

Distance based feature detection on 3D point sets

Ahmad Ramli and Ioannis Ivriissimtzis

Durham University, UK

Abstract

We propose a distance based algorithm for implicit feature detection on 3D point sets. Instead of directly determining whether a point belongs to a feature of the 3D point set or not, we first compute the distance between the point and its nearest feature. The obtained distance function is filtered, removing noise and outliers, and the features of the point set are computed as the zero set of the filtered function. Initial tests show that the proposed method is robust and can deal with amount of noise usually expected in a point set.

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

1. Introduction

Point sets are becoming an alternative to polygonal meshes and NURBS for surface representation in 3D modelling applications, even though the most efficient geometry processing and rendering algorithms still require polygonal meshes as input. The main argument in favour of using point sets is that the polygonal meshes found in several applications are polygonisations of point sets. The polygonisation of a point set imposes a sometimes arbitrary connectivity, creating problems which can be avoided if one works directly with the raw geometric data.

From this point of view, feature detection has some advantages when performed on the 3D point set, before the meshing of the data has imposed a connectivity. For example, it is known that *Marching Cubes* [LC87], the most commonly used polygonisation algorithm does not respect sharp edges [Wan06]. One remedy for this is to collect feature information on the input point set, and pass it to the polygonisation algorithm for feature reconstruction.

In this paper we propose an implicit feature detection algorithm for point sets. At the first stage of the algorithm, the goal is to compute a function giving the distance between a point and its nearest feature. For each point p , we do *Principal Component Analysis (PCA)* [Joi02] on its k -neighborhood, for an increasing sequence of k 's. We process the eigenvalues obtained from the PCA, trying to detect when the k -neighborhood has reached a feature. Then, we

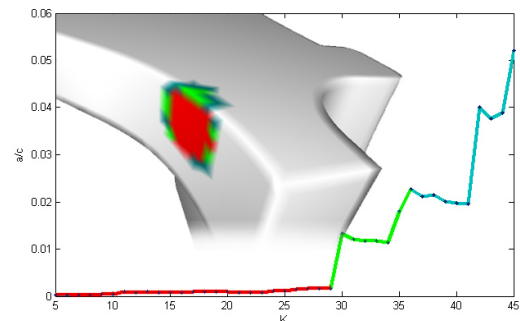


Figure 1: An expanding k -neighborhood of a point reaches a feature of the model. The graph shows the ratio of the smallest and largest eigenvalues of the PCA.

use the radius of the k -neighborhood as a measure of the distance between p and its nearest feature, see Fig. 1. At the second stage of the algorithm, we smooth the distance function using bilateral filtering. Finally, the feature points of the set are detected as the points with almost zero distance from their nearest feature.

Preliminary results show that the proposed algorithm is robust, as spurious results due to data noise can be handled by processing the values of the distance function. We believe that the algorithm exhibits the robustness to

noise usually associated with implicit surface reconstruction [HDD*92,CBC*01]. Indeed, distance functions are theoretically smooth and thus, smoothing can justifiably be used, either as post-processing or implicitly during the computations, to remove noise and outliers. One limitation of the method is its high computational cost. A second limitation is that excessive noise can distort the computations of the distance function to the extent that smoothing will only be able to restore the most basic features of the original data.

2. Related work

Given its importance in geometric modelling applications [MI97], feature detection and extraction, both for 3D point sets and polygonal meshes, has received considerable research interest.

Regarding feature detection on point sets, [YLHP06] uses PCA to compute principle curvatures on neighborhoods of varying size. The curvature information can then be used for feature detection. Similarly, [GG06] uses quadratic fitting to compute a curvature map which is used to detect and classify features. [BO08] focuses on the application of point set registration. As the main descriptor of a feature it uses the distance between a point and its neighbourhood's centre of mass. [LG05] smooths the point set with the MLS projection and then defines as features the areas where the difference between normals of neighbouring points is maximum. [GWM01] analyses the eigenvalues and eigenvectors of a neighbourhood and computes the probability that a point is a feature of a certain type, such as sharp edge, boundary or corner. [DVVR07] uses normal computations to obtain a segmentation of the point set and then detects sharp features between the segments. [DHOS07] identifies as features the points with the highest error in a RMLS polynomial fitting. [Chi08] extracts features from a voxelised model of the point data which is segmented using visibility computations.

Considerable research effort has also been directed into the computation of features on polygonal meshes. The techniques applied there are similar to those for point sets. [YBS05] uses polynomial fittings to compute curvature information which is then used for detecting feature lines on triangle meshes. [SPK*02] uses eigenanalysis on an average tensor, weighted by the geodesic distance of tensors of neighboring normals. [Wan06] analyses angles between normals to detect features and then uses bilateral filtering to reconstruct them. [OBS04] detects features by first constructing an implicit model of the data and then computing curvature information from that implicit model.

The feature detection method closer to our approach is the one proposed in [PKG03]. There, similarly to our approach, neighborhoods of varying size are analysed with PCA. However, there the extracted information is used to create a probability map, while our algorithm creates a distance map. The difference is not just in the interpretation of

the PCA results; our map explicitly encodes the distance between a point and its nearest feature, while in [PKG03] such information is only implicitly encoded through the selection of a neighborhood of optimal size. We believe that using the PCA results to create a distance map is a more robust approach because a distance map is expected to be smooth, thus allowing post-processing operations. [ACSTD07] also uses neighborhoods of varying size to compute a probability map for the estimated normals, which is then used for surface reconstruction.

3. Feature detection on point clouds

The proposed feature detection algorithm is based on the PCA of neighborhoods of varying size. For every vertex p , we calculate the eigenvalues of the covariance matrix

$$b_{jk} = \sum_i (q_{ij} - p_j)(q_{ik} - p_k) \quad (1)$$

where $p = (p_x, p_y, p_z)$ and q_i is the i -th neighbouring vertex.

Let $a \leq b \leq c$ be the three eigenvalues of the matrix in Eq. 1. The eigenvector corresponding to a is the normal vector at p . The ratio a/c can be used as a measure of how flat is the neighbourhood. A small value of a/c means that a is small with respect to c , indicating a flat area. [PKG03] has a similar approach, considering the ratio $a/(a+b+c)$ instead.

We keep increasing the i -neighbourhood, creating a function $f_{a/c}(i)$ of values of a/c , until $f_{a/c}(i)$ exceeds a user-defined threshold $T_{a/c}$. Experimentally we found that a value $T_{a/c} = 0.1$ gives satisfactory results. Even though the thresholding of $f_{a/c}(i)$ is a crude instrument for feature detection itself, we found it sufficient for the purpose of computing a distance function. If $T_{a/c}$ is not exceeded after a certain number of iterations the process terminates anyway.

Notice that, as Fig. 1 indicates, if we take into consideration the derivative of $f_{a/c}(i)$ as well as its value, for example, by thresholding the function $f_{a/c}(i) + \alpha \cdot f'_{a/c}(i)$, where α is a constant, we can compute more accurate distance functions on smooth models. However, as Fig. 2 indicates, that approach would be less robust in the presence of noise.

When the neighborhood has been computed, we assign the radius of that neighborhood as the distance h_p between the point p and its nearest feature. This is the main difference between our method and previous approaches with varying neighbourhoods, where such information is discarded.

The next step is the bilateral filtering of the distance function h . Bilateral filtering was introduced in [TM98] for edge preserving image denoising. The new distance h'_j between the vertex j and its nearest feature is given by

$$h'_j = (1 - \alpha)h_j + \alpha \frac{\sum_{i=1}^n g_a(\|p_j - q_i\|) \cdot g_b(\|h_j - h_i\|) \cdot h_i}{\sum_{i=1}^n g_a(\|p_j - q_i\|) \cdot g_b(\|h_j - h_i\|)}$$

where α is a user-defined parameter and g_a, g_b are Gaussian

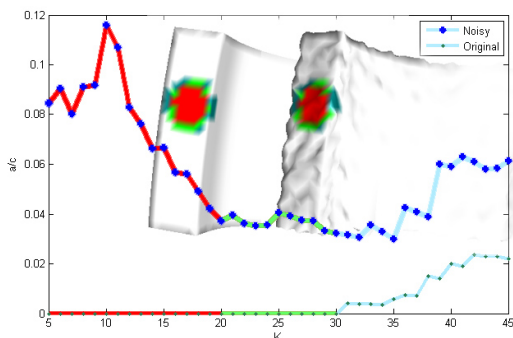


Figure 2: The computation of the function $f_{a|c}(i)$ on corresponding points of a smooth and a noisy model. The graphs of $f_{a|c}(i)$ show the problems associated with the direct application of PCA for feature detection.

functions. The standard deviation of g_a is the average, over the whole point set, of the distance between a point and its nearest neighbour, while the standard deviation of g_b was set to 0.01.

To increase the effectiveness of the method, before bilateral filtering we find the points with zero values, which are the already detected features and some noisy points, and replace their values with $-\max(\text{radius})$, that is, the negative value of the maximum neighbourhood radius. This way the features of the model are accentuated and better preserved by the filtering process. Notice that otherwise the discontinuity preserving property of the bilateral filtering would not have been fully utilised as the distance function is smooth.

After the smoothing process, the features of the point set are extracted by thresholding the distance function. The value of the threshold T_h is critical for the shape of the features. Having T_h as a user-defined parameter is typical in such applications.

4. Validation

We tested the proposed algorithm on the Fandisk and Bunny models, as well as on the Fandisk model with added noise.

Fig. 3 shows the results of the two smooth models. Notice a small part on the flat area of the Fandisk where it seems to be an error in the computation of the distance function. The reason is that there the model is thin, and the two surface sheets that are close to each other are identified as features. However, the problem is solved with a suitable thresholding of the distance function. The results of the Bunny are comparable with [PKG03].

Fig. 4 shows the results of the noisy Fandisk model. Notice the considerable improvement after the bilateral filtering. We believe that this is indicative of the robustness of the proposed method and its potential compared to direct PCA.

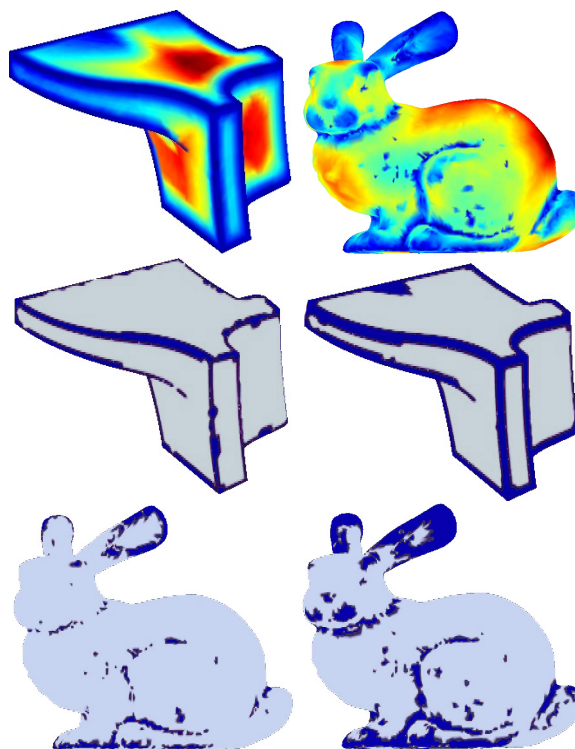


Figure 3: *Top:* Colourmaps of the distance function. *Middle:* Feature detection with the threshold value T_h set at 10% and 20% of the maximum value of h . *Bottom:* Feature detection with T_h set at 20% and 30% of the maximum of h .

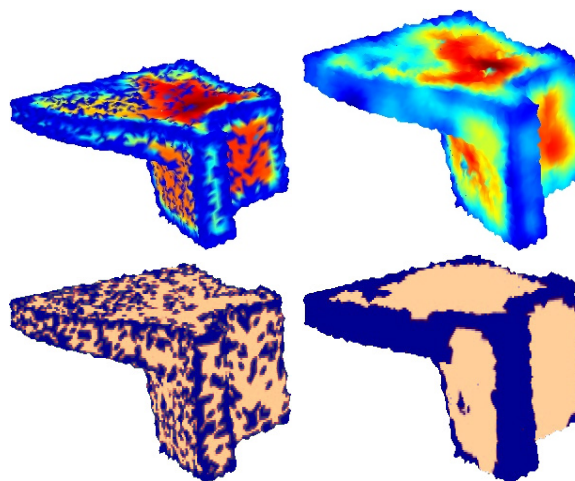


Figure 4: *Top:* Colourmap of the distance function before and after bilateral filtering. *Bottom:* Feature detection with $T_h = 0$, before and after bilateral filtering.

5. Discussion

We proposed a method for feature detection on 3D point sets based on estimating the distance between a point and its nearest feature. Initial tests show that the proposed method is robust enough to handle the amount of noise that is usually present in point set data.

In the future, we plan to further improve the algorithm by revisiting the computation of the distance function. In particular, instead of relying on the smoothing step to average out spurious values, we plan to use statistical tests to reject erroneous small values resulting from the high anisotropy of the small neighbourhoods of noisy point sets. We believe that such a refinement of the distance estimations will increase the accuracy of the method, while the robustness of the approach, which is based on the implicit nature of the algorithm, will not be affected.

In many practical applications the extracted features should be lines. For such applications we need a more sophisticated feature extraction algorithm than the current method of thresholding the values of the distance function. Candidate methods for such an improvement are line growing algorithms and snakes. Generally, as feature detection and extraction are interdependent problems, we believe that any future improvement on the computation of the distance function should be specific to the feature extraction method that will be used.

References

- [ACSTD07] ALLIEZ P., COHEN-STEINER D., TONG Y., DESBRUN M.: Voronoi-based variational reconstruction of unoriented point sets. In *Proc. Symposium on Geometry Processing* (2007), pp. 39–48.
- [BO08] BASDOGAN C., OZTIRELI A. C.: A new feature-based method for robust and efficient rigid-body registration of overlapping point clouds. *Vis. Comput.* 24, 7 (2008), 679–688.
- [CBC*01] CARR J., BEATSON R., CHERRIE J., MITCHELL T., FRIGHT W., MCCALLUM B., EVANS T.: Reconstruction and representation of 3d objects with radial basis functions. In *Proc. SIGGRAPH* (2001), pp. 67–76.
- [Chi08] CHICA A.: Visibility-based feature extraction from discrete models. In *Proc. Symposium on Solid and Physical Modeling* (2008), pp. 347–352.
- [DHOS07] DANIELS J. I., HA L. K., OCHOTTA T., SILVA C. T.: Robust smooth feature extraction from point clouds. In *Proc. Shape Modeling and Applications* (2007), pp. 123–136.
- [DVVR07] DEMARSIN K., VANDERSTRAETEN D., VOLODINE T., ROOSE D.: Detection of closed sharp edges in point clouds using normal estimation and graph theory. *Comput. Aided Des.* 39, 4 (2007), 276–283.
- [GG06] GATZKE T., GRIMM C.: Feature detection using curvature maps and the min-cut/max-flow algorithm. In *Proc. Geometric Modeling and Processing* (2006), pp. 578–584.
- [GWM01] GUMHOLD S., WANG X., MACLEOD R.: Feature extraction from point clouds. In *Proc. International Meshing Roundtable* (2001), pp. 293–305.
- [HDD*92] HOPPE H., DE ROSE T., DUCHAMP T., McDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. In *Proc. SIGGRAPH* (1992), pp. 71–78.
- [Jol02] JOLLIFFE I. T.: *Principal Component Analysis*. Springer, 2002.
- [LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3D surface construction algorithm. In *Proc. SIGGRAPH* (1987), pp. 163–168.
- [LG05] LI X., GUSKOV I.: Multi-scale features for approximate alignment of point-based surfaces. In *Proc. Symposium on Geometry Processing* (2005), p. 217.
- [MI97] MA K.-L., INTERRANTE V.: Extracting feature lines from 3d unstructured grids. In *Proc. IEEE Visualization* (1997), pp. 285–292.
- [OBS04] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph.* 23, 3 (2004), 609–612.
- [PKG03] PAULY M., KEISER R., GROSS M. H.: Multi-scale feature extraction on point-sampled surfaces. *Comput. Graph. Forum* 22, 3 (2003), 281–290.
- [SPK*02] SUN Y., PAIK J. K., KOSCHAN A., PAGE D. L., ABIDI M. A.: Triangle mesh-based edge detection and its application to surface segmentation and adaptive surface smoothing. In *Proc. ICIP* (3) (2002), pp. 825–828.
- [TM98] TOMASI C., MANDUCHI R.: Bilateral filtering for gray and color images. In *Proc. International Conference on Computer Vision* (1998), p. 839.
- [Wan06] WANG C. C. L.: Bilateral recovering of sharp edges on feature-insensitive sampled meshes. *IEEE Transactions on Visualization and Computer Graphics* 12, 4 (2006), 629–639.
- [YBS05] YOSHIZAWA S., BELYAEV A., SEIDEL H.-P.: Fast and robust detection of crest lines on meshes. In *Proc. Symposium on Solid and physical modeling* (2005), pp. 227–232.
- [YLHP06] YANG Y.-L., LAI Y.-K., HU S.-M., POTTMANN H.: Robust principal curvatures on multiple scales. In *Proc. Symposium on Geometry Processing* (2006), pp. 223–226.