Using Image Descriptors to Guide Shape Analysis

Justin Solomon and Potcharapol Suteparuk
Department of Computer Science, Stanford University
justin.solomon@stanford.edu and neungs1@stanford.edu

Abstract

Considerable research has been dedicated to the computation of robust features for points in two-dimensional images. These features have been applied to a variety of tasks, including recognition, detection, and image processing, and have been shown to be robust under assorted image transformations. In this paper, we take advantage of these "mature" features to characterize localized shape at vertices of a triangular mesh. Rather than computing geometric features directly from mesh shape, we render the mesh and pull back two-dimensional features from the resulting image to the mesh geometry. These features capture the visual appearance of a surface and are provably robust to certain choices of camera position and parameter. We test our proposed descriptor for tasks in geometry processing such as surface mapping.

1. Introduction

The field of computer vision has generated several effective descriptors for localized image structure. These descriptors, such as SIFT and SURF [16, 2], accumulate intensities, gradients, and other information that describe shapes and patterns in an image more effectively than individual pixel colors. Such descriptors form the basis for diverse vision techniques at both low and high levels, from image alignment to recognition; this variety of applications demonstrates their robustness to variation both in image appearance and to some extent over object deformation.

Tools in the field of geometry processing make use of localized descriptors to characterize the shape of a surface rather than its appearance in images. Shapes are most commonly represented using polyhedral meshes, and descriptors including [7, 23] are given as per-vertex values characterizing curvature, heat diffusion structure, and other properties. The formulation of shape descriptors that are stable under both smooth surface deformation as well as the addition of noise is an open problem requiring insight from differential geometry, machine learning, and vision; currently, descriptors usually are proven to behave well under limited conditions like isometry, conformality, or Gaussian vertex noise.

Given the relatively successful applications of image operators even to tasks in which the underlying scene has three-dimensional structure, in this paper we explore the possibility of generating a shape descriptor by pulling back image-based descriptors. In particular, we generate per-vertex SIFT descriptors by rendering the mesh near a given vertex in a small patch, computing per-pixel descriptors of the resulting image using [15], and then averaging the descriptors over the patch to increase stability. We evaluate our technique using standard mesh correspondence and analysis datasets.

2. Background

Our problem of interest is that of mesh analysis and correspondence. Considerable research in both vision and geometry has been dedicated to this task with an eye on applications including symmetry detection [17], registration for motion capture [14], feature point detection [18], segmentation [10], and so on. A common theme in popular approaches to these problems is the use of per-vertex descriptors. For instance, mapping techniques including [8, 22] optimize energy terms trading off between descriptor agreement and map continuity or elasticity. Analysis of a surface embedded into descriptor space also can be used for symmetry detection by finding points with similar descriptors [19].

Given the wide-spread usage of per-vertex descriptors for matching, a variety of geometric descriptors have been proposed. Taking inspiration from differential geometry, Gaussian, mean, and/or principal curvatures provide a complete characterization of metric geometry at a differential scale [4]; unfortunately, these descriptors can be difficult to compute reliably on a discrete surface [11]. Other PDE-based descriptors, such as the heat and wave kernel signatures [23, 1], are smoother, multiscale, and *intrinsic*, mean-

Future Distribution Permission: The author(s) of this report give permission for this document to be distributed to Stanford-affiliated students taking future courses.

ing that they are robust to surface deformations that preserve distances along the surface. Unfortunately, these descriptors depend on the global geometry of the surface, and their computation requires stable spectral analysis and geometric Laplacians.

Nearly any nontrivial deformation of a generic surface is non-isometric–meaning intrinsic geometry is not preserved–and indeed many elastic deformations observed in nature are far from isometric. Furthermore, tasks mapping between diverse objects within a class may expect global or volumetric rather than differential-geometric structure to be preserved from shape to shape. In these cases, extrinsic descriptors may be relevant to the mapping task. For instance, spin image, shape context, and shape diameter descriptors attempt to characterize the volume of the interior of a surface near a given feature point [9, 13, 7], which slippage descriptors cluster points based on their contact with simple surfaces [3].

Contrastingly, descriptors for points in two-dimensional images must deal with the fact that the underlying geometry has been projected to the image plane. Since inferring depth is a difficult and in many cases ambiguous, these descriptors are designed to be invariant to certain underlying scene changes, such as rigid motion, scaling, and rotation. Furthermore, the variety of patterns and high-frequency detail likely to appear in a photograph arguably is higher than that found on a typical piece of meshed geometry, and thus these descriptors can deal with not only geometric deformations but also some degree of noise in the image itself.

Although our technique in theory could be used to pull back *any* localized image descriptor to vertices of a mesh, we are unable to summarize the rich field of image descriptors here. In our experiments, we make use of the popular SIFT descriptor, first introduced in [16]. SIFT is designed to be robust to changes in scale and rotation, making it possible for our method to position cameras without a canonical orientation beyond eye direction. Furthermore, SIFT captures information about gradients near a given pixel, helping characterize the shape of an object in front of the camera, including its smooth structure and sharp contours.

The idea of rendering a shape and using it as a *localized* descriptor has not been explored in depth, but work on the "lightfield" descriptor provides an analogous approach for describing global geometry [6]. The lightfield descriptor renders a shape from a variety of viewpoints located at the vertices of a polyhedron enclosing the shape in question and uses properties of the resulting image to characterize the shape for applications in search. In this way, our approach might be considered a localized version of the lightfield descriptor, although we make use of not only silhouettes but also shading to characterize smooth geometry in addition to edges.

3. Technical Approach

Our approach to generating per-vertex SIFT descriptors takes place in a number of straightforward steps. We summarize and discuss each in an individual section below. Note that the first two steps below occur independently for each vertex on the mesh and thous easily could be implemented in parallel.

3.1. Input

We take as input a triangular mesh Σ representing a piece of geometry, given as a pair (X, T) of vertices $X = \{x_i\}_{i=1,\ldots,n}$ and triangles $T = \{t_i\}_{i=1,\ldots,m}$. Our goal is to associate each x_i a descriptor $d_i \in \mathbb{R}^k$ characterizing geometry and surface appearance near that point.

For convenience we also compute per-vertex mesh normals. To obtain the normal N_i for x_i, we simply average the normals of its adjacent faces and resize the resulting vector to length one. Although this strategy is not robust to noisy surfaces, for our relatively smooth tests it behaves acceptably.

3.2. Rendering

Since we wish to design d_i to capture the appearance of Σ near x_i, we take the most straightforward possible approach and simply render the surface near each vertex.

Given a user-provided parameter δ, we compute a camera position for each vertex x_i as the displacement

$$p_i = x_i + \delta N_i$$

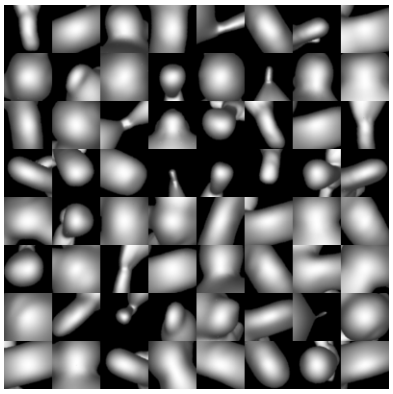
with gaze direction $-N_i$. We choose a camera "up" direction arbitrarily.

The introduction of parameter δ and arbitrary choice of camera orientation initially may appear to be considerable drawbacks of our approach. In reality, however, we make them with the application in mind. In particular, displacing the camera and/or changing its "roll" simply rotates or scales the resulting rendered image. Since SIFT descriptors are robust to such changes, we do not find a significant dependence on these choices.

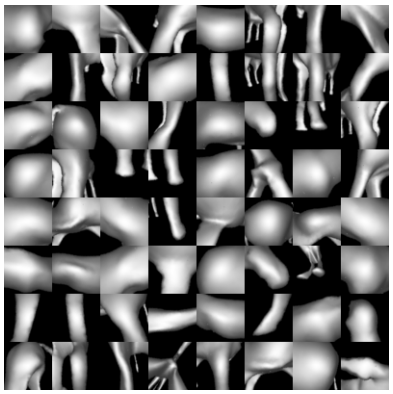
Since we wish to capture not only projection topology but also curvature, we apply a smooth shading model to each of the rendered images. In particular, we position a single undirected light source at p_i. Each surface is shaded using the Phong model, which computes intensities using the following formula [20]:

$$I = k_a i_a + k_d (L \cdot N) i_d + k_s (R \cdot V)^\alpha i_s$$

Here, the material parameters are given by ambient, diffuse, and specular components (k_a, k_d, k_s), and parallel components for the light source are given by (i_a, i_d, i_s); a parameter α controls the specularity of the surface. Unit vector L



(a)



(b)

Figure 1. Example image patches used to generate per-vertex SIFT descriptors for a (a) teddy and a (b) donkey model.

points toward the light source, N is a given surface normal, R is the reflection of L over N, and V points toward the viewer.

We purposefully chose the Phong model for a number of reasons. First of all, it is easy to evaluate and implemented in a number of hardware shaders, should one choose to implement our technique for time-critical applications; this is a strong advantage over spectral descriptors like [23, 1], which require much more involved analysis. The diffuse component of the image captures surface curvature near x_i. In particular, principal curvatures and other properties can be written as derivatives of the normal vector N along the surface, so in some sense our descriptor is attempting to formulate robust curvatures. The reflected term helps capture geometry farther from the center vertex, since the reflected lobe can be displaced from x_i.

Practically, we use the RenderMan package to generate a 50×50 rendered patch for each vertex. We use $k_a = k_d = k_s = 1$ and the same for the light source, with $\alpha = 0.1$. Figure 1 shows example image patches for teddy bear and donkey models.

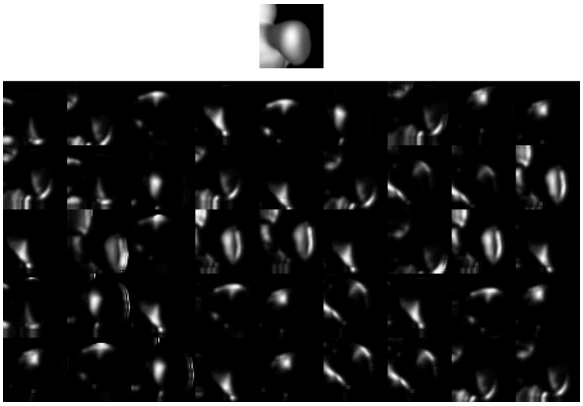


Figure 2. Assorted bins of the descriptor [15] for the 50×50 patch above. There are 128 bins total.

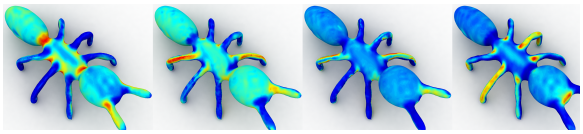


Figure 3. SIFT bins 1, 24, 100, and 127, resp., shown as scalar functions over a surface. Here, scalar values are colored between blue (lowest) and red (highest).

3.3. Generation of Image Descriptors

With each mesh vertex x_i, the strategy from §3.2 yields a 50×50 patch I_i showing a rendering of the local neighborhood about x_i. We cannot use the patch itself as a descriptor, since doing so would require the choice of a canonical camera position, orientation, and so on. Thus, we first must lift to a descriptor space where values are more likely to align.

As suggested earlier, we use the SIFT descriptor [16] to characterize the localized structure of the vertex images. Invariance of SIFT to scale and orientation resolve issues in our algorithm regarding choice of camera position and orientation. Furthermore, SIFT's success in image processing tasks involving photographs may indicate its resilience to structured noise occurring in realistic scenes and scans.

The original algorithm proposed in [16] only makes use of SIFT to describe key points that already have been established in a given image as important to the visible shapes and structures. In our application, we are unable to make such choices, since the point of interest x_i by definition is at the center of I_i. Thus, instead we make use of the per-pixel SIFT feature computations proposed in [15] to compute one SIFT descriptor per pixel of I_i, making a total of 250 descriptor vectors in total as illustrated in Figure 2. We then average the descriptors over the patch to derive the descriptor for x_i. Some bins of the SIFT descriptor shown as functions over a mesh are shown in Figure 3.

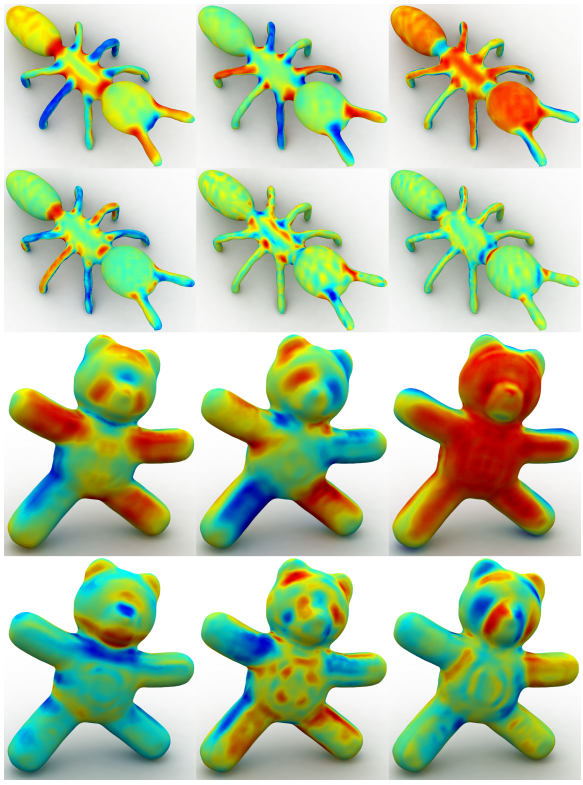


Figure 4. The first six PCA dimensions obtained by analyzing the per-vertex SIFT descriptors. Each instance of a given shape shows a single PCA dimension as a function of the surface position.

3.4. Dimension Reduction

Given the dimensionality of the SIFT descriptor, at this point we have associated each vertex x_i with a descriptor $d_i \in \mathbb{R}^{128}$. It is likely that many of these 128 dimensions are not strongly correlated with mesh geometry in any way and that some represent noise and artifacts due to the two- rather than three-dimensional computations involved in deriving SIFT. This phenomenon is illustrated in Figure 3, in which some bins of the SIFT descriptor are clearly more informative than others, and individual bins exhibit apparently random noise or variation across the surface.

For this reason, we perform principal components analysis (PCA) on the collection $\{d_i\}_{i=1,\ldots,n}$ and replace each descriptor with a lower-dimensional $\tilde{d}_i$ projected onto the most prominent PCA eigenvectors. This process helps reduce noise and highlight the most relevant parts of the descriptors for geometry rather than image appearance.

Figure 4 shows examples of the most prominent PCA dimensions on two contrasting example surfaces. It is immediately clear that the dimension-reduced SIFT descriptors provide a very different characterization of surface geometry than intrinsic approaches. For instance, the descriptor strongly disambiguates the left and right sides of the two models; descriptors based on intrinsic geometry cannot possibly provide such disambiguation since they would require the choice of a canonical orientation. Figure 5 shows a typical set of singular values for the PCA problem carried out on the per-vertex SIFT values. Empirically we find the first six to nine PCA eigenvectors contain the vast majority of information and discard the remaining dimensions.

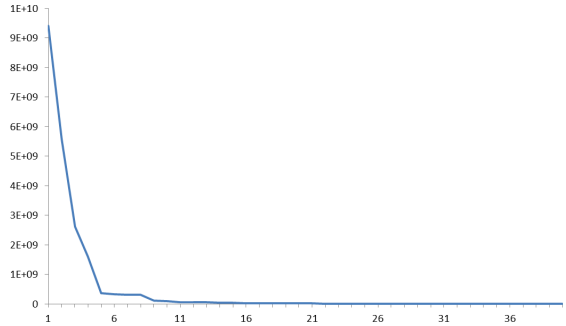


Figure 5. The first 40 singular values corresponding to the PCA on SIFT descriptors from the ant model in Figure 4.

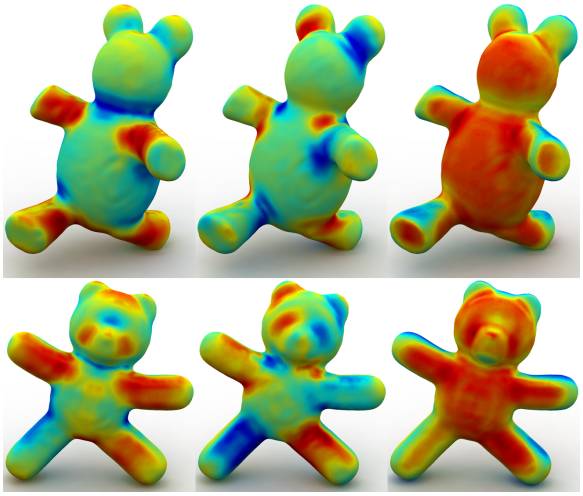


Figure 6. PCA technique applied simultaneously to two models for mapping; here each column represents a separate PCA vector.

Most encouragingly, however, it is important to note that the descriptors resulting from projection onto the first few PCA eigenvectors appear to be continuous functions of position along the surface. Assuming such a phenomenon *a priori* would not be permissible: there likely is some random noise in the choice of camera position, SIFT uses discrete bins, PCA is carried out in descriptor space with no direct knowledge of mesh geometry or topology, and so on. This observation is encouraging not only from a theoretical but also from a computational standpoint; unlike eigenvector computations or linear solves used to obtain spectral surface descriptors, our SIFT descriptor can be obtained in parallel per vertex.

Of course, each mesh is likely to exhibit unique variability among its per-vertex SIFT descriptors, and thus in some

cases the resulting PCA dimensions can be considerably different. If we wish to apply our descriptors to mapping between surfaces, however, we must ensure that the resulting dimensions align. To overcome this issue, we simply perform PCA on the set of descriptors from *both* surfaces simultaneously rather than one at a time. For substantially similar surfaces, the resulting projections are not appreciably different. Figure 6 shows one example of the mapped descriptors; by visual inspection it is clear that the global structures of the descriptors is preserved, highlighting the same front/back and left/right pieces on both surfaces.

4. Experiments and Results

Here we discuss some metrics we have used to evaluate the quality of our mesh descriptor for common geometric problems. Note that individualized examples illustrating steps of our algorithm are included within the text of §3.

4.1. Dataset

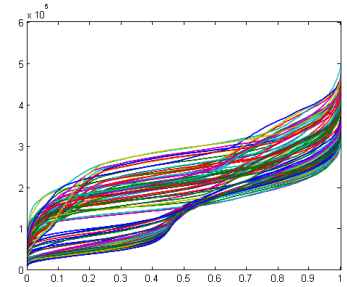
The qualitative and quantitative results in this paper are generated using the Surface Correspondence Benchmark database first introduced in [12].[1] The benchmark combines triangle mesh data from three sources, but for consistency we focus on meshes from "watertight shape retrieval" subset. Note that while the meshes in this particular database are all manifold and watertight, our algorithm only depends on the existence of a smooth normal field N and a technique for rendering, so in theory it could be applied not only in this case but also to implicit representations, point clouds, and other modes of shape data.

It is important to mention here that the problem of evaluating a surface shape descriptor is ambiguous, with limited ground truth data available. It is outside the scope of this paper to generate comprehensive novel techniques for testing descriptor matches, especially in the absence of dense ground truth mappings, but we show some straightforward heuristics to help indicate when our descriptors are effective and when they fail to identify a given point.

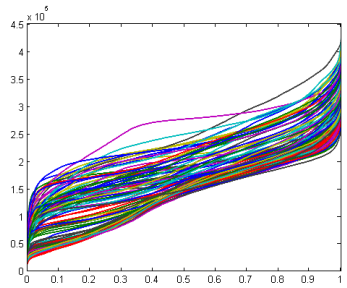
4.2. Uniqueness

A descriptor is useful only if it discriminates one vertex from another along a single surface; if the descriptor is uniform, it provides no useful information about geometry.

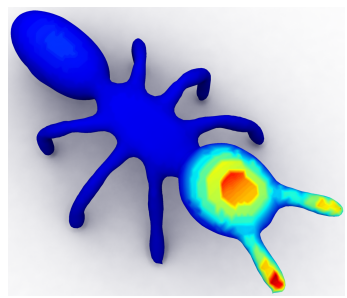
For a given surface, we can analyze how discriminative a descriptor is by examining the distribution of distances between different vertices' descriptors. Figure 7 shows one such experiment; although the results are for two meshes, we find that they exhibit similar patterns on other surfaces. In this plot, each curve represents a single vertex (we randomly select 100 examples) on a given surface. The plot shows distance from that vertex's descriptor to all others on the mesh as a function of percentile; this function is increasing, and the point at $x = 1$ represents the distance to the farthest descriptor. We see that descriptor distances are nonlinearly but fairly evenly distributed, and in particular that there are relatively few vertices whose descriptor distance to the given central vertex are close to zero. This experiment thus shows that our descriptors are unlikely to repeat along a surface, and that the places they are self-similar correspond generally to symmetric points.



(a)



(b)



(c)

Figure 7. Distance between vertex SIFT descriptors for the (a) ant and (b) teddy models. Each curve represents a single vertex from the ant mesh in Figure 4. The curve shows descriptor distance between the descriptor at that vertex as a function of percentile. An example match function is shown in (c); an Gaussian distance to the descriptor of the tip of the front right antenna is shown as a function across the surface (the best matches are on the antennae and head).

[1] http://www.cs.princeton.edu/∼vk/CorrsCode/Benchmark/

4.3. Suitability for Mapping

In the previous section we have verified that our descriptor is *discriminative*, that is, that it identifies each point on the surface uniquely. While this property may be sufficient for understanding the structure of an object, as in descriptor-based symmetry detection [17], using the descriptor to understand a collection of shapes requires that it is preserved from shape to shape.

Fortunately, the Surface Correspondence Benchmark discussed in §4.1 contains not only a collection of surfaces but also some sparse ground truth correspondences between sufficiently similar instances. These ground truth correspondences are not dense but can be used to find some indication of descriptor preservation between semantically similar meshes.

Figure 8 shows the results of experiments run on pairs of teddy, ant, and human models from the Correspondence Benchmark. In this figure, false matches more prominent than the ground truth correspondence are given by negative y values. Overall, these graphs indicate that the per-vertex post-PCA SIFT descriptors generally yield a match within the upper quartile of potential target vertices but beyond this provide limited guidance for matching. Figure 9 provides an alternative strategy for analysis by showing the ratio of the distance between ground truth matched descriptors to the distance to the closest match to a given descriptor.

The experiments provide some indication of when our algorithm performs well and when other methods may be more useful. Within the test cases we are able to try, it is easy to see that the descriptor was most effective on the ant models and least effective matching male and female human models. Such a difference seems reasonable when considering our methodology for collecting descriptors. In particular, the differences between the male and female models can manifest themselves as differences in the normal vector field along the surfaces since the two have considerably different curvature; this difference affects the choice of per-vertex camera positions, which in turn determines the images for analysis. The ant models have more similar structure and thus are less susceptible to this variability.

Overall, while the SIFT-based descriptors are strongly discriminative, they do not map well from surface to surface and can yield a significant number of false matches. The addition of a continuity term as in [22] might improve these results by culling obviously false matches, but alone the descriptor is too weak to use directly for surface mapping tasks.

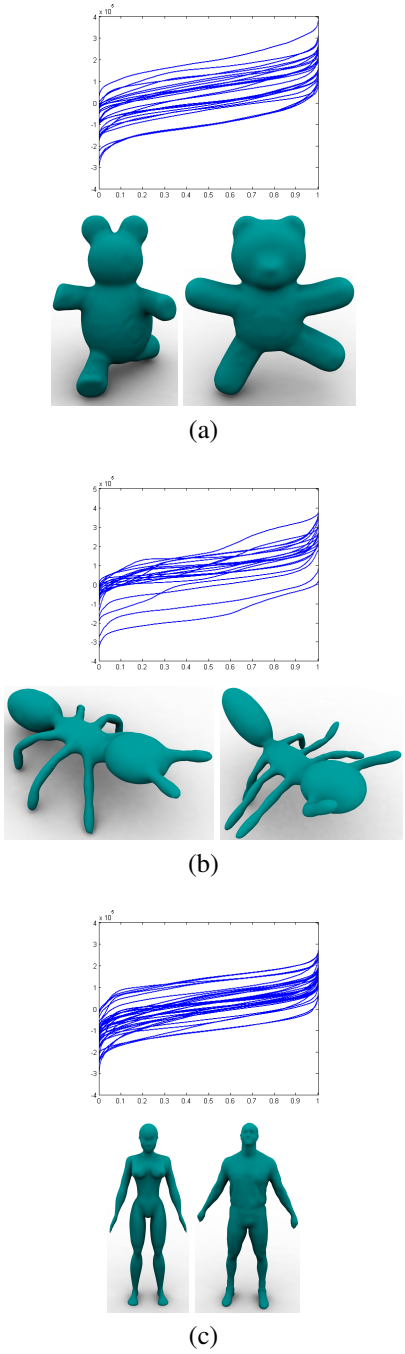


Figure 8. Descriptor distance graphs for matching (a) teddy, (b) ant, and (c) human models. Here, each curve represents a ground truth correspondence provided in the Correspondence Benchmark. As in Figure 7, the vertical axis corresponds to descriptor distance, this time between the descriptor at the source vertex of the given correspondence and every possible vertex on the target model. The curve is offset vertically by the descriptor distance to the desired target vertex from the Benchmark; thus, negative values represent false matches.

5. Discussion and Conclusion

We have introduced a technique for computing per-vertex mesh descriptors making use of two-dimensional image features via rendering. By making use of the SIFT fea-

Models	Accuracy ratio
Teddies	4.1069
Ants	5.2197
Humans	3.9672

Figure 9. Ratio of the distance to the ground truth target descriptor to the closest descriptor match, averaged over the ground truth matches given by the Correspondence Benchmark.

ture [16], we remove sensitivity due to choice of camera position relative to each mesh vertex. This technique, when combined with PCA projection, yields a set of descriptors that is relatively stable across a surface despite no explicit continuity term. Additionally, our descriptor is straightforward to compute using existing rendering and 2D image analysis tools, and the nontrivial computations can be parallelized per-vertex and–in the rendering step–even per-pixel. While in this paper we have examined a particular choice of shading models, camera parameters, and 2D image descriptors, in reality the method we describe can be regarded as a *framework* for applying image-based descriptors to surface matching in a straightforward way.

5.1. Future Research

While the coherency of our mesh descriptors shows promise for the generation of robust features, at this point the results in §4 indicate that considerable additional work remains to be done before per-vertex SIFT can be applied to practical geometric problems. Within our framework, a number of parameters and engineering decisions might be reexamined to formulate the most robust possible descriptor:

- Most prominently, the SIFT [16, 15] descriptor is by no means the only 2D image descriptor that may be applicable here. For instance, the HOG [5] and SURF [2] feature papers may yield interesting per-pixel descriptors used in their applications to find feature points.
- While we have argued that SIFT allows our feature to be robust to choices of camera scale and rotation, SIFT is *not* affine-invariant and thus has some dependence on camera projection. We could experiment with different types of projections or use a canonical orthogonal projection to find which one yields the best results.
- We have argued in §3.2 that the Phong shading model [20] is sufficient for capturing some surface structure and curvature information. Other shading models and textures, however, may give more detailed geometric characterizations of a surface.

Experiments testing each of the above parameters and tool choices may be facilitated by the implementation of a faster rendering pipeline, e.g. using hardware rendering tools to generate patch images I_i in parallel.

More generally, the technique presented here represents one application of vision tools to problems traditionally considered part of computer graphics. Although graphics and vision obviously deal with related problems, techniques that are informed from developments in both fields are only appearing recently, at least within geometry processing. Our method is inspired by methods in vision like [21] that use rendering to generate vision training data, but contrastingly it uses renderings not just for synthetic training data but instead to inform a more difficult graphics process. We hope that this model strategy can serve as inspiration for algorithms making stronger use of approaches from both fields simultaneously.

References

[1] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *ICCV Workshops*, pages 1626–1633, nov. 2011.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Proc. CVPR*, 110(3):346–359, June 2008.

[3] M. Bokeloh. *Slippage Features*. Wilhelm-Schickard-Inst. für Informatik, 2008.

[4] M. D. Carmo. *Differential geometry of curves and surfaces*. 1976.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[6] T. Funkhouser and M. Kazhdan. Shape-based retrieval and analysis of 3d models. In *SIGGRAPH 2004 Course Notes*, 2004.

[7] R. Gal, A. Shamir, and D. Cohen-Or. Pose-oblivious shape signature. *TVCG*, 13(2):261–271, March-April 2007.

[8] Q.-X. Huang, B. Adams, M. Wicke, and L. J. Guibas. Non-rigid registration under isometric deformations. In *Proc. SGP*, pages 1449–1457, 2008.

[9] A. Johnson. *Spin-images: A Representation for 3-D Surface Matching*. Technical report. Carnegie Mellon University, The Robotics Institute, 1997.

[10] E. Kalogerakis, A. Hertzmann, and K. Singh. Learning 3d mesh segmentation and labeling. *ACM Trans. Graph.*, 29(4):102:1–102:12, July 2010.

[11] E. Kalogerakis, P. Simari, D. Nowrouzezahrai, and K. Singh. Robust statistical estimation of curvature on discretized surfaces. In *Proc. SGP*, pages 13–22, 2007.

[12] V. G. Kim, Y. Lipman, and T. Funkhouser. Blended intrinsic maps. *TOG*, 30(4):79:1–79:12, July 2011.

[13] M. Körtgen, G. J. Park, M. Novotni, and R. Klein. 3d shape matching with 3d shape contexts. In *Central European Seminar on Computer Graphics*, Apr. 2003.

[14] H. Li, R. W. Sumner, and M. Pauly. Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum*, 27(5), July 2008.

[15] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 33(5):978–994, May 2011.

[16] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, volume 2, pages 1150–1157, 1999.

[17] N. J. Mitra, L. J. Guibas, and M. Pauly. Partial and approximate symmetry detection for 3d geometry. *ACM Trans. Graph.*, 25(3):560–568, July 2006.

[18] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. Guibas. One point isometric matching with the heat kernel. In *Proc. SGP*, 2010.

[19] M. Ovsjanikov, J. Sun, and L. Guibas. Global intrinsic symmetries of shapes. In *Proc. SGP*, pages 1341–1348, 2008.

[20] B. T. Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, June 1975.

[21] L. Pishchulin, A. Jain, C. Wojek, M. Andriluka, T. Thormahlen, and B. Schiele. Learning people detection models from few training samples. In *CVPR*, pages 1473–1480, june 2011.

[22] J. Solomon, A. Nguyen, A. Butscher, M. Ben-Chen, and L. Guibas. Soft maps between surfaces. *Comp. Graph. Forum*, 31(5):1617–1626, Aug. 2012.

[23] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Proc. SGP*, pages 1383–1392, 2009.

6. Appendix: Contributions

Both authors have an ongoing interest in geometry processing applications: Justin is a graduate student in the geometric computing group, and Potcharapol worked there last summer. While we have general knowledge about this field (which helped generate the "background" section), the idea for this project was entirely based on discussions resulting from CS 231a's lectures on low-level vision. Relevant references in our field have been cited as necessary within the text of this paper. This project is being used for no other courses.

Both authors contributed to both coding and testing the algorithm at hand. In particular, Potcharapol implemented tools for rendering the mesh near each vertex including calls to Renderman and wrote code for testing the descriptors using the Princeton benchmark and other data for ground truth; Justin handled the 2D vision components as well as conversion from image descriptors to mesh descriptors (and wrote much of the paper text). The tests and experiments were designed together.