Content Based 3D Shape Retrieval
A Survey

Ilknur Icke
email:iicke@gc.cuny.edu
http://www.cs.gc.cuny.edu/~iicke
The Graduate Center
City University of New York

365 Fifth Avenue
New York, NY 10016,USA

December 2004

Abstract

Large databases of 3-dimensional (3D) data are becoming available on the Internet, and in various domains such as Computer Aided Design (CAD), Molecular Biology (3D Protein Models), Computer Graphics, Medicine and Archeology to name a few. As the number and variety of the 3D models(3D shapes) continue to grow, there has been an increasing interest in applications to help people search in these large databases. Even though each model might have a file name or other textual data associated with it, most of the time such information will not be enough to fully describe what the model actually is. The solution is to use the content, namely the shape. In this paper, we give an overview of the recent research on Content Based 3D Shape Retrieval. First, general shape similarity and matching concepts are given. A review of the literature on 3D shape matching research within the content based retrieval framework is the main part of this paper. Then the issues affecting the shape retrieval performance issues are discussed.

1 Introduction

Large databases of 3-dimensional (3D) data are becoming available on the Internet, and in various domains such as Computer Aided Design (CAD), Molecular

Biology (3D Protein Models), Computer Graphics, Medicine and Archeology. Recent advances in laser scanning technology made it easier to reconstruct geometrically precise 3D models of objects. Stanford University's Digital Michelangelo [1] and Digital Formae Urbis Romae [2] projects are examples of such attempts to create archives of cultural heritage. Laser scanning is also helpful in generating realistic 3D models of human heads and bodies to be used in film industry and animation. Other kinds of domain specific archives are also available. For example, National Design Repository is an online database of CAD models [3] and Protein Data Bank[4] is an online database of 3D biological macromolecule structures. HUGO [5] is an anatomical 3D volume and surface data set based on the Visible Human Project. [6]

As the number and variety of the 3D models continue to grow there has been an increasing interest in applications to help people navigate through these large databases. Searching for a model in a large database of 3D models is not an easy task. Even though each model might have a file name or other textual data associated with it, most of the time such information will not be enough to fully describe what the model actually is. Instead of trying to label a model, a better approach is to let the model speak for itself, namely, using the content rather than the user assigned subjective textual information about the model.

3D models of most real life objects can be discriminated by their color, texture and shape information. Color and texture would not be useful for some kinds of models, such as the 3D Protein Models. Therefore, shape is the lowest common denominator in describing 3D data.

We all have an idea about the concept of shape but there is no universal definition. The most commonly cited definitions are the following :

- Merriam-Webster Dictionary[7]
  Shape(noun) :
  1. The visible makeup characteristic of a particular item or kind of item.
  2. A spatial form of contour.
  3. A standard or universally recognized spatial form.

- Kendall's definition [29]
  Shape is all the geometrical information that remains when location, scale and rotational effects are filtered out from the object.

Kendall's definition suggests that shape of an object is invariant to similarity transformations, for example, the 3D model of a car, should be considered as the same shape, even if it is rotated, scaled or moved to another location. Given two models, the intuitive way to determine if they are similar is to find

[1] `http://www.graphics.stanford.edu/data/mich`
[2] `http://www.formaurbis.stanford.edu/index.html`
[3] `http://edge.mcs.drexel.edu/repository/frameset.html`
[4] `http://www.rcsb.org/pdb`
[5] `http://www.viewtec.ch/meddiv/hugo\_e.html`
[6] `http://www.nlm.nih.gov/research/visible/visible\_human.html`
[7] `http://www.w-m.com`

correspondences between these two models and align them accordingly. The degree of alignment in the end would give an idea about the similarity of the shapes. This is known as the *shape registration problem* and a well-known method has been introduced by Besl and McKay [5]. The main application area of this technique is to align multiple views (e.g., 3D point clouds) of the same model for 3D model reconstruction. It is not an efficient method for the purpose of 3D model retrieval in large databases.

The approach in today's research on 3D model retrieval is to describe the model in a compact manner (either as feature vectors or structural descriptions like graphs) and compare these compact *descriptions* instead of the models themselves. Since shape should be invariant to rotation, translation and scaling either the descriptions should be invariant to similarity transformations or the whole database of 3D models should be put into a canonical coordinate system before any similarity matching can be done. This is known as the *pose normalization problem*.

This paper presents a survey of the emerging field of content based 3D shape retrieval. As we already established that shape is the lowest common denominator to describe the 3D data, we use the terms 3D shape, 3D model and 3D object interchangibly. Likewise, in literature, the phrases *3D Model Retrieval* or *3D Model Search Engine* refer to the same research area.

Tangelder and Veltkamp from Utrecht University, Netherlands evaluate the shape retrieval methods with respect to several aspects such as the ways shapes are represented, similarity/dissimilarity measures, retrieval performance, ability to perform partial matching, robustness and pose normalization requirements [47]. Iyer et al. from Purdue University, School of Mechanical Engineering give an overview of shape searching techniques including CAD specific methods as well [22]. Another survey that explains some of the existing techniques is given by Atmosukarto and Naval from National University of Singapore [2]. Also a Siggraph2004 course on 3D shape retrieval has been given by Funkhouser and Kazhdan from Princeton University, Department of Computer Science [14].

The organization of this document is as follows:

Section 2 gives an overview of 3D shape representation techniques. As there are different ways of 3D shape reconstruction(laser scanning, stereo vision based reconstruction, structure from motion), or modeling (for example CAD tools), ways to organize these data in digital environment differ as well. Representations for static and dynamic models (articulated or deformable) are presented but the rest of this paper reviews the similarity and matching methods for static shapes only.

Section 3 covers shape similarity and matching concepts.

Section 4 surveys different ways to describe the 3D shapes for the purpose of similarity matching and model retrieval. The methods are divided into two main categories: they either make use of the 3D model directly(model based) or work on a number of 2D projections of the 3D model (view based). Model based methods can be either purely geometric, structural or a combination of geometric and structural properties of the shapes. Geometric methods generate either global or local descriptions of the shapes.

Section 5 gives an overview of 3D shape retrieval performance measures and related issues that affect the performance.

2 Representing 3D Shapes in Digital World

Replicating the objects of the real world in digital environment has always been an interesting task for many applications. The quality of these models were limited by the capacity of the hardware and the software. Recent advances in hardware let people visualize and manipulate complex models easily. Modern scanning technologies also made it possible to generate geometrically precise models of objects. Besides the advances in hardware, modeling software (e.g., CAD tools) became more sophisticated with a lot more capabilities.

Since there are various ways of creating models of objects, the techniques to represent the data in digital environment vary as well. This section gives a brief overview of these techniques. As we have mentioned earlier, these are the representations of the shapes of the objects, therefore texture or color (photometric data) are not covered here.

This section mainly covers the methods to represent the 3D models that would be used as input data in a 3D shape retrieval system. Some of these representations are more commonly available than others, mostly because of the reasons related to the nature of the model generation process.

In a digital world, the primary task with a 3D model is to visualize it. Sometimes, it may be required to edit the model. The main concern is efficiency in storing and displaying the models. Different tasks would require different kinds of representations. For example, if the task is to recognize the objects in a scene, we might not need very detailed models of the objects. An overview of 3D model reconstruction, object recognition and related techniques are beyond the scope of this paper. Some of the related literature on these techniques are by Campbell and Flynn [7], Jain and Dorai [24], Bennamoun and Mamic [4] and Pope [41].

The shapes are divided into two main categories: static shapes and dynamic shapes. Static shapes are rigid shapes that do not change in time by deformation or articulation. For example, model of a coffee cup is a static shape, and the human face is a dynamic shape since its shape changes while speaking, smiling and so on. This paper focuses on the shape retrieval techniques for the static shapes, therefore the dynamic shape representations are going to be briefly mentioned only.

2.1 Static Shapes

There are two different paradigms in representing the objects : model based (object centered) or view based (viewer centered) methods. Model based methods directly make use of the 3D data, while view based methods store a number of 2D projections of the 3D model. This paper covers the representations that are directly used in the 3D shape retrieval literature. A more detailed review can be found in [20].

2.1.1 Model Based Representations

A 3D shape can be represented in different levels of abstraction. The first level is as a set of points in 3D space, this representation will be just raw data therefore it lacks a structure, but would be enough for visualization purposes. In 2D images, this corresponds to the pixels. The second level of abstraction is the boundary of the shape, in the case of 3D shapes, the boundaries are surfaces. In 2D this corresponds to curves. The third level of abstraction is to think of the shape in terms of the volume it occupies. In the case of 2D shapes, this corresponds to the area.

Point Based Representations These representations are rather unstructured and raw data. When viewed they seem as a shape all together.

Point Clouds **Definition** A point cloud is defined as a set of points $\mathbf{P} = \{\mathbf{p_1}, \mathbf{p_2}, ..., \mathbf{p_N}\}$ where $\mathbf{P} \epsilon \mathbf{R^3}$ and $\mathbf{p_i} = (x_i, y_i, z_i)^T$.

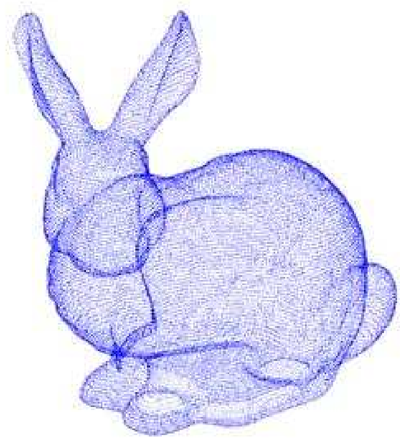

Figure 1: A 2D snaphot of Stanford Bunny point cloud

Range Images Range images are similar to intensity images in the sense they capture the shape from one point of view, but instead of the color information the pixel values carry the depth (or distance from the camera) information.

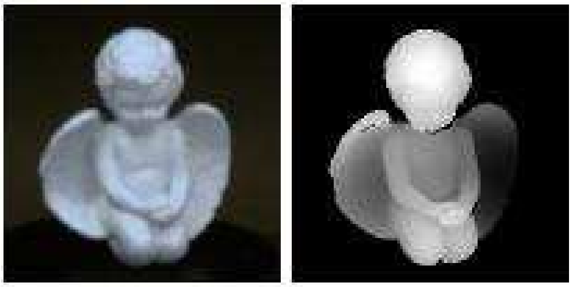

Figure 2: Angel image (intensity and range images)

This represenation is generally used in 3D model reconstruction where multiple range images of an object are merged. This is an application of 3D shape registration.

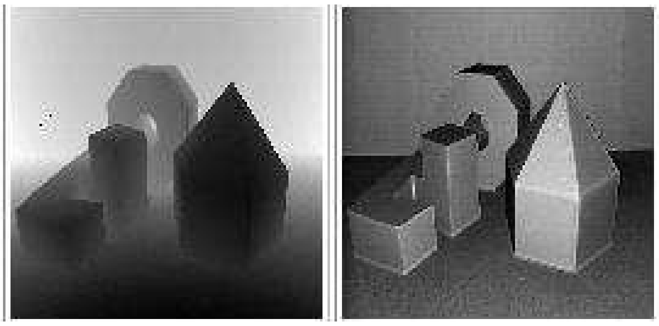

Figure 3: Polyhedral objects (range and intensity images)

In range images the encoding of depth(distance) may vary depending on the process that generates the images. For example, in figure 2.2, [8] the further away the objects from the camera, the darker the corresponding pixel values are. The opposite is true for the range image given in figure 2.3.[9]

Surface Representations A 3D shape can be represented in terms of its outer surfaces, just as a 2D shape can be represented based on its boundaries. This section will cover the mathematical models to represent the shapes as surfaces.

[8] http://marathon.csee.usf.edu/range/DataBase.html

[9] http://sampl.eng.ohio-state.edu/~sampl/data/3DDB/RID/minolta/angel.0699/index.html

Polygon Soups This type of representation is generally used by computer aided modeling tools. They are also called polygon soup models since the polygons might not be totally connected to cover a solid. In 3D model retrieval literature these are considered ill-defined models. A large number of 3D models available on the Internet are polygon soups.

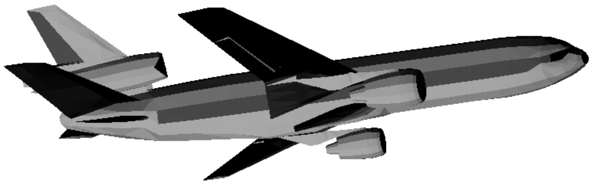

Figure 4: A CAD model as polygon soup

Polygon Meshes Polygon mesh is a very popular representation for 3D models because of its simplicity.

Definition A polygon mesh 3D model is defined by a pair of ordered lists:

$$\mathbf{M} = < \mathbf{P}, \mathbf{V} >$$

where $\mathbf{V} = \{\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_N}\}$ is the list of vertices and $\mathbf{v_n} = (x_n, y_n, z_n)^T$. $\mathbf{P} = \{\mathbf{p_1}, \mathbf{p_2}, ..., \mathbf{p_R}\}$ is the list of planar polygons and $\mathbf{p_r} = (\mathbf{v_{n,1}}, \mathbf{v_{n,2}}, ..., \mathbf{v_{n,k_r}})$. k_r is the number of vertices in polygon $\mathbf{p_r}$. If $k = 3$ for all $\mathbf{p_r}$ then the mesh is called a triangle mesh.

Unlike a polygon soup, a polygon mesh is a complete mathematical model which covers a whole volume.

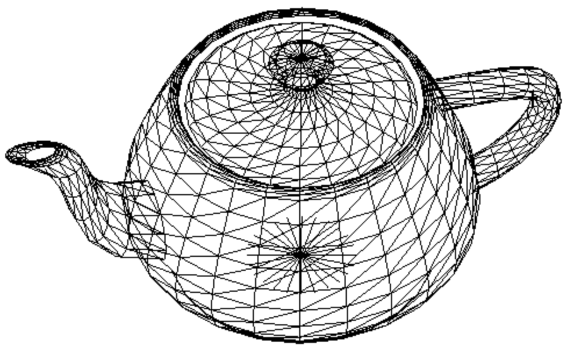

Figure 5: A teapot polygon mesh

Volumetric (Solid) Representations

Voxels Voxel is the minimum 3-D unit in a volume rendering which is equivalent to the pixel in a 2-D rendering. This is the simplest form of space subdivision based representation which is not memory efficient. This representation is generally used in medical applications.

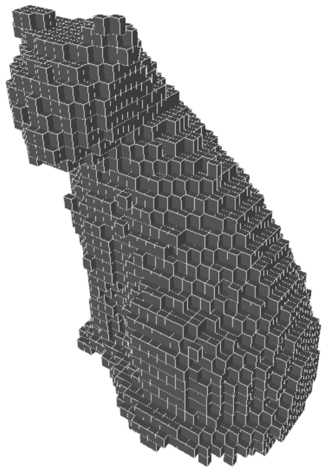

Figure 6: A voxelized cat model

Octree Octree is a space subdivison based representation in which a cubic space is recursively divided into smaller cubic volumes and an hierarchical data structure is built. The following figure shows how an octree is built for the given solid model.

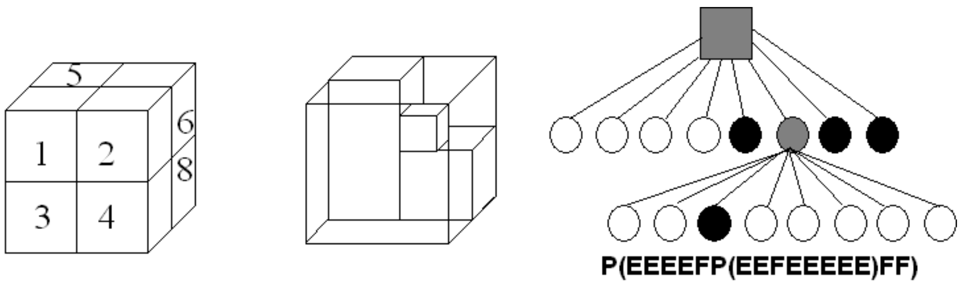

Figure 7: Octree Representation

The white nodes represent the subvolumes that are completely empty, black nodes represent subvolumes that are completely occupied, and gray nodes represent the subvolumes that are partially occupied. This is a more memory efficient representation compared to voxels.

2.1.2 View Based Representations

View based representations stem from the observation that similar 3D shapes look similar from the same viewpoints, therefore a number of views (2D projections) of objects could be used to represent the shapes. This is generally used for the purposes of object recognition. This section will briefly cover some of these techniques.

Silhouettes Silhouettes contain the boundary of a shape from one view point. In order to represent a 3D shape, a collection of silhouettes should be generated and stored. This can be seen as a more economical representation compared to model based representations.

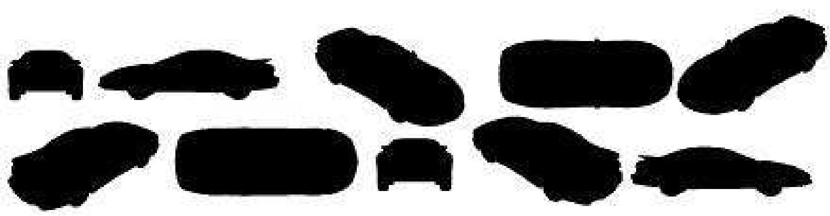

Figure 8: Silhouette images of a car

Common usage of this representation is object classification where matching is done between one view (silhouette) of a 3D shape and a database of objects represented as collection of silhouettes of models at hand. But the problem with this representation is that, in theory, different 3D shapes might have the same set of silhouette images.

Aspect Graphs 3D shapes look different when viewed from different viewpoints. For example, a cube looks like a square when viewed from the top. Based on this idea, the space of views can be partitioned into view classes or characteristic views. Within each class, the views share a certain property. A clustering algorithm might be used to generate the view classes.

A view class representation called an *aspect graph* was proposed by Koenderink and van Doorn in 1979 [30]. The nodes of the graph represent the aspects namely a class of views and the edges connect different nodes which have a certain change in aspect. These appearance changes from node to node are called visual events. Aspect graphs are complicated data structures therefore their usage is limited.

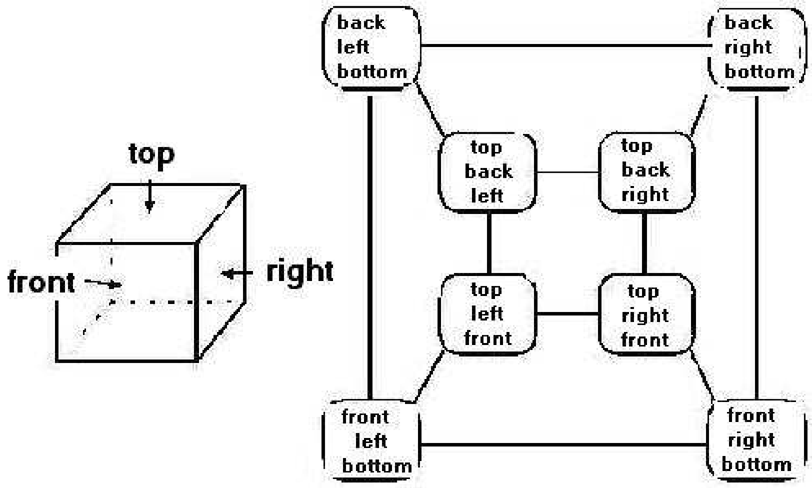

Figure 9: Aspect graph representation

2.2 Dynamic Shapes

In modeling and vision applications dynamic shapes are often encountered. These shapes may be articulated or deformable, thus they may change in time. There are various representations for this type of models. The following are some examples [42]:

- **Snakes: Active Contour Models**

  Fitting a deformable contour (snakes) to a given set of points is an optimization problem that can be cast as energy minimization subject to some constraints. Active contour models have been proposed by Kass, Witkin and Terzopoulos in 1987 [33]. They define the total energy as a sum of three components: internal contour energy by bending or streching the contour, image energy by the amount of fitness between the contour and image intensity or gradient and external energy due to the defined constraints.

- **Deformable Volumetric Models**

  Motion of the human heart has been modeled using tetrahedral volume elements that can deform based on the motion of the heart. This work has been done by Park, Metaxas and Axel [23].

- **Balloon Models**

  This is a deformable mesh representation in which the edges in the mesh are modeled as springs so that the entire mesh can deform to fit the shape either by streching or contracting. One such model is presented by Chen and Medioni [8].

3 Shape Similarity and Matching Concepts

Shape matching is an important concept in a number of applications like retrieval, recognition or registration. It is the process of determining how similar two shapes are. In general, matching is done by computing a distance in terms of similarity and small distance means small dissimilarity and large similarity.

Definition: Given a set of shapes $S = \{s_1, s_2, ..., s_N\}$, the *similarity distance* is defined as $d(s_i, s_j) : S \times S \rightarrow R^+ \cup 0$ where $s_i, s_j \epsilon S$. Function d may have some of the following properties:

- Identity: $\forall s_i \epsilon S$, $d(s_i, s_i) = 0$
- Positivity: $\forall s_i, s_j \epsilon S, s_i \neq s_j$, $d(s_i, s_j) > 0$
- Symmetry: $\forall s_i, s_j \epsilon S$, $d(s_i, s_j) = d(s_j, s_i)$
- Triangle Inequality: $\forall s_i, s_j, s_k \epsilon S$,$d(s_i, s_k) \leq d(s_i, s_j) + d(s_j, s_k)$
- Transformation Invariance: Given a transformation group G, $\forall s_i, s_j \epsilon S, g \epsilon G$, $d(s_i, g(s_j)) = d(s_i, s_j)$

Identity property means that a shape completely matches itself. Positivity property ensures that two different shapes never match completely.

Definition A distance function that has identity, positivity, symmetry and triangle inequality properties is called a *metric*.

Definition A distance function that has identity, symmetry and triangle inequality properties is called a *pseudo-metric*.

Definition A distance function that has identity, positivity and symmetry properties is called a *semi-metric*.

3.1 Classification of Shape Matching Problems

Given two shapes s_1, s_2 and a dissimilarity measure d, Veltkamp [50] gives the following classification of the shape matching problems:

- *Computation problem:* Let d be a transformation invariant dissimilarity function. Compute $d(s_1, s_2)$
- *Decision problem:* Let d be a transformation invariant dissimilarity function. Given a threshold value t, decide whether $d(s_1, s_2) < t$
- *Decision problem:* Given a threshold value t, decide whether there exists a transformation g where $d(g(s_1), s_2) < t$
- *Optimization problem:* Find the transformation g where $d(g(s_1), s_2)$ is minimum.

A number of shape matching applications can be posed based on this classification:

- *Shape based retrieval*
  Given a database of shapes $S = \{s_1, s_2, .., s_N\}$, and a query shape q, retrieve the shapes that are similar to q. This can be done in two ways:
  - Decision problem: Given a threshold value t, retrieve all the shapes where $d(q, s_i) < t$
  - Computation problem: Retrieve the top k shapes where $d(q, s_i)$ are minimum.
- *Shape recognition and classification*
  - Decision problem: Given a shape s and a model o, determine if $d(s, o)$ is sufficiently small.
  - Computation problem: Given a shape s, k classes of shapes and a representative shape for each class $r_1, r_2, ..., r_k$ find the class of r_i where $d(r_i, s)$ is minimum.
- *Shape alignment and registration*
  Optimization problem: Given two shapes $s1$ and s_2, find the transformation g such that $d(g(s_1), s_2)$ is minimum.

In 3D shape retrieval literature, the problem is generally posed as a computation problem as mentioned above. Given a query model, the system returns a number of most similar models in the database.

The ways in which the shapes are described for shape matching, guides the choice of similarity measure. An overview of the techniques used in 3D shape retrieval literature are given in the following sections. In this section, most commonly used similarity measures are presented. Veltkamp [50] gives an overview of shape matching within the computational geometry framework, that also includes similarity measures used in matching polygons and curves.

- L_p Norm (Minkowski Distance)

  This measure is used if some numerical descriptions based on the whole shape are extracted. These descriptions are in the form of fixed length vectors (feature vectors), so the similarity comparisons are made on these vectors.

  Definition Given two points $x, y \epsilon R^k$, the L_p distance is defined as:

  $$L_p = (\sum_{i=1}^{k} |x_i - y_i|^p)^{1/p}$$

  For $p \geq 1$, L_p distance is a metric.
  If $p = 1$, it is called L_1 norm or Manhattan distance or city block distance.
  If $p = 2$, it is called L_2 norm or Euclidean distance.
  L_p norm is not a transformation invariant dissimilarity measure.

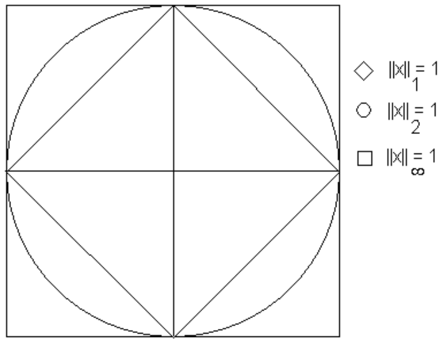

Figure 10: The plot of points in 2D satisfying $||x||_p = 1$

- Hausdorff Distance

  Definition Given two shapes represented by two sets of points:
  $X = \{x_1, x_2, ..., x_M\}$ and $Y = \{y_1, y_2, ..., y_N\}$ the Hausdorff distance between X and Y is defined as:

  $$H(X,Y) = max(h(X,Y), h(Y,X))$$

  where $h(X,Y) = \underset{x\epsilon X}{max}\ \underset{y\epsilon Y}{min}||\mathbf{x} - \mathbf{y}||$ and $||.||$ is usually Euclidean distance.

  The Hausdorff distance is a metric. However, it is not transformation invariant and also not robust against noise. The advantage of using this metric is that partial matching is possible.

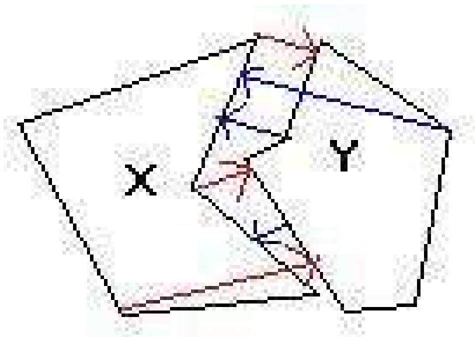

Figure 11: Visualization of Hausdorff distance

- Elastic Matching Distance

  Definition Let $A = \{a_1, a_2, ..., a_M\}$ and $B = \{b_1, b_2, ..., b_N\}$ be two finite sets of ordered contour points, and let f be a correspondence between all points in A and all points in B where the following condition holds: $\{\forall a_i, a_j \in A, a_i < a_j \Rightarrow f(a_i) \leq f(a_j)\}$

A stretch s is defined as:

$$s(a_i, b_j) = (a_i, f(a_i)) = \begin{cases} 1 & \text{if } f(a_{i-1}) = b_j \text{ or } f(a_i) = b_{j-1} \\ 0 & \text{otherwise} \end{cases}$$

The Nonlinear Elastic Matching Distance between A and B is defined as:

$$NEM(A, B) = \min_f \sum s(a_i, b_j) + d(a_i, b_j)$$

where $d(a_i, b_j)$ is the difference between the tangent angles at a_i and b_i.

A dynamic programming algorithm to compute this distance exists. Elastic Matching Distance is not a metric since it does not obey the triangle inequality.

- Earth Mover's Distance

  This is also known as transport distance.

  Definition Given two weighted point patterns
  $\mathbf{A} = \{(A_1, w(A_1)), (A_2, w(A_2)), ..., (A_M, w(A_M))\}$ and
  $\mathbf{B} = \{(B_1, w(B_1)), (B_2, w(B_2)), ..., (B_N, w(B_N))\}$
  where $A_i, B_i \in R^2$, the transport distance between $\mathbf{A}$ and $\mathbf{B}$ is the minimum amount of work it takes to transform $\mathbf{A}$ to $\mathbf{B}$.

3.2 Distance Functions for 3D Shape Matching

By definition, the shape of a 3D object is independent of any **translation, scaling and rotations** applied to it. Therefore, the distance function is desired to be *invariant* to these transformations. A distance function considering all possible transformations can be given as follows :

$$D(s_i, s_j) = \min_{g \in G} d(s_i, g(s_j))$$

where G is the group of transformations.

It is clear that, the distance function given above is not an efficient similarity measure for 3D shape matching. The following are two ways to define transformation invariant

Definition(Pose normalization):

Given a set of shapes $S = \{s_1, s_2, ..., s_N\}$, a metric $d(s_i, s_j)$ and a group of transformations G.

Let n be a many-to-one function where $\forall g \in G, s_i \in S, n(g(s_i)) = \hat{s_i}$ and $\forall s_i, s_j \in S, d(s_i, s_j) \sim d(\hat{s_i}, \hat{s_j})$.
Then

$$d(s_i, s_j) \sim d(\hat{s_i}, \hat{s_j}) = d(g(s_i), g(s_j))$$

In 3D shape matching, the group of transformations G contain any combination of translation, scaling and rotation.

The function n defined on this G is called a **pose normalization function**.

Definition(Invariant features):

Given a set of shapes $S = \{s_1, s_2, ..., s_N\}$, a metric $d(s_i, s_j)$ and a group of transformations G.

Let f^+ be a function where $\forall g \in G, s_i \in S, f^+(g(s_i)) = f^+(s_i)$ and $d(s_i, s_j) \sim d(f^+(s_i), f^+(s_j))$.

Then

$$d(s_i, s_j) \sim d(f^+(s_i), f^+(s_j)) = d(g(f^+(s_i), g(f^+(s_j))$$

The function f^+ is called an **invariant feature extraction function**.

3D Shapes in their representation forms are not well suited for matching. Therefore simplified descriptions (*shape descriptors*) capturing the significant features of the shapes are needed.

Definition(Shape descriptor generation):

Given a set of shapes $S = \{s_1, s_2, ..., s_N\}$, a metric $d(s_i, s_j)$.

Let f be a function where $\forall s_i, s_j \in S, d(s_i, s_j) \sim d(f(s_i), f(s_j))$.

The function f is called a **shape descriptor generation function**.

If f is also invariant to translation, scaling and rotations then it is called an **invariant shape descriptor generation function**.

Shape descriptors could be numerical or structural.

Numerical shape descriptors generate a mapping $\mathbf{X} \rightarrow \mathbf{R^n}$ where $\mathbf{X}$ is the space of original shape representations.

Definition (3D Shape Based Retrieval Problem) : Given a database of 3D shapes $S = \{s_1, s_2, ..., s_N\}$ and a query shape q, retrieve the shapes that are similar to q.

Solution:

(Decision Problem) Given a threshold similarity value t, retrieve all the shapes where $d(f(q), f(s_i)) < t$.

(Computation Problem) Retrieve the top k shapes where $d(f(q), f(s_i))$ are minimum where d is a distance function, preferably a metric. where f is a **shape descriptor generation function**.

If f is not invariant to translation, scale and rotations then the shapes have to be pose normalized first.

4 3D Shape Matching for Retrieval

This section gives an overview of the 3D shape retrieval literature. A great variety of methods have been proposed within the recent couple years. The 3D model representation that is widely worked on are the polygonal representations, simply because they are widely used computer graphics and CAD applications. Given a polygon model, it is also possible to create a volumetric model via a procedure known as voxelization. Therefore, almost all of the research done on

3D shape retrieval considers polygon (mesh or polygon soup) models or voxelized models as inputs.

Given a dataset of various 3D models, the main approach is to create simpler representations for these models so that similarity comparisons between the models could be made in a computationally efficient manner. This is a very important issue, especially if the database is large, since the retrieval results should be returned after a reasonable querying time. In 3D shape retrieval literature, these simplified representations that are extracted from the inital models are called *shape descriptors*. As the name suggests, shape descriptors should be descriptive enough to be able to discriminate similar and dissimilar shapes and as compact as possible so they are not aimed for visualization purposes. Shape descriptors can be either numeric (e.g., feature vectors, histograms and such) or structural (e.g., graphs).

There are two main approaches to shape matching. First approach is to use the 3D model as it is and generate shape descriptors based on the geometrical and/or topological properties of the model. This is called the model based approach. Some model based techniques require that the models to be put in a canonical coordinate system before processing. This is known as pose normalization and it is a necessary step if the shape descriptor is not invariant to similarity transformations (translation, rotation and scaling). Translation invariance is achieved by translating the center of mass of the model to the origin, scale invariance is achieved by scaling all the models to the same dimensions. Rotation invariance is a more complicated issue. The general technique used is the principal component analysis (PCA) which computes the principle axes of the model, then the model is rotated so that its principle axes would match the predefined canonical coordinate system. There are various problems with this technique. First of all, PCA does not guarantee the correct ordering of the principle axes which may result in improper alignments of some models. Also, for polygon meshes, the area of individual polygons may differ and this would affect the computation of the principal axes of the model. A weighted PCA algorithm has been proposed in order to handle this situation.

The alternative approach to model based approach is the view based approach in which a number of 2D projections of the models are used in order to generate a shape descriptor for shape matching. Generally, the 2D shape descriptors are used in the view based case. There has been extensive research on 2D shape matching and Zhang and Lu [61] give a recent overview of the 2D shape descriptors. The challenge in view based techniques is that one has to capture enough number of views to be able to cover all possible aspects of 3D model.

4.1 Model Based Techniques

Model based techniques for 3D shape retrieval make use of the 3D shape at hand. There are two main approaches in model based 3D shape retrieval. Some techniques consider the geometric properties of the shapes only and these properties can be either global or local. Other techniques consider structural properties of the shapes like the holes or the components they contain.

4.1.1 Geometric Techniques

These methods exploit the quantitative properties of the shapes. Some examples of these properties are volume, aspect ratio, surface area, curvature or other kinds of numerical descriptions extracted from the shape. These properties could be characterising the shape either globally or locally. Most global methods are computationally efficient but they do not allow partial matching, in contrast to this, local methods are not computationally efficient though they can be used in partial matching as well.

Global Shape Descriptors Global methods aim to capture the characteristics of the shape as a whole. There are a great number of approaches proposed in order to describe the global shape of an object. This section presents some of these approaches grouped by the main ideas that they use.

Features Given a shape, maybe the first approach one thinks of is to describe the shape in terms of some measurements (features) that would help discriminate the shape from other shapes easily. Examples of such features are volume, surface area or statistical moments computed based on the shape surfaces or the volume. Since these features can only describe a shape very roughly they are not very discriminative but they can be used as an initial filtering stage in 3D shape retrieval applications since they can be computed very easily and efficiently.

Elad et al. [13] propose a moments based method that works polygon meshes. They define an approximated moment according to the following formula:

$$\hat{M}[k1, k2, k3] = \frac{1}{N} \sum_{j=1}^{N} x_j^{k_1} y_j^{k_2} z_j^{k_3}$$

The authors first sample N points from the surface of the model. Invariance to translation is done by using centralized first moments. Scale and rotation invariance is achieved using a singular value decomposition technique on a 3×3 matrix that is formed by calculating the second moments on the sample points representing the model. After the normalization step, they calculate moments up to some pre-specified order and form a feature vector using these values for each 3D model. The similarity measure is the Euclidean distance between these feature vectors.

Zhang and Chen [58] describe efficient ways to compute the volume, surface area and moments of 3D models represented as polygon meshes.

Feature Distributions These methods are not directly based on some measurements on the shapes but the distributions of those measurements. This section will cover some of these techniques.

Osada et al. [38] present a method that computes a shape distribution sampled from a shape function measuring global geometric properties of the shape. Therefore shape similarity matching can be made in terms of probability distribution comparisons. They define various shape functions based on the global geometry of the shape:

- A3: Measures the angle between three randomly chose points on the surface of a 3D shape.
- D1: Measures distance between a fixed point and one randomly chosen point on the surface. They use the centroid of the boundary of shape as the fixed point.
- D2: Measures the distance between two randomly chosen points on the surface.
- D3: Measures the square root of the area of the triangle formed by three randomly chosen points on the surface.
- D4: Measures the cube root of the volume of the tetrahedron formed by four randomly chosen points on the surface.

These functions are easy to compute and they are invariant to rotations and translations. In order to generate shape distributions using these functions, they sample N points on each shape distribution generated by the functions above. Then they create histograms for B equi-width bins. These histograms are the approximations to the distributions. Shape similarity matching reduces to histogram matching which can be done in various ways among which are, the Minkowski norm, Kolmogorov-Smirnov distance, Kullback-Leibler divergence, earth mover's distance, Bhattacharyya distance, χ^2 statistic.

The authors implement eight easy to compute similarity measures. Let a, b be the two shapes to be compared and f_a, f_b be the probability distribution functions (pdf) for these shapes approximated by the histograms as explained above and let $\hat{f}_a, \hat{f}_b$ be the cumulative distribution functions. The similarity measures are:

- χ^2 : $D(a,b) = \int \frac{(f_a - f_b)^2}{f_a + f_b}$
- Bhattacharyya Distance : $D(a,b) = 1 - \int \sqrt{f_a f_b}$
- Minkowski (L_p) norm on pdf: $D(a,b) = \int |f_a - f_b|^p)^{\frac{1}{p}}$ where p=1,2,∞
- Minkowski (L_p) norm on cdf: $D(a,b) = \int |\hat{f}_a - \hat{f}_b|^p)^{\frac{1}{p}}$ where p=1,2,∞

Since the models are not scale normalized, a normalization procedure is applied to the distributions. They report that the D2 function gave the best results in their experiments. Some extensions to D2 method are given by Obhuchi et al. [35] and Ip et al [21]

Obhuchi et al. [37] describe a method that computes a number of statistics along the principle axes of the 3D models. The method work on the polygon mesh models. First, they align the models with respect to their principle axes and for each of these axes they compute the following histograms along the axes: (1) the moment of inertia about the axis, (2) the average distance to surfaces from the axis, (3) the variance of distance to surfaces from the axis. This procedure generates 9 feature vectors which are concatenated to form one feature vector per model. They use Euclidean distance and Elastic-matching distance for similarity comparisons. Their experiments show that the method performs well on rotationally symmetric models only.

Spatial Maps These methods aim to capture the spatial organization of the shapes. The 3D space is divided into sections and the distributions of model points or other features for each section are computed. The neighbourhood relations between these sections are also considered during the similarity comparison. The following is an example technique.

Ankerst et al. [1] describe a method that has two main parts. The first part introduces shape histograms based on the discrete representations of the models. The second part is the definition of a quadratic distance function. Initially the shapes are aligned to their center of mass. A set of uniformly sampled surface points are used for computing the histograms.

They propose three methods to create the shape histograms. Each method defines a different decomposition of the space. These are: a shell model in which the 3D shape is decomposed into concentric shells around the center point, a sector model in which the 3D shape is decomposed into sectors that emerge from the center point of the model and a spiderweb model as a combination of the two.

The shell model is rotationally invariant while the other two models are not. The authors argue that Euclidean distance is not a good similarity measure for their purposes because it does not consider the relationships between the components of a feature vector. Since in this case, the components reflect the spatial relationships of point distributions with respect to the space decomposition model, they use a quadratic form distance function defined as follows :

$$d_A^2 = (x - y) \cdot A \cdot (x - y)^T = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij}(x_i - y_i)(x_j - y_j)$$

where N is the dimension of the feature vector, namely the number of bins in the space decomposition model. A is the similarity matrix where the components a_{ij} represent the similarity of the components in the feature vector. As it can be seen, if A is the identity matrix then this distance is the Euclidean distance. Using this distance function, it is easy to assign similarity weights depending on the neighbourhood relationships between the bins.

Integral Transforms and Special Functions Techniques from Calculus and Analysis have been applied to various fields such as digital image processing and signal processing. In the 3D shape retrieval literature, some techniques utilizing integral transforms (by utilizing their coefficients) and some special functions have been reported. This section briefly overviews some of these techniques.

Definition A general integral transform is defined as :

$$F(s) = \int_{\alpha}^{\beta} K(s,t) f(t) d(t)$$

where the function K(s,t) is called the kernel function. Depending on the kernel function the integral transforms take different names. Some common transforms are Hough transform, Fourier transform, Wavelet transform, Radon transform and Laplace transform.

3D shape retrieval techniques utilizing these transforms apply the discrete form of the transforms since the data is discrete and make use of the coefficients in a way to create a feature vector as the shape descriptor.

Zaharia and Preteux [57] describe a 3D shape retrieval system based on the 3D Hough Transform(3DHT) as the shape descriptor on the polygon meshes. They pose normalize the 3D models. Because of the limitations of the PCA algorithm, they compute 3DHTs on the model over all possible orderings of the coordinate axes. This gives them a set of 48 3DHTs which is called the Optimized 3DHT(O3DHT) since it achieves rotation invariance. In order to compare two models, they calculate L_1 and L_2 distances on each of these 48 corresponding 3DHT shape descriptors and the distance between these two models is chosen as the minimum distance among them.

Vranic and Saupe [53] present a discrete 3D Fourier transform(3DDFT) based method to generate descriptors on polygon mesh models. Initially, they pose normalize the models. Rotation invariance is achieved using a variant of the PCA algorithm. Then 3D Fourier transform is applied to voxelized models. They generate a feature vector that contains the real valued coefficients of the transform. They experiment with L_1 and L_2 distances as similarity measure.

Paquet et al. [40] propose Wavelet transform based shape descriptors for 3D model retrieval. Daras et al. [9] describe an algorithm that uses the 3D Radon transform and a similarity measure that is based on the L_1 distance.

Some special functions have also been used in the context of 3D shape retrieval. Kazhdan et al. [28] give an algorithm that is based on the spherical harmonics. They voxelize the polygon mesh models and intersect the voxelized models with concentric spheres, describing each sphere as a spherical function in terms of how much of the model falls into that sphere. The next step is the harmonic decomposition (frequency decomposition) of these spherical functions. They sum up the harmonics within each frequency and generate a 2D map of L_2 distances indexed by the radius of the sphere and the frequency. This shape descriptor is invariant to rotations around the center of mass of the model. They also report that spherical harmonics decomposition can be applied to any shape descriptor which is defined as a function on the voxel grid to achive rotation invariance. Novotni and Klein [34] describe a method that uses 3D Zernike moments as the shape descriptors. The shape descriptors they create using 3D Zernike moments are also rotation invariant.

Information Theory Approach Page et al. [39] propose a method that measures the shape complexity of 3D model surfaces. Using the curvature information, they compute the entropy of the curvature and call this measure *shape information.* Their motivation is that, intuitively, a pin for example, is much more complicated than the sphere, so there should be a way to define this quantitively. They exploit the concept of entropy that is defined as follows for the discrete case:

$$H = -\sum_{i=1}^{M} p_i \; log_2 p_i$$

Uniformly sampling points on the polygon mesh and calculating an estimate to the Gaussian curvature on these points distributed into M equi-width bins, the authors estimate the probability density function (pdf) for the curvature on a shape. Using the definition given above, they calculate the entropy H based on the M bins they create. This gives a scalar metric that tells how complex a 3D shape is in terms of the Gaussian curvature.

The authors had claimed that a sphere should be the simplest shape in terms of curvature complexity. To support their claim, the shape information metric described above gives 0 for the sphere. Their experiments confirm that models with a variety of curvature values have more complexity than the symmetrical models or the models with repeating curvature values.

Volumetric Difference These techniques are based on the observation that different shapes occupy the volume in different ways. This can not be captured by a simple volume difference technique. Two shapes might have the same total volume, though they are not similar. Because of the nature of the comparison, all shapes have to be pose normalized before processing. Some of these techniques are presented here.

Kaku et al. [26] propose a method based on OBBTree data structure that was proposed by Gottschalk. After pose normalization they represent each 3D model in their database in terms of a binary tree where each node represents the center of an oriented bounding box (OBB). They define a similarity measure based on the sum of differences of the corresponding nodes on the trees. They also keep the aspect ratios of the original models to define another similarity measure based on the aspect ratios. The final similarity measure is a weighted combination of these two similarity measures. The authors report better retrieval performance compared to the D2 technique proposed by Osada et al [38].

Leifman et al. [32] propose an octree based volumetric difference measure. Given two octrees representing two models, the volume difference D at the root node is calculated in a recursive bottom-up manner. This is a slow algorithm compared to other algorithms in which feature vectors are compared.

Ichida et al. [19] present an interactive user interface for 3D shape retrieval. Their system is called ActiveCube in which the user builds the query shapes using the 5cm side cubes. The system automatically recognizes the shape that the user builds in real time. The models in the database and the query shape are represented in voxels. The system performs similarity matching by finding the intersections between the voxels representations of the models.

Projection (Morphing) to a Canonical Shape The idea behind the projection based methods is that the energy required to morph a shape into another could be a measure of similarity between these two shapes. In the context of 3D shape retrieval, every model in the database is morphed into a canonical shape (e.g., a sphere) and the amount of energy required to do this morphing is used as a descriptor for that model and during retrieval the descriptors are compared. There are different ways to define this energy. This section presents some of these methods.

Leifman et al. [32] describes a sphere projection algorithm. They first normalize the pose of the models in their database to ensure invariance to similarity transformations. The energy that is required to morph a model into its bounding sphere with radius R, is defined as $\int_{dist} \vec{F} \cdot \vec{dr}$ where $\vec{F}$ is the applied force and dist is the distance between the object surface and the bounding sphere. The force is assumed to be constant for all points on the surface and along the distance it's applied. Therefore, the energy is proportional to the distance between the sphere and the model's surface.

They sample points on the sphere and calculate two distances based on these points. The first distance(d_1) is the minimum Euclidean distances from the sphere to the model and the second distance(d_2) is the distance from the model to the sphere that is calculated as follows: each point p on the model is represented in spherical coordinates (α, θ, r), for each model point, a point on the sphere that has the most similar α, θ is found. After this correspondence search, for each sample point on the sphere, there is a set of corresponding points on the model surface. Then the distance is the average of the distances($|R - r|$) from the sphere point to its corresponding model points. The final distance (d) is calculated as either as the average or concatenation of d_1 and d_2.

The authors report experimental results on a database of 1068 random objects gathered from the Internet. 258 objects were manually categorized into 17 classes (people,missiles,cars and such). Their experiments give better results compared to the shape moments [13] and shape distributions [38] on most classes except for those ones that do not have a common global shape simply because this method captures the global shape properties only.

Yu et al. [56] propose a similar method which is also based on the idea of morphing the models into a sphere. They generate a distance map from to object to the bounding sphere. Pose normalization techniques are used to put the models in a canonical coordinate system before calculating the distance map. They also apply the Fast Fourier Transform (FFT) on these maps in order to handle the possible misalignments even after the pose normalization. The similarity measure they use is a weighted normalized Euclidean distance of the Fourier transformed maps.

The authors report good experimental results on a database of 52 models divided into 34 categories. No comparisons with other methods are given.

Weighted Point Sets These methods generate a set of points from the shape. The points are weighted in some manner. Different similarity measures have been proposed to match these point sets.

Tangelder and Veltkamp [46] propose three different ways of generating a weighted point set, given a pose normalized 3D polygon model, which is placed in a 3D grid. Each non-empty grid cell contains one salient point. The selection of the salient point and its weight is done in various ways: (1) pick the point in each cell that has the highest Gaussian curvature, and assign the curvature value as the weight for the point, (2) pick the area-weighted mean of the vertices in the cell as the point, and a measure of facet normal variation as the weight, (3) compute the center of mass of all vertices in the cell and assign 1 to weight.

The similarity measure they use is a variation of the earth mover's distance, unlike that measure the proposed measure satisfies the triangle inequality. The authors report better results compared to the shape distribution methods proposed by Osada et al. [38] on their own database.

Local Shape Descriptors These methods consider the local properties of the shape around the neighbourhoods of points on the surface. Curvature is an example local property which has also been used within the context of the global methods. In that context, such local properties are treated as a a collection and all together they form a global descriptor for the shape.

The methods that are presented here do not merge the local properties to form a global descriptor, therefore they are suited for partial matching. Also, they can provide better descriptions of shapes because they capture more detailed information about the shape at the expense of computation time. These methods have been mostly used for object recognition in cluttered environments and surface registration problems. Some of them have also been applied to 3D shape retrieval problem. These methods do not require prior pose normalization of the models.

Johnson and Hebert [25] present the spin images method. They rotate planes around the surface normal at chosen points and create a set of 2D histograms that capture the information on the point distribution around that point with respect to those planes. Then these histograms are used for shape matching. They describe an object recognition algorithm that works on cluttered scenes.

De Alarcon et al. [10] use spin images method in 3D shape retrieval. For each 3D model represented as polygon mesh, they generate a large number of spin images, then using a self organizing map (SOM) algorithm, they generate a reduced set of spin images per model. Furthermore, they cluster these spin images using k-means clustering algorithm in order to provide an indexing mechanism on their database. The authors report experimental results on a small database.

Yamany et al. [55] present a method that captures the surface curvature information at certain points on the surface and generates images called surface signatures for each selected point. The authors use this method for surface registration. They find at least three corresponding points on two models by matching the surface signatures and then recover the parameters to the similarity transformation that would put these surfaces in alignment.

Kortgen et al. [31] present a 3D shape matching algorithm that introduces 3D shape contexts as an extension to the 2D shape contexts concept that has been proposed by Belongie et al. [3]. They generate histograms around the N sampled points on the surface. For a sampled point a histogram contains the relative coordinates of the remaining $N-1$ points. Depending on the size of the sampled point set, this method can be considered more or less local. The binning algorithm they use decomposes the space into shells or sectors. Shape matching is done by comparing the shape contexts to find the corresponding points on the models.

4.2 Structural and Topological Techniques

Geometrical properties of 3D shapes do not tell a lot about the semantics of the shape. They describe the shape either locally or globally but they can not describe the shape in terms of the components of it. Also topologically different shapes can not be efficiently discriminated using geometric methods. For example, a torus and a sphere are different types of shapes, it is easier to discriminate between them using topological methods. Also, sometimes geometrically different but topologically similar shapes need to be classified in the same group. For example, different types of tables could be grouped together. Tables with rectangular or circular top, or four or three legs are topologically similar though they may not be geometrically similar.

Structural descriptions of shapes are more intuitive and easy to interpret but the difficulty is that matching is not a computationally efficient process compared to the geometrical methods, but their advantage over geometrical methods is that they allow partial matching which is difficult with the methods that describe the shapes in terms of their global geometry.

Surface Penetration Map Yu et al. [56] present a method that extracts topological information from a model by morphing it into a sphere. This is called a surface penetration map. The idea behind this method is the following: When an imaginary ray is shot from the center of the bounding sphere to the model surface it may penetrate one or more surfaces depending on the topology and concavity of the model. The bounding sphere is divided into sections and for each section the mean number of surfaces penetrated by the rays is computed. The authors do not report any comparions with other methods.

Graph Structures Hilaga et al. [18] describe a method that is called topology mathing. They create multiresolutional Reeb Graphs (MRG) for comparing 3D models. A Reeb graph is a skeleton determined using a continous scalar function defined on an object. The authors create the Reeb graphs utilizing the geodesic distance distribution as the continous function. This method is well suited for articulated shapes.

Tung and Schmit [48] describe a Reeb graph based algorithm which is augmented with geometrical properties such as volume and curvatures. In the context of human body matching, they argue that without geometric information, an arm could be matched to a leg because they are topologically equivalent.

Sundar et al. [44] present a skeletal graph approach for comparing 3D models for retrieval. Their method encodes the topological and geometric information about the models. There are various ways to create a skeleton representation of a 3D model. The authors compute the skeleton of a volumetric 3D model using a parameter-based thinning algorithm. The skeleton graph also contains geometric information on each part of the model such as the radius.

Relational Structures Each 3D object can be thought of a combination of primitives and a set of relations that describe the relationships among these primitives. Each primitive is described in terms of some geometric attribute like area, radius and so on. A general framework named relational matching for this kind of relational descriptions is given by Vosselman [51]. Also, Haralick and Shapiro [16] describe a consistent-labeling framework based on a relational distance definition.

4.3 View Based Techniques

Humans are claimed to utilize the appearance (or views) of the 3D shapes as well as the 3D geometrical or structural properties of them. Based on the idea that if two shapes are similar they should look similar from all viewing angles, some researchers study the problem of 3D shape retrieval using the visual similarity of the shapes. This section will cover some of these techniques in which a number of views of the 3D models are used in order to generate a descriptor for the models for the purposes of similarity matching.

Chen et al. [11] propose a light field based method. A light field is a five dimensional function representing the radiance at a given 3D point in a given direction. Given a 3D model made invariant to translation and scaling, they create a light field containing 10 sillhouettes of the model using 10 uniformly distributed points of views on an approximating bounding sphere. A combination of Zernike moments (region based descriptor) on the area and Fourier transform (contour based descriptor) of the boundary is used as the 2D descriptor for each silhouette. A set of 10 light fields resulted from 10 different rotations of the sphere is stored for each 3D model. Let a, b be two models to be compared.

The similarity measure is defined as follows:

$$D(a,b) = \underset{1 \leq i \leq 10}{min} \sum_{k=1}^{10} d(I_{a_{ik}}, I_{b_{ik}})$$

where $I_{a_{ik}}$,$I_{b_{ik}}$ are the 2D descriptors computed on the silhouettes and the distance d is L_1 norm.

The authors report experimental results in which they compare their method to the 3D spherical harmonics based method by Funkhouser et al [15]. They report better retrieval performance at the cost of much more processing time.

Obhuchi et al [36] propose a method which works on polygon soup 3D models. The models are made invariant to translation and scaling. Then they compute $N = 42$ depth buffer rendered images (a kind of range image) of the model. These set of images discretely cover all possible view aspects of the model. Then for each image a Fourier transform based descriptor is computed (2D descriptor). The set of these 42 descriptors comprises the multiple oriented shape descriptor for the 3D model. Let a, b be two models to be compared. The similarity measure is defined as follows:

$$D(a,b) = \frac{1}{N} \sum_{i=1}^{N} \underset{1 \leq j \leq N}{min} d(I_{a_i} - I_{b_j})$$

where I_{a_i}, I_{b_j} are the computed 2D descriptors and the distance d is L_1 norm. Since there is no ordering to the rotations, the similarity measure compares all possible pairings and picks the one with the minimum L_1 distance to contribute to the sum over all the 42 views.

5 3D Shape Retrieval Performance and Related Issues

In previous sections, we have covered the literature on the techniques used in shape based 3D model search applications. This section presents some aspects of the problem related to the performance of these systems in more detail and discuss the literature on each articular topic.

The outline of this section is as follows:

Section 5.1 overviews the measures for retrieval performance in the 3D model search systems.

Section 5.2 presents the techniques that consider subjective measures in retrieval. Most systems use the shape based (low-level) features for similarity matching but there is also semantic aspects of shapes and people do not always agree on similarity of different shapes. Based on this observation, a number of techniques have been developed to incorporate the user preferences into the similarity measures.

Section 5.3 presents the problem of selecting the best shape descriptor to use depending on the query.

5.1 Performance Measures and Benchmarking

In most 3D shape retrieval applications, the results are evaluated on the basis of how closely it agrees with the predefined classifications made by the system designers. Since the databases used by different systems vary as well as the classifications of the models in those databases, a unified framework to compare the different matching algorithms is needed. Princeton Shape Benchmark [43] is an effort to fill this void. It provides a test database with different types of classifications available along with the tools to compare the retrieval performance.

If a shape matching algorithm computes a distance between two shapes which is positive and small when the shapes are similar and larger when the shapes are not similar, a number of common performance measures can be applied to compare the algorithm with others provided that they work on the same 3D Model database.

Given a shape matching algorithm and a database of 3D models ($M = \{m_1, m_2, ..., m_N\}$), a distance matrix containing the distances between the pairs of models can be calculated. For any model $q \in M$, the list of k most similar models can be retrived from the distance matrix.

The following are the qualitative means to asses the performance of a 3D shape retrieval algorithm:

- Images of best matches sorted in descending order of similarity.
- Precision-Recall plot
- Distance Image
- Tier Image

5.2 Subjective Retrieval

In most 3D model search applications shape based features are extracted from the models, these are called low-level features because they do not capture the meaning of the shapes. Humans have the notion of similarity not only based on shape but also the functionality of the models, these are called the semantic features. Also each person might have a different measure of model similarities. A successful search engine should be able to adapt to the users' preferences. This section overviews a number of methods proposed for this purpose.

Suzuki et al. [45] describe a technique that creates an object feature space (OFS) and a user preference space (UPS) and a mapping between these two spaces. At feature extraction stage, they take polygon meshes and consider the vertices only and then they find the normalized bounding cube for the model and subdivide the cube into unit cells. Finally, the normalized number of vertices in each cube contributes to the feature vector for the model.

The rest of the algorithm is as follows:

1. Pick a subset of models in the database (study set) and ask the users to rate the similarity of these models. Create a matrix of similarity ratings for each user.

2. Perform dimensionality reduction using Multidimensional Scaling (MDS) on the rated similarity matrices created in previous step. This is the user preference space.

3. For the other models that were not in the study set, predictions should be made. Using multiple regression analysis create a mapping between the object feature space and each user's preference space.

Elad et al. [12] propose an iterative refinement algorithm that would let the users mark the relevant and irrelevant results to a query causing the distance measure to change adaptively. The features generated in this study are normalized moments and the similarity measure is weighted Euclidean distance.

User feedback is used to modify the weights in the distance measure so that the relevant matches become closer and irrelevant matches become further away with respect to the modified similarity distance. A Support Vector Machine (SVM) algorithm is used for training the weights of the distance measure. This way, for each user the system learns that user's subjective similarity measures between the models returning the models which are relevant according to that user's preferences.

Zhang and Chen [59, 60] present a method based on active learning concept to incorporate semantic features in retrieval process. The low-level features they generate are volume-surface ratio,moment invariants and fourier transform coefficients.

The system uses 53 pre-defined attributes like 'car', 'body', 'airplane' and so on. For each object, there is a list of probabilities that the object has the corresponding attribute. During training process, randomly sampled subset of models are presented to the person designated as the 'annotator'. The annotator assigns 0 or 1 to the probabilities for each object having a certain attribute. This process is called 'hidden annotation'. Since it is not possible to manually annotate all models, the rest of the probabilities are estimated. The authors use a biased kernel regression technique to estimate these probabilities for the models that were not annotated. The estimation is biased meaning that if an object is far from an annotated object it should not be affected by that specific annotation as a protection against the annotation propogating too far in the feature space.

The next step is to pick those models in the database that the system is most uncertain about and then present them to the annotator for further annotation. This is done using a knowledge gain measure and the aim is to reduce the amount of uncertainty in the database.

The retrival process uses a weighted distance measure of low-level feature similarity and the semantic similarity measure based on the calculated probabilities of model having the attributes. The performance of the system improves as the number of annotated models increases.

5.3 Shape Descriptor Selection

Various ways of describing the 3D shapes for matching and retrieval have been presented in the previous sections. Performance measures and benchmarking techniques presented above provide a common framework to compare the performance of individual shape descriptors.

In this section, the shape descriptor selection problem is presented as a feature subset selection problem within the pattern recognition context. Here, each shape descriptor can be thought of one feature and a combination of these features can be used together for shape retrieval. The problem is to decide what combination of these features gives the best results for retrieval.

This section covers two techniques presented in literature related to shape descriptor selection.

Vandeborre et al. [49] describe a method that generates three kinds of shape descriptors (features) on polygon mesh models. These features are: a curvature index which consists of a histogram of the principal curvatures of each face of the mesh, a histogram of distances between the faces (distance index) and a histogram of the volumes based on each face (volume index). All these features are invariant to Euclidean transformations. They use the L_1 norm as the similarity measure. Their model database is classified into object types like airplanes, cars, fish, chess pieces and so on.

The authors define two different ways to combine the shape descriptors:

Let the rank of an object in each result set be R_c, R_d and R_v using the curvature index, distance index and volume index shape descriptors respectively. N is the number of models retrieved in each query. Let F be a real number to represent the degree of each retrieved model's relevance to the query model with respect to that particular feature combination.

- OR method

$$F = 1 - \frac{(R_c - 1) * (R_d - 1) * (R_v - 1)}{N^3}$$

- MEAN method

$$F = \frac{R_c + R_d + R_v}{3N}$$

These two methods return F values between 0 and 1. Then the best N matches can be selected based on their F values.

Their experiments show that combining the shape descriptors give better retrieval results compared to using any of the three shape descriptors alone.

Bustos et al. [6] describe an entropy impurity measure based assessment for the feature selection problem. The database contains 1,838 3D models and 292 of them are pre-classified, for instance, 'car models','planes','sea animals' and so on. The classified models are used as queries and the models from the same class are considered relevant to a given query model. The authors use the L_1 norm to measure the similarity between the feature vectors representing the models.

The effectiveness of a retrieval process is measured by how coherent the result set is. The models returned as a result of the query should be of the same class as the query model. Although this may not be the case most of the times, some types of features might give better results than the others, or a combination of certain features would also perform better. The motivation here is that no single feature extraction technique is expected to give the best results for all kinds of queries. For example, the authors state that one feature extraction technique worked best on car models while another gave the best results on sea animal models.

The authors implement 15 different feature extraction techniques reported in literature that describe the models as feature vectors. The common property of all these methods is that they describe the global shape of the 3D shapes in the database. Table 4.1 lists these methods and the references in which they were introduced.

Method	**Reference**
Depth buffer	Heczko et al. [17]
Voxel	Heczko et al. [17]
Silhouette	Heczko et al. [17]
Volume	Heczko et al. [17]
Shading	Vranic and Saupe [54]
3D harmonics	Funkhouser et al. [15]
Complex function on the sphere	Vranic and Saupe [54]
Rays with spherical harmonics	Vranic and Saupe [54]
Cords	Paquet et al. [40]
Moments	Paquet et al. [40]
Shape distribution with D2	Osada et al [38]
3D FFT	Vranic and Saupe [53]
Ray based	Vranic and Saupe [52]
Rotational invariant	Kato et al. [27]
Shape spectrum	Zaharia and Preteux [57]

Table 1: Feature extraction methods used in Bustos et al.

The authors employ the well-known entropy impurity measure to estimate the performance of a feature extraction method given a query model presented as a feature vector. Based on their experiments, the authors state that the entropy impurity measure gave better results compared to Gini and the misclassification impurity.

Using the entropy impurity measure they develop two methods: query dependent selection of the best feature vector and query dependent combination of the feature vectors.

Let U be the universe of 3D models, and M be a finite set of models (the database) where $M \subset U$ and each model $m \in M$ belongs to one of the N classes

$c_1, c_2, ..., c_N$ and $U = \bigcup_{n=1}^{N} c_n$

Let $q \in U$ be a query model. Given a feature extraction function f, R_f^q is a list of models sorted in ascending order by d(f(q),f(r)), where d is the distance measure which is the L_1 norm and q is the query model and r is a retrieved model.

Let $P_k(c_n, R_f^q)$ be the fraction of models at the first k positions of R_f^q that are in class c_n.

- *Entropy impurity measure for selecting the best feature extraction function*

  The *k-entropy impurity* of a feature extraction function f given the query model q is:

  $$i(f, q, k) = -\sum_{j=1}^{N} P_k(c_n, R_f^q) log_2(P_k(c_n, R_f^q))$$

  The k-entropy impurity is 0 if all of the first k models in the result set are of one class, the maximum impurity value is reached when the number of different classes returned in the result set is maximum. Based on the calculated k-entropy impurity measure, the best feature extraction method given the query model q is selected using the following formula:

  $$arg \min_{1 \leq t \leq T} i(f_t, q, k)$$

  where $F = \{f_1, f_2, ..., f_T\}$ is the set of feature extraction functions.

- *Entropy impurity measure for combining the feature extraction functions*

  Instead of selecting the best feature extraction function f with respect to a query model q, a combination of different feature extraction functions can be selected. The authors use the k-entropy impurity measure derived above to weigh the feature extraction functions in a combination of such functions. The functions with lower impurity receive more weight. Based on the k-entropy impurity measure a new distance between the query model q and an object $o \in U$ is derived as follows:

  $$\delta(q, o) = \sum_{t=1}^{T} \frac{1}{1 + i(f_t, q, k)} \frac{d_t(q, o)}{dmax_t}$$

  where $i(f_t, q, k)$ is the k-entropy impurity given a feature extraction function f_t and the query model q. $dmax_t$ is the maximum distance($L1$ norm) between q and any model in the database using the feature extraction function f_t. $d_t(q, o)$ is the distance between q and a retrieved model o using the function f_t.

  $\delta(q, o)$ is used as the distance measure when sorting the list of retrieved models with respect to their relevance.

The authors use Precision(P) and Recall(R) graphs to compare the results of single feature extraction method used for all queries to the best feature extraction method selected using the k-entropy impurity measure for each individual query model. Likewise, a comparison is made in the case where a combination of feature extraction methods are used per query mode instead of a single feature extraction method for all queries. Reported improvement in retrieval effectiveness is almost %30 when the combination method is used to combine a small set of good feature extraction functions.

The problem with this approach is that the database of objects needs to be pre-classified manually since the classes of each model should be known beforehand for the calculations to be made. In the case of an unclassified database, the classification problem has to be solved first. If the number of classes of models in the database is not known, clustering can be used as an unsupervised classification technique. But this is not straightforward, because there may be different ways to group the objects, and these groupings maybe subjective. For example, purely shape based clustering may put irrelevant models in one group, like a missile and a pen. Therefore other information could be needed in the grouping process, this could be based on the functionality of the model or any other related textual information besides the shape.

References

[1] Mihael Ankerst, Gabi Kastenmüller, Hans-Peter Kriegel, and Thomas Seidl. 3d shape histograms for similarity search and classification in spatial databases. In Ralf Hartmut Güting, Dimitris Papadias, and Frederick H. Lochovsky, editors, *Advances in Spatial Databases, 6th International Symposium, SSD'99, Hong Kong, China, July 20-23, 1999, Proceedings*, volume 1651 of *Lecture Notes in Computer Science*, pages 207–226. Springer, 1999.

[2] I. Atmosukarto and P. Naval. A survey of 3d model retrieval systems. not published, N/A 2003. not published.

[3] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24):509–522, April 2002.

[4] M. Bennamoun and G. J. Mamic. *Object recognition: fundamentals and case studies.* Springer-Verlag New York, Inc., 2002.

[5] P. J Besl and N. D. MacKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[6] B. Bustos, D. Keim, D. Saupe, T. Schreck, and D. Vranić. Using entropy impurity for improved 3d object similarity search. In *Proc. IEEE International Conference on Multimedia and Expo (ICME'04)*, 2004.

[7] Richard J. Cambell and Patrick J. Flynn. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding*, (81):166–210, 2001.

[8] Yang Chen and Gérard Medioni. Description of complex objects from multiple range images using an inflating balloon model. *Computer Vision and Image Understanding: CVIU*, 61(3):325–334, 1995.

[9] P. Daras, D. Zarpalas, D. Tzovaras, and M.G. Strintzis. Shape matching using the 3d radon transform. In *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004*, pages 953–960, september 2004.

[10] De-Alarcon, Pascual-Montano PA, and JM Carazo. Spin images and neural networks for efficient content-based retrieval in 3d object databases. In *CIVR*, 2002.

[11] Yu-Te Shen Ding-Yun Chen, Xiao-Pei Tian and Ming Ouhyoung. On visual similarity based 3d model retrieval. In *Computer Graphics Forum (EUROGRAPHICS'03)*, volume 22, pages 223–232, September 2003.

[12] M. Elad, A. Tal, and S. Ar. Content based retrieval of vrml objects-an iterative and interactive approach. *Eurographics Multimedia Workshop*, pages 97–108, 2001.

[13] Michael Elad, Ayellet Tal, and Sigal Ar. Directed search in a 3d objects database using svm. Technical report, HP Laboratories, Israel, 2000.

[14] Thomas Funkhouser and Michael Kazhdan. Shape based retrieval and analysis of 3d models. Siggraph2004 Course 15, 2004.

[15] Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. A search engine for 3d models. *ACM Trans. Graph.*, 22(1):83–105, 2003.

[16] Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision*. Addison-Wesley Longman Publishing Co., Inc., 1993.

[17] M. Heczko, Keim, D. D., Saupe, and D. V. Vranic. Verfahren zur hnlichkeitssuche auf 3d-objekten. In *Datenbank Spektrum Zeitschrift fr Datenbanktechnologie*, volume 2, pages 54–63, 2002.

[18] Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, and Tosiyasu L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 203–212. ACM Press, 2001.

[19] Hiroyasu Ichida, Yuichi Itoh, Yoshifumi Kitamura, and Fumio Kishino. Interactive retrieval of 3d virtual shapes using physical objects. In *IEEE Virtual Reality*, 2004.

[20] I. Icke. A tutorial on 3d shape representations. Technical report, CUNY, 2004.

[21] Cheuk Yiu Ip, Daniel Lapadat, Leonard Sieger, and William C. Regli. Using shape distributions to compare solid models. In *Proceedings of the seventh ACM symposium on Solid modeling and applications*, pages 273–280. ACM Press, 2002.

[22] N. Iyer, K. Lou, S. Janyanti, Y. Kalyanaraman, and K. Ramani. Three dimensional shape searching : State-of-the-art review and future trends. *Computer Aided Design*, 2004.

[23] A. Young J. Park, D. Mataxas and L. Axel. Deformable models with parameter functions for cardiac motion analysis from tagged mri data. *IEEE Transactions on Medical Imaging*, 15:278–289, 1996.

[24] Anil J. Jain and Chitra Dorai. 3d object recognition: Representation and matching. *Statistics and Computing*, (10):167–182, 2000.

[25] A.E Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, May 1999.

[26] Keitaro Kaku, Yoshihiro Okada, and Koichi Niijima. Similarity measure based on obbtree for 3d model search. In *International Conference on Computer Graphics, Imaging and Visualization (CGIV'04)*, volume I, pages 46–51, july 2004.

[27] Toshikazu Kato, Motofumi T. Suzuki, and Nobuyuki Otsu. A similarity retrieval of 3d polygonal models using rotation invariant shape descriptors. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC2000)*, pages 2946–2952, 2000.

[28] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 156–164. Eurographics Association, 2003.

[29] D.G. Kendall, Barden D., Carne T.K., and Le H. *Shape and Shape Theory.* Wiley Series in Probability and Statistics, 1999.

[30] J. J. Koenderink and A. J. van Doorn. The internal representation of shape with respect to vision. In *Biological Cybernetics*, volume 32, pages 211–216, 1979.

[31] M. Kortgen, G.J Park, M. Novotni, and R. Klein. 3d shape matching with 3d shape contexts. *The 7th Central European Seminar on Computer Graphics*, April 2003.

[32] G. Leifman, S. Katz, A. Tal, and R. Meir. Signatures of 3d models for retrieval. *4th Israel Korea Bi-National Conference on Geometric Modeling and Computer Graphics*, pages 159–163, 2003.

[33] A. Witkin M. Kass and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

[34] M. Novotni and R. Klein. 3d zernike descriptors for content based shape retrieval. *Solid Modeling*, 2003.

[35] Ryutarou Ohbuchi, Takahiro Minamitani, and Tsuyoshi Takei. Shape-similarity search of 3d models by using enhanced shape functions. In *Proceedings of the Theory and Practice of Computer Graphics 2003*, page 97. IEEE Computer Society, 2003.

[36] Ryutarou Ohbuchi, Masatoshi Nakazawa, and Tsuyoshi Takei. Retrieving 3d shapes based on their appearance. *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 39–45, 2003.

[37] Ryutarou Ohbuchi, Tomo Otagiri, Masatoshi Ibato, and Tsuyoshi Takei. Shape-similarity search of three-dimensional models using parameterized statistics. In *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, page 265. IEEE Computer Society, 2002.

[38] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4):807–832, October 2002.

[39] D. L. Page, A. F. Koschan, J. K. Paik, and M. A. Abidi. Shape analysis algorithm based on information theory. In *Proceedings of the International Conference on Image Processing*, volume I, pages 229–232, 2003.

[40] E. Paquet, A. Murching, T. Naveen, A. Tabatabai, and M. Rioux. Description of shape information for 2-d and 3-d objects. In *Signal Processing: Image Communication*, volume 16, pages 103–122, 2000.

[41] A. R. Pope. Model-based object recognition: A survey of recent research. Technical report, Univ. of British Columbia, 1994.

[42] Linda G. Shapiro and George C. Stockman. *Computer Vision*. Prentice Hall, 2001.

[43] P. Shilane, M. Kazhdan, P. Min, and T. Funkhouser. The princeton shape benchmark. *SMI*, 2004.

[44] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *Shape Modeling International, 2003*, 2003.

[45] Motofumi T. Suzuki, Toshikazu Kato, and Hideo Tsukune. 3d object retrieval based on subjective measures. In *Proceedings of the 9th International Workshop on Database and Expert Systems Applications*, page 850. IEEE Computer Society, 1998.

[46] J. Tangelder and R. Veltkamp. Polyhedral model retrieval using weighted point sets. *Int. Journal of Image and Graphics, 3(1), pp. 209-229 (2003).*, 2003.

[47] Johan W. H Tangelder and Remco C. Veltkamp. A survey of content based 3d shape retrieval methods. *Shape Modeling Conference*, 2004.

[48] T.Tung and F.Schmitt. Augmented reeb graphs for content-based retrieval of 3d mesh models,. In *International Conference on Shape Modeling and Applications (SMI'04)*, pages 157–166, 2004.

[49] Jean-Philippe Vandeborre, Vincent Couillet, and Mohamed Daoudi. A practical approach for 3d model indexing by combining local and global invariants. In *1st International Symposium on 3D Data Processing Visualization and Transmission*, pages 644–647, 2002.

[50] R.C Veltkamp. Shape matching: Similarity measure and algorithms. In *Proceedings Shape Modelling International*, pages 188–197, 2001.

[51] G. Vosselman. *Relational Matching*. Lecture Notes in Computer Science, vol. 628, Springer Verlag., 1992.

[52] D. V. Vranic and D. Saupe. 3d model retrieval. In *Proceedings Spring Conference on Computer Graphics 2000(SCCG2000), Budmerice, Slovakia*, may 2000.

[53] D. V. Vranic and D. Saupe. 3d shape descriptor based on 3d fourier transform. In *Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services(ECMCS 2001),Budapest, Hungary*, pages 271–274, september 2001.

[54] D. V. Vranic and D. Saupe. Description of 3d-shape using a complex function on the sphere. In *Proceedings IEEE International Conference on Multimedia and Expo, Lausanne, Switzerland*, pages 177–180, August 2002.

[55] Sameh M. Yamany and Aly A. Farag. Free-form surface registration using surface signatures. In *Proceedings of the International Conference on Computer Vision-Volume 2*, page 1098. IEEE Computer Society, 1999.

[56] M. Yu, I. Atmosukarto, W. K. Leow, Z. Huang, and R. Xu. 3d model retrieval with morphing-based geometric and topological feature maps. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[57] T. Zaharia and F. Preteux. Hough transform-based 3d mesh retrieval. In *Proceedings of SPIE 4476 on Vision Geometry X, San Diego, USA*, august 2001.

[58] C. Zhang and T. Chen. Efficient feature extraction for 2d/3d objects in mesh representation. *CIP*, 2001.

[59] C. Zhang and T. Chen. Indexing and retrieval of 3d models aided by active learning. In *Proc. of ACM Multimedia*, pages 615–616, 2001.

[60] C. Zhang and T. Chen. An active learning framework for content based information retrieval. Technical report, CMU, 2002.

[61] D. S. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, 2004.