Experiments on the Accuracy of Feature Extraction

Freek Reinders[1], Hans J.W. Spoelder[2], and Frits H. Post[1]

[1] Dept. of Computer Science, Delft University of Technology
PO Box 356, 2600 AJ Delft, The Netherlands
email: {k.f.j.reinders, f.h.post}@cs.tudelft.nl
[2] Dept. of Physics and Astronomy, Vrije Universiteit
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
email: hs@nat.vu.nl

Abstract. Feature extraction is an approach to visualization that extracts important regions or objects of interest algorithmically from large data sets. In our feature extraction process, high-level attributes are calculated for the features, thus resulting in averaged quantitative measures. The usability of these measures depends on their robustness with noise and their dependency on parameters like the density of the grid that is used. In this paper experiments are described to investigate the accuracy and robustness of the feature extraction method. Synthetic data is generated with predefined features, this data is used in the feature extraction procedure, and the obtained attributes of the feature are compared to the input attributes. This has been done for several grid resolutions, for different noise levels, and with different feature extraction parameters. We present the results of the experiments, and also derive a number of guidelines for setting the extraction parameters.

Keywords: feature extraction, attribute calculation, experimental accuracy estimation.

1 Introduction

Feature extraction is a set of techniques in scientific visualization aiming at algorithmic, automated extraction of relevant features from data sets. This leads to a small set of numbers (the attributes) describing the properties of the features. Hence, feature extraction lifts the data to a higher abstraction level, and comes down to a major data reduction. Since an "interesting feature" is different for each application, many application-specific feature extraction techniques exist, examples are critical points extraction [2], vortex extraction [1], and shock wave extraction [3]. A more general approach for extracting features is introduced by Post et al [4], and [8]. It is summarized by the pipeline model in figure 1, and consists of the following stages: selection, clustering, attribute calculation, and iconic mapping.

Selection identifies all grid nodes where the data satisfies a certain selection criterion, *clustering* clusters the selected nodes into regions of interest, *attribute*

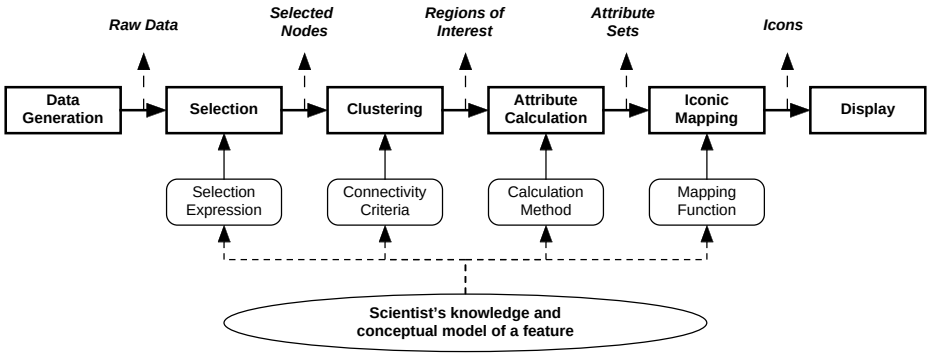

Fig. 1. The feature extraction pipeline.

calculation determines a number of attributes for each feature, and *iconic mapping* maps the calculated attributes to an icon which can be displayed. This process is controlled by the scientist in the sense that his knowledge of the data and his conceptual ideas of an interesting feature are translated into the selection expression, the connectivity criteria, the calculation method and the mapping function.

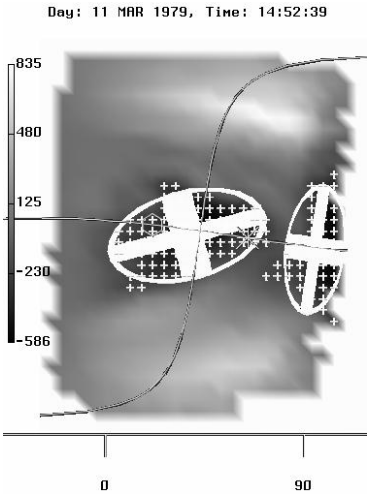

Fig. 2. Iconic presentation of cloud features on Venus.

An example of feature extraction is the detection of cloud formations in the atmosphere of Venus [6]. The clouds are visualized by ellipsoids which give a good indication of position and size (see Fig. 2). Motion of the clouds can be derived by visually matching the ellipsoids in consecutive frames. However, we believe the attributes of the ellipsoids can be used for automatic matching of features. Subsequently, it is important that the attributes are determined accurately, and that the results are robust with respect to noise. The latter will depend on feature extraction parameters like the selection threshold value, the cluster threshold, and the connected component definition.

Therefore, we wish to investigate the accuracy of the attribute calculation, and the influence of noise in combination with different extraction parameters on the calculated attributes. This is achieved by a simulation study: synthetic data is generated with synthetic features, i.e. with known attributes, and with noise with a known distribution function. This data is used as input for feature extraction, and the attributes obtained are compared to the initial settings of the attributes. This has been done for different noise levels, with different grid densities, and with different feature extraction param-

eters. In this way, we derived a number of guidelines for working with the feature extraction method in practice. Hence, it is not our intention to extract features below noise level!

The paper is organised as follows: section 2 gives a detailed description of the problem definition, section 3 discusses the generation of the synthetic data, section 4 describes the experiments performed, section 5 presents the results of the experiments, and section 6 draws some conclusions, and finally section 7 suggests work for future research.

2 Problem definition

The experiments focus on two main issues:

1. *Accuracy of the attribute calculation method.* Attribute calculation determines a number of quantitative characteristics of a feature. The attributes may be related to the data in the feature, to the geometry of the feature, or to a combination of both. In order to describe the geometry, we use ellipsoid-fitting because amorphous 3D objects can be approximated by ellipsoids [7]. The resulting attributes are the center position, the lengths and orientations of the ellipsoid axes. These can be estimated using an integration over the selected nodes: the average position of the nodes is the center position of the ellipsoid, and from the variance-covariance matrix of the node positions the axis-lenghts and orientations can be derived by solving the eigenvector/eigenvalue problem of the matrix.
   The accuracy of the ellipsoid attribute calculation depends on the integration procedure. The accuracy of the integration depends on the number of nodes within a feature; the average position and variation in position is more accurate when we integrate over a large number of nodes. Thus, the accuracy of the attributes will also depend on the (local) grid density.
2. *Robustness of the extraction method with noise.* The presence of noise in data will introduce false positives, and false negatives in the collection of selected nodes. Besides an error in the attributes, this will cause the emergence of spurious features. The latter can be eliminated by choosing the right extraction parameters. The extraction parameters consist of the selection threshold value, the cluster threshold, and the connected component definition.
   - The selection threshold value (or multiple values) decides whether the data in a grid point satisfies our selection criterion. It can be set above the noise level in order to eliminate noise effects, but this will also influence the resulting feature.
   - The cluster threshold is the minimum number of nodes of a cluster; all clusters smaller than the cluster threshold are discarded. Thus, only large features remain, and small features resulting from noise are removed. However, we may also remove small but genuine features.
   - The connected component definition can be defined as: 1D-connected (where a node has 6 neighbours), 2D-connected (18 neighbours), and 3D-connected (26 neighbours). This definition is crucial in the clustering

stage, since it determines if two adjacent nodes are in the same cluster or not. Obviously, 1D-connected will result in more and smaller clusters than 2D- or 3D-connected.

The extraction parameters must be chosen with care, therefore we will establish a number of guidelines for finding the right settings.

3 Synthetic data

In order to examine the relations between accuracy, noise, and extraction parameters, we created well-defined synthetic data on which we perform a number of experiments. The data is generated on a regular grid with a variable density. A scalar field is created on this grid with a variable initial value (set to zero by default), and noise is possibly added on top of this. The noise has a Gaussian distribution function with zero mean, and a variable standard deviation (SD); it is generated with an algorithm given by Press et al [5]. Furthermore, data values are added to grid nodes inside the synthetic features. The synthetic features are ellipsoids with given center position, axis lengths and orientations, plus a data value is defined at the center of the ellipsoid (set to 100.0 by default). The data within the ellipsoid decreases linearly from the center to the surface (value = 0.0). Thus, for each grid node inside the feature a data value is calculated and added to the present node value.

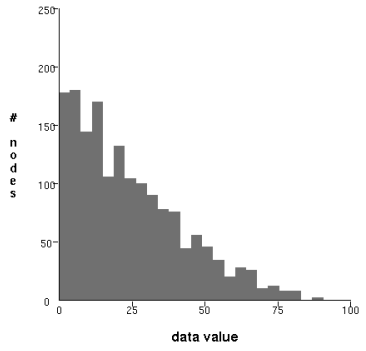

Fig. 3. Histograms of the generated synthetic ellipsoid data.

The synthetic data is used as the input in the feature extraction pipeline, where the data is thresholded, the selection clustered, and an ellipsoid-fit is performed around the clusters. Since the features in this data are predefined, the obtained attributes can be compared directly to the attributes specified as input, thus obtaining an experimental error estimation.

Figure 3 shows the histogram of the data within an ellipsoid feature (background with data = 0 is omitted from the histogram). Most of the nodes in the feature have a value close to zero and only few come close to the maximum value (=100). Additional noise will mostly affect the feature-nodes near the surface of the ellipsoid where the data values are small. Still it is possible to extract the feature since the maximum data value of the feature is significantly higher than the noise data. This is shown in figure 4, where a selection is made of nodes with a data value > 2*SD = 30.0, the figure shows the selected nodes by small crossmarks, and ellipsoids are fitted around each cluster with more than one node. One of the ellipsoids is significantly larger than the rest, this is the synthetic feature, it can be filtered out by choosing a larger cluster threshold, thus eliminating all small clusters.

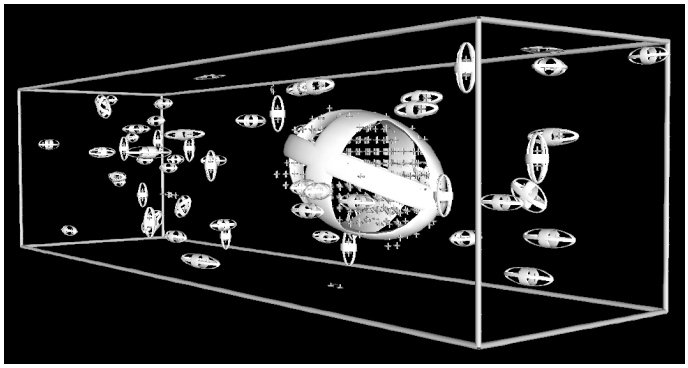

Fig. 4. Resulting selections from data with noise.

4 Experiments

4.1 Accuracy of the ellipsoid-fitting method

- **Center position.** The position detection is expected to have an error below cell-size level, which can be proved by the following experiment. Synthetic data is generated with a spherical feature with fixed radius and a position moving in 50 steps from a corner node of a cell diagonal through the cell to the center of the cell. Each of the 50 data sets are analysed by the feature extractor, and the resulting positions are used for error estimation. We expect the resulting position to move stepwise through the cell, the steps are caused by nodes entering or exiting the moving sphere. The distance to the diagonal (the real position) divided by the diagonal length, gives a relative error for the position detection. The same experiment can be repeated for different grid resolutions, i.e. a feature with a larger number of selected nodes. This will probably show that the accuracy is better for higher resolutions.
- **Axis length.** To determine the accuracy of the axis lengths, synthetic data is generated containing ellipsoids with fixed orientation and with the radius of one of the axes varying in one direction. Again, the variation is limited within a cell, and the experiment is repeated for several grid resolutions. Errors are calculated relatively to the cell size.
- **Axis orientation.** Synthetic data is generated containing ellipsoids with fixed axes ratios with an eccentricity of 3:1:1, and with varying orientation of the main axis (from 0 to 45 degrees), for several grid resolutions. Errors are calculated relatively to the maximum possible angle, i.e. 45 degrees.

4.2 Robustness of the method

As discussed in section 2, there are three important settings in the feature extraction procedure, the selection threshold, the cluster threshold and the connected component definition. The following experiments establish the relationships between these parameters, and the effects on the extracted features.

– **The selection threshold value.** Noise may introduce additional undesired clusters if the threshold value is set too low. The next experiment surveys the number of clusters found, as a function of the threshold value, and of the noise level. Synthetic data is generated with one feature, and for a number of different noise levels. Using this data, the number of clusters is monitored while slowly increasing the threshold until only one cluster (the synthetic feature) is found. The lowest threshold value that results in one feature is called the cut-off threshold value. It is an important value since it gives us the minimum threshold value that distinguishes the feature from the noise. The cut-off should be as low as possible, as higher threshold values result in smaller features. Therefore, the cut-off threshold will be used in further experiments, because it depends not only on the noise level, but also on the other feature extraction settings.
– **The cluster threshold.** The cluster threshold is a very useful parameter, since small irrelevant clusters are removed by it. In many cases (especially if noise is involved) the selection results in single unconnected nodes that just happen to satisfy the selection criteria, but are not significant. The cluster threshold is often an adequate remedy to filter out these undesired features. Therefore, we determine the cut-off threshold for different noise levels, as a function of the cluster threshold.
– **Connected component definition.** The neighbour definition will affect the number of clusters found. The 1D-definition is more strict than the others, and will result in more and smaller clusters, which amplifies the effects of the cluster threshold. In order to test this, the cut-off threshold is determined for all three definitions as a function of the cluster threshold, for one given noise level.

5 Results

5.1 The accuracy of the ellipsoid-fitting method

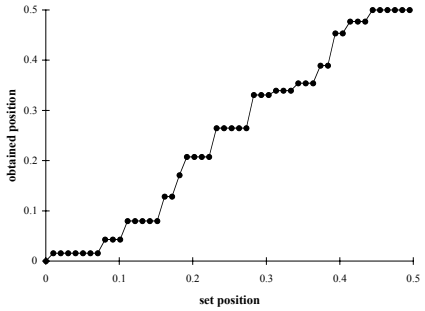

Fig. 5. Stepwise movement of the position within a cell.

First, the accuracy of the ellipsoid fitting method is established using the experiment described in section 4.1. Figure 5 shows the results of the accuracy tests for the position detection. The center position of the sphere starts at one corner node of the cell (relative position = 0), and ends in the center of the cell (relative position = 0.5). The obtained position is plotted as a function of the input position. It changes discontinuously every time a node enters or exits the moving sphere. Thus, a stepwise update of the position is found. The average distance to the diagonal is the average relative error of the position detection. Similar stepwise results are

obtained for the axis and orientation detection. Since the attributes were varied within cell size, we may conclude that the ellipsoid-fit method detects shifts within sub-cell level.

As may be expected, the accuracy becomes better when the grid is more dense. Figure 6 shows the errors as a function of the number of selected nodes in the cluster. The figure clearly shows the exponential decrease of the error with respect to the number of nodes. The errors are below 7% when the clusters consist of more than 15 nodes. Thus, the ellipsoid attributes are accurate if a cluster threshold of 15 nodes is used.

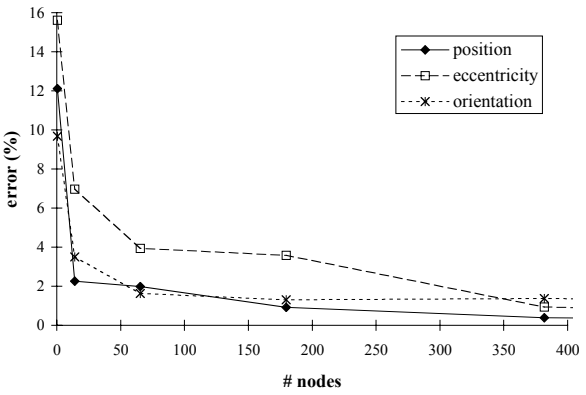

Fig. 6. Obtained errors for the ellipsoid attributes.

5.2 Robustness of the method

Now that the accuracy has been assured, the robustness of the method with respect to noise is investigated using the experiments described in section 4.2. First the number of clusters is determined as a function of the selection threshold value, and as a function of the noise level.

Figure 7 shows that for small thresholds a large number of clusters is found. This is an obvious results since the noise causes many node values to rise above the threshold level. If noise is added with an SD > 10, the number of clusters first increases with increasing threshold values because many nodes connect to form a large cluster which breaks up while increasing the threshold. In the end, a threshold value is found where only one feature remains. This value is the cut-off threshold; it becomes larger as the noise level increases.

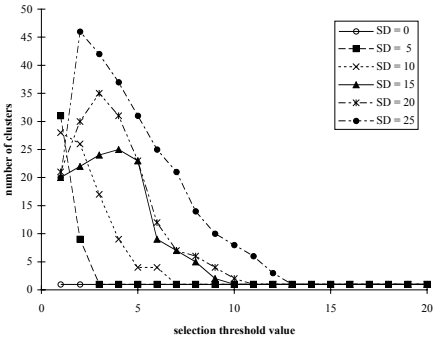

Fig. 7. The number of clusters as a function of the selection threshold value and the noise level.

The cut-off threshold is examined as a function of the noise level and the cluster threshold. Figure 8 shows that the cut-off threshold increases with increasing noise, still the cut-off threshold remains low for a large cluster threshold. Using a cluster threshold of at least 20 nodes, it suffices to use a selection threshold value of 1*SD in order to eliminate all clusters due to noise. If smaller features are expected, then a cluster threshold of 5 nodes in combination with a selection

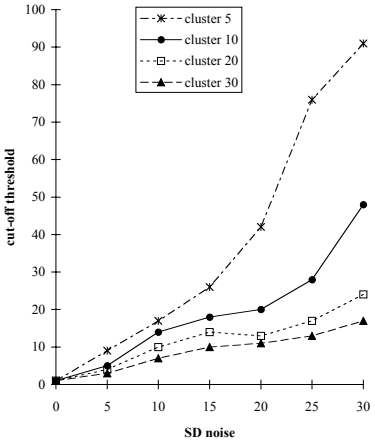

Fig. 8. The cut-off threshold value as a function of the noise level and the cluster threshold.

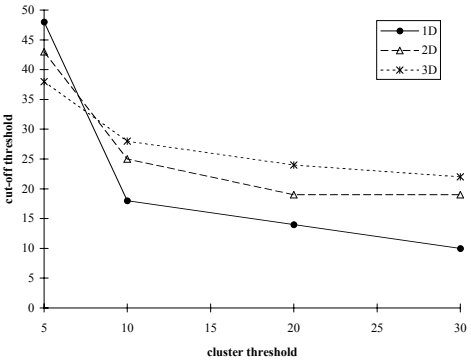

Fig. 9. The cut-off threshold value as a function of the connected component definition and the cluster threshold.

threshold value of at least 2*SD suffices, provided that the threshold value is significantly smaller than the maximum data value in the feature.

The next experiment determines the cut-off threshold for the three connected component definitions, as a function of the cluster threshold, with the noise level set to SD = 15.0. The result is plotted in figure 9, which shows that the cut-off threshold drops as the cluster threshold increases, and also that it drops faster if the connected component definition is set to 1D-connected. This definition results in more, smaller clusters which are easier to discard by the cluster threshold. Therefore, in case of noisy data, one should use the 1D-connected component definition.

5.3 Robustness of the calculated attributes

Finally, the effect of noise on the obtained ellipsoid attributes are investigated with optimal extraction parameters. Noise with an SD = 15.0 is added to the data, and 1D-connected component definition is used in combination with a cluster threshold of 15 nodes. In figure 10 errors are plotted as a function of the selection threshold value.

The figure clearly shows a large error in position for low selection threshold values. This is caused by the large cluster throughout the entire domain due to noise. Also, the error increases for large thresholds, caused by the fact that the feature is small and additional selected nodes due to noise affect the position significantly. Between the two extremes the errors are stable, thus the results of the method are relatively invariant to noise.

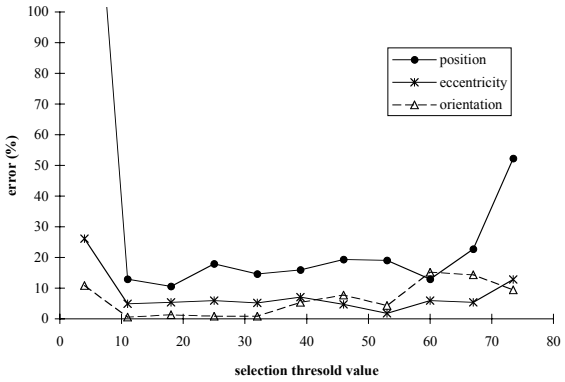

Fig. 10. The errors of the ellipsoid detection of data with noise (SD = 15.0).

6 Conclusions

During the execution of the experiments it became clear that a lot can be learned about the behaviour of the feature extraction method. Therefore, we consider this type of experiment extremely important for the exploration and validation of visualization techniques, and we recommend to do similar experiments with any new visualization method. In this case, the following conclusions can be drawn:

1. The ellipsoid attributes can be estimated with an accuracy below the cell-size level. The errors decrease for increasing grid density, i.e. for clusters with more nodes. A cluster threshold of 15 nodes results in errors below 7%.
2. In case of noisy data, the feature extraction parameters can be set in such a way that spurious features can be filtered out. A statictical analyses is needed in order to give the exact requirements, but based on these experiments the following guidelines for the feature extraction parameters can be given:
   - The cluster threshold is a powerful parameter to discard spurious features due to noise. Large cluster thresholds result in correct feature extraction, even close to the noise level. This is caused by the coherence in space of the selected nodes.
   - In case of noise, the use of the 1D-connected component definition is recommended, since this results in smaller clusters which are easier discarded by the cluster threshold.
3. The obtained ellipsoid attributes are stable despite the presence of noise. This means that the ellipsoid attributes are relatively invariant to noise.

7 Future research

The results of the experiments described in this paper pave the way for a number of interesting studies in the future.

- Small spurious features may be filtered out by morphological operators like opening and closing. This may enhance the effects of the cluster threshold.
- Further statistical analysis can be done on the extraction of features below noise level. Besides coherence in space, coherency in time may be exploited: e.g. if a feature is detected at one time, a prediction can be made of the feature some time later, this prediction can be used to extract the new feature. This suggests a predictive approach for feature tracking in time-dependent data.

Acknowledgments

This work is supported by the Netherlands Computer Science Research Foundation (SION), with financial support of the Netherlands Organization for Scientific Research.

References

1. D.C. Banks and B.A. Singer. A predictor-corrector technique for visualizing unsteady flow. *IEEE Trans. on Visualization and Computer Graphics*, 1(2):151–163, June 1995.
2. J.L. Helman and L. Hesselink. Visualization vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, 1991.
3. H.G. Pagendarm and B. Seitz. An algorithm for detection and visualization of discontinuities in scientific data fields applied to flow data with shock waves. In P. Palamidese, editor, *Scientific Visualization: Advanced Software Techniques*, pages 161–177. Ellis Horwood Limited, 1993.
4. F.J. Post, T. van Walsum, F.H. Post, and D. Silver. Iconic techniques for feature visualization. In G.M. Nielson and D. Silver, editors, *Proc. Visualization '95*, pages 288–295. IEEE Computer Society Press, 1995.
5. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, second edition, 1992.
6. F. Reinders, F.H. Post, and H.J.W. Spoelder. Feature extraction from pioneer venus ocpp data. In W. Lefer and M. Grave, editors, *Visualization in Scientific Computing '97*, pages 85–94. Springer Verlag, April 1997.
7. D. Silver and N.J. Zabusky. Quantifying visualizations for reduced modeling in nonlinear science: Extracting structures from data sets. *J. of Visual Communication and Image Presentation*, 4(1):46–61, March 1993.
8. T. van Walsum, F.H. Post, D. Silver, and F.J. Post. Feature extraction and iconic visualization. *Trans. on Visualization and Computer Graphics*, 2(2):111–119, 1996.