

# The Mesh-LBP: A Framework for Extracting Local Binary Patterns From Discrete Manifolds

Naoufel Werghi, *Member, IEEE*, Stefano Berretti, *Member, IEEE*, and Alberto del Bimbo, *Member, IEEE*

**Abstract**—In this paper, we present a novel and original framework, which we dubbed mesh-local binary pattern (LBP), for computing local binary-like-patterns on a triangular-mesh manifold. This framework can be adapted to all the LBP variants employed in 2D image analysis. As such, it allows extending the related techniques to mesh surfaces. After describing the foundations, the construction and the main features of the mesh-LBP, we derive its possible variants and show how they can extend most of the 2D-LBP variants to the mesh manifold. In the experiments, we give evidence of the presence of the uniformity aspect in the mesh-LBP, similar to the one noticed in the 2D-LBP. We also report repeatability experiments that confirm, in particular, the rotation-invariance of mesh-LBP descriptors. Furthermore, we analyze the potential of mesh-LBP for the task of 3D texture classification of triangular-mesh surfaces collected from public data sets. Comparison with state-of-the-art surface descriptors, as well as with 2D-LBP counterparts applied on depth images, also evidences the effectiveness of the proposed framework. Finally, we illustrate the robustness of the mesh-LBP with respect to the class of mesh irregularity typical to 3D surface-digitizer scans.

**Index Terms**—Local binary patterns, ordered ring facets, mesh manifold, 3D texture analysis.

## I. INTRODUCTION

THE Local Binary Pattern (LBP) is a local shape descriptor that has been introduced by Ojala et al. [1], [2] for describing 2D textures in still images. Its computational simplicity and discriminative power attracted the attention of the image processing and pattern recognition community, and rapidly it has found other applications in visual inspection [3], [4], remote sensing [5]–[7], face recognition [8]–[11], facial expression recognition [12], and motion analysis [13], [14]. However, all 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 features to depth values, and then use ordinary LBP processing on depth images [15]–[17].

Manuscript received January 30, 2014; revised June 18, 2014; accepted November 6, 2014. Date of publication November 12, 2014; date of current version December 12, 2014. This work was supported by the National Research Foundation, United Arab Emirates, under Grant ref: UIRCA 2013-24877. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nikolaos V. Boulgouris.

N. Werghi is with the Department of Electrical and Computer Engineering, Khalifa University, Abu Dhabi 127788, United Arab Emirates (e-mail: naoufel.werghi@kustar.ac.ae).

S. Berretti and A. del Bimbo are with the Department of Information Engineering, University of Florence, Florence 50121, Italy (e-mail: stefano.berretti@unifi.it; alberto.delbimbo@unifi.it).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2014.2370253

The triangular mesh manifold is a simple, compact and flexible format for encoding 3D shape information, which 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. Despite the abundance and the richness of the mesh manifold modality, to the best of our knowledge, there is no a computational support that allows the computation of LBP on this format. One factor that plagued the development of an LBP-based description on the mesh is the lack of an intrinsic order in the triangular mesh manifold, since the LBP requires an ordered support for its computation. On the contrary, computation of LBP on 2D images benefits from the implicit ordering of the pixels in the 2D image array.

Providing such a framework for computing LBP on a mesh could be of great interest for describing 3D texture reflecting the presence of repeatable geometric patterns on the mesh surface (this being a completely separate concept from photometric texture). In fact, there are many applications that require local surface shape analysis and interpretation of 3D textured surfaces. In quality control, texture description can be used for detecting local surface pattern defection. In medicine, most of the imaging data (e.g., ultrasound, microscopic images) are shifting to a 3D mesh format. Many diagnostic rules related to these modalities need description and classification of some organs local surfaces. More generally, texture description on the mesh is useful for any application that needs 3D texture analysis, classification, and retrieval. For example, a typical scenario in the last application is to have a sample of specific 3D texture pattern and detect regions which match that model in a gallery of surfaces.

Motivated by these facts, in this paper we address the challenge of computing LBP on a mesh manifold by proposing an original computational framework, which we called mesh-LBP that 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, then the main features and the contribution of our proposal are discussed.

### A. LBP Overview and Related Work

In its original definition, the LBP operator [1] assigns labels to image pixels by first thresholding the  $3 \times 3$  neighborhood

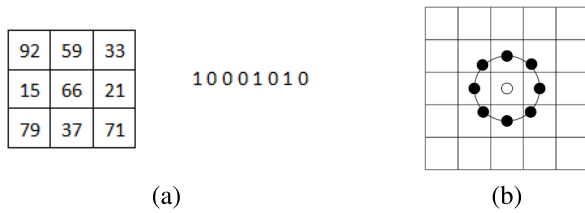


Fig. 1. (a) Computation of the basic LBP code from the  $3 \times 3$  neighborhood of a central pixel. Each pixel, starting from the upper-left corner is compared with the central pixel to produce 1 if its value is greater or equal, 0 otherwise. The result is an 8-bit binary code; (b) Example of a central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

of each pixel with the center value (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 considering the sequence of 0/1 in the pixel neighborhood as a binary number according to a positional coding convention. 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 its neighboring pixels. The histogram of the numbers obtained in such a way can then be used as a texture descriptor. This operator is distinguished by its simplicity and its invariance to monotonic gray-level transformations.

An extended LBP version that can operate on circular neighborhood of different radii, also allowing sub-pixel alterations was proposed later in [2] (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 *et al.* [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 (ELPB) [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 *et al.* [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 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, and 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 an 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 is addressed with complex techniques involving spherical correlation in the frequency domain.

## B. Paper Contribution and Organization

From the analysis above, it emerges that since its introduction 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 done 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 maps.

In this paper, we propose a framework that we call mesh-LBP, for designing and extracting local binary patterns directly from a 2D 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* – By its ability of handling mesh data, this framework can deal with a larger spectrum of surfaces (e.g., closed, open, self-occluded) as compared to its counterpart defined on depth images;
- *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 parametriza-

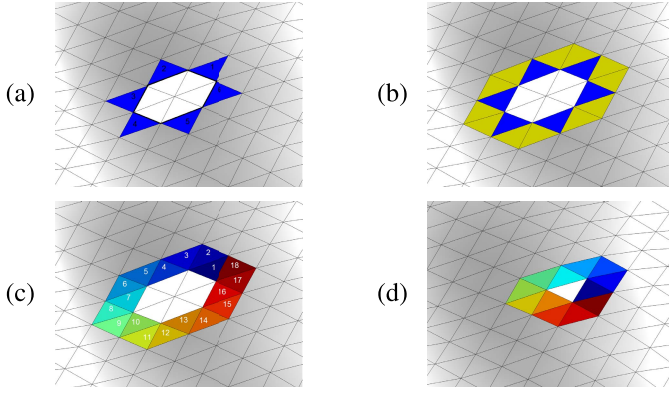


Fig. 2. Construction of an ordered ring: (a) Initial  $Fout$  facets on a convex contour; (b) Bridging the gap between the pairs of consecutive  $Fout$  facets with the  $Fgap$  facets; (c) The obtained ordered ring; (d) Ordered ring constructed around a central facet.

tion, apart the standard mesh arrangement into facets and vertex arrays, while keeping linear computational complexity.

The rest of the paper is organized as follows: In Section II, we introduce our framework by giving the foundation of the mesh-LBP and presenting its multi-resolution extension; Some mesh-LBP variants aiming to reduce the dimensionality of the descriptor are introduced in Section III (a comprehensive view of the mesh-LBP variants is provided in the Appendix), together with solutions addressing the invariance of the descriptor, and its robustness to irregular tessellations of the mesh; Experimental evidence of the potential of the mesh-LBP in different application scenarios and in comparison to state of the art solutions is reported in Section IV; Finally, concluding remarks and future research directions are drawn in Section V.

## II. THE MESH-LBP

The construction of LBP-like patterns on a mesh, first requires a scheme for constructing rings of facets around a central one 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 sets of vertices and facets of the mesh. Let us start by considering the general case of a convex contour on the mesh, which we assume regular, i.e., each vertex has a valence of six (we will show later that our framework can also cope with meshes that do not comply with this ideal case). Consider the facets that have an edge on that contour (Fig. 2(a)). We call these facets  $Fout$  facets, as they seem pointing outside the contour. Let us consider also the set of facets that are one-to-one adjacent to the  $Fout$  facets and which are located inside the convex contour. Each facet in this set, that we call  $Fin$ , shares with its corresponding  $Fout$  facet an edge located on the convex contour. Let us assume that the  $Fout$  facets are initially ordered in a circular fashion across the contour. 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 contour). We call these facets  $Fgap$  facets (see Fig. 2(b)). The “Bridge” procedure reported in pseudocode in Algorithm 1 is

---

### 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  $\langle fout_i, fout_{i+1} \rangle$ 
   $gf \leftarrow$  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

```

---



---

### Algorithm 2 GetRing

---

**Input:**  $Fout$ , set of  $n$  ordered facets,  $fout_1, fout_2, \dots, fout_n$ , 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:**  $Ring$ , ring of ordered facets

```

procedure GETRING( $Fout, Fin$ )
   $Ring = [ ]$ 
  for all  $\langle fout_i, fout_{i \% n + 1} \rangle, i \leftarrow 1, \dots, n$  do
    append  $fout_i$  to  $Ring$ 
     $Fgap_i \leftarrow$  BRIDGE( $fout_i, fout_{i \% n + 1}, fin_i$ )
    append  $Fgap_i$  to  $Ring$ 
  end for
  return  $Ring$ 
end procedure

```

---

used 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 Fig. 2(c)). The resulting arrangement of the ring facets inherits the same direction (clock-wise 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. We dubbed such obtained ordered ring, Ordered Ring Facets (ORF).

In the above discussion, we referred to the general case where the ORF is constructed around a convex contour. Actually, the usual case is constituted by an initial seed formed by an individual facet (central facet), whose three edges represent the initial convex contour. This case is considered in this work, since it corresponds to the situation where an ordered ring is constructed around the facets of a mesh surface. In this particular case, the  $Fout$  set includes the three facets adjacent to the central one, and the obtained ring is composed of 12 ordered facets (i.e., the three  $Fout$  facets, plus

the nine *Fgap* facets bridging the gap between consecutive *Fout* facets), as shown in Fig. 2(d).

Let  $h(f) : S \rightarrow \mathcal{R}$  be a scalar function defined on the mesh  $S$  (e.g., photometric data or curvature). The circular ordering of the facets obtained with ORF allows us to derive a binary pattern (i.e., sequence of 0 and 1 digits) from it, and thus to compute a local binary operator in the same way as in the standard LBP. We define the basic mesh-LBP operator at a central facet  $f_c$  by thresholding its ordered ring neighbourhood constituted by the 12 facets in the ORF:

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

where  $\alpha(k)$  is a weighting function. Different definitions of the function  $\alpha(k)$  permit us to obtain different binary patterns, and thus different mesh-LBP values can be derived from the central facet and its ring neighborhood. For example, with  $\alpha(k) = 2^k$  the basic LBP operator firstly suggested by Ojala *et al.* [1] is obtained; for  $\alpha(k) = 1$ , the sum of the digits of the pattern is computed (i.e., the number of digits equal to 1). We remark here that for the present discussion it is not necessary to detail the particular scalar function  $h(f)$ , whose values are computed on the mesh facets. The effect of different choices of this function will be investigated in Section IV.

#### A. Multi-Resolution Mesh-LBP

The mesh-LBP is extended to a multi-resolution framework by deriving a sequence of concentric rings, which preserve the ordering property. 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 (Fig. 3(b)), a new ring, which exhibits the same ordered structure of its predecessor is obtained (Fig. 3(c)). 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(d)). 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*).

It is worth mentioning that, when the regularity assumption for the mesh is satisfied, the number of facets  $\nu$  across the rings evolves according to the following arithmetic progression from ring  $i$  to ring  $i + 1$ :

$$\nu_{i+1} = \nu_i + 12. \quad (2)$$

This can be intuitively seen referring to Fig. 3: the 1st-ring comprises 12 facets (3 *Fout* plus 9 *Fgap*); 24 facets are included in the 2nd ring (i.e., 9 *Fout* plus 15 *Fgap*); 36 facets in the third ring, and so on.

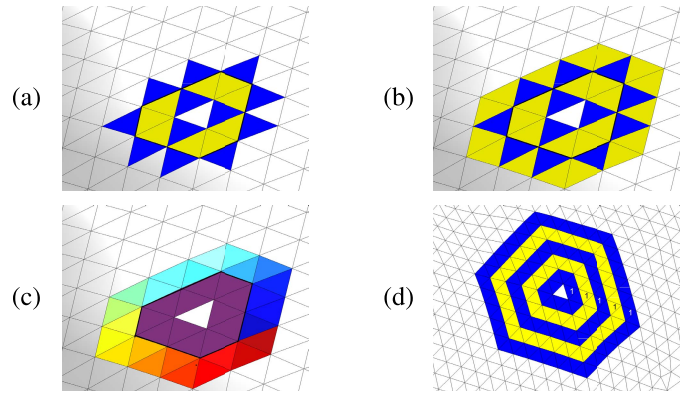


Fig. 3. Construction of multi-resolution mesh-LBP: (a) Extraction of the next set of *Fout* facets, as the facets adjacent to *Fgap* which are not part of the current ring; (b) Extracting the *Fgap* facets; (c) The second ordered ring extracted; (d) Five concentric ordered rings. Notice that the first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

---

#### 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. However, after this post-processing procedure, the arithmetic progression of the number of facets across rings is no longer satisfied, and this latter case can be used as an indicator of the local mesh irregularity. We will elaborate further on this aspect in Section III-C.

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

$$meshLBP_m^r(f_c) = \sum_{k=0}^{m-1} s(h(f_k^r) - h(f_c)) \cdot \alpha(k), \quad (3)$$

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. In principle, any predefined number of samples per ring can be used. In this work, we considered, in almost all the cases, a number of samples per ring  $m = 12$ .

### III. MESH-LBP IMPLEMENTATION

In the following, we provide more insights on the practical implementation of mesh-LBP. In particular, we propose mesh-LBP variants to reduce the descriptor size (Section III-A), together with solutions to make the mesh-LBP descriptor invariant with respect to the selection of the initial ORF facet (Section III-B), and to make it computable on meshes with non-regular tessellation (Section III-C).

#### A. Reducing Descriptor Size

The LBP operator produces rather long histograms and is therefore difficult to use as a region descriptor. A first solution to this problem was obtained by using just “uniform” patterns (i.e., binary patterns with a number of bitwise 0-1 transitions equal at most to 2) instead of all the possible ones [2].

The problem of reducing the dimensionality of the LBP descriptor also inspired the LBP variant called *center-symmetric* (CSLBP) [29], which modifies the pixels comparison scheme by computing the difference between center-symmetric pairs of pixels rather than comparing each pixel with the central pixel. This halves the number of comparisons for the same number of neighbors. In the context of mesh-LBP, the same result can be obtained using the following equation for the center symmetric mesh-LBP (mesh-CSLBP):

$$\text{meshCSLBP}_m^r(f_c) = \sum_{k=0}^{m/2-1} s(h(f_k^r) - h(f_{k+m/2}^r)) \cdot \alpha(k). \quad (4)$$

This is illustrated in the case (d) of Table II in the Appendix. In the experiments, we show the existence of the uniformity aspect in the mesh-LBP patterns, and the capability of the mesh-CSLBP of keeping virtually the same results than the basic mesh-LBP, while reducing the computational cost.

#### B. Achieving Invariance to Facets Ordering

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 (clock-wise or anti-clockwise). The last aspect can be easily fixed by orienting the normals of the mesh-manifold. For the first aspect, when the ORF are constructed around a central facet, three different orderings of the facets inside each ring can be obtained, depending from which of the three *Fout* facets, adjacent to the central facet, the first ring starts from. Therefore three different patterns can be derived from each ring. To address this ambiguity several solutions can be used:

- *Method-1*: 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];
- *Method-2*: Adopting intrinsically rotation invariant descriptors only. This set includes the number of transitions, the number of 1-valued bits (i.e., the sum of the

binary digits obtained when using  $\alpha(k) = 1$  variant), 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;

- *Method-3*: 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;
- *Method-4*: Selecting the first facet with respect to a local reference frame (LRF) determined based on the local morphology of the ring neighborhood. For this purpose, the method proposed by Tombari et al. [31], which ensures a unique and unambiguous LRF can be used. Afterwards, the nearest facet to the  $x$  or  $y$  axis of the LRF can be selected as the first facet.

From the above, the *method-4* looks the most reliable and generic, but its implementation requires histograms construction, which might burden the computational complexity. For this reason, we rather adopted a simpler yet practical solution, tailored to our problem, and which consists of the following steps: (i) First, we generate the sequence of rings starting from any arbitrary adjacent facet to the central facet; (ii) Then, from the obtained sequence of ordered rings, we select as a first facet in each ring- $r$ , the facet  $f_i$  which satisfies the following condition:

$$\min_i \text{dist}(c_o, c_i^r), \quad f_i \in \text{ring-}r, \quad (5)$$

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 rings weighted by their area; (iii) Finally, in each ring, we apply a circular shifting to the current facets ordering to bring the facet selected in step (ii) to the first position.

Fig. 4 shows the mesh-LBP maps obtained with the *method-1* and *method-2* (the number of 1-valued bits in the pattern has been used) listed above, and our proposed method for selecting the first facet of a ring. The repeatability and behavior obtained using the different methods can be appreciated. In particular, the zoomed maps in Fig. 4(b), obtained for a rectangular region at the base of the nose, show a clear overall repeatability of the mesh-LBP (last column) obtained with the proposed method. The minor disparities between the three instances emanate from the mesh variability across the scans, which in turn affects to some extent the binary patterns. The same behavior is observed for the *method-1* and *-2*. In particular, we notice the reduced range of the pattern values in *method-1*. For *method-2*, we can notice the limited description ability reflected in the similar values observed at the curve sides. On the opposite, our method looks the most effective in detecting the shape variability at that neighbors.

#### C. Mesh Quality Assessment

One issue that can hamper the repeatability of the mesh-LBP is the local irregularity of mesh tessellation, for which the assumption of vertex valence of six does not hold,



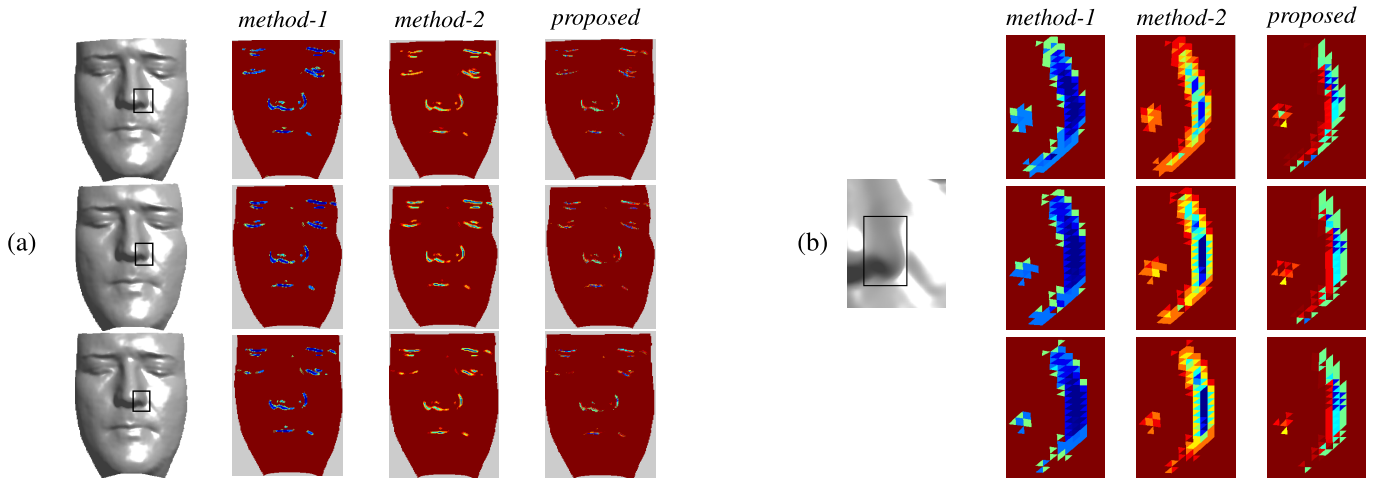


Fig. 4. Comparison of the mesh-LBP maps obtained with different methods for selecting the first facet of a ring ( $r = 1$  and  $m = 12$  are used). The maps shown represent the face surface mesh after coloring each facet in the mesh with a color representing its mesh-LBP value. (a) mesh-LBP maps obtained using, respectively, *method-1*, *method-2* (number of 1s), and our proposed method, on three different face scans of a same subject; (b) A region at the base of the nose of each scan in (a) is cropped (rectangular region framed in black), and the corresponding mesh-LBP maps are zoomed in. (The maps are best viewed on the soft-copy version).

and consequently the regular progression of Eq. (2) is not satisfied. 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 [32] to the ring vertices around the central facet.

In our experiments, we rather used a simpler technique that interpolates the scalar function used to compute mesh-LBP across each ring, so as to obtain a sequence of samples that matches the ideal progression.

We note that the progression of the number of facets across the ordered rings (see Eq. (2)), also allows establishing a simple criteria for assessing the local regularity of a triangular mesh. Indeed, given a facet neighborhood comprising  $r$  rings, we define the *local irregularity criterion* by:

$$\delta_r = \frac{\|\Gamma_r - \hat{\Gamma}_r\|}{\|\hat{\Gamma}_r\|}, \quad (6)$$

where  $\hat{\Gamma}_r$  is a vector representing the ideal sequence of the number of facets across an  $r$ -ring ORF (i.e.,  $[12, 24, \dots, 12r]$ ) according to the arithmetic progression of Eq. (2), and  $\Gamma_r$  is the actual sequence. Fig. 5(a) depicts examples of 3-ring ORF exhibiting different  $\Gamma_3$  and  $\delta_3$ .

Intuitively, the idea behind the  $\delta_r$  coefficient is that the greater is the relative deviation between the actual number of facets across the  $r$  rings with respect to its ideal number, the more the mesh is irregular in the local surface spanned by these  $r$  rings computed around a central facet. This criterion can be used to assess the local regularity of a mesh, thus to regularize the support region used in the computation of the multi-resolution mesh-LBP. Fig. 5(b)-(c) depict, respectively, a mesh sample and its corresponding map with the values of  $\delta_r$  originated using the local irregularity criterion. In the  $\delta_r$  map in (c), dark areas correspond to larger values of  $\delta_r$ ; it

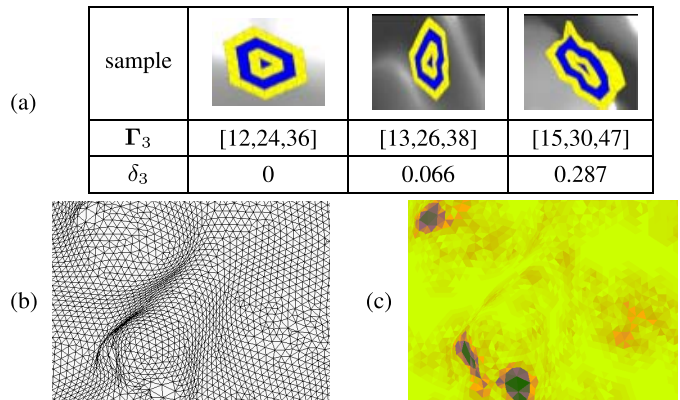


Fig. 5. (a) Examples of 3-ring ORF with their related  $\Gamma_3$  and  $\delta_3$ ; (b) Sample of a facial mesh showing local irregularities in the eye and nose regions; (c) Corresponding map obtained by computing the local irregularity criterion  $\delta_r$  at each facet.

can be observed that dark areas correspond well to the most irregular regions of the facial mesh in (b) (see, for instance, the left nostril or the right eye).

With this criteria, once an irregular mesh region is detected, a local mesh regularization approach can be applied to it so as to recover the ideal mesh tessellation for mesh-LBP computation. Using an opposite perspective, the value of  $\delta_r$  computed for the  $r$ -ring neighborhood of a facet can be used as a criteria to assess the significance of the mesh-LBP computed for the facet. According to this,  $1 - \delta_r$  could be used to weight the contribution of individual mesh-LBP values accumulated in a global histogram descriptor: the more irregular the mesh is in a facet neighborhood, the lower the contribution of the corresponding mesh-LBP to the overall descriptor. In the experiments, we found that, even without recurring to this procedure, the mesh-LBP can actually cope to a large extent with mesh irregularities.

#### IV. EXPERIMENTAL RESULTS

Experiments have two main goals: On the one hand, we investigate the basic properties of the mesh-LBP descriptor,

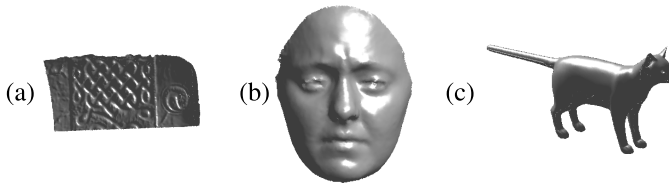


Fig. 6. Mesh models used in the uniformity experiment: (a) Portion of a pot (MIT CSAIL textured 3D models); (b) Face surface (BU-3DFE); (c) Cat model (TOSCA high-resolution).

evidencing the presence of mesh-LBP uniform patterns (Section IV-A) and the repeatability of the descriptor (Section IV-B); On the other, we evaluated the proposed framework on the specific task of 3D texture classification: first, we compare the different mesh-LBP surface functions and operators, also in comparison with some of the mesh-LBP variants (Section IV-C); then, we provide a comparative analysis of the mesh-LBP descriptor with respect to state of the art solutions which describe 3D meshes through surface descriptors, or by applying the standard 2D-LBP on depth images of the 3D meshes (Section IV-D); finally, the robustness of the mesh-LBP to mesh irregularities is also shown (see Section IV-E).

#### A. Uniform Patterns

By studying the statistics of the number of bitwise 0-1 transitions in the binary patterns, Ojala et al. [2] noticed that the majority of the patterns in textured 2D images have a number of transitions  $U$  equal at most to 2. These patterns are called “uniform”. In our investigation, we considered a representative set of three surface meshes collected from different sources. The first surface is a portion of a pot object from the “MIT CSAIL textured 3D models database” [33]. 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) [34], and shows the case of an open surface. The third one is a closed surface of a cat model from the “TOSCA high-resolution database” [35]. These models are shown in Fig. 6, from (a) to (c), respectively.

Four scalar functions ( $h(\cdot)$  in Eq. (3)) 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 of these functions, we computed the number of transitions  $U$  in the binary patterns computed by using the mesh-LBP operator of Eq. (3), 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, depicted in Fig. 7, show the percentage of facets, exhibiting a number of transitions  $U$  less or equal than 4. We can observe that this number exceeds 90% up to the third ring, across the four scalar functions, for all the three surfaces. The angle between normals is the function exhibiting the largest score 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  $U \leq 4$  above 70%. These observations provide evidence on the existence of a “uniformity” aspect of the mesh-LBP computed on triangular mesh manifolds, and thus the mesh-LBP has

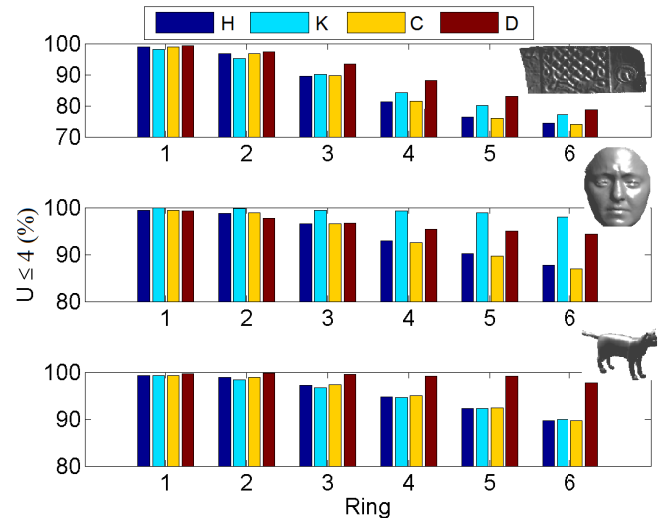


Fig. 7. Percentage of facets whose mesh-LBP have a number of transitions  $U$  less than or equal to 4 (legend: H - Mean curvature; K - Gaussian curvature; C - Curvedness; D - Angle between facets normals).

the potential of adapting to the uniformity-driven description suggested by Ojala et al. [2]. Based on the obtained results, considering an azimuthal quantization of  $m = 12$ , that is 4096 possible patterns, we define the set of uniform patterns as the set including all binary patterns for which  $U$  is at most equal to 4. This set contains exactly 1123 patterns against 2973 for the non-uniform patterns. Following the same partition scheme of [2], where all the non-uniform patterns are grouped into a single label, whereas a separate label is assigned to each non-uniform pattern, the number of labels (or classes) is reduced to 1234 for our mesh-LBP. We will adopt this partition in the rest of the experiments. Notably, this partition will be used for the mesh-LBP operator involving  $\alpha(k) = \alpha_2(k) = 2^k$ . For  $\alpha_1(k) = 1$  the distinction into uniform/non-uniform patterns does not make too much sense since the number of patterns is already small (13 patterns exactly).

#### B. Repeatability

Repeatability of mesh-LBP 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. For this experiment, we acquired 32 facial scans of a same subject with neutral or moderate facial expressions. The four scalar surface functions reported in the previous Section, namely, *mean curvature*, *gaussian curvature*, *curvedness* and *angle between facets normal* have been used for computing mesh-LBP. For each of these functions, we considered two different mesh-LBP operators, that is,  $\alpha_1(k) = 1$ ,  $\alpha_2(k) = 2^k$ . A third mesh-LBP representation has been obtained by applying the  $\alpha_2(k)$  operator just to the uniform patterns (i.e., according to the results of Section IV-A, we considered a pattern uniform if its number of transitions  $U$  is  $U \leq 4$ ). Different spatial resolutions corresponding to eight rings  $r = 1, \dots, 8$ , have been also accounted. To compute the repeatability of mesh-LBP we followed an approach similar to that proposed in [36] for 3D keypoints. With this solution, first a scan is selected as *reference*, and each of the other scans (*probe*) is aligned to the reference one using ICP registration.

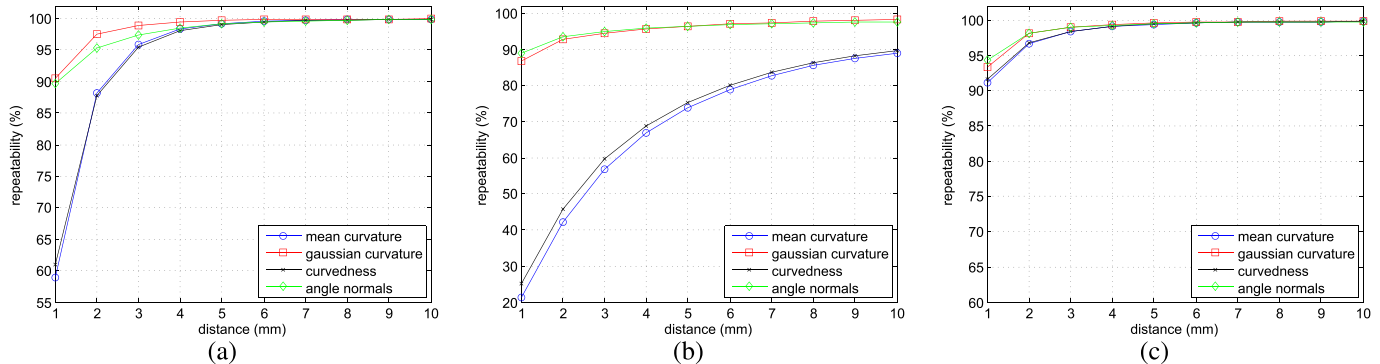


Fig. 8. Repeatability of mesh-LBP: (a)  $\alpha_1(k) = 1$ ; (b)  $\alpha_2(k) = 2^k$ ; (c)  $\alpha_2(k) = 2^k$  applied to uniform patterns (i.e., number of transitions  $U \leq 4$ ).

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 of the nearest neighbor facets in the reference 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.

Figs. 8(a)-(c) show the obtained average repeatability as a function of increasing values of the proximity radius, respectively, for the three used mesh-LBP descriptors. The plots reported in the figure concern the mesh-LBP computed on the 1st-ring, but a similar behaviour 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 similar behaviour, 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  $\alpha_1$  and  $\alpha_2$  operators. Interestingly, for all the scalar surface functions, the best repeatability is obtained for the uniform patterns  $U$  (see the plot (c) in the figure).

### C. Discriminating 3D Texture Patterns

2D-LBP has been successfully used in a number of different applications, the most notables being texture classification and face recognition. We have shown that mesh-LBP inherits many of the positive aspects of the standard LBP, further extending the range of possible applications to the direct analysis of 3D triangular meshes. As a consequence, it is expected that mesh-LBP can find application in a number of 3D scenarios, inspiring also new one. In the following, we focus on the problem of 3D texture classification. We remark here that in this study textures are intended as 3D repeatable patterns corrugating the object surface; 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 fact, 3D objects have been analyzed for classification and retrieval purposes mainly using their 3D shape. This is largely motivated by the almost complete absence of 3D textures in CAD and synthetic models

used in the majority of benchmark datasets [35], [37], [38]. Instead, the 3D surface texture is of fundamental importance to discriminate the 3D scans of real objects, which can show very similar shapes, but be well differentiable based on their 3D texture.

According to these considerations, in this experiment we investigate the potential of the mesh-LBP for discriminating texture patterns on 3D meshes. 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. For this purpose, we used surface samples exhibiting a variety of 3D shape textures, collected from eight different object models of the “MIT CSAIL textured 3D models database” [33]. These objects are *bagel*, *bird*, *gargoyle*, *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 1st row of Fig. 9 (in particular, three texture patterns were derived from the *owl* object). For each sample, we computed a 1D-histogram of the mesh-LBP operator (Eq. (3)) 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 operator function  $\alpha_1$ , the resulting mesh-LBP take values in  $[0,12]$  (i.e., in this case, the number of 1-valued bits in a pattern of 12 bits is counted), and these values are accumulated in a 1D histogram with 13 bins for each ring. For the  $\alpha_2$  operator, for which the range of mesh-LBP is  $[0,4095]$ , we adopted the uniform/non-uniform mesh-LBP partition described in Section IV-A, that is 1123 bins are used for the uniform patterns, i.e., one bin for each of the 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, which are associated, respectively, with the  $\alpha_1$  and  $\alpha_2$  operators. The histograms are computed for each sample surface by considering an area of 19 rings around the central facets in the computation of mesh-LBP, which is sufficient for covering the 3D texture variation in each sample. 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 using four scalar surface functions, namely, the *mean*



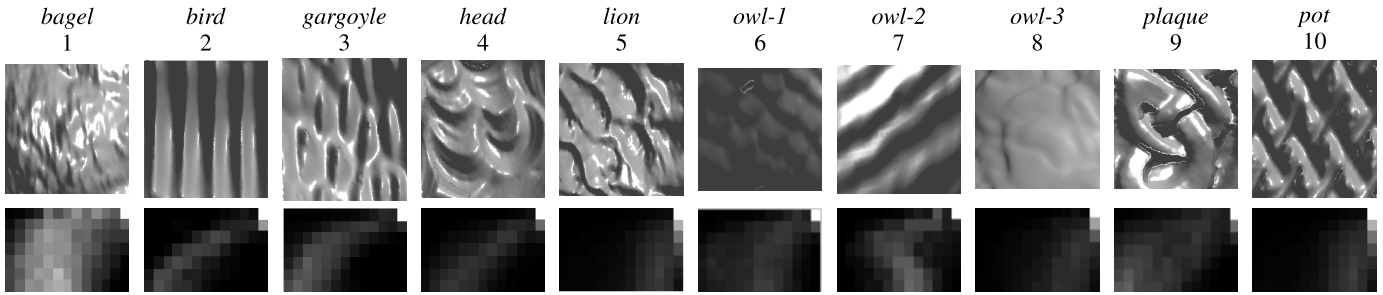


Fig. 9. Top: 3D texture samples from the ten classes. Bottom: The corresponding histograms obtained with the *angle between facets normal* and the  $\alpha_1$  weighting function using 7 rings and 12 samples per ring (i.e., histograms with 7 rows and 13 columns). Each histogram bin cumulates the frequency of a mesh-LBP pattern computed for all the facets of a sample surface (histograms are represented as gray-level images, where lighter pixels correspond to histogram bins with higher values).

curvature, the *gaussian curvature*, the *shape index* (instead of the *curvedness*) and the *angle between facets normal*. As an example, Fig. 9 (2nd row) depicts the histograms of the first type (i.e.,  $\alpha_1$  operator) obtained with the *angle between facets normal*, and computed for the sample surfaces in the first row. The histograms are obtained by reporting the frequency of the mesh-LBP patterns computed for all the facets of the sample surfaces (i.e., histograms are represented as gray level images, where lighter pixels correspond to histogram bins with higher values).

The assessment of the discriminative power of the different descriptors is performed as follows. For each texture class, we considered 30 different instances and for each of them the different descriptors have been computed. From the set associated to each texture class, we evaluate the mean and the variance. Since all the descriptors have a histogram structure, the variance we consider here is the variance of the Bhattacharyya distances between descriptor instances and their mean. For each descriptor, we compute the distance matrix of the ten texture classes, where each diagonal term is the mean intra-class distance, and the non-diagonal term is the distance between the mean of class  $i$  and the mean of class  $j$ . The so-obtained  $10 \times 10$  distance matrices provide a coarse assessment of the discriminative power of the descriptors.

Figs. 10 and 11 depict, respectively, the distance matrices related to the different mesh-LBP surface descriptors for  $\alpha_1(k) = 1$  and  $\alpha_2(k) = 2^k$ . For the mesh-LBP descriptor, we notice that the intra-class distance is quite below the inter-class distance across all the different descriptors and the two operator functions. To evidence this behavior, in the confusion matrices reported for the different cases, we highlighted the intra-class and inter-class distances that are less separated (in gray and yellow, respectively), and so that are more susceptible to be confused with each other. Even in the worst cases, it can be observed that the ratio between the inter-class distances and the corresponding intra-class distance is greater than 2.33, for  $\alpha_1$  and SI, and of 3.37 for  $\alpha_2$  and SI. This is a clear indication of the potential and the appropriateness of the mesh-LBP descriptors for discriminating textured shapes.

Fig. 12 reports the distance matrices between all the classes' instances (i.e., 30 instances for each of the 10 classes). Results for the mesh-LBP computed with the scalar functions  $H$ ,  $K$ ,  $SI$  and  $D$ , for the  $\alpha_1$  and  $\alpha_2$  operators are depicted in the top and bottom row, respectively. In the mesh-LBP

$$H$$

	1	2	3	4	5	6	7	8	9	10
1	0.01	0.13	0.08	0.08	0.05	0.10	0.07	0.06	0.08	0.07
2	-	0.03	0.13	0.13	0.13	0.17	0.13	0.14	0.14	0.12
3	-	-	0.01	0.05	0.06	0.13	0.07	0.08	0.06	0.06
4	-	-	-	0.01	0.07	0.14	0.07	0.09	0.06	0.06
5	-	-	-	-	0.01	0.11	0.06	0.05	0.07	0.05
6	-	-	-	-	-	0.02	0.10	0.09	0.13	0.12
7	-	-	-	-	-	-	0.01	0.07	0.08	0.06
8	-	-	-	-	-	-	-	0.02	0.08	0.07
9	-	-	-	-	-	-	-	-	0.02	0.06
10	-	-	-	-	-	-	-	-	-	0.01

$$K$$

	1	2	3	4	5	6	7	8	9	10
1	0.01	0.16	0.16	0.07	0.35	0.49	0.66	0.23	0.07	0.19
2	-	0.02	0.12	0.16	0.40	0.54	0.69	0.32	0.16	0.16
3	-	-	0.01	0.16	0.43	0.55	0.70	0.35	0.15	0.06
4	-	-	-	0.02	0.34	0.49	0.66	0.23	0.07	0.19
5	-	-	-	-	0.02	0.25	0.51	0.18	0.36	0.45
6	-	-	-	-	-	0.02	0.28	0.38	0.51	0.56
7	-	-	-	-	-	-	0.02	0.60	0.68	0.69
8	-	-	-	-	-	-	-	0.01	0.25	0.37
9	-	-	-	-	-	-	-	-	0.02	0.17
10	-	-	-	-	-	-	-	-	-	0.02

$$SI$$

	1	2	3	4	5	6	7	8	9	10
1	0.01	0.11	0.06	0.07	0.04	0.07	0.05	0.04	0.06	0.07
2	-	0.03	0.13	0.12	0.11	0.14	0.12	0.12	0.11	0.13
3	-	-	0.01	0.06	0.06	0.07	0.06	0.07	0.05	0.06
4	-	-	-	0.01	0.06	0.09	0.05	0.08	0.06	0.05
5	-	-	-	-	0.01	0.07	0.05	0.06	0.05	0.07
6	-	-	-	-	-	0.03	0.07	0.07	0.08	0.10
7	-	-	-	-	-	-	0.01	0.06	0.06	0.06
8	-	-	-	-	-	-	-	0.01	0.07	0.08
9	-	-	-	-	-	-	-	-	0.01	0.07
10	-	-	-	-	-	-	-	-	-	0.01

$$D$$

	1	2	3	4	5	6	7	8	9	10
1	0.01	0.38	0.28	0.33	0.50	0.27	0.48	0.21	0.18	0.30
2	-	0.03	0.27	0.24	0.54	0.42	0.51	0.33	0.34	0.26
3	-	-	0.02	0.34	0.61	0.42	0.59	0.33	0.23	0.08
4	-	-	-	0.01	0.39	0.32	0.35	0.22	0.32	0.34
5	-	-	-	-	0.02	0.43	0.10	0.37	0.49	0.62
6	-	-	-	-	-	0.02	0.38	0.21	0.37	0.42
7	-	-	-	-	-	-	0.01	0.34	0.49	0.60
8	-	-	-	-	-	-	-	0.03	0.23	0.35
9	-	-	-	-	-	-	-	-	0.02	0.26
10	-	-	-	-	-	-	-	-	-	0.02

Fig. 10. Distance matrices between the 3D texture classes. The  $\alpha_1$  operator is used in the mesh-LBP computation using the following descriptors (from top): mean curvature ( $H$ ), gaussian curvature ( $K$ ), shape index ( $SI$ ) and angle between facets normal ( $D$ ). The intra-class and inter-class distances that are less separated are highlighted in gray and yellow, respectively.

distance matrices, we can easily distinguish the  $30 \times 30$  blocks related to the inter-class distances between class pairs. This observation confirms the discriminant capability of the mesh-LBP descriptors. 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. A perfect classification of 100% is obtained in all the cases.

1) *Mesh-LBP Variants*: We conducted the same texture classification experiment with the mesh-CSLBP variant (Eq. (4)). In this variant, we kept the same spatial resolution

<i>H</i>										
	1	2	3	4	5	6	7	8	9	10
1	0.10	0.58	0.47	0.48	0.45	0.58	0.49	0.46	0.51	0.47
2	-	0.13	0.56	0.56	0.55	0.63	0.59	0.56	0.61	0.54
3	-	-	0.11	<b>0.43</b>	<b>0.43</b>	0.59	0.46	0.46	0.45	<b>0.43</b>
4	-	-	-	0.10	0.44	0.61	0.50	0.48	0.48	0.45
5	-	-	-	-	0.11	0.56	0.46	0.44	0.48	<b>0.43</b>
6	-	-	-	-	-	0.13	0.57	0.55	0.61	0.57
7	-	-	-	-	-	-	0.11	0.47	0.48	0.46
8	-	-	-	-	-	-	-	0.10	0.48	0.44
9	-	-	-	-	-	-	-	-	0.11	0.46
10	-	-	-	-	-	-	-	-	-	0.11

<i>K</i>										
	1	2	3	4	5	6	7	8	9	10
1	0.10	0.59	0.50	0.47	0.52	0.62	0.69	0.50	0.52	0.51
2	-	0.16	<b>0.55</b>	0.56	0.63	0.70	0.75	0.62	0.62	0.57
3	-	-	-	0.11	0.46	0.56	0.67	0.72	0.55	0.48
4	-	-	-	0.10	0.48	0.62	0.67	0.50	0.47	0.49
5	-	-	-	-	0.08	0.49	0.54	0.44	0.56	0.59
6	-	-	-	-	-	0.06	0.44	0.54	0.68	0.68
7	-	-	-	-	-	-	0.06	0.64	0.71	0.73
8	-	-	-	-	-	-	-	0.10	0.56	0.58
9	-	-	-	-	-	-	-	-	0.12	0.52
10	-	-	-	-	-	-	-	-	-	0.11

<i>SI</i>										
	1	2	3	4	5	6	7	8	9	10
1	0.09	0.57	0.45	0.45	0.43	0.51	0.45	0.42	0.46	0.46
2	-	0.16	0.56	<b>0.54</b>	<b>0.54</b>	0.62	0.55	0.57	0.55	0.56
3	-	-	0.11	0.40	0.42	0.52	0.44	0.45	0.43	0.43
4	-	-	-	0.10	0.40	0.53	0.44	0.45	0.42	0.41
5	-	-	-	-	0.11	0.51	0.42	0.43	0.42	0.44
6	-	-	-	-	-	0.14	0.53	0.51	0.53	0.54
7	-	-	-	-	-	-	0.11	0.45	0.45	0.46
8	-	-	-	-	-	-	-	0.10	0.46	0.47
9	-	-	-	-	-	-	-	-	0.12	0.44
10	-	-	-	-	-	-	-	-	-	0.10

<i>D</i>										
	1	2	3	4	5	6	7	8	9	10
1	0.10	0.68	0.56	0.63	0.67	0.69	0.77	0.69	0.58	0.56
2	-	0.11	0.60	0.58	0.69	0.75	0.76	0.72	0.66	0.58
3	-	-	0.10	0.53	0.71	0.75	0.79	0.73	0.48	<b>0.41</b>
4	-	-	-	0.09	0.55	0.77	0.71	0.73	0.55	0.53
5	-	-	-	-	0.09	0.76	0.64	0.73	0.68	0.71
6	-	-	-	-	-	0.05	0.49	0.35	0.79	0.74
7	-	-	-	-	-	-	0.04	0.40	0.78	0.79
8	-	-	-	-	-	-	-	0.04	0.74	0.72
9	-	-	-	-	-	-	-	-	0.11	0.52
10	-	-	-	-	-	-	-	-	-	0.11

Fig. 11. Distance matrices between the 3D texture classes. In this case, the  $\alpha_2$  operator function is used in the mesh-LBP computation.

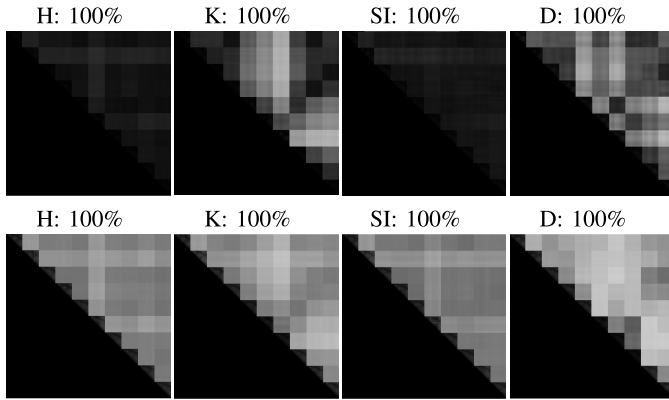


Fig. 12. Matrices reporting the distances between all the instances of the texture classes (30 instances per class). Distances are computed for the mesh-LBP obtained using *H*, *K*, *SI*, and *D* scalar surface descriptors (top row for  $\alpha_1$  and bottom row for  $\alpha_2$ , respectively). The 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.

and the azimuthal quantization ( $r = 7, m = 12$ ), but the mesh-LBP patterns are now coded on 6 digits, setting thus their ranges to  $[0,6]$  and  $[0,63]$  for  $\alpha_1$  and  $\alpha_2$ , respectively. We also adopted the uniform/non-uniform partition as for mesh-LBP, though the resulting number of classes (i.e., the histogram bins) is not significantly reduced (62 instead of 64). Fig. 13 depicts the distance matrices between all the 30 classes' instances, computed with the four previously used scalar functions (*H*, *K*, *SI* and *D*), together with their corresponding

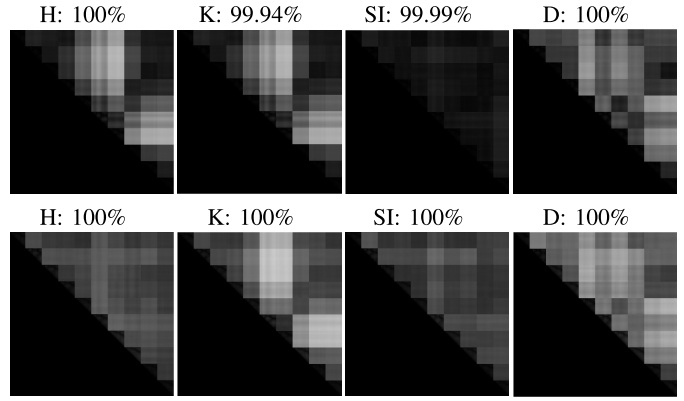


Fig. 13. Distance matrices obtained for the mesh-CSLBP using *H*, *K*, *SI* and *D* scalar surface descriptors (top row for  $\alpha_1$  and bottom row for  $\alpha_2$ , respectively), and their related classification accuracies.

<i>H</i>										
	1	2	3	4	5	6	7	8	9	10
1	0.02	0.13	0.07	0.07	0.07	0.16	0.13	0.09	0.13	0.09
2	-	0.02	0.10	0.11	0.15	0.15	0.12	0.11	0.13	0.09
3	-	-	-	0.01	0.08	0.10	0.11	0.08	0.04	0.08
4	-	-	-	-	0.01	0.08	0.16	0.13	0.09	0.14
5	-	-	-	-	-	0.01	0.17	0.15	0.12	0.16
6	-	-	-	-	-	-	0.02	0.09	0.10	0.09
7	-	-	-	-	-	-	-	0.02	0.07	<b>0.05</b>
8	-	-	-	-	-	-	-	-	0.01	0.08
9	-	-	-	-	-	-	-	-	-	0.01
10	-	-	-	-	-	-	-	-	-	0.01

<i>K</i>										
	1	2	3	4	5	6	7	8	9	10
1	0.01	0.13	0.11	0.09	0.30	0.49	0.61	0.19	0.10	0.09
2	-	0.01	0.09	0.19	0.40	0.59	0.71	0.27	0.12	0.11
3	-	-	0.01	0.16	0.38	0.57	0.69	0.25	0.07	0.09
4	-	-	-	0.01	0.23	0.43	0.56	0.12	0.14	0.16
5	-	-	-	-	0.02	0.21	0.34	0.19	0.35	0.37
6	-	-	-	-	-	0.05	<b>0.16</b>	0.38	0.54	0.56
7	-	-	-	-	-	-	-	0.03	0.51	0.66
8	-	-	-	-	-	-	-	-	0.01	0.22
9	-	-	-	-	-	-	-	-	-	0.01
10	-	-	-	-	-	-	-	-	-	0.01

<i>SI</i>										
	1	2	3	4	5	6	7	8	9	10
1	0.01	0.11	0.08	0.07	0.04	0.10	0.06	0.04	0.06	0.13
2	-	0.01	0.09	0.06	0.11	0.14	0.11	0.11	0.08	0.07
3	-	-	0.01	0.06	0.07	0.09	0.06	0.07	0.04	0.09
4	-	-	-	0.01	0.06	0.11	0.08	0.07	0.05	0.07
5	-	-	-	-	0.01	0.09	0.06	<b>0.03</b>	0.05	0.11
6	-	-	-	-	-	0.02	0.07	0.09	0.09	0.14
7	-	-	-	-	-	-	0.02	0.06	0.06	0.12
8	-	-	-	-	-	-	-	0.01	0.05	0.12
9	-	-	-	-	-	-	-	-	0.01	0.09
10	-	-	-	-	-	-	-	-	-	0.01

<i>D</i>										
	1	2	3	4	5	6	7	8	9	10
1	0.02	0.27	0.25	0.29	0.50	0.21	0.41	0.22	0.27	0.27
2	-	0.02	0.21	0.21	0.55	0.35	0.49	0.38	0.22	0.22
3	-	-	0.02	0.21	0.60	0.35	0.53	0.39	0.17	<b>0.08</b>
4	-	-	-	0.02	0.48	0.30	0.42	0.36	0.19	0.22
5	-	-	-	-	0.02	0.36	0.14	0.34	0.59	0.64
6	-	-	-	-	-	0.03	0.27	0.17	0.35	0.38
7	-	-	-	-	-	-	0.02	0.25	0.53	0.57
8	-	-	-	-	-	-	-	0.02	0.41	0.43
9	-	-	-	-	-	-	-	-	0.02	0.15
10	-	-	-	-	-	-	-	-	-	0.02

Fig. 14. Distance matrices between the 3D texture classes computed with the mesh-CSLBP,  $\alpha_1$  operator, and the four descriptors.

accuracy rates. The  $10 \times 10$  distance matrices related to  $\alpha_1$  and  $\alpha_2$  are also depicted in Fig. 14 and 15, respectively.

We notice that the accuracy rate is virtually 100% and exactly 100% for  $\alpha_1$  and  $\alpha_2$ , respectively, across the four descriptors. Also, in the worst cases, the ratio between the interclass distances and the corresponding intra-class distance is scoring 2.4 for both  $\alpha_1$  (*SI*) and  $\alpha_2$  (*SI*). These scores confirm the discriminant capability of the mesh-CSLBP, though to a less extent than the mesh-LBP, for which the corresponding ratios are 2.33 and 3.37. However, this inferiority is expected because of the lower range of the mesh-CSLBP pattern.

H										
	1	2	3	4	5	6	7	8	9	10
1	0.04	0.28	0.21	0.20	0.18	0.32	0.26	0.20	0.25	0.21
2	-	0.06	0.30	0.29	0.30	0.35	0.34	0.30	0.35	0.26
3	-	-	0.06	0.20	0.25	0.34	0.27	0.24	0.21	0.19
4	-	-	-	0.05	0.24	0.36	0.27	0.24	0.25	0.21
5	-	-	-	-	0.05	0.35	0.30	0.24	0.30	0.22
6	-	-	-	-	-	0.07	0.36	0.32	0.34	0.31
7	-	-	-	-	-	-	0.08	0.26	0.30	0.26
8	-	-	-	-	-	-	-	0.08	0.25	0.22
9	-	-	-	-	-	-	-	-	0.04	0.21
10	-	-	-	-	-	-	-	-	-	0.04

K										
	1	2	3	4	5	6	7	8	9	10
1	0.04	0.27	0.22	0.20	0.34	0.66	0.66	0.27	0.26	0.20
2	-	0.05	0.27	0.30	0.46	0.75	0.76	0.38	0.31	0.29
3	-	-	0.05	0.24	0.44	0.75	0.75	0.34	0.21	0.21
4	-	-	-	0.06	0.30	0.63	0.63	0.24	0.26	0.25
5	-	-	-	-	0.05	0.40	0.42	0.26	0.42	0.40
6	-	-	-	-	-	0.08	0.16	0.55	0.70	0.71
7	-	-	-	-	-	-	0.05	0.55	0.71	0.72
8	-	-	-	-	-	-	-	0.08	0.33	0.35
9	-	-	-	-	-	-	-	-	0.03	0.27
10	-	-	-	-	-	-	-	-	-	0.04

SI										
	1	2	3	4	5	6	7	8	9	10
1	0.04	0.32	0.19	0.21	0.17	0.25	0.18	0.17	0.23	0.25
2	-	0.05	0.28	0.27	0.27	0.39	0.31	0.35	0.23	0.28
3	-	-	0.05	0.17	0.18	0.27	0.19	0.23	0.17	0.20
4	-	-	-	0.05	0.18	0.31	0.22	0.25	0.17	0.19
5	-	-	-	-	0.05	0.27	0.19	0.21	0.18	0.22
6	-	-	-	-	-	0.08	0.27	0.26	0.30	0.30
7	-	-	-	-	-	-	0.07	0.23	0.22	0.24
8	-	-	-	-	-	-	-	0.06	0.26	0.27
9	-	-	-	-	-	-	-	-	0.04	0.21
10	-	-	-	-	-	-	-	-	-	0.04

D										
	1	2	3	4	5	6	7	8	9	10
1	0.04	0.47	0.38	0.40	0.55	0.38	0.51	0.31	0.35	0.37
2	-	0.04	0.35	0.39	0.63	0.55	0.63	0.54	0.38	0.38
3	-	-	0.07	0.31	0.64	0.48	0.59	0.46	0.25	0.22
4	-	-	-	0.06	0.55	0.44	0.53	0.45	0.28	0.28
5	-	-	-	-	0.06	0.44	0.29	0.42	0.64	0.68
6	-	-	-	-	-	0.07	0.42	0.36	0.48	0.51
7	-	-	-	-	-	-	0.09	0.39	0.59	0.64
8	-	-	-	-	-	-	-	0.07	0.46	0.48
9	-	-	-	-	-	-	-	-	0.04	0.24
10	-	-	-	-	-	-	-	-	-	0.05

Fig. 15. Distance matrices between the 3D texture classes computed with the mesh-CSLBP,  $\alpha_2$  operator, and the four descriptors.

2) *Mesh-LBP Discriminative Power*: In order to compare quantitatively the different cases using a synthetic performance indicator, the discriminative power of the mesh-LBP descriptors and the mesh-CSLBP descriptors has been evaluated according to the following criterion:

$$\mathcal{J} = \sum_{i=1}^M \sum_{j=i+1}^M \mathcal{D}_{ij}, \quad (7)$$

where  $M$  is the number of texture classes.  $\mathcal{D}_{ij}$  is the probabilistic-like inter-class separation between texture classes  $i$  and  $j$  defined as follows:

$$\mathcal{D}_{ij} = \frac{1}{2} \text{dist}(\bar{H}_i, \bar{H}_j)^2 \left( \frac{1}{\sigma_{H_i}^2} + \frac{1}{\sigma_{H_j}^2} \right) + \frac{1}{2} \left( \frac{\sigma_{H_i}^2}{\sigma_{H_j}^2} + \frac{\sigma_{H_j}^2}{\sigma_{H_i}^2} - 2 \right),$$

where  $(\bar{H}_i, \bar{H}_j)$  and  $(\sigma_{H_i}, \sigma_{H_j})$  are the mean histograms and the variances of the texture classes  $i$  and  $j$ , respectively.

The criterion  $\mathcal{J}$  computed for the different mesh-LBP and mesh-CSLBP descriptors is reported in Table I. For both variants, we notice that  $K$  and  $D$  score the best performance for the  $\alpha_1$  and  $\alpha_2$  operators, respectively. The same ranking is kept for the other descriptors  $H$  and  $SI$ .

### D. Comparative Evaluation

In the following, we compared the mesh-LBP descriptors performance, in terms of 3D texture classification, with other

TABLE I  
DISCRIMINATIVE POWER  $\mathcal{J}$  COMPUTED FOR THE DIFFERENT MESH-LBP AND MESH-CSLBP DESCRIPTORS

		H	K	SI	D
mesh-LBP	$\mathcal{J}(\alpha_1)$	16.8	228.3	9.7	128.4
	$\mathcal{J}(\alpha_2)$	61.1	100.0	54.4	172.8
mesh-CSLBP	$\mathcal{J}(\alpha_1)$	20.4	233.0	15.0	187.1
	$\mathcal{J}(\alpha_2)$	41.4	111.2	34.5	105.0

standard 3D surface descriptors (Section IV-D.1) and the 2D-LBP applied to depth images (Section IV-D.2).

1) *3D Surface Descriptors*: In this analysis, we considered the following 3D surface descriptors: the *Geometric Histograms* (GH) [39]; the *Shape Distribution* variants [40], 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 ( $A3$ ); the *Spin-Images* [41]; and the *mesh-HOG* [42]. Using these descriptors, we performed the same experiments discussed above for the mesh-LBP. The distance matrices between all the classes' instances are reported in Fig. 16. Comparing these distance matrices with those obtained for the mesh-LBP using different descriptors and reported in Fig. 12, it clearly emerges the performance improvement obtained using the mesh-LBP approach.

Similarly to the results presented for the mesh-LBP, we also provide the distance matrices obtained between the different texture classes. In this case, we report just the results for the best competing solutions as resulted from Fig. 16, that is, the descriptor  $D4$ , which resulted the best among the *Shape Distributions*, and the *Spin Images* that resulted the most effective among the other descriptors. In these matrices, depicted in Fig. 17, the cases in which the intra-class distance is greater than the corresponding inter-class distances are highlighted in gray and red, respectively. It can be observed that this case occurs for several pairs of texture classes for both the matrices, whereas this is never the case for the distance matrices obtained for the mesh-LBP descriptors, where the intra-class distances are lower than the corresponding inter-class distances across all the cases.

2) *2D-LBP on Depth Images*: We conducted an additional experiment to assess the mesh-LBP performance with respect to the 2D-LBP counterpart applied on depth images [25]. For this purpose, we considered 30 depth image samples for each texture surface (see Fig. 18). In each set, the samples were constructed at different rotation angles, varying from 0 to  $2\pi/3$ , around the surface's principal orientation, to avoid self-occlusion effects. For each sample, we computed multi-resolution 2D-LBP patterns with nearly the same 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. In addition, we adopted the local descriptors  $H$ ,  $K$ ,  $SI$ , and  $C$ , rather than the depth value (e.g.,  $z$  coordinate) used usually in the standard 3D-LBP.

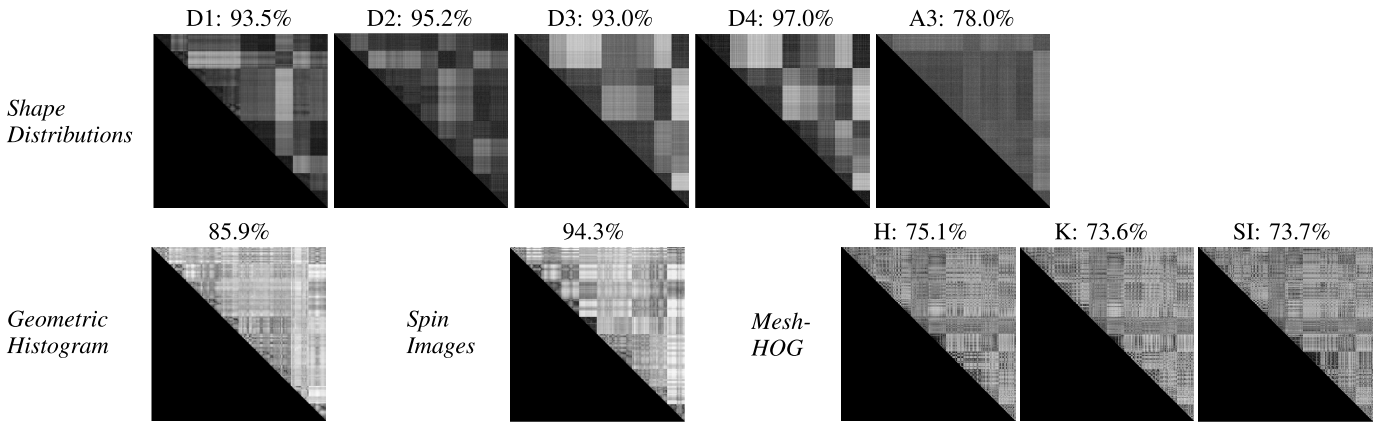


Fig. 16. Matrices reporting the distances between all the instances of the texture classes (30 instances per class). Distances are computed for: *Shape Distributions* (top); *Geometric Histogram* (bottom left); *Spin Images* (bottom middle); *mesh-HOG* (bottom right) computed for different surface scalar functions, namely, H, K and SI. For each descriptor, the overall classification accuracy is also reported in percentage.

	1	2	3	4	5	6	7	8	9	10	
1	-	0.11	0.09	0.56	0.73	0.68	0.17	0.31	0.45	0.66	0.11
2	-	-	0.13	0.55	0.71	0.66	0.20	0.31	0.43	0.64	0.12
3	-	-	-	0.18	0.24	0.19	0.46	0.32	0.14	0.13	0.61
4	-	-	-	-	0.23	0.09	0.65	0.53	0.36	0.22	0.76
5	-	-	-	-	-	0.22	0.60	0.48	0.30	0.18	0.71
6	-	-	-	-	-	-	0.13	0.17	0.34	0.57	0.25
7	-	-	-	-	-	-	-	0.15	0.19	0.42	0.38
8	-	-	-	-	-	-	-	-	0.23	0.25	0.50
9	-	-	-	-	-	-	-	-	-	0.18	0.70
10	-	-	-	-	-	-	-	-	-	-	0.13

	1	2	3	4	5	6	7	8	9	10	
1	-	0.64	0.54	0.37	0.60	0.65	0.59	0.67	0.60	0.44	0.41
2	-	-	0.38	0.50	0.33	0.74	0.64	0.72	0.55	0.33	0.72
3	-	-	-	0.30	0.57	0.48	0.61	0.62	0.62	0.33	0.37
4	-	-	-	-	0.35	0.74	0.60	0.68	0.48	0.34	0.80
5	-	-	-	-	-	0.16	0.70	0.60	0.73	0.55	0.67
6	-	-	-	-	-	-	0.50	0.35	0.27	0.56	0.70
7	-	-	-	-	-	-	-	0.42	0.46	0.60	0.73
8	-	-	-	-	-	-	-	-	0.55	0.52	0.75
9	-	-	-	-	-	-	-	-	-	0.52	0.60
10	-	-	-	-	-	-	-	-	-	-	0.39

Fig. 17. Distance matrices for the *Shape Distribution D4* and the *Spin Images*. The cases in which the intra-class distance is greater than the corresponding inter-class distances are highlighted (gray and red colors are used for, respectively, the intra- and inter-class distances).

We computed, the classification rate for the three 2D-LBP variants, namely, the uniform LBP ( $u2$ ), the rotation invariant LBP ( $ri$ ) and the uniform rotation-invariant LBP ( $riu2$ ). Fig. 19 depicts the obtained classification rates and the distance matrices for each variant. First, we notice the low performance of the  $u2$  variant, which naturally is expected because its sensitivity to surface rotation. The  $ri$  and the  $riu2$  show much better performance, but they remain lower than their mesh-LBP counterpart across all the instances. We also computed the distance matrices, showing inter-class and the intra-class distances, between the different texture classes. We reported only the results related to the best competing variants, namely,  $K_{riu2}$  and  $SI_{ri}$ , which are depicted in Fig. 20.

### E. 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 arithmetic progression of the number of facets across the rings, expressed in Eq. (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 wanted to 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 following corruption procedure reported in Algorithm 4.

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

$$t(v) = v + \sigma \vec{u}, \quad (8)$$

where  $\sigma$  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. The combination of this transformation 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 ten 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. In turn, each set contains the 30 instances of the class. Fig. 21 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 obtained classification rates are depicted in Fig. 22. 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  $\alpha(k) = 2^k$  operator (Fig. 22(b)). For this category,



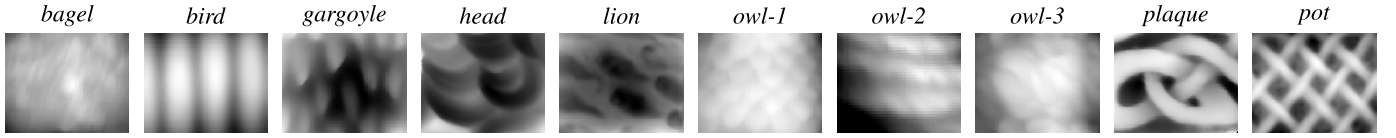


Fig. 18. Depth images of 10 3D texture classes.

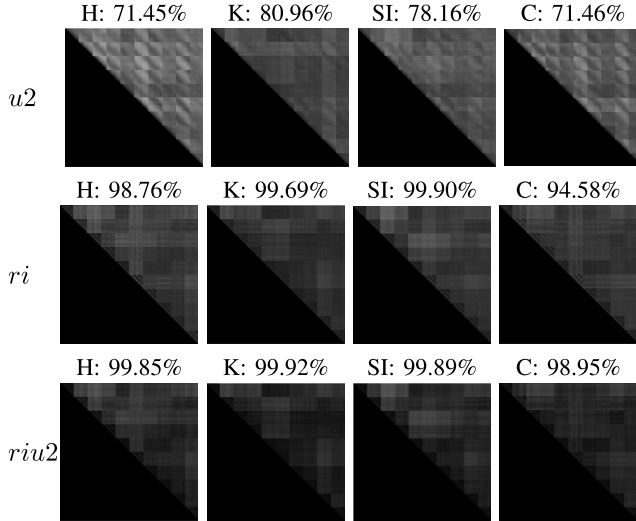


Fig. 19. Matrices reporting the distances between all the instances of the texture classes computed from depth images (30 depth images per class) using 2D-LBP patterns. Distances are computed for the uniform LBP ( $u2$ ), rotation invariant ( $ri$ ), and uniform rotation-invariant ( $riu2$ ). The 2D-LBP patterns were computed for each of the scalar functions  $H$ ,  $K$ ,  $SI$  and  $C$ . The overall classification accuracy is also reported in percentage.

	1	2	3	4	5	6	7	8	9	10
1	0.06	0.23	0.18	0.21	0.08	0.09	0.14	0.16	0.22	0.18
2	-	0.07	0.11	0.14	0.19	0.21	0.15	0.14	0.10	0.13
3	-	-	0.06	0.07	0.13	0.14	0.08	0.07	0.07	0.07
4	-	-	-	0.05	0.15	0.15	0.10	0.09	0.10	0.09
5	-	-	-	-	0.05	0.08	0.10	0.10	0.16	0.13
6	-	-	-	-	-	0.06	0.10	0.11	0.19	0.14
7	-	-	-	-	-	-	0.05	0.06	0.12	0.07
8	-	-	-	-	-	-	-	0.05	0.12	0.08
9	-	-	-	-	-	-	-	-	0.05	0.12
10	-	-	-	-	-	-	-	-	-	0.05

	1	2	3	4	5	6	7	8	9	10
1	0.13	0.21	0.35	0.27	0.13	0.20	0.16	0.20	0.22	0.22
2	-	0.16	0.26	0.19	0.16	0.22	0.17	0.19	0.14	0.13
3	-	-	0.13	0.15	0.28	0.30	0.27	0.21	0.19	0.23
4	-	-	-	0.13	0.20	0.22	0.19	0.15	0.14	0.16
5	-	-	-	-	0.13	0.16	0.11	0.13	0.15	0.15
6	-	-	-	-	-	0.16	0.15	0.16	0.22	0.19
7	-	-	-	-	-	-	0.13	0.13	0.15	0.14
8	-	-	-	-	-	-	-	0.14	0.14	0.14
9	-	-	-	-	-	-	-	-	0.13	0.12
10	-	-	-	-	-	-	-	-	-	0.13

Fig. 20. Distance matrices for the 2D-LBP:  $K_{riu2}$  and  $SI_{ri}$ . Cases when the intra-class distance is greater than the corresponding inter-class distances are highlighted (gray and red colors are used for, respectively, the intra- and inter-class distances).

we notice in particular that with *gaussian* curvature, the descriptor keeps above 99% accuracy across all the corruption levels, seconded by the *Shape Index* (SI), which is showing similar performance up to the 37th level. In the first category (Fig. 22(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.

#### Algorithm 4 Triangular Mesh Corruption Procedure

```

procedure MESHCORRUPT( )
  for  $m = 10 : 10 : 80$  do
    for  $n = 10 : 10 : 50$  do
      Apply random perturbation to  $m\%$  randomly
      selected facet's vertex
      Collapse  $n\%$  randomly selected edges
    end for
  end for
end procedure

```

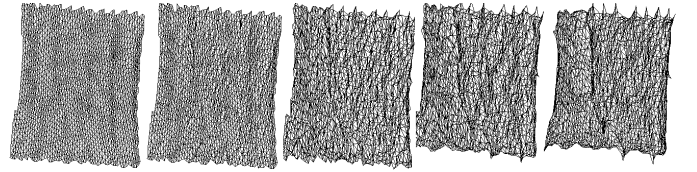


Fig. 21. The original mesh (left) and 4 corrupted instances at levels 1, 11, 21, and 31.

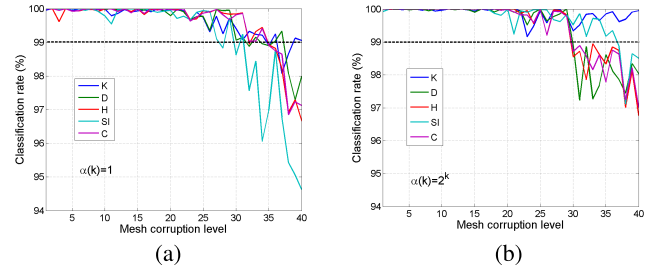


Fig. 22. Classification accuracy obtained for the different mesh corruption levels: (a)  $\alpha(k) = 1$ ; (b)  $\alpha(k) = 2^k$ .

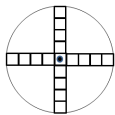
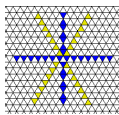
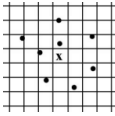
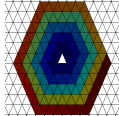
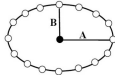
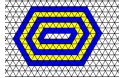
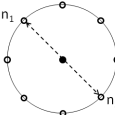

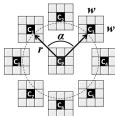
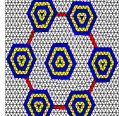
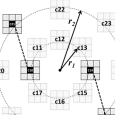
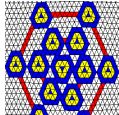
## V. DISCUSSION AND CONCLUSIONS

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 experimental tests revealed that mesh-LBP exhibits a “uniformity” aspect for the different types of scalar functions, pretty similar to the one noticed in 2D-LBP. We also provided a simple method for addressing rotation invariance that proved to be effective as was confirmed by repeatability and the other subsequent experiments.

Experiments 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 popular 3D descriptors. Experiments related to the mesh-CSLBP variant showed that we can keep virtually the same performance, while reducing the computational cost. Regarding the choice of the scalar function, in summary, the angle between facets normal and the gaussian curvature seem

TABLE II  
DIFFERENT VARIANTS OF THE LBP OPERATOR, GROUPED IN FOUR CATEGORIES, AND THE CORRESPONDING PATTERNS OBTAINED FOR THE 2D-LBP AND THE MESH-LBP ARE REPORTED. UNDER THE COLUMN “CLASS OF VARIATION”, WE EVIDENCE THE MAIN ASPECT OF THE LBP COMPUTATION FRAMEWORK, WHICH IS VARIED BY A PARTICULAR SOLUTION

Class of variation	LBP variant	2D-LBP pattern	mesh-LBP pattern
Browsing path	(a) Local-line [43]		
	(b) Archimedian Spiral [44]		
Contour / initial seed	(c) Elongated [18]		
Comparison scheme	(d) Center-symmetric [29]		
Structural element	(e) Three-Patch [10]		
	(f) Four-Patch [10]		

the more effective surface descriptors to be used within the mesh-LBP framework as emerges throughout the different experiments. The same experiments that were carried out with depth image modalities, confirmed also the superiority of our mesh-LBP, noting also the constraints on the depth image construction procedure that we had to consider to obtain the desired quality in terms of pattern visibility. It is also noticeable, in particular, that the rotation invariant mesh-LBP  $\alpha_1$  outperforms its 2D-LBP ‘ri’ variant despite the lower size of its associate histogram (13 against 36 for ‘ri’).

The re-sampling scheme of the scalar function over each ORF ring proved to be an effective mechanism for addressing mesh irregularities. In the related experiment, the gaussian curvature and the shape index exhibited the best robustness score.

The comparison of the  $\alpha_1$  and  $\alpha_2$  operators does not provide conclusive results, apart that they perform best with gaussian curvature and the angle between facets normal, respectively. However, the compactness of the descriptor obtained with the  $\alpha_1$  operator, and the resulting lower computational complexity required to compare descriptors, vote for this solution especially in the cases where time constraints are relevant.

As future work, we plan extending the mesh-LBP to global analysis. One potential approach is extracting ordered blocks from the mesh surfaces and then construct from them, by concatenation, a global histogram. We believe that mesh-LBP will open-up new perspectives for mesh manifold analysis and

will be an appropriate complement to other mesh manifold analysis techniques.

APPENDIX

In this appendix, we show that most, if not all, the different LBP neighborhood and operator variants proposed in the literature [24] can be easily derived from the ordered rings structure of the mesh-LBP. In fact, one important feature of the mesh-LBP is that the topology of the neighborhood from which the descriptor is computed can be changed to accommodate the specificities of a given shape analysis application. Some of the most effective and used LBP variants, their structure and the related mesh-LBP patterns are summarized in Table II, where the LBP variants are organized in four categories, according to which aspect of the basic LBP computation framework is varied. In the following, we provide more details about the definition and computation of the mesh-LBP variants:

*Browsing Path:* Considering a set of directions  $D_j$ , a mesh-LBP operator can be defined which uniformly samples  $m$  facets along the directions  $D_j$ :

$$meshLBP_m^{D_j}(f_c) = \sum_{k=0}^{m-1} s(h(f_k^j) - h(f_c)) \cdot \alpha(k). \quad (9)$$

This directional extension of the mesh-LBP can be regarded as a generalization to the mesh case of the Local Line Binary Pattern (LLBP) [43], introduced in the context of face

recognition to encode anisotropic information of neighbouring pixels by computing LBP across vertical and horizontal directions (Table II, case (a)). A variety of operators can be further derived from Eq. (9) by combining the different directional operators. Among these, the ORF framework can be used to arrange the facets according to a spiral-wise topology, thus allowing the derivation of the equivalent of the Archimedean spiral-like LBP, as originally defined in [44] (Table II, case (b)).

*Contour/Initial Seed:* Several LBP variants can be obtained by using a non-circular neighbourhood of the central facet, through a particular setting of the initial contour. For example, selecting the set of *Fin\_root* facets of Algorithm 3 in a bar-like shape fashion produces elongated ORFs. This pattern can be viewed as the mesh-LBP version of the elongated local binary pattern (ELBP) proposed in [18] (Table II, case (c)).

*Comparison Strategy:* This category includes the variants aiming to reduce the dimensionality of the LBP descriptor. The uniform patterns and the central symmetric mesh-LBP (mesh-CSLBP in Table II, case (d)) variants have been presented in Section III-A and experimented in Section IV. More recently, a dimensionality reduction method for LBP, denoted as *orthogonal combination* of LBP (OC-LBP) has been proposed in [45]. In this case, the basic idea is to first split the neighboring pixels of the original LBP operator into several non-overlapped orthogonal groups, then compute the LBP code separately for each group, and finally concatenate them together. The same computation procedure can be used in the mesh-LBP framework, resulting in an equivalent mesh-OC-LBP operator.

*Structural Element:* The *three-patch* and *four-patch* LBP (TPLBP and FPLBP, respectively) have been proposed by Wolf et al. [10] as an extension of the *center-symmetric* LBP (CSLBP) [29] for the purpose of extracting complementary information to pixel-based descriptors. In the mesh-LBP framework, we define a mesh-TPLBP like structure by constructing ORF patches of  $w$  rings at the central facet, and at  $m$  equally spaced facets on the  $r$ -th ring around the central facet. The case (e) of Table II depicts a mesh-TPLBP composed of ORF with 3-ring patches ( $w = 4$ ), one at the central facet and six at equally spaced positions on the 12-th ring (e.g.,  $r = 12$ ). Varying the parameters  $w$ ,  $r$  and  $m$  other mesh-TPLBP can be obtained as well. Formally, we express the mesh-TPLBP operator as follows:

$$\text{meshTPLBP}_{r,m,w}(f_c) = \sum_{k=0}^{m-1} s(Y) \cdot \alpha(k)$$

$$\text{with } Y = d(P_k, P_{f_c}) - d(P_{k+\delta \bmod m}, P_{f_c}),$$

where  $d(\cdot)$  is any distance function between two patches constructed on  $w$  rings (for example,  $d(\cdot)$  can be the  $L_2$  norm or the Bhattacharyya distance between the geometric histogram [39] associated to the two patches); and  $\delta$  controls the arc-length distance between the patches of a pair.

The FPLBP construction follows a similar approach to the three-patch solution, but considering four patches on two concentric rings (see Table II, case (f)). The construction of the mesh-FPLBP version of this operator follows virtually the same steps of the mesh-TPLBP, except that two groups, rather than one, of equally spaced ORF with  $w$ -rings are generated

at two different radii (e.g., the inner ring with radius  $r_1$ , and the outer ring with radius  $r_2$ ). The mesh-FPLBP operator is defined as follows:

$$\text{meshFPLBP}_{r_1,r_2,m,w}(f_c) = \sum_{k=0}^{m/2-1} s(Y) \cdot \alpha(k)$$

$$Y = d(P_k^1, P_{k+\delta \bmod m}^2) - d(P_{k+m/2}^1, P_{k+m/2+\delta \bmod m}^2).$$

Different variants of the mesh-FPLBP can be constructed by tuning the parameters  $r_1$ ,  $r_2$ ,  $m$ ,  $w$  and  $\delta$ . An example is shown in Table II, case (f), using  $r_1 = 5$ ,  $r_2 = 10$ ,  $m = 6$ ,  $w = 2$  and  $\delta = 0$ .

#### ACKNOWLEDGMENTS

Preliminary ideas and experiments of the mesh-LBP framework appeared in [46].

#### REFERENCES

- [1] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, Jan. 1996.
- [2] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
- [3] T. Mäenpää, J. Viertola, and M. Pietikäinen, "Optimising colour and texture features for real-time visual inspection," *Pattern Anal. Appl.*, vol. 6, no. 3, pp. 169–175, Mar. 2003.
- [4] L. Cao, J. Luo, F. Liang, and T. S. Huang, "Heterogeneous feature machines for visual recognition," in *Proc. 12th Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep./Oct. 2009, pp. 1095–1102.
- [5] D. Guo, V. Atluri, and N. Adam, "Texture-based remote-sensing image segmentation," in *Proc. IEEE Int. Conf. Multimedia Expo*, Amsterdam, The Netherlands, Jul. 2005, pp. 1472–1475.
- [6] C. Song, F. Yang, and P. Li, "Rotation invariant texture measured by local binary pattern for remote sensing image classification," in *Proc. IEEE 2nd Int. Workshop Edu. Technol. Comput. Sci.*, vol. 3. Wuhan, China, Mar. 2010, pp. 3–6.
- [7] A. Lucieer, A. Stein, and P. Fisher, "Multivariate texture-based segmentation of remotely sensed imagery for extraction of objects and their uncertainty," *Int. J. Remote Sens.*, vol. 26, no. 14, pp. 2917–2936, 2005.
- [8] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Proc. 8th Eur. Conf. Comput. Vis.*, Prague, Czech Republic, May 2004, pp. 469–481.
- [9] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.
- [10] L. Wolf, T. Hassner, and Y. Taigman, "Descriptor based methods in the wild," in *Proc. ECCV Workshop Faces Real-Life Images*, Marseille, France, Oct. 2008, pp. 1–14.
- [11] G. Zhang, X. Huang, S. Z. Li, Y. Wang, and X. Wu, "Boosting local binary pattern (LBP)-based face recognition," in *Proc. 5th Int. Workshop Adv. Biometric Pers. Authentication*, Beijing, China, Oct. 2005, pp. 179–186.
- [12] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on local binary patterns: A comprehensive study," *Image Vis. Comput.*, vol. 27, no. 6, pp. 803–816, May 2009.
- [13] L. Cai, C. Ge, Y.-M. Zhao, and X. Yang, "Fast tracking of object contour based on color and texture," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 7, pp. 1421–1438, Nov. 2009.
- [14] X. Wang and M. Mirmehdi, "Archive film restoration based on spatiotemporal random walks," in *Proc. 11th Eur. Conf. Comput. Vis.*, Crete, Greece, 2010, pp. 478–491.
- [15] G. Sandbach, S. Zafeiriou, and M. Pantic, "Local normal binary patterns for 3D facial action unit detection," in *Proc. 19th IEEE Int. Conf. Image Process.*, Orlando, FL, USA, Sep./Oct. 2012, pp. 1813–1816.
- [16] H. Li, L. Chen, D. Huang, Y. Wang, and J. Morvan, "3D facial expression recognition via multiple kernel learning of multi-scale local normal patterns," in *Proc. 1st Int. Conf. Pattern Recognit.*, Nov. 2012, pp. 2577–2580.
- [17] G. Sandbach, S. Zafeiriou, and M. Pantic, "Binary pattern analysis for 3D facial action unit detection," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Guildford, U.K., Sep. 2012, pp. 119.1–119.12.

- [18] S. Liao and A. C. S. Chung, "Face recognition by using elongated local binary patterns with average maximum distance gradient magnitude," in *Proc. 8th Asian Conf. Comput. Vis.*, Tokyo, Japan, Nov. 2007, pp. 672–679.
- [19] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Li, "Face detection based on multi-block LBP representation," in *Proc. Int. Conf. Biometrics*, Washington, DC, USA, Sep. 2007, pp. 11–18.
- [20] H. Jin, Q. Liu, H. Lu, and X. Tong, "Face detection using improved LBP under Bayesian framework," in *Proc. 1st Int. Conf. Image Graph.*, Hong Kong, Dec. 2004, pp. 306–309.
- [21] D. Huang, Y. Wang, and Y. Wang, "A robust method for near infrared face recognition based on extended local binary pattern," in *Proc. 3rd Int. Symp. Vis. Comput.*, Lake Tahoe, CA, USA, Nov. 2007, pp. 437–446.
- [22] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," in *Proc. 3rd Int. Workshop Anal. Modelling Faces Gestures*, Rio de Janeiro, Brazil, Oct. 2007, pp. 168–182.
- [23] T. Ahonen and M. Pietikäinen, "Soft histograms for local binary patterns," in *Proc. Finnish Signal Process. Symp.*, Oulu, Finland, Aug. 2007, pp. 1–4.
- [24] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen, *Computer Vision using Local Binary Patterns*. Berlin, Germany: Springer-Verlag, 2011.
- [25] S. Z. Li, C. Zhao, M. Ao, and Z. Lei, "Learning to fuse 3D+2D based face recognition at both feature and decision levels," in *Proc. 2nd Int. Workshop Anal. Modeling Faces Gestures*, Beijing, China, Oct. 2005, pp. 44–54.
- [26] Y. Huang, Y. Wang, and T. Tan, "Combining statistics of geometrical and correlative features for 3D face recognition," in *Proc. Brit. Mach. Vis. Conf.*, Edinburgh, U.K., Sep. 2006, pp. 879–888.
- [27] D. Huang, M. Ardabilian, Y. Wang, and L. Chen, "3D face recognition using eLBP-based facial description and local feature hybrid matching," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 5, pp. 1551–1565, Oct. 2012.
- [28] J. Fehr and H. Burkhardt, "3D rotation invariant local binary patterns," in *Proc. Int. Conf. Pattern Recognit.*, Tampa, FL, USA, Dec. 2008, pp. 1–4.
- [29] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognit.*, vol. 42, no. 3, pp. 425–436, Mar. 2009.
- [30] M. Pietikäinen, T. Ojala, and Z. Xu, "Rotation-invariant texture classification using feature distributions," *Pattern Recognit.*, vol. 33, no. 1, pp. 43–52, Jan. 2000.
- [31] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proc. 11th Eur. Conf. Comput. Vis.*, vol. 3. Crete, Greece, 2010, pp. 347–360.
- [32] T. Darom and Y. Keller, "Scale-invariant features for 3D mesh models," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2758–2769, May 2012. (2008). *MIT CSAIL Database*. [Online]. Available: <http://people.csail.mit.edu/tmertens/texttransfer/data/>
- [33] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, "A 3D facial expression database for facial behavior research," in *Proc. IEEE 7th Int. Conf. Autom. Face Gesture Recognit.*, Southampton, U.K., Apr. 2006, pp. 211–216.
- [34] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Numerical Geometry of Non-Rigid Shapes*. New York, NY, USA: Springer-Verlag, 2008.
- [35] A. S. Mian, M. Bennamoun, and R. Owens, "Keypoint detection and local feature matching for textured 3D face recognition," *Int. J. Comput. Vis.*, vol. 79, no. 1, pp. 1–12, Aug. 2008.
- [36] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton shape benchmark," in *Proc. Shape Modeling Int.*, Genoa, Italy, Jun. 2004, pp. 1–12.
- [37] A. Cerri *et al.*, "SHREC'13 track: Retrieval on textured 3D models," in *Proc. 6th Eurograph. Workshop 3D Object Retr.*, Girona, Spain, May 2013, pp. 73–80.
- [38] A. P. Ashbrook, R. B. Fisher, C. Robertson, and N. Werghi, "Finding surface correspondence for object recognition and registration using pairwise geometric histograms," in *Proc. 5th Eur. Conf. Comput. Vis.*, Freiburg im Breisgau, Germany, Jun. 1998, pp. 674–686.
- [39] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Trans. Graph.*, vol. 21, no. 4, pp. 807–832, Oct. 2002.
- [40] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.
- [41] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud, "Surface feature detection and description with applications to mesh matching," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 373–380.
- [42] A. Petpon and S. Srisuk, "Face recognition with local line binary pattern," in *Proc. 5th Int. Conf. Image Graph.*, Xi'an, China, Sep. 2009, pp. 533–539.
- [43] L. Nanni, A. Lumini, and A. Brahmam, "Local binary patterns variants as texture descriptors for medical image analysis," *Artif. Intell. Med.*, vol. 49, no. 2, pp. 117–125, Jun. 2010.
- [44] C. Zhu, C.-E. Bichot, and L. Chen, "Image region description using orthogonal combination of local binary patterns enhanced with color information," *Pattern Recognit.*, vol. 46, no. 7, pp. 1949–1963, Jul. 2013.
- [45] N. Werghi, S. Berretti, A. Del Bimbo, and P. Pala, "The mesh-LBP: Computing local binary patterns on discrete manifolds," in *Proc. IEEE Int. Workshop 3D Represent. Recognit.*, Sydney, Australia, Dec. 2013, pp. 562–569.



**Naoufel Werghi** (SM'14) received the Ph.D. degree in computer vision from the University of Strasbourg, Strasbourg, France. He has been a Research Fellow with the Division of Informatics, University of Edinburgh, Edinburgh, U.K., and a Lecturer with the Department of Computer Sciences, University of Glasgow, Glasgow, U.K. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Khalifa University, Abu Dhabi, United Arab Emirates. His main research area is

2D/3D image analysis and interpretation, where he has led several funded projects in the areas of biometrics, medical imaging, geometrical reverse engineering, and intelligent systems. He has been a Visiting Professor with the Department of Electrical and Computer Engineering, University of Louisville, Louisville, U.K., in 2002, and the Media Integration and Communication Center, University of Florence, Florence, Italy, in 2012. He has authored over 70 journal and conference papers.



**Stefano Berretti** received the Ph.D. degree in information and telecommunications engineering from the University of Florence, Florence, Italy, in 2001, where he is currently an Associate Professor with the Department of Information Engineering and the Media Integration and Communication Center. His current research interests focus on 3D object retrieval and partitioning, face recognition and facial expression recognition from 3D and 4D data, 3D face superresolution, and human action recognition from 3D data. He has been a Visiting Researcher

with IIT Bombay, Mumbai, India, and a Visiting Professor with the Institute Telecom, TELECOM Lille 1, Lille, France, and the Khalifa University, Abu Dhabi, United Arab Emirates. He has authored over 100 papers appeared in conference proceedings and international journals in the area of pattern recognition, computer vision, and multimedia. He is on the Program Committee of several international conferences and serves as a frequent reviewer of many international journals. He has been the Co-Chair of the Fifth Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment in 2012, held in conjunction with the European Conference on Computer Vision in 2012.



**Alberto del Bimbo** is currently a Full Professor of Computer Engineering, the Director of the Master in Multimedia, and the Director of the Media Integration and Communication Center with the University of Florence, Florence, Italy, where he was the Deputy Rector for Research and Innovation Transfer from 2000 to 2006. His scientific interests are multimedia information retrieval, pattern recognition, image and video analysis, and natural human-computer interaction. He has authored over 350 publications in some of the most distinguished

scientific journals and international conferences, and a monograph entitled *Visual Information Retrieval*. From 1996 to 2000, he was the President of the IAPR Italian Chapter and a Member-at-Large of the IEEE Publication Board from 1998 to 2000. He was the General Chair of the IAPR International Conference on Image Analysis and Processing in 1997 and the IEEE International Conference on Multimedia Computing and Systems in 1999, and the Program Co-Chair of the ACM Multimedia in 2008. He is the General Co-Chair of the ACM Multimedia in 2010 and the European Conference on Computer Vision in 2012. He is a fellow of the International Association for Pattern Recognition and an Associate Editor of *Multimedia Tools and Applications*, *Pattern Analysis and Applications*, the *Journal of Visual Languages and Computing*, and the *International Journal of Image and Video Processing*, and was an Associate Editor of *Pattern Recognition*, the *IEEE TRANSACTIONS ON MULTIMEDIA*, and the *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*.