MESH RETRIEVAL BY COMPONENTS

Ayellet Tal *
Department of Electrical Engineering
Technion
Email: ayellet@ee.technion.ac.il

Emanuel Zuckerberger
Department of Electrical Engineering
Technion

Keywords: 3D shape retrieval, 3D shape matching.

Abstract: This paper examines the application of the human vision theories of Marr and Biederman to the retrieval of three-dimensional objects. The key idea is to represent an object by an attributed graph that consists of the object's meaningful components as nodes, where each node is fit to a basic shape. A system that realizes this approach was built and tested on a database of about 400 objects and achieves promising results. It is shown that this representation of 3D objects is very compact. Moreover, it gives rise to a retrieval algorithm that is invariant to non-rigid transformations and does not require normalization.

1 INTRODUCTION

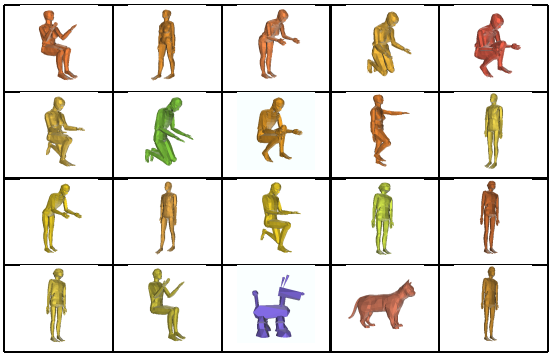


Figure 1: Retrieval of the top 20 objects similar to to the top left-most human figure

In his seminal work (Marr, 1982), Marr claims that the human brain constructs a 3D viewpoint-independent model of the image seen. This model consists of objects and spatial inter-relations between them. Every 3D object is segmented into primitives, which can be well approximated by a few simple shapes. Biederman's Recognition-By-Components (RBC) theory (Biederman, 1987; Biederman, 1988) claims that the human visual system tends to segment complex objects at regions of deep concavities into simple basic shapes, *geons*. The simple attributed shapes along with the relations between them form a stable 3D mental representation of an object.

The current paper proposes a retrieval approach that attempts to succeed these theories. The key idea is to decompose each object into its "meaningful" components at the object's deep concavities, and to match each component to a basic shape. After determining the relations between these components, an attributed graph that represents the decomposition is constructed and considered the object's *signature*. Given a database of signatures and one specific signature, the latter is compared to other signatures in the database, and the most similar objects are retrieved.

Retrieving 3D objects has become a lively topic of research in recent years (Veltkamp, 2001). A common practice is to represent each object by a few properties – a *signature* – and base the retrieval on the similarity of the signatures. Some signatures consist of local properties of the shapes such as histograms of colors and normals (Paquet et al., 2000), probability shape distributions (Osada et al., 2001), reflective symmetry (Kazhdan et al., 2003a), spherical harmonics (Vranic and Saupe, 2002; Kazhdan et al., 2003b) and more. Other papers consider global properties, such as shape moments (Elad et al., 2001) or *sphere projection* (Leifman et al., 2005). In these cases, the objects need to be normalized ahead of time.

*This work was partially supported by European FP6 NoE grant 506766 (AIM@SHAPE) and by the Smoler Research Funds.

Our global approach is mostly related to graph-based algorithms. The Reeb graph is a skeleton determined using a scalar function, which is chosen in this case to be the geodesic distance (Hilaga et al., 2001). In (Sundar et al., 2003; Cornea et al., 2005), it is proposed to represent an object by its skeleton and an algorithm for comparing shock graphs is presented. Our approach succeeds these methods, yet differs in several ways. First, the graphs are constructed differently, focusing on segmentation at deep concavities, following (Marr, 1982; Biederman, 1995). Second, each node and edge in the graph is associated with properties, resulting in an attributed graph. Third, a different graph matching procedure is utilized.

Our proposed signature has a few important properties. First, it is invariant to non-rigid-transformations. For instance, given a human object, we expect its signature to be similar to signatures of other humans, whether they bend, fold their legs or point forward, as illustrated in Figure 1. In this figure, all the 19 humans in a database consisting of 388 objects, were ranked among the top 21 objects, and 17 among the top 17. Invariance to non-rigid-transformations is hard to achieve when only geometry is considered.

Second, normalization is not required, since the signature is a graph that is invariant to rigid transformations.

Third, the signature tolerates degenerated meshes and noise. This is so because the object is represented by its general structure, ignoring small features.

Finally, the proposed signature is very compact. Thus, signatures can be easily stored and transfered.

The remaining of the paper is structured as follows. Section 2 outlines our approach. Sections 3–4 address the main issues involved in the construction of a signatures. In particular, Section 3 discusses mesh decomposition into meaningful components while Section 4 describes the determination of basic shapes. Section 5 presents our experimental results. Section 6 concludes the paper.

2 SYSTEM OVERVIEW

Given a database of meshes in a standard representation consisting of vertices and faces (e.g., VRML) and one specific object O, the goal is to retrieve from the database objects similar to O.

This section starts by outlining the signature computation technique. This is the main contribution of the paper and thus the next sections elaborate on the steps involved. Then, the section briefly describes the graph matching algorithm used during retrieval.

SIGNATURE COMPUTATION: Let S be an orientable mesh.

Definition 2.1 ***Decomposition:*** *$S_1, S_2, \ldots S_k$ is a decomposition of S iff (i) $\forall i, 1 \leq i \leq k$, $S_i \subseteq S$, (ii) $\forall i$, S_i is connected, (iii) $\forall i \neq j, 1 \leq i, j \leq k$, S_i and S_j are face-wise disjoint and (iv) $\cup_{i=1}^{k} S_i = S$.*

Definition 2.2 Decomposition graph: *Given a decomposition $S_1, S_2, \cdots S_k$ of a mesh S, a graph $G(V,E)$ is its corresponding decomposition graph iff each component S_i is represented by a node $v_i \in V$ and there is an arc between two nodes in the graph iff the two corresponding components share an edge in S.*

Definition 2.3 Attributed decomposition graph: *Given a decomposition graph $G(V,E)$, $G = (V,E,\mu,\nu)$ is an attributed decomposition graph if μ is a function which assigns attributes to nodes and ν is a function which assigns attributes to arcs.*

For each object in the database, its attributed decomposition graph, the object's *signature*, is computed and stored. Signature computation is done in three steps. First, the object is decomposed into a small number of meaningful components. Second, a decomposition graph is constructed. Third, each node and arc of the decomposition graph is given attributes, following (Biederman, 1987; Biederman, 1988). Specifically, each component is classified as a basic shape: a spherical surface, a cylindrical surface, a cone surface or a planar surface. The corresponding graph node is given the appropriate shape attribute. Each graph arc is attributed by the relative surface area of its endpoint components (i.e., greater, smaller, equal). We elaborate on signature construction in the next couple of sections.

RETRIEVAL: Given a specific object by the user, the goal of the system is to retrieve from the database the most similar objects to this object.

This step requires the comparison of graphs. Graph matching and subgraph isomorphism has been applied to many problems in computer vision and pattern recognition e.g., (Rocha and Pavlidis, 1994; Wang et al., 1997; Lee et al., 1990; Pearce et al., 1994; Wong, 1992). In the current paper, we follow (Messmer, 1995), which uses error-correcting subgraph isomorphism.

The key idea of error correction algorithm is as follows. A graph edit operation is defined for each possible error type. Possible operations are deletion, insertion and substitution (i.e., changing attributes) of nodes and arcs. A cost function is associated with each type of edit operation. Given a couple graphs, the algorithm aims at finding a sequence of edit operations with a minimal cost, such that applying the sequence to one graph results in a subgraph isomorphism with the other.

Formally, the algorithm is given two graphs, $G = (V,E,\mu,\nu)$ and $G' = (V',E',\mu',\nu')$, where V (V') is the set of nodes of G (G'), E (E') is its set of arcs,

μ (μ') is a function which assigns attributes to nodes and ν (ν') is a function which assigns attributes to arcs. It is also given a set of graph edit operations and their corresponding cost functions. The goal is to find the optimal error-correcting subgraph isomorphism (Δ, g), where Δ is a sequence of edit operations and g is an isomorphism, such that there is a subgraph isomorphism g from $\Delta(G)$ to G' and the cost $C(\Delta)$ of Δ is minimal.

The algorithm maintains a search tree. The root of the search tree contains an empty mapping and is associated with cost 0. At the next level of the search tree, the first node of G is mapped onto nodes in G'. Each such mapping, along with its corresponding cost of the relevant edit operation, is a node in the search tree. The generation of the next nodes is guided by the cost of the edit operations. The node representing the mapping with the lowest cost in the current search tree is explored by mapping a new node of G onto every node of G' that has not yet been used in the path and the corresponding costs are calculated.

When the first mapping γ' describing a complete subgraph isomorphism from G to G' is found, a threshold parameter is set to the cost $C(\gamma')$ of γ'. A node having a cost greater than the threshold is never explored. Other nodes are explored until a mapping with the minimal cost is found.

This procedure is applied to the graph representing the query object against each graph in the database. It returns a corresponding error value for each pair. The lower the error, the less edit operations are required (or the "cheaper" these operations are), and thus the more similar the objects are. The objects are therefore retrieved in an ascending order of their error values.

3 MESH DECOMPOSITION

The first step in signature construction is mesh decomposition into its meaningful components. In recent years, there have been several papers addressing this problem, e.g. (Katz and Tal, 2003; Li et al., 2001; Lee et al., 2005; Shamir, 2004; Katz et al., 2005). These techniques produce very nice decompositions. However, we will show below that simpler, linear algorithms are sufficient for retrieval.

Our approach follows Biederman's observation that "*the human visual system tends to segment complex objects at regions of deep concavities into simple basic shapes*". Thus, algorithms that generate rough decompositions at deep concavities are used.

In (Chazelle et al., 1997), a sub-mesh is called *convex* if it lies entirely on the boundary of its convex hull. It is proved that the optimization problem is NP-complete. Nevertheless, linear greedy flooding heuristics are used for generating convex decompositions. These heuristics work on the dual graph H of mesh S, where nodes represents facets and arcs join nodes associated with adjacent facets. The algorithm starts from some node in H and traverses H, collecting nodes along the way as long as the associated facets form a convex sub-mesh. When no adjacent nodes can be added to the current component, a new component is started and the traversal resumes.

Another simple linear decomposition algorithm is *Watershed decomposition* (Mangan and Whitaker, 1999) which decomposes a mesh into *catchment basins*, or *watersheds*. Let $h : E \rightarrow R$ be a discrete height function defined over E, the set of elements (vertices, edges or faces) of the mesh. A *watershed* is a subset of E, consisting of elements whose path of steepest descent terminates in the same local minimum of h. In our implementation, the height function is defined over the edges and is a function of the dihedral angle.

The key idea of the Watershed decomposition algorithm is to let the elements descend until a labeled region is encountered, where all the minima are labeled as a first step.

The major problem with watershed as well as with convex decomposition is over-segmentation (i.e., obtaining a large number of components), due to many small concavities. The goal of our application, however, is to obtain only a handful of components.

To solve over-segmentation, it is proposed in (Mangan and Whitaker, 1999) to merge regions whose watershed depth is below a certain threshold. A couple of other possible solutions are studied in (Zuckerberger et al., 2002) and described below.

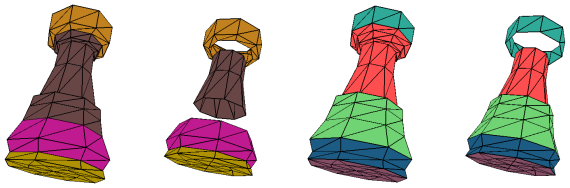
First, since small components are less vital to recognition (Biederman, 1987), the components are merged based on their surface areas. Thus, a small component is merged with a neighboring component having the largest surface area. This process is done in ascending order of surface areas and continues until all the components become sufficiently large.

The drawback of merging is that it might result with complex shapes, which might not fit any basic shape.

Another solution is to ignore the small components altogether. Only the original large components are taken into account both in the construction of the decomposition graph and in determining the components' basic shapes. The small components are used only to determine the adjacency relations between the large components.

Figure 2 presents an example of the results, obtained by four variants of the general scheme: Convex vs. Watershed decomposition and merging vs. ignoring small components. As can be seen, even when the small components are ignored, there is still sufficient information to visually recognize the rook. Figures 2(c) demonstrates the drawback of merging – the

red component does not resemble any basic shape.



Convex merging | Convex ignoring | Watershed merging | Watershed ignoring

Figure 2: Decompositions of a rook

In summary, the first step in constructing a signature of an object is to decompose it into a handful of meaningful components. This can be done by augmenting linear algorithms – the *watershed decomposition* and a *greedy convex decomposition* – with a post-processing step which either eliminates small components or merges them with their neighbors.

4 BASIC SHAPE DETERMINATION

The second issue in the construction of a signature is basic shape determination. Given a sub-mesh, which basic shape better fits this component? In this paper four basic shapes are considered – a spherical surface, a cylindrical surface, a cone surface and a planar surface.

Our problem is related to the problem of fitting implicit polynomials to data and using polynomial invariants to recognize three-dimensional objects. In (Taubin, 1991), a method based on minimizing the mean square distance of the data points to the surface is described. A first-order approximation of the real distance is used. In (Keren et al., 1994), a fourth-degree polynomial $f(x,y,z)$ is sought, such that the zero set of $f(x,y,z)$ is stably bounded and approximates the object's boundary. A probabilistic framework with an asymptotic Bayesian approximation is used in (Subrahmonia et al., 1996).

In order to fit a basic shape to a component, the given component is first sampled. A non-linear least-squares optimization problem, which fits each basic shape to the set of sample points, is then solved. The approximate mean square distance from the sample points to each of the basic surfaces is minimized with respect to a few parameters specific for each basic shape. The basic shape with the minimal fitting error represents the shape attribute of the component. The algorithm for fitting the points to a surface is based on (Taubin, 1991). We formalize it below.

Let $f : R^n \rightarrow R^k$ be a smooth map, having continuous first and second derivatives at every point. The set of zeros of f, $Z(f) = \{Y | f(Y) = 0\}$, $Y \in R^n$ is defined by the implicit equations $f_1(Y) = 0, \cdots, f_k(Y) = 0$ where $f_i(Y)$ is the i-th element of f, $1 \leq i \leq k$.

The goal is to find the approximate distance from a point $X \in R^n$ to the set of zeros $Z(f)$ of f. In the linear case, the Jacobian matrix $Jf(X)$ of f with respect to X is a constant $Jf(X) = \mathbf{C}$, and $f(Y) = f(X) + \mathbf{C}(Y - X)$. The unique point $\hat{Y}$ that minimizes the distance $\|Y - X\|$, constrained by $f(Y) = 0$, is given by $\hat{Y} = X - \mathbf{C}^{\dagger} f(X)$, where $\mathbf{C}^{\dagger} = \mathbf{C^T}(\mathbf{CC^T})^{-1}$ is the pseudo-inverse (Duda et al., 2000). If C is invertible then $\mathbf{C}^{\dagger} = \mathbf{C^{-1}}$. Finally, the square of the distance from X to $Z(f)$ is given by

$$dist(X, Z(f))^2 = \|\hat{Y} - X\|^2 = f(X)^T (\mathbf{CC}^T)^{-1} f(X).$$

For the nonlinear case, Taubin (Taubin, 1991) proposes to approximate the distance from X to $Z(f)$ with the distance from X to the set of zeros of a linear model of f at X, $\tilde{f} : R^n \rightarrow R^k$, where $\tilde{f}$ is defined by the truncated Taylor series expansion of f, $\tilde{f}(Y) = f(X) + Jf(X)(Y - X)$. But, $\tilde{f}(X) = f(X)$, $J\tilde{f}(X) = Jf(X)$, and the square of the approximated distance from a point $X \in R^n$ to the set of zeros $Z(f)$ of f is given by

$$dist(X, Z(f))^2 \approx f(X)^T (Jf(X) Jf(X)^T)^{-1} f(X).$$

Specifically, for the basic shapes we are considering, $n = 3$, $k = 1$, and the set of zeros $Z(f)$ of f is a surface in three-dimensions. The Jacobian $Jf(X)$ has only one row and $Jf(X) = (\nabla f(X))^T$, where $\nabla f(X)$ is the gradient of $f(X)$.

In this case, the approximated distance becomes

$$dist(X, Z(f))^2 \approx f(X)^2 / \|\nabla f(X)\|^2.$$

Moreover, we are interested in maps described by a finite number of parameters $(\alpha_1, \cdots, \alpha_r)$. Let $\phi : R^{n+r} \rightarrow R^k$ be a smooth function, and consider maps $f : R^n \rightarrow R^k$, which can be written as $f(X) \equiv \phi(\alpha, X)$, where $\alpha = (\alpha_1, \cdots, \alpha_r)^T$, $X = (X_1, , \cdots, X_n)$ and $\alpha_1, \cdots, \alpha_r$ are the parameters.

The approximated distance from X to $Z(\phi(\alpha, X))$ is then

$$dist(X, Z(\phi(\alpha, X)))^2 = \delta_{\phi}(\alpha, X)^2$$
$$\approx \phi(\alpha, X)^T (J\phi(\alpha, X) J\phi(\alpha, X)^T)^{-1} \phi(\alpha, X).$$

In particular, in three-dimensional space

$$\delta_{\phi}(\alpha, X)^2 \approx \phi(\alpha, X)^2 / \|\nabla \phi(\alpha, X)\|^2.$$

We can now formalize the fitting problem. Let $P = \{p_1, \cdots, p_m\}$ be a set of n-dimensional data points and $Z(\phi(\alpha, X))$ the set of zeros of the smooth function $\phi(\alpha, X)$. In order to fit P to $Z(\phi(\alpha, X))$ we need to minimize the approximated mean square distance $\Delta_P^2(\alpha)$ from P to $Z(\phi(\alpha, X))$:

$$\Delta_P^2(\alpha) = \frac{1}{m} \sum_{i=1}^{m} \delta_{\phi}(\alpha, p_i)^2$$

with respect to the unknown parameters $\alpha = (\alpha_1, \cdots, \alpha_r)^T$.

This is equivalent to minimizing the length of the residual vector $Q = (Q_1, \cdots, Q_m)^T$

$$\|Q(\alpha)\|^2 = \sum_{i=1}^{m} Q_i(\alpha)^2 = m\Delta_P^2(\alpha)$$

where $Q_i(\alpha) = \delta_\phi(\alpha, p_i)$, $i = 1, \cdots, m$.

The Levenberg-Marquardt algorithm can be used to solve this nonlinear least squares problem (Bates and Watts, 1988). This algorithm iterates the following step

$$\alpha^{n+1} = \alpha^n - (JQ(\alpha^n)JQ(\alpha^n)^T + \mu_n I_m)^{-1} JQ(\alpha^n)^T Q(\alpha^n),$$

where $JQ(\alpha)$ is the Jacobian of Q with respect to α: $J_{ij}Q(\alpha) = \frac{\partial Q_i}{\partial \alpha_j}(\alpha)$, for $i = 1, \cdots, m$, and $j = 1, \cdots, r$, and μ_n is a small nonnegative constant which makes the matrix $JQ(\alpha^n)JQ(\alpha^n)^T + \mu_n I_m$ positive defined.

At each iteration, the algorithm reduces the length of the residual vector, converging to a local minimum.

4.1 Distance 3D Point – Basic Shape

We can now explicitly define the square of the distance $\delta_\phi(\alpha, X)$ from a three-dimensional point X to the set of zeros $Z(\phi(\alpha, X))$ of $\phi(\alpha, X)$ for our basic shapes, three of which are quadrics (i.e., sphere, cylinder, cone) and the fourth is linear (i.e., plane).

A quadric, in homogeneous coordinates, is given by $X^T MX = 0$ in the global coordinate system, where M is a 4×4 matrix and X is a vector in R^4. In its local coordinate system, it is given by $X'^T M' X' = 0$, where $X = T_r R_x R_y R_z S_c X'$, T_r is a translation matrix, R_x, R_y, R_z are rotation matrices and S_c is a scale matrix.

If M' is known, M can be calculated and the equation of the quadric in the global coordinate system can be obtained.

$$\phi(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z, s_x, s_y, s_z, X) = X^T MX = 0,$$

where the parameters are the translation, rotation and scale.

Then, for each basic quadric, the square of the approximated distance $\delta_\phi(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z, s_x, s_y, s_z, X_p)$ from a three-dimensional point X_p to the quadric can be determined by

$$\begin{aligned} &\delta_\phi(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z, s_x, s_y, s_z, X_p)^2 \approx \qquad (1)\\ &\approx \frac{\phi(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z, s_x, s_y, s_z, X_p)^2}{\|\nabla\phi(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z, s_x, s_y, s_z, X_p)\|^2} = \\ &= \frac{\phi(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z, s_x, s_y, s_z, X_p)^2}{(\frac{\partial\phi}{\partial x})^2 + (\frac{\partial\phi}{\partial y})^2 + (\frac{\partial\phi}{\partial z})^2} \end{aligned}$$

Hereafter we use the above equation to calculate δ_ϕ for each quadric basic shape, which are all special cases of the above.

For a spherical surface with radius $r_0 = 1$, defined in its local coordinate system centered at the center of the sphere, we have

$$M' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

$$\phi(t_x, t_y, t_z, r, x, y, z) = (x-t_x)^2 + (y-t_y)^2 + (z-t_z)^2 - r^2 = 0.$$

For a cylindrical surface with radius $r_0 = 1$, defined in its local coordinate system, where the z axis is the axis of the cylinder,

$$M' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

The implicit equation in the global coordinate system is

$$\begin{aligned} &\phi(t_x, t_y, t_z, \theta_x, \theta_y, r, x, y, z) = \qquad (2)\\ &= D_1(x-t_x)^2 + D_2(y-t_y)^2 + D_3(z-t_z)^2 + \\ &+2C_1(x-t_x)(y-t_y) + 2C_2(x-t_x)(z-t_z) + \\ &+2C_3(y-t_y)(z-t_z) - r^2 = 0 \end{aligned}$$

where

$$\begin{aligned} D_1 &= \cos^2\theta_y, \\ D_2 &= \cos^2\theta_x + \sin^2\theta_x \sin^2\theta_y, \\ D_3 &= \sin^2\theta_x + \cos^2\theta_x \sin^2\theta_y, \\ C_1 &= \sin\theta_x \sin\theta_y \cos\theta_y, \\ C_2 &= -\cos\theta_x \sin\theta_y \cos\theta_y, \\ C_3 &= \sin\theta_x \cos\theta_x \cos^2\theta_y, \\ B_1 &= -t_x D_1 - t_y C_1 - t_z C_2, \\ B_2 &= -t_x C_1 - t_y D_2 - t_z C_3, \\ B_3 &= -t_x C_2 - t_y C_3 - t_z D_3. \end{aligned}$$

Note that (t_x, t_y, t_z) can be any point on the cylinder axis, thus the cylinder is over parameterized. This can be solved by setting one of these three parameters to zero.

For a cone surface with $g_0 = r_0/h_0 = 1$, where r_0 is the radius and h_0 is the height, defined in its local coordinate system, where the z axis is the axis of the cone and the origin of the coordinate system is the apex of the cone,

$$M' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The implicit equation in the global coordinate system is

$$\begin{aligned} &\phi(t_x, t_y, t_z, \theta_x, \theta_y, g, x, y, z) = \qquad (3)\\ &= D_1(x-t_x)^2 + D_2(y-t_y)^2 + D_3(z-t_z)^2 + \\ &+2C_1(x-t_x)(y-t_y) + 2C_2(x-t_x)(z-t_z) + \\ &+2C_3(y-t_y)(z-t_z) = 0 \end{aligned}$$

where

$$
\begin{aligned}
D_1 &= \cos^2\theta_y - g^2\sin^2\theta_y,\\
D_2 &= \cos^2\theta_x + \sin^2\theta_x\sin^2\theta_y - g^2\sin^2\theta_x\cos^2\theta_y,\\
D_3 &= \sin^2\theta_x + \cos^2\theta_x\sin^2\theta_y - g^2\cos^2\theta_x\cos^2\theta_y,\\
C_1 &= (1+g^2)\sin\theta_x\sin\theta_y\cos\theta_y,\\
C_2 &= -(1+g^2)\cos\theta_x\sin\theta_y\cos\theta_y,\\
C_3 &= (1+g^2)\sin\theta_x\cos\theta_x\cos^2\theta_y,\\
B_1 &= -t_xD_1 - t_yC_1 - t_zC_2,\\
B_2 &= -t_xC_1 - t_yD_2 - t_zC_3,\\
B_3 &= -t_xC_2 - t_yC_3 - t_zD_3.
\end{aligned}
$$

Finally, a plane is defined by the equation $ax+by+cz+d=0$. The square of the distance from a point $p=(x_p,y_p,z_p)$ to the plane is simply

$$\delta_\phi(a,b,c,d,x_p,y_p,z_p)^2 = \frac{(ax_p+by_p+cz_p+d)^2}{a^2+b^2+c^2}.$$

5 EXPERIMENTAL RESULTS

Our goal is to examine whether Biederman's observation, claiming that recognition can be accurate even if only a few geons of a complex object are visible (Biederman, 1995), is indeed feasible.

We tested our retrieval algorithm on a database consisting of 388 objects. Among the 388 objects we identified six classes: 19 models of human figures, 18 models of four-legged animals, 9 models of knives, 8 models of airplanes, 7 models of missiles and 7 models of bottles. The other models are unclassified.

Four different decomposition techniques were used in our experiments: (1) Greedy convex decomposition, where small patches are ignored; (2) Greedy convex decomposition, where small patches are merged with their neighbors; (3) Watershed decomposition, where small patches are ignored; (4) Watershed decomposition, where small patches are merged with their neighbors.

Based on these four decomposition techniques, four signature databases were built. Identical retrieval experiments were applied to each database. In each experiment, a test object was chosen and the system was queried to retrieve the most similar objects to this test object in ascending order. At least one member from each of the six classes was used as a test object.

Figures 3– 6 demonstrate some of our results. In each figure, the test object is the left-most, top object, and the objects retrieved are ranked from left to right. In particular, Figure 3 presents the most similar objects to *Detpl* (at the top-left), as retrieved by our algorithm. All the eight airplanes of the class were retrieved among the top eleven. Figure 4 presents the results of retrieving objects similar to *Cat2*. Sixteen out the eighteen members of the 4-legged animal class were retrieved among the top twenty. Figure 5 presents the retrieved most similar objects to *Knifech*. Eight out of the nine knifes of the class were retrieved among the top ten. Figure 6 demonstrates the most similar objects to the missile at the top left, as retrieved by our algorithm. Six out of the the seven class members were retrieved among the top nine. Note that in all the above cases the members of each class differ geometrically. Yet, their decomposition graphs are similar and therefore they were found to be similar.

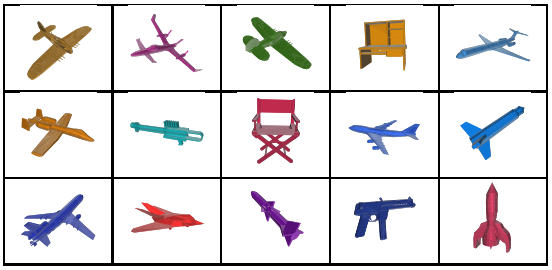


Figure 3: The most similar objects to *Detpl* (top left)

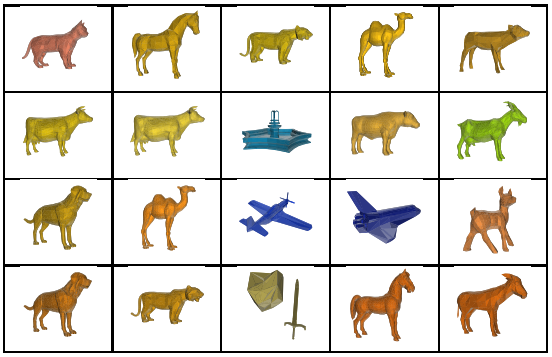


Figure 4: The most similar objects to *Cat2*

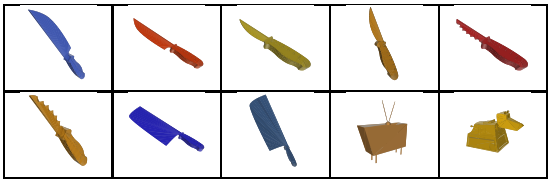


Figure 5: The most similar objects to *Knifech*

On the class of bottles, the algorithm does not perform as well. This class contains seven members (see Figure 7). Though the objects seem similar geometrically, their connectivity differs. The *Beer, Ketchup* and *Tabasco* bottles consist each of 4-8 disconnected components while *Bottle3, Champagne, Whiskey* and

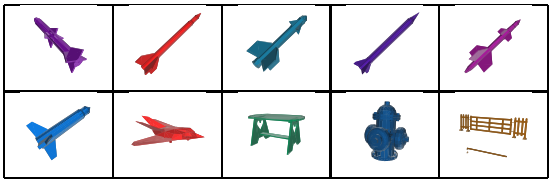


Figure 6: The most similar objects to *Aram*

Figure 7: The bottle class

Plastbtl consist each of only one or two components. Since connectivity determines the graph structure and the graphs differ, the results of the retrieval experiments are inferior to the other classes.

All four sub-methods performed well. The Watershed decomposition performed slightly better than convex decomposition. This fact might be surprising since convexity is the main factor in human segmentation. This can be explained by the fact that optimal convex decomposition cannot be achieved. Moreover, the height function used in the Watershed algorithm considers convexity as well.

Considering only the original large components and ignoring the small ones performs better than merging small components with their neighbors, both for watershed decomposition and for convex decomposition. This can be explained by the fact that merging results in complex shapes which might cause a failure of the basic shape determination procedure.

Table 1 shows some of our results for one sub-method – Watershed, ignoring small components. The first column shows the classes and the test objects. For each class, the number of members of the class N is shown. The next column of the table summarizes the results obtained for each test object. Each result (n/m) represents the number of the members of the same class n retrieved among the top m objects.

Class(N)/ Object	**Retrieved / Top results**		
Airplanes(8)			
Detplane	5/6	7/9	8/16
Worldw	6/6	6/9	8/16
747	5/6	5/9	7/16
Animals(18)			
Cat2	6/8	11/14	14/20
Tiger3	7/8	11/14	13/20
Deer	8/8	11/14	15/20
Humans(19)			
Woman2	10/10	17/17	19/24
Child3y	10/10	15/17	19/24
Knives(9)			
Knifech	6/6	8/8	8/15
Knifest	6/6	6/8	8/15
Missiles(7)			
Aram	3/6	5/10	
Bottles(7)			
Beer	1/3	1/6	

Table 1: Summary of the experimental results for the Watershed / ignore sub-method

6 CONCLUSION

This paper examines the adaptation of the human vision theories of Marr and Biederman to three dimensions. According to these theories, an object is represented by an attributed graph, where each node represents a meaningful component of the object, and there are arcs between nodes whose corresponding components are adjacent in the model. Every node is attributed with the basic shape found to best match the component, while each arc is attributed with the relative surface area of its adjacent nodes.

It was demonstrated that simple and efficient decomposition algorithms suffice to construct such a signature. We examined a couple of post-processing steps on top of well-known segmentation algorithms, in order to get only a handful of components. Moreover, a technique was presented for finding the best match between a given sub-mesh and pre-defined basic shapes. An error-correcting subgraph isomorphism algorithm was used for matching.

The experimental results presented in the paper are generally good. The major benefits of the signature is being invariant to non-rigid transformations and avoiding normalization as a pre-processing step. In addition, the algorithm for generating signatures is simple and efficient and produces very compact signatures.

The technique has a couple of drawbacks. First, the signature depends on the connectivity of the given objects, which might cause geometrically-similar objects to be considered different. Second, the graph matching algorithm we use is relatively slow. While the first drawback can be solved by fixing the models, the second problem is inherent to graph-based representations. More efficient graph matching algorithms should be sought.

REFERENCES

Bates, D. and Watts, D. (1988). *Nonlinear Regression and Its Applications*. John Wiley & Sons, New York.

Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147.

Biederman, I. (1988). Aspects and extensions of a theory of human image understanding. *Pylyshyn Z. editor, Computational Processes in Human Vision: An Interdisciplinary Perspective*, pages 370–428.

Biederman, I. (1995). Visual object recognition. *S. Kosslyn, D. Osherson, editors. An Invitation to Cognitive Science*, 2:121–165.

Chazelle, B., Dobkin, D., Shourhura, N., and Tal, A. (1997). Strategies for polyhedral surface decomposition: An experimental study. *Computational Geometry: Theory and Applications*, 7(4-5):327–342.

Cornea, N., Demirci, M., Silver, D., Shokoufandeh, A., Dickinson, S., and Kantor, P. (2005). 3D object retrieval using many-to-many matching of curve skeletons. In *IEEE International Conference on Shape Modeling and Applications*, pages 368–373.

Duda, R., Hart, P., and Stork, D. (2000). *Pattern Classification*. John Wiley & Sons, New York.

Elad, M., Tal, A., and Ar, S. (2001). Content based retrieval of vrml objects - an iterative and interactive approach. *EG Multimedia*, 39:97–108.

Hilaga, M., Shinagawa, Y., Kohmura, T., and Kunii, T. (2001). Topology matching for fully automatic similarity estimation of 3D shapes. *SIGGRAPH*, pages 203–212.

Katz, S., Leifman, G., and Tal, A. (2005). Mesh segmentation using feature point and core extraction. *The Visual Computer*, 21(8-10):865–875.

Katz, S. and Tal, A. (2003). Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. Graph. (SIGGRAPH)*, 22(3):954–961.

Kazhdan, M., Chazelle, B., Dobkin, D., and Funkhouser, T. (2003a). A reflective symmetry descriptor for 3D models. *Algorithmica*, page to appear.

Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S. (2003b). Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Symposium on Geometry Processing*.

Keren, D., Cooper, D., and Subrahmonia., J. (1994). Describing complicated objects by implicit polynomials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):38–53.

Lee, S., Kim, J., and Groen, F. (1990). Translation-, rotation-, and scale-invariant recognition of hand-drawn symbols in schematic diagrams. *Int. J. Pattern Recognition and Artificial Intelligence*, 4(1):1–15.

Lee, Y., Lee, S., Shamir, A., Cohen-Or, D., and Seidel, H.-P. (2005). Mesh scissoring with minima rule and part salience. *Computer Aided Geometric Design*.

Leifman, G., Meir, R., and Tal, A. (2005). Semantic-oriented 3D shape retrieval using relevance feedback. *The Visual Computer*, 21(8-10):649–658.

Li, X., Toon, T., Tan, T., and Huang, Z. (2001). Decomposing polygon meshes for interactive applications. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 35–42.

Mangan, A. and Whitaker, R. (1999). Partitioning 3D surface meshes using watershed segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):308–321.

Marr, D. (1982). *Vision - A computational investigation into the human representation and processing of visual information*. W.H. Freeman, San Francisco.

Messmer, B. (1995). *GMT - Graph Matching Toolkit*. PhD thesis, University of Bern.

Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. (2001). Matching 3D models with shape distributions. In *Proceedings of the International Conference on Shape Modeling and Applications*, pages 154–166.

Paquet, E., Murching, A., Naveen, T., Tabatabai, A., and Rioux, M. (2000). Description of shape information for 2-D and 3-D objects. *Signal Processing: Image Communication*, pages 103–122.

Pearce, A., Caelli, T., and Bischof, W. (1994). Rulegraphs for graph matching in pattern recognition. *Pattern Recognition*, 27(9):1231–1246.

Rocha, J. and Pavlidis, T. (1994). A shape analysis model with applications to a character recognition system. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16:393–404.

Shamir, A. (2004). A formalization of boundary mesh segmentation. In *Proceedings of the second International Symposium on 3DPVT*.

Subrahmonia, J., Cooper, D., and Keren, D. (1996). Practical reliable bayesian recognition of 2d and 3D objects using implicit polynomials and algebraic invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):7505–519.

Sundar, H., Silver, D., Gagvani, N., and Dickinson, S. (2003). Skeleton based shape matching and retrieval. In *Shape Modelling and Applications*.

Taubin, G. (1991). Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1115–1138.

Veltkamp, R. (2001). Shape matching: Similarity measures and algorithms. In *Shape Modelling International*, pages 188–197.

Vranic, D. and Saupe, D. (2002). Description of 3D-shape using a complex function on the sphere. In *Proceedings IEEE International Conference on Multimedia and Expo*, pages 177–180.

Wang, Y.-K., Fan, K.-C., and Horng, J.-T. (1997). Genetic-based search for error-correcting graph isomorphism. *IEEE Trans. Systems, Man, and Cybernetics*, 27:588–597.

Wong, E. (1992). Model matching in robot vision by subgraph isomorphism. *Pattern Recognition*, 25(3):287–304.

Zuckerberger, E., Tal, A., and Shlafman, S. (2002). Polyhedral surface decomposition with applications. *Computers & Graphics*, 26(5):733–743.