Eurographics Workshop on 3D Object Retrieval (2008)
I. Pratikakis and T. Theoharis (Editors)

Characterizing Shape Using Conformal Factors

Mirela Ben-Chen and Craig Gotsman

Technion – Israel Institute of Technology

Abstract

We present a new 3D shape descriptor based on conformal geometry. Our descriptor is invariant under non-rigid quasi-isometric transformations, such as pose changes of articulated models, and is both compact and efficient to compute. We demonstrate the performance of our descriptor on a database of watertight models, and show it is comparable with state-of-the-art descriptors.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

1. Introduction

As the number of available 3D models grows, the need for efficient shape indexing and retrieval has emerged. 3D models are usually represented in the simplest manner: a set of vertices and faces of a polygonal mesh. However, such a representation cannot be used for a direct comparison of two shapes, since it is not unique: the same shape can be represented by different sets of vertices and faces, with different resolutions. To reliably compare two shapes, search engines usually use a *shape signature* or *descriptor*, a numerical representation of the shape, invariant to the specific tessellation used to represent it as a polygonal mesh. In addition, shape descriptors should be invariant to rigid transformations and should produce "close" descriptors for "similar" shapes. An important invariance property for a shape descriptor is invariance to pose changes. A pose-invariant shape descriptor should be able to identify different poses of the same 3D shape, such as a running, sitting or standing human.

Pose changes are a quasi-isometric transformation of the 3D mesh, in the sense that edge lengths do not change much as a result of the transformation. The Gaussian curvature of a surface is invariant to isometric transformations, so it seems natural that some version of the Gaussian curvature may be used as a pose-invariant shape descriptor. Unfortunately, the computation of the (discrete) Gaussian curvature on a discrete surface is noisy, and the resulting function is not useful in practice as a shape descriptor.

In addition to Gaussian curvature, conformal geometry provides additional intrinsic measures for the surface geometry. The celebrated *uniformization theorem* (see [Abi81] for a modern proof) shows that any 2-manifold can be conformally mapped to a surface with the same topology having constant Gaussian curvature. This mapping can be described by a scalar function on the surface – the *conformal factor* – which is invariant to isometric transformations. As we will show, this function is both efficient to compute, and considerably less noisy than the Gaussian curvature. Consequently, a histogram of this function may serve as a robust pose-invariant signature of a 3D shape.

1.1. Previous work

The body of research dedicated to shape descriptors is quite vast, and a survey of it is beyond the scope of this paper. We will focus here only on recent descriptors which are most relevant to our work. The interested reader is referred to [SMKF04, TV08, DBP06] for comprehensive surveys of

different shape descriptors and evaluations of their performance**.**

As we shall see later, our shape descriptor makes use of the discrete Laplace-Beltrami operator. Several pose-invariant shape signatures based on this operator have been proposed recently. Reuter et al. [RWP05] use the spectrum (i.e. the eigenvalues) of this operator and Rustamov [Rus07] uses its eigenvectors. Xiang et al. [XHGC07] use the histogram of the solution to the volumetric Poisson equation $\nabla^2 U = -1$ as a pose-invariant shape descriptor. This equation also involves the Laplace-Beltrami operator (∇^2). Superficially similar to what we shall propose, it is different since it solves an equation on the *volume* of the shape, whereas we work on the boundary surface alone.

Another spectral-related shape signature was introduced by Jain and Zhang [JZ07], who use the eigenvectors and eigenvalues of the geodesic distances matrix. A pose invariant descriptor based on histograms of surface functions is presented by Gal et al. [GSCO07]. They use two scalar functions on the mesh. The algorithm performs quite well, even on meshes which contain more than one connected component. This shape descriptor is, however, relatively heavy to compute.

Some pose invariant shape descriptors are derived from geodesic distances on the mesh, which are invariant to isometric transformations. Elad and Kimmel [EK03] embed these distances in Euclidean space, where two such embeddings can be compared as rigid objects. In a more recent work, Tung and Schmitt [TS05] use geodesic distances for building a multiresolution Reeb graph. The discriminative power of such descriptors is usually very good, however this comes at the price of a high computational cost for computing the geodesic distances.

Conformal geometry has been applied to shape retrieval in the past. Wang et al. [WWJ*06] propose to do face recognition using the 2D conformal maps of 3D shapes, instead of comparing the 3D models directly. This method is applicable only to shapes which are a topological disc, since general shapes need to be cut (to form a boundary) before they can be embedded in 2D. As the cuts might not be compatible between different 3D shapes, the resulting 2D parameterization might be very different even if the original 3D shapes were similar. Since a 2D parameterization can be generated using the conformal scaling factor and the original 3D edge lengths [BGB08], using the conformal factor as a signature can be viewed as a generalization of Wang et al.'s method to general 3D meshes. Gu et al. [GV04] and Wang et al. [WCT05] combine the conformal factor with the mean curvature for shape matching in medical applications. However, since the mean curvature is an extrinsic measure dependent on the embedding, it is not invariant to isometric transformations. Recently, Jin et al. [JLYG07] suggested the use of the conformal class of a shape as a signature. Although the signature is invariant to isometries, it is not discriminative enough – for example, all closed genus zero surfaces belong to the same conformal class.

The rest of the paper is organized as follows. In the next section, we describe how the conformal factor is computed, and how the signature is generated and compared. In Section 3 we experimentally evaluate the performance of our descriptor. We conclude with a short discussion and future directions in Section 4.

2. The Shape Descriptor

2.1 Conformal Geometry

The celebrated uniformization theorem [Abi81] states that any 2-manifold surface can be conformally mapped to a surface with the same topology having constant Gaussian curvature. Such a mapping can be achieved by defining a positive scalar function on the surface, and locally changing the surface metric (which can be thought of as scaling infinitesimal patches of the surface) using this function. The scaling function ϕ (also known as the *conformal factor*) that achieves this depends only on the Gaussian curvature of the surface, hence is invariant to isometric transformations. In the continuous setting, conformal maps are well defined and understood, and ϕ is the solution to the non-linear partial differential equation:

$$\nabla^2\phi = \kappa - \exp(2\phi)\kappa_u$$

where κ is the Gaussian curvature of the surface, and κ_u is the (uniform) target Gaussian curvature.

In the discrete setting, conformal maps are less straight forward. There are a number of definitions of discrete "conformal" maps, and a number of methods to compute such mappings. Recently, Ben-Chen et al. [BGB08] proposed an efficient way of computing a conformal map using a discrete conformal scaling factor. This scaling factor is then used to scale the edges of the original mesh, to achieve edge lengths which induce the target curvature. For the sake of completeness, we repeat here the main algorithmic steps for computing the conformal factor.

2.2. The Discrete Conformal Factor

A triangular mesh M is given by sets of its vertices, faces and edges, which we denote by V,F and E respectively. The boundary vertices of the mesh M is given by the set B. The discrete Gaussian curvature [MDSB02] of a vertex v in the mesh (sometimes referred to as the angle defect) is defined as:

$$k_v = \begin{cases} 2\pi - \sum_{f \in F_v} \alpha_v^f & v \in V \setminus B \\ \pi - \sum_{f \in F_v} \alpha_v^f & v \in B \end{cases}$$

where F_v is the set of faces in F which share the vertex v, and α^f_v is the angle near the vertex v in the face f.

The conformal factor ϕ is the solution to the following sparse set of linear equations:

$$L\phi = K^T - K^{orig} \quad (1)$$

where L is the discrete Laplace-Beltrami operator with cotangent weights [HPW06], K^{orig} is the Gaussian curvature of the mesh, and K^T is the target Gaussian curvature. In our case, the target is uniform Gaussian curvature, so we set:

$$k_v^T = \left(\sum_{i \in V} \kappa_i \right) \frac{\sum_{f \in F_v} \frac{1}{3} area(f)}{\sum_{f \in F} area(f)}$$

This formula assigns to each vertex a portion of the total curvature. Although the target curvature is supposedly uniform, it is not uniformly distributed among the discrete vertices. Were this the case, the descriptor would not be tessellation invariant. Hence, the portion of the curvature assigned to a vertex is determined by the "influence area" of the vertex – a third of the area of the faces near the vertex, divided by the total surface area of the mesh. This also guarantees that the sum of the target curvatures on all vertices is identical to the sum of the original vertex curvatures.

For a connected mesh, the Laplacian has co-rank 1, so the solution ϕ is defined only up to an additive constant. This is resolved by requiring ϕ to have zero mean. Since the Laplacian is sparse and symmetric, Equation (1) may be solved for ϕ very efficiently, e.g. with Matlab's *mldivide* operator, which uses the CHOLMOD package [Dav05]

Figure 1: *Color-coding of the conformal factor of two hand models and two dancer models from the AIM@SHAPE shape repository [http://shapes.aim-at-shape.net/index.php] (normalized to the range* [0,1]*).*

Figure 1 shows the color coding of the conformal factor (normalized to the range [0,1]) computed for two dancer models, and two hand models from the AIM@SHAPE shape repository. As the figure demonstrates, the conformal factor identifies well the features of the shape - such as the fingers of the hand and the feet of the dancer. This is because the conformal factor represents how much local "work" is involved in globally transforming the model into a sphere (in the case of a genus zero model). Long extrusions such as the fingers and the feet require more "work", which is expressed as a larger conformal factor.

2.3. The Signature

Given the conformal factor we can now describe how to generate the signature of a given mesh, and how to compare two signatures.

As is commonly done when defining a shape descriptor based on a real function on the mesh [GSCO07, XHGC07,Rus07], we use the histogram of the conformal factor as our shape signature. Since the conformal factors are invariant to rigid transformations, so are their histograms. To make the signature invariant also to the tessellation of the mesh, we sample ϕ at $5|F|$ random points on the surface of the mesh. This is achieved by selecting a random triangle with probability proportional to its area, and then randomly generating a point uniformly distributed in the triangle interior. The value of ϕ at this point is defined to be the linear interpolation of ϕ at the vertices of the triangle. To this sample set we add the original mesh vertices, and compute the histogram of ϕ at all these points, which is now almost independent of the mesh' tessellation. Finally, we compute the histogram with 200 uniform bins spanning the range [-99,100], resulting in a signature with 200 values. See Figures 2 and 3 for some examples..

To compare two signatures, S_1 and S_2 we use the L_1 norm:

$$d(S^1, S^2) = \sum_i | S^1{}_i - S^2{}_i |$$

3. Evaluation of the Shape Descriptor

In this section we evaluate the performance of our shape descriptor, from both a theoretical and experimental point of view. First we explain why it is indeed pose-invariant, and then we show some experimental results demonstrating its robustness to noise. Next, we evaluate its performance in retrieval from a shape database of a few hundred watertight models. Finally, we compare it to other state-of-the-art shape descriptors, both pose-invariant and pose-dependent.

3.1. Pose Invariance

Gaussian curvature is a good starting point for a pose-invariant shape descriptor. Being an intrinsic measure, dependent only on the *metric* (i.e. the edge lengths) of the mesh, and not its embedding, it is invariant under *isometric transformations* - transformations which preserve edge lengths. Pose changes of articulated shapes are one example of such transformations, but also non-articulated shapes exhibit such transformations. For example, unfolding a developable surface to a plane is an isometric transformation, and it indeed preserves the (vanishing) Gaussian curvature during the transformation.

Unfortunately, standard discrete Gaussian curvature measures are too noisy to allow comparison using a simple histogram descriptor, since they are computed locally for each vertex. The conformal factor, on the other hand, despite being derived directly from the Gaussian curvature, depends also on the *metric* of the mesh (through the weights of the Laplacian matrix), Since the conformal factor is the result of a global computation on the entire mesh, it is much smoother than the discrete Gaussian curvature, and hence can be easily used in a simple histogram descriptor. Since the Gaussian curvature is invariant under isometric transformations, so is the conformal factor.

Figure 2 shows the conformal factors and their histograms, computed as described in the previous section, for

three models in different poses. Evidently the same model in different poses results in very similar histograms, and different models have quite different histograms – both in shape, and in the range of the values the conformal factor achieves.

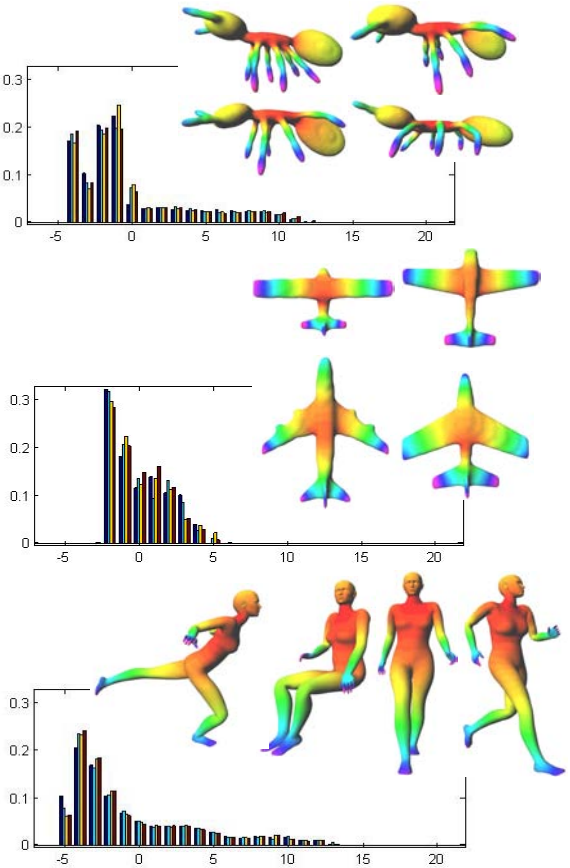


Figure 2: *Color-coding of conformal factors and histograms of the conformal factors, for three model classes in various poses (the values are normalized to the range* [0,1] *for the color-coding).*

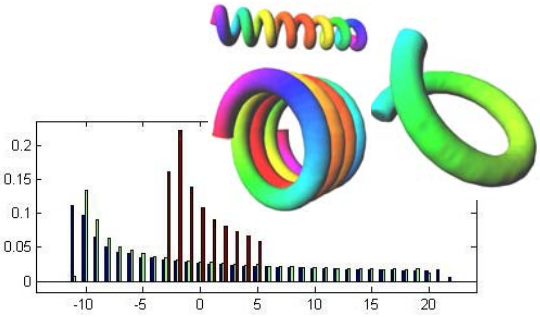


Figure 3: *Color coding of conformal factors, and the histograms of the conformal factors of three springs. The two springs of the same length have similar histograms. The shorter spring has a histogram with the same shape, but different range of values.*

Figure 3 shows the discriminative power of the conformal factor. The conformal factor tends to be positive on long extrusions - the longer the extrusion, the larger the conformal factor. This means that similar objects, which are not isometric transformations of each other – for example two springs of different length – will have similar-shaped histograms, but with different ranges of values. Hence, two springs of different length will be considered different. Normalizing the conformal factor to the range [0,1] will result in shapes with the same *number* of extrusions, but different extrusion *lengths* to be considered similar. This can be useful in certain cases, for example, for the springs from the watertight shapes benchmark [VtH07,GBP07], as can be seen in Figure 4. Unfortunately, for most models, normalizing the conformal factor may have undesirable effects. For example, chairs and humans might be considered similar, as they have more or less the same number of extrusions. Empirically, we have noticed that the un-normalized conformal factor performs better on most models.

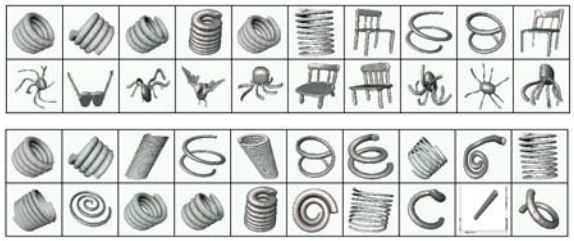


Figure 4: *Shape retrieval results from the watertight shapes benchmark* [VtH07,GBP07] *using the conformal factor (upper two rows) and the conformal factor normalized to* [0,1] *(lower two rows). In both cases the first model on the upper left is the query model. The database contains exactly 20 springs.*

3.2 Robustness to Noise

An important property of any shape descriptor is the ability to match similar shapes even in the presence of noise. Since our descriptor is based on Gaussian curvature, which is notorious for being sensitive to noisy geometry, verifying that our descriptor is robust is a major concern.

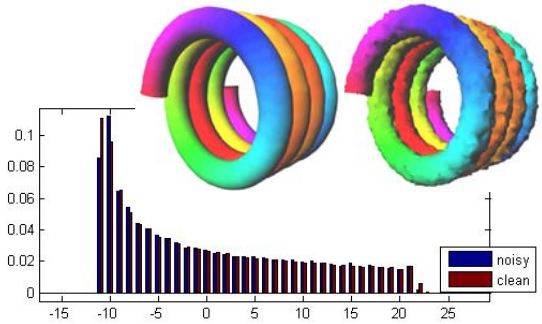


Figure 5: *Color-coded conformal factors (values normalized to the range* [0.1]*), and histograms of the conformal factors, for a spring model, and a noisy version of it.*

We contaminated the geometry of one of the springs from the watertight benchmark with Gaussian noise with $\sigma = 0.01$, which is about 20% of the mean edge length of the model. We then computed the conformal factor and its

histogram on the noisy spring, and compared it to the original. Figure 5 shows the results of this comparison – both the histogram and the color coding show that the difference between the signature of the original model and the noisy model is minor.

3.3 Sensitivity to Topology

A drawback of our descriptor is that it is sensitive to small topological differences. Since the descriptor is based on the conformal mapping to a surface with uniform curvature of the *same genus*, mapping between shapes of different genera is possible, but not necessarily successful. For example, the watertight shapes benchmark [VtH07,GBP07] contains two versions of the teapot, one of genus 0, and the other genus 1.

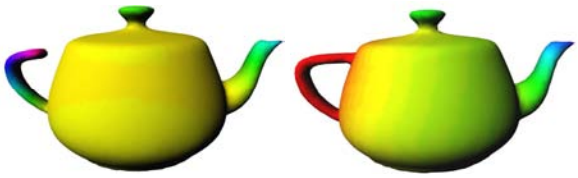
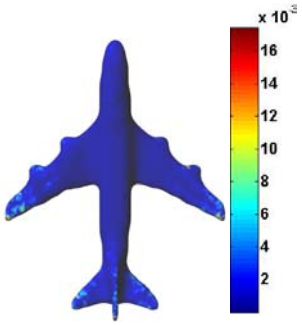


Figure 6: *The conformal factor for two similar models with a small topological difference.*

As Figure 6 shows, the distributions of the conformal factor of these models are quite different (as are their histograms). In this case, a simple visual-based descriptor (such as the light field descriptor [COTS03] for example) will do a better job. Of course, those two surfaces are not isometric transformations of each other. It remains to be seen whether this descriptor can be modified to be invariant to such genus changes.

3.4 Curvature Error

The discrete conformal factor computed by the linear method from [BGB08] is only an approximation of the true continuous conformal factor, and is accurate only for small curvature changes. Therefore, it is interesting to check the distribution of the error in the conformal factor over the mesh, and how it affects our shape descriptor. To test this, we computed the *curvature error*, as $|K^T - K^R|$, where K^T is the target curvature we wanted to obtain, as defined in section 2.2, and K^R is the obtained curvature, computed after scaling the original edge lengths with the computed conformal factor. The inset figure shows the distribution of this color-coded error over one of the meshes from the watertight shapes benchmark [VtH07,GBP07]. The average curvature error on all the mesh, defined as $\|K^T - K^R\|/|V|$ is 1.8e-5π. As the figure shows, the error is close to zero on most of the mesh, and is localized in areas where the difference between the original and target curvatures were relatively high. Hence, in practice, the error in the conformal factor resulting from the discretization will not have much effect on our shape descriptor.



3.5 The Watertight Shapes Benchmark

We tested our conformal factor (CF) descriptor on a large database of watertight meshes [VtH07,GBP07], and compared its performance with that of two state-of-the-art descriptors which are not pose-invariant: the spherical harmonics (SH) descriptor [KFR03] and light field (LF) descriptor [COTS03]. We also compared to the global point signature (GPS) - a more recent pose-invariant descriptor [Rus07], and to the augmented multiresolution Reeb graph (aMRG) – a pose invariant descriptor [TS05] which won the SHREC 2007 competition on the database of watertight meshes [VtH07,GBP07].

The implementations of SH and LF were taken from the respective Web sites of the authors. The comparison with GPS was based on the simulated search engine at the author's Web site. The comparison with aMRG was based on the results published in [GBP07].

The database contains 20 classes of objects, each containing 20 models. Some classes, e.g. the "ants" class, contain models which are indeed a result of quasi-isometric transformations. In these cases, our CF descriptor performs better than the non pose-invariant descriptors SH and LF, and almost as good as the pose-invariant GPS. On almost all classes however, the aMRG descriptor outperforms our descriptor. This superior retrieval performance comes with a price, as the aMRG descriptor is graph-based, and relatively computationally heavy both to compute and to compare. For example, the authors state [GBP07] that the computation time for the aMRG for a model containing 15,000 vertices is about 20 seconds on a 1.6GHz laptop. In contrast, computing our CF descriptor for the same model takes about 0.9 seconds on a 1.4GHz laptop. Hence, our descriptor can be efficiently computed even for relatively large meshes. Comparing two signatures is also much more efficient using our descriptor, as it is vector based, and requires only the computation of a simple L1 sum.

To quantitatively compare the performance of our descriptor with other shape signatures, we used a standard measure: the *precision/recall graph* [SMG83]. This describes the relationship between precision and recall in a ranked list of matches. For each query model in the class and any number *k* of top matches, “recall” (the horizontal axis) represents the ratio of models in the class returned within the top *k* matches, while “precision” (the vertical axis) indicates the ratio of the top *k* matches that are members of the class. A perfect retrieval result produces a horizontal curve (at precision = 1.0), indicating that all the models within the query object’s class are returned as the top ranked matches. In general, curves that stay close to the precision 1.0 mark represent superior retrieval results. Different points on the graph represent different values of *k*.

Figure 7 shows the precision/recall graph for the ants class of CF versus aMRG, SH and LF. Since the comparison with GPS was based on the results from the simulated search engine, this particular descriptor is not shown in the precision/recall graph. As discussed before, for the ants

class, our descriptor performs better than the non pose invariant descriptors, but worse than the aMRG descriptor.

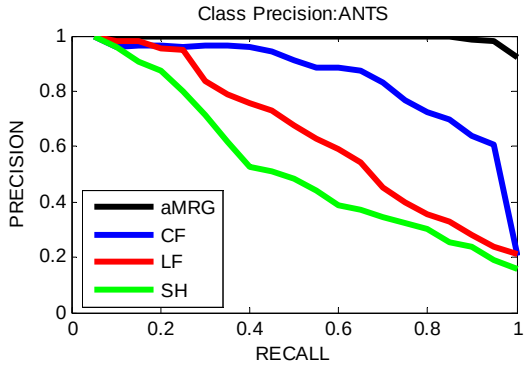


Figure 7: *Precision/recall graphs of various shape descriptors applied to the ants class in the watertight shapes benchmark* [VtH07,GBP07].

Figure 8 shows the query result for a specific ant using CF, GPS, LF and SH. The aMRG descriptor was not compared using query results. As each class contains exactly 20 models, showing the first 20 results of the query is quite informative.

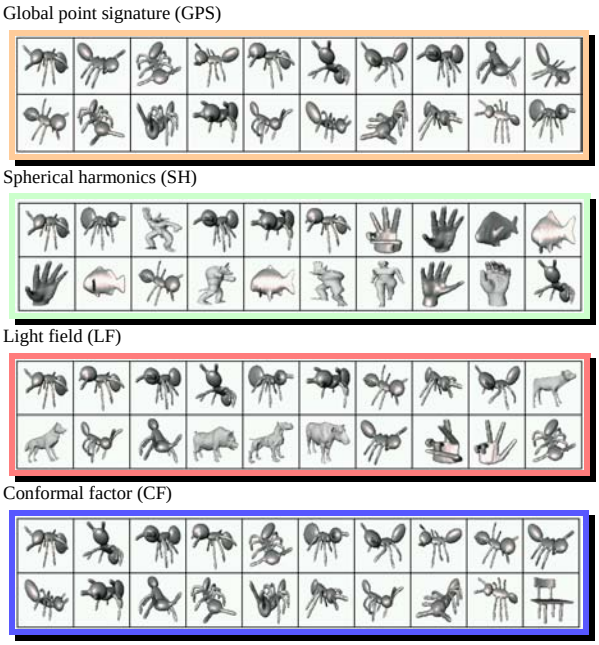


Figure 8: *Comparison of the retrieval of an ant in the watertight shapes benchmark* [VtH07,GBP07]. *The query model is in the upper left corner, followed by the retrieved shapes in decreasing order of similarity.*

The GPS descriptor manages to retrieve all the ants in the class, achieving a perfect precision/recall ratio of 1.0. Our CF retrieval is very similar (all ants except one were retrieved), yet the CF descriptor is much simpler to compute, requiring only the solution of a sparse symmetric linear system, instead of an eigenvector problem..

For most of the other classes, the results are not that clear cut. For example, in the armadillo class, our CF descriptor finds most of the deformations of the armadillo, which do not include missing parts. This is natural, since severing parts of a surface is not an isometric deformation. The aMRG descriptor, however, achieves the best performance on almost all the classes. Figures 9 and 10 show the precision/recall graph and the retrieval results for a representative model using the four descriptors, for the armadillos and heads classes.

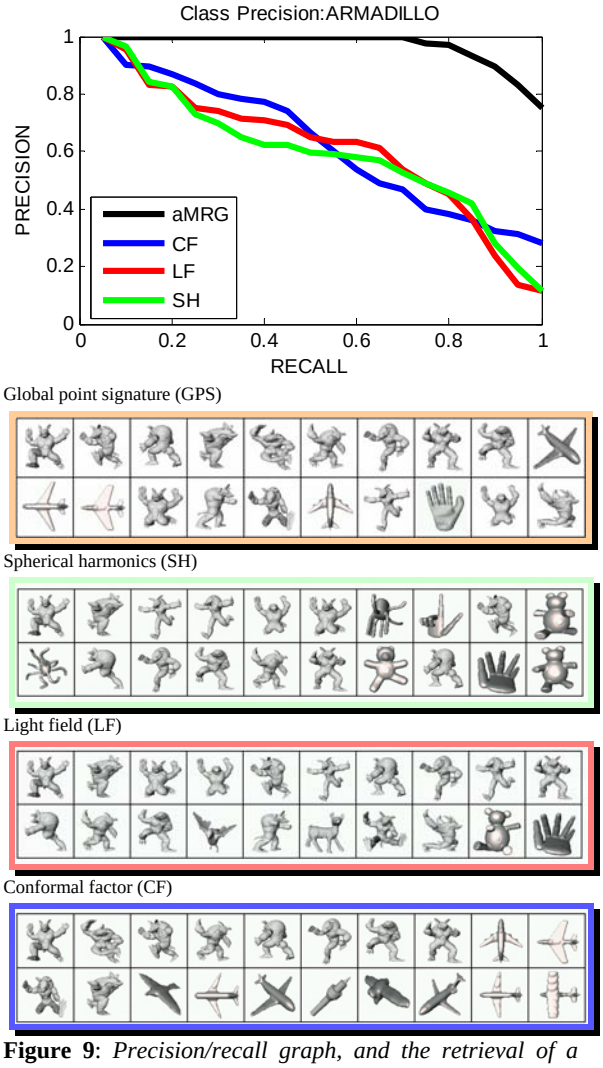


Figure 9: *Precision/recall graph, and the retrieval of a single model in the watertight shapes benchmark* [VtH07,GBP07] *for the armadillo class.*

Figure 11 shows some more precision/recall graphs for other classes. Figure 12 shows the results of sample queries, one query per each model class. Correct retrieval results (results from the same class as the query model) are colored green, and incorrect results are colored orange.

4. Conclusions and Discussion

We have presented a new shape descriptor, based on conformal geometry, which is invariant to isometric transformations. The descriptor is simple and very easy to compute and compare. It is based on solving a sparse linear set of equations, and does not require complex computations, such as computation of geodesic distances, solution of eigenvalue problems, or computing a Reeb graph. It performs comparably to state-of-the-art shape descriptors, both pose-dependent and pose-invariant.

Our descriptor has a few drawbacks. The most important is probably that the descriptor is applicable only to a manifold mesh. Otherwise, discrete geometric notions – such as the Laplace-Beltrami operator and the discrete curvature – are not well defined. As the majority of models available on the web are not manifolds, it is important to make this descriptor robust to non-manifold meshes. Another robustness issue is the problem of instability under local genus changes. As we conformally map between surfaces of the same genus, similar meshes of different genus might have a different signature.

A natural extension would be to use the conformal factor for full and partial shape matching. As the color coded meshes show, similar parts of shapes – for example the fingers of the armadillo model – match on different deformations of the mesh. We believe that the conformal factor, together with other intrinsic measures, may be a useful tool for such applications.

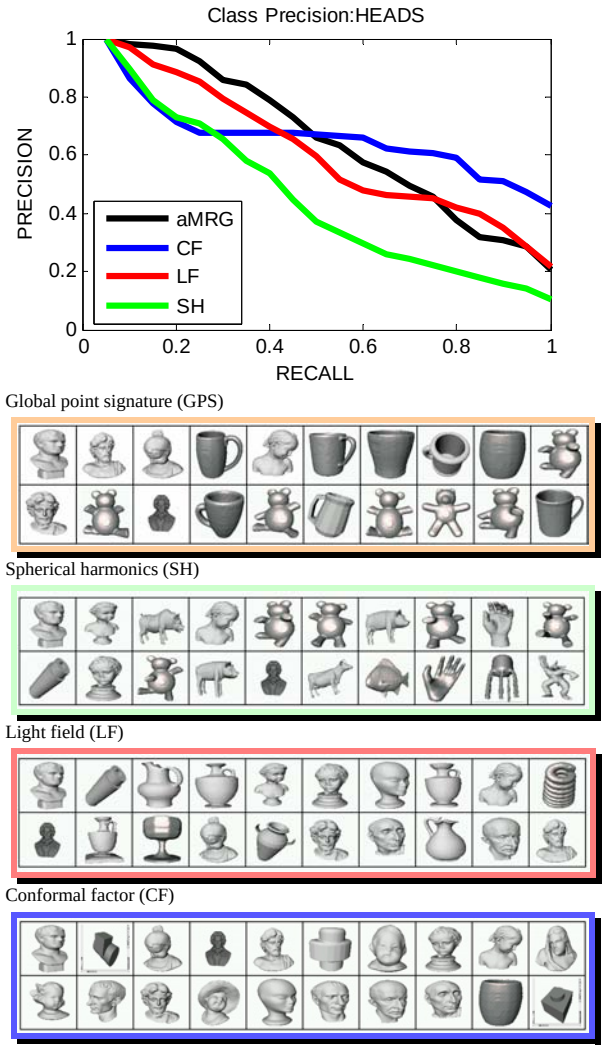


Figure 10: *Precision/recall graph, and the retrieval of a single model in the watertight shapes benchmark* [VtH07,GBP07] *for the heads class.*

References

[Abi81] ABIKOFF, W.: The uniformization theorem. *The American Mathematical Monthly*, 88, 8, (1981), 574-592.

[BGB08] BEN-CHEN, M., GOTSMAN, C., BUNIN, G.: Conformal flattening by curvature prescription and metric scaling. *Computer Graphics Forum*, 27, 2, (2008). .(*Proc. Eurographics*).

[COTS03] CHEN, D.-Y., OUHYOUNG, M., TIAN, X.-P., SHEN, Y.-T.: On visual similarity based 3D model retrieval. *Computer Graphics Forum*, (2003), 223–232.

[Dav05] DAVIS T.-A.: CHOLMOD Version 1.0 User Guide (http://www.cise.ufl.edu/research/sparse/cholmod). (2005).

[DBP06] DEL BIMBO, A., PALA, P.: Content-based retrieval of 3D models. *ACM Transactions on Multimedia Computing, Communications and Applications* 2, 1, (2006), 20-43.

[EK03] ELAD, A., KIMMEL, R.: On bending invariant signatures for surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, 10, (2003), 1285-1295.

[GBP07] GIORGI. D, BIASOTTI, S., PARABOSCHI, L.: Watertight models track, *Technical Report IMATI-CNR-GE 9/07*. http://watertight.ge.imati.cnr.it/watertight-global.pdf. (2007)

[GSCO07] GAL R., SHAMIR A., COHEN-OR D.: Pose oblivious shape signature. *IEEE Transactions on Visualization and Computer Graphics* 13, 2 (2007), 261–271.

[GV04] GU, X., VEMURI, B.- C.: Matching 3D shapes using 2D conformal representations. *Lecture Notes in Computer Science*, 3216 (2004), 771-780.

[HPW06] HILDEBRANDT, K., POLTHIER, K., WARDETZKY, M.: On the convergence of metric and geometric properties of polyhedral surfaces. *Geometria Dedicata* 123, 1, (2006), 89-112.

[JLYG07] JIN, M., LUO, F., YAU, S., GU, X.: Computing geodesic spectra of surfaces. *Proc. ACM Symposium on Solid and Physical Modeling* (2007), 387-393.

[JZ07] JAIN V., ZHANG H. A spectral approach to shape-based retrieval of articulated 3D models. *Computer Aided Design 39* (2007), 398-407.

[KFR03] KAZHDAN, M., FUNKHOUSER, T., RUSINKIEWICZ, S.: Rotation invariant spherical harmonic representation of 3D shape descriptors. *Proc. Symposium on Geometry Processing* (2003).

[MDSB02] MEYER, M., DESBRUN, M., SCHRÖDER, P., BARR, A.-H.: Discrete differential geometry operators for triangulated 2-manifolds. *Proc. International Workshop on Visualization and Mathematics* (2002)

[Rus07] RUSTAMOV, R.- M.: Laplace-Beltrami eigenfunctions for deformation invariant shape representation. *Proc. Symposium on Geometry Processing*, (2007)

[RWP05] REUTER M., WOLTER F.-E., PEINECKE N.: Laplace-spectra as fingerprints for shape matching. *Solid and Physical Modeling* (2005), 101–106.

[SMG83] SALTON, G., MCGILL, M.: *Introduction to modern information retrieval*, McGraw Hill (1983)

[SMKF04] SHILANE, P., MIN, P., KAZHDAN, M., FUNKHOUSER, T.: The Princeton shape benchmark. *Proc. Shape Modeling International*, (2004)

[TS05] TUNG, T., SCHMITT, F.: The augmented multiresolution Reeb graph approach for content-based retrieval of 3D shapes. *International Journal of Shape Modeling* 11, 1, (2005)

[TV08] TANGELDER, J.-W.H., VELTKAMP, R.-C.: A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications*, in press, (2008)

[VtH07] VELTKAMP, R.-C., TER HAAR, F.- B.: SHREC 2007 3D retrieval contest. *Technical Report UU-CS-2007-015.* http://www.cs.uu.nl/people/remcov/UU-CS-2007-15.pdf (2007)

[WCT05] WANG, Y., CHIANG, M.- C., THOMPSON, P.- M.: Mutual information-based 3D surface matching with applications to face recognition and brain mapping. *Proc. IEEE International Conference on Computer Vision*, (2005), 527-534.

[WWJ*06] WANG, S., WANG, Y., JIN, M., GU, X., SAMARAS, D.: 3D surface matching and recognition using conformal geometry. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2 (2006), 2453-2460.

[XHGC07] XIANG, P., HUA, C.- O., GANG, F.- X., CHUAN, Z.- B.: Pose insensitive 3D retrieval by Poisson shape histogram. *Lecture Notes in Computer Science*, 4488 (2007), 25-32.

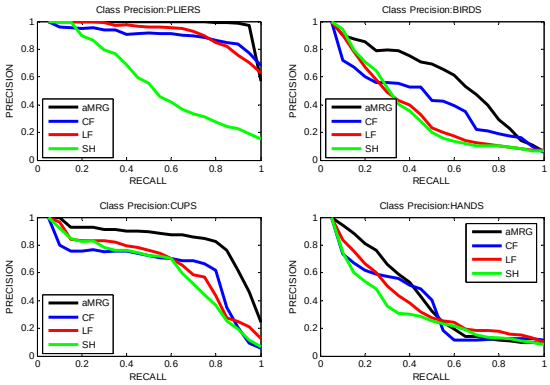


Figure 11: *Precision/recall graphs for some more classes from the watertight shapes benchmark* [VtH07,GBP07].

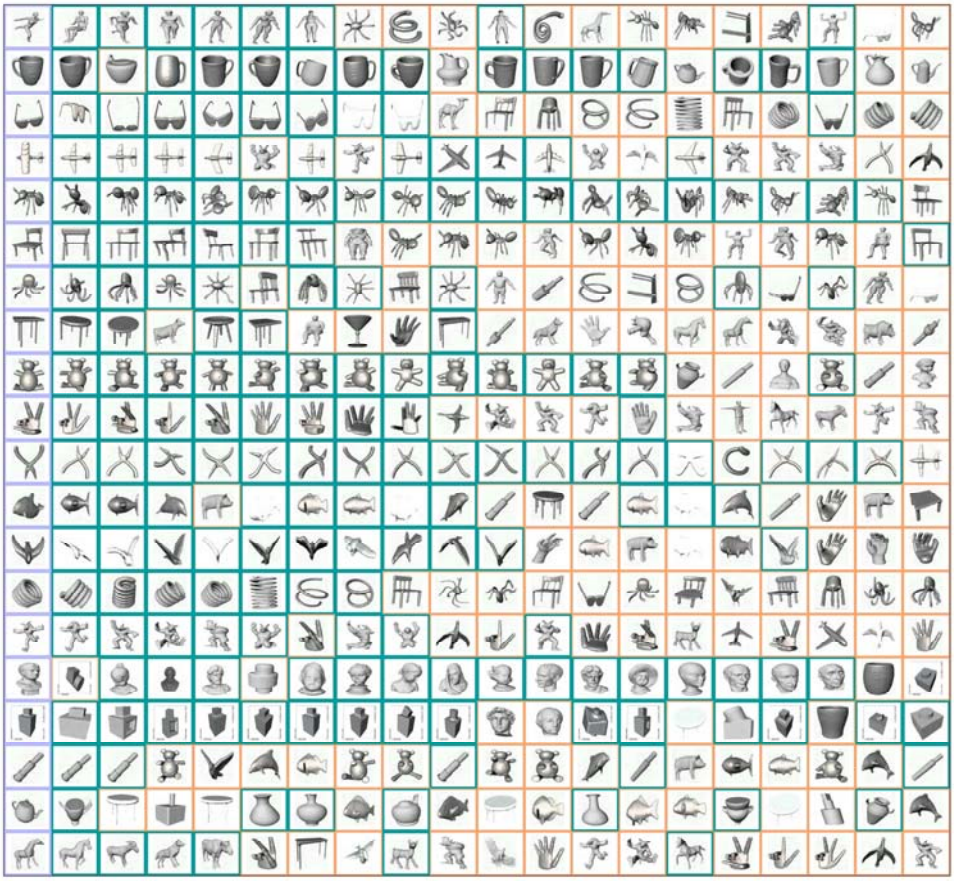


Figure 12: *Retrieval results for some query shapes, one from each class of the watertight shapes benchmark. Each row is the result of a single query. The leftmost shape (highlighted in purple) is the query image. Shapes highlighted in green are shapes from the correct class (the same as the class of the query shape). Shapes highlighted in orange are shapes from wrong classes.*