In the proceedings of the *Pacific Graphics 2003*, Canmore, Canada, Oct. 2003

Shape-Similarity Comparison of 3D Models Using Alpha Shapes

Ryutarou Ohbuchi, Tsuyoshi Takei
ohbuchi@acm.org, f8058@kki.yamanashi.ac.jp
Interdisciplinary Graduate School of Medicine and Engineering,
University of Yamanashi, 4-3-11 Takeda, Yamanashi-shi, Japan

Abstract

As the number of in-house and public-domain 3D shape models increase, importance of shape-similarity based search and retrieval for 3D shapes models has increased rapidly. In this paper, we describe our preliminary findings in applying a multiresolution analysis technique to the task of shape similarity comparison of polygon soup models. We used the 3D alpha shapes *algorithm to create a multiresolution hierarchy of shapes from the given 3D model. We then applied a (single resolution) shape descriptor to each of the models at multiple resolution levels to derive a multiresolution shape descriptor. According to our evaluation experiments, the retrieval performance of our multiresolution descriptor outperformed its single resolution counterpart, proving the effectiveness of the basic approach.*

Keywords: content-based search and retrieval, geometric modeling, 3D alpha shapes, polygon soup.

1. Introduction

Proliferation of 3D models prompted development of the technology for effective content-based search and retrieval of three-dimensional (3D) models. A 3D model could be searched by textual annotation by using a conventional text-based search engine. This approach wouldn't work in many of the application scenarios. The annotations added by human beings depend on language, culture, age, sex, and other factors. It is also very difficult to describe by words shapes that are not in the well-known shape or semantic categories. It is thus necessary to have a content-based search and retrieval systems for 3D models that are based on the features intrinsic to the 3D models, most important of which is the shape [1, 13, 24, 27, 8, 15, 21, 18, 11, 30, 31, 32, 4, 17, 19, 22, 33, 9, 16, 20].

One of the first questions one would ask in developing a shape similarity matching system is the kind of shape representation(s) the system accepts. 3D CAD models, many of which are defined as either 3D solids or manifold surfaces, are quite important. Solids and manifolds (e.g., mesh surfaces) are amenable to many mathematically sound analysis techniques. However, a majority of the important and useful models around are defined as polygon soup, examples being the VRML, MPEG-4 SNHC and file formats for many proprietary computer-animation software packages. As it is one of the most dominant shape representations currently in use, we must develop a shape similarity search algorithm for polygon soup models.

In the field of content-based search and retrieval of 2D images, shape of an object is one of the most important of the features, in addition to color, texture and others [28, 29]. In analyzing shape and texture of objects in the images, many *Frequency domain* and *multiresolution* image analysis techniques are found to be very powerful tools. Examples of such techniques include, but not limited to, Fourier transform, various wavelet-based pyramids, Laplacian pyramid, Gaussian pyramid, and morphological pyramids [10, 25]. Frequency domain and multiresolution analysis techniques found their ways into processing of 2-manifold surfaces embedded in 3D space. However, these 2-manifold based analysis techniques are not directly applicable to 3D shapes defined as polygon soup, which are not manifolds.

One possible approach to apply a multiresolution analysis technique to a polygon soup 3D shape is to scan-convert the shape into a voxel-based representation, as in [31, 9]. A whole battery of existing image analysis techniques could then be applied to the voxel-based model, since it is essentially a 3D image.

This paper presents a new alternative method to shape comparison of 3D models that employs multiresolution analysis approach without relying on voxelization of the 3D models. From an input polygon-soup 3D model, our method computes a 3D multiresolution representation (MRR) based on *n-simplices* by using the *three-dimensional alpha-shapes* proposed by Edelsbrunner [7]. To compute the MRR, we specify a set of alpha value spaced at power of 2 intervals to the alpha shapes algorithm. This multiresolution scheme is reminiscent of the morphological multiresolution analysis for 2D image analysis. Our method then computes, for each model in the MRR, a shape descriptor by applying the *Absolute Angle Distance* (AAD) shape descriptor [20]. An ordered set of multiple AAD shape descriptors for the MRR is our multiresolution shape descriptor for the input 3D model.

Experiments showed that the proposed multi-resolution shape similarity comparison method outperforms in retrieval performance the single-resolution method using the AAD. As a tradeoff, computational cost of our multiresolution method is higher than that of the AAD.

The rest of this paper is organized as follows. In the following section, we review the previous work on shape similarity matching of 3D shapes, especially those that are defined as polygon soup. In Section 3, we describe the proposed algorithm. In Section 4, we evaluate the performance of the proposed method through experiments. We conclude the paper in Section 5.

2. Previous Work

There are four major steps in shape-based retrieval of 3D models from a 3D model database (See Figure 1.)

(1) **Query formation**: Form and present a query. Alternatives for querying 3D shapes include a 3D shape given a priori, a 3D sketch of the 3D shape, or a 2D sketch of a 3D shape.

(2) **Feature extraction**: Extract feature vectors, or shape descriptor, from the model to be used for shape similarity (more often, dissimilarity) computation. Shape representation of the target 3D models influences the shape features that can be employed.

(3) **Dissimilarity computation**: Compute dissimilarity value between shapes. Usually, the dissimilarity values are expected to reflect human judgments.

(4) **Retrieval**: Efficiently retrieve, from the database, the models having the lowest dissimilarity values.

In the following, we review the item (2) and (3) above, as our focus in this paper is a shape descriptor and its distance computation method.

A shape descriptor can be classified by 3D shape representations it accepts. Our targets are surfaces based 3D shape representations. Roughly speaking, a surface based 3D shape representation can be classified into solid, manifold, or polygon soup.

I call the first two, solid and manifold, "well-defined" representations. A solid is a real 3D object, whose volume can be properly computed. Many geometric CAD models fall in this category, and shape features have been proposed for this class of 3D models [17, 18, 15, 4]. Other 3D models may not be solid, but are defined as sets of manifolds. A manifold model allows computation of such differential geometric properties as surface curvature (e.g., [32]). It also allows computation of topological properties, e.g., an extension of the *Reeb graph* proposed by Hilaga, et al [11].

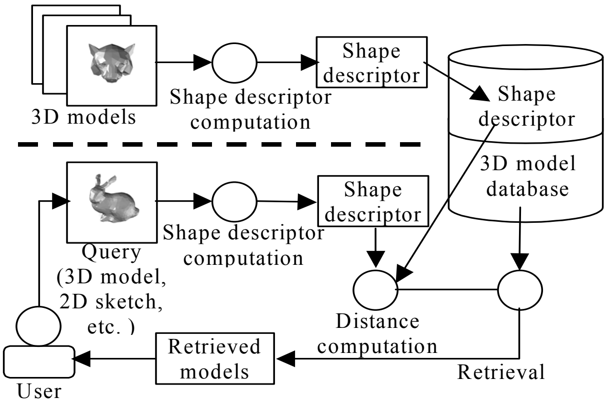


Figure 1. A generic diagram for a shape similarity search database for 3D geometric models.

The last kind, polygon soup, is "ill-defined" in that it does not really define a 3D object or manifold surface. It is a collection of points, lines, independent polygons, and meshes that create an *illusion* of 3D shape. A VRML model may also contain such parameterized shapes as sphere and cone. Despite its "ill-defined" nature, most of the models available on the net (e.g., VRML models), or the models used in the entertainment industry are polygon soup models. We have to somehow deal with polygon soup model.

2.1. Shape descriptors for polygon soup models

In this section, we review the shape descriptors for polygon soup models. A method in this class can be classified further depending on if it requires pose normalization to compute shape descriptor.

Some of the previous methods required pose normalization prior to shape comparison. Paquet et al [23] computed, after pose normalization, a set of geometrical features. They combined the shape with other features such as color for their shape similarity search. Suzuki et al [26, 27] computed, after pose normalization, distribution of vertices in the uniformly subdivided axis-aligned grid. Zaharia [33] employed a 3D Hough transformation as the shape feature, after pose normalization. Both [27] and [33] took advantage of symmetries of their shape features so that their pose normalization can be simplified. Elad et al [8] also normalized pose and computed various moments from the points generated randomly on the surface. The method by Ohbuchi et al [19] first normalizes pose by using moments. Then it computes moment of inertia, average and variance of distance from the (inertial) axis to the model surface. According to our experiment, which is yet to be published, the retrieval performance of this method [19] is better than the Osada's D2 [21, 22], and is comparable to the AD and AAD [20] described below. However, these and other methods, which require pose normalization, could run into trouble if pose normalization fails.

Other shape features targeting polygon soup are invariant to similarity or other transformations, and thus do not require pose normalization [1, 31, 21, 22, 9].

Osada et al [21, 22] proposed and compared statistical features that are invariant to rigid body transformation. The best performing one both in terms of computation cost and retrieval performance is the D2 shape function. The D2 is a 1D histogram of Euclidian distances between randomly selected pairs of points located on the model's surface. The points are generated at random location on each polygon. Their method is quite robust, and for its simplicity, performed quite well both in terms of computational cost and retrieval performance.

Ohbuchi et al [20] adopted Osada's approach and improved retrieval performance of the D2 with only modest increases in computational costs. Their shape descriptors, called *Angle Distance* (*AD*) and *Absolute Angle Distance* (*AAD*) histograms are 2D histogram having the axes of distance and angle. Both AD and AAD take into account not only the distance between the point pair but the angle formed by the surface normal vectors at the pair of points. Their experiments showed that the AAD significantly outperformed the Osada's D2. The method described in this paper uses the AAD as a component.

Ankerst [1] proposed one of the first 3D shape similarity matching algorithms. Their algorithm is not directly applicable to 3D polygonal models, as they targeted 3D molecular database, in which each model is a collection of points (atoms) with properties (e.g., a mass or a label "hydrogen".) Nonetheless, their basic approach is applicable to 3D polygonal models as well. One of their shape descriptor parameterizes a 3D space using concentric shells, making the feature invariant to rotation. Their other parameterizations subdivided the space radially, which made them rotationally variant.

The shape descriptor of Funkhouser et al [9] combines the concentric parameterization of Ankerst [1] with the spherical harmonic approximation of Vranić et al. [31]. They first scan-converts polygons of a 3D model into a voxel buffer of size 64^3 in a Cartesian coordinate system. Then, at concentric shells, the distribution of voxels in the voxel buffer is approximated by a set of spherical harmonic coefficients. Experimental evaluation showed that their method outperformed several other methods, including those by Osada [21, 22] and Ankerst [1]. Significance of their work also lies in the fact that they constructed a well-developed 3D model database system. The system has a publicly accessible web-based query interface that accepts keyword text, 2D sketch, 3D sketch (using *Teddy* [12]), and 3D example model as the query. Min et al describes their initial experience with using the database [16].

Some of the shape descriptors, for example, those by Vranić [31] and Funkhouser [9] employed "frequency domain" or "multiresolution" approaches. Our method falls in this category. We idea was to apply an analogue of morphological multiresolution analysis to polygon soup models for their shape similarity comparison.

3. The Shape Similarity Comparison Algorithm

Our method assumes, as its input, models defined as polygon soup. The method does not use connectivity of the faces in comparing the shapes. The method accepts, but simply ignores, anything that does not have surface area, e.g., a zero-area polygon and a polyline.

The proposed method uses a multiresolution shape descriptor for shape comparison. To do so, the method first computes, from a polygon-soup 3D model, a 3D multiresolution representation (MRR) based on *n-simplices* by using the *three-dimensional alpha-shapes* proposed by Edelsbrunner [7]. (For brevity, in the following of this paper, we will call this *alpha-Multiresolution Representation, or AMRR* for short.) Computation of AMRR starts by converting the surface-based model into a point set model by using a Monte-Carlo sampling of the surface geometry. Then the AMRR, which is a set of alpha shapes, is computed by using L alpha value spaced at power-of-2 intervals.

Our multiresolution shape descriptor is called the *Alpha Multiresolution AAD* (*AMR-AAD*) for we combined the *Absolute Angle Distance* (AAD) shape descriptor [20] with the AMRR. An ordered set of the AAD descriptors computed for multiple resolution levels forms the AMR-AAD. Note here that the AMRR is a multiresolution framework not tied to the AAD shape descriptor. Any shape descriptor that satisfies certain requirements can be used with the AMRR.

A distance between a pair of AMR-AAD shape descriptors is the distance, or *dissimilarity* of the corresponding pair of 3D models.

To repeat, the method compares 3D models for their shape similarity by following the steps below;

1. **Converting to a point set model**: Convert the input surface-based model into a point-based model by Monte-Carlo sampling the surface of the model.
2. **Generating multiresolution representation**: First, a set of L-1 scale values α_i $(i = 1, 2, ..., L-1)$ for the

multiresolution representation is found based on the size statistics of the model. This is necessary to compare models having different sizes. Then, compute *L*-1 3D alpha shapes [7] from the point set model by using the *L*-1 scale.

3. **Computing the shape descriptor**: Compute the AMR-AAD consisting of *L* AAD shape descriptors. Of L shape descriptors, *L*-1 are computed from the *L*-1 alpha shapes above, and the remaining one is computed from the original (input) face-based model.
4. **Computing distance between models**: Compute the distance between the shape descriptors of a pair of models to be compared for their shape (dis-)similarity.

We will explain each of the steps above in detail in the following sections.

3.1. Converting to a point set model

To compute an AMRR for a 3D polygon-soup model, we first create a point set model for the polygon-soup model. Instead of just using the original vertices, we sampled the surface geometry by generating points located (*quasi*-) randomly on the surfaces of the model. This sampling is necessary to avoid problems caused by variation in surface tessellation.

First, if the model contained non-triangular polygons, they are triangulated prior to the point generation. Then, a set of points on a polygon is generated. Coordinate **P** of a point on a triangle is determined by using the equation below by Osada et al. [Osada02].

$$\mathbf{P} = \left(1-\sqrt{r_1}\right)\ \mathbf{t}_1 + \sqrt{r_1}\ \left(1-r_2\right)\ \mathbf{t}_2 + \sqrt{r_1}\ \left(r_2 \cdot \mathbf{t}_3\right). \quad (2)$$

In the formula, $\mathbf{t}_1$, $\mathbf{t}_2$, and $\mathbf{t}_3$ are vertices of the triangle, and r_1 and r_2 are *Sobol*'s quasi-random number sequences (QRNS, also known as low-discrepancy sequences). The number of points per triangle is proportional to the area of the triangle. Instead of the pseudo-random number sequence (PRNS) employed by Osada, et al, we used the *Sobol*'s QRNS for QRNS provides faster conversion when applied to Monte-Carlo sampling. We typically approximate a surface based 3D model by using 1000 to 2000 points before computing 3D alpha shapes.

3.3 Generating multiresolution representation

A set of *3D alpha shapes* [7] is a family of 3D shapes computed from a 3D point set given various values of real-valued parameter α. If $\alpha = \infty$, the alpha shape is equal to convex hull of the point set, and if $\alpha = 0$, the alpha shape equals the input point set. If $\infty > \alpha > 0$, the alpha shape is a set of points (0-simplex), lines (1-simplex), and triangles (2-simplex).

Figure 2 illustrates an example of a 2D alpha shape (a polygon drawn in solid line) generated from the 2D point set. Given the point set and α, a circle is moved around on the point set. When the circle "rests" on a pair of points, a straight line is drawn between the pair of points. The result in this case is a concave polygon. It can be imagined easily that if $\alpha = \infty$, the resulting alpha shape is a convex hull.

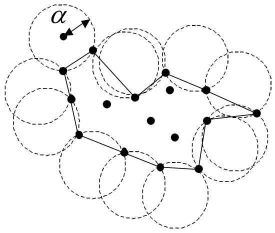


Figure 2. A 2D example of an alpha shape (the polygon in solid line) computed from the point set.

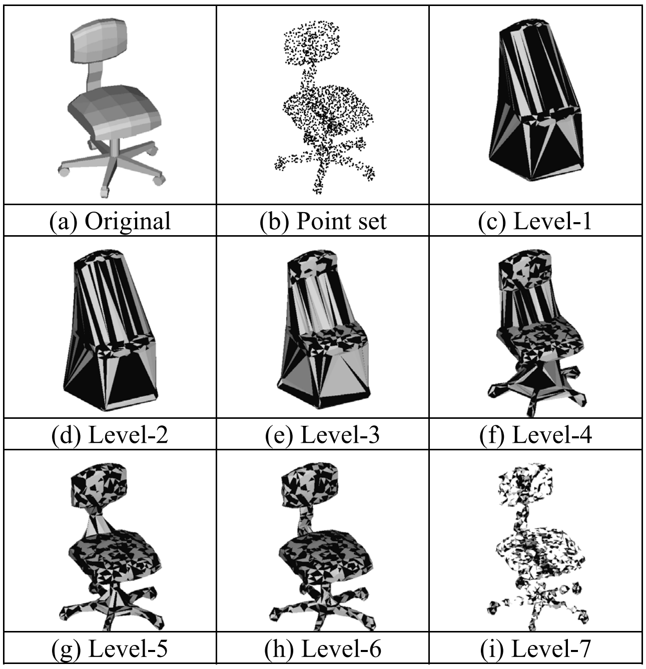
In order to compare the models having different sizes, shape descriptors must be invariant to size of the models. We achieve this by making the values of α proportional to the size of each model. Given the point set, we first determine the minimum d_{min} and the maximum d_{max} distances among all the possible pairs of the points generated on the model (e.g., $N_p(N_p$-1)/2 pairs for the N_p points). The values of alpha α_i $(i = 1, 2, ..., L-1)$ for the levels *i* are computed as follows;

$$\alpha_i = \frac{d_{\max} - d_{\min}}{2^{i-1}} \qquad (i = 1, 2, ..., L-1) \qquad (1)$$

That is, an increase in the level *i* halves the value α_i. For example, a model at level 1 is very close to the convex hull of the original model. As the level *i* increases, that is, as the value of α_i decreases, smaller features appear in the model. At $i = \infty$, α_i becomes 0, and the resulting alpha shape is a point set. This set of *alpha shapes* generated from a 3D point set by using multiple values of α is somewhat analogous to a morphological multiresolution representation in 2D image processing.

For the experiments described in Section 4, we computed AMRR at 5 different α values. The use of 5 levels is reasonable since an alpha shape produced at the level 5 is analogous in spatial resolution to a voxel model having 32^3 voxels. We believe that a voxel model of resolution 32^3 could typically capture significant part of the important shape features in a good majority of 3D shapes. In addition to these models at 5 resolution levels, AMR-AAD uses the original model so that features are not lost during the point sampling and reconstruction using the alpha shapes algorithm. In the following, for convenience, we call the original level as the "Level 6" model. (This name is somewhat justified as the original model would contain the most detailed shape.)

Figure 3 shows an example of an AMRR generated from a face-based 3D model of an office chair. Figure 3a and Figure 3b show, respectively, the original model and its point set model. Figure 3c-3i show, respectively for the eight levels $i = 1, 2, 3, 4, 5, 6, 7$, alpha shapes computed from the point set model of Figure 3b. A model generated by using a large α (e.g., level i=1 or 2) captures global shape, while a model generated by using a small α (i.e., level i=5 or 6) captures detail. As the value of α becomes even smaller, alpha-shapes produced become fragmented (Figure3i), and in the end becomes the input point set.



(a) Original (b) Point set (c) Level-1

(d) Level-2 (e) Level-3 (f) Level-4

(g) Level-5 (h) Level-6 (i) Level-7

Figure 3. A set of alpha shapes generated from a set of 1024 point generated on the surface of the original model.

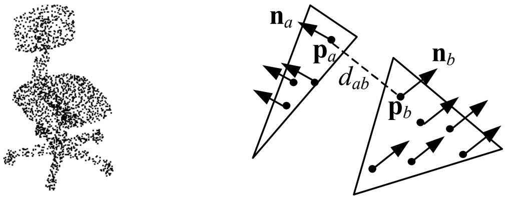
3.3. Computing the shape descriptor

The AMRR requires a shape descriptor that is insensitive to inconsistent surface (normal) orientations, as the surfaces in the alpha shapes are not oriented. We chose to combine the AMRR with the AAD shape descriptor [20] since the AAD satisfies the requirement above, and that the AAD has a good retrieval performance for its relatively low computational cost. An additional reason is the saving in computational costs, as both AAD and AMRR uses points generated (quasi-) randomly on the surfaces of the 3D model. Note, however, that the AMRR is not tied to the AAD; the AMRR may be combined with any (single-resolution) shape descriptor that satisfies the surface orientation insensitivity requirement mentioned above.

The AAD [20] is based on the Osada's D2 shape function [21, 22]. The advantages of the D2 are that it is robust against topological and geometrical irregularities and degeneracies, and that it does not require pose normalization. It is also relatively low cost to extract and compare D2 shape descriptors.

To compute the D2 shape function for a 3D model, points are generated at random location on every surface of the model. The number of point per polygon is made proportional to the area of the polygon. This step is the same as the point generation in creating AMRR. Then, Euclidian distance (in 3D) is computed for every possible point pair, *i.e.*, the $N_p(N_p\text{-}1)/2$ pairs for the N_p points generated. The feature vector of the D2 shape function is the 1D histogram generated by counting the population of pairs that falls within a certain distance interval.

We extended the D2 by adding the histogram of surface orientation to create the AAD shape descriptor [20]. When points are generated on the surfaces of a model (Figure 4a), each point is attached with the normal vector of the surface on which the point is generated (Figure 4b). As with the D2, for every pair of points, the AAD computes the 3D Euclidian distance between the pair of points. In addition, the AAD computes the inner product of the surface normal vectors attached to the points. The AAD is a 2D histogram, whose axes are the absolute value of the inner product of the surface normal vectors and the distance between the pair of points. The AAD was shown to have higher retrieval performance than the D2, while having the computational cost comparable to the D2.



(a) Points generated on the surface of the office chair model.

(b) Each point has its 3D coordinate and the normal vector of the surface it is on.

Figure 4. The AAD computes the histogram of distance and mutual orientation of the points on the model's surfaces.

The 2D histogram still requires normalization for compare models having different size. We used the distance average based normalization of the distance axis of the 2D histogram, which performed the best among the four methods we tried [20]. Assume that the 2D histogram has the I_d distance intervals and I_a angular (or, absolute value of the inner-product) intervals. The normalization method first finds the average distance, and subdivides the distance axis of the histogram (having the total of I_d intervals) into two, which are, above and below the average distance. Then, each of the upper and lower half of the histogram is divided into $I_d/2$ equally spaced intervals. The sizes of the interval may differ above and below the average. In the angular axis, the range of the absolute value of the inner product [0,1] is subdivided into I_a equally spaced intervals. The resulting AAD is a 2D histogram having $I_a \times I_d$ elements.

An AMR-AAD descriptor having L resolution levels is a 3D histogram of dimension $L \times I_a \times I_d$, which can be written as 3D vector $\mathbf{x} = (x_{ijk})$, in which $(i = 1, 2, ..., L)$ $(j = 1, 2, ..., I_a)$, and $(k = 1, 2, ..., I_d)$.

Figure 5 shows an example of the AAD shape descriptor computed for the office chair model of Figure 3 using the average-based distance normalization method. It is computed by using N_p=1024 points generated by the *Sobol*'s QRNS, and the numbers of histogram interval are I_d = 64, and I_a = 4. Figure 6 shows, as 2D images, AMR-AAD descriptors for the office chair model and a car model. The histograms are computed by generating N_p=1024 points and using the number of intervals I_d = 64, and I_a = 4. The darker the color, the higher the histogram values are. (Note that the images are histogram-equalized for the illustration purpose.)

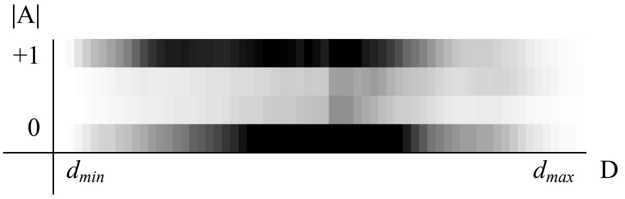


Figure 5. The AAD descriptor for the Level-1 model of the office chair in Figure 3.

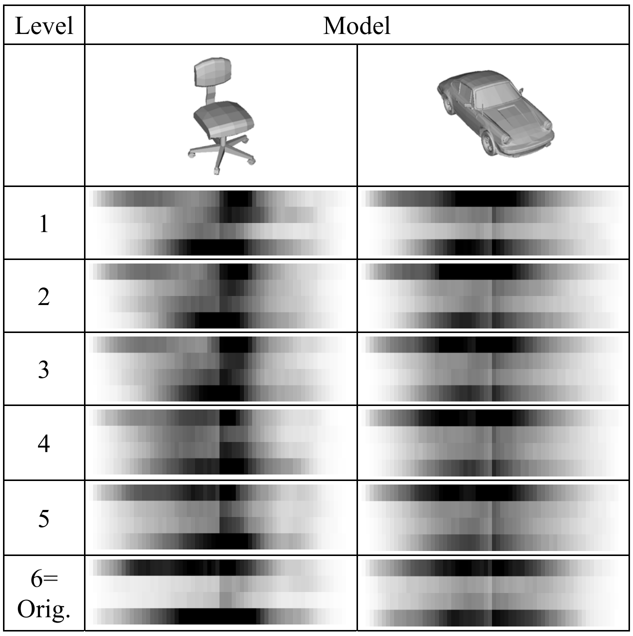


Figure 6. A pair of AMR-AAD shape descriptors, each of which uses AMRR models at 5 resolution levels (L=1…5) as well as the original model. The original model is referred to as the level-6 model for convenience.

3.4. Distance computation

A distance, or dissimilarity, among a pair of models X and Y are computed by using their respective shape descriptors **x**=(x_{ijk}) and **y**=(y_{ijk}). The distance $d(\mathbf{x},\mathbf{y})$ between the shape descriptor **x** and **y** are defined as follows;

$$d(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{L} w_i \sum_{j=1}^{I_d} \sqrt{\sum_{k=1}^{I_a} \left(x_{ijk} - y_{ijk}\right)^2} \qquad (3)$$

It is a simple weighted average of the distance at each resolution level. The smaller the distance value $d(\mathbf{x},\mathbf{y})$, the more similar the pair of models is. In the equation, the number of AMRR level is L, the number of distance intervals is I_d, and the number of angular intervals is I_a.

The w_i are the weights of the descriptor at multiple AMRR levels for weighted averaging. A larger value of w_i for a resolution level means that the level has the larger effect on the result.

4. Experiments and Results

We have implemented the proposed algorithm using C++ language on a Linux platform. To compute alpha shapes, we used a robust and efficient implementation of the alpha shapes algorithm called the *Hull* by Clarkson et al [3].

4.1. Experimental Method

In the case of content-based search and retrieval of 2D images, several de-facto-standard image databases with known categories exist. In the case of shape similarity retrieval of 3D models, most researchers must be satisfied with a few hundreds to a thousand free 3D mesh models collected from the Internet. Obviously, performance figures would depend very much on the model database and the categories used for the experiment. Changes in the models contained in the database or changes in the categorization will produce results different from those reported below. Performance comparison among methods without actually implementing them requires a standard database and standard category. Us researchers should work together to create such a database (or two or three!)

In our case, we manually collected 1200 free polygon soup 3D models, most of which are VRML models, from the Internet. We then converted their format to that of the VRML 97 using conversion tools.

To measure performance, we classified the 1200 models into 35 categories based on the judgment of two adult male persons. Of the1200 models, 861 are categorized into one of the 34 "*known*" categories. The "known" categories include such shape and/or semantic categories as "Car", "Lamp", "Chair", "Officechair", "Humanoid", "Plane1", "Head", and "Mug". The 35th category is the "Other" kind, which contained remaining 339 hard-to-classify models. In the experiments below, we queried models in the known categories only. If a query using a model from a known category produced models from the "Other" category, those models from the "Other" category are counted as failures. Thus, large number of model in the "Other" category push down the performance figures, such as FT, ST, NN, figures and recall-precision plots.

Retrieval performance of the proof-of-the-concept system has been evaluated by using the *First Tier (FT)*, *Second Tier (ST)*, and *Nearest Neighbor (NN)*, as well as the *recall-precision* plot.

First Tier (FT): Assume that the query belongs to the class C_q containing k models. The FT figure is the percentage of the models from the class C_q in the top (k-1) matches. As the query model is excluded, (k-1) models from the class C_q in the top (k-1) results produces the figure 100%.

Second Tier (ST): The ST figure is the percentage of the models from the class C_q in the top 2(k-1) matches.

Nearest Neighbor (NN): The percentage of the cases in which the top match is drawn from the query's class C_q.

Recall and precision are defined as follows. Let M be the set of all the models in the database, and the query presented by the user is an element in the set $m_i \in M$. Let the category to which m_i belongs to be C_i . The category C_i contains $\left|C_i\right|$ models including the m_i itself. Assume that the systems returned the set of models S, in which the models the system determined most similar models to the query m_i are included. Then, the recall and precision values averaged over all the models, that are, $\hat{R}$ and $\hat{P}$ and are computed as follows;

$$\hat{R} = \frac{1}{|M|}\sum_{i=1}^{|M|}\frac{|S \text{ I } C_i|}{|C_i|} \tag{4}$$

$$\hat{P} = \frac{1}{|M|}\sum_{i=1}^{|M|}\frac{|S \text{ I } C_i|}{|S|} \tag{5}$$

The maximum values for both $\hat{R}$ and $\hat{P}$ are 1.0. Recall and precision are in trade-off relationship so that improving one degrades the other. For example, recall value $\hat{R}$ =1.0 can be trivially achieved if all the $|M|$ models are retrieved. But such a retrieval result is useless as the result include every model in the database.

To plot the tradeoff between the recall and precision, we computed $\hat{R}$ and $\hat{P}$ by varying the number of retrieval $|S|$ in the range $1 \le |S| \le |M|$.

4.2. Retrieval performance

There are two major parameters to be considered in the AMR-AAD shape descriptor, the number of resolution levels L, and the weights w_{i} for the resolution level i (See equation (1)).

As mentioned before, we used L=5, but added the original model (called "level 6") to make the total number of resolution levels 6. To compute the AMR-AAD, all but the case H used N_p=1024, I_d=64, and I_a=4, for each resolution level. That is, if two resolution levels are used, total number of points used for the descriptor is 2048. In the case H, the N_p=6144, I_d=64, and I_a=4 are used.

As for the weights, we compared seven different combinations of weights that are listed in Table 1 (The cases A and H have the same weights but different N_p values.). For example, case A used $w_i = 0.0$ for i=1, 2, 3, 4 and 5, and $w_i = 1.0$ for $i = 6$. That is, the case A is a single resolution shape descriptor that is exactly the same as the AAD shape descriptor [Ohbuchi03]. As another example, the case F employed weights 1.0, 2.0, 3.0, 4.0, 5.0, and 6.0, respectively, for the levels 1, 2, 3, 4, 5, 6(=original), respectively.

4.2.1. Effect of the multiresolution approach

According to the experiments listed in Table 1, the multiresolution approach appears to improve performance. The performance figures for those cases (cases B, C, D, E, F, G) employing multiple resolution levels are better than the case using only one resolution (case A). For example, the case B, which employed both the original model and its convex hull, outperformed the case A, which used the original model only. Further improvements in performance was observed as more resolution levels are recruited, as seen in the cases C, D, E, and F.

Note that the AMR-AAD actually uses more points for the shape descriptor; if it uses 6 resolution levels, 6 times more points ($6,144 = 1,024 \times 6$) are used. We did an experiment to know whether the multiresolution approach or the mere increase in number of points is the main source of performance gain. We used N_p=6144 for the AAD (case H) and compared its performance with the others. The result shows that any non-degenerate AMR-AAD (cases B, C, D, E, F, G) performed better than the single resolution AAD using a larger number of points (N_p=6144, case H). Note that the AAD using N_p=6144 cost 6 times more to compute than the AMR-AAD using 6 levels, as the computational cost of AAD is $O\left(N_p^2\right)$.

Note also that the AAD that used N_p=6144 (case H) performed only slightly better than the AAD using N_p=1,024 (case A) while costing 36 times more in terms of computation. Changes in the number of histogram intervals I_d and I_a could affect the performance of the AAD.

But increases in I_d and I_a would actually decrease the overall performance by making the descriptor too sensitive to minute differences in shape (that is, a higher precision but a lower recall.)[20].

Table 1. Weighting of the resolution levels in AMR-AAD shape descriptors and retrieval performance. All but the case H used N_p=1024 for computing the AAD shape descriptor for each resolution level. The case H used N_p=6144 just for one AAD descriptor at level 6.

Weights for the resolution levels							Performance		
Level / Case	1	2	3	4	5	6	FT [%]	ST [%]	NN [%]
A (=AAD)	0	0	0	0	0	1	24.0	35.7	43.1
B	1	0	0	0	0	1	25.9	37.7	46.3
C	1	0	1	0	0	1	26.7	38.4	46.6
D	1	0	0	1	0	1	27.2	38.8	49.5
E	1	1	1	1	1	1	27.3	39.0	50.0
F	1	2	3	4	5	6	27.9	39.9	51.8
G	1	2	4	8	16	32	27.5	39.8	51.2
H (=AAD)	0	0	0	0	0	1	24.0	36.0	44.3

4.2.2. Weighting and retrieval performance

Table 1 shows that, compared to the even weighting across the resolution levels (the case E), weighting the resolution levels unevenly (the case F) could improve performance. Also, putting more weight on the higher resolution descriptor(s) improved performance. This effect can be observed by comparing the case C and D, and by comparing the case E and F. However, there is a limit to this; too much weight on the higher resolution descriptors actually degraded performance, as seen in the cases G compared to the case F.

4.3. Performance variations due to category

Obviously, there are variations in performance from one category to another. Figure 7 shows the recall-precision plots from three categories, “4 legged animal”, “head”, and “humanoid”. The “4 legged animal” category and “head” category included models that are geometrically similar. However, the “humanoid” category contained “humanoid” in many different postures. Hence, retrieval performance of the “4 legged animal” category and “head” category are much better than the “humanoid” category. A category that is of semantic in nature produces a low score.

4.4. Comparison with other shape descriptors

We compared the performance of the best-performing AMR-AAD descriptor (the case F in Table 1) with the AAD [20] and our implementation of the D2 [21, 22]. We call our implementation of the Osada’s D2 as *mD2*, for the implementation details differ from that of Osada’s. (For example, we used QRNS to compute mD2, instead of the PRNS in the Osada’s D2.)

The result is shown in Table 2 for the FT, ST, and NN figures, and in Figure 8 for the recall-precision plot. The AMR-AAD multiresolution shape outperformed AAD and mD2 in every performance measures. For example, the FT figure improved by more than 7% over the mD2, and the NN figure improved more than 14% over the mD2. The recall-precision plot shows that the AMR-AAD outperformed both AAD and mD2 in all the plotted range.

Please note that the comparing these numbers or the plot with that of the other methods measured by using different databases and categories is meaningless. For example, a large number of models in the “Other” category will bring down the performance figure, as the models retrieved from the “Other” category are counted as failures in our experiment. We need a standard test 3D model database with an accompanying category set.

Table 2. Comparison of retrieval performance among three shape descriptors by means of FT, ST, and NN figures.

Methods	FT	ST	NN
mD2	20.5%	31.4%	37.1%
AAD	24.0%	35.7%	43.1%
AMR-AAD	27.9%	39.9%	51.8%

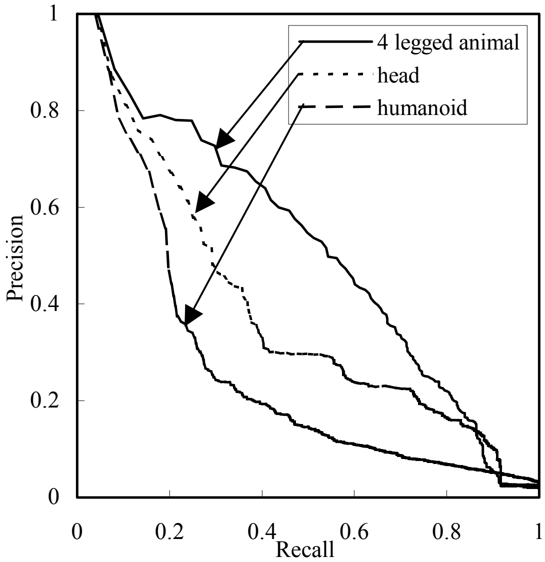


Figure 7. Significant performance variation exists among categories.

4.5. Retrieval examples

Figure 9 shows an example query results by using models of a car (82porsch.wrl) and an office chair (officechair.wrl) as queries. In each figure, the query model is positioned at the upper left corner, and top 20 matches are shown to the right in a left-to-right, top-to-bottom scan-line order. (The top-left model is the closest model to the query. In both Figure 9a and Figure 9b, the top match is the query model itself.)

Black polygons that appear in some of the models indicate that the surface normal vectors of these polygons are flipped compared to the others.

5. Summary and conclusion

In this paper, we proposed a new multiresolution approach to 3D shape similarity comparison of polygon shape models by using *3D alpha-shapes* [7]. To derive a multiresolution representation from a 3D polygon soup model, our method first converts the model into a point set model by stochastically sampling the model's surfaces. The method then reconstructs a set of 3D alpha-shapes by using multiple predetermined scale parameters, or *alpha* values. A multiresolution shape descriptor, an ordered set of single resolution shape descriptors, can then be derived. We employed the AAD shape descriptor [20] for the single-resolution shape descriptor. Our evaluation experiments showed that the proposed multiresolution method has significantly better retrieval performance than the single resolution AAD shape descriptor.

While the multiresolution approach is quite powerful, our current algorithm that employs the 3D alpha shapes is slow. We somehow must improve the computational efficiency of multiresolution shape descriptors in order to deal with a large database of 3D models. We intend to investigate better alternatives to the *Sobol*'s quasi-random number sequence for creating point set models and to compute the AAD shape descriptor. By using a better QRNS, we will be able to use fewer sample points, thereby reducing the computational costs. A candidate would be the *Niederreiter* sequence, which worked quite well in computing volume and surface area of solid models [5, 14].

We also intend to explore other (single-resolution) shape descriptors to be combined with the multiresolution shape-similarity comparison framework proposed in this paper.

Acknowledgements

We thank Prof. Shigeo Takahashi and anonymous reviewers for their valuable comments. This research has been funded, in part, by the grants from the *Ministry of Education, Culture, Sports, Sciences, and Technology of Japan* (No. 12680432), *Okawa Foundation for Information and Telecommunications*, and the *Artificial Intelligence Research Promotion Foundation*.

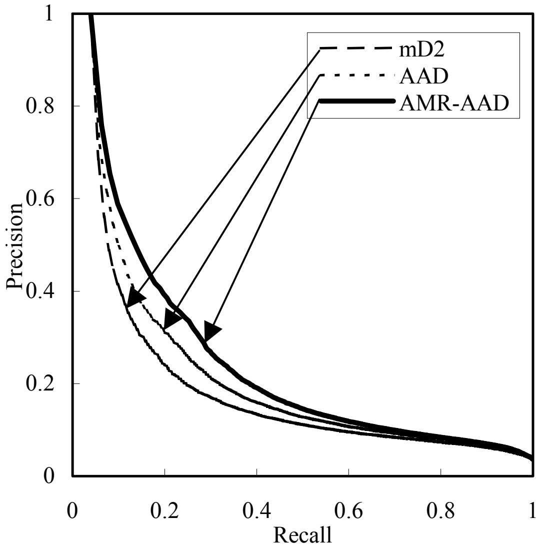
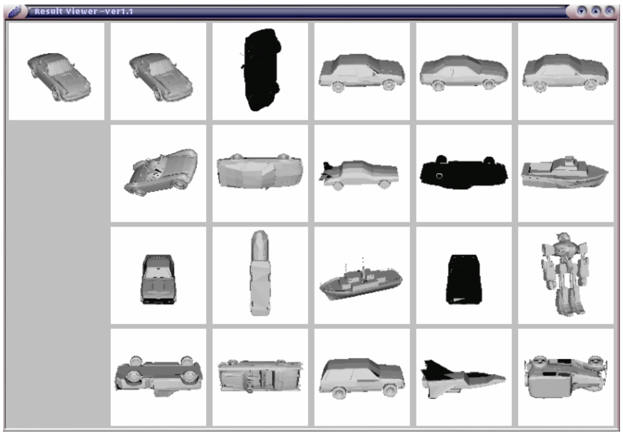
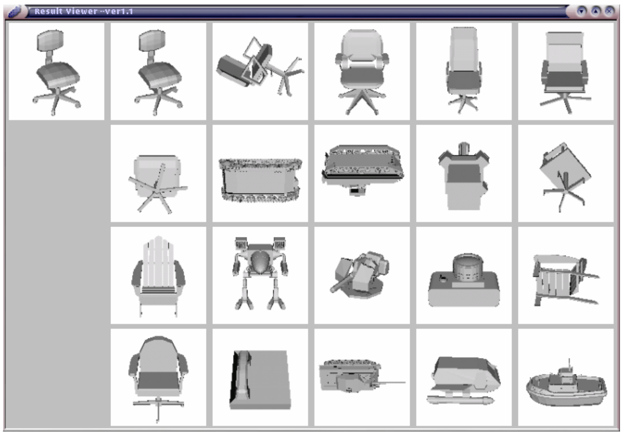


Figure 8. The recall-precision plot averaged over 1200 models in 35 categories.



(a) A car model (82porsch.wrl) is queried.



(b) An office chair model (officechair.wrl) is queried.

Figure 9. Query examples.

References

[1] M. Ankerst, G. Kastenmuller, H-P. Kriegel, T. Seidl, 3D Shape Histogram for Similarity Search and Classification in Spatial Databases, Proc. *Int'l Symp. Spatial Databases (SSD '99)*, Hong Kong, China, July 1999.

[2] Burt P.J., Adelson E.H., The Laplacian Pyramid as a Compact Image Code. *IEEE transactions on communications*, **31**(4), 1983.

[3] Ken Clarkson, A program for convex hulls,
http://www.netlib.org/voronoi/hull.html

[4] J. Corney, H. Rea, D. Clark, John Pritchard, M. Breaks, R. MacLeod, Coarse Filter for Shape Matching, *IEEE CG&A*, pp. 65-73, May/June, 2002.

[5] T. J. G. Davies, R. R. Martin, A. Bowyer, Computing Volume Properties using Low-Discrepancy Sequences, *Computing* [Suppl] **14**, pp. 55-72, 2001.

[6] Tony D. DeRose, Michael Lounsbery, Joe Warren. Multiresolution Analysis for Surfaces of Arbitrary Topological Type. Department of Computer Science and Engineering, University of Washington, Technical Report TR 93-10-05, October 29, 1993.

[7] Herbert Edelsbrunner, Ernst P. Mücke, Three-dimensional Alpha Shapes, *ACM TOG,* **13**(1), pp. 43-72, (1994)

[8] M. Elad, A. Tal, S. Ar., Content Based Retrieval of VRML Objects - An Iterative and Interactive Approach, Proc. *6th Eurographics workshop on Multimedia*, Manchester, UK., September 2001.

[9] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, D. Jacobs, A search engine for 3D models, *ACM TOG*, **22**(1), pp. 83-105, (January, 2003).

[10] H. J. A. M. Heijmans and J. B. T. M. Roerdink (eds.), Mathematical Morphology and its Applications to Image and Signal Processing, Kluwer Academic Publishers, Dordrecht, 1998 (Proceedings of the *ISMM'98*).

[11] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii. Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes. *Proc. SIGGRAPH 2001*, pp. 203-212, Los Angeles, USA. 2001.

[12] Takeo Igarashi, Hidehiko Tanaka, Satoshi Matusoka, Teddy: A Sketching Interface for 3D Freeform Design, Proc. *SIGGRAPH '99*, pp. 409-416, 1999.

[13] D. Keim, Efficient Geometry-based Similarity Search of 3D Spatial Databases, Proc. *ACM SIGMOD Int. Conf. On Management of Data*, pp. 419-430, Philadelphia, PA, 1999.

[14] X. Li, W. Wang, R. R. Martin, A. Bowyer, Using Low-discrepancy Sequences and the Crofton Formula to Compute Surface Areas of Geometric Models, *Computer Aided Design*, **35** (9), pp. 771-782, 2003.

[15] D. McWherter, M. Peabody, W. Regli, A. Shokoufandeh, Transformation Invariant Shape Similarity Comparison of Solid Models, Proc. *ASME DETC '2001*, September 2002, Pittsburgh, Pennsylvania.

[16] Patrick Min, John A. Halderman, Michael Kazhdan, Thomas A. Funkhouser, Early Experiences with a 3D Model Search Engine, Proc. *Web3D 2003*, pp. 101-112, Saint Malo, France, March 2003.

[17] S. Mukai, S. Furukawa, M. Kuroda, An Algorithm for Deciding Similarities of 3-D Objects, Proc. *ACM Symp. on Solid Modelling and Applications 2002*, Saarbrücken, Germany, June 2002.

[18] M. Novotni, R. Klein. A Geometric Approach to 3D Object Comparison. Proc. *Int'l Conf. on Shape Modeling and Applications 2001*, pp. 167-175, Genova, Italy, May, 2001.

[19] R. Ohbuchi, T. Otagiri, M. Ibato, T. Takei, Shape-Similarity Search of Three-Dimensional Models Using Parameterized Statistics, Proc. *Pacific Graphics 2002*, pp. 265-274, October 2002, Beijing, China.

[20] R. Ohbuchi, T. Minamitani, T. Takei, Shape-Similarity Search of 3D Models by using Enhanced Shape Functions, Proc. *Theory and Practice of Computer Graphics 2003* (TP.CG.03), pp. 97-104, Birmingham, U.K., June 2003.

[21] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin. Matching 3D Models with Shape Distributions. Proc. *Int'l Conf. on Shape Modeling and Applications 2001*, pp. 154-166, Genova, Italy, May, 2001.

[22] R. Osada, T. Funkhouser, Bernard Chazelle, and David Dobkin Shape Distributions, *ACM TOG*, **21**(4), pp. 807-832, (October 2002).

[23] E. Paquet and M. Rioux, Nefertiti: a Query by Content Software for Three-Dimensional Databases Management, Proc. *Int'l Conf. on Recent Advances in 3-D Digital Imaging and Modeling*, pp. 345-352, Ottawa, Canada, May 12-15, 1997.

[24] W. Regli, V. Cicirello, Managing Digital Libraries for Computer-Aided Design, *Computer Aided Design*, pp. 110-132, **32**(2), 2000.

[25] Jean-Luc Starck, Fionn Murtagh and Albert Bijaoui, *Image and Data Analysis: The Multiscale Approach*, Cambridge University Press, 1998.

[26] M. T. Suzuki, T. Kato, H. Tsukune. 3D Object Retrieval based on subject measures, Proc. *9th Int'l Conf. and Workshop on Database and Expert Systems Applications (DEXA98)*, pp. 850-856, IEEE-PR08353, Vienna, Austria, Aug. 1998.

[27] M. T. Suzuki, T. Kato, N. Otsu. A similarity retrieval of 3D polygonal models using rotation invariant shape descriptors. *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC2000)*, Nashville, Tennessee, pp. 2946-2952, 2000.

[28] R. C. Veltkamp. State of the art in shape matching. Technical Report UU-CS-1999-27, U-Utrecht, 1999.

[29] R. C. Veltkamp, M. Tanasek. Content-Based Image Retrieval Systems: A Survey. Technical Report UU-CS-2000-34, U-Utrecht, 2000.

[30] R. C. Veltkamp. Shape Matching: Similarity Measures and Algorithms, invited talk, Proc. *Int'l Conf. on Shape Modelling and Applications 2001*, pp. 188-197, Genova, Italy, May, 2001.

[31] D. V. Vranić, D. Saupe, and J. Richter. Tools for 3D-object retrieval: Karhunen-Loeve Transform and spherical harmonics. Proc. *IEEE 2001 Workshop on Multimedia Signal Processing*, Cannes, France, pp. 293-298, October 2001.

[32] T. Zaharia, F. Préteux, Three-dimensional shape-based retrieval within the MPEG-7 framework, Proc. *SPIE Conference 4304 on Nonlinear Image Processing and Pattern Analysis XII*, San Jose, CA, January 2001, pp. 133-145.

[33] T. Zaharia, F. Préteux, Shape-based retrieval of 3D mesh models, Proc. *IEEE ICME 2002*, Lausanne, Switzerland, August 2002.