New Applications of the Verdict Library for Standardized Mesh Verification
Pre, Post, and End-to-End Processing

Philippe P. Pébay[1*], David Thompson[1*], Jason Shepherd[2*], Patrick Knupp[2*], Curtis Lisle[3‡], Vincent A. Magnotta[4‡], and Nicole M. Grosland[4‡]

[1] Sandia National Laboratories, P.O. Box 969, Livermore CA 94551, U.S.A. {pppebay,dcthomp}@sandia.gov
[2] Sandia National Laboratories, P.O. Box 5800, Albuquerque NM 87185, U.S.A. {jfsheph,pknupp}@sandia.gov
[3] Knowledgevis, Llc, 1905 South Blvd, Maitland, FL 32751, U.S.A. clisle@ieee.org
[4] The University of Iowa, Iowa City, IA 52242, U.S.A. {vincent-magnotta,nicole-grosland}@uiowa.edu

Summary. Verdict is a collection of subroutines for evaluating the geometric qualities of triangles, quadrilaterals, tetrahedra, and hexahedra using a variety of functions. A quality is a real number assigned to one of these shapes depending on its particular vertex coordinates. These functions are used to evaluate the input to finite element, finite volume, boundary element, and other types of solvers that approximate the solution to partial differential equations defined over regions of space. This article describes the most recent version of Verdict and provides a summary of the main properties of the quality functions offered by the library. It finally demonstrates the versatility and applicability of Verdict by illustrating its use in several scientific applications that pertain to pre, post, and end-to-end processing.

1 Introduction

Verdict is a library for evaluating the geometric qualities of regions of space. A region of space can be, for example, a finite element or a volume associated with a finite volume mesh. This paper presents the design of the library and its application to several problems of interest. This introduction briefly

* These authors were supported by the United States Department of Energy, Office of Defense Programs. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under contract DE-AC04-94-AL85000.
‡ These authors gratefully acknowledge the financial support provided by NIH Awards EB005973, EB001501, and U54 EB005149.

defines quality functions and discusses the history and motivation behind the creation of Verdict. The paper continues with the practical aspects of obtaining and using the library. Finally, several applications currently using Verdict are reviewed and followed with a list of quality functions provided by the library. The goals of this paper are to increase the visibility Verdict by illustrating its use in several scientific applications, and to propose it as a reference implementation for geometric qualities.

1.1 Quality Functions

In general, one can state that

Definition 1. *A* quality function *is a function that maps any region of space – possibly along with a parametrization defined over it – to a non-negative real number.*

Since a quality function maps an entire region of space to a single real number, it cannot completely describe the shape of its corresponding region. For example, the set of triangular shapes (*i.e.*, equivalence classes of triangles that are identical up to similarity) can be described as a 3-dimensional space. Therefore, some information is necessarily lost when a triangular shape is mapped to the real line. Thus, most quality functions are used to identify a single type of problem with a region's shape. Moreover, it is widely recognized that acceptable quality of regions depends on the particular problem of interest and its solution. For example, equilateral triangles do not always constitute the best region shape for every finite element simulation using triangular elements. The quality functions presently in Verdict do not take the PDE nor the solution into account when evaluating region quality. Therefore, when the desired optimal shapes can be specified by means of an appropriate geometric transform, regions should be transformed accordingly before quality is evaluated. This ensures, in this case, that distances are computed in the correct anisotropic metric [14]. Nevertheless, in applications which possess physics that are nearly homogeneous and isotropic, Verdict can also be used directly, independent of an application, for detecting inverted and other problematic regions.

Each quality function may take on any value on the real number line, but a typical use of quality functions is to focus on subsets of this range that are of interest for the application at hand. In particular, if one is interested in filtering out bad elements, as illustrated by applications in §3, a quality function with range $[1, \infty[$, where the value 1 is attained at and only at optimal geometric shapes, and tends to ∞ where regions are degenerate, may be advantageous. Therefore, we say that quality functions that have this property are *filterable*.

1.2 Assessing Quality

Verdict groups quality functions by the topological definition of the region on which they operate. The topological definition of a region is related to the number and type of geometric discontinuities on its boundary. For instance, regions with 8 boundary corners, 12 boundary edges, and 6 boundary faces (hexahedra) will have one corresponding set of quality functions. This is to be expected since the number and type of degenerate configurations vary with the topological definition. It is usually these degenerate boundary configurations where filterable quality functions should tend to $+\infty$. Examples where filterable quality functions should tend to $+\infty$ include crossed quadrilaterals, triangles with edges of vastly different lengths, and regions with coincident corner vertices. Some quality functions defined on regions of a given topology place additional caveats on their use. For instance, many quadrilateral quality functions are intended for planar quadrilaterals, and thus the numbers they may yield for non-planar regions are not guaranteed to be meaningful.

1.3 History of Verdict

Verdict has its roots in VERDE[5], a simple program to read ExodusII [33] meshes, and analyze them for possible problems, in particular in terms of finite element quality. VERDE, in turn, has roots in the CUBIT[6] mesh generation code. Many of the original quadrilateral and hexahedral quality functions, for example, were first coded in CUBIT based on the papers of [32] and later transferred over to VERDE. After a period of independent development, it was realized that VERDE and CUBIT did not yield the same results when assessing mesh quality. Consequently, Verdict was created so that both applications could share the same code and produce consistent results. Verdict was later extended to include quality functions for simplicial regions and the algebraic quality functions [16]. Verdict was initially licensed under the LGPL.

Meanwhile, the Visualization Tool Kit (VTK)[7] did not support general purpose mesh quality assessment. Due to the need for such a tool and the incompatibility between licenses, P. Pébay and D. Thompson generalized the `vtkMeshQuality`[8] class in 2004 to compute the quality of triangles, quadrilaterals, tetrahedra, and hexahedra using a variety of quality functions.

Following a change in Verdict's licensing scheme, from LGPL to BSD-style, it was decided in late 2006 to use Verdict in VTK for the same reasons that Verdict was initially created, by moving all quality functions from `vtkMeshQuality` to Verdict while retaining the best implementation when the same quality function was implemented in both software packages, using

[5] http://www.cs.sandia.gov/capabilities/VerdeMeshVerificationSuite/
[6] http://cubit.sandia.gov/
[7] http://www.vtk.org/
[8] http://www.vtk.org/doc/nightly/html/classvtkMeshQuality.html

vtkMeshQuality as a wrapper around Verdict, and resolving naming inconsistencies and redundancies. It is important to former Verdict users to note that the latter action has resulted in changes to Verdict's API, although efforts have been made to preserve backwards-compatibility as often as possible (*cf.* [39] for a summary of these changes).

2 Practicalities

The main goals of the library are (1) correctness of the implementation; (2) order invariance; and (3) computational efficiency. Correctness of the implementation is assessed empirically as Verdict has undergone an extensive amount of testing and debugging, and users are invited to report bugs and other problems they may encounter. Note that the quality functions in Verdict are all checked for overflow as follows: given a double-precision quality value q, if $q > 0$, then $q \leftarrow \min(q, D_{\max})$; otherwise $q \leftarrow \max(q, -D_{\max})$. An algorithm for computing quality is *order invariant* if the same result (to within machine truncation error) is returned regardless of the order in which nodes are specified, as long as the nodes specify the same region[9]. Where applicable, the quality functions in Verdict are verified against theory for order invariance. When multiple quality functions are requested for the same geometric region, an efficient implementation will only compute intermediate results used by different quality functions one time.

2.1 Obtaining Verdict

The Verdict repository now resides at Kitware, Inc. and is publicly available. A formal release has not yet been made since the repository has been moved and so Verdict source code must be obtained from Kitware's CVS server at www.vtk.org. If you intend to build VTK, you need not obtain or compile Verdict separately since it is included with VTK.

Verdict can be built and installed on most systems, including Linux, Mac OS X, and even Windows systems. For configuration, compilation, and installation details, please refer to [39].

2.2 Application Programming Interface (API)

Verdict was designed with a C interface so that it can be used in a variety of applications. It provides a number of quality functions, implemented as routines whose input consists of a list of coordinates corresponding to a list of nodes or vertices within a region and of the cardinality of this list. Note that this interface does not allow to pass the parametrizations mentioned in

[9] A corollary to this is that regions with different arrangements, such as inverted finite elements, may take on different values.

Definition 1, but the implementation may be expanded in this order, *e.g.*, to accommodate higher order elements. Each available quality function has a corresponding routine; for instance, the hexahedron “Condition Number” is:

```
double v_hex_condition(int num_nodes,
                       double node_coordinates[][3])
```

and it may be used as follows:

```
double coords[8][3];
...
double condition_value = v_hex_condition(8, coords);
```

If a region’s quality must be assessed with multiple functions, it is less computationally expensive to take advantage of common expressions: *e.g.*, the hexahedron “Jacobian” and “shape” quality functions both use the Jacobian matrix. Therefore, to improve computational efficiency, one function for each type of region computes multiple quality functions at the same time:

```
double v_hex_quality(int num_nodes,
                     double node_coordinates[][3],
                     unsigned int request_flag,
                     struct HexMetricVals* quality_vals)
```

e.g., to obtain together “Jacobian” and “shape” of a hexahedron:

```
double coords[8][3];
HexMetricVals vals;
double jacobian_value;
double shape_value;
int request = V_HEX_JACOBIAN | V_HEX_SHAPE;
...
v_hex_quality(8, coords, request, &vals);
double jacobian_value = vals.jacobian;
double shape_value = vals.shape;
```

2.3 Available Quality Functions

Verdict provides a total of 78 quality functions: triangles (13), quadrilaterals (23), tetrahedra (16), hexahedra (23), pyramids (1), wedges (1), and knives (1), which are now summarized in tabulated form. In principle, one should be able to reduce the 78 quality functions to a list of at most a dozen quality functions that correspond to basic geometric properties. However, the quality functions available in Verdict represent a collection of quality functions that have accumulated over the years by various practitioners at various times. Some quality functions may therefore be redundant.

A brief summary of the available qualities and their main properties is given in the Appendix; *cf.* Verdict’s Reference Manual [39] for the explicit definitions of these functions. The summary provides two tables for each type of region, with the exception of pyramids, wedges, and knives which are lumped into the same tables. The first table first indicates the function's dimension, in the sense of the associated units: L (resp. A) denotes dimensions of length (resp. angle). When a quality function has a repeated dimension unit, an exponent is used to show the count: *e.g.*, volume has dimension L^3. While the precise units of length depend on the input coordinates, angles are always reported in degrees. Note that filterable quality functions are dimensionless (denoted 1), but the converse is not true. Second, a bibliographic reference where the function is defined and discussed is provided, if available; otherwise, the formula is one that is traditionally used but not readily available in the literature. Last, the corresponding Verdict function name is indicated. For each type of region, the second table presents information on the acceptable, normal, and full ranges of values taken on. The acceptable ranges are subjective and are provided as a suggestion for people unfamiliar with the meaning of the value of a particular quality function. The ranges were selected based on visual appearance – poor-looking regions are considered to have unacceptable quality. The acceptable ranges should not be taken as authoritative and they should be used with care for quality verification of application meshes. The normal column describes the range of values taken on by all valid, non-degenerate regions. The full range column includes the values taken on by invalid and/or degenerate regions. These tables also contain an entry for the reference value that the quality takes on for the corresponding optimal element. For triangular, quadrilateral, tetrahedral, and hexahedral shapes, this is respectively an equilateral triangle, a square, a regular tetrahedron, and a cube. For filterable quality functions, this value is 1.

3 Applications

Geometric quality is important in simulations. Typically an initial quality must be met and be maintained as the simulation progresses. When the simulation domain is deformed during the course of a simulation, this can require mesh relaxation or re-meshing. When the quality is not maintained, simulations can diverge. This makes quality important during pre-processing (for achieving the initial quality desired), during simulation (for maintaining the quality), and during post-processing (for inspecting the results to verify the impact of geometric quality on the results). In this section, five applications that build on Verdict to evaluate mesh quality in all phases of mesh generation and analysis are used to illustrate the efficiency and versatility of this library at providing support for geometric quality assessment in a variety of contexts.

3.1 Pre-processing: VERDE

VERDE is a pre-processing tool that uses Verdict for verifying the quality of finite element models. Utilizing Verdict, VERDE provides a wide range of

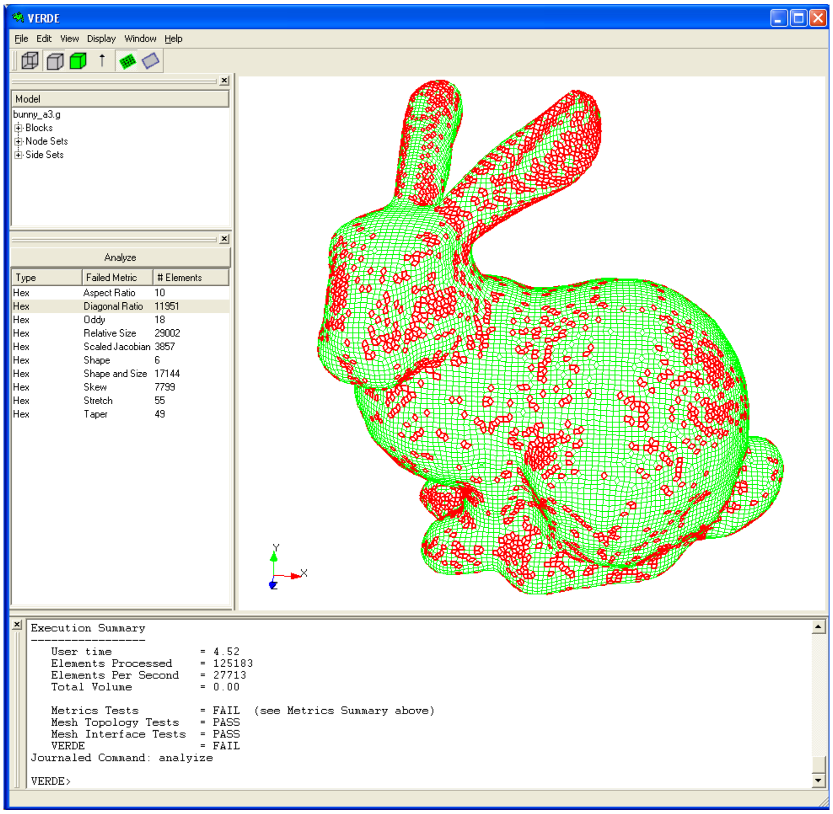

Fig. 1. Example filtering of mesh quality using VERDE and Verdict.

element qualities, including state of the art algebraic quality functions for calculating model topology, interface continuity, and locating mesh connectivity problems. It features a cross-platform graphical user interface and a graphical and numeric output useful for evaluating the quality of a finite element model, and was also designed to be a testbed for additional mesh quality research. Fig. 1 shows an example mesh loaded in the VERDE environment with highlighted elements filtered using mesh qualities calculated in Verdict.

3.2 Pre-processing: CUBIT

CUBIT[11] is a pre-processing tool for model and mesh generation that uses Verdict for verifying the mesh quality of finite element models. CUBIT is a full-featured software toolkit for geometric model generation and robust generation of 2-D and 3-D finite element meshes. The main development goal for CUBIT is to dramatically reduce the time required to generate meshes, particularly large hex meshes of complicated, interlocking assemblies. CUBIT incorporates a solid-model based preprocessor for generating meshes that conform

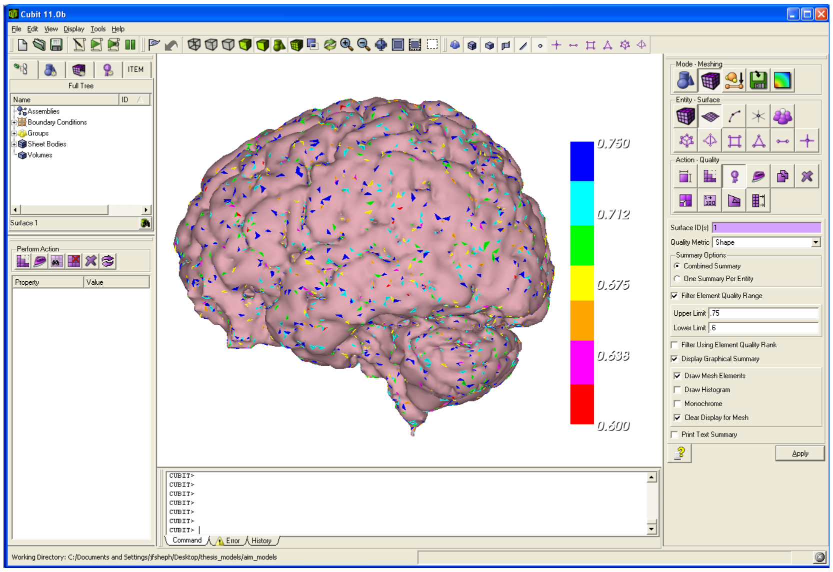

Fig. 2. Example filtering of mesh quality using CUBIT and Verdict. (The triangle mesh for the brain model is provided courtesy of INRIA by the AIM@SHAPE Shape Repository (`http://shapes.aim-at-shape.net/index.php`).)

to the geometric and topological features defined by the solid model. Mesh generation algorithms include quadrilateral and triangular paving [7], 2-D and 3-D mapping [9, 42, 41], hex sweeping [17, 35] and multi-sweeping [24, 23, 37], tetrahedral meshing [25], and various special purpose primitives. CUBIT also contains many algorithms for controlling and automating much of the meshing process, such as automatic scheme selection [43], interval matching [27], sweep grouping and sweep verification [37, 26], and includes state-of-the-art smoothing algorithms [8, 19, 20, 18, 21]. Fig. 2 shows a mesh loaded in the CUBIT environment with highlighted elements filtered using Verdict quality functions.

3.3 Post-processing: ParaView

ParaView[10] is a post-processing visualization tool that uses Verdict to provide mesh quality inspection. More specifically, the `vtkMeshQuality` class provides an interface to Verdict, whose functionalities are exposed as a filter in ParaView. `vtkMeshQuality` computes quality average, minimum, maximum, and variance over the entire mesh for each type of region, and stores these statistics in the output mesh's `FieldData`. In addition, the per-cell qualities are added to the mesh cell's data in an array named `Quality` to allow for

[10] `http://www.paraview.org/`

visualization and/or further processing as shown in Fig. 3. This figure demonstrates techniques we have found useful during post-processing. We are often interested in spatial trends in quality and their correlation with other variables such as simulation error estimates; a logarithmic scale with a user-adjustable color range helps to make these correlations along with volume rendering regions using a nearly transparent palette (resp. opaque) for good (resp. poor) quality values. This is especially true when the quality of interest is related to skew or torsion of elements. However, when quality functions meant to detect coincident vertices, slivers, or other disparities in scale are used, thresholding is more apropos since these elements are likely to take up very little screen space and thus require close-up inspection without nearby elements obscuring the rendering.

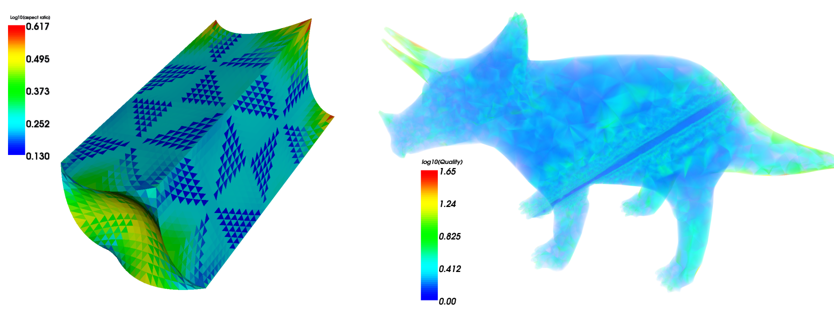

Fig. 3. Surface (left) and volume (right) renderings with ParaView of the per-region base-10 logarithms of the aspect ratios for two tetrahedral meshes.

3.4 End-to-end-processing: SCIRun

SCIRun [34] is a scientific programming environment which provides for the interactive construction, debugging and steering of large scale computer simulations. A visual programming interface allows scientific computations to be composed, executed, controlled, and tuned via a visual programming interface. Users may extend the set of scientific computations provided. In order to make the data flow paradigm applicable to large simulations, ways have been identified to avoid excessive memory use. Most importantly, SCIRun has been applied to solving large scale problems in computational biomedical imaging, [2, 4, 6] simulations [1] and visualization [3], explosives and fire simulations [31], and fusion reactions [5]. Verdict has been added to this environment to provide mesh quality verification and inspection. Fig. 4 shows an example mesh loaded in the SCIRun environment with highlighted regions filtered using mesh qualities calculated in Verdict.

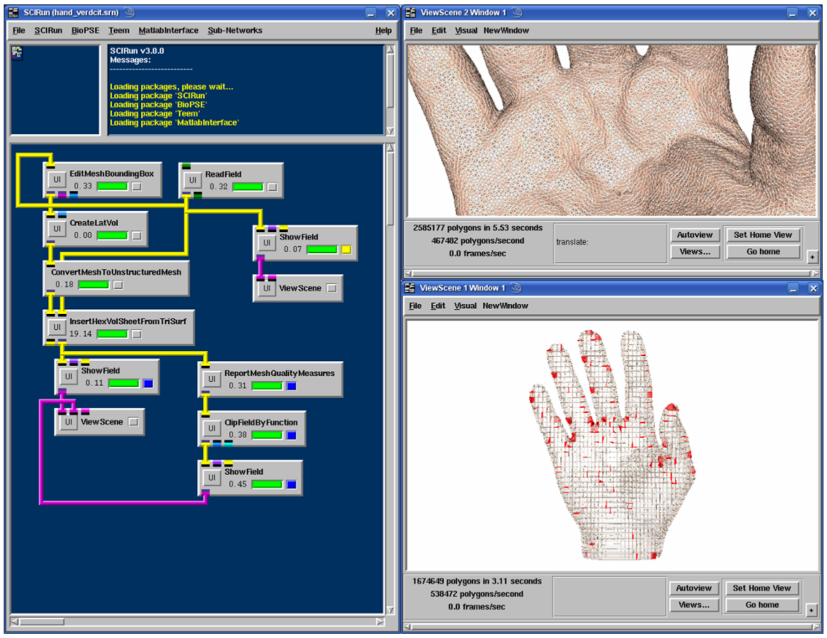

Fig. 4. Example filtering of mesh quality using SCIRun and Verdict. The triangle mesh for the hand model shown in the upper right frame is provided courtesy of INRIA by the AIM@SHAPE Shape Repository (http://shapes.aim-at-shap.net/index.php). The hexahedral mesh with mesh quality filtering shown in the lower right frame was generated by Jason Shepherd.

3.5 IA-FEMesh

IA-FEMesh[11] is a meshing tool designed to generate hexahedral finite element models appropriate for surface contact analyses in orthopedic applications. The mesh definitions initiate with segmented (manual or automated) medical image datasets, from which a surface representation of the region or regions of interest is generated. Two main meshing algorithms have been developed to generate a hexahedral mesh bounded by the triangulated isosurface representation. The first technique maps a block, or blocks, of elements onto the surface definition [38]. This algorithm has been used to mesh structures ranging from the relatively cylindrical phalanx bones of the hand to the complex geometries of the spinal vertebrae. Secondly, a mapped meshing technique has also been implemented. The objective is to map a predefined mesh of high quality (template) directly onto a new (subject-specific) bony surface definition, thereby yielding a similar mesh with minimal user intervention.

Regardless of the meshing technique, a check of the resulting mesh quality is imperative; consequently, a stand alone mesh quality viewer has been inte-

[11] `www.ccad.uiowa.edu/mimx`

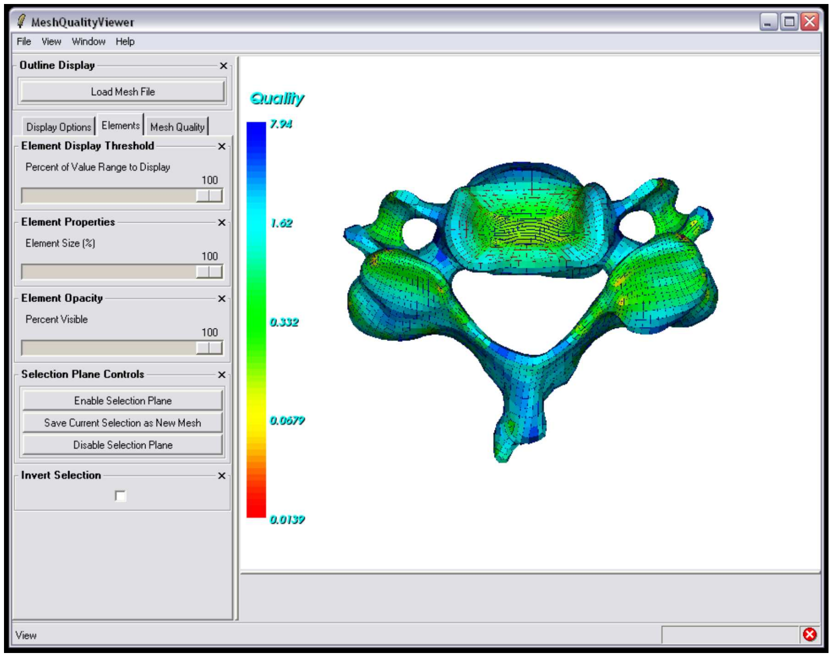

Fig. 5. Evaluation of the mesh quality for a cervical vertebra using IA-FEMesh and Verdict.

grated into the IA-FEMesh program suite. The objective is to make the mesh generation process more efficient by providing rapid, visual feedback to the user (Fig. 5). Our mesh quality application utilizes Verdict to analyze element quality according to several functions including: volume, Jacobian, scaled Jacobian, edge-ratio, and the Frobenius aspect. The tool supports the ability to display qualities and to interact with the mesh in real time using VTK. Since datasets generally contain a large number of elements, interactive tools have been developed that allow the operator to isolate and examine both individual elements and element groupings. A Laplacian smoothing algorithm has also been implemented to help improve quality in regions exhibiting elements of sub par quality. This work is currently being integrated into Slicer3, the core tool of the National Alliance for Medical Image Computing (NA-MIC).

4 Conclusions

This paper has outlined several applications that use Verdict for a variety of tasks in the context of simulations. We believe its adoption is evidence of an easy-to-use programmatic interface and an attention to speed and accuracy of calculation. Verdict is now distributed under a modified BSD license which makes it available to both open- and closed-source projects. We hope this will foster its use in other applications and encourage the contribution of new and useful quality functions.

In fact, in order to maintain Verdict as a reference library, we feel it is necessary to involve the meshing and simulation communities at large; we propose the formation of an open committee to adjudicate the definition of a standard set of quality functions with Verdict serving as a free reference implementation. It is hoped that Verdict will be accepted and supported by the community as a tool for standardizing the reporting of mesh quality and for allowing apples-to-apples comparisons of mesh quality between different mesh generation tools. Additionally, the use of Verdict will give credence and trust to mesh qualities reported in the research literature.

Another aspect of maintaining Verdict as a living code base is to push it into new application areas such as end-to-end processing, non-traditional quality functions, and further research into traditional quality functions. This paper has presented some of that work and we hope to continue into new areas in the future such as the development of quality functions for arbitrary polyhedra, such as those generated by mimetic finite difference schemes [22].

Acknowledgement

The authors would like to acknowledge Corey Ernst, Robert Kerr, Rob Leland, Ray Meyers, Karl Merkley, Scott Mitchell, Will Schroeder, Greg Sjaardema, Clinton Stimpson, Tim Tautges, and David White.

References

1. BioFEM: Bioelectric field finite element method software. Scientific Computing and Imaging Institute (SCI). URL `www.sci.utah.edu/cibc/software/index.html`.
2. BioImage: Volumetric image analysis and visualization. Scientific Computing and Imaging Institute (SCI). URL `www.sci.utah.edu/cibc/software/index.html`.
3. BioPSE: Bioelectric problem solving environment for modeling, simulation, image processing, and visualization of bioelectric fields. Scientific Computing and Imaging Institute (SCI). URL `www.sci.utah.edu/cibc/software/index.html`.
4. BioTensor: A SCIRun Power App for processing and visualizing diffusion tensor images. Scientific Computing and Imaging Institute (SCI). URL `www.sci.utah.edu/cibc/software/index.html`.
5. FusionView: Problem solving environment for MHD visualization. Scientific Computing and Imaging Institute (SCI). URL `software.sci.utah.edu/fusion.html`.
6. Seg3D: Volumetric image segmentation and visualization. Scientific Computing and Imaging Institute (SCI). URL `www.sci.utah.edu/cibc/software/index.html`.
7. Ted D. Blacker. Paving: A new approach to automated quadrilateral mesh generation. *Intl. J. for Numerical Methods in Engineering*, 32:811–847, 1991.

8. Michael Brewer, Lori Freitag-Diachin, Patrick Knupp, Thomas Leurent, and Darryl J. Melander. The MESQUITE mesh quality improvement toolkit. In *Proc. 12th Intl. Meshing Roundtable*, pages 239–250. Sandia National Laboratories, September 2003.
9. W. A. Cook and W. R. Oakes. Mapping methods for generating three-dimensional meshes. *Computers In Mechanical Engineering*, CIME Research Supplement:67–72, August 1982.
10. MSC Software Corporation. *Finite Element Modeling*, volume 3 of *MSC.PATRAN Reference Manual*. MSC.Software, Los Angeles, CA, 2003.
11. The CUBIT geometry and mesh generation toolkit, Sandia National Laboratories. URL `cubit.sandia.gov`.
12. V. N. Parthasarathy et al. A comparison of tetrahedron quality measures. *Finite Elem. Anal. Des.*, 15:255–261, 1993.
13. Inc. Fluent. *Fluent FIMESH User's Manual*. Fluent, Inc.
14. P.J. Frey and P.-L. George. *Mesh Generation*. Hermes Science Publishing, Oxford & Paris, 2000.
15. P. Knupp. Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. *Intl. J. Numer. Meth. Engng.*, 48: 1165–1185, 2000.
16. P. Knupp. Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elements in Analysis and Design*, 39:217–241, 2003.
17. Patrick Knupp. Next-generation sweep tool: A method for generating all-hex meshes on two-and-one-half dimensional geometries. In *Proc. 7th Intl. Meshing Roundtable*, pages 505–513. Sandia National Laboratories, October 1998.
18. Patrick Knupp and Scott A. Mitchell. Integration of mesh optimization with 3D all-hex mesh generation, LDRD subcase 3504340000, final report. SAND 99-2852, October 1999.
19. Patrick M. Knupp. Winslow smoothing on two-dimensional unstructured meshes. In *Proc. 7th Intl. Meshing Roundtable*, pages 449–457. Sandia National Laboratories, 1998.
20. Patrick M. Knupp. Hexahedral and tetrahedral mesh shape optimization. *Intl. J. for Numerical Methods in Engineering*, 58(1):319–332, 2003.
21. Patrick M. Knupp. Hexahedral mesh untangling and algebraic mesh quality metrics. In *Proc. 9th Intl. Meshing Roundtable*, pages 173–183. Sandia National Laboratories, October 2000.
22. Yuri Kuznetsov, Konstantin Lipnikov, and Mikhail Shashkov. Mimetic finite difference method on polygonal meshes. Technical Report LA-UR-03-7608, Los Alamos National Laboratories, 2003.
23. Mingwu Lai, Steven E. Benzley, Gregory D. Sjaardema, and Timothy J. Tautges. A multiple source and target sweeping method for generating all-hexahedral finite element meshes. In *Proc. 5th Intl. Meshing Roundtable*, pages 217–228. Sandia National Laboratories, October 1996.
24. Mingwu Lai, Steven E. Benzley, and David R. White. Automated hexahedral mesh generation by generalized multiple source to multiple target sweeping. *Intl. J. for Numerical Methods in Engineering*, 49(1):261–275, September 2000.
25. Mark Loriot. TetMesh-GHS3D v3.1 the fast, reliable, high quality tetrahedral mesh generator and optimiser. URL `www.simulog.fr/mesh/tetmesh3p1d-wp.pdf`.

26. Scott A. Mitchell. Choosing corners of rectangles for mapped meshing. In *13th Annual Symposium on Computational Geometry*, pages 87–93. ACM Press, June 1997.
27. Scott A. Mitchell. High fidelity interval assignment. In *Proc. 6th Intl. Meshing Roundtable*, pages 33–44. Sandia National Laboratories, October 1997.
28. A. Oddy, J. Goldak, M. McDill, and M. Bibby. A distortion metric for isoparametric finite elements. *Trans. CSME*, 38-CSME-32, 1988. Accession No. 2161.
29. P. P. Pébay. Planar quadrangle quality measures. *Engineering with Computers*, 20(2):157–173, 2004. URL `dx.doi.org/10.1007/s00366-004-0280-8`.
30. P. P. Pébay and T. J. Baker. Analysis of triangle quality measures. *AMS Mathematics of Computation*, 72(244):1817–1839, 2003. URL `dx.doi.org/10.1090/S0025-5718-03-01485-6`.
31. Combustion research group of University of Utah. C-Safe - Center for the Simulation of Accidental Fires and Explosions. URL `www.csafe.utah.edu/`.
32. J. Robinson. CRE method of element testing and the Jacobian shape parameters. *Eng. Comput.*, 4, 1987.
33. Larry A. Schoof and Victor R. Yarberry. EXODUS II: A finite element data model. Technical Report SAND92-2137, Sandia National Laboratories, 1992.
34. Scirun: A scientific computing problem solving environment. Scientific Computing and Imaging Institute (SCI). URL `software.sci.utah.edu/scirun.html, 2007.`.
35. Michael A. Scott, Matthew N. Earp, Steven E. Benzley, and Michael B. Stephenson. Adaptive sweeping techniques. In *Proc. 14th Intl. Meshing Roundtable*, pages 417–432. Sandia National Laboratories, September 2005.
36. SDRC. *SDRC/IDEAS Simulation: Finite Element Modeling – User's Guide*. SDRC.
37. Jason F. Shepherd, Scott A. Mitchell, Patrick Knupp, and David R. White. Methods for multisweep automation. In *Proc. 9th Intl. Meshing Roundtable*, pages 77–87. Sandia National Laboratories, October 2000.
38. K.H. Shivanna, B.D. Adams, V.A. Magnotta, and N.M. Grosland. Towards automating patient-specific finite element model development. In K. Miller and D. Poulikakos, editors, *Computational Biomechanics For Medicine 2006*, pages 85–93. Medical Image Computing and Computer Assisted Intervention Conference, 2006.
39. C. Simpson, C. D. Ernst, P. Knupp, P. P. Pébay, and D. C. Thompson. *The Verdict Library Reference Manual*. Sandia National Laboratories, April 2007. URL `www.vtk.org/Wiki/images/6/6b/VerdictManual-revA.pdf`.
40. L. M. Taylor and D. P. Flanagan. Pronto3D - a three dimensional transient solid dynamics program. Technical Report SAND87-1912, Sandia National Laboratories, 1989.
41. David R. White. *Automatic Quadrilateral and Hexahedral Meshing of Pseudo-Cartesian Geometries Using Virtual Subdivision*. Published Master's Thesis, Brigham Young University, June 1996.
42. David R. White, Mingwu Lai, Steven E. Benzley, and Gregory D. Sjaardema. Automated hexahedral mesh generation by virtual decomposition. In *Proc. 4th Intl. Meshing Roundtable*, pages 165–176. Sandia National Laboratories, October 1995.
43. David R. White and Timothy J. Tautges. Automatic scheme selection for toolkit hex meshing. *Intl. J. for Numerical Methods in Engineering*, 49(1):127–144, September 2000.

Appendix: Summary of **Verdict** Quality Functions

Quality name	**Dim.**	**Reference**	**Verdict function**
area	L^2	–	v_tri_area
aspect Frobenius	1	[30]	v_tri_aspect_frobenius
aspect ratio	1	[30]	v_tri_aspect_ratio
condition	1	[15, 16]	v_tri_condition
distortion	1	[36] (adapt.)	v_tri_distortion
edge ratio	1	[30]	v_tri_edge_ratio
maximum included angle	A^1	–	v_tri_maximum_angle
minimum included angle	A^1	[30]	v_tri_minimum_angle
radius ratio	1	[30]	v_tri_radius_ratio
relative size squared	1	[16]	v_tri_relative_size_squared
scaled Jacobian	1	[15]	v_tri_scaled_jacobian
shape and size	1	[16]	v_tri_shape_and_size
relative size squared	1	[16]	v_tri_shape

Table 1. Verdict triangle quality functions.

Quality name	**Dim.**	**Reference**	**Verdict function**
area	L^2	–	v_quad_area
aspect ratio	1	[29]	v_quad_aspect_ratio
condition	1	[15]	v_quad_condition
distortion	1	[36]	v_quad_distortion
edge ratio	1	[29]	v_quad_edge_ratio
Jacobian	L^2	[15]	v_quad_jacobian
maximum aspect frobenius	1	[29]	v_quad_max_aspect_frobenius
maximum included angle	A^1	–	v_quad_maximum_angle
maximum edge ratio	1	[32]	v_quad_max_edge_ratio
mean aspect frobenius	1	[29]	v_quad_med_aspect_frobenius
minimum included angle	A^1	–	v_quad_minimum_angle
Oddy	1	[28]	v_quad_oddy
radius ratio	1	[29]	v_quad_radius_ratio
relative size squared	1	[16]	v_quad_relative_size_squared
scaled Jacobian	1	[15]	v_quad_scaled_jacobian
shape and size	1	[16]	v_quad_shape_and_size
shape	1	[16]	v_quad_shape
shear and size	1	[16]	v_quad_shear_and_size
shear	1	[16]	v_quad_shear
skew	1	[32] (adap.)	v_quad_skew
stretch	1	[13]	v_quad_stretch
taper	1	[32] (adap.)	v_quad_taper
warpage	1	–	v_quad_warpage

Table 2. Verdict quadrilateral quality functions.

Quality name	**Dim.**	**Reference**	**Verdict function**
aspect β	1	[12]	`v_tet_aspect_beta`
aspect Frobenius	1	[15]	`v_tet_aspect_frobenius`
aspect γ	1	[12]	`v_tet_aspect_gamma`
aspect ratio	1	[14]	`v_tet_aspect_ratio`
collapse ratio	1	[10]	`v_tet_collapse_ratio`
condition	1	[15]	`v_tet_condition`
distortion	1	[36] (adap.)	`v_tet_distortion`
edge ratio	1	–	`v_tet_edge_ratio`
Jacobian	L^3	[16]	`v_tet_jacobian`
minimum dihedral angle	A^1	–	`v_tet_minimum_angle`
radius ratio	1	[12]	`v_tet_radius_ratio`
relative size squared	1	[16]	`v_tet_relative_size_squared`
scaled Jacobian	1	[15]	`v_tet_scaled_jacobian`
shape and size	1	[16]	`v_tet_shape_and_size`
shape	1	[16]	`v_tet_shape`
volume	L^3	[12]	`v_tet_volume`

Table 3. Verdict tetrahedron quality functions.

Quality name	**Dim.**	**Reference**	**Verdict function**
diagonal	1	–	`v_hex_diagonal`
dimension	L^1	[40] (adap.)	`v_hex_dimension`
distortion	L^3	[36] (adap.)	`v_hex_distortion`
edge ratio	1	–	`v_hex_edge_ratio`
Jacobian	L^3	[15]	`v_hex_jacobian`
maximum aspect frobenius	1	[15]	`v_hex_max_aspect_frobenius`
maximum edge ratio	1	[40] (adap.)	`v_hex_max_edge_ratio`
mean aspect frobenius	1	–	`v_hex_med_aspect_frobenius`
Oddy	1	[28] (adap.)	`v_hex_oddy`
relative size squared	1	[16]	`v_hex_relative_size_squared`
scaled Jacobian	1	[15]	`v_hex_scaled_jacobian`
shape and size	1	[16]	`v_hex_shape_and_size`
shape	1	[16]	`v_hex_shape`
shear and size	1	[16]	`v_hex_shear_and_size`
shear	1	[16]	`v_hex_shear`
skew	1	[40] (adap.)	`v_hex_skew`
stretch	1	[13] (adap.)	`v_hex_stretch`
taper	1	[40] (adap.)	`v_hex_taper`
volume	L^3	–	`v_hex_volume`

Table 4. Verdict hexahedron quality functions.

Quality name	**Dim.**	**Reference**	**Verdict function**
pyramid volume	L^3	–	v_pyramid_volume
wedge volume	L^3	–	v_wedge_volume
knife volume	L^3	–	v_knife_volume

Table 5. Other Verdict quality functions.

Quality name	Accpt.	Normal	Full	Ideal
area	[0, $D_{\max}$]	[0, $D_{\max}$]	[0, $D_{\max}$]	$\frac{\sqrt{3}}{4}$
aspect Frobenius	[1, 1.3]	[1, $D_{\max}$]	[1, $D_{\max}$]	1
aspect ratio	[1, 1.3]	[1, $D_{\max}$]	[1, $D_{\max}$]	1
condition	[1, 1.3]	[1, $D_{\max}$]	[1, $D_{\max}$]	1
distortion	[0.5, 1]	[0, 1]	[$-D_{\max}$, $D_{\max}$]	1
edge ratio	[1, 1.3]	[1, $D_{\max}$]	[1, $D_{\max}$]	1
maximum included angle	[60°, 90°]	[60°, 180°]	[0°, 180°]	60°
minimum included angle	[30°, 60°]	[0°, 60°]	[0°, 360°]	60°
radius ratio	[1, 3]	[1, $D_{\max}$]	[1, $D_{\max}$]	1
relative size squared	[0.25, 1]	[0, 1]	[0, 1]	N/A
scaled Jacobian	[0.2, $\frac{2\sqrt{3}}{3}$]	[$-\frac{2\sqrt{3}}{3}$, $\frac{2\sqrt{3}}{3}$]	[$-D_{\max}$, $D_{\max}$]	1
shape and size	[0.2, 1]	[0, 1]	[0, 1]	N/A
relative size squared	[0.25, 1]	[0, 1]	[0, 1]	1

Table 6. Properties of Verdict triangle quality functions.

Quality name	Accpt.	Normal	Full	Ideal
area	[0, $D_{\max}$]	[0, $D_{\max}$]	[$-D_{\max}$, $D_{\max}$]	1
aspect ratio	[1, 1.3]	[1, $D_{\max}$]	[1, $D_{\max}$]	1
condition	[1, 4]	[1, $D_{\max}$]	[1, $D_{\max}$]	1
distortion	[0.5, 1]	[0, 1]	[$-D_{\max}$, $D_{\max}$]	1
edge ratio	[1, 1.3]	[1, $D_{\max}$]	[1, $D_{\max}$]	1
Jacobian	[0, $D_{\max}$]	[0, $D_{\max}$]	[$-D_{\max}$, $D_{\max}$]	1
maximum aspect frobenius	[1, 1.3]	[1, $D_{\max}$]	[1, $D_{\max}$]	1
maximum included angle	[90°, 135°]	[90°, 360°]	[0°, 360°]	90°
maximum edge ratio	[1, 1.3]	[1, $D_{\max}$]	[1, $D_{\max}$]	1
mean aspect frobenius	[1, 1.3]	[1, $D_{\max}$]	[1, $D_{\max}$]	1
minimum included angle	[45°, 90°]	[0°, 90°]	[0°, 360°]	90°
Oddy	[0, $\frac{1}{2}$]	[0, $D_{\max}$]	[0, $D_{\max}$]	0
radius ratio	[1, 1.3]	[1, $D_{\max}$]	[1, $D_{\max}$]	1
relative size squared	[0.3, 1]	[0, 1]	[0, 1]	N/A
scaled Jacobian	[0.2, 1]	[−1, 1]	[−1, 1]	1
shape and size	[0.2, 1]	[0, 1]	[0, 1]	N/A
shape	[0.2, 1]	[0, 1]	[0, 1]	1
shear and size	[0.2, 1]	[0, 1]	[0, 1]	N/A
shear	[0.3, 1]	[0, 1]	[0, 1]	1
skew	[0.5, 1]	[0, 1]	[0, 1]	1
stretch	[0.25, 1]	[0, 1]	[0, $D_{\max}$]	1
taper	[0, 0.7]	[0, $D_{\max}$]	[0, $D_{\max}$]	0
warpage	[0, 0.7]	[0, 2]	[0, $D_{\max}$]	0

Table 7. Properties of Verdict quadrilateral quality functions.

Quality name	Accpt.	Normal	Full	Ideal
aspect β	$[1, 3]$	$[1, D_{\max}]$	$[1, D_{\max}]$	1
aspect Frobenius	$[1, 1.3]$	$[1, D_{\max}]$	$[1, D_{\max}]$	1
aspect γ	$[1, 3]$	$[1, D_{\max}]$	$[1, D_{\max}]$	1
aspect ratio	$[1, 3]$	$[1, D_{\max}]$	$[1, D_{\max}]$	1
collapse ratio	$[0.1, D_{\max}]$	$[0, D_{\max}]$	$[0, D_{\max}]$	$\frac{\sqrt{6}}{3}$
condition	$[1, 3]$	$[1, D_{\max}]$	$[1, D_{\max}]$	1
distortion	$[0.5, 1]$	$[0, 1]$	$[-D_{\max}, D_{\max}]$	0
edge ratio	$[1, 3]$	$[1, D_{\max}]$	$[1, D_{\max}]$	1
Jacobian	$[0, D_{\max}]$	$[0, D_{\max}]$	$[-D_{\max}, D_{\max}]$	$\frac{\sqrt{2}}{2}$
minimum dihedral angle	$[40^\circ, \alpha^*]$	$[0^\circ, \alpha^*]$	$[0^\circ, 360^\circ]$	α^*
radius ratio	$[1, 3]$	$[1, D_{\max}]$	$[1, D_{\max}]$	1
relative size squared	$[0.3, 1]$	$[0, 1]$	$[0, 1]$	N/A
scaled Jacobian	$[0.2, \frac{\sqrt{2}}{2}]$	$[-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}]$	$[-D_{\max}, D_{\max}]$	1
shape and size	$[0.2, 1]$	$[0, 1]$	$[0, 1]$	N/A
shape	$[0.3, 1]$	$[0, 1]$	$[0, 1]$	1
volume	$[0, D_{\max}]$	$[-D_{\max}, D_{\max}]$	$[-D_{\max}, D_{\max}]$	N/A

Table 8. Properties of Verdict tetrahedron quality functions ($\alpha^* = \frac{180^\circ}{\pi} \arccos \frac{1}{3} \approx 70.53^\circ$).

Quality name	Accpt.	Normal	Full	Ideal
diagonal	$[0.65, 1]$	$[0, 1]$	$[1, D_{\max}]$	1
dimension	$[1, D_{\max}]$	$[0, D_{\max}]$	$[0, D_{\max}]$	1
distortion	$[0.5, 1]$	$[0, 1]$	$[-D_{\max}, D_{\max}]$	1
edge ratio	$[1, 1.3]$	$[1, D_{\max}]$	$[1, D_{\max}]$	1
Jacobian	$[0, D_{\max}]$	$[0, D_{\max}]$	$[-D_{\max}, D_{\max}]$	1
maximum aspect frobenius	$[1, 3]$	$[1, D_{\max}]$	$[1, D_{\max}]$	1
maximum edge ratio	$[1, 1.3]$	$[1, D_{\max}]$	$[1, D_{\max}]$	1
mean aspect frobenius	$[1, 3]$	$[1, D_{\max}]$	$[1, D_{\max}]$	1
Oddy	$[0, \frac{1}{2}]$	$[0, D_{\max}]$	$[0, D_{\max}]$	0
relative size squared	$[0.5, 1]$	$[0, 1]$	$[0, 1]$	N/A
scaled Jacobian	$[0.2, 1]$	$[-1, 1]$	$[-1, D_{\max}]$	1
shape and size	$[0.2, 1]$	$[0, 1]$	$[0, 1]$	N/A
shape	$[0.3, 1]$	$[0, 1]$	$[0, 1]$	1
shear and size	$[0.2, 1]$	$[0, 1]$	$[0, 1]$	N/A
shear	$[0.3, 1]$	$[0, 1]$	$[0, 1]$	1
skew	$[0, \frac{1}{2}]$	$[0, 1]$	$[0, D_{\max}]$	0
stretch	$[0.25, 1]$	$[0, 1]$	$[0, D_{\max}]$	1
taper	$[0, \frac{1}{2}]$	$[0, D_{\max}]$	$[0, D_{\max}]$	0
volume	$[0, D_{\max}]$	$[0, D_{\max}]$	$[-D_{\max}, D_{\max}]$	N/A

Table 9. Properties of Verdict hexahedron quality functions.

Quality name	Accpt.	Normal	Full	Ideal
pyramid volume	$[0, D_{\max}]$	$[-D_{\max}, D_{\max}]$	$[-D_{\max}, D_{\max}]$	N/A
wedge volume	$[0, D_{\max}]$	$[-D_{\max}, D_{\max}]$	$[-D_{\max}, D_{\max}]$	N/A
knife volume	$[0, D_{\max}]$	$[-D_{\max}, D_{\max}]$	$[-D_{\max}, D_{\max}]$	N/A

Table 10. Properties of other Verdict quality functions.