 across all the radii. This analysis has been performed for three 2D-LBP variants, namely, the uniform LBP ($u2$), the rotation invariant LBP (ri) and the uniform rotation-invariant LBP ($riu2$).

The distance matrices between all the classes' instances, related to the aforementioned standard descriptors are reported in Fig. 9. Comparing these distance matrices with those obtained for the mesh-LBP using different descriptors and reported in Fig. 8, it clearly emerges the performance improvement obtained using the mesh-LBP approach.

To best understand the structure of the distance matrix, while appreciating the discriminative superiority of the mesh-LBP, we depicted in Fig. 10 the first row of the distance matrix for each of the mesh-LBP $\langle \alpha_2, D \rangle$ and spin image descriptors.

4.1.2. Robustness to mesh irregularities

Ideally, a mesh is formed entirely by equal-sized triangles (not necessarily equilateral), and 6-valence vertices. As we mentioned previously, though nowadays triangle mesh surfaces acquired by shape digitizers have overall good quality in terms of uniformity, they often contain areas of non-uniform tessellation showing extremum triangles, such as needle or flat triangles, and whereby the assumption of vertex valence of six does not hold. These two aspects make the

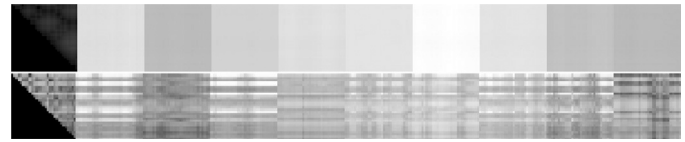


Fig. 10. First bloc-rows of the distance matrices computed for the $\langle \alpha_2, D \rangle$ mesh-LBP descriptor (see Fig. 8) and the spin images descriptor (see Fig. 9). Blocks of the first descriptor are quite homogeneous and distinguishable as compared to the second descriptor. We can also notice, by zooming, the different distances in each 30×30 block in the first block of each row.

arithmetic progression of the number of facets across the rings, expressed by Eq. (4) in Section 3.2 no longer satisfied. We addressed this issue by interpolating or sub-sampling the scalar function on the mesh across the rings. In this experimentation, we assess to what extent this procedure can cope with mesh irregularities that can be encountered in real mesh data. To simulate the two aforementioned aspects that corrupt the mesh uniformity, we propose the corruption procedure reported in Algorithm 4.

Algorithm 4 Triangular mesh corruption procedure.

```

procedure MESHCORRUPT( )
  for  $m = 1 : 8$  do
    for  $n = 1 : 5$  do
      Apply random perturbation to  $m \cdot 10\%$  randomly selected
      vertices
      Collapse  $n \cdot 10\%$  randomly selected edges
    end for
  end for
end procedure

```

The random perturbation consists of applying the following transformation to one of the vertex v of the facet:

$$t(v) = v + \sigma \vec{u}, \quad (5)$$

where σ is a random positive variable taking values in the range $[0.2, 0.8]$, and \vec{u} is a unit vector collinear with the line joining the vertex v to the middle point of its opposite edge. Fig. 11 illustrates the effect of the mesh corruption procedure of Algorithm 4, by reporting examples of the transformation of Eq. (5), used for randomly perturb a vertex position, and the mesh collapse procedure. In particular, the effects of perturbing a vertex, of collapsing together the vertices of an edge, and the combination of a vertex perturbation followed by an edge collapse are shown in Fig. 11(a), (b) and (c), respectively.

Combining the transformation of Eq. (5) and the edge collapsing aims to obtain mesh irregularity instances close to the ones encountered in real mesh data. The extreme case of this corruption scheme is represented by meshes where 80% of the facets and 50% of the edges have undergone vertex perturbation and edge collapsing, respectively. Though real mesh data rarely exhibit such extreme corruption, at least after a basic pre-processing, considering such extreme cases, allows us to best assess the extent to which the adopted interpolation/subsampling procedure can address mesh irregularities. We applied this corrupting procedure to the textured shape surfaces included in the 10 classes employed in the 3D texture matching experiments discussed above. For each texture class, we obtained 40 sets of mesh instances at increasing corruption amplitudes. Referring to Algorithm 4, these sets are obtained by combining the 8 percentages of vertex perturbation (for $m = 1, \dots, 8$), with the 5 percentages of edge collapsing (for $n = 1, \dots, 5$), each combination representing a different level l of corruption ($l = (m - 1) * 5 + n$). In turn, each set contains the 30 instances of the class. Fig. 12 depicts an original mesh surface and four samples of corrupted instances at different levels.

For each mesh corruption level, we performed the full classification procedure involving all the 30 instances of each class. The

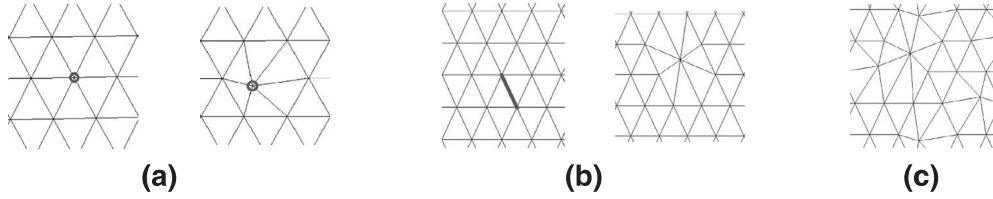


Fig. 11. Mesh corruption procedure. (a) Transformation of amplitude $\sigma = 0.2$ applied to a vertex. (b) Edge collapse. (c) Vertex transformation followed by an edge collapse.

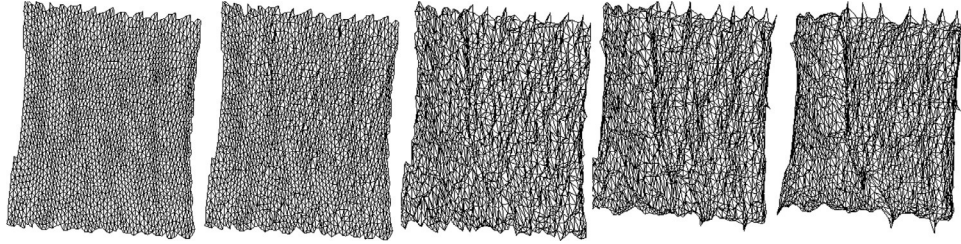


Fig. 12. The original mesh (left) and four corrupted instances at levels 1, 11, 21, and 31 (e.g., according to Algorithm 4, a level of corruption equal to 21, correspond to randomly perturbing 50% of vertices and 10% of edges).

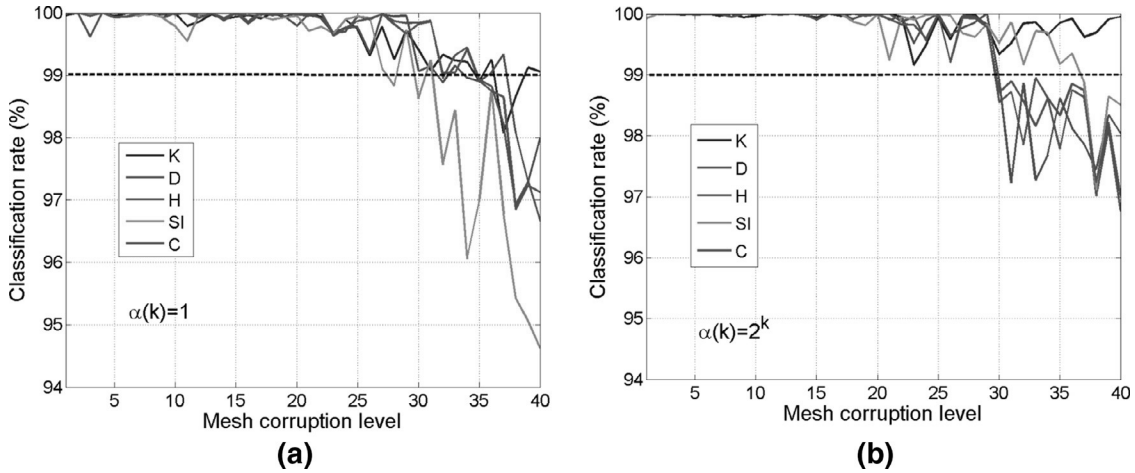


Fig. 13. Classification accuracy obtained for the different mesh corruption levels: (a) $\alpha(k) = 1$; (b) $\alpha(k) = 2^k$.

obtained classification rates are depicted in Fig. 13(a) and (b). It can be observed that all the mesh-LBP descriptors keep a classification accuracy above 99% up to the 30th corruption level, and practically 100% up to the 20th level, especially for the $\alpha(k) = 2^k$ variants (Fig. 13(b)). For this category, we notice in particular that with Gaussian curvature (K), the descriptor keeps above 99% accuracy across all the corruption levels, seconded by the SI, which shows similar performance up to the 37th level. In the first category (Fig. 13(a)), the angle between facets normal is virtually scoring 100% till the 29th level. Overall, the results indicate a clear resistance of the mesh-LBP descriptors to mesh irregularities, and bring evidence of the validity of the proposed interpolation/subsampling procedure.

4.1.3. Global deformation

In this experiment, we studied the behavior of mesh-LBP descriptors when the surface undergoes a global and uniform deformation, that is a deformation that affects the overall shape of the surface, while preserving the 3D texture of the shape.

For this purpose, we applied a series of circular bending at increasing amplitudes to the 10 3D textured surfaces. The bending is performed around the major axis of the surface. Such operation simulates wrapping the textured surface on a cylinder whose axes is aligned with the major axis of the textured surface. The bending



Fig. 14. Original surface (left), and bent samples corresponding to amplitude levels 1, 16 and 22, respectively.

transformation is defined by the following equations:

$$\begin{aligned} x' &= x \cos\left(\frac{x}{2r}\right) \\ y' &= y \cos\left(\frac{y}{2r}\right) \\ z' &= z + x' \sin\left(\frac{x'}{2r}\right) + y' \sin\left(\frac{y'}{2r}\right), \end{aligned} \tag{6}$$

where r is the radius of the cylinder, which defines the bending amplitude. The range of the radius values that can be applied must have a lower bound, which we estimated to 0.8λ , being λ the surface dimension along the major axis. Below this value, the local shape of the tested surfaces is affected, and thus the 3D texture characteristics are no longer preserved. The bending has been applied for 22 decreasing radius values from 5λ (which corresponds to a moderate bending, level 1) down to 0.8λ (level 22). Fig. 14 depicts some samples illustrating surface bending at different amplitudes.

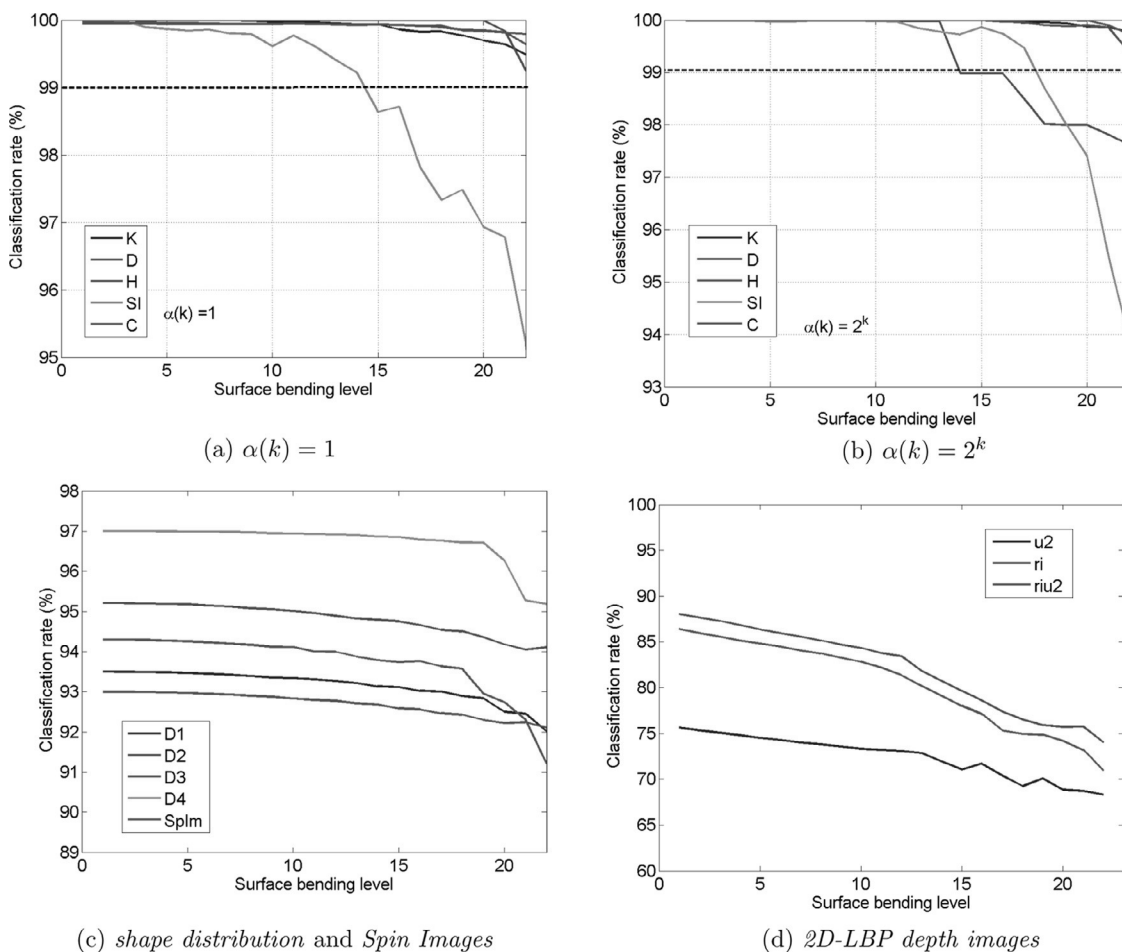


Fig. 15. Classification rates obtained at different bending levels are reported in (a) and (b) for the mesh-LBP with $\alpha(k) = 1$ and $\alpha(k) = 2^k$ operators, respectively; in (c) and (d) for the other shape descriptors (i.e., shape distribution D_1, D_2, D_3, D_4 , and spin images), and 2D-LBP depth images, respectively.

The experimental data have been prepared as follows: for each bending level, we grouped together 30 samples of the bent surface with 30 samples from its original counterpart. Each texture class is thus represented by 60 samples with equal normal and bent instances. We assessed the classification accuracy as in the previous experiments, that is by computing the distance matrices between all the groups of 60 class instances, and deriving from them the percentage of correct matching instances. The results are depicted in Fig. 15. For $\alpha(k) = 1$ (Fig. 15(a)), all the mesh-LBP descriptors, except the SI, keep a classification rate above 99% and practically 100% till the 15th level. A better performance is observed in the results for $\alpha(k) = 2^k$ (Fig. 15(b)), for the Gaussian curvature (K), the mean curvature (H), and the curvedness (C). The difference between normals (D) and the shape index (SI) behave well till the 15th level, then degrade significantly afterwards.

We performed the same experiment with the other standard descriptors that include shape distributions (D_1, D_2, D_3, D_4), the spin image ($Splm$), and 2D-LBP ($u2, ri, riu2$). As we can notice in Fig. 15(c), D_1, D_2, D_3 show some steadiness, whereas the $Splm$ and D_4 performance starts to decrease significantly from the level 18 and 19, respectively. For the 2D-LBP, we first notice a significant drop of the performance from the very first bending level, then the performance decrease nearly at linear rate. Overall, while the mesh-LBP and shape distribution descriptors look coping reasonably with bending, the mesh-LBP keeps its superiority across the different levels. The 2D-LBP descriptor performance, on the opposite, seem quite affected by the bending.

4.2. 3D texture retrieval

In the following, we propose 3D retrieval based on the geometric texture of the surface. We consider the 3D geometric texture as a property of the surface, distinct from the shape, characterized by the presence of repeatable geometric patterns. These patterns can be seen as geometric corrugations of the surface that do not alter the overall 3D shape, but rather change the local smoothness and appearance of the surface. This can result in 3D objects that show similar or equal shape, but very different 3D geometric texture. To the best of our knowledge, the potential of a 3D retrieval based on this surface feature has been not addressed before. This concept can find applications in distinguishing and retrieving 3D objects where the information of interest lies in the geometric texture of the surface, rather than in the shape (an application is reported in Section 4.2.1); or in the identification of textured query patches in large textured surfaces where the shape feature cannot be effectively used, being it vague or even impossible to represent (the terrain models of Section 4.2.2 represent such a case).

4.2.1. Textured objects

This experiment aimed to assess the mesh-LBP potential for detecting specific type of texture in a given surface. Such capacity is useful in “3D shape texture retrieval” applications, where a sample of specific 3D texture (probe texture) is available and we want to automatically detect regions, in a gallery surface, matching that particular model. To the best of our knowledge, we are the first to attempt

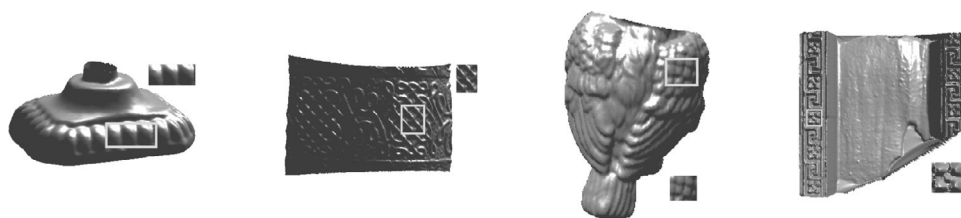


Fig. 16. Surfaces extracted from the bird, pot, owl, and mural models, and their corresponding position, highlighted with a rectangle, in the probe models.

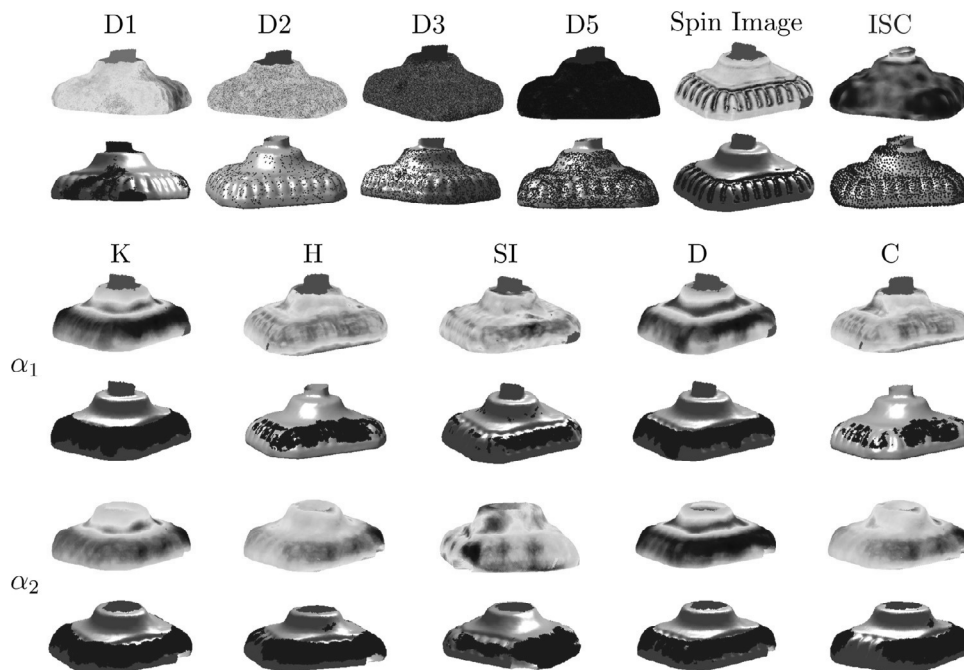


Fig. 17. Results for the *bird* surface. Two rows are reported in each case; the upper row represents the distance map obtained with the Bhattacharyya distance; in the lower row, the region on the mesh where the probe texture is best identified is highlighted in blue. For mesh-LBP the α_1 and α_2 operator are evaluated, in combination with the surface functions K , H , SI , D and C . (For the interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

retrieving shape texture on a mesh manifold. However, we point out that elaborating a full scheme for 3D texture retrieval is beyond the scope of this paper. Rather, we aim to showcase the potential of the mesh-LBP and its performance for such a task in comparison with other standard descriptors. Therefore, we just used a naive template-matching-like method, where the gallery mesh surface is browsed, and at each facet a texture descriptor is computed and compared to its probe texture model counterpart using a given metric (i.e., the Bhattacharyya distance in this application). Facets exhibiting a low distance (i.e., below a certain threshold) are selected as a potential match.

In the experiments, we considered as gallery a representative set of four surfaces (Fig. 16), exhibiting different global and local shape characteristics. These surfaces were extracted from the *bird*, *pot*, *owl*, and the *murail* objects in the MIT CSAIL database [32]. The order of the aforementioned objects reflects an ascending level of 3D texture retrieval complexity. The *bird* instance contains basically two free-from surfaces, one is smooth and the other is textured. The *pot* surface is composed of a single cylindrical surface exhibiting different types of 3D texture patterns. The *owl* surface encompasses different free-from textured surfaces. The set of probe textures for these three models is composed of three instances of textured surface extracted from them, as shown in Fig. 16. The fourth *murail* model is a U-shaped surface, composed of harsh flat bottom surface, and two border textured bands. The texture retrieval is deemed the most complex for this object, because what we want to retrieve here, is not the textured areas, but rather a particular 3D shape pattern in the textured surface, shown in the probe sample in Fig. 16.

The experiment consists in searching each probe within its corresponding surface and then assessing the detection and retrieval capacity of the different descriptors.

In addition to the shape distribution and the spin-image descriptors, we also tested the recently proposed intrinsic shape context (ISC) [42]. Figs. 17–20 show the maps of the Bhattacharyya distance computed at each facet and the related retrieval results for the bird, pot, owl, and murail objects, respectively. Referring to the distance maps (first row), we can assert that the shape distributions practically show no possibility for detecting the searched texture. The spin-image (first row, fifth sample) looks partially spotting the textured regions in the distance maps for the bird and the pot surfaces, whereas its corresponding owl map indicates neat incapacity. For the murail model, none of shape distribution descriptors seems capable of detecting the 3D probe pattern. The spin-image could only achieve a partial retrieval of the textured areas, with some false positives detected at flat surface though. The ISC does not indicate a particular ability for spotting the probe texture apart for the owl model, but with a significant false detection rate. These observations, related to the shape distribution, the spin-image and the ISC maps are confirmed in the texture retrieval results (second row), which indicate a nearly total failure in recovering the searched texture.

The mesh-LBP distance maps, on the opposite, indicate a neat superior performance across the four models, though with different levels. For the α_1 variant, the K and D results clearly indicate an ability of detecting the searched texture for the bird and the pot surfaces, as compared with the H , SI and C . The same is noticed for the

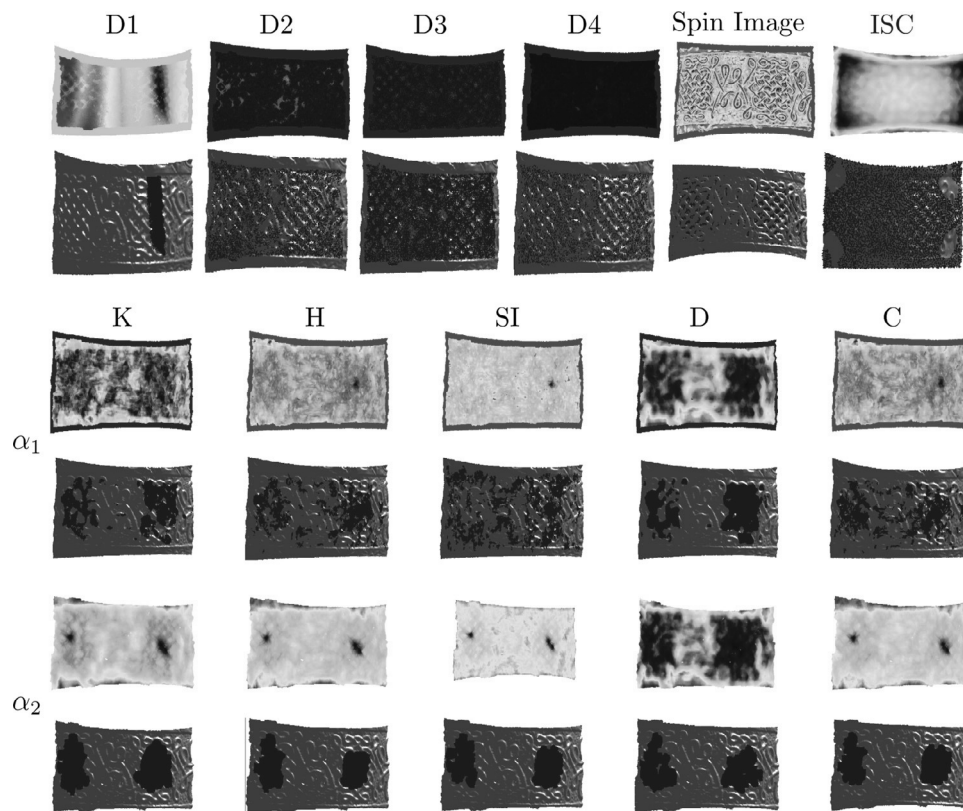


Fig. 18. Results for the *pot* surface.

owl surface. These observations are confirmed by the retrieval results (fourth row), whereby *K* and *D* instances show the best performance across the three surfaces. For the variant using the α_2 operator, the distance maps (fifth row) show an overall improvement, particularly visible in the *pot* and the *owl* results. We can observe that for the three surfaces, the regions in related maps look more compact and localized when compared with their α_1 counterparts. The appearances of these maps suggest an even more ability in texture retrieval, which has been confirmed in the detection results depicted in the last row.

Results of *murail* model experiments depicted in Fig. 20 indicate clearly the ability of the mesh-LBP descriptors for detecting and retrieving the 3D pattern. The *H* and *C* variants exhibit the best performance whereby the eight 3D pattern instances have been successfully detected. We can appreciate this performance by observing their distance-maps showing saliently the 3D texture pattern locations. The distance maps and the retrieval figures, overall, confirm again the superiority of the α_2 operator.

4.2.2. Terrain models

In this experiment, we considered a remote sensing application. Here the retrieval task is described as follows: given a 3D terrain query representing a specific area, find its corresponding match in a gallery of 3D terrain models. We used terrain models from the public “The Visualization Virtual Services databases”.¹ These terrain models are originally digital elevation models (DEM) converted into mesh models. A set of 21 terrain mesh models have been set as gallery models. From these models, we generated three rotated sets at an angle of 45°, 90° and 135°, thus obtaining 63 query models simulating different sensor poses. Also, to simulate the effect of distortion in the mesh model construction from the DEM model, which might result

from the sensor pose change, we applied the rotation on the cloud of points derived from the terrain model after perturbing their positions by small amount, then we construct a new mesh model out of them. Fig. 21 depicts samples of gallery and query terrain models.

In each query model, we selected a sample area to be used as probe. The area is a geodesic disc around a given point, which is approximated, in the terrain mesh model, by the facets confined within a given sphere. The histogram of mesh-LBP descriptors computed at this area is compared with its counterpart computed at each facet of the gallery models, looking for the instance that produces the minimum distance. Table 1 shows the rate of correctly retrieved models with the different mesh-LBP descriptors. We restricted the comparison to the *spin image* descriptor in view of the previous experiment results, in which the *spin image* performed quite above the others. We first notice the neat superiority of the mesh-LBP over the *spin image*. The *K* and *D* performed best at both α_1 and α_2 variants. The overall results confirms also the preeminence of α_2 variants.

4.3. 3D face matching

The mesh-LBP has been also used for constructing a 3D counterpart of the LBP based face description proposed in [9]. In that work, a face representation is constructed, by dividing the 2D face image into regions (e.g., 2D blocks), and histograms of the LBP descriptors are extracted from each region, and then concatenated to form a global description of the face (see Fig. 22(a)). In the case of 3D face scans, the methodology for constructing a mesh-LBP representation of the facial shape is a two-phase process, namely: (1) Extracting a grid of points from the facial surface, and generating regions around these points; (2) computing the histograms of mesh-LBP descriptors for each region, and concatenate them to obtain a global mesh-LBP representation of the face.

As depicted in Fig. 22(b) the extraction of the grid is performed as follows: First, the plane formed by the nose tip and the two eyes

¹ <http://shapes.aimatshape.net/ontologies/shapes/>

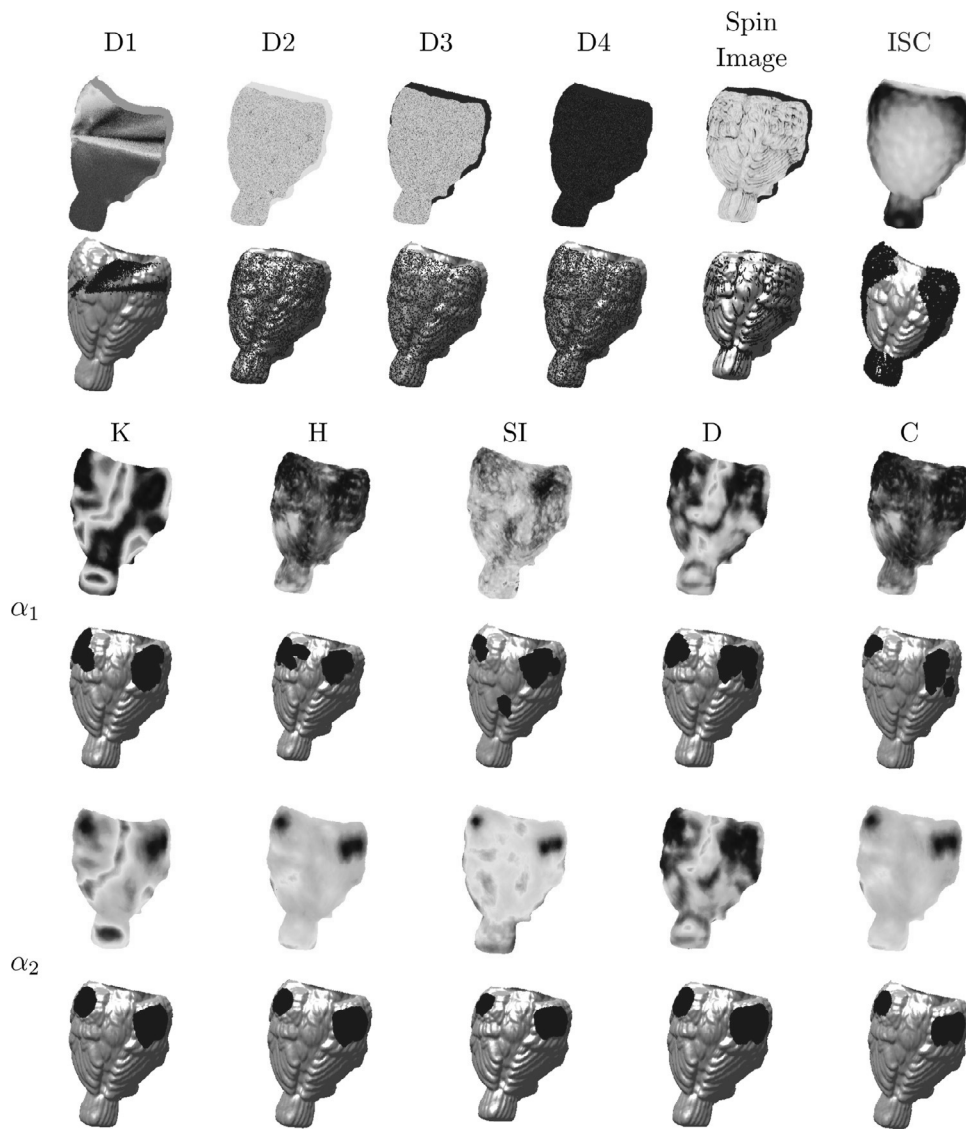


Fig. 19. Results for the owl surface.

inner-corner landmark points is computed. These landmarks have been selected since they are among the most accurate detectable landmarks on the face, in addition to their robustness to facial expressions. From these three landmarks we derive, via simple geometric calculation, an ordered and uniform set of points on that plane. Afterward, we project this set of points on the face surface, along the plane's normal direction. The outcome of this procedure is an ordered grid of facets, which defines an atlas for the facial regions dividing the facial surface. Once the grid of points is defined, a multi-ring structure of s_r rings is generated at each facet of the grid, thus obtaining an ordered sequence of regions covering the facial surface. Then, for each facet in a region we compute a multi-resolution mesh-LBP descriptor using Eq. (2). In the final step, the histograms of these descriptors are computed and concatenated into a single histogram describing the overall face (see Fig. 22(c)).

From the above, the set of parameters of this representation that can be used to optimize face matching encompasses the number and the size of the regions (n_r and s_r , respectively) in addition to the parameters of the multi-resolution mesh-LBP, which include the radial resolution r , the azimuthal quantization m , and the type of mesh-LBP operator (α_1 or α_2), and the local shape descriptor (H , K , SI , D , and C). For the face matching metric, we used different distances that include,

the Bhattacharyya distance, the cosine distance, and the Chi-square distance.

For the experimentation, we used the BU-3DFE database constructed at Binghamton University [33]. The data encompass 56 males and 44 females. The subjects are well distributed across different ethnic groups or racial ancestries, including White, Black, East-Asian, Middle-East Asian, Hispanic-Latino, and others. Face scans are acquired with different facial expressions, including anger, disgust, fear, happiness, sadness, and surprise. Each facial expression has four levels of intensity, respectively, low, middle, high and highest, except the neutral facial expression that has only one intensity level. Thus, there are 25 3D facial expression scans for each subject, resulting in 2500 3D facial expression scans in the database. As an example, Fig. 23 shows the 3D scans of a sample subject showing the neutral expression and the six basic facial expressions at the lowest level.

While the BU-3DFE dataset has been used to investigate the robustness of face recognition algorithms with respect to facial expressions, our main purpose from using it here is to have an assessment of the mesh-LBP potential for global face matching. For that reason, in the experimentation scheme, we considered as gallery and probe the sets of neutral scans and the level-1 expression scans, respectively. We note, however, that level-1 scans actually exhibit significant

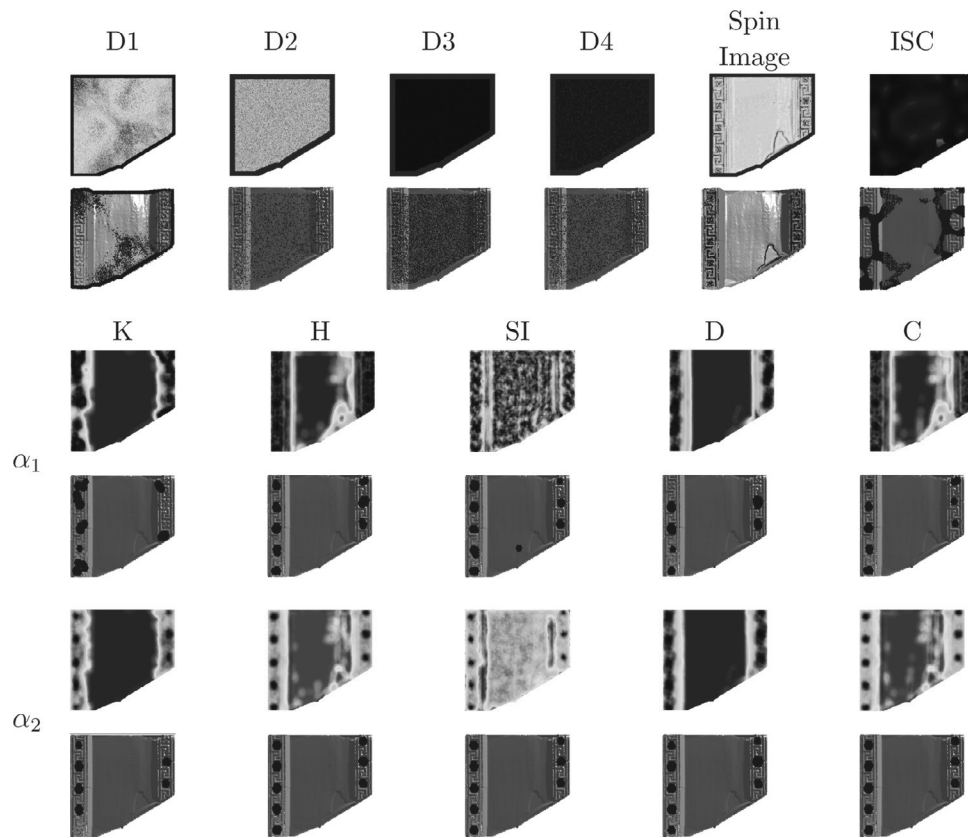


Fig. 20. Results for the *murail* surface.

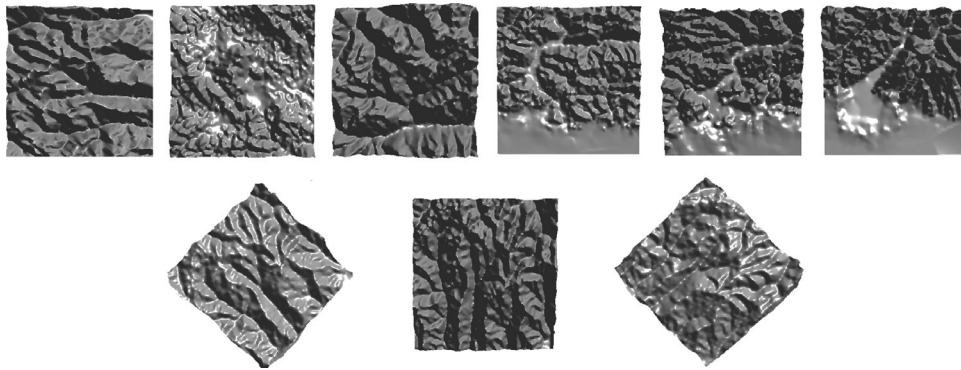


Fig. 21. Samples of gallery terrain models (top). Samples of probe models corresponding to 45°, 90° and 135° rotations (bottom).

Table 1
Accuracy of retrieval for terrain models (values in percentage).

Spin image	α_1					α_2				
	<i>K</i>	<i>H</i>	SI	<i>D</i>	<i>C</i>	<i>K</i>	<i>H</i>	SI	<i>D</i>	<i>C</i>
84.1	96.8	93.7	92.1	98.4	95.2	100	98.4	95.2	100	96.8

disparity from the neutral expression, especially for the disgust, fear, happy and surprise expressions, as it can be noticed in Fig. 23. Using these probe and gallery sets, we performed recognition experiments based on our mesh-LBP descriptors, for a radial resolution r and an azimuthal quantization m set to 7 and 12, respectively. We tested all the combinations of the local shape descriptors H , K , SI, D , and C , and the mesh-LBP operator functions α_1 and α_2 . We set the region size s_r to 7. Also to account for the effects of facial expressions, we segmented the grid points into three bands, dubbed, *top* (T), *middle* (M) and *bottom* (B), then we tested our recognition considering the full grid (TMB) and

the top and middle bands (TM) only (see Fig. 22(d)). The TM option allows us to neutralize to some extent the shape changes manifesting in the lower part of the face, and caused by the mouth in particular. The TMB and the TM grids contain 35 and 26 points, respectively.

Table 2 reports the best obtained rank-1 recognition accuracies. We found that the mean curvature (H) and the curvedness (C) perform the best across the different combinations. From the table, we can clearly notice the best results obtained with the TM grid in comparison with the TMB counterparts, confirming the effect on the shape of the face caused by the mouth (e.g., mouth open). Looking to the mesh-LBP

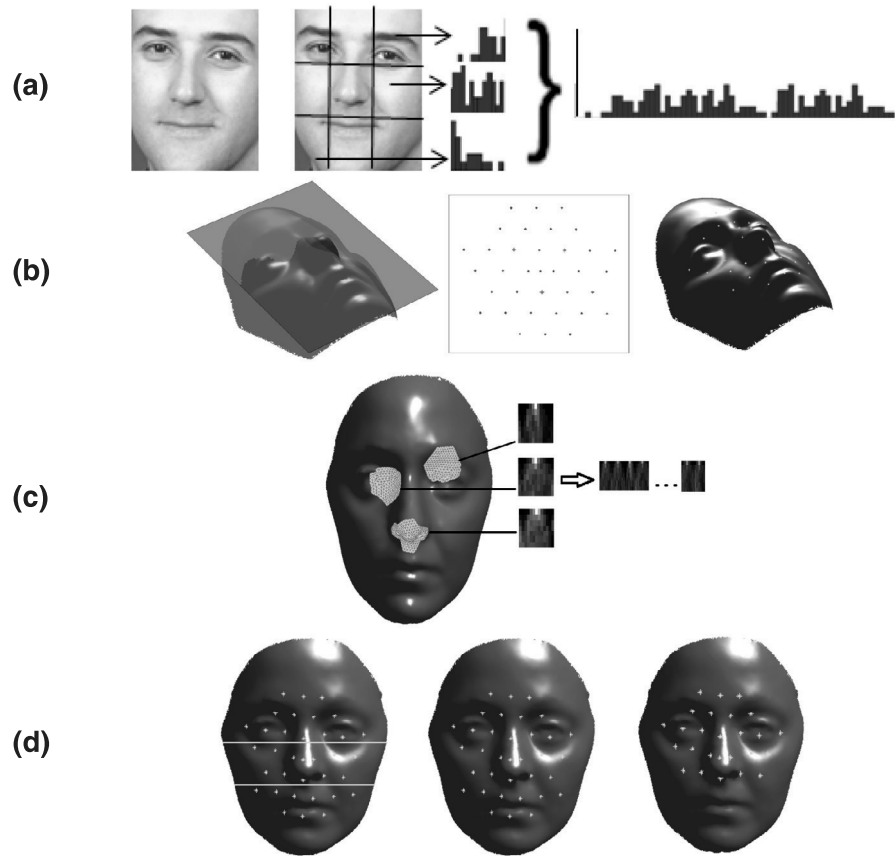


Fig. 22. Face representation. (a) 2D LBP-based face representation. (b) Construction of the face grid. (c) Region samples and their corresponding histograms, which are composed together into a global histogram. (d) On the left scan, the separation of the grid points into a top, middle and bottom band is reported, whereas in the center and right scans all the points in the three bands, and only the points in the top and middle bands are shown, respectively.

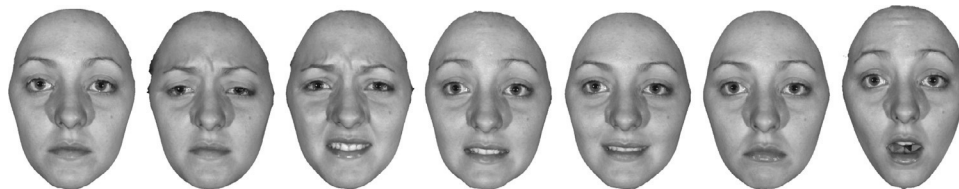


Fig. 23. BU-3DFE-3D face scans (with texture) of a sample subject showing, from left, the neutral, anger, disgust, fear, happy, sad, and surprise expressions. The level-1 of expression intensity is shown in each case.

Table 2

BU-3DFE - Rank-1 recognition accuracy obtained with different operators, surface functions, distances, and points grid.

		TMB			TM		
		Bhatt	cos	χ^2	Bhatt	cos	χ^2
α_1	H	85.74%	90.61%	88.00%	87.48%	92.52%	89.57%
	C	84.47%	89.57%	87.13%	88.00%	90.43%	89.04%
α_2	H	92.00%	82.96%	92.70%	93.91%	87.83%	95.48%
	C	93.04%	85.22%	93.57%	94.61%	89.39%	95.13%

operator functions, we can notice a net superiority of the α_2 operator, for which the top overall performance is obtained in combination with the chi-square distance. The cosine distance instead, seems better coupled with the α_1 operator function.

Table 3 shows on the left-most columns the results obtained on the same database using the three variants of interest-points method recently proposed by [43]. We can see easily that our method performs better. We note, however, that the sets used in their experiments

Table 3

BU-3DFE - Rank-1 recognition rate of our method for the six expressions, obtained with H and C descriptors for each of the local binary operator functions α_1 and α_2 and using TM bands grid. The results of the interest-points method [43], using the HOG, SHOT and GH descriptors are also reported. The best obtained rate in each expression is marked in Bold.

	[43]			α_1		α_2	
	HOG	SHOT	GH	H	C	H	C
Angry	90.0%	93.8%	90.6%	95.88%	91.75%	96.07%	96.91%
Disgust	87.5%	81.6%	85.0%	83.33%	79.17%	87.50%	88.54%
Fear	88.8%	91.9%	84.4%	87.62%	85.57%	93.91%	93.81%
Happy	88.1%	90.0%	85.6%	92.86%	92.86%	95.92%	94.90%
Sad	90.6%	94.4%	90.6%	96.94%	95.92%	98.98%	96.94%
Surprise	85.0%	88.8%	82.5%	98.88%	97.75%	100.00%	100.00%
Overall	88.3%	91.6%	86.5%	92.52%	91.43%	95.48%	95.13%

include level-1 and level-2 expressions. Yet, the small difference between these two levels, allows us to assert the superiority of the proposed approach. In the same table, the recognition accuracies obtained separately, with the TM grid, for the six expressions, together

with the counterpart results in [43] are depicted. Again, we can observe that our method performs best across all the expressions. We notice in particular the high performance obtained in the sad and surprise expressions for the α_2 operator function.

5. Discussion and conclusions

In this paper, we presented mesh-LBP as a novel framework for computing local binary patterns on triangular mesh manifolds. This framework keeps the simplicity and the elegance characterizing the original LBP and allows the extension of all its variants, developed in 2D image analysis, to the mesh manifold. The mesh-LBP relieves object surface data from normalization and registration procedure required when using depth images, while it extends the spectrum of LBP analysis to closed surfaces.

The experiments performed on mesh surfaces collected from different sources, brought evidence of a mesh-LBP “uniformity” aspect for the different types of scalar functions, pretty similar to the one noticed in 2D-LBP. We also demonstrated that mesh-LBP descriptors can cope with mesh-irregularities to a great extent. We showcased the ability of the mesh-LBP to be deployed in both local and global shape analysis. The experiments carried on 3D texture classification showed clear evidence of the appropriateness of the mesh-LBP descriptors for such a task, and their superior discriminative power as compared to other competitive shape descriptors. In summary, the angle between facets normal and the Gaussian curvature seem to be the more effective surface descriptors to be used within the mesh-LBP framework as emerges throughout all the performed experiments. The mesh-LBP performance superiority of 3D texture classification is nearly preserved for moderate and uniform global deformation apart for the shape index variant. While qualitative, and despite employing basic techniques, the experiments on 3D texture retrieval revealed their great potential of the mesh-LBP descriptors and the incapacity of the standard descriptors for such a task.

The mesh-LBP allows a straightforward extension of the 2D-LBP face matching paradigm to the facial mesh surface modality. We proposed a simple yet effective method for constructing global facial shape signature. Compared to its depth image counterpart, our framework relieves the process from the constraining data normalization and depth map extraction tasks, in addition of avoiding self-occlusion effect. We proposed presentation composed of a three-band grid of facial points to which are attached mesh-LBP region descriptors. The experiments showed a net improvement when using the top and middle bottom only as the bottom band is the most sensitive to facial expressions. On the other hand, considering the lower part of the face as missing data, the obtained results revealed the ability of our method to successfully handle partial face matching to a great extent, even in the presence of facial expressions. Overall the obtained results are quite promising and showed to outperform other competitive methods.

The comparison of the α_1 and α_2 operator functions across the different mesh-LBP applications results, gives more credentials the second function, especially in 3D texture retrieval and face recognition. However, the compactness of the descriptor obtained with the α_1 operator, and the resulting lower computational complexity required to compare descriptors, vote for this solution when time constraints are relevant.

In conclusion, we believe that our proposed mesh-LBP opened-up new perspectives in mesh manifold analysis, as showcased by the newly investigated tasks, namely 3D texture classification and retrieval, that have not been approached before, to the best of our knowledge, at least using mesh manifold modality. The texture retrieval experiments evidence the ability of the mesh-LBP not only for retrieving textured surfaces, but also detecting specific 3D shape patterns. As a final remark, we want also to point out our framework is not meant to handle general 3D shapes. Being based upon a notion of

local description, our mesh-LBP framework is rather appropriate for the class of manifold objects exhibiting texture or shape variation. So, it is more related to intra-class classification/retrieval, where objects in that class have a very similar global structure and different local shape characteristics.

As future work, we plan undertaking a deeper investigation of the different applications approached in this work. In addition, to exploring further the different mesh-LBP variants and their appropriate and optimal usage.

Acknowledgments

Preliminary ideas and experiments of this research appeared in [44]. Part of this work is supported by a National Research Foundation Grant UIRCA 2013-24877, and C4 Advanced Solutions, UAE.

References

- [1] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distribution, *Pattern Recogn.* 29 (1) (1996) 51–59.
- [2] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (7) (2002) 971–987.
- [3] T. Mäenpää, J. Viertola, M. Pietikäinen, Optimizing color and texture features for real-time visual inspection, *Pattern Anal. Appl.* 6 (3) (2003) 169–175.
- [4] L. Cao, J. Luo, F. Liang, T. Huang, Heterogeneous feature machines for visual recognition, in: *Proceedings of the International Conference on Computer Vision*, Kyoto, Japan, 2009, pp. 1095–1102.
- [5] D. Guo, V. Atluri, N. Adam, Texture-based remote-sensing image segmentation, in: *Proceedings of the International Conference on Multimedia and Expo*, Amsterdam, The Netherlands, 2005, pp. 1472–1475.
- [6] C. Song, F. Yang, P. Li, Rotation invariant texture measured by local binary pattern for remote sensing image classification, in: *Proceedings of the IEEE International Workshop on Education Technology and Computer Science*, Vol. 3, Wuhan, Hubei, China, 2010, pp. 3–6.
- [7] A. Lucieer, A. Stein, P. Fisher, Multivariate texture-based segmentation of remotely sensed imagery for extraction of objects and their uncertainty, *Int. J. Remote Sens.* 26 (14) (2005) 2917–2936.
- [8] T. Ahonen, A. Hadid, M. Pietikäinen, Face recognition with local binary patterns, in: *Proceedings of the European Conference on Computer Vision*, Prague, Czech Republic, 2004, pp. 469–481.
- [9] T. Ahonen, A. Hadid, M. Pietikäinen, Face description with local binary patterns: Application to face recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (12) (2006) 2037–2041.
- [10] L. Wolf, T. Hassner, Y. Taigman, Descriptor based methods in the wild, in: *Proceedings of the ECCV Workshop on Faces in Real-Life Images*, Marseille, France, 2008, pp. 1–14.
- [11] G. Zhang, X. Huang, S. Li, Y. Wang, X. Wu, Boosting local binary pattern (lbp)-based face recognition, in: *Proceedings of the International Workshop on Advances in Biometric Person Authentication*, Beijing, China, 2005, pp. 179–186.
- [12] C. Shan, S. Gong, P.W. McOwan, Facial expression recognition based on local binary patterns: A comprehensive study, *Image Vis. Comput.* 27 (6) (2009) 803–816.
- [13] L. Cai, C. Ge, Y. Zhao, X. Yang, Fast tracking of object contour based on color and texture, *Int. J. Pattern Recog. Artif. Intell.* 23 (7) (2009) 1421–1438.
- [14] X. Wang, M. Mirmehdi, Archive film restoration based on spatiotemporal random walks, in: *Proceedings of the European Conference on Computer Vision*, Crete, Greece, 2010, pp. 478–491.
- [15] G. Sandbach, S. Zafeiriou, M. Pantic, Local normal binary patterns for 3D facial action unit detection, in: *Proceedings of the IEEE International Conference on Image Processing*, Orlando, FL, 2012, pp. 1813–1816.
- [16] H. Li, L. Chen, D. Huang, Y. Wang, J. Morvan, 3d facial expression recognition via multiple kernel learning of multi-scale local normal patterns, in: *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2012, pp. 2577–2580.
- [17] G. Sandbach, S. Zafeiriou, M. Pantic, Binary pattern analysis for 3d facial action unit detection, in: *Proceedings of the British Machine Vision Conference (BMVC)*, Guildford, UK, 2012, pp. 1–12.
- [18] S. Liao, A. Chung, Face recognition by using elongated local binary patterns with average maximum distance gradient magnitude, in: *Proceedings of the Asian Conference on Computer Vision*, Tokyo, Japan, 2007, pp. 672–679.
- [19] L. Zhang, R. Chu, S. Xiang, S. Li, Face detection based on multi-block LBP representation, in: *Proceedings of the International Conference on Biometrics*, Washington, DC, 2007, pp. 11–18.
- [20] H. Jin, Q. Liu, H. Lu, X. Tong, Face detection using improved LBP under bayesian framework, in: *Proceedings of the International Conference on Image Graphics*, Hong-Kong, 2004, pp. 306–309.
- [21] D. Huang, Y. Wang, Y. Wang, A robust method for near infrared face recognition based on extended local binary pattern, in: *Proceedings of the International Symposium on Visual Computing*, Lake Tahoe, CA/NV, 2007, pp. 437–446.

- [22] X. Tan, B. Triggs, Enhanced local texture feature sets for face recognition under difficult lighting conditions, in: Proceedings of the International Workshop on Analysis and Modelling of Faces and Gestures, Rio de Janeiro, Brazil, 2007, pp. 168–182.
- [23] T. Ahonen, M. Pietikäinen, Soft histograms for local binary patterns, in: Proceedings of the Finnish Signal Processing Symposium, Oulu, Finland, 2007, pp. 1–4.
- [24] M. Pietikäinen, A. Hadid, G. Zhao, T. Ahonen, *Computer Vision using Local Binary Patterns*, Springer, Berlin, 2011.
- [25] S. Li, C. Zhao, M. Ao, Z. Lei, Learning to fuse 3D+2D based face recognition at both feature and decision levels, in: Proceedings of the International Workshop on Analysis and Modeling of Faces and Gestures, Beijing, China, 2005, pp. 44–54.
- [26] Y. Huang, Y. Wang, T. Tan, Combining statistics of geometrical and correlative features for 3D face recognition, in: Proceedings of the British Machine Vision Conference, Edinburgh, UK, 2006, pp. 879–888.
- [27] D. Huang, M. Ardabilian, Y. Wang, L. Chen, 3-D face recognition using eLBP-based facial description and local feature hybrid matching, *IEEE Trans. Inform. Foren. Secur.* 7 (5) (2012) 1551–1565.
- [28] J. Fehr, H. Burkhardt, 3D rotation invariant local binary patterns, in: Proceedings of the International Conference on Pattern Recognition, Tampa, Florida, 2008, pp. 1–4.
- [29] N. Werghi, S. Berretti, A. Del Bimbo, The mesh-lbp: a framework for extracting local binary patterns from discrete manifolds, *IEEE Trans. Image Process.* 24 (1) (2015) 220–235.
- [30] M. Pietikäinen, T. Ojala, Z. Xu, Rotation-invariant texture classification using feature distributions, *Pattern Recogn.* 33 (1) (2000) 727–739.
- [31] T. Darom, Y. Keller, Scale-invariant features for 3-D mesh models, *IEEE Trans. Image Process.* 21 (5) (2012) 2758–2769.
- [32] MIT CSAIL database, 2008. URL <http://people.csail.mit.edu/tmertens/texttransfer/data/>
- [33] L. Yin, X. Wei, Y. Sun, J. Wang, M. Rosato, A 3D facial expression database for facial behavior research, in: Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, Southampton, UK, 2006, pp. 211–216.
- [34] A. Bronstein, M. Bronstein, R. Kimmel, *Numerical Geometry of Non-Rigid Shapes*, Springer-Verlag, New-York, 2008.
- [35] A.S. Mian, M. Bennamoun, R. Owens, Keypoint detection and local feature matching for textured 3D face recognition, *Int. J. Comput. Vis.* 79 (1) (2008) 1–12.
- [36] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, The princeton shape benchmark, in: Proceedings of the Shape Modeling International (SMI'04), IEEE Computer Society, Washington, DC, USA, 2004, pp. 167–178.
- [37] A.M. Bronstein, M.M. Bronstein, U. Castellani, A. Dubrovina, L.J. Guibas, R.P. Horaud, R. Kimmel, D. Knossow, E. von Lavante, D. Mateus, M. Ovsjanikov, A. Sharma, SHREC 2010: robust correspondence benchmark, in: Proceedings of the EUROGRAPHICS Workshop on 3D Object Retrieval (3DOR), Norrkping, Sweden, 2010.
- [38] A. Ashbrook, R. Fisher, C. Robertson, N. Werghi, Finding surface correspondance for object recognition and registration using pairwise geometric histograms, in: Proceedings of the European Conference on Computer Vision, Friburg, Germany, 1998, pp. 674–686.
- [39] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Shape distributions, *ACM Trans. Graph.* 21 (4) (2002) 807–832.
- [40] A.E. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3D scenes, *IEEE Trans. Pattern Anal. Mach. Intell.* 21 (5) (1999) 433–449.
- [41] A. Zaharescu, E. Boyer, K. Varanasi, R. Horaud, Surface feature detection and description with applications to mesh matching, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, 2009, pp. 373–380.
- [42] I. Kokkinos, M. Bronstein, R. Littman, A. Bronstein, Intrinsic shape context descriptors for deformable shapes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, Rhode Island, USA, 2012, pp. 159–166.
- [43] S. Berretti, N. Werghi, A. Del Bimbo, P. Pala, Matching 3D face scans using interest points and local histogram descriptors, *Comput. Graph.* 37 (5) (2013) 509–525.
- [44] N. Werghi, S. Berretti, A. Del Bimbo, P. Pala, The mesh-lbp: computing local binary patterns on discrete manifolds, in: Proceedings of the International Workshop on 3D Representation and Recognition, Sydney, Australia, 2013, pp. 562–569.