Graph-based 3D object classification

Sajjad Baloch, and Hamid Krim

ECE Department, North Carolina State University, Raleigh, NC, USA

ABSTRACT

In this paper, we propose a novel method for the classification of 3D shapes, based on topo-geometric shape descriptors. Topo-geometric models have an advantage over existing shape descriptors that they capture complete shape information — capturing topology through skeletal graphs, and geometry via edge weights. The resulting weighted graph representation allows shape classification by establishing error correcting subgraph isomorphisms between the test graph and model graphs, where the best match is the one that corresponds to largest subgraph isomorphism. We propose various cost assignments for graph edit operations for error correction, which in turn takes into account any shape variations arising due to noise and measurement errors.

Keywords: 3D Shape Modeling, Skeletal graph, Reeb graph, Morse theory

1. INTRODUCTION

Classification and recognition of 3D objects constitute an important problem in computer vision applications. 3D objects are usually represented by their surface boundaries, which in turn are normally parameterized by triangulated meshes. This representation is, however, unsuitable for shape recognition. The idea is then to extract some shape features, which form a characteristic of a shape, and subsequently employ them for recognition. A number of techniques have been proposed in literature. Shinagawa *et al.*[14] use *height function* based Reeb graphs for shape representation. Ben et al.[3] and Osada et al.[12] represent 3D objects by shape distributions and subsequently employ a dissimilarity measure on distributions for shape classification. Lazarus et al.[7] propose skeletonization based on geodesic distance from a manually chosen *source point.* The graphs obtained this way are called *level set diagrams.* Hilaga et al.[4] extend this approach by eliminating the need of the manual selection of a source point and propose a matching algorithm based on multiresolution Reeb graphs. Tung and Schmitt[15] capitalized on this approach to present augmented multiresolution Reeb graphs thereby capturing additional attribute features which yield better recognition rates. In Kazhdan et al.,[5] global properties of 3D objects are captured through the reflective symmetry descriptor that is defined over a certain parameterization, which the authors term as canonical. Kazhdan and Funkhouser[6] use rotation-invariant spherical harmonics as a shape descriptor for recognition purposes.

A limitation of these shape descriptors is that they do not represent a shape completely and, therefore, fall short of a unique shape signature. This non-uniqueness motivates us to utilize a better shape model for classification and recognition purposes. To that end, we propose to employ topo-geometric shape models (TGSM),[2] which take into account both topological as well as geometric information. These shape models have already been employed in reconstruction and representation applications.[2] In this paper, we exploit their uniqueness of representation, and invariance to rigid body transformation for shape classification.

The paper is organized as follows. We start with a review of topo-geometric shape model given in Section 1, along with a description of their construction, and their topology and geometry capturing capabilities. Shape classification using weighted graph matching is discussed in Section 3. We conclude the chapter with experimental results in Section 4.

Further author information: (Send correspondence to Sajjad Baloch)
Sajjad Baloch: E-mail: shbaloch@ncsu.edu,
Hamid Krim: E-mail: ahk@ncsu.edu

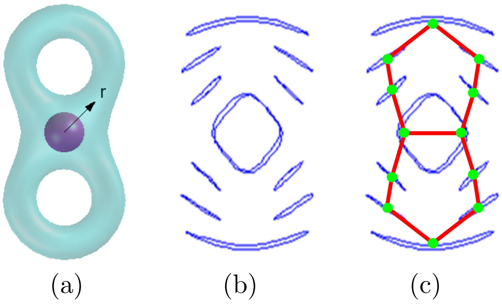


Figure 1. Skeletal graph of double torus: (a) Surface analyzed with an evolving sphere; (b) Intersections of the sphere and the surface; (c) Node assignment in the graph.

2. TOPO-GEOMETRIC SHAPE MODEL

As mentioned earlier, topo-geometric shape model captures both topology and geometry of 3D shapes. Topology is captured in a Morse theoretic framework,[8, 10] where a distance function is employed as a Morse function. This leads to a rigid transformation invariant topological graph of a given surface. Geometry on the other hand is captured by parametrically modeling the evolution of level curves of the distance function along topologically homogeneous parts of the surface. Curve evolution model parameters are then used to learn some weights, which are eventually assigned to various graph edges.

2.1. Topological Model

Consider a smooth compact 2D manifold $\mathcal{M}$ embedded in $\mathbb{R}^3$, and parameterized as $\phi : \Omega \to \mathcal{M}$, where $\Omega \ni u \mapsto \phi(u) \in \mathcal{M}$, with $\Omega \subset \mathbb{R}^2$ being an open connected set representing the parameter space. For capturing topology, TGSM capitalizes on Morse theory, which relates the topology of $\mathcal{M}$ with the number of critical points of a *Morse function* $f : \mathcal{M} \to \mathscr{R} \subseteq \mathbb{R}$. Specifically, TGSM employs distance function as a Morse function, which makes it invariant to rigid body transformations.

2.1.1. The distance function

Distance function defined on a surface $\mathcal{M}$ maps each point $\mathbf{p}$ on a surface $\mathcal{M}$ to its distance from the origin, i.e., $d : \mathbf{p} \mapsto \|\mathbf{p}\|, \quad \forall \mathbf{p} \in \mathcal{M}$. One can show that for generic surfaces $\mathcal{M} \subset \mathbb{R}^3$, the restriction of the distance function $d : \mathcal{M} \to \mathscr{R} \subseteq \mathbb{R}_+$ on $\mathcal{M}$ is a Morse function. We, therefore, use it for constructing skeletal graphs.

To analyze a compact surface with distance function as a Morse function, the surface is scanned with the level sets of the distance function by gradually increasing it in K steps from 0 to a sufficiently large value, say b. The integer K is, therefore, called the *resolution* of the skeletal graph. Since level sets of d are concentric spheres, intersections of the surface with spheres of radii r, for all $r \in [0, b]$, are evaluated and a node is assigned to each connected component in each intersection as illustrated in Fig. 1. The skeletal graph associated with the distance function may, hence, be described as a quotient space $\mathcal{M}/\sim$, where the equivalence relation $\sim$ is defined as follows:

Definition 2.1. (Equivalence) *Any two points $\mathbf{p}$ and $\mathbf{q} \in \mathcal{M}$ are equivalent, i.e., $\mathbf{p} \sim \mathbf{q}$, if they belong to the same connected component of a level set of the function d, i.e., $d(\mathbf{p}) = d(\mathbf{q})$ and $\mathbf{p} \in$* `ConnComp(LevelSet(q))`.

Distance function based topological graph is, therefore, a quotient space $\mathcal{M}/\sim := \{[\mathbf{p}] \mid \mathbf{p} \in \mathcal{M}\}$, where the equivalence class $[\mathbf{p}]$ of the point $\mathbf{p} \in \mathcal{M}$ is the set of all points $\mathbf{q} \in \mathcal{M}$ such that $\mathbf{q} \sim \mathbf{p}$.

Note that the function d given above is *not* invariant with respect to translation and scaling. In order to achieve this invariance, we take the origin at the centroid $\boldsymbol{\mu}$ of the surface of interest and scale the surface accordingly to get:

$$d_{\boldsymbol{\mu}}(\mathbf{p}) := \|\mathbf{p} - \boldsymbol{\mu}\|,$$
$$\tilde{d}_{\boldsymbol{\mu}}(\mathbf{p}) = \frac{d_{\boldsymbol{\mu}}(\mathbf{p}) - d_{\min}}{d_{\max} - d_{\min}}. \tag{1}$$

Proposition 2.2. (Invariance) *The distance function $\tilde{d}$ given by Eq. (1) is rotation, translation and scale invariant.*

Proof. The proof follows trivially from the definition of distance function. □

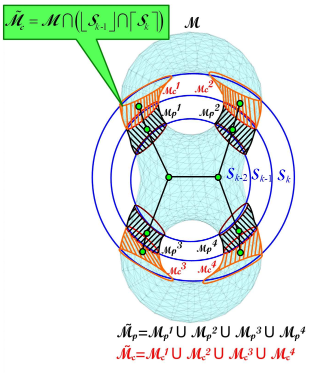

$\tilde{\mathcal{M}}_c = \mathcal{M} \cap (\lfloor \mathcal{S}_{k-1} \rfloor \cap \lceil \mathcal{S}_k \rceil)$
$\mathcal{M}$
$\mathcal{M}_c^1$
$\mathcal{M}_c^2$
$\mathcal{M}_p^1$
$\mathcal{M}_p^2$
$\mathcal{S}_{k-2}$
$\mathcal{S}_{k-1}$
$\mathcal{S}_k$
$\mathcal{M}_p^3$
$\mathcal{M}_p^4$
$\mathcal{M}_c^3$
$\mathcal{M}_c^4$
$\tilde{\mathcal{M}}_p = \mathcal{M}_p^1 \cup \mathcal{M}_p^2 \cup \mathcal{M}_p^3 \cup \mathcal{M}_p^4$
$\tilde{\mathcal{M}}_c = \mathcal{M}_c^1 \cup \mathcal{M}_c^2 \cup \mathcal{M}_c^3 \cup \mathcal{M}_c^4$

Figure 2. Skeletonization of a 3D surface $\mathcal{M}$.

2.1.2. Algorithm

The definition of distance function based topological graph given above leads to the algorithm for its construction, which is illustrated in Fig. 2 and given in Table 1.

Table 1. Algorithm for Constructing Topological Graph

- Find the centroid of the surface $\mathcal{M}$ as the arithmetic mean of the vertices of the triangulated mesh and place the origin at the centroid
- Find $d_{\max}$, the maximum distance from the centroid to $\mathcal{M}$
- Given K, define:

$$r_k := k\frac{d_{\max}}{K}, \quad k = 1, \ldots, K$$

- Generate the spheres $\mathcal{S}_1$ and $\mathcal{S}_2$ with radii $R = r_1$ and $R = r_2$, respectively
- Find $\tilde{\mathcal{M}}_p = \mathcal{M} \cap (\lfloor \mathcal{S}_1 \rfloor \cap \lceil \mathcal{S}_2 \rceil)$, where $\lceil . \rceil$ and $\lfloor . \rfloor$ identify the interior and exterior of a closed surface; $\tilde{\mathcal{M}}_p$ is, therefore, the part of $\mathcal{M}$ that lies between $\mathcal{S}_1$ and $\mathcal{S}_2$
- Assign a node $N_{\mathcal{M}_p}$ to each connected component $\mathcal{M}_p$ of $\tilde{\mathcal{M}}_p$ at the centroid of $\mathcal{M}_p$
- For $k = 3$ to K
  - Generate the "current" sphere $\mathcal{S}_k$ with radius $R = r_k$
  - Find $\tilde{\mathcal{M}}_c = \mathcal{M} \cap (\lfloor \mathcal{S}_{k-1} \rfloor \cap \lceil \mathcal{S}_k \rceil)$. Hence, $\tilde{\mathcal{M}}_c$ is the portion of $\mathcal{M}$ that lies in between $\mathcal{S}_{k-1}$ and $\mathcal{S}_k$
  - Find the connected components $\mathcal{M}_c$ of $\tilde{\mathcal{M}}_c$
  - For each $\mathcal{M}_c \in \tilde{\mathcal{M}}_c$ do
    - Assign a node $N_{\mathcal{M}_c}$ at the centroid of $\mathcal{M}_c$
    - Find the connected region $\mathcal{M}_p \in \tilde{\mathcal{M}}_p$ such that $\mathcal{M}_c \cup \mathcal{M}_p$ is a single connected region. Add an *edge segment* between $N_{\mathcal{M}_c}$ and $N_{\mathcal{M}_p}$
  - end for
  - $\tilde{\mathcal{M}}_p = \tilde{\mathcal{M}}_c$
- end for.

The algorithm yields a graph similar to the one shown in Fig. 3(a), which is composed of *edge segments* between various nodes. Not all of these nodes correspond to critical points. We, therefore, simplify topological graph by merging the nodes (and the edge segments), which lie on a topologically homogeneous path along the graph as shown in Fig. 3(b). Note that vertices in a simplified graph correspond to critical points of the distance function and mark a change in the topology of level curves. In subsequent discussion, we only consider the simplified graph.

2.2. Geometric Model

Although the skeletal model described in Section 2.1 completely represents topology, it contains minimal geometric information. In order to completely represent shape, we model the evolution of level curves along

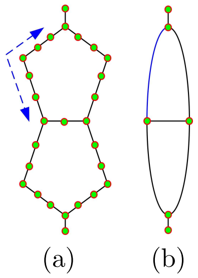


Figure 3. Simplification of a skeletal graph: (a) A graph learned by the algorithm of Section 2.1.2. ; (b) Simplified graph. Note how a topologically homogeneous sequence of edge segments marked in (a) maps to an edge in (b).

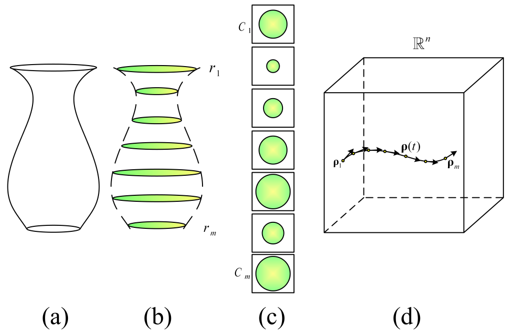


Figure 4. Illustration of geometric modeling: (a) A 3D object; (b) Object sampled at levels $r_1, \ldots, r_m$; (c) Intersections $C_1, \ldots, C_m$ embedded in Λ bounding box to compute the distance field; (d) Vectorizing the elements of the distance field yields an n dimensional vector $\boldsymbol{\rho}_i$ for each $\mathcal{C}_i$.

each graph edge to evaluate a weight vector, which is then assigned to the corresponding edge, thus capturing geometry. Note that these level curves have already been extracted by the graph construction algorithm given above.

These level curves are spatial curves, each of which is a subset of a sphere. Spherical coordinates, therefore, map these level curves onto the curves in $\Lambda = [-\pi, \pi] \times [-\frac{\pi}{2}, \frac{\pi}{2}]$. Idea is, then, to view each curve as a point in a high dimensional space, and fit a trajectory that passes through these points while maintaining curve topology and surface smoothness.

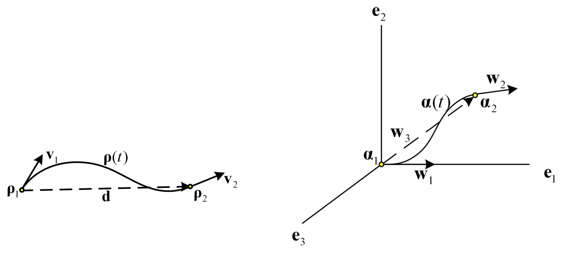
To preserve topology, we employ signed distance field, which is always bounded, since surfaces of interest are compact. *Signed distance field* $\rho_r : \Lambda \to \mathbb{R}$ for a closed curve $\mathcal{C}_r \in \Lambda$ corresponding to the r-level curve of the distance function, is defined as:

$$\rho_r(x, y) = \begin{cases} +D((x, y), \mathcal{C}_r) & \text{if} \quad (x, y) \in \lceil \mathcal{C}_r \rceil \\ -D((x, y), \mathcal{C}_r) & \text{if} \quad (x, y) \in \lfloor \mathcal{C}_r \rfloor, \end{cases} \tag{2}$$

where $D((x, y), \mathcal{C}_r)$ denotes the Euclidean distance from any point $(x, y) \in \Lambda$ to the set $\mathcal{C}_r$, and $\lfloor \mathcal{C}_r \rfloor$ and $\lceil \mathcal{C}_r \rceil$ represent the interior and exterior of $\mathcal{C}_r$, respectively. $\mathcal{C}_r$, itself, corresponds to the isoset $\rho_r^{-1}(0)$.

In practice, Λ is a discrete $n_1 \times n_2$ grid, and vectorizing the $n = n_1 n_2$ elements of the distance field defined on Λ yields a function $\boldsymbol{\rho} : \Lambda \to \mathbb{R}^n$ whose components are $(\rho_1, \ldots, \rho_n)$. This is illustrated in Fig. 4, where a vase is sampled by m horizontal planes at levels $r_1, \ldots, r_m$ to get intersection curves $\mathcal{C}_1, \ldots, \mathcal{C}_m$ shown in Fig. 4(b) which are then embedded in the bounding box Λ as in Fig. 4(c) to compute the distance field. Note that the choice of height function is only for illustration purposes, and in actual practice we use the distance function. The vectorization of the corresponding distance fields yields a collection of m points, $\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_m$ in $\mathbb{R}^n$ as depicted in Fig. 4(d). Our goal is to model a trajectory that best fits these points in $\mathbb{R}^n$ according to some criterion such that the original vase may be reconstructed from the level curves.

We adopt a piecewise interpolation approach, where for all i, we fit an arc $\boldsymbol{\rho}^{(i)}$ between the points $\boldsymbol{\rho}_i$ and



(a) (b)

Figure 5. Optimal trajectory between each pair of curve vector: (a) Elasticæ fitted between $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$ with constraints end point constraints $\mathbf{v}_1$ and $\mathbf{v}_2$; (b) Problem transformed to $\mathbb{R}^3$, where tangential constraints become $\mathbf{w}_1$ and $\mathbf{w}_2$ and the displacement between end points is mapped to $\mathbf{w}_3$.

$\boldsymbol{\rho}_{i+1}$ in $\mathbb{R}^n$ subject to the constraints

$$\begin{aligned} \boldsymbol{\rho}^{(i)}(r_i) &= \boldsymbol{\rho}_i, \\ \boldsymbol{\rho}^{(i)\prime}(r_i) &= \mathbf{v}_i, \\ \boldsymbol{\rho}^{(i)}(r_{i+1}) &= \boldsymbol{\rho}_{i+1}, \\ \boldsymbol{\rho}^{(i)\prime}(r_{i+1}) &= \mathbf{v}_{i+1}. \end{aligned} \tag{3}$$

where the end point tangent vectors $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$ are computed from the data, and where the arc $\boldsymbol{\rho}^{(i)}$ is parameterized by r. The purpose of introducing tangent constraints is to ensure smoothness at joints (end points), when later we combine arcs for all such consecutive pair of points. The optimal arc, $\boldsymbol{\rho}^{(i)}(r)$, is the one that minimizes a certain energy functional subject to the stated constraints. We, thus, construct a sequence of arcs between successive points which when put together yields a C^1 trajectory from $\boldsymbol{\rho}_1$ to $\boldsymbol{\rho}_m$, passing through $\boldsymbol{\rho}_2, \boldsymbol{\rho}_3, \ldots, \boldsymbol{\rho}_{m-1}$.

Finding the optimal arc in $\mathbb{R}^n$ with $n \gg 3$ is computationally very intensive. This complexity may, however, be greatly reduced by projecting the problem to $\mathbb{R}^3$ by a set of transformations, which are derived by noting that the arc belongs to an affine subspace of $\mathbb{R}^n$ through the point $\boldsymbol{\rho}_i$ and spanned by vectors $\mathbf{v}_i$, $\mathbf{v}_{i+1}$, and $\mathbf{d}_i := \boldsymbol{\rho}_{i+1} - \boldsymbol{\rho}_i$. These vectors are generically independent but not necessarily orthogonal. We, therefore, employ Gram–Schmidt orthogonalization to get an orthogonal set of basis vectors $\{\mathbf{b}_k, k = 1, 2, 3\}$ spanning the above mentioned subspace. This allows us to map the vectors $\mathbf{v}_i, \mathbf{v}_{i+1}, \mathbf{d}_i \in \mathbb{R}^n$ onto vectors:

$$\begin{aligned} \mathbf{w}_1 &= \mathbf{e}_1, \\ \mathbf{w}_2 &= \langle \mathbf{v}_{i+1}, \mathbf{b}_1 \rangle \mathbf{e}_1 + \langle \mathbf{v}_{i+1}, \mathbf{b}_2 \rangle \mathbf{e}_2 + \langle \mathbf{v}_{i+1}, \mathbf{b}_3 \rangle \mathbf{e}_3, \\ \mathbf{w}_3 &= \langle \mathbf{d}_i, \mathbf{b}_1 \rangle \mathbf{e}_1 + \langle \mathbf{d}_i, \mathbf{b}_2 \rangle \mathbf{e}_2 + \langle \mathbf{d}_i, \mathbf{b}_3 \rangle \mathbf{e}_3, \end{aligned} \tag{4}$$

where $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ form the canonical basis for $\mathbb{R}^3$ and $\mathbf{w}_k \in \mathbb{R}^3, k = 1, 2, 3$. The problem is now reduced to finding elasticæ $\boldsymbol{\alpha}^{(i)} : I = [r_i, r_{i+1}] \to \mathbb{R}^3$ satisfying $\boldsymbol{\alpha}^{(i)}(r_i) = \mathbf{0}$, $\boldsymbol{\alpha}^{(i)}(r_{i+1}) = \mathbf{w}_3$ with starting and ending tangents $\frac{\boldsymbol{\alpha}^{(i)\prime}(r_i)}{\|\boldsymbol{\alpha}^{(i)\prime}(r_i)\|} = \mathbf{w}_1$ and $\frac{\boldsymbol{\alpha}^{(i)\prime}(r_{i+1})}{\|\boldsymbol{\alpha}^{(i)\prime}(r_{i+1})\|} = \mathbf{w}_2$, respectively as shown in Fig. 5(b). The optimal solution to the problem is the one that minimizes the bending energy:

$$E(\boldsymbol{\alpha}^{(i)}) = (s_{i+1} - s_i) \int_{s_i}^{s_{i+1}} \kappa^2_{\boldsymbol{\alpha}^{(i)}}(s)\, ds, \tag{5}$$

where s is the arc length and $\kappa_{\boldsymbol{\alpha}^{(i)}}$ is the curvature. Since $ds = \|\boldsymbol{\alpha}^{(i)\prime}(r)\| dr$, we get:

$$E(\boldsymbol{\alpha}^{(i)}) = (r_{i+1} - r_i) \int_{r_i}^{r_{i+1}} \frac{\|\boldsymbol{\alpha}^{(i)\prime}(r) \times \boldsymbol{\alpha}^{(i)\prime\prime}(r)\|}{\|\boldsymbol{\alpha}^{(i)\prime}(r)\|^2}\, dr. \tag{6}$$

To reconstruct the part of the surface corresponding to the interval $[r_i, r_{i+1}]$, we need to traverse this trajectory. Hence, for any $r \in [r_i, r_{i+1}]$, $\boldsymbol{\alpha}^{(i)}(r) \in \mathbb{R}^3$ is mapped to a unique $\boldsymbol{\rho}^{(i)}(r) \in \mathbb{R}^n$:

$$\boldsymbol{\rho}^{(i)}(r) = \boldsymbol{\rho}_i + \langle \boldsymbol{\alpha}^{(i)}(r), \mathbf{e}_1 \rangle \mathbf{b}_1 + \langle \boldsymbol{\alpha}^{(i)}(r), \mathbf{e}_2 \rangle \mathbf{b}_2 + \langle \boldsymbol{\alpha}^{(i)}(r), \mathbf{e}_3 \rangle \mathbf{b}_3, \tag{7}$$

Each $\boldsymbol{\rho}^{(i)}(r)$, therefore, models the vectorized distance fields of the curves for $r \in [r_i, r_{i+1}]$. To recover a level curve from $\boldsymbol{\rho}^{(i)}(r)$, we first need to unvectorize it to get the corresponding distance field which is defined on Λ, and then to find the zero level set of this distance field. For a complete representation of an entire topologically homogenous part of a surface, we glue together the corresponding $\boldsymbol{\rho}^{(i)}(r)$ to get a trajectory $\boldsymbol{\rho}(r) \subset \mathbb{R}^n, r \in [r_1, r_m]$, which is C^1 smooth and in essence is a piecewise curve modeling approach. Each graph edge is finally assigned a distance field trajectory $\boldsymbol{\rho}(r) \subset \mathbb{R}^n$.

2.3. Topo-Geometric Model

We now encode the trajectory $\boldsymbol{\rho}(r)$ by a finite dimensional *weight vector*. Note that segments of $\boldsymbol{\rho}(r) \in \mathbb{R}^n$ have one-to-one mapping with corresponding $\boldsymbol{\alpha}^{(i)}(r) \in \mathbb{R}^3$. A smooth trajectory $\boldsymbol{\alpha}$ in $\mathbb{R}^3$ corresponding to $\boldsymbol{\rho}$ is obtained by gluing these $\boldsymbol{\alpha}^{(i)}$, while simultaneously translating and rotating them to have the end points aligned.[2] The coefficients of a polynomial approximation of this trajectory are then assigned as a weight vector to each homogenous part of the graph. A skeletal graph equipped with such weights contains sufficient information to reconstruct the original surface with desired precision. A weighted skeletal graph may, therefore, be used for storage and classification of objects.

3. SHAPE CLASSIFICATION

For shape classification, we employ weighted graph matching, where we proceed to find the best match of a test shape among model shapes using their skeletal representations. Note that an exact subgraph isomorphism between test graph and model graphs may not exist, due to shape distortions arising from noise or measurement errors. In order to account for such distortions, we distort model graphs by a sequence of edit operations, such that a subgraph isomorphism is found between the test graph and the distorted model graph. Each edit operation, however, has an associated cost and the optimal graph distortion is the one that minimizes cumulative cost over all sequences of edit operations. The best match is the one, which establishes largest subgraph isomorphism with minimal cost.

Edit operations that we are interested in are insertion and/or deletion of vertices and edges and modification of their weight attributes. One approach for cost assignment is to employ uniform cost for all edit operations. Such an approach, however, fails to take into account likelihoods of individual edit operations. Another approach is to assign variable cost to each edit operation. To that end, we derive our cost as a weighted sum of three cost components, C_c, C_p and C_g, taking into account both topological and geometric features of a surface:

$$C(\delta) = \alpha_c C_c(\delta) + \alpha_p C_p(\delta) + \alpha_g C_g(\delta), \tag{8}$$

where C is the cost of an edit operation δ, and α_c, α_p and α_g are the weights corresponding to three costs defined as follows. The first component, the *component cost* C_c, is defined as the fraction of edge segments in an edge and reflects its relative importance:

$$C_c(\delta) = \frac{N}{M}, \tag{9}$$

where N is the number of edge segments in the edge and M is the total number of edge segments in the graph. The second component, the *proximity cost* C_p, of an edit operation δ involving an edge between vertices v_i and v_j is defined as the proximity of the vertices, i.e., the length of the edge, weighted by maximum of their degrees:

$$C_p(\delta) = \max\{\mathscr{D}_i, \mathscr{D}_j\} d(v_i, v_j), \tag{10}$$

where $d(v_i, v_j)$ is the Euclidean distance between the physical locations of v_i and v_j, and $\mathscr{D}_i$ and $\mathscr{D}_j$ are the degrees of v_i and v_j respectively. The third component, the geometric cost C_g, depends on geometric attributes of edges and is defined as:

$$C_g(\delta) = d(\mathbf{w}_i, \mathbf{w}_j), \tag{11}$$

where $\mathbf{w}_i$ and $\mathbf{w}_j$ are geometric attribute vectors corresponding to e_i and e_j.

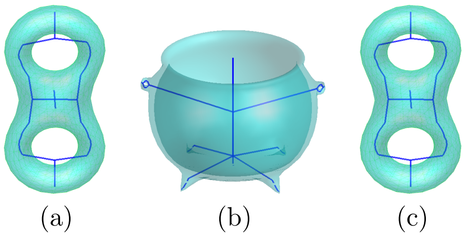


Figure 6. First level search for input shape (a) resulted in two candidate shapes (b) and (c) with identical genus.

4. EXPERIMENTAL RESULTS

The results demonstrating topology capturing and surface reconstruction capabilities of the model have been presented in Baloch et al.[2] In this paper, we focus our attention towards classification of shapes. We employ decomposition based error correcting subgraph isomorphism[9] for finding similarities between test and model shapes. Instead of using the variable cost, we employ a simplified matching criterion, which leads to a hierarchical model search. The idea is to exploit the simplicity of the constant cost together with important features of the variable cost, without enhancing complexity of the problem. This leads to the following hierarchical approach:

1. The first step is to simplify a graph by eliminating two types of edges, (a) those are composed of one edge segment even if the length of the edge is "too" large and (b) those with degree one vertices, and whose removal does not produce additional degree one vertices. This effectively takes into account low cost associated with the two types. The former compensates for component cost, while the latter compensates for proximity cost, by removing spurious edges possibly arising from shape distortions.

2. The second step involves carrying out a first level search in the database for the best match on the basis of genus of a test shape. This immediately prunes the set of shapes to those with topological type similar to that of the test shape.

3. The third step is related to the second level search, where the optimal isomorphisms from the model graphs to a test graph is found, assuming constant cost of an edit operation without considering edge attributes. This consequently gives a model which is the closest match to the input graph in the graph distance sense.

This strategy assumes zero geometric cost, corresponding to $\gamma = 0$ in Eq. (8), and considering all vertices of degree two and above to be equally important. In addition, Step 1 is based on infinite cost for edit operations that yield topological changes in a graph. In some cases, however, one may want to allow topological changes at a relatively high but finite penalty. In such a case, another term should be included in the total cost that takes topological changes into account.

For classification, test shapes are first represented by their simplified skeletal representation, before carrying out the first level hierarchical search, where all shapes in the database with same genus are found. In order to determine the genus, we adopt a depth-first-search based approach, thereby greatly reducing the number of models in some cases that would eventually be considered as candidates for graph matching. For instance, when searching a match for double torus, this strategy resulted in only two candidates as shown in Fig. 6. Since graph matching procedure is quite expensive, the above scheme, therefore, considerably speeded up search process.

Eventually, classification is carried out according to Step 3, with results presented in Fig. 7. Note that in one experiment a pear was classified as a vase. The reason is that we have not invoked geometric information and similar topological graphs for both shapes led to the ambiguity. This indicates that complete geometric information is essential for correct recognition.

ACKNOWLEDGMENTS

We acknowledge AFSOR for supporting this work by grant number F49620-98-1-0190.

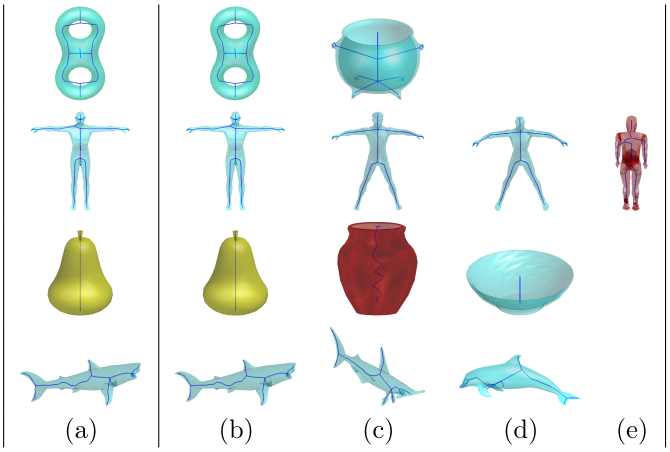


Figure 7. Shape recognition: (a) Test shape; (b) Best match; (c)-(e) Other good matches.

5. CONCLUSIONS

In this paper, we presented an application of topo-geometric shape model for shape classification. The ideas for shape classification via skeletal representations were borrowed from the rich field of graph theory, where we employed error-correcting graph matching. We, however, fell short of utilizing comprehensive skeletal models for recognition application, in spite of presenting a strategy that employs a variable cost function. Although it allowed classification, correct recognition demands utilizing complete geometric information, for which geometric component of variable cost seems to be of prime importance.

REFERENCES

1. S. H. Baloch, H. Krim, I. Kogan, and D. V. Zenkov, "Rotation invariant topology coding of 2D and 3D objects using Morse theory", *Proc. ICIP 2005.*
2. S. H. Baloch, H. Krim, I. Kogan, and D. Zenkov, "Topological-geometric shape model for object representation", *IEEE Trans. on Image Processing*, Under review.
3. A. B. Hamza, and H. Krim, "Geodesic object representation and recognition", *Proc. DGCI*, pp. 378–387, November 2003.
4. M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii, "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes", *Proc. SIGGRAPH*, pp. 203–212, August 2001.
5. M. Kazhdan, B. Chazelle, D. Dobkin, A. Finkelstein, and Thomas Funkhouser, "A Reflective Symmetry Descriptor", *European Conference on Computer Vision*, May 2002.
6. M. Kazhdan, and T. Funkhouser, "Harmonic 3D Shape Matching", *SIGGRAPH 2002 Technical Sketches*, pp. 191, July, 2002.
7. F. Lazarus, and A. Verroust, "Level set diagrams of polyheral objects", *Proc. Fifth ACM symposium on Solid Modeling and Applications*, pp. 130–140, June 1999.
8. Y. Matsumoto, *An Introduction to Morse Theory*, American Mathematical Society, 1997.
9. B. T. Messmer, and H. Bunke, "A New Algorithm for Error-Tolerant Subgraph Isomorphism Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 5, pp. 493-504, May 1998.
10. J. Milnor, *Morse Theory*, Princeton University Press, Princeton, NJ, 1963.
11. W. Mio, A. Srivastava, and E. Klassen, Interpolations with elasticæ in Euclidean spaces, *To appear in Quarterly of Applied Mathematics.*
12. R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape Distributions", *ACM Transactions on Graphics*, 21(4), pp. 807–832, October 2002.
13. G. Reeb, "Sur les points singuliers d'une forme de pfaff complèment intégrable ou d'une fonction numérique", *Comptes Rendus de L'Académie ses Séances* , Paris, 222, pp. 847–849, 1946.
14. Y. Shinagawa, and T. L. Kunii, "Constructing a Reeb graph automatically from cross sections", *IEEE Computer Graphics and Applications*, Vol. 11, No. 6, pp. 44-51, November 1991.
15. T. Tung, F. Schmitt, "Augmented Reeb Graphs for Content-Based Retrieval of 3D Mesh Models", *International Conference on Shape Modeling and Applications*, pp. 157-166, 2004.