



ELSEVIER

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Euclidean-distance-based canonical forms for non-rigid 3D shape retrieval

David Pickup^{*}, Xianfang Sun¹, Paul L. Rosin², Ralph R. Martin³

School of Computer Science and Informatics, Cardiff University, Queen's Buildings, 5 The Parade, Roath, Cardiff CF24 3AA, United Kingdom

ARTICLE INFO

Article history:

Received 11 November 2013

Received in revised form

28 November 2014

Accepted 21 February 2015

Available online 2 March 2015

Keywords:

Shape retrieval

Canonical forms

ABSTRACT

Retrieval of 3D shapes is a challenging problem, especially for non-rigid shapes. One approach giving favourable results uses multidimensional scaling (MDS) to compute a canonical form for each mesh, after which rigid shape matching can be applied. However, a drawback of this method is that it requires geodesic distances to be computed between all pairs of mesh vertices. Due to the super-quadratic computational complexity, canonical forms can only be computed for low-resolution meshes. We suggest a linear time complexity method for computing a canonical form, using Euclidean distances between pairs of a small subset of vertices. This approach has comparable retrieval accuracy but lower time complexity than using global geodesic distances, allowing it to be used on higher resolution meshes, or for more meshes to be considered within a time budget.

© 2015 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Content-based 3D shape retrieval is a key research topic, as the large and ever increasing number of available 3D models makes effectively searching for models with a desired shape an increasingly important task. In the case of non-rigid models, where similar objects may appear in many different poses, the task of accurately comparing shapes is an especially challenging problem. One of the most effective approaches to solving this non-rigid retrieval problem is to convert each shape to a *canonical form*, i.e. a standard pose, and perform rigid shape retrieval on those [1,2]. Lian et al. [3] gave a method which computes a canonical form (see Fig. 1) for a mesh using the method of Elad and Kimmel [4] to map the geodesic distances between every pair of points on the surface to 3D Euclidean distances using multidimensional scaling (MDS). A view-based method is then used to perform shape retrieval. The drawback of this method is the high, *super-quadratic*, computational cost of geodesic distance computation, which requires the models to be simplified to approximately 2000 vertices to achieve a reasonable run-time.

^{*} Corresponding author. Tel.: +44 29 20870389.

E-mail addresses: PickupD@cardiff.ac.uk (D. Pickup), SunX2@cardiff.ac.uk (X. Sun), RosinPL@cardiff.ac.uk (P.L. Rosin), MartinRR@cardiff.ac.uk (R.R. Martin).

¹ Tel.: +44 29 20879355.

² Tel.: +44 29 20875585.

³ Tel.: +44 29 20875536.

Our method computes a canonical form without the need for geodesic distances. Instead of mapping geodesic distances to Euclidean distances, we instead maximise the Euclidean distances between a subset of feature points while attempting to preserve the original mesh edge lengths. These feature points are selected based upon the conformal factor of the vertices [5]. This is a pose-invariant measure that represents the amount of local work required to globally transform the mesh into a sphere.

We select \sqrt{N} feature points, where N is the number of mesh vertices, which allows the distances between each pair of feature points to be computed in *linear* time. This reduced computational complexity compared to using geodesic distances means that our algorithm can not only produce canonical forms much faster for simplified meshes, but also allows canonical forms to be computed for higher resolution meshes within a reasonable time. When substituting our canonical forms into Lian et al.'s retrieval method [3], we achieve comparable results to those using Elad and Kimmel's canonical forms [4], but much more quickly. We are also able to produce canonical forms for the dataset used by Lian et al., but without the need to simplify the models first. Another way in which greater speed can be put to use is to allow a larger number of meshes to be compared within a fixed time budget if the search space is a large database.

2. Related work

Many works consider rigid shape matching, including ones based on *lightfield* descriptors [6] and *spherical harmonics* [7]. For a

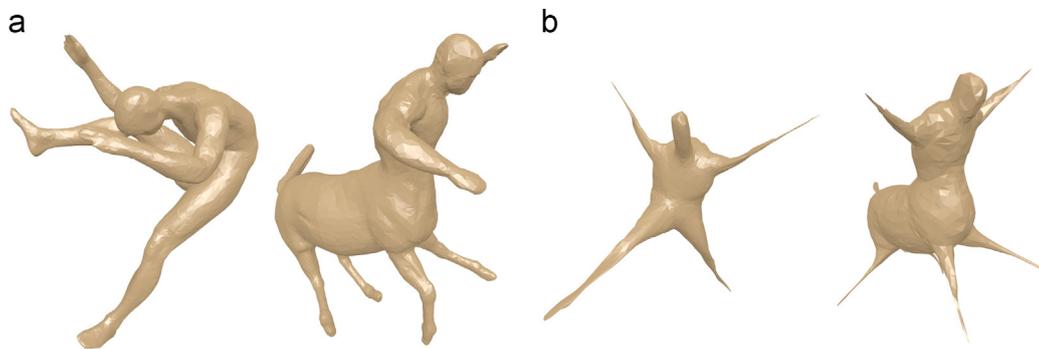


Fig. 1. Canonical forms produced using geodesic distances and multidimensional scaling. The limbs of the models are stretched so that their extremities are distant from one another. (a) Original meshes and (b) Canonical forms.

detailed review of rigid shape retrieval methods, we refer the reader to two survey papers [8,9].

Since our work contributes to non-rigid shape retrieval, we concentrate our review on this topic. Such work is based on shape invariance under isometric transformations. A popular approach compares models based on their local features. Some of these methods directly follow successful methods in computer vision, including *meshSIFT* [10] and *meshHOG* [11]. Sun et al. [12] proposed *heat kernel signatures* (HKS) as a shape descriptor. Inspired by *Video Google* [13], Bronstein et al. [14] used the heat kernel signature and the bags-of-features approach to produce shape signatures for retrieval. Abdelrahman et al. [15] also proposed a similar shape retrieval method based on HKS. Hou et al. [16] used HKS and a novel *bag-of-features graph* method which incorporates spatial relationships of local features. Wang et al. [17] proposed an extension to *spin images* [18] to allow for non-rigid matching, computing intrinsic spin images from a high-dimensional MDS embedding of a 3D model. Both Tabia et al. [19] and Koury et al. [20] describe each detected feature point of a model in terms of a set of local closed curves on its surface.

A second category of techniques uses graphs to represent objects. Hilaga et al. [21] use *multiresolution Reeb graphs* to match the topology of 3D shapes. Sundar et al. [22] use a *thinning* technique to produce an object's *skeleton*, and graph matching to compare skeletons. Sfikas et al. [23] produce a graph representation using geometry-based *discrete conformal factors* [5].

Methods using global information have also been proposed. The *shapeDNA* method of Reuter et al. [24] uses the Laplace–Beltrami operator to provide an isometrically invariant shape descriptor. Smeets et al. [25,26] showed that both singular value decomposition and a histogram of the *geodesic distance matrix* can be used as global shape descriptors. They showed that singular value decomposition is especially effective for shape retrieval. Jain and Zhang [27] rely on a *spectral embedding* of a 3D object using the eigenvalues of the geodesic distance matrix. Bronstein et al. [28] directly compute the distance between two non-rigid shapes by formulating the *Gromov–Hausdorff distance* as a multidimensional-scaling-like continuous optimisation problem. Tam and Lau [29] showed that manifold learning can be used to improve the shape retrieval accuracy of existing methods.

Elad and Kimmel [4] proposed computing a canonical form for a mesh using *multidimensional scaling* (MDS) to map the geodesic distances of a mesh into three-dimensional Euclidean distances. Rigid matching can then be performed on the resulting canonical forms. Lian et al. [3] used these canonical forms, along with a *view-based* retrieval method to match non-rigid shapes. It renders depth images of 66 different views of an object, extracts SIFT features from each of these images, and uses the bag of features approach to form a shape descriptor from these features. Recently, Lian et al. [30] created a *feature preserving* canonical form by transforming

parts of the original mesh to positions and orientations that correspond to equivalent segments of the mesh's canonical form. Lian et al. [31] have also used a *convexity measurement* of the canonical forms as a global feature for shape retrieval. The drawback of these methods is the *super-quadratic* computational complexity of finding the geodesic distances between all pairs of vertices, requiring the canonical forms to be computed from a simplified version of the original mesh in practice.

Ying et al. [32] gave a parallel geodesic distance algorithm to improve the efficiency of computing these distances. This algorithm still exhibits higher computational complexity than our method described below, but claims an order of magnitude improvement of computation time for meshes containing more than 500,000 faces. However, even if the computation time of the fastest geodesic-distance-based retrieval method we use as a basis for comparison in Section 6 were an order of magnitude faster, our method would still be 3.5 times faster than an approach using Ying's method, even for meshes with a resolution of about 10,000 vertices. Unfortunately, as the authors of [32] have not made their code available, we are unable to directly compare computation speeds with a canonical form method using their algorithm. Ultimately, however parallelisation can only lead to a constant factor speed-up, and not a decrease in computational complexity.

A similar approach to ours has been proposed by Wang and Zha [33], who also avoid computing geodesic distances between all pairs of vertices. They also select feature points, but based on local maxima and minima of geodesic distances to the two most *geodesic distant vertices* [34]. They then use the geodesic distances between all pairs of feature points to create target axes used to align local sets of geodesic contours. They compared the result of using their canonical forms for shape retrieval against classical MDS with all-pairs geodesic distances, but not using *least squares MDS*, which has been shown to produce improved results [4,3]. Their method is also much more complex than ours, and requires more parameters to be defined by the user. While their method is faster than ours for meshes of approximately 9000 vertices, they did not examine how their method scales to finer meshes. We are unable to determine the exact time complexity of their method, as they omit details of how to minimise certain cost functions. We can deduce however that their geodesic computations have time complexity at least $O(MN \log N)$, where M is the number of feature points, and N is the number of mesh vertices. Our distance calculations have time complexity of $O(N)$, and so will eventually be faster for large enough meshes. Our retrieval results are also more accurate than theirs for each dataset tested later, but these are based on the use of a different retrieval method.

For further insight, we refer the reader to Lian et al. [2] for an in-depth comparison of several non-rigid 3D shape retrieval methods.

3. Preliminaries

We first give a brief overview of the canonical form computation by Elad and Kimmel [4], which uses least squares multidimensional scaling to map geodesic distances into Euclidean distances. We also give an overview of the shape retrieval method by Lian et al. [3], which successfully uses these canonical forms for non-rigid shape retrieval, as we later substitute our canonical forms into this method in order to evaluate their effectiveness against those of Elad and Kimmel [4].

3.1. Multidimensional scaling for canonical form computation

Multidimensional scaling (MDS) maps a chosen distance measure between pairs of points to Euclidean distances in a low-dimensional space. The use of MDS to compute a pose invariant canonical form of a mesh was first proposed by Elad and Kimmel [4]. They use least squares MDS to map the *geodesic* distances between all points on a 3D mesh to 3D *Euclidean* distances. The geodesic distances are computed using the *fast marching method* of Kimmel and Sethian [35]. They use the *SMACOF* (scaling by maximising a convex function) algorithm [36] to compute the MDS. SMACOF minimises the following functional:

$$S(X) = \sum_{i=1}^N \sum_{j=i+1}^N w_{ij} (\delta_{ij} - d_{ij}(X))^2, \quad (1)$$

where N is the number of vertices, w_{ij} are weighting coefficients, δ_{ij} is the geodesic distance between vertices i and j of the original mesh, and d_{ij} is the Euclidean distance between vertices i and j of the resulting canonical mesh X .

This is solved by iteratively computing

$$X_i = V^+ B(X_{i-1}) X_{i-1}, \quad (2)$$

where the elements of matrix $B(X_{i-1})$ are

$$b_{ij} = \begin{cases} -w_{ij} \delta_{ij} / d_{ij}(X_{i-1}) & i \neq j \text{ and } d_{ij}(X_{i-1}) \neq 0 \\ 0 & i = j \text{ and } d_{ij}(X_{i-1}) = 0 \end{cases} \quad (3)$$

$$b_{i,i} = \sum_{j=1, j \neq i}^N b_{ij}. \quad (4)$$

The matrix V is given by

$$V = \sum_{i=1}^N \sum_{j=i+1}^N w_{ij} A_{ij}, \quad (5)$$

$$A_{ij} = (e_i - e_j)(e_i - e_j)^T, \quad (6)$$

where e_i is the i th column of the identity matrix. V^+ is the Moore–Penrose pseudo-inverse of V . The algorithm iterates until $S(X_i) - S(X_{i-1})$ is less than a user specified value ϵ . For more details, see [4].

This canonical form computation is used for 3D shape retrieval by Lian et al. [3]. Due to the super-quadratic computational expense, they first simplify the input meshes to approximately 2000 vertices. In Section 4 we give a method for computing canonical forms which instead of using a *quadratic* number of global *geodesic* distances, only uses the edge lengths of the mesh plus a *linear* number of *Euclidean* distances.

3.2. Non-rigid shape retrieval using canonical forms

The state-of-the-art method for using canonical forms for non-rigid shape retrieval is due to Lian et al. [3]; their method performed very well in the 2011 Shape Retrieval Competition (SHREC'11) [2,37]. They extract a shape descriptor from a model using canonical forms and visual features.

An outline of the steps used is shown in Fig. 2. They first simplify the original model (Fig. 2(a)) to produce a new mesh with approximately 2000 vertices (Fig. 2(b)). They next produce a canonical form of the mesh (Fig. 2(c)), as explained in Section 3.1. It takes too long to process higher resolution meshes without simplification due to the use of all pairs of geodesic distances.

Once the canonical form has been computed, the position and orientation of the model are normalised using a combination of PCA and rectilinearity [38]. A set of 66 depth images are captured of the model, from viewpoints located at the vertices of a geodesic sphere (Fig. 2(d)). For each of these depth images a set of SIFT features [39] are extracted (Fig. 2(e)). The set of SIFT features is used to generate a histogram (bag-of-features) for each image (Fig. 2(f)).

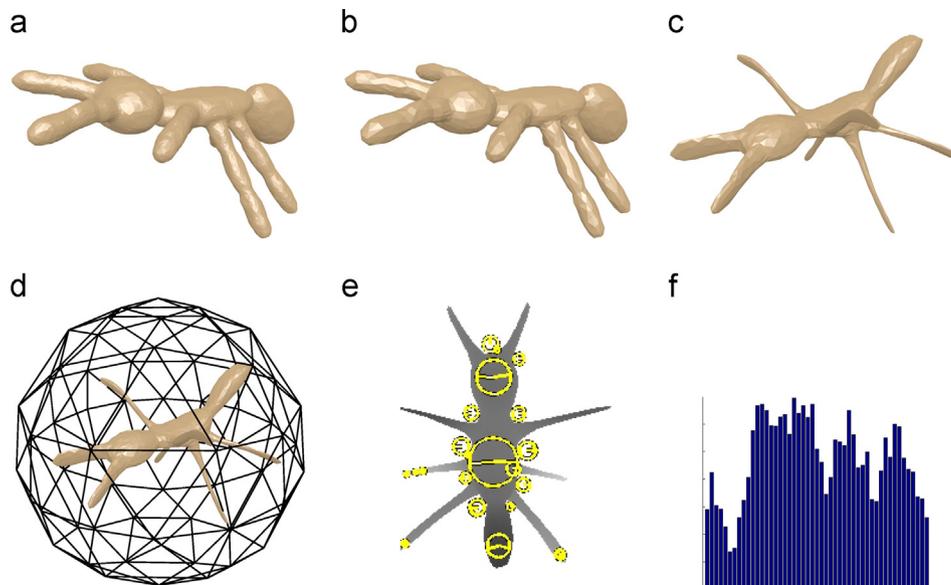


Fig. 2. Computing an object descriptor, according to Lian et al. [3]. (a) Original mesh. (b) Simplify mesh. (c) Compute canonical form. (d) Capture multiple depth images. (e) Extract SIFT features. (f) Construct bag of features.

The distance between two models is computed by comparing the feature histograms for the models using the *maximum dissimilarity histogram intersection distance*.

Our method can be used in this framework to provide alternative canonical forms to those of Elad and Kimmel, significantly speeding up this bottleneck step of the matching algorithm, and eliminating the need to simplify the meshes to such a small number of vertices. We later show that we can compute canonical forms for the models in the SHREC’11 dataset without the need for simplification.

4. Euclidean distance based canonical form computation

Fig. 1 shows two canonical forms produced by Lian et al. [3] from their original meshes. The limbs and similar parts of articulated objects are stretched out, so that their extremities are distant from one another. We achieve this effect more efficiently by maximising the Euclidean distances between certain selected feature points on the mesh, while attempting to preserving the original edge lengths to ensure isometric deformation. The feature points are selected based upon the conformal factors of the vertices [5]. The conformal factor increases along the length of mesh protrusions, which results in high values at the extremities of the mesh.

We first scale the mesh so that the maximum distance of any point on its surface to the centroid of all vertices is one. We then use the method of Ben-Chen and Gotsman [5] to calculate the conformal factor ϕ of the mesh as

$$L\phi = K^T - K^0, \tag{7}$$

where L is the discrete Laplace–Beltrami operator with cotangent weights [40], K^0 is a vector containing the Gaussian curvature for each vertex, and K^T is a vector containing a target Gaussian curvature for each vertex.

The Gaussian curvature κ_v^0 at a vertex v is defined as

$$\kappa_v^0 = \begin{cases} 2\pi - \sum_{t \in T_v} \theta_t & v \notin B \\ \pi - \sum_{t \in T_v} \theta_t & v \in B \end{cases}, \tag{8}$$

where θ_t is the angle subtended at vertex v in triangle t , T_v is the set of triangles connected to v , and B is the set of all vertices on the mesh boundary (if any). The target curvature κ_v^t at a vertex v is set to

$$\kappa_v^t = \left(\sum_{i \in V} \kappa_i^0 \right) \frac{\sum_{t \in T_v} \text{Area}(t)/3}{\sum_{t \in T} \text{Area}(t)}, \tag{9}$$

where V is the set of all vertices and T is the set of all triangles.



Fig. 3. Conformal factors for a human mesh model in varying poses. The conformal factor increases towards the ends of the model's limbs.

Eq. (7) can be solved for ϕ using Cholesky factorisation. An example of the conformal factors of a mesh in various poses is shown in Fig. 3.

To obtain a set of feature points for a mesh with N vertices, we sample the \sqrt{N} vertices which have the largest conformal factors and also satisfy the requirement that they are local maxima. A vertex is defined to be a local maximum if its conformal factor is greater than that of all its neighbours in a 2-ring neighbourhood. Our experiments show that points at the extremities of the mesh are likely to be present within this set of chosen feature points (Fig. 4). Our experiments on a previous dataset [41] show that using 1, 2, or 3-ring neighbourhoods all result in similar final retrieval results; using 2-ring gives marginally better performance. We select \sqrt{N} feature points, as this is the largest number of features we can have while being able to compute the distances between all pairs of feature points in linear time (with respect to the number of mesh vertices). Reducing the number of feature points further would not reduce the overall time complexity of our algorithm, as other aspects of our method also have a linear time complexity. On the other hand, our experiments demonstrate that \sqrt{N} feature points are sufficient to give good shape retrieval in our tests. A further reason for using exactly \sqrt{N} feature points is given later.

We compute the canonical form of the mesh by setting the value of δ_{ij} in Eq. (1) for all connected vertices i and j equal to the length of the edge connecting them. This aims to preserve the edge lengths of the mesh, to ensure isometric deformation. In order to maximise the distance between feature points, the value of δ_{ij} for each pair of the \sqrt{N} sampled vertices is set to a high value α . We want this value to be large enough to straighten all the limbs of the model, and our experiments show 10 is large enough. As long as α is large enough and the parameter β discussed below is optimised accordingly, any value of α can be chosen.

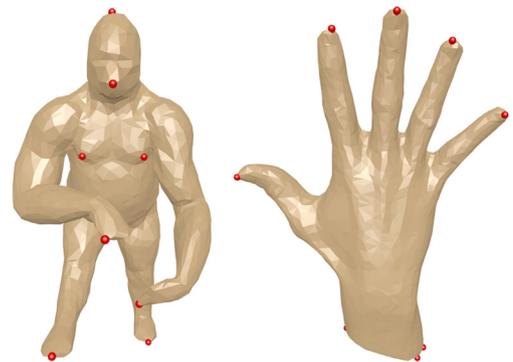


Fig. 4. Feature points selected using conformal factors. The mesh extremities are successfully included in this set.

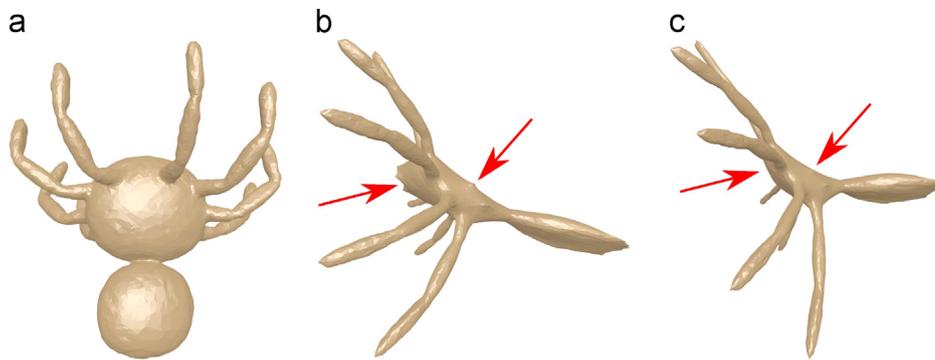


Fig. 5. Setting the MDS weighting for each pair of feature points to the mean of their conformal factors avoids undesirable extrusions at parts of the mesh where feature points do not lie at extremities. (a) Original mesh. (b) Weightings set to 1. (c) Weightings set to the mean of the conformal factors.

If the two vertices i and j are neither a pair of feature points nor connected by an edge, we do not enforce a target distance between them, so δ_{ij} and w_{ij} are both set to zero for such cases. Not having to compute and optimise the distances between these points is crucial in keeping the linear time complexity of our distance calculations. The weights w_{ij} in Eq. (1) for all i and j that are connected by an edge are set to β/δ_{ij}^2 , where β is a user defined parameter for preserving edge lengths. (We consider the optimal choice for β in the Experiments section later.) We divide by the square of the edge length δ_{ij}^2 so that the distance in Eq. (1) becomes a relative, rather than absolute, difference, making the weighting independent of the length of the edge. The conformal factor is normalised to lie in the interval $[0, 1]$, and the entries in the weighting matrix w_{ij} for each pair of feature points are set to the mean of their conformal factors. This results in vertices which are nearer the ends of long ‘limbs’ of the object having a higher impact on the resulting canonical form, and avoids stretching out inappropriate parts of the mesh (Fig. 5). The SMACOF algorithm can then be used to minimise Eq. (1) as previously described in Section 3.1.

In many cases the number of local maxima of conformal factor is less than \sqrt{N} . We want the number of feature points to be exactly \sqrt{N} so that the number of edges connecting pairs of feature points grows at the same rate as the number of mesh vertices. This in turn ensures that we can use the same value for the parameter β for preserving edge lengths, regardless of mesh resolution. We offer two different solutions to handling this issue. The first is to increase the number of feature points to \sqrt{N} by adding extra randomly selected vertices as feature points.

The second is to separately normalise the weightings w_{ij} used for pairs of feature points, and for adjacent vertices. We normalise the weights for adjacent vertices by dividing by the total number of edges, and we normalise the feature point pair weights by dividing by the sum of all feature point pair weights. Thus, we may rewrite the final functional to be minimised as

$$S(X) = \sum_{i \in F} \sum_{j \in F, j \neq i} \frac{0.5(\Phi_i + \Phi_j)}{\sum_{i \in F} \sum_{j \in F, j \neq i} 0.5(\Phi_i + \Phi_j)} (\delta_{ij} - d_{ij}(X))^2 + \sum_{(i,j) \in E} \frac{\beta}{|E| \delta_{ij}^2} (\delta_{ij} - d_{ij}(X))^2, \quad (10)$$

where F is the set of all feature points, E is the set of all edges, and Φ_i is the conformal factor of vertex i .

Examples of canonical forms produced by both versions of our method are shown later in Fig. 8.

5. Computational complexity

In this section we analyse the computational complexity of our canonical form computation and compare it to the original

method of Elad and Kimmel described in Section 3.1. We first examine the computational complexity of calculating the target distance matrix δ , used in Eq. (1). Elad and Kimmel used the geodesic distance matrix produced by the fast marching method [35], which has a time complexity of $O(N^2 \log N)$, where N is the number of vertices in the mesh.

Our method instead uses a constant high value α for N entries in the distance matrix. These N entries correspond to pairs of feature points. To locate feature points we calculate the conformal factor of each vertex on the mesh, identify at most \sqrt{N} local maxima, and in the first version of our method we add randomly selected vertices if we have fewer than \sqrt{N} local maxima. The conformal factor has several components. Both the Laplace–Beltrami and Gaussian curvature computations are $O(NM)$, where M is the average valence of a vertex on the mesh. Euler’s formula tells us that the average valence of a vertex on a triangular mesh is 6 [42]. The computational complexity of locating the feature points is therefore equal to $O(N)$ as M may be considered constant. The computation of the target curvature has a complexity of $O(N+T)$, where T is the number of triangles in the mesh. Again, this is $O(N)$ as the number of triangles is linearly related to the number of vertices. As the matrix L in Eq. (7) has size $N \times N$ but is sparse with $O(N)$ non-zero elements, the linear system can be solved for the conformal factor in $O(N)$ time.

Our method also requires calculation of the edge lengths for all pairs of connected vertices, which has a computational complexity of $O(N)$. Computing the set of feature points, along with all the distance computations is significantly more efficient—linear—than computing the geodesic distances—super-quadratic—required by previous canonical form computations.

Calculating the values of b_{ij} and d_{ij} for each iteration of the SMACOF algorithm (Section 3.1) has a computational complexity of $O(N^2)$ when the distance between all pairs of points is used. Our method uses all pairs of \sqrt{N} feature points plus all edge lengths, which lowers the computational complexity to $O(N)$.

6. Experiments

Here we present the results of our experiments. In Section 6.1 we experimentally decide on the best value for the edge preservation weight β . Section 6.2 compares the run-time of our method with that of the method by Elad and Kimmel [4] based on MDS and geodesic distances (Section 3.1). Finally we substitute our canonical forms into the shape retrieval method of Lian et al. [3], described in Section 3.2, to analyse their effectiveness for non-rigid shape retrieval.

6.1. Parameter optimisation

We used the TOSCA non-rigid models dataset [41] to determine the best value to use for the edge preserving weight β . This dataset

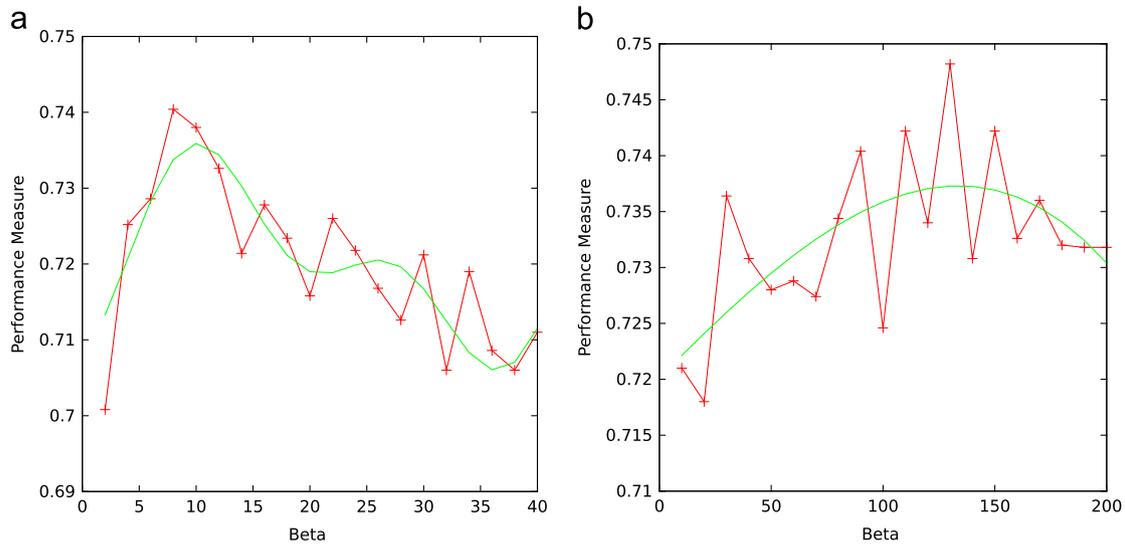


Fig. 6. Shape retrieval performance using our canonical forms for different values for the edge length preserving parameter β . (a) Additional random feature points. (b) Normalised weightings.

consists of 148 meshes, divided into 12 classes. Each class consists of the same model in different articulated poses. We performed shape retrieval on this dataset using Lian et al.'s method, but using our canonical forms instead of Elad and Kimmel's. The output of the shape retrieval test is a distance matrix between all pairs of models.

We performed shape retrieval several times, each time using a different β value. We evaluated the distance matrices produced using each of these results using five quantitative measures of how well they perform: nearest neighbour (NN), 1-tier, 2-tier, e-measure, and discounted cumulative gain (DCG); see [2] for a description of these measures and their use in assessing shape retrieval performance. The measures all lie in the range $[0, 1]$, and we average over them all and plot the performance against β . We do this for both variants of our method which deal differently with meshes of varying resolution (adding random features, or normalising the weights in the MDS formulation) (see Fig. 6). This test shows that $\beta=10$ is approximately the best choice when adding random features, while performance only drops off by less than 2% if we choose β anywhere between 5 and 15. We also come to the same conclusion when studying each statistical measure individually, except for the e-measure which shows high performance for a wider range of parameter values. When using the weight normalisation variant of our method, $\beta=130$ is the best choice, but again, the method is insensitive to changes in β .

6.2. Run-time

The main advantage of our technique over the existing state of the art is that we do not require the expensive computation of geodesic distances to produce canonical forms. In this section, we therefore present a run-time comparison between our method, and a related method using geodesic distances. Our run-time tests were performed on a Linux PC with a 3.2 GHz Intel Core i7-3930k CPU and 32 GB of memory. In Table 1 the timings for computing our canonical forms are compared to the times taken to produce the canonical forms using geodesic distances, for all the models in the SHREC'11 non-rigid dataset [37]. This dataset consists of 600 watertight meshes, with an average of 9300 vertices. The method by Lian et al. [3] simplified these meshes to contain approximately 2000 vertices, before computing their canonical forms using least squares MDS and geodesic distances. The run-time for both our method and the geodesic distance based method is shown for

Table 1

Run-time for computing canonical forms for the SHREC'11 dataset. Our method takes significantly less time, especially for the original meshes. Our method is faster using the original meshes, than the geodesic distance based method using the simplified meshes.

| Method | All models | Average/model |
|--|------------|---------------------|
| Geodesic distances (C, simplified meshes) | 8.5 h | 51 s |
| Geodesic distances (C, original meshes) | – | 21 min ^a |
| Geodesic distances (F, simplified meshes) | 8.5 h | 51 s |
| Geodesic distances (F, original meshes) | – | 19 min ^a |
| Geodesic distances (LS, simplified meshes) | 11 h | 66 s |
| Geodesic distances (LS, original meshes) | – | 24 min ^a |
| Our method (simplified meshes) | 20 min | 2 s |
| Our method (original meshes) | 5.5 h | 33 s |

^a Geodesic distances (original meshes) were assessed for a single mesh only due to the length of time needed. Key: C, Classical MDS; F, Fast MDS; LS, Least Squares MDS.

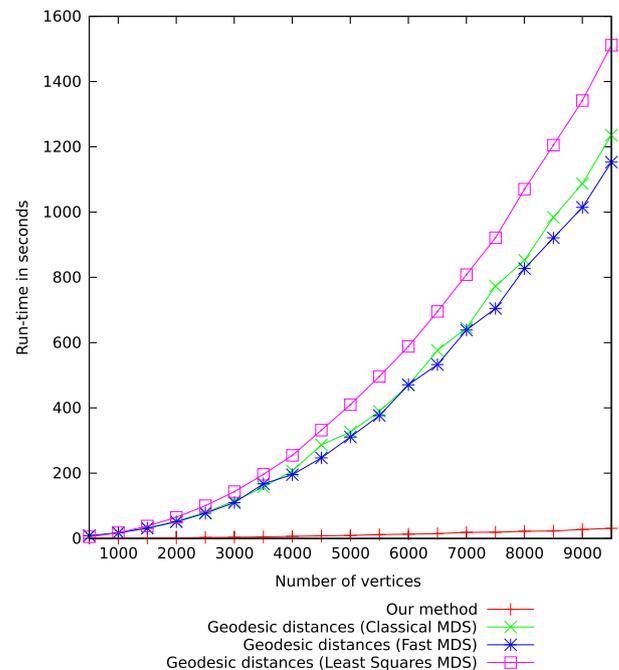


Fig. 7. Run-time for a single model, resampled to different mesh resolutions. Run-times grow significantly faster when using geodesic distances than for our method.

these simplified meshes, but only our method to compute canonical forms could be run on the original meshes in a reasonable enough time for all 600 models. We therefore only tested the run-time when using geodesic distances for one of the original models, which consists of 9300 vertices. The timings for the geodesic distance based method are shown for classical, fast [43] and least squares MDS. We do not include the timings for the simplification process, as they are negligible.

We also compared how the run-time of both our method and using geodesic distances grows as the number of vertices is increased. We selected a single 'typical' model, and created several different versions of it, each simplified to a different number of vertices. A graph showing time taken versus the number of vertices is presented in Fig. 7. When computing canonical forms with geodesic distances, the run-time grows significantly faster than our method as the number of vertices is increased. Our

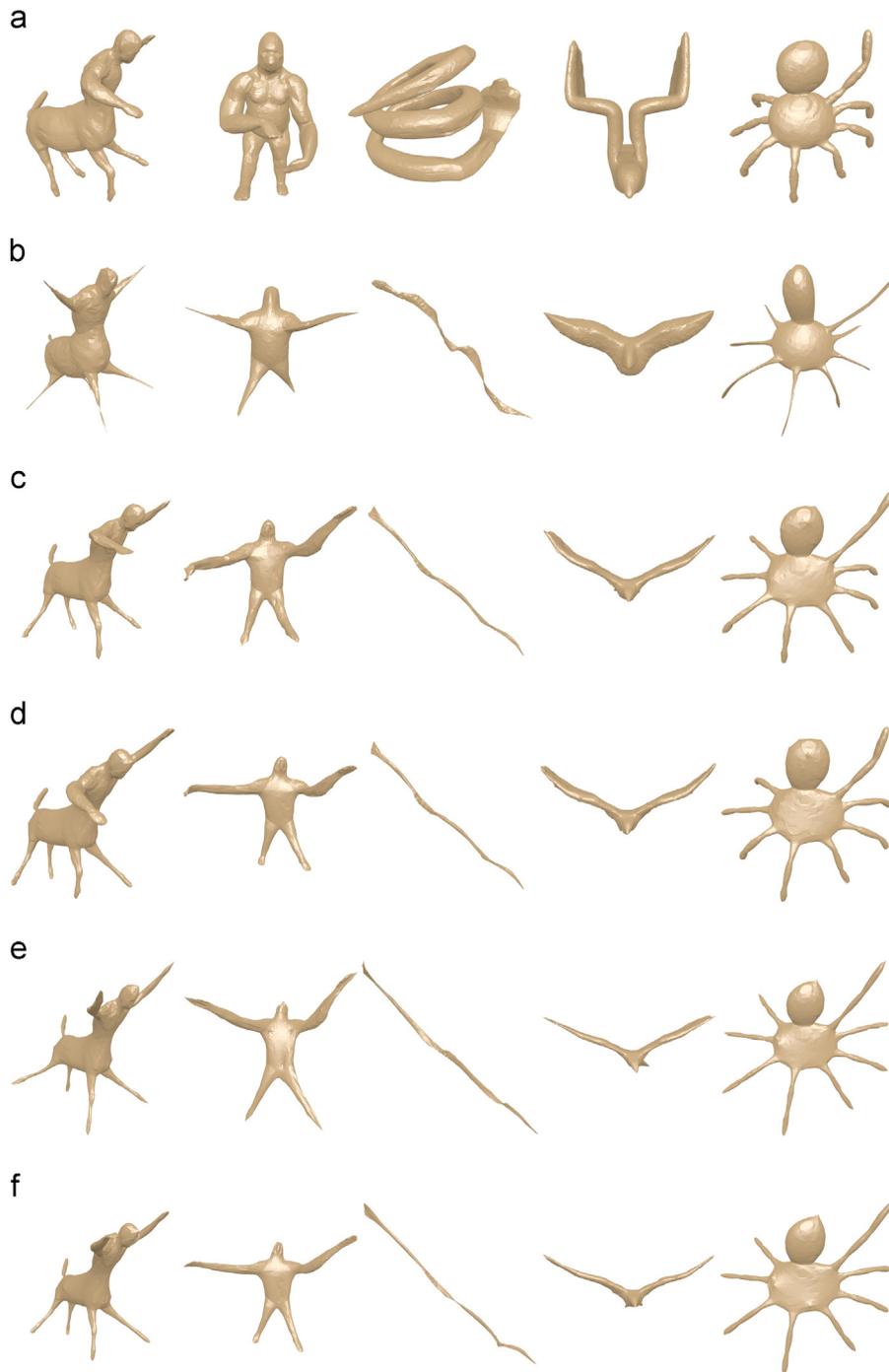


Fig. 8. Comparison of canonical forms for a selection of the SHREC'11 dataset, produced by Elad's method (b), and our methods, both for simplified (c, e) and unsimplified (d, f) meshes. Our method stretches out the limbs of the models, with less distortion along the 'limbs'. Key: S, simplified mesh; O, original mesh; A, additional random features; N, normalised weightings. (a) Original meshes, (b) Canonical forms produced using Least Squares MDS and geodesic distances with simplified meshes, (c) Canonical forms produced using our method (SA), (d) Canonical forms produced using our method (OA), (e) Canonical forms produced using our method (SN) and (f) Canonical forms produced using our method (ON)

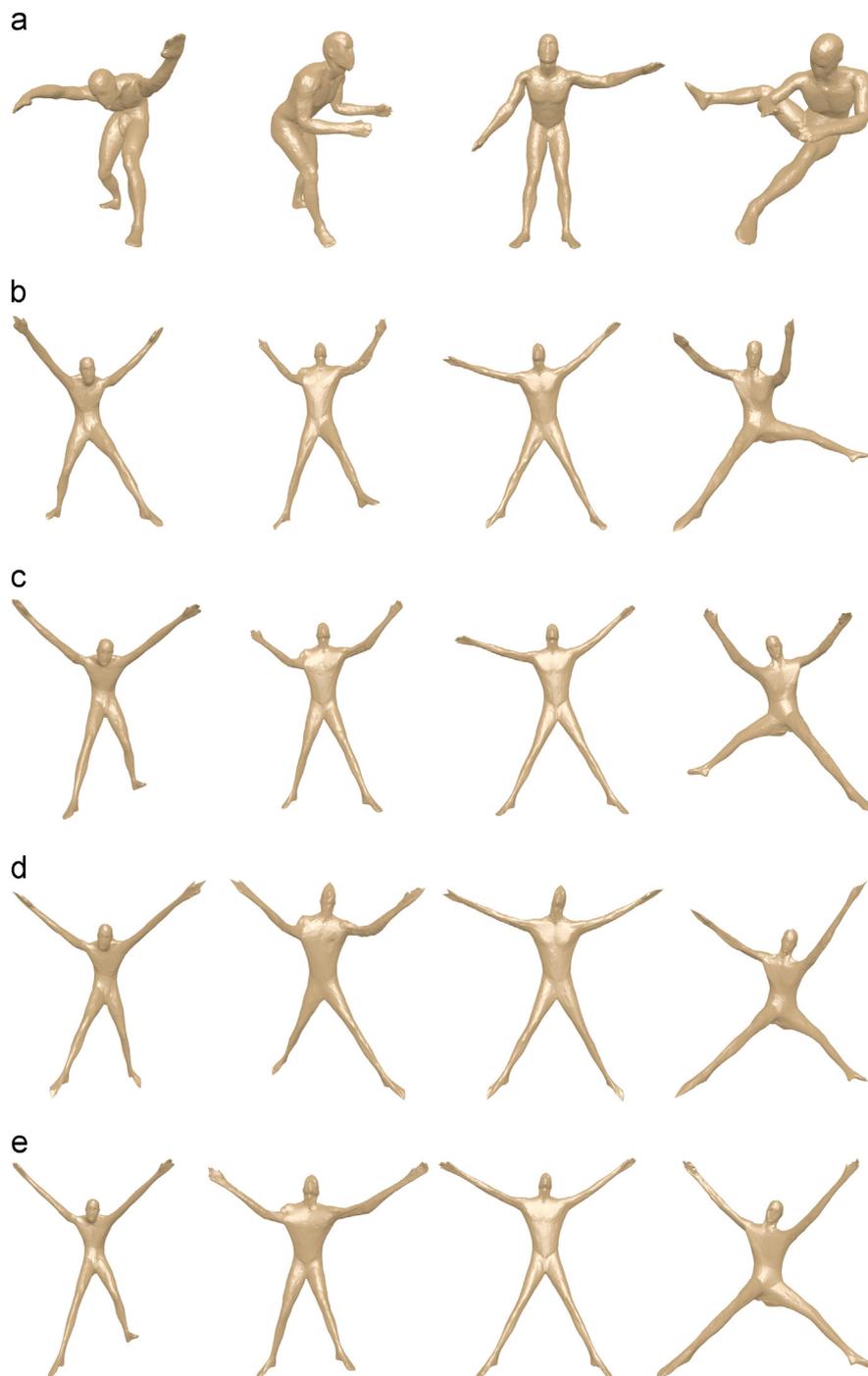


Fig. 9. Canonical forms of the same object initially in different poses, produced by our method, both for simplified (b, d) and unsimplified (c, e) meshes. Key: S, simplified mesh; O, original mesh; A, additional random features; N, normalised weightings. (a) Original meshes, (b) Canonical forms produced using our method (SA), (c) Canonical forms produced using our method (OA), (d) Canonical forms produced using our method (SN) and (e) Canonical forms produced using our method (ON).

method scales better for higher resolution meshes, as predicted theoretically.

6.3. Shape retrieval

We used the models from the non-rigid track of the 2011 Shape Retrieval Contest (SHREC'11) [37] to evaluate the effectiveness of our method at producing canonical forms suitable for shape retrieval. This dataset consists of 600 watertight meshes, divided into 30 classes. Each class consists of the same shape in 20

different poses. We evaluated the canonical forms using both variants of our method. In experiments using our method with additional random feature points we set $\beta=10$, and with weightings normalised, we set $\beta=130$.

Fig. 8 displays a selection of canonical forms produced by our method, for comparison with the canonical forms calculated using least squares MDS and geodesic distances [4] as used by Lian et al. [3] (Section 3.1). The canonical forms using geodesic distances were computed from simplified meshes consisting of approximately 2000 vertices; the original meshes were simplified using

MeshLab [44]. For our method, we show both the canonical forms computed using the simplified and original meshes, which on average had approximately 9300 vertices.

Our canonical forms produced using feature points based on conformal factors (Fig. 8(c)) successfully stretch out the ‘limbs’ of the models, as do the canonical forms produced using geodesic distances (see Fig. 8(b)). Our canonical forms however result in less distortion towards the ends of the extremities. Several canonical forms of the same object, but in different poses, are shown in Fig. 9.

We next demonstrate the results of using our method for non-rigid shape retrieval. We computed the distance matrix between all models from the SHREC’11 non-rigid track [37] by substituting our canonical forms into Lian’s view-based retrieval method. We compare our results to those from their original method, which uses least squares MDS and geodesic distances to calculate the canonical forms, we also show results for classical and fast MDS. Results based on comparing the original meshes without any canonical form computations are also shown. We evaluate the distance matrices produced using each of these methods using the same five quantitative measures used earlier when tuning the parameter β , and analyse them using precision-recall plots. We have also tested our canonical forms using a different view-based method [6], and the results exhibit the same trend. We have therefore not included them in our paper.

Table 2 shows the qualitative measures for each method; higher numbers are better. Using original meshes instead of canonical forms consistently performs worse on all measures, demonstrating the effectiveness of using canonical forms. When using additional random points to allow for varying numbers of feature points, our experiments show that using the original full resolution meshes achieves better results than when the meshes are simplified. The opposite appears true when we instead normalised the weightings of the MDS formulation. The original least squares MDS and geodesic distance method outperforms our method on most of the performance measures, but only by a small amount. Our method outperforms the geodesic based method used with classical or fast MDS.

Table 2 also shows a comparison with the highest performing submission of each research group to the SHREC’11 non-rigid track. This table shows that our method is ranked in the same position (2nd) as Lian et al.’s method with geodesic distance based canonical forms. Note that the best performing method (SD-GDM+meshSIFT [25,10]) also uses all pairs of geodesic distances and therefore suffers from high computational complexity.

Table 2
Comparison of retrieval results on the SHREC’11 non-rigid dataset. Our method achieves the same ranking as Lian et al.’s method [3] with geodesic distances. Key: C, Classical MDS; F, Fast MDS; LS, Least Squares MDS; S, simplified mesh; O, original mesh; A, additional random features; N, normalised weightings.

| Method | NN | 1-Tier | 2-Tier | e-Measure | DCG | |
|-----------------|-------------------------|--------------|--------------|--------------|--------------|--------------|
| Canonical | Original meshes | 0.985 | 0.746 | 0.863 | 0.627 | 0.931 |
| | Geodesic distances (C) | 0.987 | 0.855 | 0.943 | 0.691 | 0.964 |
| | Geodesic distances (F) | 0.978 | 0.795 | 0.905 | 0.657 | 0.945 |
| | Geodesic distances (LS) | 0.995 | 0.913 | 0.969 | 0.717 | 0.982 |
| | Our method (SA) | 0.983 | 0.867 | 0.966 | 0.709 | 0.974 |
| | Our method (OA) | 0.988 | 0.870 | 0.969 | 0.710 | 0.976 |
| | Our method (SN) | 0.995 | 0.875 | 0.970 | 0.713 | 0.977 |
| Our method (ON) | 0.982 | 0.867 | 0.964 | 0.708 | 0.974 | |
| SHREC’11 | SD-GDM+meshSIFT | 1.000 | 0.972 | 0.990 | 0.736 | 0.996 |
| | ShapeDNA | 0.992 | 0.915 | 0.957 | 0.705 | 0.978 |
| | FOG+MRR | 0.960 | 0.881 | 0.946 | 0.696 | 0.959 |
| | BOGH | 0.993 | 0.811 | 0.884 | 0.647 | 0.949 |
| | LSF | 0.995 | 0.799 | 0.863 | 0.633 | 0.943 |
| | BOW-LSD | 0.955 | 0.672 | 0.803 | 0.579 | 0.897 |
| | PatchBOF | 0.748 | 0.642 | 0.833 | 0.588 | 0.837 |
| | HKS | 0.837 | 0.406 | 0.497 | 0.353 | 0.730 |

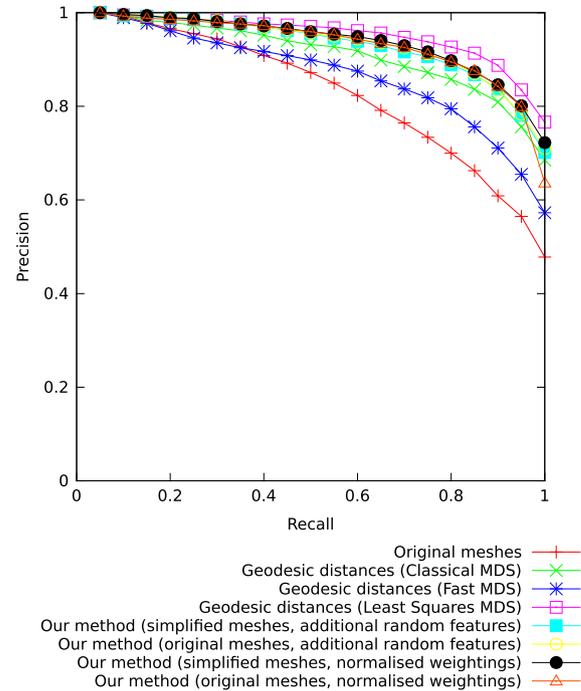


Fig. 10. Precision-recall plots of retrieval results on the SHREC’11 non-rigid dataset.

Table 3
Retrieval results for the McGill dataset. Key: C, Classical MDS; F, Fast MDS; LS, Least Squares MDS; S, simplified mesh; O, original mesh; A, additional random features; N, normalised weightings.

| Method | NN | 1-Tier | 2-Tier | e-Measure | DCG |
|-------------------------|--------------|--------------|--------------|--------------|--------------|
| Original meshes | 0.980 | 0.744 | 0.896 | 0.691 | 0.940 |
| Geodesic distances (C) | 0.961 | 0.728 | 0.868 | 0.678 | 0.931 |
| Geodesic distances (F) | 0.918 | 0.692 | 0.860 | 0.649 | 0.909 |
| Geodesic distances (LS) | 0.996 | 0.830 | 0.947 | 0.778 | 0.970 |
| Our method (SA) | 0.988 | 0.771 | 0.919 | 0.728 | 0.951 |
| Our method (OA) | 0.973 | 0.793 | 0.925 | 0.737 | 0.954 |
| Our method (SN) | 0.977 | 0.780 | 0.909 | 0.731 | 0.945 |
| Our method (ON) | 0.969 | 0.761 | 0.891 | 0.710 | 0.940 |
| Wang and Zha [33] | 0.980 | 0.746 | 0.895 | 0.707 | 0.938 |

Fig. 10 displays precision-recall plots for both the geodesic distance based method, and our method. Again, any canonical form method performs better than using only the original meshes. Our method produced very similar precision-recall plots when using simplified or original meshes. Using least squares MDS with geodesic distances shows a slight advantage over our method.

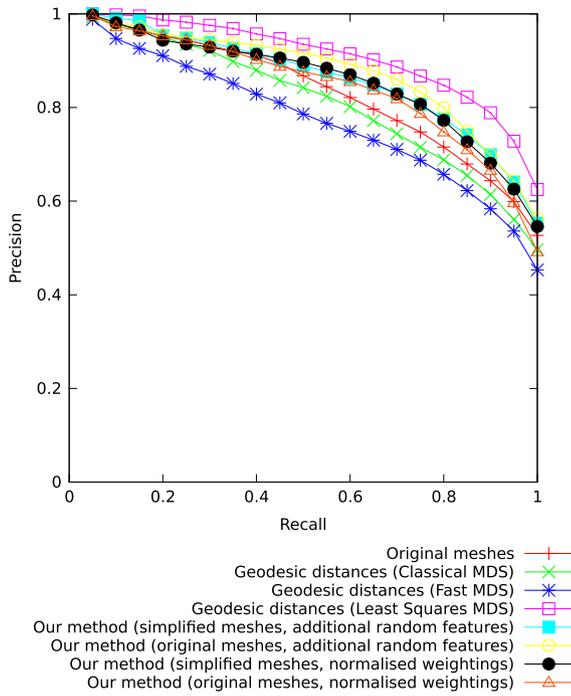


Fig. 11. Precision-recall plots of retrieval results on the McGill articulated dataset.

We have also produced retrieval results for the McGill articulated dataset [45]. This dataset consists of 255 models, made up of 10 different shape classes, with an average mesh resolution of about 14,000 vertices. As with the SHREC'11 dataset, the models were simplified to 2000 vertices to be able to produce canonical forms with the geodesic distance based method. Table 3 shows the retrieval performance of each method, and Fig. 11 displays precision-recall plots. On this dataset the use of canonical forms does not show as large an advantage over using the original articulated meshes. This may be because the differences between each class are still mostly visible, even when the meshes are in different poses. Both the classical and fast MDS with geodesic distances perform worse than when using the original articulated meshes. This may be because the large distortions they apply to the mesh cause greater retrieval error than the non-rigid articulations. For this dataset our method with additional random features outperforms using weighting normalisation when using the original full resolution meshes. Table 3 also shows the retrieval results presented by Wang and Zha [33]. Our method produces more accurate results than those which they report, but we use a different retrieval framework to compare canonical forms.

As our method for computing canonical forms does not distort the local details of the mesh as much as the original geodesic distances method, we would expect our method to perform better when such details are important. We tested this hypothesis on a specialised subset of the SHREC'11 dataset, comprising models in which all share the same basic shape (humanoid), but which vary in local detail. There are 80 models in this subset, split equally into the classes armadillo, gorilla, man, and woman. An example from each of these classes is shown in Fig. 12.

Results of running shape retrieval on the humanoid subset are shown in Table 4: our method outperforms the best geodesic distance method on all but one measure, both when using the simplified models with normalised weightings, and when using the full resolution models with additional random features. When



Fig. 12. Examples from the humanoid subset of the SHREC'11 dataset.

Table 4

Retrieval results for the SHREC'11 humanoid subset. Variants of our method outperform the original method using MDS and geodesic distances. Key: C, Classical MDS; F, Fast MDS; LS, Least Squares MDS; S, simplified mesh; O, original mesh; A, additional random features; N, normalised weightings.

| Method | NN | 1-Tier | 2-Tier | e-Measure | DCG |
|-------------------------|--------------|--------------|--------------|--------------|--------------|
| Original meshes | 0.963 | 0.657 | 0.870 | 0.607 | 0.880 |
| Geodesic distances (C) | 0.975 | 0.691 | 0.910 | 0.649 | 0.919 |
| Geodesic distances (F) | 0.925 | 0.688 | 0.936 | 0.658 | 0.916 |
| Geodesic distances (LS) | 0.975 | 0.761 | 0.949 | 0.678 | 0.928 |
| Our method (SA) | 0.913 | 0.773 | 0.948 | 0.677 | 0.938 |
| Our method (OA) | 0.963 | 0.767 | 0.952 | 0.679 | 0.945 |
| Our method (SN) | 1.000 | 0.753 | 0.959 | 0.685 | 0.948 |
| Our method (ON) | 0.913 | 0.731 | 0.938 | 0.667 | 0.935 |

using normalised weightings, we achieve a score of 100% for the nearest neighbour performance measure. The least squares MDS geodesic based method performs slightly better overall than the

other two runs of our algorithm, but not across all measures. We are able to both outperform best-of-class previous methods on such data, and produce results much more quickly.

Precision-recall plots for these results are shown in Fig. 13; again results for our method are still very similar for both the original and simplified meshes.

For both datasets, using our method to produce canonical forms using either the original or simplified meshes leads to very similar shape retrieval accuracy. This is likely to be for two reasons. The first is that the feature point locations chosen using conformal factors are stable over a wide range of mesh resolutions, leading to the same standardised pose. The second is that there is a large dissimilarity between the classes in the SHREC'11 non-rigid dataset, and so the different shapes are distinguishable even without the finer details discarded during simplification.

7. Limitations

Our method restricts the number of feature points to \sqrt{N} , and in the unlikely case that the mesh has more than \sqrt{N} extremities, our method will not be able to detect them all, so sections of the mesh will not be straightened out. All the models in the datasets we have used have fewer than \sqrt{N} extremities, and therefore we have not encountered this problem.

To test how our method deals with noise, we have used the normalised weightings version of our method to produce canonical forms for the SHREC'11 dataset, but with noise added to each of the meshes. The noise was added by randomly displacing each vertex by a small amount using Meshlab [44]. We have also

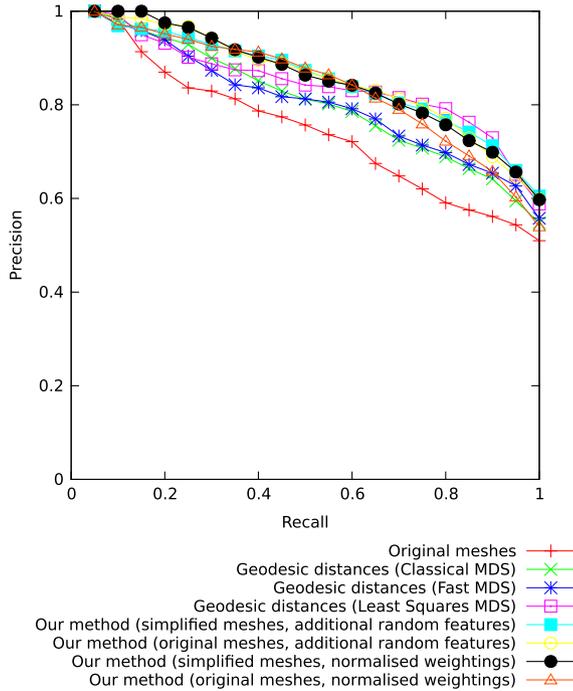


Fig. 13. Precision-recall plots of the retrieval results on the SHREC'11 humanoid subset.

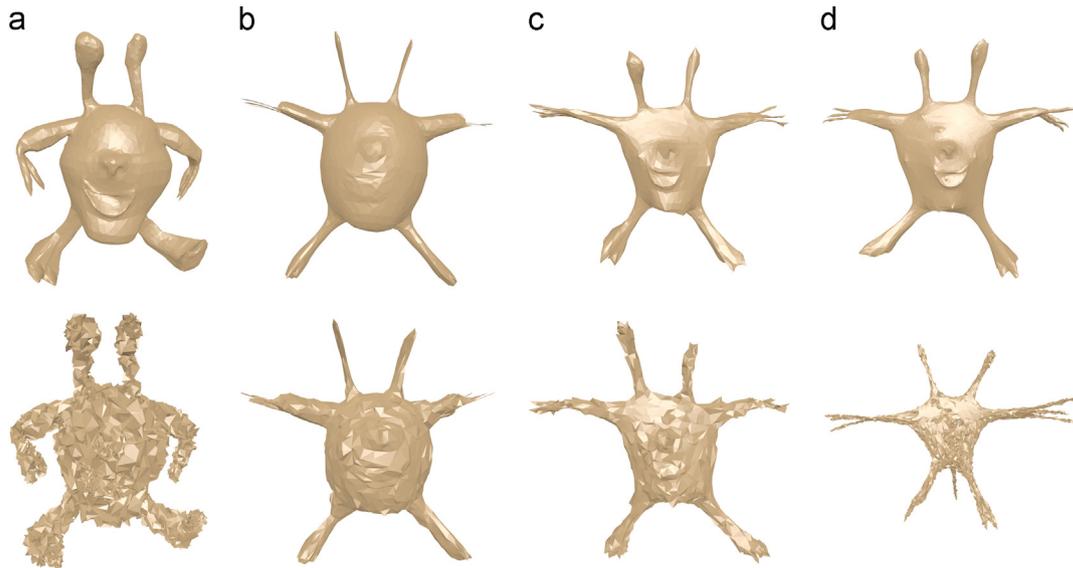


Fig. 14. Canonical forms of the same object with and without additional noise. Key: S, simplified mesh; O, original mesh; N, normalised weightings. Top: No noise, bottom: additional noise. (a) Original meshes, (b) Geodesic distance, (c) Our method (SN) and (d) Our method (ON).

Table 5 Retrieval results for the SHREC'11 dataset, with added noise. The noise has resulted in a decline of retrieval results for all methods. Key: C, Classical MDS; F, Fast MDS; LS, Least Squares MDS; S, simplified mesh; O, original mesh; A, additional random features; N, normalised weightings.

| Method | NN | 1-Tier | 2-Tier | e-Measure | DCG |
|-------------------------|--------------|--------------|--------------|--------------|--------------|
| Geodesic distances (C) | 0.985 | 0.802 | 0.918 | 0.670 | 0.948 |
| Geodesic distances (F) | 0.920 | 0.687 | 0.839 | 0.599 | 0.895 |
| Geodesic distances (LS) | 0.992 | 0.863 | 0.960 | 0.701 | 0.970 |
| Our method (SN) | 0.963 | 0.797 | 0.909 | 0.664 | 0.944 |
| Our method (ON) | 0.947 | 0.736 | 0.868 | 0.630 | 0.918 |

produced canonical forms of these meshes using the geodesic based method by Elad and Kimmel [4]. Fig. 14 shows an example of the effect of noise on the canonical forms for a single mesh. The canonical forms for the meshes with additional noise are still reasonable, but they are not completely consistent with the canonical forms produced from the original meshes. Table 5 shows the retrieval results when using the noisy meshes. The results show a drop in performance for each method, but the results using our method show a larger drop in performance than when using the least squares or classical MDS geodesic distance based methods. It is unclear, however, how much the drop in retrieval performance is caused by deterioration in the canonical forms, or by noise affecting the view-based retrieval method.

8. Conclusions

We have presented a novel linear-time method for producing canonical forms of meshes for non-rigid shape retrieval. Our method maximises the Euclidean distance between a small number of feature points, while attempting to preserve the original edge lengths. The feature points are chosen based on the conformal factors of the mesh vertices, which concentrates the feature points at the extremities of the model. Our method has lower computational complexity, and is much faster in practice, than methods that require all pairs of geodesic distances to be computed, while resulting in only a small drop in performance for both the SHREC'11 and McGill datasets. We furthermore show that when considering models with a similar basic shape (humanoids), our method can provide the best retrieval performance.

Using the original meshes of both datasets only produces a very small improvement over using simplified versions of the models when using one variant of our method, and produces a small decline in retrieval results for another. This may be because of the large dissimilarity between the classes in these datasets. This means that any finer details of the models, which are lost during simplification, are mostly unnecessary for distinguishing between models of different classes. Our method for finding feature points based on the conformal factors of the mesh vertices is able to find very similar points at different mesh resolutions, therefore the pose of the canonical forms will be similar over a wide range of mesh resolutions. The higher resolution meshes may also exhibit more noise, which could negatively affect the retrieval process.

Conflict of interest statement

None declared.

Acknowledgements

This work was supported by EPSRC Research Grant EP/J02211X/1.

References

- [1] Z. Lian, A. Godil, T. Fabry, T. Furuya, J. Hermans, R. Ohbuchi, C. Shu, D. Smeets, P. Suetens, D. Vandermeulen, S. Wuhrer, SHREC'10 track: non-rigid 3d shape retrieval, in: Proceedings of the 3rd Eurographics Conference on 3D Object Retrieval, EG 3DOR'10, Eurographics Association, 2010, pp. 101–108.
- [2] Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoué, H.V. Nguyen, R. Ohbuchi, Y. Ohkita, Y. Ohishi, F. Porikli, M. Reuter, I. Sipiran, D. Smeets, P. Suetens, H. Tabia, D. Vandermeulen, A comparison of methods for non-rigid 3d shape retrieval, *Pattern Recognit.* 46 (2013) 449–461.
- [3] Z. Lian, A. Godil, X. Sun, J. Xiao, CM-BOF: visual similarity-based 3d shape retrieval using clock matching and bag-of-features, *Mach. Vis. Appl.* (2013) 1–20.
- [4] A. Elad, R. Kimmel, On bending invariant signatures for surfaces, *IEEE Trans. Pattern Anal. Mach. Intell.* 25 (2003) 1285–1295.
- [5] M. Ben-Chen, C. Gotsman, Characterizing shape using conformal factors, in: Proceedings of the 1st Eurographics Conference on 3D Object Retrieval, EG 3DOR'08, Eurographics Association, 2008, pp. 1–8.
- [6] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, M. Ouhyoung, On visual similarity based 3d model retrieval, *Comput. Graph. Forum* 22 (2003) 223–232.
- [7] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3d shape descriptors, in: Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, SGP '03, Eurographics Association, 2003, pp. 156–164.
- [8] J. Tangelder, R. Veltkamp, A survey of content based 3d shape retrieval methods, *Multim. Tools Appl.* 39 (2008) 441–471.
- [9] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, The Princeton shape benchmark, in: International Conference on Shape Modeling and Applications, 2004, pp. 167–178.
- [10] C. Maes, T. Fabry, J. Keustermans, D. Smeets, P. Suetens, D. Vandermeulen, Feature detection on 3d face surfaces for pose normalisation and recognition, in: Fourth IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS), 2010, pp. 1–6.
- [11] A. Zaharescu, E. Boyer, K. Varanasi, R. Horaud, Surface feature detection and description with applications to mesh matching, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 373–380.
- [12] J. Sun, M. Ovsjanikov, L. Guibas, A concise and provably informative multi-scale signature based on heat diffusion, *Comput. Graph. Forum* 28 (2009) 1383–1392.
- [13] J. Sivic, A. Zisserman, Video google: a text retrieval approach to object matching in videos, in: Proceedings of Ninth IEEE International Conference on Computer Vision, vol. 2, 2003, pp. 1470–1477.
- [14] A.M. Bronstein, M.M. Bronstein, L.J. Guibas, M. Ovsjanikov, Shape google: geometric words and expressions for invariant shape retrieval, *ACM Trans. Graph.* 30 (2011) 1:1–1:20.
- [15] M. Abdelrahman, M. El-Melegy, A. Farag, Heat kernels for non-rigid shape retrieval: sparse representation and efficient classification, in: Ninth Conference on Computer and Robot Vision (CRV), 2012, pp. 153–160.
- [16] T. Hou, X. Hou, M. Zhong, H. Qin, Bag-of-feature-graphs: a new paradigm for non-rigid shape retrieval, in: 21st International Conference on Pattern Recognition (ICPR), 2012, pp. 1513–1516.
- [17] X. Wang, Y. Liu, H. Zha, Intrinsic spin images: a subspace decomposition approach to understanding 3d deformable shapes, in: Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission, 3DPVT'10, 2010, pp. 17–20.
- [18] A. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3d scenes, *IEEE Trans. Pattern Anal. Mach. Intell.* 21 (1999) 433–449.
- [19] H. Tabia, M. Daoudi, J.-P. Vandeboorde, O. Colot, A new 3d-matching method of nonrigid and partially similar models using curve analysis, *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (2011) 852–858.
- [20] R. Khoury, J.-P. Vandeboorde, M. Daoudi, Indexed heat curves for 3d-model retrieval, in: 21st International Conference on Pattern Recognition (ICPR), 2012, pp. 1964–1967.
- [21] M. Hilaga, Y. Shinagawa, T. Kohmura, T.L. Kunii, Topology matching for fully automatic similarity estimation of 3d shapes, in: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01, Los Angeles, ACM, New York, 2001, pp. 203–212.
- [22] H. Sundar, D. Silver, N. Gagvani, S. Dickinson, Skeleton based shape matching and retrieval, in: Shape Modeling International, 2003, pp. 130–139.
- [23] K. Sfikas, T. Theoharis, I. Pratikakis, Non-rigid 3d object retrieval using topological information guided by conformal factors, *Vis. Comput.* 28 (2012) 943–955.
- [24] M. Reuter, F.-E. Wolter, N. Peinecke, Laplace–Beltrami spectra as shape-DNA of surfaces and solids, *Comput.-Aided Des.* 38 (2006) 342–366.
- [25] D. Smeets, T. Fabry, J. Hermans, D. Vandermeulen, P. Suetens, Isometric deformation modelling for object recognition, in: X. Jiang, N. Petkov (Eds.), *Computer Analysis of Images and Patterns, Lecture Notes in Computer Science*, vol. 5702, Springer, Berlin, Heidelberg, 2009, pp. 757–765.
- [26] D. Smeets, J. Hermans, D. Vandermeulen, P. Suetens, Isometric deformation invariant 3d shape recognition, *Pattern Recognit.* 45 (2012) 2817–2831.
- [27] V. Jain, H. Zhang, A spectral approach to shape-based retrieval of articulated 3d models, *Comput.-Aided Des.* 39 (2007) 398–407.
- [28] A. Bronstein, M. Bronstein, R. Kimmel, Efficient computation of isometry-invariant distances between surfaces, *SIAM J. Sci. Comput.* 28 (2006) 1812–1836.
- [29] G. Tam, R. Lau, Embedding retrieval of articulated geometry models, *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (2012) 2134–2146.
- [30] Z. Lian, A. Godil, J. Xiao, Feature-preserved 3d canonical form, *Int. J. Comput. Vis.* 102 (2013) 221–238.
- [31] Z. Lian, A. Godil, P. Rosin, X. Sun, A new convexity measurement for 3d meshes, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 119–126.
- [32] X. Ying, S.-Q. Xin, Y. He, Parallel chen-han (pch) algorithm for discrete geodesics, *ACM Trans. Graph.* 33 (2014) 9:1–9:11.
- [33] X.-L. Wang, H. Zha, Contour canonical form: an efficient intrinsic embedding approach to matching non-rigid 3d objects, in: Proceedings of the 2nd ACM International Conference on Multimedia Retrieval, ICMR '12, Hong Kong, ACM, New York, 2012, pp. 31:1–31:8.

- [34] J. Tierny, J.-P. Vandeborrel, M. Daoudi, 3D mesh skeleton extraction using topological and geometrical analyses, in: 14th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2006), 2006, pp. 85–94.
- [35] R. Kimmel, J.A. Sethian, Computing geodesic paths on manifolds, *Proc. Natl. Acad. Sci.* 95 (1998) 8431–8435.
- [36] I. Borg, P. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, Springer Series in Statistics, Springer, New York, 2005.
- [37] Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoué, H. V. Nguyen, R. Ohbuchi, Y. Ohkita, Y. Ohishi, F. Porikli, M. Reuter, I. Sipiran, D. Smeets, P. Suetens, H. Tabia, D. Vandermeulen, SHREC'11 track: shape retrieval on non-rigid 3d watertight meshes, in: Proceedings of the 4th Eurographics Conference on 3D Object Retrieval, EG 3DOR'11, Eurographics Association, 2011, pp. 79–88.
- [38] Z. Lian, P. Rosin, X. Sun, Rectilinearity of 3d meshes, *Int. J. Comput. Vis.* 89 (2010) 130–151.
- [39] D. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vis.* 60 (2004) 91–110.
- [40] K. Hildebrandt, K. Polthier, M. Wardetzky, On the convergence of metric and geometric properties of polyhedral surfaces, *Geom. Dedic.* 123 (2006) 89–112.
- [41] A.M. Bronstein, M.M. Bronstein, R. Kimmel, *Numerical Geometry of Non-rigid Shapes*, Monographs in Computer Science, Springer, New York, 2008.
- [42] X. Sun, E.R. Hancock, Quasi-isometric parameterization for texture mapping, *Pattern Recognit.* 41 (2008) 1732–1743.
- [43] C. Faloutsos, K.-I. Lin, Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets, in: Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, SIGMOD '95, ACM, New York, NY, USA, 1995, pp. 163–174. URL: (<http://doi.acm.org/10.1145/223784.223812>), <http://dx.doi.org/10.1145/223784.223812>.
- [44] Meshlab v1.3.0, 2011. URL: (<http://meshlab.sourceforge.net>).
- [45] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, S. Dickinson, Retrieving articulated 3-d models using medial surfaces, *Mach. Vis. Appl.* 19 (2008) 261–275.

David Pickup obtained an MEng degree from the University of Bristol and a PhD from the University of Bath. He is currently a post-doc research associate within the Visual Computing research group at Cardiff University. His current research centres on three-dimensional shape retrieval.

Xianfang Sun received his PhD from the Institute of Automation, Chinese Academy of Sciences in 1994. He is currently a lecturer at the School of Computer Science, Cardiff University, Wales, UK. His main research interests include computer vision, computer graphics, pattern recognition and artificial intelligence.

Paul Rosin is Professor at the School of Computer Science & Informatics, Cardiff University. Previous posts include lecturer at the Department of Information Systems and Computing, Brunel University London, UK, research scientist at the Institute for Remote Sensing Applications, Joint Research Centre, Ispra, Italy, and lecturer at Curtin University of Technology, Perth, Australia.

His research interests include the representation, segmentation, and grouping of curves, knowledge-based vision systems, early image representations, low level image processing, machine vision approaches to remote sensing, methods for evaluation of approximation algorithms, etc., medical and biological image analysis, mesh processing, non-photorealistic rendering and the analysis of shape in art and architecture.

Ralph Martin obtained his PhD in 1983 from Cambridge University. Since then he has been at Cardiff University, where he now holds a Chair and leads the Visual Computing research group. He is also a Guest Professor at Tsinghua and two other Universities in China, and is Director of Scientific Programmes of the One Wales Research Institute of Visual Computing. His publications include over 250 papers and 14 books covering such topics as solid modelling, surface modelling, reverse engineering, intelligent sketch input, mesh processing, video processing, computer graphics, vision based geometric inspection, and geometric reasoning. He is a Fellow of the Learned Society of Wales, the Institute of Mathematics and its Applications, and the British Computer Society. He is on the editorial boards of “Computer Aided Design”, “Computer Aided Geometric Design”, “Geometric Models”, and other journals.