Surface Feature Detection and Description with Applications to Mesh Matching

Andrei Zaharescu, Edmond Boyer, Kiran Varanasi, Radu Horaud

Surface Feature Detection and Description with Applications to Mesh Matching

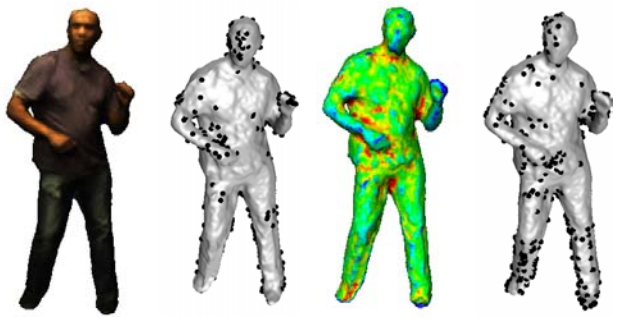
Andrei Zaharescu, Edmond Boyer, Kiran Varanasi and Radu Horaud
Perception Team, INRIA Grenoble Rhône-Alpes
655 Avenue de l'Europe, Montbonnot, 38 334 Saint Ismier Cedex, France
firstname.lastname@inria.fr

Abstract

In this paper we revisit local feature detectors/descriptors developed for 2D images and extend them to the more general framework of scalar fields defined on 2D manifolds. We provide methods and tools to detect and describe features on surfaces equiped with scalar functions, such as photometric information. This is motivated by the growing need for matching and tracking photometric surfaces over temporal sequences, due to recent advancements in multiple camera 3D reconstruction. We propose a 3D feature detector (MeshDOG) and a 3D feature descriptor (MeshHOG) for uniformly triangulated meshes, invariant to changes in rotation, translation, and scale. The descriptor is able to capture the local geometric and/or photometric properties in a succinct fashion. Moreover, the method is defined generically for any scalar function, e.g., local curvature. Results with matching rigid and non-rigid meshes demonstrate the interest of the proposed framework.

1. Introduction

The detection, characterization, and matching of various 2D or 3D features from visual observations is of great importance for a large variety of applications such as modeling, tracking, recognition or indexing, among others. The vast majority of existing methods detect features using either photometric information available with 2D images or geometric information available with 3D surfaces. However, recent progress in image based 3D modeling and rendering allows to recover both photometric and geometric information from multiple images [19]. Whenever such models are available, photometric 2D features or geometric 3D features, if taken separately, have limited informative capabilities with respect to the potential richness of the data. This is the case, for example, with deformable and/or articulated objects, since image appearance is only partially robust to motions and geometric properties alone are not always robust, e.g., the topology of the model can change considerably with varying object poses. Therefore, we believe that photometric and geometric information need to be handled in a consistent and simultaneous manner. To this purpose, we observe that photometric information available with 3D models can be viewed as scalar functions defined over 2D manifolds and, as such, represent a generalization of planar image domains to non-planar domains. We can thus build on the existing image feature extraction theories and investigate their extensions to 2D manifolds.



(a) (b) (c) (d)

Figure 1. The feature detection method described in this paper can be applied to any scalar function defined over a 2D manifold such as the meshed surface shown here: photometric data (a) and associated points of interest (b); mean surface curvature (c) and the detected features (d).

The contribution of this paper is twofold: first we develop a methodology for feature-based characterization using operators acting on scalar functions defined over 2D manifolds; second, we derive a novel family of interest point detectors and descriptors that take into account both the surface geometry and the photometric information. To this aim, operators such as the *discrete convolution* and the *discrete gradient*, are defined for scalar functions on discrete surfaces, i.e., meshes, thus taking into account both the functions' differential properties as well as the surfaces' intrinsic geometry. Based on these operators, a new inter-

est point detector and a new local descriptor are introduced, namely MeshDOG and MeshHOG. MeshDOG is a generalization of the DOG operator [14, 13] and it seeks the extrema of the Laplacian of a scale-space representation of any scalar function defined on a discrete manifold. MeshHOG is a generalization of the histogram of oriented gradients (HOG) descriptor recently introduced for describing 2D images [3]. The new descriptor is defined with respect to the measurements available at each of the discrete surface's vertices and it can work with features photometric features, as well as with geometric feature, such as curvature, geodesic integral, etc.

As it is the case with the more classical image operators, detectors and descriptors are not uniquely defined over surfaces and MeshDOG and MeshHOG were chosen in light of their quasi-invariance to transformations such as rotation and scale. In addition, they exhibit a number of attractive properties: (**i**) there are no perspective distortions, since computations are achieved in 3D; (**ii**) there are no false detections due to occlusions; (**iii**) the descriptor captures both the local 3D geometry and the local gradient information of the scalar function; (**iv**) no planar mesh embedding is necessary; (**v**) within a multiple-camera setting, the descriptor can fuse the photometric information coming from different images in order to provide more robust image-invariant photometric information.

The organization of the paper is as follows. Section 2 discusses related works. Section 3 describes the mathematical formulation used to to build a number of operators on discrete manifolds. Section 4 and 5 introduce the local feature detector and descriptor, respectively. Section 6 presents and discusses the results, before concluding in section 7.

2. Related Work

Photometric functions over planar domains (local image features): Developing robust 2D features, invariant under changes in illumination, viewpoint, scale and orientation has been one of the long term research goals in the area. Currently, SIFT [13] and HOG (histogram of oriented gradients) [3] are among the most widely used descriptors for their robustness to the transformations just cited. Interest points may coincide to the extrema of the Laplacian of the photometric function, and they are detected at various resolution scales using the difference of Gaussians (DOG) approximation of the Laplacian, see [15] for a detailed review. Alternatively, spatio-temporal descriptors have also been proposed [24, 9], by considering the 3D spatio-temporal volume defined by a short image sequence over time. Such space-time features can be seen as local features defined over 3D grids. We extend the DOG operator to non-planar surfaces instead of dealing with volumetric grids.

Geometric functions over surfaces (local geometric features): 3D spin images [8] and 3D shape contexts [11, 5] are among the most successful surface descriptors. These are descriptors that rely solely on the surface geometry. See [22, 2] for a detailed survey. Typically these descriptors characterize the neighbourhood of a specified surface region. A number of methods have been proposed for automatic identification of interest regions on surfaces, taking into account geometrical features. Scale-space extrema based on the averaged mean curvature flow are proposed in [18]. Alternatively, [16] defines the scale space in a planar parametrization of the surface using the normal map and searches for the extrema. Gradient operators are defined over a planar vector field. While this formulation could be used as an alternative mathematical framework in current work, the required planar parameterization introduces an additional level of complexity that the currently proposed method avoids. [12] proposes a mesh saliency method, based on the center-surround operator, adapted from the visual attention literature. Photometric information is not taken into account by these methods.

Photometric functions over surfaces (local augmented surface features): In [25] a SIFT-based descriptor on 3D oriented patches is proposed, i.e., VIP (Viewpoint Invariant Patches), which was used for 3D model matching. It constitutes a first attempt to devise a descriptor that includes both geometry (normal orientation) and photometric information. In [21] the authors propose a concatenated surface descriptor taking into account both geometry (a region descriptor based on geodesic-intensity histograms), and photometric information (edge and corner descriptors that take into account the local isometric mapping to $\mathbb{R}^2$). The approach proposed in this paper is similar in spirit to [25], but, instead, considers full 3D gradients and histograms.

Many applications make use of local features, in particular in the context of surfaces: surface registration, non-rigid shape matching and object recognition. For instance [17] proposes an image-based descriptor using the local $\mathbb{R}^2$ embedding of the normal information on the mesh in order to perform surface registration. Also, a recent number of works, e.g. [6, 1, 4, 23], address the non-rigid mesh matching problem using observations from multiple views. The vast majority of the proposed methods (the only notable exception being [6]) uses both geometric information extracted from surfaces and photometric data available with images. The latter is first extracted using 2D image descriptors (such as SIFT [13]), and subsequently backprojected onto the mesh. This sparse description is generally used to bootstrap dense matching. Surface descriptors may well be used for 3D object recognition, as it has been already done in [20] using the Princeton shape benchmarking database [1].

[1] http://shape.cs.princeton.edu/benchmark/

Our work contributes to these efforts by taking a different, yet complimentary approach, namely image-feature detection and description methodologies are extended to features defined onto 2D manifolds.

3. Problem formulation

Let $\mathbb{S}$ denote the set of all possible discrete parametrizations of the admissible 2D manifolds in $\mathbb{R}^3$. We will consider in particular *uniformly sampled triangulated meshes* $S \in \mathbb{S}$, namely meshes whose facets are triangles of approximately the same area and whose vertices' valence is close to 6. We notice that such an uniform mesh can be obtained from a non-uniform mesh through simple mesh operations, as proposed in [10]. This absolves us of the necessity of complex techniques that ensure proper samplings of scalar fields over S, while keeping generality. It is interesting to notice that an image can be viewed as a "flat" uniformly sampled mesh, i.e., a grid of vertices with valence 4 and whose facets are squares or rectangles.

S can also be viewed as a graph $S(V, E)$, where $V = \{v_i\}_{1 \leq i \leq N}$ is the set of mesh vertices and $E = \{e_{ij}\}$ is the set of mesh edges between adjacent vertices. We denote by e_{avg} the average edge length. We associate a 3D point $\mathbf{v} \in \mathbb{R}^3$ with each vertex v. The ring of a vertex $rg(v, n)$ is the set of vertices that are at distance n from v on S, where the distance n is the minimum number of edges between two vertices. Thus $rg(v, 0)$ is v itself and $rg(v, 1)$ is the set of direct neighbours of v (see Figure 2). The neighbourhood $N_n(v)$ is then the set of rings $\{rg(v, i)\}_{0 \leq i \leq n}$. We further denote $\overrightarrow{\mathbf{n}}_v$ the unit vector normal to the surface S at vertex v, computed as the average direction of the normals of the triangles incident to v.

We consider a scalar function $f : \mathbb{S} \rightarrow \mathbb{R}$. In order to be able to estimate discrete gradient information, we first recall the definition of the directional derivative of a scalar function on a manifold [7]:

Definition 1 (Directional Derivative) *Let $\nabla_S f$ denote the gradient operator of f on S, the directional derivative of f at $v \in S$ is defined as:*

$$D_{\overrightarrow{u}} f(\mathbf{v}) = \nabla_S f(\mathbf{v}) \cdot \overrightarrow{u}, \tag{1}$$

for any direction $\overrightarrow{u}$ in the tangent plane of S at v.

Using the fact that up to first order: $f(\mathbf{v}_j) - f(\mathbf{v}_i) = \nabla_S f(\mathbf{v}_i) \cdot (\mathbf{v}_j - \mathbf{v}_i)$ around v_i, we have the following definition:

Definition 2 (Discrete Directional Derivative) *The discrete directional derivative of f is defined as:*

$$D_{\overrightarrow{e_{ij}}} f(\mathbf{v}_i) = \frac{1}{||\overrightarrow{v_i v_j}||}(f(\mathbf{v}_j) - f(\mathbf{v}_i)), \tag{2}$$

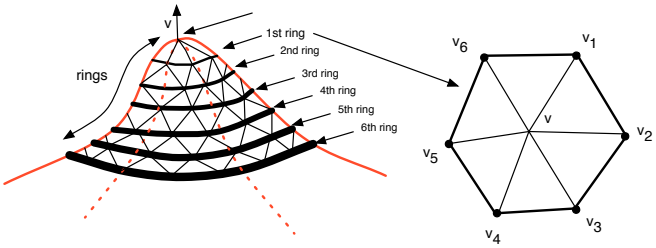


Figure 2. A vertex v and its rings (left) and the first ring of v (right).

$\forall e_{ij} \in E$ *and where* $||\overrightarrow{v_i v_j}|| = ||\mathbf{v}_j - \mathbf{v}_i||$.

$\nabla_S f(\mathbf{v}_i)$ is by definition a vector in the tangent plane at v_i and the above definition allows us to estimate its directional values around v_i. Hence, two such non-null local directional gradients are, in principle, sufficient to estimate the gradient $\nabla_S f(\mathbf{v}_i)$ at v_i. This is a generalization of the classical way of computing gradients in the image using two orthogonal directions. In practice however, we prefer to use all the directional gradients provided by the first ring of a vertex: indeed, this redundancy guarantees a more robust operator:

Definition 3 (Discrete Gradient) *the gradient operator $\nabla_S f(\mathbf{v}_i)$ of f at $v_i \in S$ is defined as:*

$$\nabla_S f(\mathbf{v}_i) = \sum_{v_j \in rg(v_i, 1)} (w_{ij} D_{\overrightarrow{e_{ij}}} f(\mathbf{v}_i))\, \overrightarrow{u_{ij}}, \tag{3}$$

where w_{ij} weighs the contribution of $D_{\overrightarrow{e_{ij}}}$ and $\overrightarrow{u_{ij}}$ is the normalized projected direction of $\overrightarrow{v_i v_j}$ in the tangent plane at v_i.

The weights w_{ij} should be chosen in order to balance the contributions of the local directional derivatives with respect to their associated directions in the tangent plane. The gradient is defined as a weighted mean of directional derivatives, since directional derivatives are projections of the gradient onto given directions. Assuming that S is uniformly sampled and thus that neighbours around v_i are equally spaced we get: $w_{ij} = \frac{1}{val(v_i)}$ where $val(v_i)$ is the valence of v_i. For non uniformly sampled meshes, the weights are a function of the angles between the directions $\overrightarrow{u_{ij}}$ around v_i in the tangent plane at v_i.

Finally, we define the discrete convolution operator on a mesh:

Definition 4 (Discrete Convolution). *The convolution of the function f with a kernel k is:*

$$(f * k)(v_i) = \frac{1}{K} \sum_{v_j \in N_n(v_i)} k(||\overrightarrow{v_i v_j}||) f(\mathbf{v}_j), \tag{4}$$

where the kernel weighs the participation of neighbouring vertices v_j as a function of their distances from vertex v_i and $K = \sum_{v_j \in N_n(v_i)} k(||\overrightarrow{v_i v_j}||)$ is a normalization factor. Notice that, as for the discrete gradient, we assume a uniformly sampled mesh and thus that contributions of neighbouring vertices v_j in the above expression are equally weighted with respect to their spatial arrangements. Another remark is that, generally, we use the above definition with the first ring only, i.e., $n = 1$.

4. Feature Detection (MeshDOG)

Feature detection is comprised of three steps, as illustrated in Figure 3. First, the extrema of the function's Laplacian (DOG) are found across scales using a one-ring neighbourhood. Second, the extrema thus detected are thresholded. Third, the unstable extrema are eliminated, thus retaining those mesh locations exhibiting some degree of cornerness.

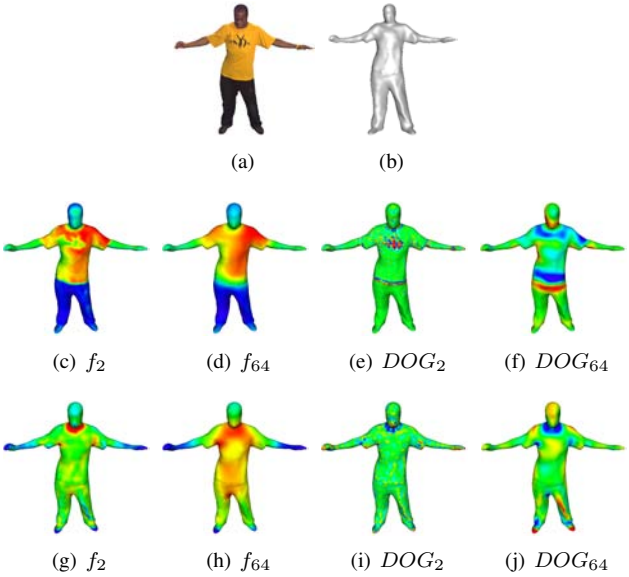
Scale-space extrema. We propose a scale-space representation of scalar function f defined on a mesh. We consider the convolution operation on meshes (see Definition 4) using a Gaussian kernel, defined as:

$$g_\sigma(x) = \frac{\exp\left(-x^2/2\sigma^2\right)}{\sigma\sqrt{2\pi}}.$$

The scale space of f is built progressively: $f_0 = f$, $f_1 = f_0 * g_\sigma$, $f_2 = f_1 * g_\sigma$, etc. Convolved functions are subtracted, e.g., $DOG_1 = f_1 - f_0$, $DOG_2 = f_2 - f_1$, etc., in order to obtain the difference of Gaussian operator. An example can be observed in Figure 4, where the model used is frame 30 from *pop2lock* sequence from the University of Surrey, and the features being shown are colour and mean curvature. An important observation is that, *when building the scale space, the mesh geometry does not change*, but the different scalar functions defined on the mesh, i.e. f_1, f_2, DOG_1, DOG_2. We have chosen $\sigma = 2^{\frac{1}{3}} e_{avg}$ and have performed 93 convolutions.

(a) (b) (c) (d)

Figure 3. Feature detection shown with photometric data. (a) Original mesh (27240 vertices); (b) Scale-space extrema (5760 vertices left); (c) Thresholding (1360 vertices left); (d) Corner detection (650 vertices left).



(a) (b) (c) f_2 (d) f_{64} (e) DOG_2 (f) DOG_{64} (g) f_2 (h) f_{64} (i) DOG_2 (j) DOG_{64}

Figure 4. An example of processing (a) photometry and (b) mean curvature. Scale space photometric representation (c)-(f) and scale space representation of mean curvature (e)-(j).

The feature points are selected as the maxima of the scale space across scales, followed by non-maximum-suppression, using the one ring neighbourhood, in the current and in the adjacent scales.

Thresholding. From the extrema of the scale space, only the top $\beta = 5\%$ of the maximum number of vertices are being considered, sorted by magnitude. We have chosen a percentage value versus a hard value threshold in order to keep the detector flexible, no matter which feature is being considered, without the need for normalization.

Corner Detection. Additionally, in order to eliminate more non-stable responses, we retain the features that exhibit corner characteristics. As proposed in [13] this can be done using the Hessian operator: :

$$\mathbf{H}(v) = \begin{bmatrix} d_{xx}(v) & d_{xy}(v) \\ d_{yx}(v) & d_{yy}(v) \end{bmatrix}, \tag{5}$$

where d_{xx}, d_{xy} and d_{yy} are second partial derivatives. We estimate them by applying the definition of directional derivatives (1) twice, e.g. $d_{xy} = \nabla_S D_{\overrightarrow{x}} f(\mathbf{v}) \cdot \overrightarrow{y}$, where the gradient is computed using (3). The directions $\overrightarrow{x}$ and $\overrightarrow{y}$ represent here a local coordinate system in the tangent plane of v, typically the gradient direction for $\overrightarrow{x}$ and its orthogonal direction for $\overrightarrow{y}$. The ratio between the largest λ_{max} and the lowest λ_{min} eigenvalues of the Hessian matrix is a good indication of a corner response, which is independent of the local coordinate frame. We typically use $\lambda_{max}/\lambda_{min} = 10$ as a minimum value to threshold responses.

5. Feature Descriptor (MeshHOG)

The descriptor $\mathbf{t}_v$ for vertex v is computed using a support region, defined using a neighbourhood ring size r, as depicted in Figure 2. For each vertex from the neighbourhood $v_i \in N_r(v)$, the gradient information $\nabla_S f(v_i)$ is computed using (3). As a first step, a local coordinate system is chosen, in order to make the descriptor invariant to rotation. Then, a histogram of gradient is computed, both spatially, at a coarse level, in order to maintain a certain high-level spatial ordering, and using orientations, at a finer level. Since the gradient vectors are 3 dimensional, the histograms are computed in 3D.

Neighborhood size. The number of rings r for the support region is chosen adaptively based on a more global measure, such that the descriptor is robust to different spatial samplings and to scaling. The value of r is chosen such that it covers a proportion α_r from the the total mesh surface, where $\alpha_r \in (0, 1)$. By denoting A_S as the total surface area of the mesh S, which can be computed as the sum of all triangle areas, the ring size r is:

$$r = round\left(\frac{1}{e_{avg}}\sqrt{\frac{\alpha_r A_S}{\Pi}}\right), \tag{6}$$

assuming that the surface covering the ring neighbourhood can be approximated with a circle and that the mesh S is equally sampled, with the average edge size e_{avg}. In practice, we use an r corresponding to $\alpha_r = 1\%$.

Local Coordinate System. A local coordinate system can be devised using the normal $\overrightarrow{\mathbf{n}}_v$ and two other unit vectors, residing in tangent plane $\mathcal{P}_v$ of v. Given a unit vector $\overrightarrow{\mathbf{a}}_v \in \mathcal{P}_v$, the local coordinate system is given by $\{\overrightarrow{\mathbf{a}}_v, \overrightarrow{\mathbf{n}}_v, \overrightarrow{\mathbf{a}}_v \times \overrightarrow{\mathbf{n}}_v\}$. Vector $\overrightarrow{\mathbf{a}}_v$ is computed as the direction associated to the dominant bin in a polar histogram, with $b_a = 36$ bins. The histogram is computed by considering the projected vertices v_i in $\mathcal{P}_v$ and taking into account their gradient magnitudes. We weigh $||\nabla_S f(v_i)||$ by a Gaussian with $\sigma = e_{avg} r/2$, based on the geodesic distance from v. In order to reduce aliasing and boundary effects of binning, votes are interpolated bilinearly between neighbouring bins when computing the histograms. We use the same weighting and interpolation technique for any further binning.

Histograms. Instead of computing full 3D orientation histograms, as proposed in [9], we project the gradient vectors to the 3 orthonormal planes, describing the local coordinate system. This provides us with a more compact representation of the descriptor. For each of the three planes, we compute a 2 level histogram. Firstly, the plane is divided in $b_s = 4$ polar slices, starting with an origin and continuing in the direction dictated by the right hand rule with respect to the other orthonormal axis vector. When projected onto the plane, each vertex v_i will fall within one of the spatial slices. For each spatial slice, we compute orientation histograms with $b_o = 8$ bins for each of the projected gradient vectors $\nabla_S f(v_i)$ of the vertices v_i that projected onto that spatial slice, as shown in Figure 5(d).

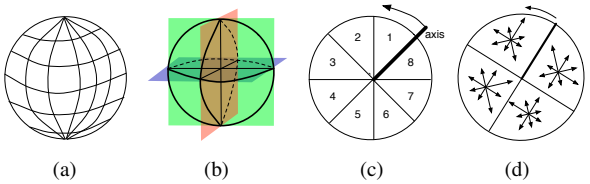


Figure 5. a) 3D Histogram - polar mapping used for creating histograms via binning of 3D vectors; b) Choosing 3 orthogonal planes onto which to project the 3D Histogram. c) Polar Coordinate system used for creating histograms via binning of 2D vectors, shown in this example with 8 polar slices. d) Example of a typical spatial and orientation histograms, using 4 spatial polar slices and 8 orientation slices.

Descriptor. The final descriptor is obtained by concatenating $b_s \times b_o$ histogram values for each of the three planes, followed by L-2 normalization.

6. Mesh Matching

We are validating the proposed detector and descriptor using a mesh matching approach. Let us consider two meshes S_1 and S_2 of the same object. The two meshes do not necessarily have the same number of vertices. Using the proposed approach, n_1 interest points are detected on S_1, which are characterised by descriptors $\mathbf{t}_i^1$, with $i \in [1..n_1]$. Similarly, n_2 interest points are detected on S_2, characterised by descriptors $\mathbf{t}_j^2$, with $j \in [1..n_2]$.

Matching. We use an intuitive greedy heuristic in order to select the a set of best matches. For each descriptor $\mathbf{t}_i^1$ from surface S_1, we find the best matching descriptor $\mathbf{t}_j^2$ from surface S_2 in terms of the Euclidean distance $d_{ij} = ||\mathbf{t}_i^1 - \mathbf{t}_j^2||$. We perform cross validation, by checking that $\mathbf{t}_j^2$'s best match is indeed $\mathbf{t}_i^1$. Finally, we only accept the candidate match is the second best match is significantly worse ($\gamma = 0.7$ or less from the best match score). This is not meant to fully solve the matching problem, as would a global approach [21]. It is merely intended for validation and for evaluation of our detector and descriptor.

Datasets. In our evaluation we consider the following scenarios: (i) the two meshes are representations of the same rigid object, which can thus be aligned using a rotation, translation and scale; (ii) the two shapes are representations of the same non-rigid object, i.e. a moving person. In this context, we are introducing the datasets.

- Matching rigid objects: we are considering reconstructions of the same object using different camera sets. In particular, we are using meshes obtained employing the method described in [27], using the publicly available datasets from the Middleburry Multi-View Stereo site [19]. The *Dino* datasets contains two meshes, one with 27,240 vertices obtained from 16 cameras and the other of 31,268 vertices generated from 47 cameras. Similarly, the *Temple* datasets contains two meshes, one with 78,019 vertices obtained from 16 cameras and the other of 80,981 vertices generated from 47 cameras.

- Matching non-rigid objects from synthetic data: we consider a synthetically generated dataset entitled *Synth-Dance* of a human mesh with 7,061 vertices moving across 200 frames.

- Matching non-rigid objects from real data: additionally, we use frames 515-550 from the INRIA *Dance-1* sequence [2], where the same reconstruction method [27] was employed to recover models using 32 cameras. The models have vertices ranging between 16,212 and 18,332.

Photometric information. The colour of each vertex of the surface is computed by considering the median colour in the visible images. We assume that the colours of a vertex follow a non-Gaussian distribution, due to errors that can occur around occluding contours. In the *Synth-Dance* dataset the vertices are randomly coloured.

6.1. Examples of Matching Rigid Objects

We present our results on the *Dino* and *Temple* datasets in Figure 6, where we have run tests where the colour and the mean curvature were used as features, as well as cases in which we have created a new descriptor by concatenating the MeshHOG descriptors for colour and mean curvature. The results are interesting. Even when just curvature is used for the descriptor, there seems to be enough discriminability to account for a number of correct matches varying between 10-30, depending on the detector and the dataset. Both the *Dino* and the *Temple* datasets are rather challenging, due to the fact that, at a first glance, they do not have a large number of distinguishing non-repetitive features in terms of their visual aspect. Additionally, it seems that using just the colour as a feature provides the best results in terms of the number of matches. This is so, we can argue, because the descriptor inherently incorporates certain mesh geometry information by design of the operators.

[2] https://charibdis.inrialpes.fr/

These are the only results presented in the paper where different features were used for the descriptor. All the other results are generated using colour information.

6.2. Examples of Matching Non-Rigid Objects

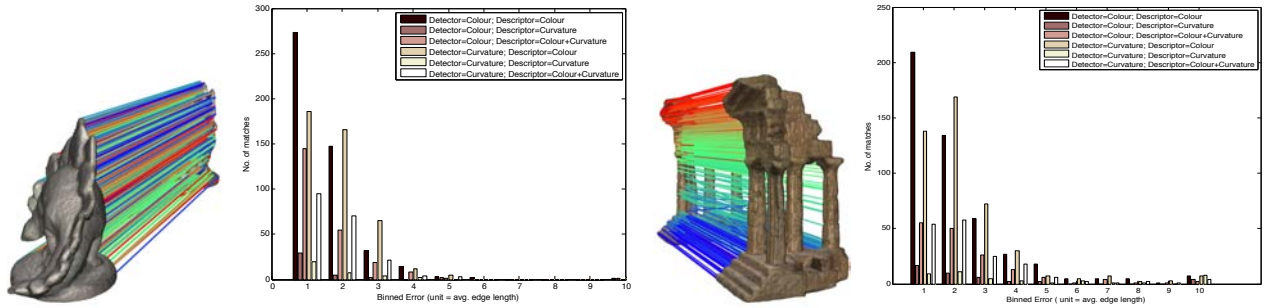
Comparison with back-projected 2D features. We present a comparison between the proposed mesh matching framework using MeshHOG descriptor with another framework, currently employed in a number of mesh matching methods (see Section 2), that uses back-projected image descriptors. In the image based framework, the matching is performed in the images and only then is back-projected onto the surface. In our comparisons, we used the SIFT image descriptor. When matching the two surfaces, only matches from the same cameras are considered. In order to be able to carry such a comparison for the *Synth-Dance* dataset, we have generated images for 16 virtual cameras, distributed in a circular pattern around the object.

Synthetic comparative results are presented in Figure 7. The mesh in the first frame was matched with the mesh at any of the other 199 frames across the sequence. As it can be observed, the MeshHOG descriptor generates very few false positives in comparison with the SIFT equivalent, clearly demonstrating the advantages of the proposed approach.

In addition, we present empirical results in Figure 8 for for the INRIA *Dance-1* sequence. As it can be observed, the second best match ratio threshold $\gamma = 0.7$ tends to be more aggressive for SIFT. There are only 54 matched found using the SIFT back-projected method between frame 525 and 526, whereas MeshHOG finds 119 matches. Even when matching across distant frames (530 and 550), our proposed method finds 13 correct matches, versus the SIFT descriptor, that fails. It is to be expected, since most of the inter-frame matches are due to local creases formed by the clothes.The head is the only unique feature that can be robustly matched across time.

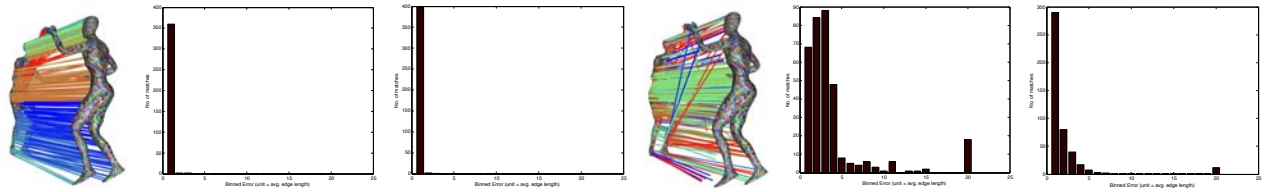
6.3. Resilience to Noise

There are two kinds of uniformly distributed noise being applied: geometry noise (changing the vertices v) and colour noise (changing values $f(v)$ held in each vertex). The colour noise relates to % of the total amount of a maximum 255 RGB value noise, whereas the geometry noise relates to the % of the total amount of a maximum e_{avg} noise level. As it can be observed in Figure 9, the method does not generate more false positives when the amount of noise increases. The *Dino* dataset has a larger number of false positives, since the two meshes are not perfectly identical, being the result of a 3D reconstruction method from multiple



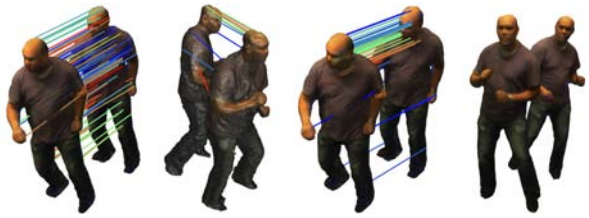
(a) Dino - Matches (b) Dino - Errors (c) Temple - Matches (d) Temple - Errors

Figure 6. "Rigid" matching results - *Dino* and *Temple* datasets. (a) (c) Matching results when using the colour both as a detector and as a feature; (b) (d) Error distribution when using different combinations of features for both detection and matching.



(a) MeshHOG (b) MeshHOG (c) MeshHOG (d) SIFT (e) SIFT (f) SIFT

Figure 7. "Non-rigid" matching using synthetic data - *dancer-synth* dataset. Comparison between MeshHOG and SIFT matching results. Matches between frames 1 and 50 are visually depicted in (a),(d). There are 364 matches for MeshHOG and 343 matches for SIFT. They are also quantified in the error histograms (b),(e). The histogram bins are of size equal to e_{avg}. The last bin groups all the errors greater than $20 * e_{avg}$. Additionally, the average histogram errors are shown in (c),(f) for matching frame 1 with x, where $x \in [2..200]$.



(a) MeshHOG (b) MeshHOG (c) SIFT (d) SIFT

Figure 8. "Non-rigid" matching using real data - *Dance-1* sequence. (a) Matches between frames 525 and 526 using MeshHOG (119 matches); (b) Matches between frames 530 and 550 using MeshHOG (13 matches); (c) Matches between frames 525 and 526 using SIFT (54 matches); (d) Matches between frames 530 and 550 using SIFT (0 matches).

images, which introduces some errors. In the *Synth-dance* dataset, the colour noise influences the descriptor accuracy more than the geometry noise, whereas in the *Dino* dataset the situation is reversed. This stems from the fact that the meshes in the two datasets have a relatively different number of vertices, which will in turn directly influence the ring neighbourhood size r ($r = 7$ for *Synth-dance*, and $r = 15$ and $r = 16$ for *Dino*), always chosen to represent α_r of the total mesh area.

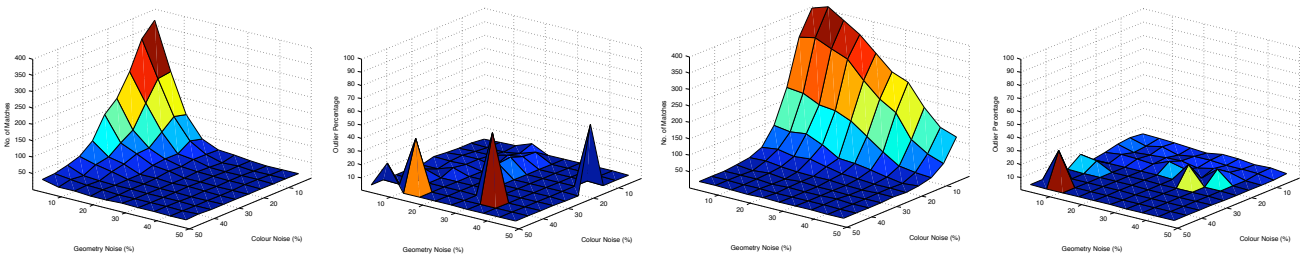
Integration with mesh tracking. We have integrated the MeshHOG descriptor within an existing mesh tracking approach, described in [23], by replacing the sparse matching step based on back projected SURF descriptors with the currently introduced descriptor. For more details, see [26].

The running time of computing such a descriptor depends on the descriptor neighbourhood size. For example, in the *synth-dance* dataset, computing 706 descriptors using a neighbourhood size $r = 7$ took under 1 second, while computing 2724 descriptors using a ring neighbourhood size $r = 15$ took 35 seconds. The machine used for the test was a Core2Duo 2.4GHz Intel with 2 Gigs of RAM running Mac OS.X. The code has been developed in C++ and it is available for download from [3].

7. Conclusion

We have introduced MeshDOG and MeshHOG, a new 3D interest point detector and a new 3D descriptor, defined on uniformly sampled triangular meshes. The descriptor is able to capture the local geometric and/or photometric properties in a succinct fashion. It is robust to changes in orientation, rotation, translation and scale. We have presented results of matching various rigid and non rigid datasets, both on real sequences and on synthetically generated data. They demonstrate that local features detected on meshes using

[3] http://perception.inrialpes.fr/



(a) Synth - TP (b) Synth - FP Ratio (c) Dino - TP (d) Dino - FP Ratio

Figure 9. Resilience to noise. Two kinds of noise are being applied: geometry noise (changing the vertices v) and colour noise (changing values $f(v)$ held in each vertex). a) *Synth-dance* dataset (frame 1 and 50) - True Positive (TP); b) *Synth-dance* dataset (frame 1 and 50) - False Positive (FP) Ratio; c) *Dino* dataset - True Positives (TP); d) *Dino* dataset - False Positive (FP) Ratio.

both photometric and geometric information are more robust than traditional purely photometric features detected in images.

Acknowledgements. This research was supported by the VISIONTRAIN RTN-CT-2004-005439 Marie Curie Action within the European Community's Sixth Framework Programme.

References

[1] N. Ahmed, C. Theobalt, C. Rossl, S. Thrun, and H.-P. Seidel. Dense correspondence finding for parametrization-free animation reconstruction from video. In *CVPR*, 2008.

[2] B. Bustos, D. A. Keim, D. Saupe, T. Schreck, and D. V. Vranic. Feature-based similarity search in 3D object databases. *ACM Computing Surveys*, 34(4):345–387, 2005.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[4] E. de Aguilar, C. Theobalt, C. Stoll, and H.-P. Seidel. Marker-less 3D feature tracking for mesh-based human motion capture. In *Human Motion – Understanding, Modeling, Capture and Animation*, 2007.

[5] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *ECCV*, 2004.

[6] Y. Furukawa and J. Ponce. Dense 3D motion capture from synchronized video streams. In *CVPR*, 2008.

[7] A. N. Hirani. *Discrete Exterior Calculus*. PhD thesis, California Institute of Technology, 2003.

[8] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *PAMI*, 21(5):433–449, 1999.

[9] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *BMVC*, 2008.

[10] L. Kobbelt, T. Bareuther, and H.-P. Seidel. Multiresolution shape deformations for meshes with dynamic vertex connectivity. In *Eurographics*, 2000.

[11] M. Körtgen, G.-J. Park, M. Novotny, and R. Klein. 3D shape matching with 3D shape contexts. *Central European Seminar on Computer Graphics*, 2003.

[12] C. H. Lee, A. Varshney, and D. Jacobs. Mesh saliency. *SIGGRAPH*, 2005.

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[14] D. Marr and E. Hildreth. Theory of edge detection. In *Proc. Roy. Soc. London*, volume B 207, pages 187–217, 1980.

[15] K. Mikolajczyk and C. Schmidt. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, 2005.

[16] J. Novatnack and K. Nishino. Scale-dependent 3D geometric features. In *ICCV*, 2007.

[17] J. Novatnack and K. Nishino. Scale-dependent/invariant local 3D shape descriptors for fully automatic registration of multiple sets of range images. In *ECCV*, 2008.

[18] M. Schlattmann, P. Degener, and R. Klein. Scale space based feature point detection on surfaces. *Journal of WSCG*, 16(1-3), February 2008.

[19] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006.

[20] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. In *Shape Modeling International*, 2008.

[21] J. Starck and A. Hilton. Correspondence labelling for wide-time free-form surface matching. In *ICCV*, 2007.

[22] J. W. H. Tangelder and R. C. Veltkamp. A survey of content based 3D shape retrieval methods. *Shape Modeling International*, 2004.

[23] K. Varanasi, A. Zaharescu, E. Boyer, and R. P. Horaud. Temporal surface tracking using mesh evolution. In *ECCV*, 2008.

[24] S.-F. Wong and R. Cipolla. Extracting Spatiotemporal Interest Points using Global Information. In *ICCV*, 2007.

[25] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys. 3D model matching with viewpoint invariant patches (vips). In *CVPR*, 2008.

[26] A. Zaharescu. *Constributions to Spatial and Temporal 3-D Reconstruction from Multiple Cameras*. PhD thesis, INPG, 2008.

[27] A. Zaharescu, E. Boyer, and R. P. Horaud. Transformesh: a topology-adaptive mesh-based approach to surface evolution. In *ACCV*, 2007.