

Extracting Feature Lines from 3D Unstructured Grids

Kwan-Liu Ma and Victoria Interrante

Institute for Computer Applications in Science and Engineering (ICASE)
NASA Langley Research Center, Hampton, Virginia 23681-0001
kma@icase.edu, interran@icase.edu

ABSTRACT

This paper discusses techniques for extracting feature lines from three-dimensional unstructured grids. Our twin objectives are to facilitate the interactive manipulation of these typically very large and dense meshes, and to clarify the visualization of the solution data that accompanies them.

We describe the perceptual importance of specific viewpoint-dependent and view-independent features, discuss the relative advantages and disadvantages of several alternative algorithms for identifying these features (taking into consideration both local and global criteria), and demonstrate the results of these methods on a variety of different data sets.

1. INTRODUCTION

The three-dimensional unstructured grids used for the numerical simulation of 3D flow are generally very large in size and irregular in both shape and resolution. Even the simplest renderings of many of these meshes can be time-consuming to generate on an average desktop workstation, and once an image is produced it can be difficult to adequately perceive relevant geometric structure through the tangle of overlapping lines.

By way of example, figure 1a shows a typical rendering of a surface mesh from a CFD grid. This mesh consists of 127,544 triangles, and was extracted from an unstructured volume grid containing 4,607,585 tetrahedra across 804,056 points. Figures 1b-d illustrate some of the standard techniques that are commonly used to decrease the rendering time required for the display of such meshes. These techniques include skipping every n th element, rendering only the gridpoints of the surface mesh, and substituting a coarser grid.

We propose that by directly identifying and extracting a small set of perceptually significant geometric features from the surface mesh and displaying these in place of the full model, we may both considerably decrease the rendering latency, compared to the above-mentioned methods, and at the same time improve the

comprehensibility of the structural data. For example, using a feature line-only display, researchers can interactively reorient a very large dataset, focus in on a particular area of interest, and then switch back to a slower-rendering full mesh display after the desired viewing parameters have been chosen.

When rendering time is not of critical concern, it can be useful to display the feature lines of a grid in conjunction with the polygonal or volume-rendered solution data to help highlight the essential structural detail of the underlying geometry while preserving the visual prominence of the flow information.

2. PREVIOUS WORK

The use of feature lines to enhance the communication of geometrical information has a long history in computer graphics. Dooley and Cohen [7] stressed the perceptual importance of silhouette, contour, discontinuity and, in certain cases, isoparametric lines for clarifying geometrical structure in a complex model. They showed how various techniques from technical illustration could be used to successfully represent these feature lines in images of CAD models containing multiple overlapping surfaces, but did not discuss how such lines might be identified. Pearson and Robinson [21] showed how images resembling artists' line drawings could be computed from two-dimensional digital photographs and they described how the use of such representations could enable improved bandwidth for visual communication across low speed data lines. Saito and Takahashi [23] proposed enhancing shaded renderings of polygonal models with lines representing the locus of first and second order depth discontinuities, and they described how these lines could be calculated by applying standard gradient operators to a two-dimensional depth map of the scene. Banks [1] demonstrated the value of emphasizing silhouette regions and lines of intersection in renderings of transparent two-dimensional surfaces in 4D, and Interrante, Fuchs and Pizer [12] suggested explicitly marking ridge and valley lines on transparent skin surfaces in radiation treatment plans to facilitate the intuitive appreciation of familiar shape-based features.

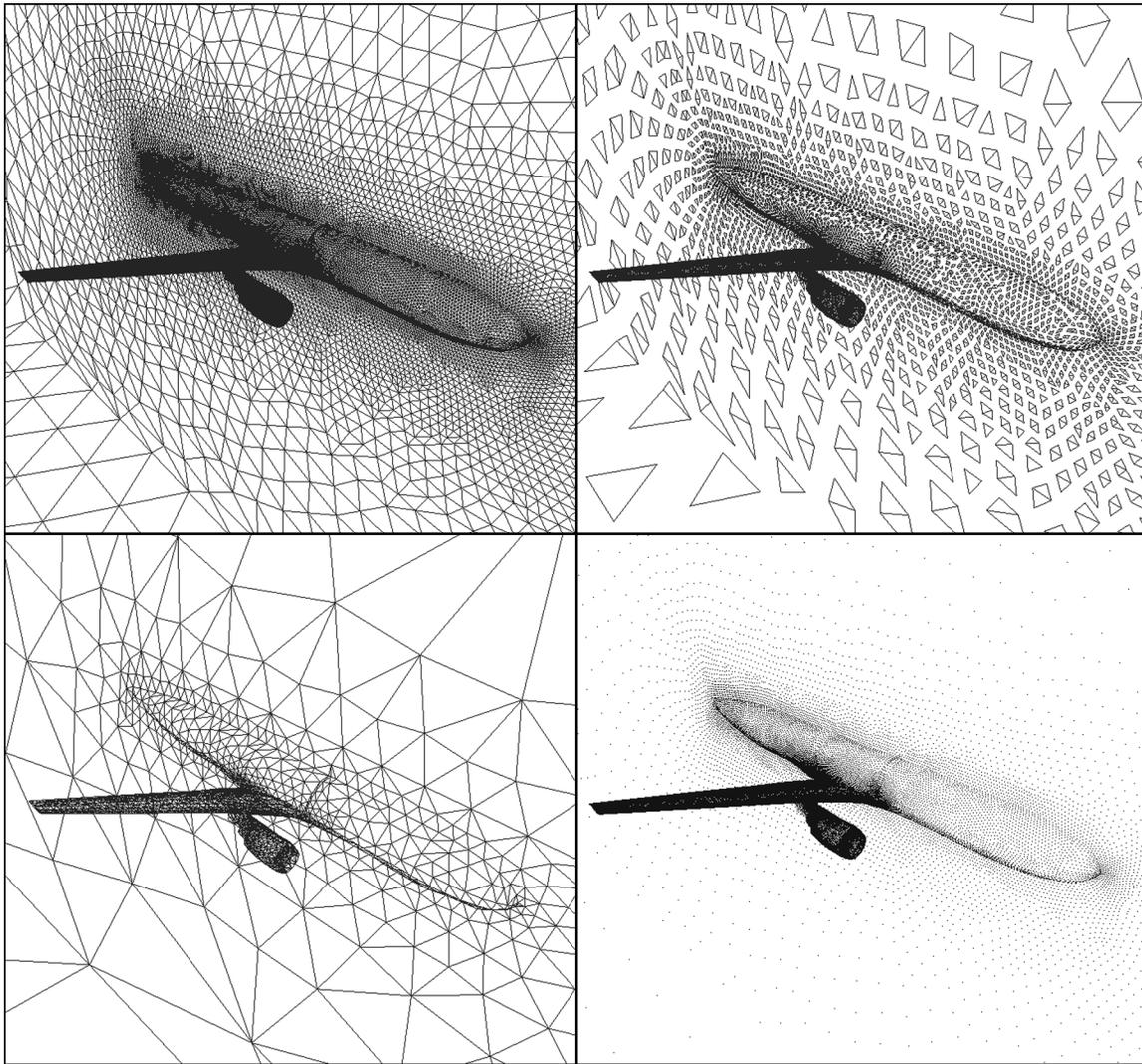


Figure 1: Previous methods for rendering large grids. Clockwise from upper left: a full surface mesh for the numerical simulation of airflow over a low wing transport aircraft; the same mesh, with every 4th grid element displayed; with only the grid points displayed; a lower resolution version of the mesh.

Mesh simplification also offers the possibility of considerable data reduction while maintaining fidelity to the original model; the best of these methods preserve sharp edges either implicitly [6] or explicitly [10]. However, mesh simplification offers only a partial solution to the problem of effectively visualizing dense grids when the objective is to highlight only the most perceptually relevant geometric features of a model.

When part boundary definitions are predefined and included along with the mesh representation, the boundary elements may of course be explicitly displayed [26], and in some cases such a representation will fulfill the purposes our algorithms are intended to address. In many cases, however, adequate segmentation information is not readily available, and grid zone boundaries, when indicated, do not always correlate well with the structural features of the

object. The techniques described in this paper are particularly applicable under these circumstances.

3. FEATURE LINES

Cross-cultural research in pictorial representation [13] indicates that line drawings are a natural and universally understood means of graphical communication, and visual theorists have suggested the possibility of an intrinsic relationship between this type of representation and the way our visual system processes and stores information (e.g. Marr's [16] "primal sketch" theory of visual information processing). The two types of geometric features most often represented in line drawings are discontinuities of depth (silhouettes and contours) and discontinuities of curvature (sharp ridges and valleys).

3.1 Silhouettes and Contours

Probably the most essential lines in any line drawing are the silhouette and contour lines. Silhouette and contour curves are the two-dimensional projection of the points on a surface in 3-space where the surface normal is orthogonal to the line of sight [14]. Contour lines mark the internal depth discontinuities in a two-dimensional image, and silhouette lines separate the figure from the ground. Our perception of 3D shape, and of object coherence in the presence of foreground occluders [19], is induced and affected by this boundary information. Richards et al. [22] describe how the 3D shape of a smoothly curving surface can be directly inferred, in the vicinity of an occluding contour, solely from the sign of the curvature of the silhouette line; Barrow and Tenenbaum [2], and many others, describe how various aspects of the 3D structure of an object or a scene can be inferred from relational characteristics of the depth and intensity discontinuity lines in a 2D image.

Procedures for identifying silhouette edges in polygonal models are very straightforward and have been known for many years. In the case of orthographic projection, one may easily identify silhouette edges in a connected mesh by simply taking the dot product of the transformed view direction with the surface normal of each triangle and looking for differences in sign between adjacent elements. When a perspective projection is used things get slightly more complicated since the viewing direction is not constant over all areas of the scene. However Hultquist [11] gives a simple algorithm that can be used to identify silhouette edges under these conditions, based on changes in the direction (clockwise vs. counterclockwise) of the ordering of the vertices of adjacent triangles after projection onto the image plane.

Because they are viewpoint-dependent, silhouette and contour curves have to be recalculated continuously as an object is repositioned in space. For large datasets, and particularly under perspective projection, it can be prohibitively expensive to perform all of the indicated computations on the fly. However, Markosian *et al.* [15]

have recently demonstrated a very effective technique for identifying and displaying nearly all of the silhouette and contour edges of a reasonably complex polygonally-defined object at interactive frame rates. Their method succeeds by relying on a scattered sampling of a very small fraction (i.e. 1%) of the possible edges to detect silhouette fragments, which can then be followed, based on connectivity information, to complete the curves. The odds of detecting a silhouette fragment are improved by weighting the probability of selecting a particular edge for testing according to the dihedral angle between the triangles that share it. Inter-frame coherence is also exploited to improve the likelihood of detecting and retaining silhouette and contour fragments.

Using some form of hidden line removal can considerably improve the clarity of an edge-based representation. The line rendering style itself may also be adjusted to achieve various different effects, as shown in figure 2.

3.2 Ridges and Valleys

A second class of feature lines that are often included in line drawing representations are the lines of intensity discontinuity. These lines are generally viewpoint-independent and correspond to the places on an object where the surface normal changes direction abruptly. On polygonally-defined surfaces such as the numerical calculation grids that we are concerned with here, however, there is a one-to-one correspondence between discontinuity edges and the lines of the mesh. How can we differentiate the few perceptually relevant edges from the many others? There are several considerations that need to be weighed. The results of psychophysical experiments in shape perception [4] support theories of object recognition [3] based on the subdivision of complex shapes into components along the lines defined by local minima of negative curvature [9]. These are the “valley” lines, and they are important to shape understanding regardless of the sharpness of the curvature across them.

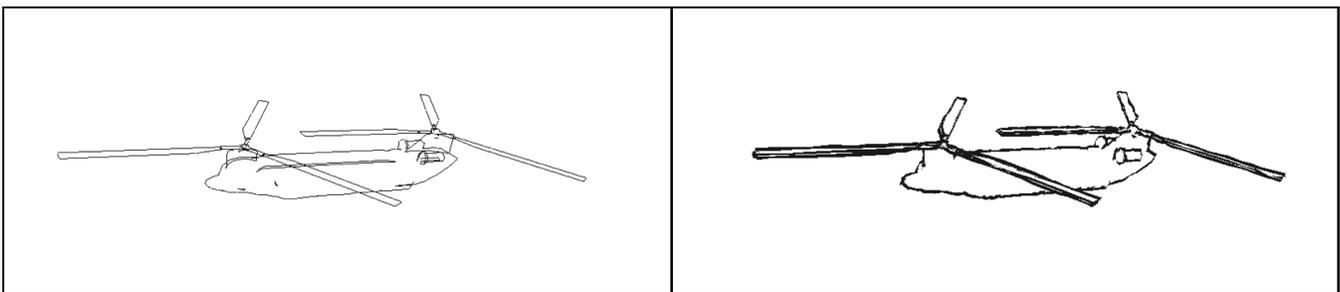


Figure 2: Silhouette curves can be drawn in different styles to achieve different visual effects. Left: the silhouette edges in a model of a tandem helicopter, without hidden line removal. Right: the same edges, rendered in a more “artistic” style, and with hidden lines removed.

Ridge lines, which are the convex analogue of valleys, appear on the other hand to be perceptually relevant only to the extent that they mark areas of significant curvature discontinuity.

Ridge and valley lines may be identified as the locus of points where the normal curvature assumes an extreme value along a line of curvature [14], and several authors [17, 24, 12] have described techniques, based upon this criterion, for computing ridge and valley lines on isovalue surfaces defined in volumetric data. The possibility of using such an approach to identify ridge and valley lines on surfaces defined solely by planar facets hinges on the ability to obtain sufficiently accurate curvature estimates at arbitrary points across these surfaces. Taubin [24] describes an efficient method for estimating principal curvatures and principal directions at the vertices of a polygonally-defined object; Chen and Schmitt [5] describe an earlier, alternative approach, and other, related algorithms, primarily applicable to height functions, are summarized in [8].

Since our primary objective for this application is speed, and approximate results will be sufficiently adequate for our purposes, we look instead at some far simpler approaches.

One very simple way to extract a small set of potentially important edges from a connected mesh is to calculate, for each pair of adjacent triangles, the scalar difference between each of the axial components of the two surface normal directions and to define as feature lines all edges across which any one of these values exceeds some user-defined global threshold. Figure 3 shows the results of using this approach to extract potential feature lines from a surface mesh over a commanche helicopter.

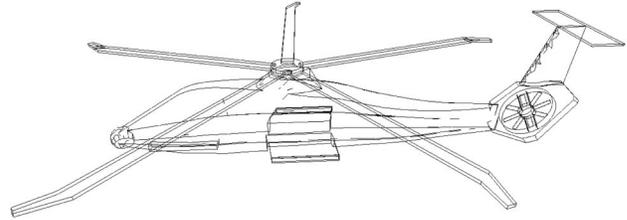


Figure 3: Feature lines defined by a global threshold on the change in the direction of the surface normal across each edge.

While the overall effect is encouraging, particularly for this model, there are obviously many instances in which a method like this will fail to produce satisfactory results. In particular, where the surface contains features that exist at different scales, there will be no single threshold angle that can successfully differentiate the desired edges from the undesired ones. In tightly curved areas, all of the mesh elements will be oriented at a relatively sharp angle to their neighbors, yet it may not be appropriate to display any of these edges. One very nice advantage of this method, however, is that, due to its simplicity, it succeeds on models (such as the one shown above) in which the orientation of the triangles is inconsistent. The problem of computing consistently outward-facing surface normals across a complex object in which the vertices of each polygon are specified in inconsistent order is not trivially resolved; Murali and Funkhouser [18] suggest one new approach and briefly review some others.

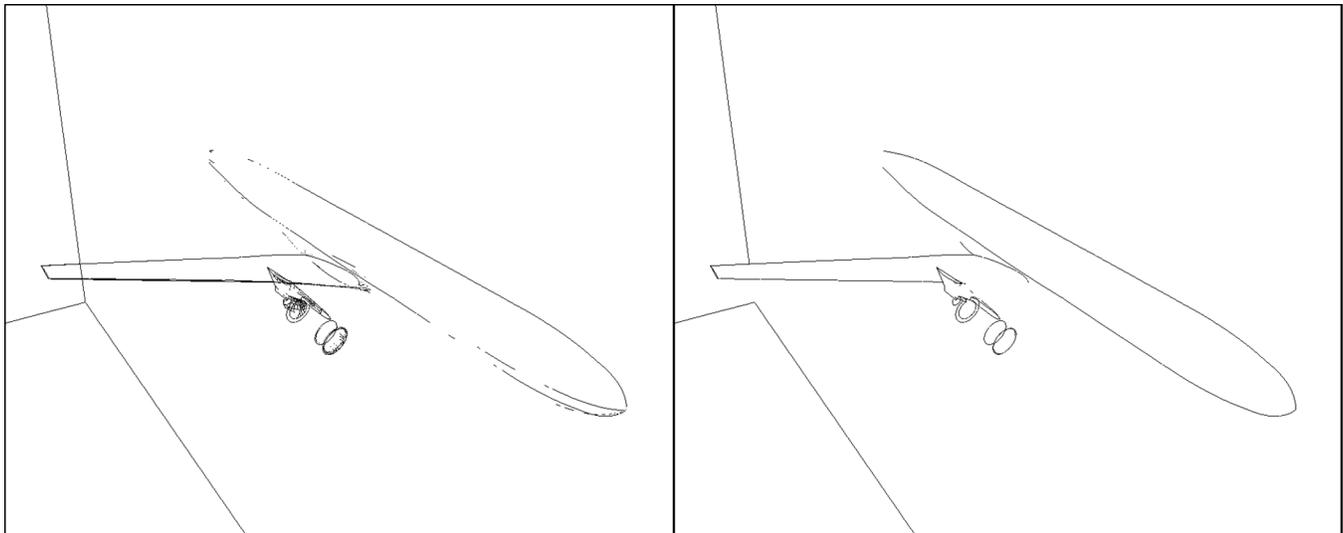


Figure 4: A comparison of images obtained using global vs. local discontinuity detection criteria. Left: the visibility of an edge is determined according to the magnitude of the difference in the directions of normals of the adjacent triangles. Right: an edge is shown if the angle across it is greater, by a certain amount, than the angles across the other two edges of each of the triangles that share it.

An alternative approach to the definition of major discontinuity lines begins from the hypothesis that the visual importance of any particular edge can be related to the extent to which the curvature across that edge (indicated by the angle between the normals) is significantly larger than the curvatures in other directions in that vicinity (indicated by the angles between the normals across the other two edges of the same triangles). Such an approach more successfully resolves the problem of reliably detecting feature lines simultaneously across multiple scales, but it has the disadvantage of breaking down at corners, where a triangle may have more than one locally important edge.

Figure 4 illustrates some of the differences between the two feature line definition procedures describe above. Note the cluster of small triangles in the highly curved region of the engine in the leftmost image. A threshold that is high enough to remove these lines will obliterate the more subtle discontinuities between the aircraft body and the side of the mesh. The image on the right, which is calculated using a locally rather than globally-defined threshold, does not share this problem. We have found in practice that good results can be achieved using a combination of both local and global criteria.

It is sometimes beneficial to obtain larger scale curvature estimates in a locally smooth region by averaging the normal directions of neighboring triangles. Special precautions need to be taken, if this is done, to avoid averaging across significant discontinuity edges.

4. FURTHER RESULTS

Figures 5 and 6 demonstrate how feature lines can be used in conjunction with surface or volume rendering to clarify the essential structure of a grid while preserving the visual prominence of the solution data over it. When feature lines are rendered together with opaque polygonal data, it is necessary to explicitly ensure that the lines will not be hidden by the polygons over which they are supposed to lie. A very simple, approximate solution is to slightly displace each of the feature lines in the direction of the surface normal at each edge, and this is what we have done in figure 5; more accurate results can be achieved by using a “stencil buffer” to explicitly enforce the precedence of line data over polygonal data wherever the two coexist at the same depth [20].

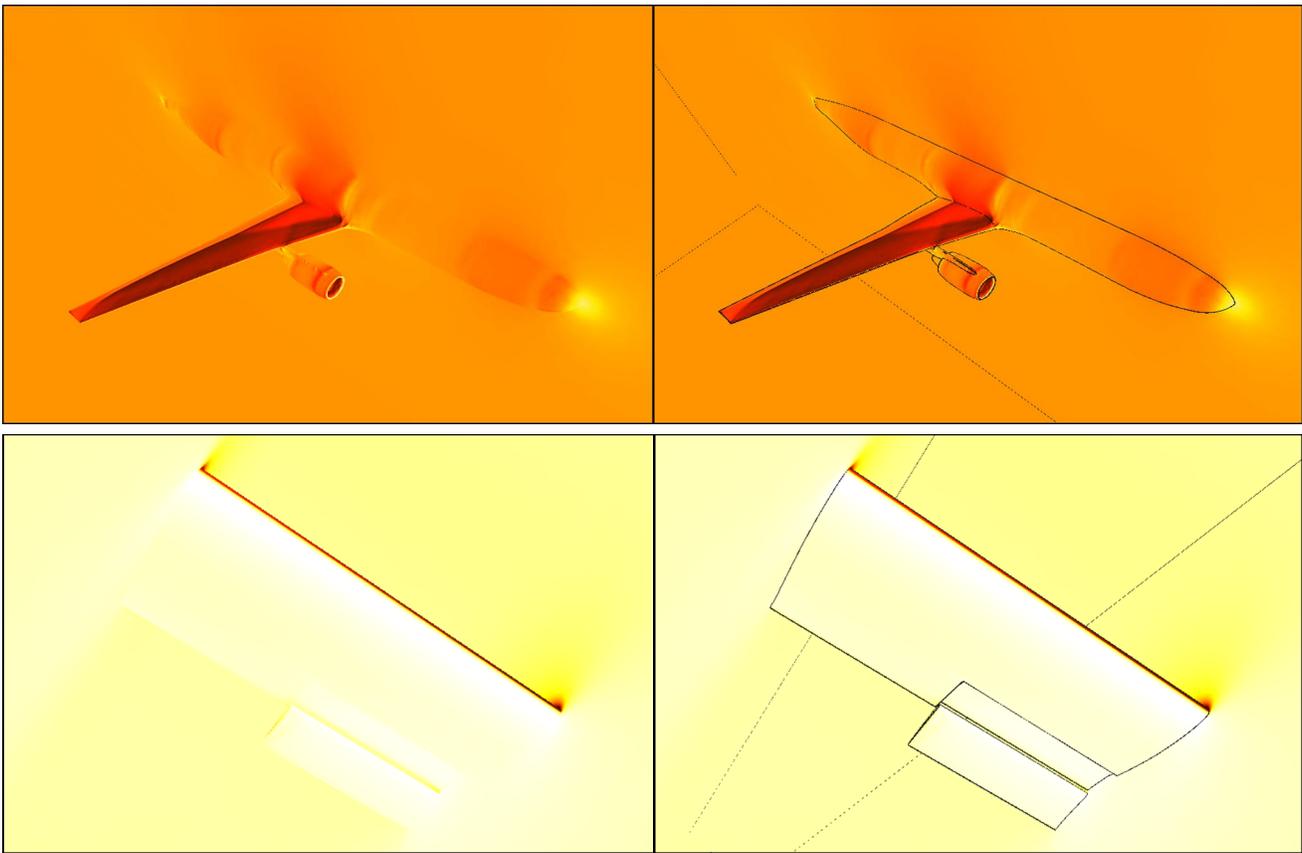


Figure 5: An illustration of the clarity that a display of feature lines can lend to a shaded rendering. The feature lines in each case are the silhouette and ridge/valley edges. Surfaces are colored according to the density of the flow across the mesh.

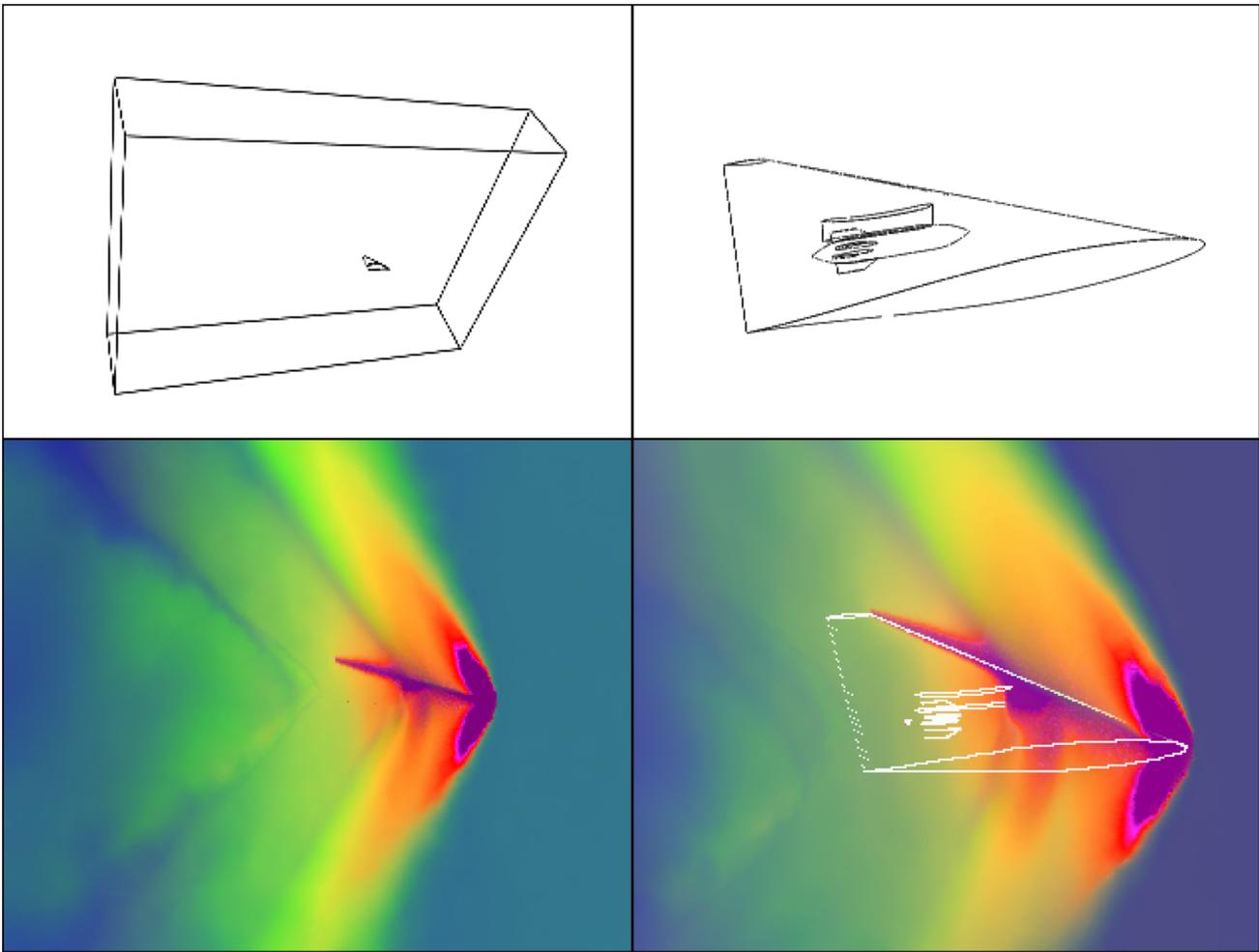


Figure 6: An illustration of the clarity that feature lines can add to volume rendered images. Upper left: a distant view places the object of interest in context; upper right: a closer view reveals the structural details of the wing and underlying missile; lower right: rendered in conjunction with the volume data, the feature lines explicitly define the location of the wing while preserving the visibility of the transparently rendered solution information; lower left: without this structural information, the volume rendered solution data can be difficult to interpret.

5. DIRECTIONS FOR FUTURE WORK

Neither spherical nor cylindrical structures are especially well described by either of the view-independent feature line detection algorithms that we have implemented. Meshes containing a principal spherical or cylindrical component may be so inadequately represented that whole pieces can be inadvertently left out, as can be seen from the central image of figure 7. Although indications of the missing segments can be restored by complementing the ridge- and valley-like feature lines of section 3.2 with feature lines representing silhouette edges, as shown in the bottom-most image of figure 7, it would be preferable to come up with a fully viewpoint independent technique that is capable of elegantly portraying these nongeneric structures at unambiguously interactive rates. A

representation based on the illustration of object cross-sections locally perpendicular to the direction of the medial axis may show some promise in this regard.

6. CONCLUSIONS

We have discussed the motivation for extracting and displaying perceptually relevant feature lines from unstructured triangular meshes. We described several alternative methods for defining these feature lines, and outlined their relative strengths and weaknesses. Each of these methods is very fast and requires little or no preprocessing of the data. They allow the clarity of the display to be enhanced while enabling complex models to be interactively manipulated in a fraction of the time that is ordinarily required for a rendering of the complete dataset.

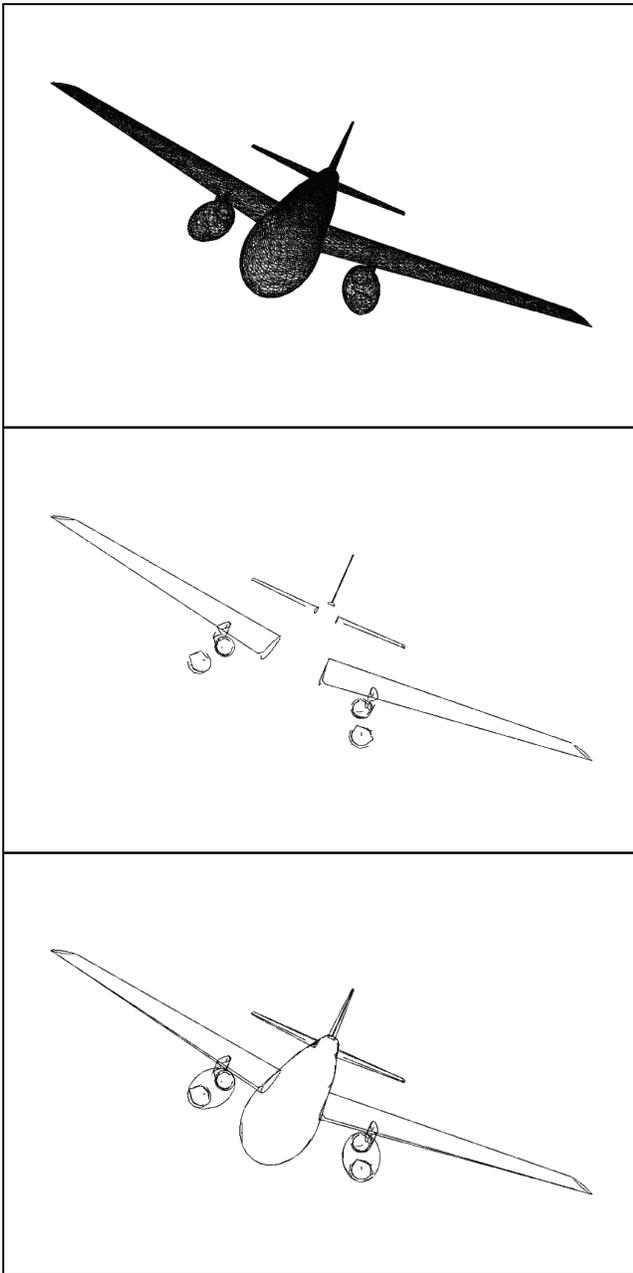


Figure 7: An example of the kind of model that cannot be successfully depicted by either of the viewpoint-independent feature line detection algorithms that we describe. Top: the full surface mesh over an airplane; center: the ridge and valley lines alone; bottom: the ridge and valley lines in combination with the silhouette edges.

ACKNOWLEDGMENTS

This work was supported by ICASE under NASA contract NAS1-19480. The low wing transport and wing flap grid datasets were generated by Dimitri Mavriplis. The commanche and tandem helicopter datasets were generated by Can Ozturan.

REFERENCES

- [1] Banks, D. "Interactive Manipulation and Display of Two-Dimensional Surfaces in Four-Dimensional Space", in *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, pp. 197-207.
- [2] Barrow, H. and Tenenbaum, J. "Interpreting Line Drawings as Three-Dimensional Surfaces", *Artificial Intelligence*, vol. 17, (1981), pp. 75-116.
- [3] Biederman, I. "Human Image Understanding: Recent Research and a Theory", in Human and Machine Vision, A. Rosenfeld, ed. Academic Press, 1985, pp. 13-57.
- [4] Braunstein, M. L., Hoffman, D. D., and Saidpour, A. "Parts of Visual Objects: an experimental test of the minima rule", *Perception*, vol. 18, (1989), pp. 817-826.
- [5] Chen, X. and Schmitt, F. "Intrinsic Surface Properties from Surface Triangulation", *Proceedings of the 2nd European Conference on Computer Vision (ECCV)*, Springer-Verlag, 1992, pp. 739-743.
- [6] Cohen, J., Varshney, A., Manocha, D., Turk, G., Weber, H., Agarwal, P., Brooks, F., and Wright, W. "Simplification Envelopes", in *Proceedings of SIGGRAPH '96*, pp. 119-128.
- [7] Dooley, D., and Cohen, M. "Automatic Illustration of 3D Geometric Models: Lines", in *Proceedings of the 1990 Symposium on Interactive 3D Graphics*, pp. 77-82.
- [8] Flynn, P. and Jain, A. "On Reliable Curvature Estimation", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1989, pp. 110-116.
- [9] Hoffman, D. D., and Richards, W. A. "Parts of Recognition", *Cognition*, vol. 18, (1984), pp. 65-96.
- [10] Hoppe, H. "Progressive Meshes", in *Proceedings of SIGGRAPH '96*, pp. 99-108.
- [11] Hultquist, J. "Backface Culling", in Graphics Gems, A. S. Glassner, ed. Academic Press, 1990, pp. 346-347.
- [12] Interrante, V., Fuchs, H., and Pizer, S. "Enhancing Transparent Skin Surfaces with Ridge and Valley Lines", in *Proceedings of Visualization '95*, pp. 52-59.
- [13] Kennedy, J. M. A Psychology of Picture Perception, Jossey-Bass, 1974.
- [14] Koenderink, J. Solid Shape, MIT Press, 1990.
- [15] Markosian, L., Kowalski, M., Trychin, S., Bourdev, L., Goldstein, D., and Hughes, J. "Real-Time Nonphotorealistic Rendering", to appear in *Proceedings of SIGGRAPH '97*.

- [16] Marr, D. "Early Processing of Visual Information", *Philosophical Transactions of the Royal Society of London, Series B. Biological Sciences*, vol. 275, (October 1976), pp. 483-519.
- [17] Monga, O., Benayoun, S., and Faugeras, O. "From Partial Derivatives of 3D Density Images to Ridge Lines", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, June 1992, pp. 354-359.
- [18] Murali, T., and Funkhouser, T. "Consistent Solid and Boundary Representations from Arbitrary Polygonal Data", *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, pp. 155-162.
- [19] Nakayama, K., Shimojo, S., and Silverman, G. "Stereoscopic Depth: its relation to image segmentation, grouping, and the recognition of occluded objects", *Perception*, vol. 18, (1989), pp. 55-68.
- [20] Neider, J., Davis, T., and Woo, M. OpenGL Programming Guide, Addison-Wesley, 1993, p. 397.
- [21] Pearson, D. E., and Robinson, J. A. "Visual Communication at Very Low Data Rates", *Proceedings of the IEEE*, vol. 4, (April 1985), pp. 795-812.
- [22] Richards, W., Koenderink, J., and Hoffman, D. "Inferring Three-Dimensional Shapes from Two-Dimensional Silhouettes", *Journal of the Optical Society of America, A*, vol. 4, (July 1987), pp. 1168-1175.
- [23] Saito, S., and Takahashi, T. "Comprehensible Rendering of 3-D Shapes", in *Proceedings of SIGGRAPH '90*, pp. 197-206.
- [24] Taubin, G. "Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation", *Proceedings of the 5th International Conference on Computer Vision (ICCV)*, 1995, pp. 902-907.
- [25] Thirion, J.-P., and Gourdon, A. "The Marching Lines Algorithm: new results and proofs", *rapport de recherche INRIA*, no. 1881-1, April 1993, 27 pages.
- [26] Walatka, P., Buning, P., Pierce, L., and Elson, P. PLOT3D User's Manual, NASA Technical Memorandum 101067, March 1990.