

Tech report WUCSE-2006-22: Feature Detection Using Curvature Maps and the Min-Cut/Max-Flow Graph Cut Algorithm

Timothy Gatzke¹ and Cindy Grimm¹

Washington University in St. Louis, St. Louis MO 63130, USA

Abstract. Automatic detection of features in three-dimensional objects is a critical part of shape matching tasks such as object registration and recognition. Previous approaches to local surface matching have either focused on man-made objects, where features are generally well-defined, or required some type of user interaction to select features. Manual selection of corresponding features and subjective determination of the difference between objects are time consuming processes requiring a high level of expertise. Curvature is a useful property of a surface, but curvature calculation on a discrete mesh is often noisy and not always accurate. However, the *Curvature Map*, which represents shape information for a point and its surrounding region, is robust with respect to grid resolution and mesh regularity. It can be used as a measure of local surface similarity. We use these curvature map properties to extract features and segment the surface accordingly. Although thresholding techniques can be used to generate reasonable features, the choice of a threshold is very subjective and the results may be very sensitive to this choice. To avoid the threshold dilemma and to make the selection of the feature region less subjective, we employ a min-cut/max-flow graph cut algorithm, with vertex weights derived from the curvature map property. A multi-scale approach is used to minimize the dependence on user defined parameters. We show that by combining curvature maps and graph cuts in a multi-scale framework, we can extract meaningful features in a robust way.

1 Introduction

Automatic detection of features in three-dimensional (3-D) objects is critical for tasks such as object registration and recognition. For example, identifying corresponding regions between two similar surfaces is a necessary first step toward alignment and registration of those surfaces. A fundamental question is: What constitutes a feature? Man-made objects often have well-defined features such as edges, but features on natural shapes, such as the wrist bones shown in Figure 1, are more subjective. Furthermore, such shapes can have subtle variations, the importance of which may not be obvious.

We aim to detect subtle shape features in a robust way with a fully automated process. The types of features we expect to be useful are peaks, pits, ridges, and valleys. Important features may be of various sizes within one object. These features may or may not be unique, as long as there are enough features to resolve any ambiguities during shape matching.

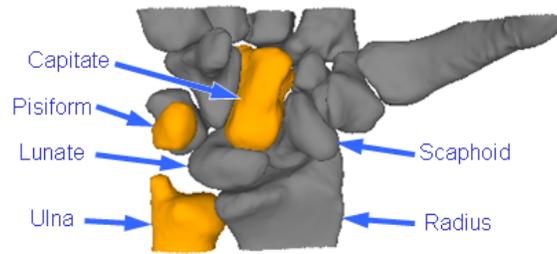


Fig. 1. Bones making up the human wrist. Natural objects have subtle shape variations that are challenging to characterize.

Previous approaches to local surface matching have either focused on man-made objects, where features tend to be more well-defined, or required some type of user interaction to select features. Manual selection of corresponding features and subjective determination of the difference between objects are time consuming processes requiring a high level of expertise. In contrast, our approach is entirely automatic.

Several two-dimensional techniques have been developed for medical imaging, image analysis, and computer vision. However, advances in 3-D scanning capability are providing ready access to 3-D data. In order to make effective use of this data, more advanced analysis methods, including automated feature detection, are required. It is desirable for feature detection to be consistent, robust, independent of the mesh resolution, and relatively insensitive to noise. In addition, we would like to eliminate the need for user interaction during the process, as well as dependence on parameter tuning. On the other hand, we do not require detection of every feature (indeed, we cannot even define every feature). Rather, we wish to find some set of geometrically interesting regions that is sufficient for shape matching tasks.

1.1 Approach

In this paper we present a feature detection algorithm based on the *Curvature Map*, combined with a robust segmentation approach in a multi-scale framework. The curvature map at a point represents shape information for the point and its surrounding region. A min-cut/max-flow graph cut algorithm, popular for image segmentation tasks, is employed to identify features at various scales. Results from multiple cuts are combined in a novel manner to produce a final feature set. The multi-scale approach eliminates the need for user interaction, and for tuning parameters based on a particular application.

The proposed algorithm is robust to noise and mesh variations. The process is automatic, with no user controlled parameters. We demonstrate the algorithm on several shapes represented by triangular meshes. These shapes include a test shape with and without noise, bones from the wrist, data from face scans, and the Stanford bunny. The features detected represent a reasonable sampling of the interesting shape regions occurring in the meshes. We show that with curvature maps we can extract meaningful features in a robust way.

Section 2 focuses on related work in object recognition, feature detection, and segmentation. In Section 3 we discuss curvature maps, and the local shape property to which the min-cut/max-flow graph cut algorithm is applied. Section 4 gives the details of the feature detection algorithm, the graph cut parameters, and the multi-scale approach. Results for various shapes, and conclusions and possible areas for future work, are presented in Sections 5 and 6 respectively.

2 Related Work

There are three main areas of research related to this work, (1) shape representations or signatures, (2) feature segmentation, and (3) multi-scale approaches to thresholding. Object recognition, correspondence, and registration often rely on similarity measures to quantify the similarity or dissimilarity between objects by computing distances between shape representations, such as sets of points, feature vectors, histograms, signatures, or graph representations. A number of these shape representations have grown out of image analysis for range data [1, 2] or medical images [3].

Graph representations, such as skeletons [4, 5] and multi-resolution Reeb graphs [6], like algorithms based on point sets [7–12], can be useful for computing similarity and registration. But these methods are primarily global rather than local and do not identify local features of interest. Often, they are also sensitive to the distribution of the mesh points.

Several signatures have been defined for shape matching. Signatures may be global or local, and provide a compact representation that results in more efficient comparison at the expense of their ability to discriminate shape. Methods used for shape retrieval, such as shape distributions [13], spin images [14], and spherical spin images [15], tend to be global measures, and generally provide limited discrimination between similar shapes.

Signatures of a more local nature include statistical signatures [16] and shape contexts [17, 18], but the use of local point-to-point distances and angles, and sampling of points respectively, limits the suitability of these methods for detailed shape comparison. The point fingerprint [19] is a signature made by projecting concentric geodesic circles to a tangent plane, and selecting one of the resulting contours based on an irregularity measure. Shape similarity is computed by comparing corresponding normals and contour radii along each contour. A surface curvature signature [20] is defined that relies on high curvature feature points. Unlike these approaches, we are looking for subtle shape differences that require more than signatures just at ‘interesting’ points.

For this work, we use the Curvature Map [21] as our shape representation. By incorporating information over a larger region of the surface, the curvature map is more stable, and better represents the local shape around a given point on the object. It is also easily adapted for a multi-scale approach, as will be described in Section 4.

Shape decomposition methods for 3-D volumes have been developed based on semantics [22], topology [23, 24], and morphological tools [25]. However, volume decomposition provides volumetric features rather than surface features, and also is only applicable to closed objects.

Segmentation identifies local regions of an object. A number of segmentation methods use curvature, particularly the sign of the curvature [26, 27], isosurfaces and extreme curvatures [28], or watersheds of a curvature function [29, 30]. Other researchers have developed curvature-based methods to identify salient features using local surface descriptors [31], which build high level features from non-trivial local shapes, and center-surround filters with Gaussian weighted curvatures [32]. However, these segmentation methods do not yield the types of features we are interested in for shape matching.

Feature regions can also be extracted based on critical points (peaks, pits, and passes) and associated ridge and valley lines. In [33], smoothing [34] was required as a preprocessing step. Peak (pit) areas surrounded by valley (ridge) cycles then provide the candidate feature areas which were selected interactively for their work. The uncertainty as to an appropriate amount of smoothing and the narrow definition of a feature are drawbacks to this approach.

Graph cut algorithms have been used to segment images [35] and medical datasets [36, 37]. They are effective at assigning the vertices of a graph to either a feature (foreground) or background set, based on graph properties such as the gradient of the image intensity. Some of these methods employ an interactive step, where the user identifies feature and background seed points, to guide the algorithm to the objects that are to be separated. We treat our mesh as a graph and apply the graph cut algorithm described in Section 4.1, and identify features based on the resulting segmentation.

Several techniques have been used to either avoid the need to select parameter values and thresholds, or to automate the selection process. One approach is to use the uniqueness of features [38] to determine matching thresholds, but this method has limited applicability in cases where uniqueness cannot be assumed, for example, where several features may be very similar or even identical. One can also check the quality of the match [39] to pick the best threshold. A similar multi-scale approach [40] has been applied to medical image analysis, with classification based on the sign of mean and Gaussian curvature. In addition, multi-step algorithms have been used successfully with curvature based registration, where an initial coarse computation is followed by a second refinement step. Our method combines such a multi-scale approach with a two-step process in order to eliminate the need for user selection or tuning of parameters.

3 Local Shape Property

The basic feature shapes we are looking for include the peak, pit, ridge, and valley. The common link between these features is the dependence on the magnitude of the mean curvature. In order to identify these features, we need some measure of the likelihood that a vertex should be classified as belonging to one of these features. This measure needs to incorporate information about the neighborhood around the vertex, as well as at the vertex itself. The curvature map [21] provides this context.

In order to calculate the curvature map, we first need to calculate the curvature at each of the mesh vertices. Because we can expect object scans to contain some amount of noise, we utilize a quadric fit of a two-ring neighborhood, based on a natural parameterization [41]. For a vertex p , the 1-D curvature map, $Kmap(p)$, is de-

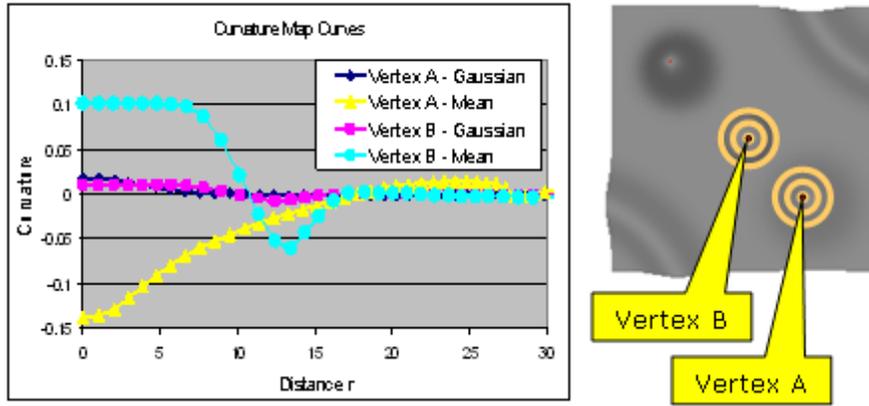


Fig. 2. The 1-D Curvature Map is defined by a mean and a Gaussian curvature curve. These curves are shown here for two sample vertices, A and B . The distance can be considered as expanding concentric rings as shown on the right, with the curvature value found by averaging the curvature values in the associated ring.

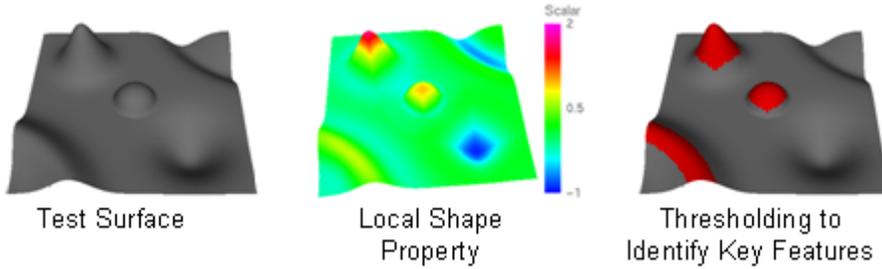


Fig. 3. Feature Detection Test Surface. Left: Surface shape with peaks, pit, ridge, and valley. Center: Mean curvature scalar function. Right: Features highlighted by selecting a function threshold. With the proper threshold, this function can highlight useful features, however, the threshold must be found by experimentation.

defined by two curves representing the average mean and Gaussian curvature as functions of distance from the vertex. We will refer to these curves as $Mean(Kmap(p))$ and $Gauss(Kmap(p))$ respectively. The curvature maps for two sample vertices are shown in Figure 2. We define our local shape property S as

$$S(p) = \int_0^R Mean(Kmap(p))(r) dr$$

where R represents the radius corresponding to the maximum feature size. A test surface, the local shape property, and features resulting from applying a threshold to S are shown in Figure 3.

We also considered functions based on the Gaussian curvature component of the curvature map and combinations of mean and Gaussian curvature, but given a suitable threshold, the mean curvature function gave the most consistent identification of the features in the test case. This is due to the primary relationship of these features to the mean curvature. We also experimented with the range over which the curves were integrated by finding the sign changes in the function value, but these variations did not improve the ability to detect features.

Although this local shape property often highlights the expected features, finding an appropriate threshold requires manual adjustment, and the results still depend on the curvature map radius R . In addition, no single threshold could extract both the positive curvature features (peak and ridge) and the negative curvature features (pit and valley). These factors motivated our search for an improved feature detection approach.

4 Multi-Scale Feature Detection

We combine our local shape property with the min-cut/max-flow graph cutting technique of Boykov and Kolmogorov [35], to create a multi-scale approach for feature detection. The min-cut/max-flow algorithm operates on a graph with weighted edges, along with weights for edges that are added to connect the graph vertices to a feature node and a background node. The algorithm then finds the minimum cost set of edges to delete, such that there remains no path from the feature node to the background node.

The primary benefit of the graph cut algorithm is its efficiency, and the compact boundary produced. We note that when we run the graph cut algorithm with a range of radii for our local shape property, different features may be identified. So our goal is to run the graph cut multiple times, and extract the most significant features overall.

The graph cut algorithm is applied first to the absolute value of our shape property. For the default graph cut weights, only the most prominent features are detected. To detect less prominent features, we multiply the weights by a scale factor, which takes on a range of values. Since the larger of the positive or negative shape property magnitudes may dominate the absolute value graph cuts, we also perform graph cuts separately on the positive and negative values of the local shape property. So we apply the graph cut technique three times at each scale in order to make sure we capture key positive and negative curvature features. This results in three categories of graph cuts: absolute value, positive, and negative.

The variations of curvature map radii and scale factors for the three graph cut categories generate a large number of possible feature sets. In order to simplify the process of extracting a master feature set from this data, we first count the number of times each vertex is identified as part of a feature in each of these categories. Then we run the graph cut algorithm on the normalized frequency counts, again varying the scale factor. This yields a smaller set of features that are then merged to create the master feature set. This process is shown in Algorithm 1.

Algorithm 1 Multi-Scale Feature Detection

```
Read Curvature Map ( $K_{map}$ ) for Mesh  $M$ 
for  $K_{map}$  radius  $R$  from  $R_{min}$  to  $R_{max}$  do
  Compute  $S$  as the integral of the  $K_{map}$  mean curvature component from 0 to  $R$ 
  for a range of weight factor  $\alpha$  do
    Create graph cuts  $C_{abs}, C_{pos}, C_{neg}$  on the absolute, positive, and negative values of  $S$ 
    Identify the features in  $C_{abs}, C_{pos}, C_{neg}$ 
    for each vertex  $v$  in Mesh  $M$  do
      Count feature occurrences  $N_{abs}, N_{pos}, N_{neg}$  in  $C_{abs}, C_{pos}, C_{neg}$ 
    end for
    for each edge do
      Count how many times both endpoints occur in the same region
      Note: Used to generate edge weights for the later max-flow/min-cut runs
    end for
  end for
end for
for a range of weight factor  $\alpha$  do
  Create graph cuts  $C_{abs}, C_{pos}, C_{neg}$  from normalized counts  $N_{abs}, N_{pos}, N_{neg}$ 
  Identify and merge features from  $C_{abs}, C_{pos}, C_{neg}$  into composite feature sets
   $G_{abs}, G_{pos}, G_{neg}$ 
end for
Merge  $G_{neg}$  and  $G_{pos}$  into  $G_{abs}$  to create the Master Feature Set  $G$ 
```

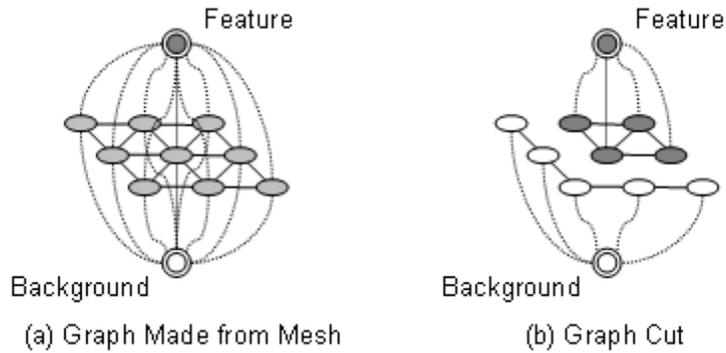


Fig. 4. The min-cut/max-flow graph cutting algorithm finds an optimal separation of the vertices of a mesh into a feature group and a background group. The cut is based on weights assigned to the mesh edges (solid lines) and to edges connecting the graph vertices to the feature and background nodes (dotted lines).

4.1 Graph Cut Parameters

In order to run the graph cut algorithm, we need to assign weights to the mesh edges and to connections from the mesh vertices to a ‘feature’ node and a ‘background’ node, as shown in Figure 4. These weights, given in Table 1, represent the cost of breaking the

Table 1. Graph Cut Weights

	<i>Shape Property Based</i>		<i>Feature Frequency Based</i>	
<i>Edge</i>	<i>Weight(cost)</i>	<i>for</i>	<i>Weight(cost)</i>	<i>for</i>
$\{p, q\}$	$E_1\{p, q\}$	$\{p, q\} \in Edges$	$E_2\{p, q\}$	$\{p, q\} \in Edges$
$\{p, F\}$	$-\log(1 - S(p))\sqrt{\alpha}$	$\forall p$	$-\log(1 - N(p))\sqrt{\alpha}$	$\forall p$
$\{p, B\}$	$-\log(S(p))/\sqrt{\alpha}$	$\forall p$	$-\log(N(p))/\sqrt{\alpha}$	$\forall p$

$$E_1 = \exp\left(-\frac{(S(p)-S(q))^2}{2(dist(p,q)\sigma)^2}\right) \frac{1}{dist(p,q)} \text{ if } S(p)S(q) < 0, \frac{1}{dist(p,q)} \text{ otherwise}$$

$$E_2 = \exp\left(-\frac{N_T - N_S}{N_T}\right) \text{ where } N_T \text{ is the number of cuts, and}$$

N_S is the number for which $Feature_p = Feature_q$

F and B are the feature and background nodes respectively, p, q are mesh vertices

α is the scale factor for the feature node weights

$S(p)$ is the local shape property value at p , limited to $\epsilon \leq S(p) \leq 1 - \epsilon$

$N(p)$ is the normalized frequency count at p

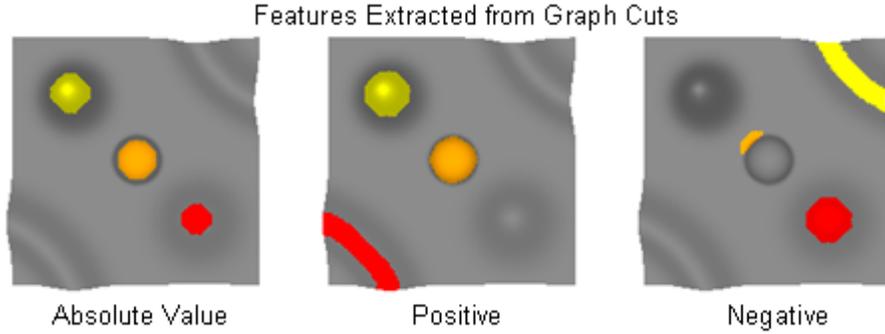


Fig. 5. Graph cuts generated by the min-cut/max-flow algorithm on the local shape property for three graph cut categories: absolute value (left), positive (center), and negative (right). The absolute value graph cut picks up the peak and pit features, while the valley is only found in the negative graph cut and the ridge is only found in the positive graph cut.

edge in order to separate the graph vertices into the feature and background sets. Note that the vertices within a set need not form a single contiguous region of the graph.

Once we have created the graph cut, we search for contiguous groups of vertices in the feature set of the graph cut. These contiguous groups of vertices are our features. Figure 5 shows features extracted from selected graph cuts of a test surface.

4.2 Multi-Scale Parameters

The two parameters that are varied are the curvature map radius R and the scale factor α . R is varied from small to large, with the size of the largest region based on the radius R_{max} used for the original curvature map calculation. R_{max} is assumed to be large enough to capture the largest desired feature. For example, on our human face scans, we use a maximum radius of about two inches. Smaller radii are defined by successively

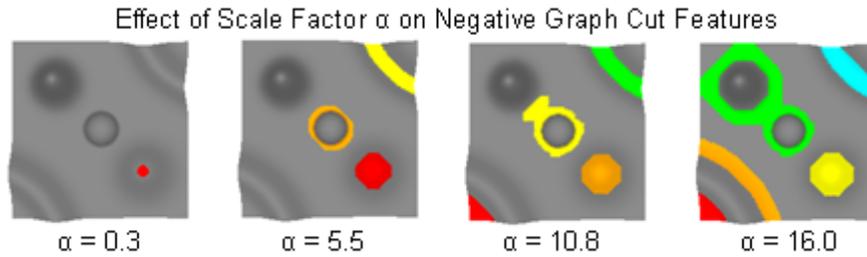


Fig. 6. Effect of the scale factor α on features identified using the Min-Cut/Max-Flow graph cutting algorithm. Representative cuts from the negative of the local shape property are shown. As α increases, more features are detected, and existing features become larger. At larger α the saddle region at the base of the peaks is detected.

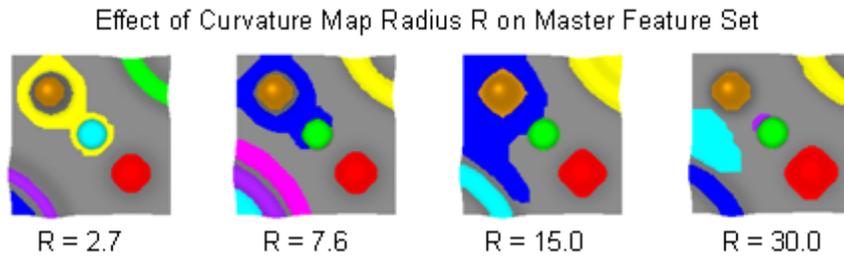


Fig. 7. Effect of the curvature map radius R on features identified using the Min-Cut/Max-Flow graph cutting algorithm. As R changes, different features are detected, and some features merge.

scaling by $1/\sqrt{2}$. For our cases, using eight levels was sufficient to make the minimum R comparable to the shortest edge of the mesh.

The weights for the connections to the feature node are scaled by $\sqrt{\alpha}$, while the connections to the background node are scaled by $1/\sqrt{\alpha}$. We determine α by trial and error. We first decrease α until we get only one group. Then we increase α until the number of groups reaches a peak. We then take uniformly spaced values for α in this range. For our examples, we use ten divisions. For these examples, 8 K_{map} radii cross the 10 scale factors results in 80 graph cuts for each category in the first step, with an additional 30 graph cuts in the second step, for a total of 270 graph cuts. Fortunately, the graph cut algorithm is very efficient, with the 270 graph cuts on a 10,000 vertex mesh taking less than 40 seconds on a 2.8GHz Pentium 4 processor.

Figure 6 shows the groups produced for selected scale factors for the negative graph cuts of our local shape property with a curvature map radius of 3.8. As the scale factor is increased, individual features tend to get larger, and new features may show up. Changes to the feature set as the curvature map radius R is increased are shown in Figure 7. The feature frequency counts are shown in Figure 8.

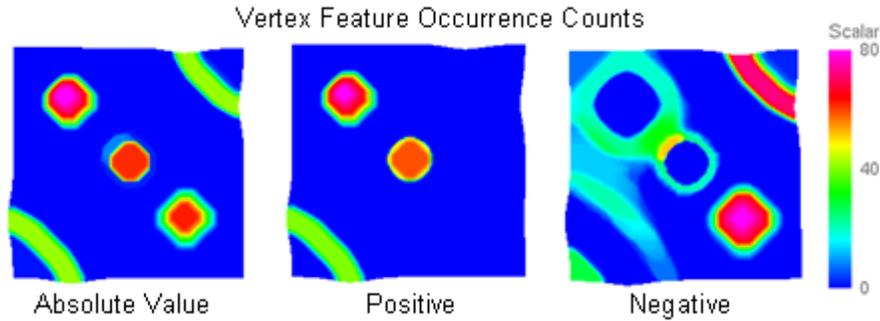


Fig. 8. Feature counts for the absolute value (left), positive (center), and negative (right) graph cut categories. Maintaining separate frequency counts for the three graph cut categories allows extraction of more well-defined features.

4.3 Group Merging Criteria

Algorithm 2 Merging Feature Set A into Feature Set B

Require: Feature Sets A and B on Mesh M

```

for each feature  $F_i$  in Set  $A$  do
    Determine how many features  $n$  in Set  $B$  overlap  $F_i$ 
    if  $n = 0$  then
        Add  $F_i$  as an additional feature in Set  $B$ 
    else if  $n = 1$  then
        Take the union of  $F_i$  with its overlapping feature in Set  $B$ 
    else
        Ignore the feature  $F_i$ 
    end if
end for

```

In order to merge feature groups together, we use a simple greedy approach as shown in Algorithm 2. When combining cuts from progressively larger source weights to form composite feature sets, the groups tend to grow, but without allowing neighboring groups to merge. This makes sure all of the features do not get merged together, as might occur for a very large scale factor. The same algorithm is applied when merging the composite feature sets for the absolute value, negative, and positive portions of the function. The composite feature sets for each of the function variations, and the final feature set made by combining them, are shown in Figure 9.

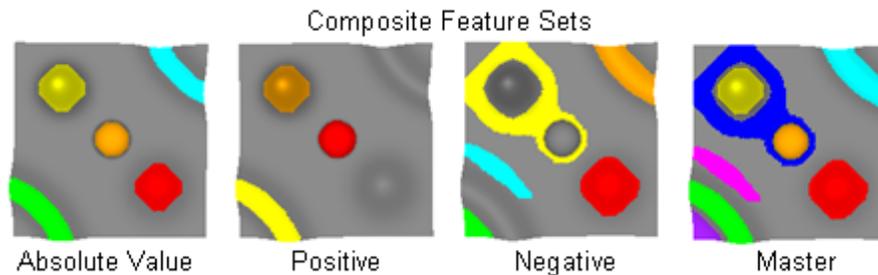


Fig. 9. Composite feature sets for the absolute value, positive, and negative graph cuts, and the master feature set made by merging them.

5 Results

Figure 10 shows the test surface with Gaussian noise added. In spite of the noise, we get a very similar feature structure to that of the case without noise shown in Figure 9.

Figure 11 shows features for several bone meshes with fairly subtle features. Note the similarity of the feature layout for Ulna A (View 2) and Ulna B in spite of a significant difference in mesh resolution and being from different subjects.

Figure 12 shows feature detection applied to a low resolution scan of a human face. The coarseness of the mesh has a smoothing effect that eliminates many details. It also highlights the benefit of running the absolute value, positive, and negative graph cuts to identify features for the master set that would be missed otherwise. In Figure 13 we compare our feature detection method with segmentation based on the signs of the mean and Gaussian curvature for a higher resolution human face. Even after smoothing the curvature data, the segmentation on the left shows quite a bit of noise. This is improved by setting a zero threshold so that large regions of low curvature are separated from the higher curvature features, as shown in the center segmentation. However, the resulting features depend strongly on the amount of smoothing applied and the zero threshold, and are still less well-defined than the master feature set shown on the right.

Features for the Stanford bunny are presented in Figure 14. While this case produced a number of very small features, the larger feature regions, such as in the ears, face, feet, and tail, seem to be features that could be useful for shape matching. Because the features can be ordered by strength, weaker features will only be used for tasks such as shape matching if there are not enough strong features detected.

6 Conclusions and Future Work

We have presented a two-step multi-scale feature detection approach that uses a local shape function based on the *Curvature Map*. It employs an efficient min-cut/max-flow graph cutting algorithm and greedy algorithm to merge feature sets. The method is robust with respect to noise, and consistently yields a reasonable set of features. Most importantly, there is no user interaction or parameter tuning required.

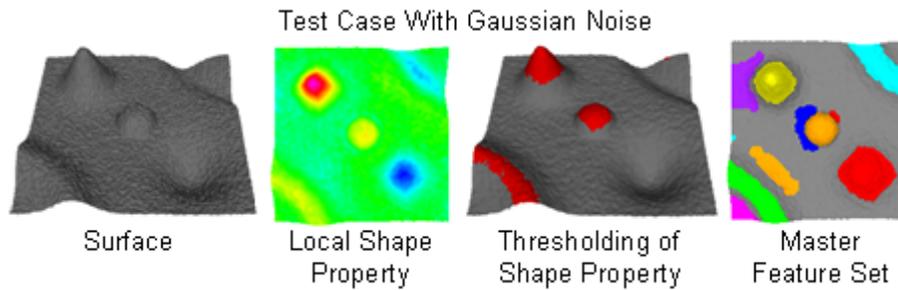


Fig. 10. Test case with Gaussian noise added. The function and final feature set are similar to the test case without noise, especially for the strongest features.

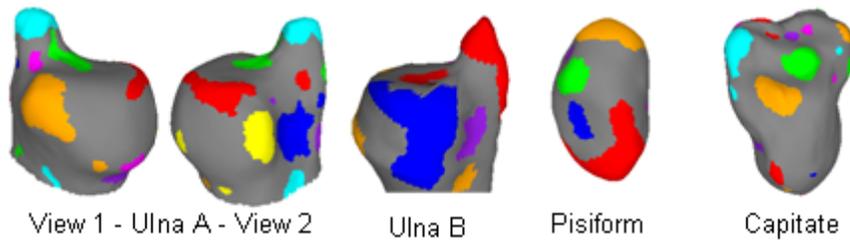


Fig. 11. Master Feature Sets for selected bone meshes. The Ulna is challenging due to the limited number of pronounced features and the significant difference between the scales of the features. Similar features were detected for Cases A and B even though the resolution of the meshes is very different. Reasonable features were also identified for the Pisiform (second from right) and Capitate (far right).

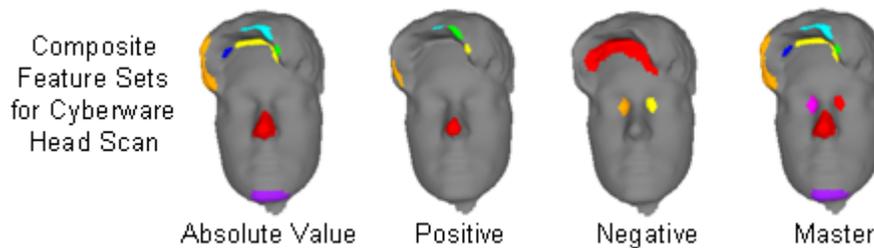


Fig. 12. Features detected on a Cyberware low resolution female face scan. The absolute value graph cuts pick up the nose chin and hair features, while the negative cuts detect the eyes. In spite of the smoothness of the mesh, the master feature set captures the prominent features of the face.

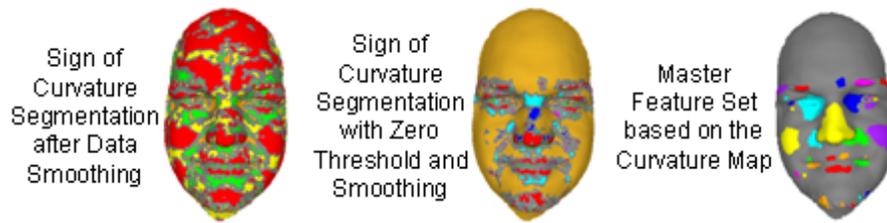


Fig. 13. Comparison of graph cut feature detection with sign of curvature segmentation for a high resolution Cyberware face scan. Before coloring by the sign of Gaussian and mean curvature, the curvature values were smoothed. The segmentation in the center uses a zero threshold to separate low curvature regions from higher curvature features. However, the master feature set provides more well-defined features.

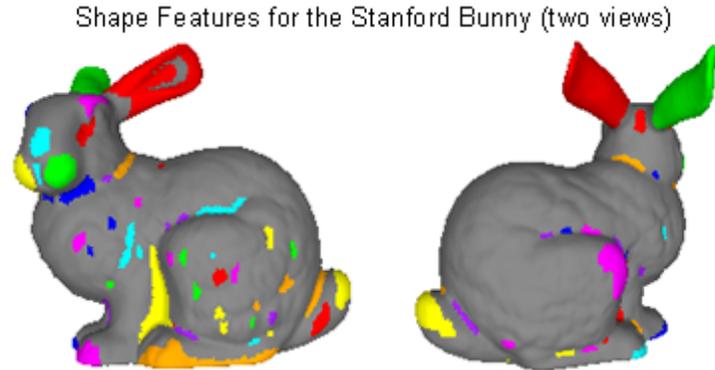


Fig. 14. Features detected for the Stanford bunny. Several features, such as the large sections of the ears and the features in the face region, are very intuitive.

The method could benefit from alternate algorithms for merging feature sets. The greedy approach works fairly well, but may cause some over-segmentation, since it does not allow two features to coalesce into one, which might be desirable in some instances.

Because the local shape property is based on the integral of mean curvature, it detects primarily higher curvature features. We will look at adding the capability to detect flat or nearly flat regions, although these are less useful for identifying shape similarity.

Acknowledgments This work was partially supported by NSF Grant 049856. The bone data was provided through NIH Grant AR44005, PI: J.J. Crisco, Brown Medical School /Rhode Island Hospital. The authors would also like to thank Vladimir Kolmogorov for the min-cut/max-flow code, Cyberware for the human head scans, and the Stanford Scanning Repository for the bunny data set.

References

1. Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J.: Recognizing objects in range data using regional point descriptors. In: ECCV (3). (2004) 224–237
2. Hetzel, G., Leibe, B., Levi, P., Schiele, B.: 3d object recognition from range images using local feature histograms. In: CVPR (2). (2001) 394–399
3. van den Elsen, P., Maintz, J., Pol, E., Viergever, M.: Medical image matching - a review with classification. In: IEEE Engineering in Medicine and Biology. (1993) 26–39
4. Bloomenthal, J., Lim, C.: Skeletal methods of shape manipulation. In: International Conference on Shape Modeling and Applications. (1999) 44
5. Klein, P.N., Sebastian, T.B., Kimia, B.B.: Shape matching using edit-distance: an implementation. In: Symposium on Discrete Algorithms. (2001) 781–790
6. Hilaga, M., Schinagawa, Y., Kohmura, T., Kuni, T.L.: Topology matching for fully automatic similarity estimation of 3d shapes. In: Computer Graphics (Proc. SIGGRAPH 2001). (2001) 203–212
7. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence **14**(2) (1992) 239–256
8. Rusinkiewicz, S., Levoy, M.: Efficient variants of the icp algorithm. In: Third International Conference on 3D Digital Imaging and Modeling (3DIM). (2001) 145–152
9. Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. In: Computer Vision and Image Understanding. Volume 89. (2003) 114–141
10. Angelov, D., Koller, D., Srinivasan, P., Thrun, S., Pang, H.C., Davis, J.: The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In: Advances in Neural Information Processing Systems (NIPS 2004), Vancouver, Canada (2005)
11. Brown, B., Rusinkiewicz, S.: Non-rigid range-scan alignment using thin-plate splines. In: Symposium on 3D Data Processing, Visualization, and Transmission. (2004) 759–765
12. Lu, X., Jain, A.K.: Deformation analysis for 3d face matching. In: WACV/MOTION. (2005) 99–104
13. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Shape distributions. In: ACM Transactions on Graphics, **21**(4). (2002) 807–832
14. Johnson, A., Hebert, M.: Recognizing objects by matching oriented points. In: CVPR '97. (1997) 684–689
15. Ruiz-Correa, S., Shapiro, L.G., Meila, M.: A new signature-based method for efficient 3-d object recognition. In: CVPR (1). (2001) 769–776
16. Planitz, B.M., Maeder, A.J., Williams, J.A.: Intrinsic correspondence using statistical signature-based matching for 3d surfaces. In: Australian Pattern Recognition Society (APRS) Workshop on Digital Image Computing (WDIC). (2003)
17. G., M., Belongie, S., Malik, H.: Shape contexts enable efficient retrieval of similar shapes. In: CVPR 1. (2001) 723–730
18. Körtgen, M., Park, G.J., Novotni, M., Klein, R.: 3d shape matching with 3d shape contexts. In: The 7th Central European Seminar on Computer Graphics. (2003)
19. Sun, Y., Paik, J.K., Koschan, A., Page, D.L., Abidi, M.A.: Point fingerprint: A new 3-d object representation scheme. IEEE Transactions on Systems, Man, and Cybernetics, Part B **33**(4) (2003) 712–717
20. Yamany, S.M., Farag, A.A.: Surface signatures: An orientation independent free-form surface representation scheme for the purpose of objects registration and matching. IEEE Trans. Pattern Anal. Mach. Intell. **24**(8) (2002) 1105–1120
21. Gatzke, T., Zelinka, S., Grimm, C., Garland, M.: Curvature maps for local shape comparison. In: Shape Modeling International. (2005) 244–256

22. Kim, D.H.: A Semantic Decomposition Scheme for Both 2-D and 3-D Shapes. PhD thesis, Seoul National Univ. (0000)
23. Mortara, M., Patanè, G., Spagnuolo, M., Falcidieno, B., Rossignac, J.: Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. *Algorithmica* **38**(1) (2003) 227–248
24. Dey, T., Giesen, J., Goswami, S.: Shape segmentation and matching with flow discretization. In Dehne, F.K.H.A., Sack, J.R., Smid, M.H.M., eds.: *Proceedings of the Workshop on Algorithms and Data Structures (WADS)*. Lecture Notes in Computer Science 2748. (2003) 25–36
25. Maintz, J.B.A., van den Elsen, P.A., Viergever, M.A.: Registration of 3d medical images using simple morphological tools. In: *IPMI*. (1997) 204–217
26. McIvor, A.M., Penman, D.W., Waltenberg, P.T.: Simple surface segmentation. In: *ICTA/IVCNZ97*, Massey University, New Zealand (1997) 141–146
27. Wilson, R.C., Hancock, E.R.: Consistent topographic surface labelling. *Pattern Recognition* **32**(7) (1999) 1211–1223
28. Vivodtzev, F., Linsen, L., Bonneau, G.P., Hamann, B., Joy, K.I., Olshausen, B.A.: Hierarchical isosurface segmentation based on discrete curvature. In: *Proceedings of VisSym'03-Data Visualization 2003*, New York, New York, ACM Press (2003) 249–258
29. Mangan, A.P., Whitaker, R.T.: Partitioning 3d surface meshes using watershed segmentation. In: *IEEE Transactions on Visualization and Computer Graphics* **5**(4). (1999) 308–321
30. Page, D.L., Koschan, A.F., Abidi, M.A.: Perception-based 3d triangle mesh segmentation using fast marching watersheds. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)* **2** (2003) 27–32
31. Gal, R., Cohen-Or, D.: Salient geometric features for partial shape matching and similarity. *J-TOG* **25**(1) (2006) 130–150
32. Lee, C.H., Varshney, A., Jacobs, D.W.: Mesh saliency. *Trans. Graph.* **24**(3) (2005) 659–666
33. Takahashi, S., Ohta, N., Nakamura, H., Takeshima, Y., Fujishiro, I.: Modeling surperspective projection of landscapes for geographical guide-map generation. *Computer Graphics Forum* **21**(3) (2002) 259–268
34. Taubin, G., Zhang, T., Golub, G.: Optimal surface smoothing as filter design. In: *Tech Report RC-20404*. (0000)
35. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(9) (2004) 1124–1137
36. Boykov, Y., Jolly, M.P.: Interactive organ segmentation using graph cuts. In: *MICCAI '00: Proceedings of the Third International Conference on Medical Image Computing and Computer-Assisted Intervention*, London, UK, Springer-Verlag (2000) 276–286
37. Kim, J., Zabih, R.: A segmentation algorithm for contrast-enhanced images. In: *Ninth IEEE International Conference on Computer Vision (ICCV'03)*. Volume 1. (2003) 502–509
38. Biber, P., Straßer, W.: Solving the correspondence problem by finding unique features. In: *16th International Conference on Vision Interface*. (2003)
39. Mus, P., Sur, F., Cao, F., Gousseau, Y.: Unsupervised thresholds for shape matching. In: *IEEE Intl. Conf. on Image Processing, ICIP, 2003*. (2003) 647–650
40. Avants, B.B., Gee, J.C.: The shape operator for differential analysis of images. In: *IPMI*. (2003) 101–113
41. Gatzke, T., Grimm, C.: Estimating curvature on triangular meshes. *Int. Journal of Shape Modeling to appear* (2006)