Surface Mesh Qualities

Marco Attene*
IMATI-GE / CNR, Italy

Keywords: Shape representation, triangle mesh, benchmark, data set

Abstract: 3D surface models are often stored as indexed face sets, whereas algorithms can typically treat only a subset of the so-representable surfaces. Besides this gap between "potential" input and "allowed" input, more and more algorithms specify their input requirements at a high level of abstraction (e.g. "natural" shapes), and this makes it difficult to automatically assess the appropriateness of a specific model.
In this article, surface meshes are analyzed in terms of indexed face sets and are characterized based on a collection of "qualities". Qualities of a surface mesh represent the peculiarities of both the surface and its combinatorial representation, can be combined to formally define input requirements of algorithms, can be exploited to effectively estimate high-level categorizations of 3D objects in a collection, and can be used to quickly produce benchmarks for geometry processing algorithms starting from unstructured digital libraries.

1 Introduction

While developing new algorithms to treat polygon meshes, researchers make often several assumptions on the input. Unfortunately, checking the characteristics of a mesh may be complicated, it is often not the focus of the main research and, since it would require a significant additional effort, it is typically left unimplemented. Even worse, the application domain of an algorithm might be defined at a very high level of abstraction (e.g. segmentation of *man-made* objects) for which even a preliminary formalization becomes a hard task. Furthermore, new algorithms need to be tested using relevent benchmarks, and even if several repositories are available on the Internet, collecting a sufficient number of *appropriate* meshes might become a time-consuming task. To the best of our knowledge, no existing repository provides the means to resolve queries such as "find meshes having a uniform distribution of vertices and which unambiguously enclose a polyhedron". Thus, there is a need to characterize the input of geometric algorithms, which in turn requires to (1) define automatic procedures to check the appropriateness of an input mesh, and (2) enable the automatic creation of significant benchmarks starting from uncharacterized mesh collections.

We provide a classification of a number of shape characteristics and show how to use them to create useful benchmarks. We only deal with characteristics that can be formally defined in terms of elements of a representative indexed face set, and term them "surface mesh qualities". Even if accounting for all the possibile qualities is clearly unfeasible, some characteristics are in widespread use, thus we identify a set of "atomic" qualities that can be combined to derive what the community mostly needs. The definition and classification of these qualities is the main contribution of the paper. Also, we have developed algorithms to compute them and have shown how to use them to automatically produce benchmarks (see fig. 1).

*e-mail: attene@ge.imati.cnr.it

2 Background and Related Work

Quality criteria for meshes have been defined for rather specific contexts, such as finite element analysis (Shewchuk, 2001) or surface remeshing (Alliez et al., 2008). Some quality values can be considered to be actual "defects" (Attene et al., 2013) that prevent the mesh to be used in a specific application.

In AIM@SHAPE's Digital Shape Workbench (AIM@SHAPE, 2004) some metadata can be used to query the Shape Repository using a *semantic search engine*. Though this is one of the most advanced shape query methods we are aware of, it still has several limitations: on the one hand, the set of metadata employed is far from being exhaustive enough to enable a systematic production of benchmarks; on the other hand, most of the available metadata must be entered manually while uploading a model, while just a few values can be computed automatically using the *TriMeshInfo* tool (Borgo et al., 2005).

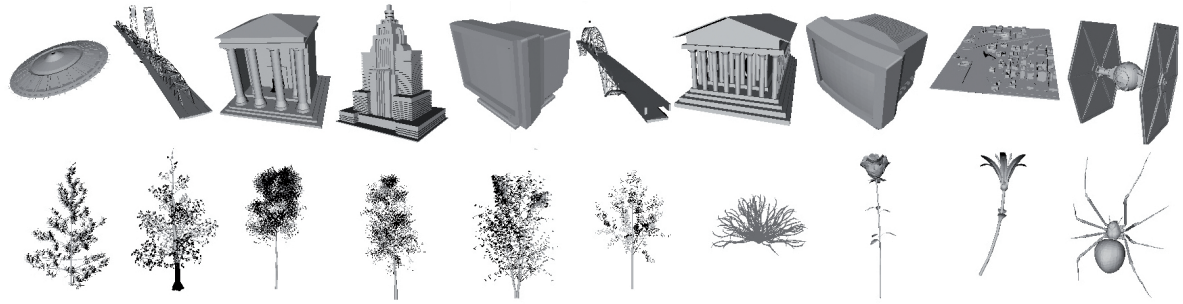

Figure 1: Top row: the "10 most man-made objects" automatically extracted from the Princeton Shape Benchmark based on a combination of qualities forming a ranking function. Bottom row: the "10 most natural objects" extracted by inverting the sign of the same ranking function.

Definitions In the remainder terms such as *abstract simplicial complex*, *simplex* and $face$ of a simplex are freely used. For formal definitions we point the reader to (Attene, 2010). We denote a triangle mesh as a pair $M = (P, \Sigma)$, where P is a set of N vertex positions $\mathbf{p}_i = (x_i, y_i, z_i) \in \mathbf{R}^3$ with $1 \leq i \leq N$, and Σ is an abstract simplicial complex over these vertices. Even without considering P at all, a number of $topological$ characteristics of M can be defined based on Σ only.

M is *homogeneous* (or *pure*) if all the edges and vertices in Σ are faces of at least one triangle in Σ. A simplex belongs to $Star(\sigma)$ if it is incident to σ. A simplex belongs to $Link(\sigma)$ if it is not incident to σ but it is a face of a simplex in $Star(\sigma)$. M is *combinatorially* manifold if Σ is a combinatorial manifold, which means that the *Link* of each of its vertices is made of a single ring of edges (Glaser, 1970). If the *Link* of all vertices is a single chain of edges (either open or closed to form a ring), then Σ is a *manifold with boundary*, and we say that M is *combinatorially manifold with boundary*. A combinatorially manifold mesh M is *orientable* if all its triangles can be oriented consistently (Attene, 2010). Based on the notion of *genus of a graph* (Goodman and Joseph, 2004), we say that the *combinatorial genus* of an orientable mesh $M = (P, \Sigma)$ is the minimal integer n such that Σ can be drawn without crossing itself on a sphere with n handles (i.e. Σ can be PL-embedded onto an oriented manifold of genus n).

The *geometric realization* $|M|$ of M (Attene, 2010) is a subset of $\mathbb{R}^3$ for which a Euclidean topology exists, and we say that M is *geometrically* manifold iff the neighborhood of each point in $|M|$ is homeomorphic to a disk in this topology. A triangle mesh may be manifold in the combinatorial sense and not in the Euclidean one, e.g. when the mesh self-intersects. Also, a geometrically manifold mesh may be not combinatorially manifold, e.g. when a topologically singular edge has three incident facets, but two of them coincide geometrically.

3 Surface mesh qualities

Qualities must be exactly defined and measureable without ambiguities. Thus, we must be able to say that in a given mesh M a quality Q has a unique value V. In other words, each quality is a function over the space of all the possible meshes. To express that, we will make use of predicates of the form $Q(M) = V$. The set of values that a quality Q can assume is called the *range* or *codomain* of Q. For example, for $Q = number\_of\_facets$ the range is the set of natural numbers $\mathbb{N}$, for $Q = minimum\_triangle\_angle$ the range is the real interval $[0, \pi]$, for $Q = is\_orientable$ the range is the set $\mathbb{B} = \{true, false\}$.

We deal with three classes of characteristics that we call *mesh qualities*, *surface qualities*, and *shape category indicators*. Mesh qualities quantify features of the mesh such as, for example, the number of vertices or the average length of edges. Surface qualities characterize the surface, and thus are independent of the specific represention used: the topological genus or the total area are examples of surface qualities. If the shape model is represented by a mesh, surface qualities may be derived from (a combination of) mesh qualities. Shape category indicators are also related to the shape, but they are independent of the model used and characterize objects in terms of high-level classifications such as, for example, models which are *articulated* or *architectural*. Normally, shape category indicators cannot be computed by simply combining mesh qualities or surface qualities. Nonetheless, in some cases it is possible to compute useful estimates (see section 3.3).

3.1 Mesh qualities

Most graphic formats encode surface meshes through *indexed face sets*, where a first block specifies the vertex positions and a second block represents polygons as sequences of indices. Clearly, files of this type

are not guaranteed to represent a well-defined polyhedron, while they may easily encode non-manifold and/or non-orientable sets of polygons. Thus, herewith we compute mesh qualities to provide useful information about the connectivity and geometry of these indexed mesh representations. Computing these qualities involves counting elements, measuring their attributes and performing statistical analysis.

Connectivity Besides counting the basic elements constituting the mesh (i.e. vertices, edges and facets), some applications need to ensure that their input is an actual simplicial complex, possibly after triangulation of non-simplicial facets. Thus, it is important to verify that all the indexed facets and edges are actually *valid* (i.e. facet indices must be different and stay within the limited range of vertices), and that no simplex is defined more than once. Also, it might be important to know if these *defects* are sparse or systematic within the file (e.g. to decide if it is worth to attempt a repairing). For these reasons, our list of mesh qualities includes the number of duplicated edges and facets and the number of invalid edges and facets. Note that a vertex should be considered a *duplication* only if its has the same coordinates of another vertex, thus it implies an analysis of the embedding. Conversely, for indexed edges and facets there are purely combinatorial possibilities which include: several instances of the same set of indexes; different orderings of the same set of indexes. In order to account for applications that can only deal with *homogeneous* (or pure) simplicial complexes, we also count isolated vertices and *naked* edges having no incident facets. Finally, we also take into account the large number of algorithms that assume that the input is a two-manifold. Here it is important to distinguish between two classes of algorithms: those that require the mesh to be combinatorially manifold (e.g. because their internal data structure can only treat these objects) and those that require geometrical manifoldness (e.g. because they need to unambiguously separate the embedding space into interior and exterior). To account for combinatorial manifoldness, our list of mesh qualities includes the number of singular vertices and the number of singular edges. We remind that a vertex is singular if its *Link* is not a single chain of edges, while an edge is singular if it has more than two indicent facets. Being a characteristic of the represented surface, geometrical manifoldness falls in the domain of surface qualities (see section 3.2). The following list summarizes mesh qualities characterizing the connectivity; the range is the set of natural numbers $\mathbb{N}$ in all the cases, and the symbol "#" means "Number Of": *#_duplicated_edges, #_duplicated_facets, #_invalid_edges, #_naked_edges, #_invalid_facets, #_isolated_vertices, #_singular_vertices, #_singular_edges, #_vertices, #_edges, #_facets.*

Geometry Bad embeddings may easily lead to meshes which are not appropriate for several applications. For this reason, our mesh qualities include the number of duplicated vertices, but also the number of duplications in the *realized* edges and facets which are induced by duplications of their vertices. Duplicated vertices may easily lead to degenerate facets with zero area, which are clearly an issue for several applications that need, for example, to compute surface normals. Degenerate facets, however, are not only due to coincident vertices, while they can be produced when the embedding aligns the vertices. In any case, it is important to count degenerate facets and edges. It is also important to count possible intersecting facets. Here it is worth to distinguish between the mesh quality *number_of_intersecting_factes* and the surface quality *is_self-intersecting*, even though the former implies the latter: when a surface model is self-intersecting, indeed, the number of intersecting facets may vary depending on the specific discretization used to represent the model. So, the *number* of intersecting facets is a *mesh* quality, whereas the fact that the surface *is* self-intersecting is a $surface$ quality. Knowing the number of such intersecting elements may help in deciding if it is worth to attempt a repairing to make the mesh exploitable. Besides counting things, depending on the specific embedding other interesting qualities assume varying values within continuous domains. To start with, we define a number of qualities that characterize the vertex sampling, which include uniformity, isotropy, and gradation. Indeed, several algorithms discussed in the literature are claimed to be robust against non-uniform sampling densities. In this regard, we observe that there is no standard definition for concepts such as *sampling uniformity* when applied to meshes. To avoid forcing the adoption of any specific definition, we tackle the problem indirectly by measuring things that can be exactly defined, but that can be also easily combined to represent the various possible definitions. In this case, for triangular meshes we consider the lengths of all the edges and measure their standard deviation and their average value. These two qualities can be later combined to represent the *sampling non-uniformity* as, for example, the ratio between the *edge length standard deviation* and the *average edge length*, which will be 0 for perfectly uniform samplings. If there are non triangular facets, they are first triangulated through a

scheme that minimizes the total edge length (Attene and Falcidieno, 2006). In some contexts (e.g. some remeshing algorithms), meshes are said to be uniform in relation to the uniformity of triangle areas, thus we proceed as in the case of edge lengths: we compute the standard deviation of these areas and encode it as a quality of the mesh. Note that the average triangle area does not need to be encoded explicitly because it can be derived as the ratio between the total surface area (see section 3.2) and the total number of facets, which are both encoded qualities. Among the algorithms that work well on non-uniformly sampled meshes, some might have problems in case of extreme sampling anisotropy (e.g. because they discard the available connectivity and use k-nearest neighbors to reproduce a local topology). Thus, we compute the sampling anisotropy around each vertex and encode the average, minimum and maximum values as mesh qualities. The sampling anisotropy around a vertex v is computed as the ratio between the 2^{nd} and 3^{rd} eigenvalues of the 3×3 coveriance matrix $C(v) = \sum_i (v_i - v)(v_i - v)^T$, where the v_is are all the vertices in $Link(v)$. To account for all the applications that expect mesh triangles to be *well-shaped* (e.g. FEM), we measure all the triangle angles and encode the minimum and maximum as mesh qualities. Similarly, we also compute all the dihedral angles formed by pairs of adjacent triangles and encode their minimum and maximum values. To make it possible to assess the presence of other extreme configurations such as e.g. *spikes*, for each vertex v we calculate the *excess angle* as the sum $K_v = 2\pi - \sum_i \alpha_i$, where the α_is are the incident triangle angles at v, and we encode both the maximum and minimum values as mesh qualities. The following list summarizes mesh qualities characterizing the geometry; the range is indicated in parentheses, and *SDev* means "Standard deviation": *#_duplicated_vertices (ℕ), #_multiply_realized_edges (ℕ), #_multiply_realized_facets (ℕ), #_degenerate_edges (ℕ), #_degenerate_facets (ℕ), #_intersecting_facets (ℕ), average_edge_length (ℝ), edge_length_SDev (ℝ), triangle_area_SDev (ℝ), min_sampling_anisotropy (ℝ), averaage_sampling_anisotropy (ℝ), max_sampling_anisotropy (ℝ), min_triangle_angle (ℝ), max_triangle_angle (ℝ), min_diherdal_angle (ℝ), max_dihedral_angle (ℝ), min_excess_angle (ℝ), max_excess_angle (ℝ).*

Topology Other interesting mesh qualities characterize the overall topology of the simplicial complex. Counting the number C of connected components is useful while analyzing meshes for applications that expect a connected input, but also to challenge tools that are claimed to work with multiple components. Furthermore, such a count helps in distinguishing between an actual *polygon soup* and a mesh which is intentionally made of several pieces. We also count the number B of connected components of boundary edges, which we shortly call *number of boundary components*. A mesh is combinatorially manifold if it is homogeneous (i.e. there are neither isolated vertices nor naked edges) and there are no singular elements; in this case each boundary component is actually a boundary loop, and their number B can be used to derive the combinatorial genus g of the mesh through the generalized Euler-Poincaré equality $V - E + F = 2(C - g) - B$, where V, E and F are the numbers of vertices, edges and facets respecitively. Orientation and orientability are also important qualities as they may help in deriving other characteristics (see section 3.2). The following list summarizes mesh qualities characterizing the overall topology: *#_connected_components (ℕ), #_boundary_components (ℕ), combinatorial_genus (ℕ), is_orientable (𝔹), is_oriented (𝔹), is_combinatorial_manifold (𝔹).*

3.2 Surface qualities

Some surface qualities can be derived based on mesh qualities and existing theorems, e.g. a mesh which is combinatorially manifold and orientable is also geometrically manifold if there are no self-intersections. Also, if such a mesh has no boundary then it encloses a polyhedron. This is often required by advanced applications that, e.g., need to tetrahedrize the inner shape. On the contrary, a non-orientable closed mesh cannot be realized without self-intersections even if it is combinatorially manifold (e.g. a Klein bottle) and, therefore, can be declared to be not geometrically manifold even without performing a direct analysis of the geometry. Similar strategies can be employed for other topological characteristics of the surface, e.g. to count the number of connected components of the realized mesh we start from the number of combinatorial components and look for intersections representing connections that have no counterpart in the abstract complex.

In several shape matching and similarity applications, algorithms use some forms of normalization to perform scale-invariant comparisons. For polyhedra, the total volume is mostly often used, but in other cases the total surface area or the radius of the smallest bounding sphere can be employed effectively. Thus, it is worth encoding these three qualities.

The analysis of the Gauss map provides interesting information about the represented surface. For a

polygon mesh M we can model a "discrete" Gauss map as a finite set of points on the unit sphere S^2 representing the facet normals. We loosely call this set of points the Gauss map of M, and indicate it with $N(M)$. To start with our analysis, we calculate the standard deviation of the sampling density of $N(M)$. To do this, we cannot rely on a useful connectivity within $N(M)$, thus we uniformly subdivide the sphere S^2 in k regions ($k = 128$ in our prototype), and just count how many points of $N(M)$ belong to each region. The standard deviation of the resulting histogram is an indicator of how the sampling density of $N(M)$ varies around the sphere, and can be exploited to derive higher-level characteristics of the shape (see section 3.3). To make this indicator less sensitive to the mesh sampling, instead of simply *counting* the number of points in each region, we use the areas of the facets that originated the points as weights in the sum.

Also, it is interesting to know if all the surface normals point towards a unique direction d, such as in the case of Digital Terrain Models or of digitized range images. Unfortunately such a direction d may exist while being unknown, thus we must employ a generalized approach that does not rely on such information. However, we observe that if such a d actually exists, then the Gauss map of the surface is limited within a hemisphere. Thus, as a solution, we just check whether the convex hull of $N(M)$ contains the origin: it it does, then M has at least two facets whose normal vectors point towards opposite directions (i.e. their dot-product is negative).

Finally, we compute a quality which measures how normal vectors change after having moved all the vertices to the center-of-mass of their neighbors. For each triangle t we calculate its normal $\mathbf{n_o}$, then we move its three vertices to the centers-of-mass of their neighbors, and calculate the so-modified triangle's normal $\mathbf{n_m}$. The average over all the triangles of the quantity $1 - < \mathbf{n_o}, \mathbf{n_m} >$ is what we call *average normal instability*, and can be exploited to estimate the amount of noise in a digitized mesh. The following list summarizes the discussed surface qualities: *total_surface_area* ($\mathbb{R}$), *enclosed_volume* ($\mathbb{R}$), *bounding_ball_radius* ($\mathbb{R}$), *is_geometric_manifold* ($\mathbb{B}$), *is_self_intersecting* ($\mathbb{B}$), *#_geometric_components* ($\mathbb{N}$), *Gauss_map_density_SDev* ($\mathbb{R}$), *is_origin_in_Gauss_map_CH* ($\mathbb{B}$), *average_normal_instability* ($\mathbb{R}$).

3.3 Shape category indicators

All the qualities described above are at a rather low level of abstraction, and indeed have an exact meaning and can be computed automatically. The definition of higher level characteristics (e.g. we want to test an algorithm on "man-made" objects) is much harder to define and compute exactly, but since we can compute a number of low-level qualities, we still have two possibilities to tackle the problem: (1) *Declarative approach*: the low-level qualities are combined to form predicates that, based on the experience of an expert, have high probability of representing the high-level characteristic. E.g. a high-resolution mesh with a high *Gauss_map_density_SDev* is likely to be a man-made object. (2) *Machine-learning approach*: a user appropriately tags a sufficiently large training set of meshes (e.g. as being man-made or not) and, for each of them, the values of a number of qualities are stacked to form a feature vector. These vectors are used to train a learning system (e.g. a Support Vector Machine) which will be eventually able to provide an estimate of the high-level classification for the query objects. Clearly, the accuracy of such an estimate depends on a number of factors, including the quality of the training set (number of meshes and discriminatory power) and the actual correlation between the qualities in the feature vector and the high-level characteristic.

4 Benchmark creation

When extracting a specific dataset (e.g. a benchmark) from a large repository we can collect meshes responding to combinations of query terms of two types: **Requirements** $Q1 = V1$ (or $Q1 > V1$, or $Q1 < V1$). All the meshes in the dataset *must* respond to these query terms. For example, $\#\_singular\_vertices = 0$ must be true when creating a set of manifold meshes. **Preferences** $Q1$ *As_Close_As_Possible_To* $V1$ (or $Q1$ *As_High_As_Possible*). If R is the set of all the meshes that satisfy the requirements, these query terms are used to actually rank elements in R. If the objective is creating a benchmark made of 100 manifold meshes preferably with a high-resolution vertex sampling, a ranking based on the $\#\_of\_vertices$ quality would make it possible to simply cut the sorted list of retrieved models to have the most appropriate benchmark.

5 Results and Discussion

We have implemented a tool that calculates all the possible mesh and surface qualities described, and have run experiments on three public repositories: a SHREC dataset (Biasotti and Attene, 2008), the Mesh

Segmentation Benchmark (Chen et al., 2009), and the Princeton Shape Benchmark (Shilane et al., 2004). The considered SHREC dataset was designed to challenge the robustness of shape matching algorithms against various forms of "defects"; it is made of 1500 meshes with variations in shape, smoothness, sampling density/uniformity, topology, etc. After having computed all the quality files (in about 30 minutes), we could successfully extract both the "10 most noisy meshes" and the "10 least noisy meshes" through the preference query term $average\_normal\_instability = As\_High\_As\_Possible$ and its inverse respectively. Similarly, we could extract the 10 "less uniformly sampled" meshes through the query term $edge\_length\_SDev/average\_edge\_length = As\_High\_As\_Possible$ and all the meshes that "enclose a polyhedron and that do not have *spikes*" (i.e. $is\_geometric\_manifold$ AND $\#\_boundary\_components = 0$, $-max\_excess\_angle = As\_High\_As\_Possible$).

The Mesh Segmentation Benchmark (MSB) collects meshes from other repositories, and its objective is to evaluate segmentation algorithms. All the 380 meshes in the MSB enclose valid polyhedra, and computing the quality files took 12 minutes. Here we could extract the "10 models having the highest genus". Note that, since we can assume that meshes enclose polyhedra, the usual notion of genus and that of combinatorial genus are equivalent, so we could simply use $combinatorial\_genus = As\_High\_As\_Possible$. Meshes in the *Princeton Shape Benchmark* (PSB) are collected from the Web, thus are rather heterogeneous from several points of view. Here we computed the quality files in about 16 minutes (note that meshes in the PSB are less complex than those in SHREC). We could successfully extract up to "10 range maps" using the requirement $is\_origin\_in\_Gauss\_map\_CH = false$, the "10 most man-made objects" and the "10 least man-made objects" (figure 1)through the preference term $Gauss\_map\_density\_SDev = As\_High\_As\_Possible$ and its inverse respectively.

See (Attene, 2012) for additional figures and details about the aforementioned experiments.

6 Conclusions

Surface mesh qualities make it possible to automatically analyze the input of an algorithm to assess its suitability and/or appropriateness. Atomic qualities can be combined to express derived properties, to estimate high-level classifications of a shape and to automatically produce benchmarks out of unclassified mesh collections. The importance of shape benchmarks is demostrated by the continuous birth of application-specific examples (e.g. (Chen et al., 2009) for segmentation, (Biasotti and Attene, 2008; Shilane et al., 2004) for matching and retrieval, (Wang et al., 2010) for watermarking).

Acknowledgements

This work is partly supported by the EU FP7 Project N. 26204 (VISIONAIR) and by the MIUR-PRIN Project N. 2009B3SAFK_002. Thanks are due to the SMG members at IMATI for helpful discussions.

REFERENCES

AIM@SHAPE (2004). Digital shape workbench, shape repository - http://shapes.aimatshape.net.

Alliez, P., Ucelli, G., Gotsman, C., and Attene, M. (2008). Recent advances in remeshing of surfaces. In De Floriani, L. and Spagnuolo, M., editors, *Shape Analysis and Structuring*, chapter 2, pages 53–82. Springer.

Attene, M. (2010). A lightweight approach to repairing digitized polygon meshes. *The Visual Computer*, pages 1393–1406.

Attene, M. (2012). Quality-based characterization of surface meshes. Technical Report 04, CNR-IMATI-GE.

Attene, M., Campen, M., and Kobbelt, L. (2013). Polygon mesh repairing: an application perspective. *ACM Computing Surveys (scheduled to appear)*, 45(2).

Attene, M. and Falcidieno, B. (2006). ReMESH: An interactive environment to edit and repair triangle meshes. In *Shape Modeling and Applications*, pages 271–276.

Biasotti, S. and Attene, M. (2008). Shrec'08: Stability of watertight models. In *SMI*, pages 217–218.

Borgo, R., Dellepiane, M., Cignoni, P., Papaleo, L., and Spagnuolo, M. (2005). Extracting meta-information from 3-dimensional shapes with protégé. In *Intnl. Protégé Conference*.

Chen, X., Golovinskiy, A., and Funkhouser, T. (2009). A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3).

Glaser, L. C. (1970). *Geometrical Combinatorial Topology*, volume 1. Van Nostrand Reinhold, New York.

Goodman, J. E. and Joseph, O., editors (2004). *Handbook of discrete and computational geometry (2nd edition)*. Chapman and Hall/CRC.

Shewchuk, J. R. (2001). Global optimization of mesh quality. In *Intnl. Meshing Roundtable (Tutorials)*.

Shilane, P., Min, P., Kazhdan, M., and Funkhouser, T. (2004). The princeton shape benchmark. In *Shape Modeling International, Genova, Italy*.

Wang, K., Lavoué, G., Denis, F., Baskurt, A., and He, X. (2010). A benchmark for 3D mesh watermarking. In *Proc. of SMI*, pages 231–235.