3

3D Indexing and Retrieval

Stefano Berretti, Mohamed Daoudi, Alberto Del Bimbo, Tarik Filali Ansary, Pietro Pala, Julien Tierny, Jean-Phillippe Vandeborre

3.1 Introduction

This chapter introduces the problem of 3D retrieval by shape descriptors and the criteria needed to evaluate the 3D retrieval algorithms. The role of shape descriptors is to represent the original data in a very short way. Intuitively, this means that the index should be invariant to some geometric transformations of the object (translation, rotation, scaling), and should have a certain robustness to noise.

This chapter also details different criteria commonly used to compare 3D shape descriptors. We distinguish two main criteria: *performance criteria* that evaluate quantitatively and objectively the performances of the 3D shape descriptors, and *algorithmic criteria* that evaluate the algorithmic properties of the 3D shape descriptors.

The use of 3D image and model databases throughout the Internet is growing in both number and size. Indeed, the development of modeling tools, 3D scanners (Figure 3.1), 3D graphic accelerated hardware, Web3D, high-quality PDAs (Personal Data Assistants) and even cellular phones with a fast CPU are powerful enough to visualize 3D models interactively. Recently,

3D Object Processing: Compression, Indexing and Watermarking J.-L. Dugelay, A. Baskurt & M. Daoudi

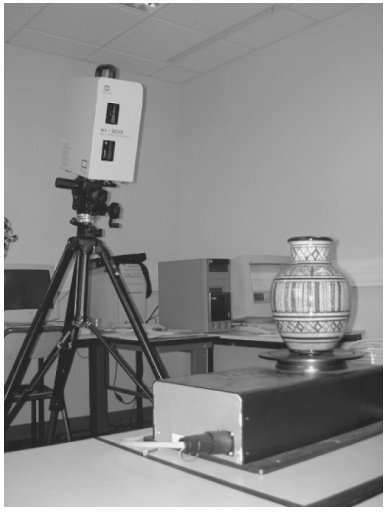

Figure 3.1 A 3D scanner

Acrobat 3D software proposed to insert and to publish 3D designs from major computer-aided design (CAD) applications in Adobe PDF documents.

Exploiting the information contents of digital collections poses several problems. In order to create value out of these collections, users need to find information that matches certain expectations, a notoriously hard problem due to the inherent difficulties of describing visual information content. Nowadays, people facing the problem of finding multimedia information are typically using text-based search engines. The content of a multimedia database is described using every day or – more or less – technical words depending on the usage of the database. Hence, this kind of search engine relies on human operators, with a certain expertise in the domain concerning the multimedia database, who are manually describing the multimedia content using keywords or annotations. The end-user is also experimenting with the problem of language level depending on his or her technical skills and expertise in the domain. Moreover, textual descriptions of multimedia content are inherently subjective and consequently are not a reliable solution to the problem of multimedia data indexing and retrieval.

In recent years, as a solution to these problems, many systems have been proposed that enable effective information retrieval from digital collections of images and videos (Del Bimbo 1999; Stamou and Kollias 2005).

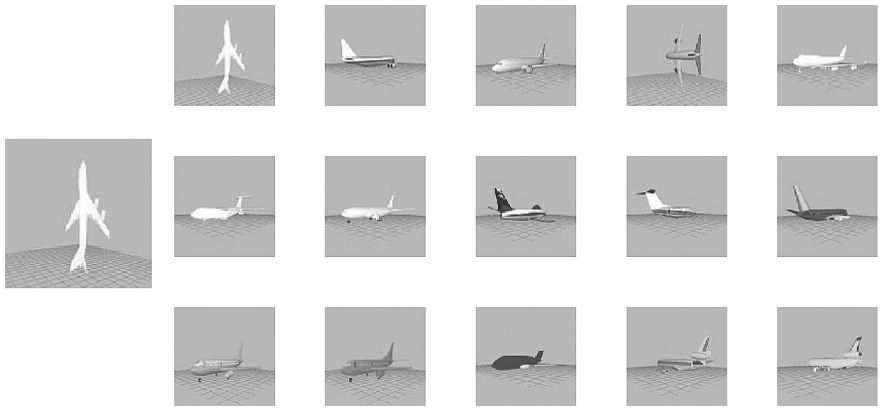

Figure 3.2 Example of a similarity search on a database of 3D models, showing a query object and a set of relevant retrieval results

However, solutions proposed so far to support retrieval of images and videos are not always effective in application contexts where the information is intrinsically 3D. A similarity metric has to be defined to compute the visual similarity between two 3D models, given their descriptions.

Figure 3.2 illustrates the concept of content-based 3D retrieval. The query object is a 3D model of an aircraft (left side of Figure 3.2). The system is expected to retrieve similar 3D models from the database as shown on the right side of Figure 3.2.

Recent advances in 3D scanner acquisition technology and 3D graphics rendering have boosted the creation of 3D model archives for several application contexts. These include:

- **Games/entertainment**: The 3D models are used to enhance realism in entertainment applications. Reuse and adaptation of 3D models by similarity search in existing databases is a promising approach to reduce production costs.
- **Medicine**: In medicine, the detection of similar 3D organ deformation can be used for diagnostics.
- **CAD**: Technicians and engineers in manufacturing companies need to exploit the large CAD mechanical parts database during the design of an automobile, to update the documentary corpus, to exploit engineering data and to create spare parts from original vehicle parts. Presently, many of

these steps demand human intervention for manual and visual inspection. Consequently, these processes are time consuming and very expensive. To improve productivity it is important to develop CAD search algorithms so as to automate significant parts of the process. Indeed, when a new product is designed, it can be composed of many small parts that fit together to form the product. If some of these parts are similar to one of the standard parts already designed, then the possible replacement of the original part with the standard part can lead to a reduction in production costs.

- **3D-face recognition**: Automatic face recognition has been actively researched in recent years, and various techniques using ideas from 2D image analysis have been presented. Although significant progress has been made, the task of automated, robust face recognition is still a distant goal. The 2D image-based methods are inherently limited by variability in imaging factors such as illumination and pose. An emerging solution is to use laser scanners for capturing the surfaces of human faces, and use these data to perform face recognition. Such observations are relatively invariant to illumination and pose, although they do vary with facial expressions. As the technology for measuring facial surfaces becomes simpler and cheaper, the use of 3D facial scans will be increasingly prominent.
- **Cultural heritage**: One particular interest is the possibility of exploiting 3D-based technologies to enable monitoring, cataloguing and remote fruition (periodic acquisition and comparison to evidence for deformation of masterpieces due to pollution, inadequate preservation, etc.).
- **Bioinformatics**: A 3D structural comparison and structural database searching of proteins play important roles. In many cases, merely comparing the amino acid sequences of the proteins cannot provide sufficient information required by the biologist. In particular, one cannot detect the similarity of two remotely homologous proteins by sequence comparison alone. Instead, a comparison of their 3D structures needs to be made in order to determine their similarity since the 3D structures are better preserved than the sequences throughout evolution. Indeed, there are now more than 25 000 protein structure files in Protein Data Bank[1] (PDB), with an additional 100 added every week – hence the increasing necessity for protein structural retrieval.

A variety of retrieval methods have been proposed that enable the efficient querying of model repositories for a desired 3D shape, many of which use a 3D model as a query and attempt to retrieve models with matching shape

[1] http://www.pdb.org/.

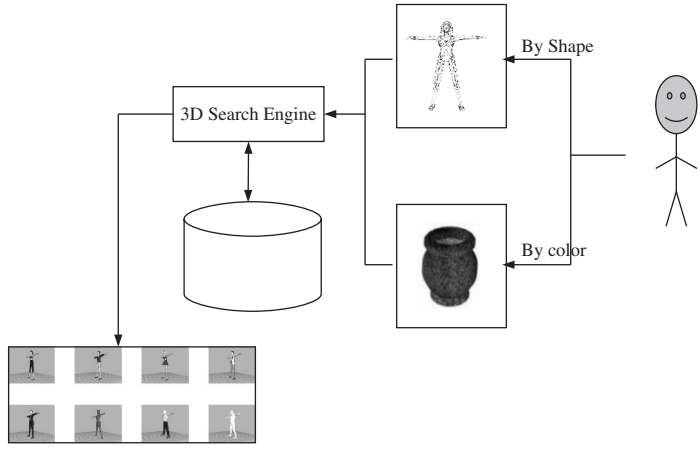

Figure 3.3 Examples of visual query

from the database. As above, an example of such an application is shown in Figure 3.2. The user specifies an aircraft as a query model (left). The system then compares the query to every model in the database, returning pointers to the models that are most similar (right).

In practice, the visual descriptor (shape, color) (Figure 3.3) is incorporated into the search engine as shown in Figure 3.4. In this chapter, we will restrict our discussion to an analysis of shapes to study their contribution in indexing. In a pre processing step the shape descriptor of each model in the database is computed (*offline step*). Then, at run-time, a query is presented to the system, its shape descriptor is computed (*online step*), the query descriptor is compared to the descriptors of the models in the database (3D search engine), and the database models with descriptors that are most similar to the query descriptor are returned as matches.

Figure 3.4 shows an overview of the 3D retrieval system:

- A subsystem for the manual annotation of the 3D data; in general, all the concepts which cannot be extracted manually.
- A subsystem for the extraction of low-level features; 3D processing is used to extract automatically descriptors of 3D data.
- An interface for browsing the 3D model database.
- A graphical query interface which permits retrieval of 3D models by a drawing or sketch, by a photo or by a 3D model. Figure 3.5 shows an example of the 3D retrieval graphical interface.

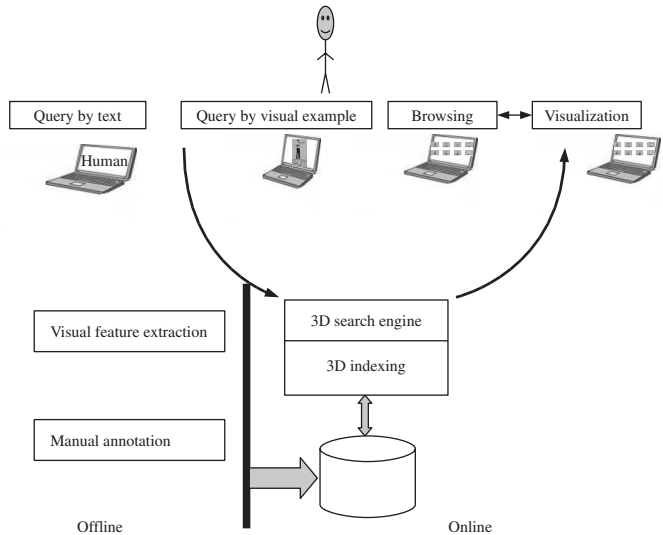

Figure 3.4 An overview of a 3D retrieval system

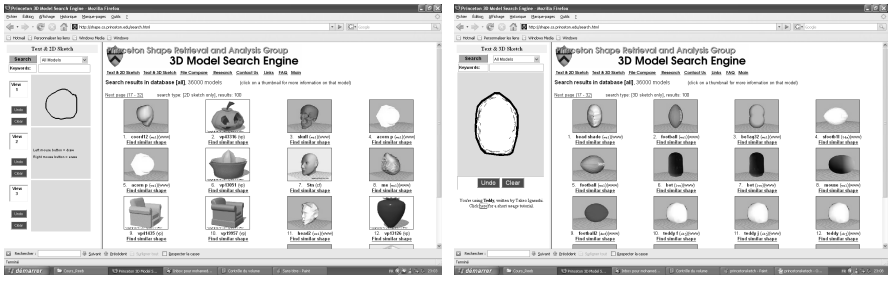

Figure 3.5 Examples from the Princeton Search Engine of 3D retrieval by sketch (2D and 3D sketch)

3.1.1 3D Shape-invariant Descriptors and Algorithmic Comparison Criteria of 3D Descriptors

Given 3D models, the goal now is to develop metrics and mechanisms for comparing their shapes.

Definition 3.1.1 *The shape is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object (Kendall 1984).*

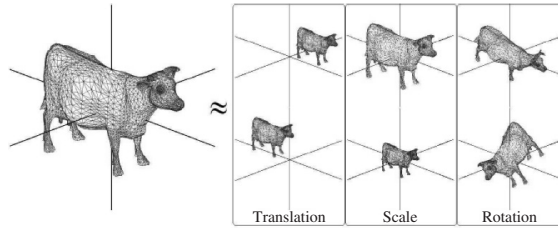

Figure 3.6 Some 3D models which have different locations, rotations and scales but the same shape. Courtesy of M Misha

So, an object's shape is invariant under the Euclidean similarity transformations of translation, scaling and rotation. This is reflected in Figure 3.6.

Definition 3.1.2 *A sequence of scalars* $\{I_k\}$ *is an invariant descriptor set with regard to a group of transformations G if and only if , for two 3D models* O_1 *and* O_2 *having the same shape,* $I_k(O_1) = I_k(O_2)$ *for all integer k.*

Two objects have the same shape if they can be translated, scaled and rotated to each other so they match exactly, i.e. if the objects are similar. In practice, we are interested in comparing objects with different shapes and so we require a way of measuring shape, some notion of distance between two shapes. This distance should be relatively easy to compute. The 3D scans of objects provide large amounts of data (approximately 20–30 000 vertices, edges, and so on), and a successful method should be able to analyze these data efficiently and compute the metric quickly.

It is important that the invariant descriptors fulfill certain criteria such as real-time computation, completeness and stability which express the fact that a small distortion of the shape does not induce a noticeable divergence. The descriptors must be independent of similarity transformations.

Several invariant descriptors have been proposed in the literature (Bustos *et al*. 2005). To compare the invariant descriptors, six algorithmic criteria have been defined: complexity, invariance under transformations, conciseness, robustness to noise as shown in Figure 3.7, etc, as follows:

- Size, *T*, representing the size of the vector corresponding to the descriptor of the 3D model.
- Extraction complexity, *CE*, representing the complexity of the algorithm that extracts the 3D model descriptor.

Figure 3.7 A 3D model from the Princeton Shape Benchmark with decimation and different noise effects

- Comparison complexity, *CMS*, representing the complexity of the similarity measure between two descriptors.
- Generality, *G*, specifying if the 3D model descriptor can be applied to topologically ill-defined 3D models and polygon soup.
- Geometrical invariance, *GI*, specifying if the descriptor is invariant to geometrical transformations (as illustrated in Figure 3.6).
- Topological invariance, *TI*, specifying if the descriptor is independent of the polygonal representations.

3.1.2 Evaluation Criteria

The efficiency of a 3D retrieval algorithm can be evaluated by the following statistics:

- **Nearest neighbor**: The percentage of the closest matches that belong to the same class as the query. This statistic provides an indication of how well a nearest neighbor classifier would perform. Obviously, an ideal score is 100 %; higher scores are associated with good results.
- **First tier and second tier**: The percentage of models in the query's class that appear within the top K matches, where K depends on the size of the query's class. Specifically, for a class with $|C|$ members, $K = |C| - 1$ for the first tier and $K = 2 \times (|C| - 1)$ for the second tier. The first-tier statistic indicates the recall for the smallest K that could possibly include 100 % of the models in the query class, while the second tier is a little less stringent (i.e. K is twice as big). These statistics are similar to the Bull Eye Percentage Score $K = 2 \times |C|$, which has been adopted by the MPEG-7 visual shape descriptors (SDs). In all cases, an ideal matching result gives a score of 100 %; again higher values indicate better matches.
- ***E*-measure**: A composite measure of the precision and recall for a fixed number of retrieved results. The intuition is that a user of a search engine is more interested in the first page of query results than in later pages. So, this measure considers only the first 32 retrieved models for every query

and calculates the precision and recall over those results. The E-Measure is defined as:

$$E = \frac{2}{\frac{1}{P} + \frac{1}{R}}$$

The E-measure is equivalent to subtracting van Rijsbergen's definition of the E-measure from 1. The maximum score is 1.0, and higher values indicate better results.

- **Discounted cumulative gain (DCG)**: A statistic that weights correct results near the front of the list more than correct results later in the ranked list under the assumption that a user is less likely to consider elements near the end of the list. Specifically, the ranked list R is converted to a list G, where element G_i has a value of 1 if element R_i is in the correct class and 0 otherwise. Discounted cumulative gain is then defined as follows:

$$\mathrm{DCG}_1 = G_1; \quad \mathrm{DCG}_i = \mathrm{DCG}_{i-1} + \frac{G_i}{\log_2(i)}, \text{ if } i > 1$$

The result is then divided by the maximum possible DCG (i.e. that would be achieved if the first C elements were in the correct class, where $|C|$ is the size of the class) to give the final score:

$$\mathrm{DCG} = \frac{\mathrm{DCG}_k}{1 + \sum_{j=2}^{|C|} \frac{1}{\log_2(j)}}$$

where k is the number of models in the database. The entire query result list is incorporated in an intuitive manner by the discounted cumulative gain (Leifman *et al*. 2003).

- **Recall vs. precision curves**: These curves are well known in the literature of content-based search and retrieval. The recall and precision are defined as follows:

$$\mathrm{Recall} = N/Q, \quad \mathrm{Precision} = N/A$$

where N is the number of relevant models retrieved in the top A retrievals, and Q is the number of relevant models in the collection, which is the number of models to which the query belongs.

Exercise 1 *Figure 3.8 shows five models of the ant–insect class from the* Princeton Shape Benchmark *classified by the user. The user provides the request illustrated in Figure 3.8 (top left) and the 3D search engine gives the first 16 results represented by Figure 3.9. Compute the statistics for nearest*

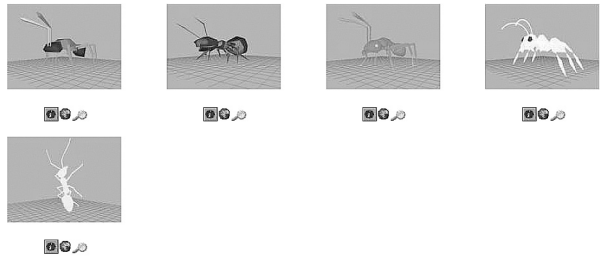

Figure 3.8 Insect class

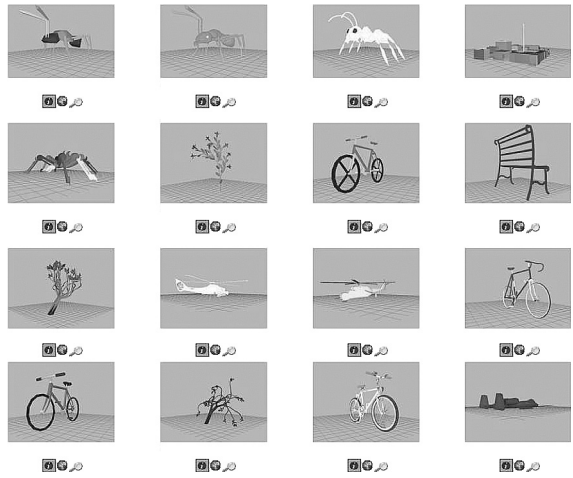

Figure 3.9 The first 16 relevant retrieval results

neighbor, first tier, second tier, E*-measure, and draw the recall/precision curve.*

3.2 Statistical Shape Retrieval

The 3D shape retrieval methods are not directly based on some measurements of shapes, but on the distribution of those measurements. This kind of distribution is generally represented by histograms of local or global features. The features could be curvatures on the surface representing the model, the Euclidean or geodesic distance between points on the surface,

angle measurements, elementary volumes, and so on. Each 3D model is then described as a distribution of such features. Therefore the principle of comparing two 3D shapes is very simple because it is reduced to a simple comparison of histograms.

In this section, we describe and analyze some methods for computing statistical 3D shape descriptors and dissimilarity measures. The first part introduces some differential geometry definitions. The second and third parts of this section are respectively about local and global features used to compute a histogram. The fourth part discusses the use of what we call hybrid approaches that consist of mixing histograms of local and global features to enhance the possibilities of these two kinds of features. The last part reports on the experiments and results of these three families of methods on a collection of 50 3D models.

3.2.1 Differential Geometries of Surfaces

Let S be a surface denoting a 3D model. Although in practice S is a triangulated mesh with a collection of edges and vertices, we start the discussion by assuming that it is a continuous surface. We recall the main surface properties.

Definition 3.2.1 *A subset S of $\mathbb{R}^3$ is called a regular surface if for each point p in S, there exists a neighborhood V of p in $\mathbb{R}^3$ and a map $\phi : U \to \mathbb{R}^3$ of an open set U, subset $\mathbb{R}^2$, onto V intersection S such that*

1. *ϕ is differentiable.*
2. *$\phi : U \to V \cap S$ is a homeomorphism.*
3. *Each map $\phi : U \to S$ is a regular patch.*

Any open subset of a regular surface is also a regular surface.

Property 1 *If $f : U \to \mathbb{R}$ is a differentiable function in an open set U of $\mathbb{R}^2$, then the graph of f, i.e. the subset of $\mathbb{R}^3$ given by $(x, y, f(x, y))$ for $(x, y) \in U$, is a regular surface.*

Definition 3.2.2 *Let $S \subset R^3$, be a regular and orientable surface. The principal curvatures k^1 and k^2 are the eigenvalues of the Weingarten endomorphism W defined by:*

$$W = I^{-1} \cdot II$$

where I and II are the two fundamental forms (do Carmo 1976) defined as follows:

$$I = \begin{bmatrix} \langle S_u, S_u \rangle & \langle S_u, S_v \rangle \\ \langle S_u, S_v \rangle & \langle S_v, S_v \rangle \end{bmatrix}, \qquad II = \begin{bmatrix} \langle S_{uu}, N \rangle & \langle S_{uv}, N \rangle \\ \langle S_{uv}, N \rangle & \langle S_{vv}, N \rangle \end{bmatrix}$$

where the scalar $\langle a\ , b \rangle$ represents the scalar product of two vectors a and b. The vector $a \wedge b$ represents the cross-product of two vectors a and b:

$$S_u = \frac{\partial S(u_0, v_0)}{\partial u}, \quad S_v = \frac{\partial S(u_0, v_0)}{\partial v}, \quad N = \frac{S_u \wedge S_v}{\|S_u \wedge S_v\|},$$

$$S_{uu} = \frac{\partial^2 S}{\partial u^2}, \quad S_{vv} = \frac{\partial^2 S}{\partial v^2} \quad \text{and} \quad S_{uv} = \frac{\partial^2 S}{\partial u v}$$

Exercise 2 *The surface of the cylinder (see Figure 3.12) admits the parametrization:*

$$S(u, v) = (\cos u, \sin u, v), U = \left\{(u, v) \in \mathbb{R}^3; 0 < u < 2\pi, -\infty < v < \infty\right\}$$

Compute the values S_u, S_{uu}, S_{uv}, S_{vv}, S_u, S_{uu}, S_{uv}, S_{vv}.

3.2.2 Local Approaches

The local feature histograms aim at using local feature measurements on the surface representing the 3D shape. These features can be understood as information from the 3D model when it is observed and analyzed very closely.

Many local features have been studied in the literature for use as local descriptors for distribution: elementary volumes extracted from each face of a 3D mesh (Zhang and Chen 2001), model area as a function of spherical angles (Ankerst *et al.* 1999) or curvatures extracted at each vertex of a 3D mesh (Zaharia and Prêteux 2001) are some well-known examples. It is also important to note that the curvature histogram has been chosen as an MPEG-7 3D descriptor and has been called the *3D shape spectrum descriptor* in this context.

Shape index (Koenderink and van Doorn 1992)

The 3D shape spectrum descriptor aims at providing an intrinsic shape description of 3D mesh models. The histogram is based on the *shape index* introduced by (Koenderink and van Doorn 1992). The shape index is defined

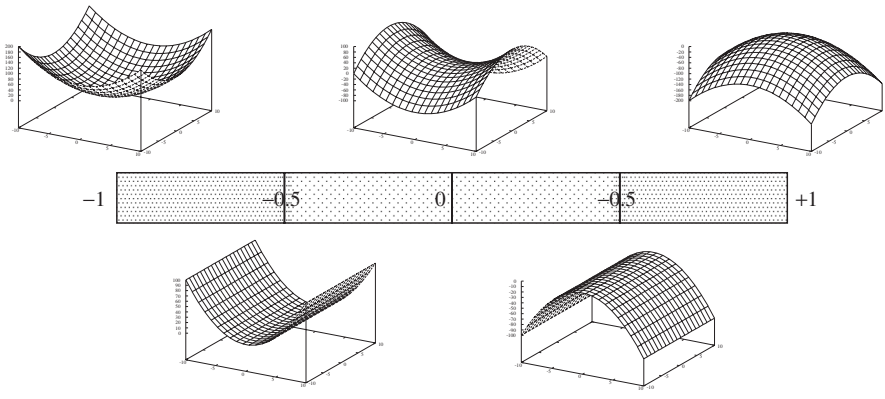

Figure 3.10 The curvature index values for some well-known curves. From left to right: spherical cup, rut, saddle, ridge, spherical cap

as a function of the two principal curvatures of the surface. The main advantage of this index is that it gives the possibility to describe the shape of the object at a given point. The drawback is that it loses the information about the magnitude of the surface shape.

Let us give more details about the computation of the curvature histogram. Let p be a point on the 3D surface. Let us denote by k_p^1 and k_p^2 the principal curvatures associated with the point p. The shape index at point p, denoted by I_p, is defined as:

$$I_p = \frac{2}{\pi} \arctan \frac{k_p^1 + k_p^2}{k_p^1 - k_p^2} \qquad \text{with} \quad k_p^1 \geq k_p^2 \tag{3.1}$$

The shape index value belongs to the interval $[0, 1]$ and is not defined for planar surfaces. Special values, outside of the interval, can be chosen for special surfaces like planes. Figure 3.10 illustrates some well-known surfaces and their values over the interval of shape index definition.

The shape spectrum of the 3D mesh is the histogram of the shape indexes calculated over the entire 3D mesh.

Estimation of the principal curvatures

Very often a surface is given as the graph of a differentiable function. Let $z = f(x, y)$ belong to an open set $U \subset \mathbb{R}^2$, and $S(u, v) = (u, v, f(u, v))$, $(u, v) \in U$, where $x = u$, $v = y$. A simple computation shows that:

$S_u = (1, 0, f_u)$, $S_u = (0, 1, f_v)$, $S_{uu} = (0, 0, f_{uu})$, $S_{uv} = (0, 0, f_{uv})$, $S_{vv} = (0, 0, f_{vv})$

$$I = \begin{bmatrix} 1 + {f_u}^2 & f_u f_v \\ f_u f_v & 1 + {f_v}^2 \end{bmatrix}, \qquad II = \frac{1}{\sqrt{(1 + {f_u}^2 + {f_v}^2)}} \begin{bmatrix} f_{uu} & f_{uv} \\ f_{uv} & f_{vv} \end{bmatrix}$$

The estimation of the principal curvatures at a point p is the key step of the shape spectrum extraction. Indeed, the spectrum extraction performances strongly depend on the accuracy of the estimation. Computing these curvatures can be done in different ways, each with advantages and drawbacks. Stokely and Wu (1992) proposed five practical methods to compute them: the Sander–Zucker approach (Sander and Zucker 1990), two methods based on direct surface mapping, a piece wise linear manifold technique, and a turtle geometry method. All these methods are well adapted to specialized uses (e.g. tomographic medical images).

Here, we choose to present a very simple and efficient method to compute the curvature at each vertex of the 3D mesh by fitting a quadric to the neighborhood of this vertex using the least squares method.

The parametric surface approximation is achieved by fitting a quadric surface through the cloud of the m points $\{(x_i, y_i, z_i)\}_{i=1}^{m}$ made at the centroids of the considered face and its 1-adjacent faces. Here, f is expressed as a second-order polynomial:

$$f(x, y) = a_0 x^2 + a_1 y^2 + a_2 xy + a_3 x + a_4 y + a_5$$

where the a_i coefficients are real values.

By denoting $a = (a_0, a_1, a_2, a_3, a_4, a_5)^t$ and $b(x, y) = (x^2, y^2, xy, x, y, 1)^t$, the previous equation can be expressed by using the standard matrix notation:

$$f(x, y) = a^t \cdot b(x, y)$$

The parameter vector $a = (a_0, a_1, a_2, a_3, a_4, a_5)^t$ is determined by applying a linear regression procedure. Given the data points denoted by $\{(x_i, y_i, z_i)\}_{i=1}^{m}$, the parameter vector corresponding to the optimal fit (in the mean squared error sense) is given by the following equation:

$$a = \left(\sum_{i=1}^{N} b(x_i, y_i) b^t(x_i, y_i) \right)^{-1} \cdot \left(\sum_{i=1}^{N} z_i b(x_i, y_i) \right)$$

From this quadric approximation, it becomes easy to compute the principal curvatures k^1 and k^2.

The curvature histogram

The shape index can now be computed by Equation (3.1) at each point of the 3D mesh. The different I_p values are put together in a histogram representing the curvature distribution for the 3D model.

For more details on curves, surfaces and curvature, we refer the reader to the reference book on differential geometry of curves and surfaces by do Carmo (1976). Another very interesting reference book has been written by Koenderink (1990) who introduced the shape index detailed in this part.

Invariance and robustness of the curvature histogram

Because the shape index (I_p value) used to compute the curvature histogram is a local descriptor, it uses only the neighborhood of the point to be calculated. This neighborhood is obviously invariant to rigid translations and rotations.

The I_p value is calculated with a quotient of the principal curvatures. Then the amplitude of a curve has no influence on the I_p value. Hence, the curvature histogram is also invariant to scaling.

However, the curvature histogram is not robust to tessellation of the 3D mesh. The computation of the shape index depends on the neighborhood of a point. If the resolution of the mesh (i.e. the size of the polygonal faces) is low, then the neighborhood will not be precise enough to approximate the local curve accurately. One idea to fix this is to compute a new tessellation of the 3D mesh with smaller and homothetic triangles.

We summarize the curvature histogram in Algorithm 1.

Algorithm 1 The curvature histogram

Given a surface S:

1. *Compute the values* $a_0, a_1, a_2, a_3, a_4, a_5$ *of the quadric* f *on each point of the surface.*
2. *Compute* $S_u = (1, 0, f_u)$, $S_u = (0, 1, f_v)$, $S_{uu} = (0, 0, f_{uu})$, $S_{uv} = (0, 0, f_{uv})$, $S_{vv} = (0, 0, f_{vv})$.
3. *Compute the two fundamental forms I and II, and the matrix W.*
4. *Compute the eigenvalues of the matrix W, and the values of the curvatures* k_1 *and* k_2.
5. *Compute* I_p.
6. *Compute the curvature histogram.*

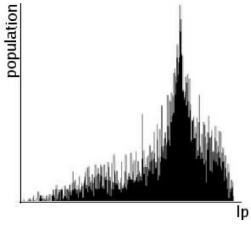

Figure 3.11 Curvature histogram for a 3D model cow (5804 faces)

Figure 3.11 shows an example of a curvature histogram for a 3D-model made of 5804 faces, computed by the algorithm 1.

Exercise 3 *Compute for the cylinder shown in Figure 3.12:*

1. *The first fundamental form.*
2. *The second fundamental form.*
3. *The curvatures k_1 and k_2.*
4. *The shape index I_p.*

The right cylinder over the circle $x^2 + y^2 = 1$ admits the parameterization $\phi : U \to \mathbb{R}^3$, where:

$$\phi(u, v) = (\cos u; \sin u, v),$$
$$U = \left\{(u, v) \in \mathbb{R}^3; 0 < u < 2\pi, -\infty < v < \infty\right\}$$

Let $R_{x,\theta} : \mathbb{R}^3 \to \mathbb{R}^3$ be the rotation of angle θ about the x axis. Compute the new value of the shape index I_p of the cylinder.

The correlogram of curvature

The main limitation of using curvature histograms for the description and retrieval of 3D models relates to the fact that any information about the spatial distribution of curvature values on the object surface is lost. Correlograms of curvature (Antini *et al.* 2005) have been proposed as a solution to this problem. In particular, with respect to a description based on curvature histograms, correlograms also enable encoding of information about the relative localization of curvature values.

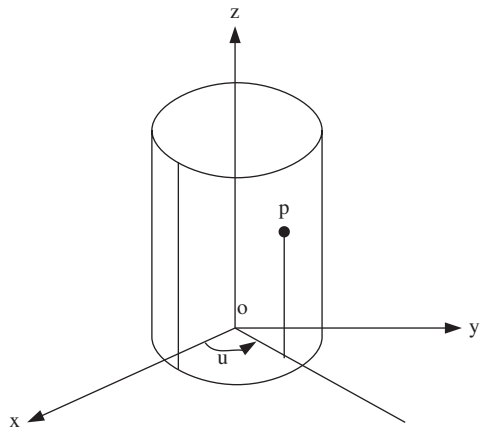

Figure 3.12 Cylinder parameterization

To compute correlograms of curvature, values of the mean curvature are quantized into $2N + 1$ classes of discrete values. For this purpose, a quantization module processes the mean curvature value through a stair–step function so that many neighboring values are mapped to one output value:

$$\mathcal{Q}(\overline{k}) = \begin{cases} N\Delta & \text{if } \overline{k} > N\Delta \\ i\Delta & \text{if } \overline{k} \in [i\Delta, (i+1)\Delta) \\ -i\Delta & \text{if } \overline{k} \in [-i\Delta, -(i+1)\Delta) \\ -N\Delta & \text{if } \overline{k} < -N\Delta \end{cases} \tag{3.2}$$

with $i \in \{0, \ldots, N-1\}$ and Δ a suitable quantization parameter (in the experiments reported in the following, $N = 100$ and $\Delta = 0.15$). Function $\mathcal{Q}(\cdot)$ quantizes values of $\overline{k}$ into $2N + 1$ distinct classes $\{c_i\}_{i=-N}^{N}$.

To simplify the notation, $v \in S_i$ is synonymous with $v \in S$ and $\mathcal{Q}(\overline{k}_v) = c_i$.

The correlogram of curvature is defined with respect to a predefined distance value δ. In particular, the correlogram of curvature $\gamma_{c_i c_j}^{(\delta)}$ of a mesh S is defined as:

$$\gamma_{c_i,c_j}^{(\delta)}(S) = \Pr_{v_1,v_2 \in S}[(v_1 \in S_{c_i}, v_2 \in S_{c_j}) \mid ||v_1 - v_2|| = \delta]$$

In this way, $\gamma_{c_i,c_j}^{(\delta)}(S)$ is the probability that two vertices that are δ far away from each other have curvature belonging to class c_i and c_j, respectively.

Ideally, $||v_1 - v_2||$ should be the geodesic distance between vertices v_1 and v_2. However, this can be approximated by the k-ring distance if the mesh S is regular and triangulated.

Definition 3.2.3 (1-ring) *Given a generic vertex $v_i \in S$, the* neighborhood *or* 1-ring *of v_i is the set:*

$$V^{v_i} = \{v_j \in S : \exists e_{ij} \in E\}$$

where E is the set of all mesh edges (if $e_{ij} \in E$ there is an edge that links vertices v_i and v_j).

The set V^{v_i} can be easily computed using the morphological operator *dilate*:

$$V^{v_i} = \text{dilate}(v_i)$$

Practically, the dilation operator is obtained by using a 1-ring as 'structure element'. A set of mesh vertices is dilated by extending the set with the 1-ring of any vertex of the set (Rossl *et al*. 2000).

Through the dilate operator, the concept of 1-ring can be used to define, recursively, a generic kth-order neighborhood:

$$\text{ring}_k = \text{dilate}^k \cap \text{dilate}^{k-1}$$

The definition of the kth-order neighborhood enables the definition of a true metric between vertices of a mesh. This metric can be used for the purpose of computing curvature correlograms as an approximation of the usual geodesic distance (which is computationally much more time consuming). According to this, we define the k-ring distance between two mesh vertices as $d_{ring}(v_1, v_2) = k$ if $v_2 \in \text{ring}_k(v_1)$.

Function $d_{ring}(v_1, v_2) = k$ is a true metric. In fact:

1. $d_{ring}(u, v) \geq 0$ and $d_{ring}(u, v) = 0$ if and only if $u = v$.
2. $d_{ring}(u, v) = d_{ring}(v, u)$.
3. $\forall w \in S, \quad d(u, v) \leq d(u, w) + d(w, v)$.

Based on the $d_{ring}(\cdot)$ distance, the correlogram of curvature can be redefined as follows:

$$\gamma^{(k)}_{c_i, c_j}(S) = \Pr_{v_1, v_2 \in S}[(v_1 \in S_{c_i}, v_2 \in S_{c_j}) \,|\, d_{ring}(v_1, v_2) = k]$$

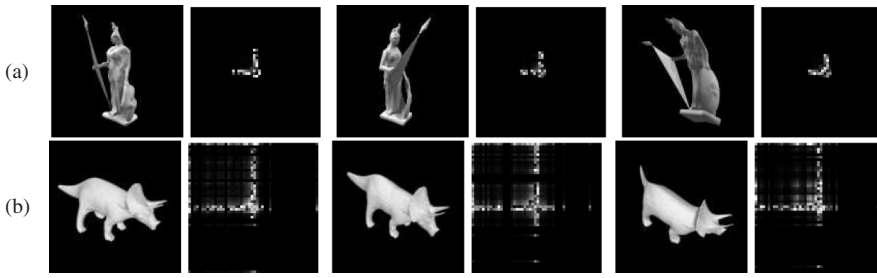

Figure 3.13 Correlograms of curvature extracted from sample 3D models: (a) statue; (b) dinosaur

Figure 3.13 shows some 3D models belonging to two classes of objects: statue and dinosaur. For each object the corresponding correlogram of curvature is also shown. It can be noted that values of the correlograms extracted from objects of the same class exhibit a common pattern.

Several distance measures have been proposed to compute the dissimilarity of distribution functions. In order to compute the similarity between curvature correlograms of two distinct meshes $\gamma_{c_i,c_j}^{(k)}(S_1)$ and $\gamma_{c_i,c_j}^{(k)}(S_2)$, we experimented with the following distance measures:

- *Minkowski-form distance*:

$$d_{\mathcal{L}_p} = \left[\sum_{i,j=-N}^{N} \left| \gamma_{c_i,c_j}^{(k)}(S_1) - \gamma_{c_i,c_j}^{(k)}(S_2) \right|^p \right]^{1/p}$$

- *Histogram intersection*:

$$d_{HI} = 1 - \frac{\sum_{i,j=-N}^{N} \min\left(\gamma_{c_i,c_j}^{(k)}(S_1), \gamma_{c_i,c_j}^{(k)}(S_2) \right)}{\sum_{i,j=-N}^{N} \gamma_{c_i,c_j}^{(k)}(S_2)}$$

- χ^2*-statistics*:

$$d_{\chi^2} = \sum_{i,j=-N}^{N} \frac{\left(\gamma_{c_i,c_j}^{(k)}(S_1) - \gamma_{c_i,c_j}^{(k)}(S_2) \right)^2}{2\left(\gamma_{c_i,c_j}^{(k)}(S_1) + \gamma_{c_i,c_j}^{(k)}(S_2) \right)}$$

- *Kullback–Leibler divergence*:

$$d_{KL} = \sum_{i,j=-N}^{N} \gamma_{c_i,c_j}^{(k)}(S_1) \log \frac{\gamma_{c_i,c_j}^{(k)}(S_1)}{\gamma_{c_i,c_j}^{(k)}(S_2)}$$

3.2.3 Global Approaches

Global descriptors are a means to handle the global nature of the object. This means that, rather than the details of the object, more importance is given to its general aspect. The moments (Lo and Don 1989; Sadjadi and Hall 1980) here are the traditional mathematical tool. However, the moments present a drawback: they rely on the object being described by a function, defined in major 3D space, which could for instance associate 1 or 0 to each 3D point, depending on whether it is inside or outside the object. Very few objects are defined by or convertible to such a function, which makes the moments method often impossible to use. To avoid this problem, it is also possible to grab the Z-buffer of an object's view, which consists of the 2D data that could be used to generate the moments. But in this section we want to deal with 3D space, so this method is not presented here.

Other global descriptor distributions have been studied in the literature. The most interesting ones have been proposed and developed by Osada *et al.* (2001). They are based on shape distributions. The main idea is to focus on the statistical distributions of a shape function measuring geometrical properties of the 3D model. They are represented as histograms, as in the local approach. The range of possible functions is very wide. The authors have proposed and tested five shape functions:

- A3: measures the angle between three random points on the surface of a 3D model;
- D1: measures the distance between a fixed point (e.g. the centroid of the boundary of the model) and one random point on the surface;
- D2: measures the distance between two random 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 shape functions have been chosen for their computational simplicity and invariance (see below for a discussion on the invariance and robustness of the distance histogram). According to Osada *et al.* (2001), the distribution of the distance between pairs of random points (D2) gives the best results compared to other simple methods.

Other authors have investigated the D2 descriptor to enhance its already very good performance. Ip *et al.* (2002) refined the D2 descriptor in the

context of CAD models with a classification of the two random points used to compute the descriptor. Ohbuchi *et al.* (2003a) also extended the D2 descriptor with a distribution called the Absolute Angle–Distance Histogram. This histogram is parameterized by one parameter denoting the distance between two random points (as in the original D2 shape descriptor) and by another parameter denoting the angle between the surfaces on which the two random points are located.

In the next sections, we give more details about the distance histogram (D2), its computation and its properties.

Distance histogram (D2)

The computation of the distance histogram is based on a stochastic method and is particularly simple.

Two random faces of the 3D model are taken, then two random points are taken on those two faces. Finally, the Euclidean distance between those two points is computed. The method is iterated N times, N being big enough to give a good approximation of the distribution. Empirically, Osada *et al.* (2001) found that $N = 1024^2$ is a good compromise between the computation time and storage and the precision of the distribution.

Figure 3.14 shows an example of a curvature histogram for a 3D model made of 5804 faces. Figure 3.15 shows distance histograms for some canonical shapes.

Invariance and robustness of the distance histogram (D2)

It can be seen that all the shape functions proposed by Osada *et al.* (2001) – including D2 – are invariant to rigid motions (translations and rotations). They are also robust to tessellation of the 3D mesh, since points are randomly selected on the surface of the 3D model. They are also relatively insensitive to small perturbations due to noise, cracks, etc., since sampling is area weighted.

But one should notice that the Euclidean distance, and thus the D2 index, are sensitive to scaling. This is obviously because the Euclidean distance is not either. So, the histogram has to be normalized before using it. One way to do this is to normalize the mean value of the distribution.

This value is first computed:

$$\mathrm{MV}(f) = \int_0^\infty x \cdot f(x) \cdot dx$$

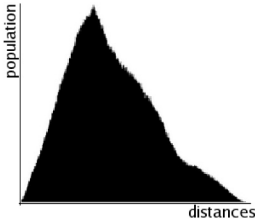

Figure 3.14 Distance histogram (D2) for the 3D model of a cow (5804 faces), O. Croquette, J-P. Vandeborre, M. Daoudi, C. Chaillou "Indexing and retrieval VRML models" *SPIE Electronic Imaging 2002, in proceedings volume 4672, pp. 95–106*. Reproduced by permission of SPIE

Then the new histogram is defined by:

$$f_{norm}(x) = f\left(\frac{x}{\mathrm{MV}(f)}\right)$$

The f_{norm} is the new distribution, invariant to scaling.

3.2.4 Hybrid Approaches

The previously presented local and global methods have advantages and drawbacks (see the conclusion of this section). Some authors proposed *hybrid* – or *mixing* – approaches to enhance advantages and to reduce the problems induced by these methods.

Paquet *et al*. (2000) proposed to use different descriptors: namely, bounding boxes, cords based, moments based and wavelet based. Paquet and Rioux (1997) also proposed, in the context of the Nefertiti project, to combine geometry and color/texture appearance to describe a 3D-model and express a user request.

More in keeping with the previously presented local and global approaches, Vandeborre *et al*. (2002) described the 3D models with three distributions: a curvature histogram, a distance histogram and an elementary volume histogram. In order to compare two 3D models, the authors compute three distances between pairs of similar histograms (one distance for the curvature histograms, one distance for the distance histograms and one distance for the elementary volume histograms) as shown in Figure 3.16. Then, the rank of each object is calculated according to each descriptor, sorting them by decreasing values (three integers between 1 and *NbObjects*,

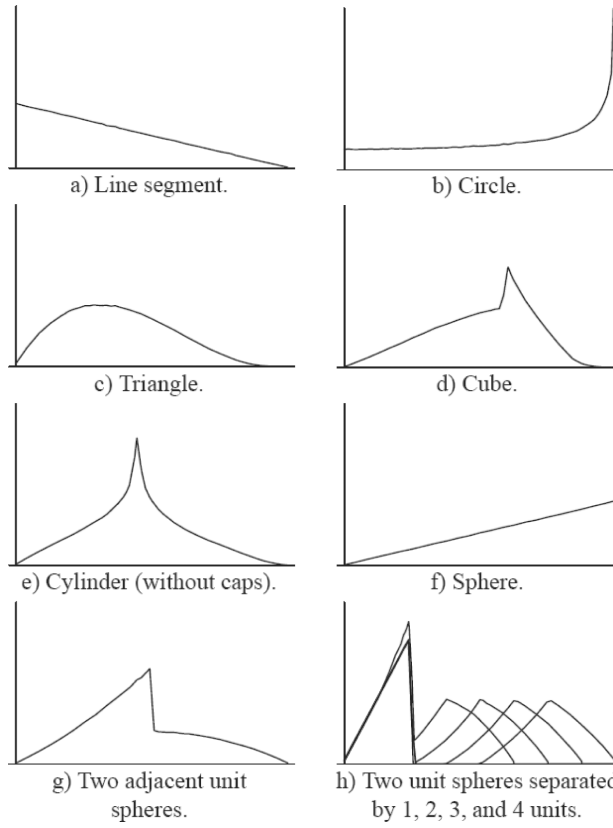

Figure 3.15 D2 distance histograms for some canonical shapes Osada *et al*. (2001). © 2001 IEEE

named $Rank_c$ for the curvature index, $Rank_d$ for the distance index and $Rank_v$ for the volume index). Those values are merged into a single value, using a formula. Intuitively, there are different strategies for merging these values:

- Support the objects having satisfactory results with one approach, the other ones having less importance. This is what the authors called the 'OR' method:

$$F = 1 - \left(\frac{(Rank_s - 1) \cdot (Rank_d - 1) \cdot (Rank_v - 1)}{NbObjects \times NbObjects \times NbObjects} \right)$$

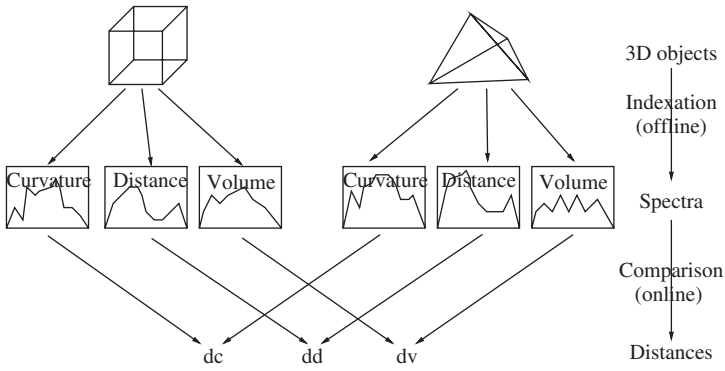

Figure 3.16 Comparison of two 3D models described by three histogram. O. Croquette, J-P. Vandeborre, M. Daoudi, C. Chaillou "Indexing and retrieval VRML models" *SPIE Electronic Imaging 2002, in proceedings volume 4672, pp. 95–106*. Reproduced by permission of SPIE

- Use the mean of the three results, called the 'MEAN' method:

$$F = \frac{(Rank_s + Rank_d + Rank_v)}{3 \cdot NbObjects}$$

This way, they obtain one final real number, between 0 and 1, which represents the confidence one could have in the result. Note that the same kind of formula can be used to merge just two rank values.

3.2.5 Experiments and Results

The methods described in the previous sections (i.e. curvature histogram, distance histogram, mixing methods) have been implemented and tested on a 3D model collection containing 50 models. Experiments and results are briefly reported in this section to show the effectiveness, advantages and drawbacks of the three families of approaches presented in the previous sections.

Before detailing the 3D model test collection and the results, let us explain the way the histograms are compared.

Comparison of histograms

There are several ways to compare distribution histograms: the Minkowski L_n norms, Kolmogorov–Smirnov distance, match distances, and many others.

One of the most used is the L_1 norm because of its simplicity and its accurate results.

The Minkowski L_n norm is given by:

$$D_{L_1}(f_1, f_2) = \int_{-\infty}^{+\infty} |f_1 - f_2|$$

where f_1 and f_2 are the two functions to be compared.

The first step before calculating the Minkowski L_n norm on two histograms is to interpolate them in a fixed number of linear segments by the least squares method in order to circumvent some problems like quantization, noise, etc. Then, a simple integration of these interpolations is done, giving a real number as a result.

3D model test collection

The 3D model test collection contains 50 models which have been collected from the MPEG-7 3D shape core experiments (Zaharia and Prêteux 2000) and arranged into seven classes. This manual classification has only been done for the purposes of our experiment. This classification is never used to improve the retrieval task of the search engine. The classes are the following: A-class (eight *aircraft* objects), M-class (considered just as noise, five *misc.* objects), P-class (seven *chess piece* objects), H-class (eight *human* objects), F-class (six *fish* objects), Q-class (eight *quadrupede* objects) and C-class (eight *car* objects). The models are 3D meshes ('soup' of polygons) of approximately 500 to 25 000 faces. Moreover, the mesh levels of detail are very different from one object to another.

Classification matrix

As the 3D model collection is classified, one can assume, for each request, that objects from the same class as the request are relevant, and consequently others are not. To evaluate the global performances of the different stand-alone descriptors (curvature and distance histogram) and of the mixing methods, classification matrices have been used.

Each column of a classification matrix (e.g. Figure 3.17) represents an object of the database (the first column corresponds to the first object, and so on) which has been used as a request in the search engine. Each small square shows how the object of the given row was ranked: the darker the square, the better the rank. For example, object 23 was ranked last when object 50 was

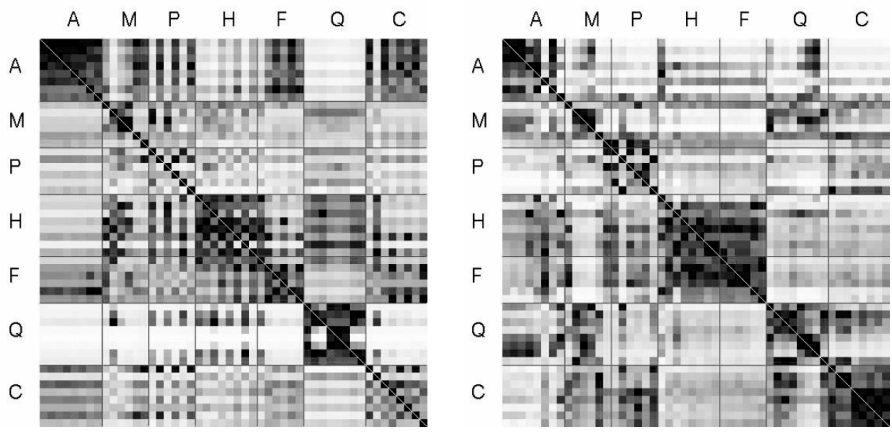

Figure 3.17 Classification matrices for the curvature (left) and distance (right) Histograms. O. Croquette, J-P. Vandeborre, M. Daoudi, C. Chaillou "Indexing and retrieval VRML models" *SPIE Electronic Imaging 2002, in proceedings volume 4672, pp. 95–106.* Reproduced by permission of SPIE

requested, 50 the square at the intersection of column 50 and line 23 is white in color. This also means that the diagonal is entirely black because each object is always the best result of its own request. Moreover, the matrices are not symmetric because the squares do not represent the distance between two objects, but a rank.

Results

The classification matrices for the curvature and distance histograms are shown in Figure 3.17. The first observation is that the two descriptors have difficulties in retrieving some classes. For example, the curvature histogram has no difficulty in classify the aircraft, but has many problems with the car class. Similarly, the distance index gives an accurate classification for the cars, but cannot give satisfactory results for the aircraft. Both curvature and distance indices have advantages and drawbacks.

Firstly, one can see that the two mixing methods are quite equivalent in terms of results. Figure 3.18 shows the classification matrices for the OR and the MEAN methods with the curvature and the distance histograms. Comparing these matrices (Figure 3.18) to the ones for the stand-alone histograms (Figure 3.17), it is clear that the two mixing methods can correct some weaknesses of the stand-alone histogram methods. For example, the curvature

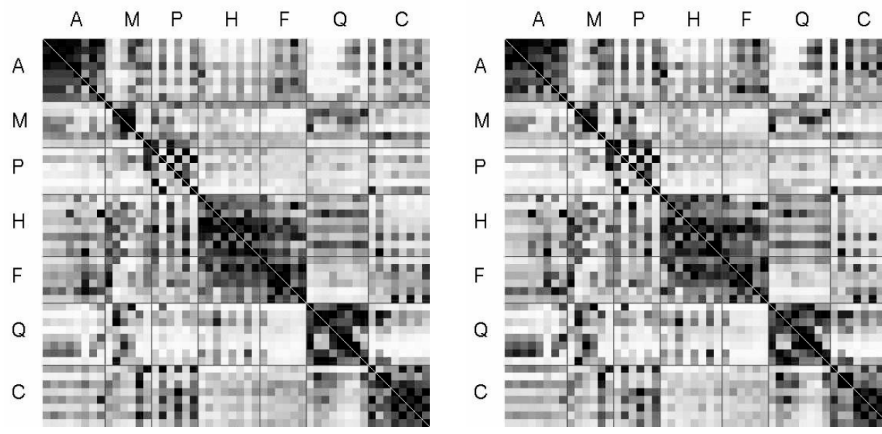

Figure 3.18 Classification matrices for the curvature (left) and distance (right) Histograms. O. Croquette, J-P. Vandeborre, M. Daoudi, C. Chaillou "Indexing and retrieval VRML models" *SPIE Electronic Imaging 2002, in proceedings volume 4672, pp. 95–106.* Reproduced by permission of SPIE

histogram matrix shows some difficulties with the cars, and the distance matrix shows some difficulties with the aircraft. Those problems no longer exist when any of the mixing methods are used. On the other hand, the curvature histogram is able to classify humans and fishes, whereas the distance histogram has some problems with these classes. So, the mixing methods cannot entirely organize these two classes correctly. Nevertheless, any mixing methods give better results than any of the single histogram methods.

3.2.6 Conclusion on Statistical Methods

Statistical methods have the main advantage of reducing the 3D model similarity measure to a very simple comparison of histograms. The previous shape functions used to compute these histograms generally have the long-awaited properties of invariance to rotation and to translation. Invariance to scaling is generally a matter of histogram normalization. Local approaches (such as the curvature histogram) are able to distinguish different classes of objects, but are sensitive to noise. Global approaches are robust to noise and can distinguish models in wide categories, but are not efficient at discriminating objects that are globally similar but with different small details in their shapes. Hybrid methods can be used to combine different statistical descriptors and enhance the performance of 3D model retrieval.

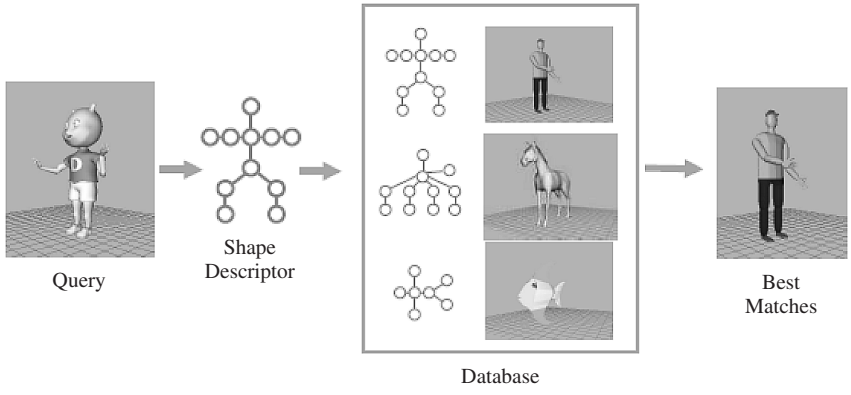

Figure 3.19 The method of 3D retrieval by parts

3.3 Structural Approaches

The aim of this section is to present structural methods for 3D retrieval. These methods use high-level information to describe the structure of 3D objects. We will mainly present the graph-based methods, as shown in Figure 3.19 illustrating a 3D retrieval system based on such information. The statistical shape descriptors represent the geometry of the model and they can be used to compare rigid body transformations and to compare whole models. However, they cannot be used to compare articulated models and parts of a model. The general approach proposes to construct graphs where nodes represent parts and edges represent relationships between parts. The proposed algorithms compare nodes and match graphs.

In this section, we are interested in the topology of the shape. Indeed, the topology describes properties of shape that are invariant under articulations. We can investigate topology by investigating critical points of Morse functions.

Definition 3.3.1 (Morse function) *Let $p(u) \in M \subset \mathbb{R}^3$ be a point on a closed embedded 2-manifold, in a neighborhood continuously parameterized by $u = (u_1, u_2)$. Let $f : M \to \mathbb{R}$ be any smooth function on the manifold. A point is critical if its gradient $\partial f/\partial u$ vanishes; otherwise, it is regular. A critical point is Morse if its Hessian matrix $H(p)$ is non-singular; otherwise, it is degenerate. If and only if all its critical points are Morse, then the function f is a Morse function.*

The Hessian of the function f at the point p is the matrix of second derivatives defined by:

$$\begin{bmatrix} \frac{\partial^2 f}{\partial u_1 \partial u_1}(p) & \frac{\partial^2 f}{\partial u_1 \partial u_2}(p) & \dots & \frac{\partial^2 f}{\partial u_1 \partial u_d}(p) \\ \frac{\partial^2 f}{\partial u_2 \partial u_1}(p) & \frac{\partial^2 f}{\partial u_2 \partial u_2}(p) & \dots & \frac{\partial^2 f}{\partial u_2 \partial u_d}(p) \\ \vdots & & \ddots & \\ \frac{\partial^2 f}{\partial u_d \partial u_1}(p) & \frac{\partial^2 f}{\partial u_d \partial u_2}(p) & \dots & \frac{\partial^2 f}{\partial u_d \partial u_d}(p) \end{bmatrix}$$

Exercise 4 *Compute the Hessian and, if defined, the index of the origin, which is critical for each function in the list below:*

1. $f(u_1, u_2) = u_1^2 + u_2^2$.
2. $f(u_1, u_2, u_3) = u_1 u_2 + u_1 u_3 + u_2 u_3$.

3.3.1 Introduction to Reeb Graph

A Reeb graph (Reeb 1946) is a topological structure that encodes the connectivity relations of the critical points of a Morse function defined on an input surface.

The Reeb graph is a schematic way of presenting a Morse function where the vertices of the graph are critical points and the arcs of the graph are connected components of the level sets of f, contracted to points. More formally, Reeb graphs are defined as follows.

Definition 3.3.2 (Reeb graph) *Let $f : M \to \mathbb{R}$ be a simple Morse function defined on a compact manifold M. The Reeb graph of f is the quotient space of f in $M \times \mathbb{R}$ from the equivalence relation $(p_1, f(p_1)) \sim (p_2, f(p2))$, if and only if:*

$$\begin{cases} f(p_1) = f(p_2) \\ p_1 \text{ and } p_2 \text{ belong to the same connected} \\ \qquad \text{component of } f^{-1}(f(p_1)) \end{cases}$$

Figure 3.20 gives an example of a Reeb graph computed on a bi-torus with regard to the height function and illustrates well the fact that Reeb graphs can be used as skeletons.

Constructing a Reeb graph from a Morse function f computed on a triangulated surface first requires identification of the set of vertices corresponding to critical points. With this aim, several formulations have been proposed (Edelsbrunner and Mücke 1990; Takahashi *et al.* 1995) to identify local

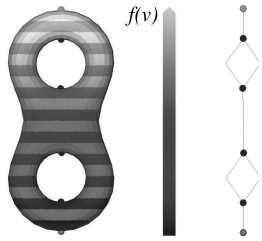

Figure 3.20 Evolution of the level lines, using the height function as $f(v)$, on a bi-torus, its critical points and its Reeb graph, J. Tierny, J-P. Vandeborre, M. Daoudi "Invariant high-level Reeb graphs of 3D polygonal meshes" IEEE 3DPVT 2006, 3rd International Symposium on 3D Data Processing, Visualization and Transmission, Chapel Hill, North Carolina, USA, June 14–16, 2006, pp. 105–112. © 2006 IEEE

maxima, minima and saddles, observing for each vertex the evolution of f at its direct neighbors. Several algorithms have been developed to construct Reeb graphs from the connectivity relations of these critical points (Carr *et al*. 2004; Cole-McLaughlin *et al*. 2003), most of them in $O(n \times log(n))$ steps, with n the number of vertices in the mesh.

However, they assume that all the information brought by the Morse function f is relevant (Carr *et al*. 2004; Ni *et al*. 2004). Consequently, they assume that all the identified critical points are meaningful, while in practice this hypothesis can lead to intractably large Reeb graphs. To overcome this issue, (Ni *et al*. 2004) developed a user-controlled simplification algorithm. (Bremer *et al*. 2004) proposed an interesting critical point cancellation technique based on a *persistence* threshold. (Attene *et al*. 2003) proposed a seducing approach, unifying the graph construction and simplification, but it is conditioned by a *slicing* parameter.

Lemma 1 *The Reeb graph of a Morse function on a compact, closed, orientable 2-manifold of genus g has g loops.*

Exercise 5 *Consider the Morse function $f : S \to \mathbb{R}$ whose level lines have been drawn in Figure 3.21. Compute the Reeb graph for the following surface (Figure 3.21).*

3.3.2 Reeb Graphs for Shape Matching

A Reeb graph encodes the behavior of a Morse function on the shape and also tells us about the topology of the shape. The main properties of 3D shape

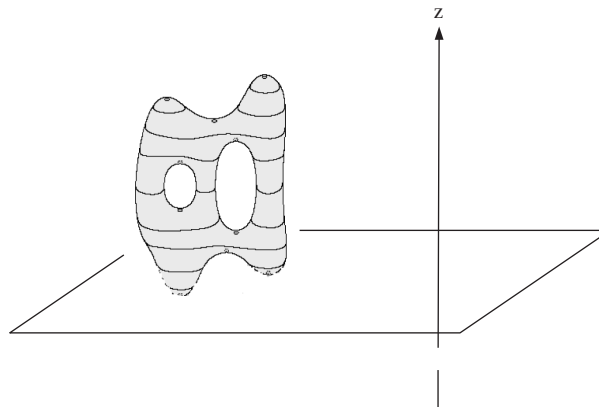

Figure 3.21 Example of geodesic distance

matching algorithms are the invariance to rotation and robustness to noise and variability in model pose. However, the height function is not appropriate for 3D shape matching because it is not invariant to transformations such as rotation. In order to avoid these problems many authors propose to use a geodesic distance: that is, the distance from point to point on a surface. Figure 3.22 gives an example of geodesic distance computed between two

Figure 3.22 Example of geodesic distance

points. Using geodesic distance provides rotation invariance and robustness against problems due to noise.

Hilaga *et al.* (2001) presented an approach to describe the topology of 3D models by a graph structure and show how to use it for matching and retrieval. They proposed the geodesic function μ defined by:

$$\mu(v) = \int_{p \in s} g(v, p) ds$$

where the function $g(v,p)$ returns the distance between v and p on s. The discrete version of this function is defined by:

$$\mu : \mu(v) = \sum g(v, b_i) \times \text{area}(b_i)$$

In Tierny *et al.* (2007b), first the mesh feature points in Figure 3.23(a) are automatically extracted, intersecting geodesic-based map extrema. Then, for each vertex in the mesh, a mapping function f_m is defined as the geodesic distance to the closest feature point (Figure 3.22). Next, for each vertex $v \in S$, an upper value approximation of $f_m^{-1}(f_m(v))$, denoted $\Gamma(v)$, is computed along the edges of S. In particular, the connected component of $\Gamma(v)$ containing v is identified and denoted $\gamma(v)$. Analyzing the evolution of the number of connected subsets of $\Gamma(v)$ as f_m evolves enables the construction of a Reeb graph (Reeb 1946) (Figure 3.23(b)). At this stage, each connected component of the Reeb graph is modeled by an ordered collection of closed curves $\gamma(v)$. The next step consists of identifying *constrictions* (Hétroy 2005) within these collections. For each connected component of the Reeb graph, the average Gaussian curvature on each curve $\gamma(v)$ is computed. Then, local

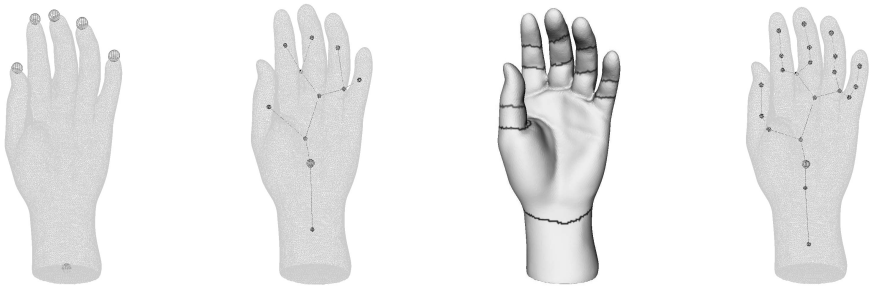

Figure 3.23 Enhanced topological skeleton extraction Tierny *et al.* (2007b). © 2007 IEEE

(a) (b) (c) (d)

Figure 3.24 Segmentation of a hand surface model into its sub parts Tierny *et al*. (2007a). © 2007 Blackwell Publishing

negative minima are identified as constrictions (Figure 3.23(c)). Finally, the connected components of the Reeb graph are subdivided using these constrictions as boundaries between sub parts (Figure 3.23(d)). As a conclusion of this algorithm, the input surface is represented by an enhanced topological skeleton (Figure 3.23(d)), which encodes the topological and geometrical evolution of the contours of the mapping function f_m.

Once the Reeb graph is computed, the surface is segmented to a set of sub parts of the surface. Figure 3.24 shows an example of the results obtained by this algorithm.

Tierny *et al*. (2007a) proposed a new idea to compare two surfaces. It is based on the key idea that two surface models are similar if their sub parts are similar. For each subpart its signature is computed as a function of the distortion introduced by its mapping to its canonical planar domain.

They proposed to compare two subpart signatures by using an L_1 distance. Then, they computed the distance between two models by running a bipartite matching algorithm (Tam and Lau 2007) that matches pairs of topology-equivalent sub parts that minimize their distance, minimizing the overall sum of distances, denoted d. Finally, the distance between the two models is given by d.

Figure 3.25 shows a typical query and the results retrieved by the system. Sub parts that have been matched together have been displayed with the same gray level. Notice that, except in one case, the tail of the horse query model has been matched by the tail of each retrieved result. Similar comments can be made for the legs, or the neck, which demonstrates the efficiency of the proposed signature. Moreover, this figure shows that the proposed signature is clearly pose insensitive, since horses in different poses have been retrieved as the top results.

Figure 3.26 zooms in on the subpart matchings between a hand and a human surface model, displaying some of the sub parts that have been

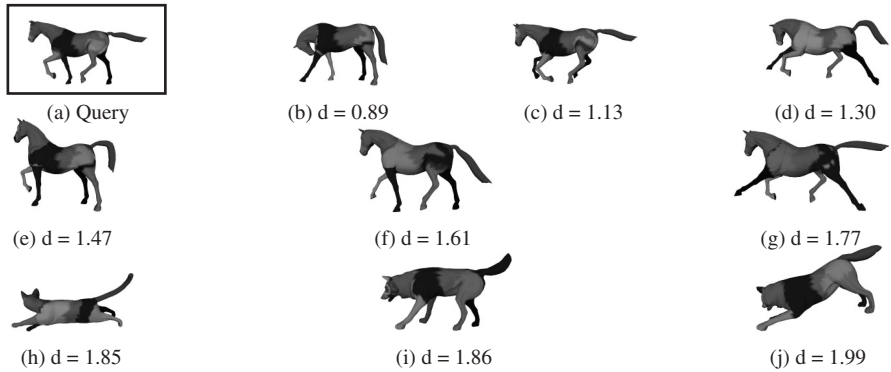

Figure 3.25 Subpart similarity matchings between a horse query model and the retrieved results Tierny *et al*. (2007a). © 2007 Blackwell Publishing

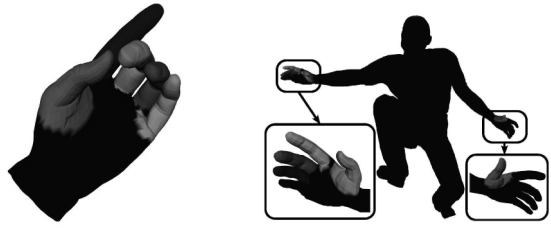

Figure 3.26 Subpart similarity matchings between a hand and a human model Tierny *et al*. (2007a). © 2007 Blackwell Publishing

matched together. Notice that the thumb has been matched with the correct thumb of the humanoid and that the remaining fingers have been matched with the fingers of the humanoid, despite their difference in pose. These results can be extended using graph-based matching algorithms so that, for example, the hand model can be matched by its corresponding hand in the humanoid, achieving partial shape similarity (Funkhouser *et al*. 2004).

3.4 Transform-based Approaches: Spin Images

This section provides an overview of transform-based 3D shape descriptors. These descriptors, such as spherical harmonics, spin images, Zernike 3D, etc, are based on the transformation of the 3D shape from 3D Euclidean space representation to another space representation.

A description of 3D models based on spin images has been proposed by Johnson and Hebert (1999) with the aim of supporting a view-independent recognition of objects by capturing both their global and local features.

Such a description of 3D models based on spin images can be regarded as a compromise between view-based and structure-based approaches to object description. On the one hand, a description based on spin images shares relevant common traits with view-based approaches. For instance, each spin image is obtained by projecting information of interest according to a particular viewpoint on the object surface. On the other hand, information that is projected on a spin image does not reflect just object parts that are visible from a given viewpoint as is typical of view-based approaches. Rather, projected data encode geometric features of visible as well as cluttered object parts.

Basically, spin images encode the density of mesh vertices projected onto a 2D space: the 3D mesh vertices are first mapped onto a 2D space defined with respect to one point in the 3D space; then, the resulting coordinates are used to build a 2D histogram.

More precisely, let $O = \langle \boldsymbol{p}, \boldsymbol{n} \rangle$ be an *oriented point* on the surface of the object, where $\boldsymbol{p}$ is a point on the surface of the object and $\boldsymbol{n}$ the normal of the tangent plane in $\boldsymbol{p}$. For a generic oriented point O, a *spin map* can be defined, which maps any point $\boldsymbol{x}$ in the 3D space onto a 2D space according to the following formula (see also Figure 3.27 for the notation):

$$S_O(\boldsymbol{x}) \rightarrow [\alpha, \beta] = [\sqrt{\|\boldsymbol{x} - \boldsymbol{p}\|^2 - (\boldsymbol{n} \cdot (\boldsymbol{x} - \boldsymbol{p}))^2}, \boldsymbol{n} \cdot (\boldsymbol{x} - \boldsymbol{p})]$$

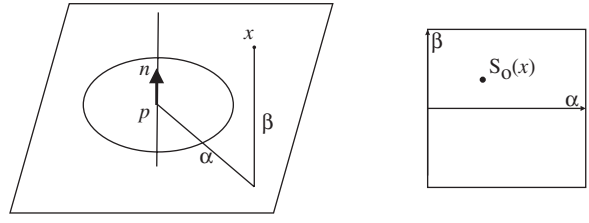

Figure 3.27 Given an oriented point $\langle \boldsymbol{p}, \boldsymbol{n} \rangle$ on the object surface, a generic point $\boldsymbol{x}$ is mapped on to point $[\alpha, \beta]$ on the spin map, $[\alpha, \beta]$ being the radial distance and the elevation of $\boldsymbol{x}$ w.r.t. to $\langle \boldsymbol{p}, \boldsymbol{n} \rangle$: (a) the object-centered 3D coordinate system, (b) the spin map coordinate system, "Content-based Retrieval of 3D Models A. Del Bimbo, P. Pala ACM Transactions on Multimedia Computing, Communications and Applications, Vol. 2, No. 1, February 2006", 2006 ACM, Inc. Reprinted by permission

In other words, the oriented point defines a family of cylindrical coordinate systems, with the origin at $\boldsymbol{p}$, and with the axis along $\boldsymbol{n}$. The spin map projection of $\boldsymbol{x}$ retains the radial distance (α) and the elevation (β), while it discards the polar angle. In so doing, the ambiguity is resolved, which results from the fact that the oriented point does not define a unique cylindrical coordinate system. This projection ensures that, for any given oriented point, a unique spin map exists.

To produce a *spin image* of an object, a spin map is applied to points of the object surface. Hence, given a mesh representation of the object, the spin image is obtained by applying the spin map to vertices of the mesh. Since several vertices can be mapped into the same (α, β) coordinates, the spin image retains information about the number of vertices mapped onto each (α, β) value. Hence, the spin image is a 2D histogram $I(i, j)$ that is constructed by projecting coordinates α and β of each mesh vertex according to a bilinear interpolation scheme. The purpose of the interpolation scheme is to spread the contribution of each vertex over the nearest points on the grid obtained by quantizing values of α and β. By spreading the contribution of each vertex, a smooth representation is obtained in contrast to the 'dotted' map that would be obtained by simply accumulating indexed values of α and β (see Figure 3.28).

Most relevant characteristics of spin images are invariance to rigid transformations (as a consequence of the adoption of an object-centered coordinate system), limited sensitivity to variations in the position of mesh vertices (which might result from the adoption of different sampling schemes), flexibility (since no hypotheses are made on the surface representation) and ease of computation.

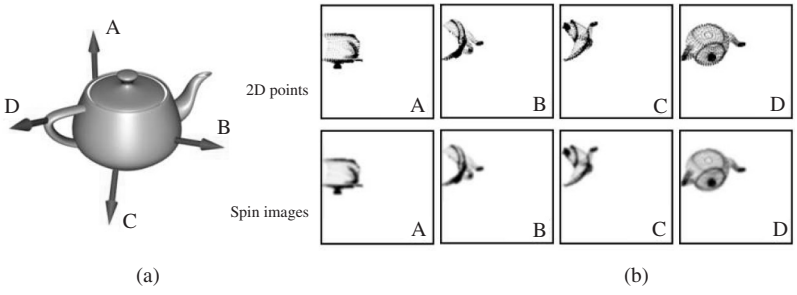

Figure 3.28 A 3D object (a), together with 2D point images and spin images computed from four distinct points on the object surface, 2006 ACM, Inc. Reprinted by permission

Object description based on spin images entails a huge amount of information which makes it difficult to use them for applications addressing retrieval by similarity from large databases rather than object recognition in 3D scenes. To overcome these limitations, spin image signatures have been proposed in Assfalg *et al*. (2007). Spin image signatures make use of feature extraction and clustering techniques to meet the storage and efficiency requirements of content-based retrieval from large 3D model repositories.

3.5 View-based Approach

The human visual system has an uncanny ability to recognize objects from single views, even when presented monocularly under fixed viewing conditions. For example, the identity of most of the 3D models in Figure 3.29 is obvious.

The issue of whether 3D model recognition should rely on internal representations that are inherently 3D or on collections of 2D views has been explored by Riesenhuber and Poggio (2000). They showed that, in a human visual system, a 3D model is represented by a set of 2D views.

As shown in Figure 3.30, the process of 3D model comparison using views can be broken into two phases: a phase of indexing and a phase of retrieval. In the phase of indexing, for every 3D model of the database, we calculate the characteristic views and their associated descriptors. In the retrieval phase, the query undergoes a similar treatment as the 3D models of the database, following which a descriptor (invariant to some deformations) is calculated

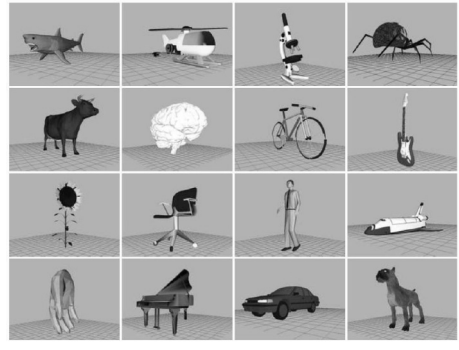

Figure 3.29 Some 3Dmodels from the Princeton Shape Benchmark. Reproduced from Princeton Shape Benchmark

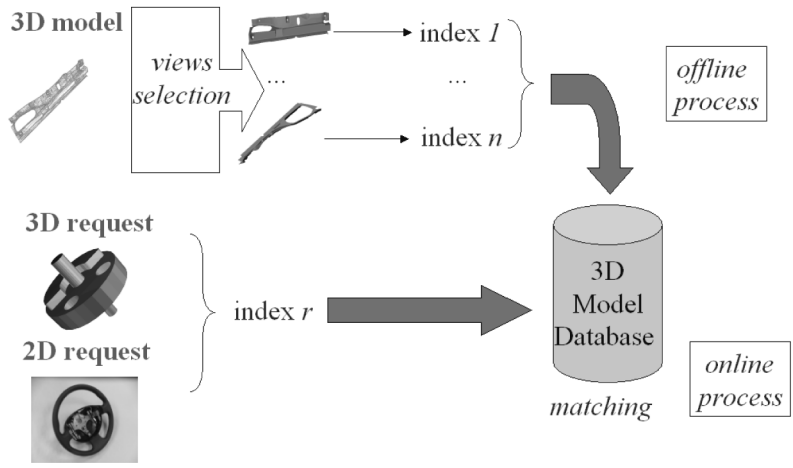

Figure 3.30 The view selection process Filali Ansary *et al*. (2007). © 2007 IEEE

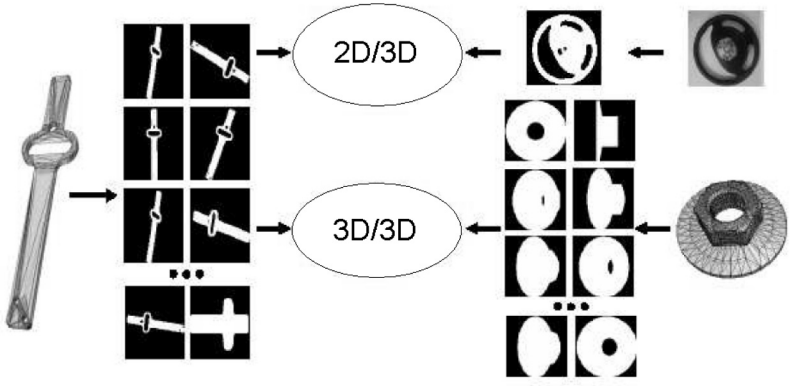

Figure 3.31 A 3D model characterized by a set of 2D views and two types of 3D retrieval. Courtesy of J Ricard

and compared to the descriptors of the database objects. Figure 3.31 shows an example of a 3D model characterized by a set of 2D views and the two types of realizable retrieval: 2D/3D by comparing the set of view descriptors to a query image: and 3D/3D by comparing the two sets of views.

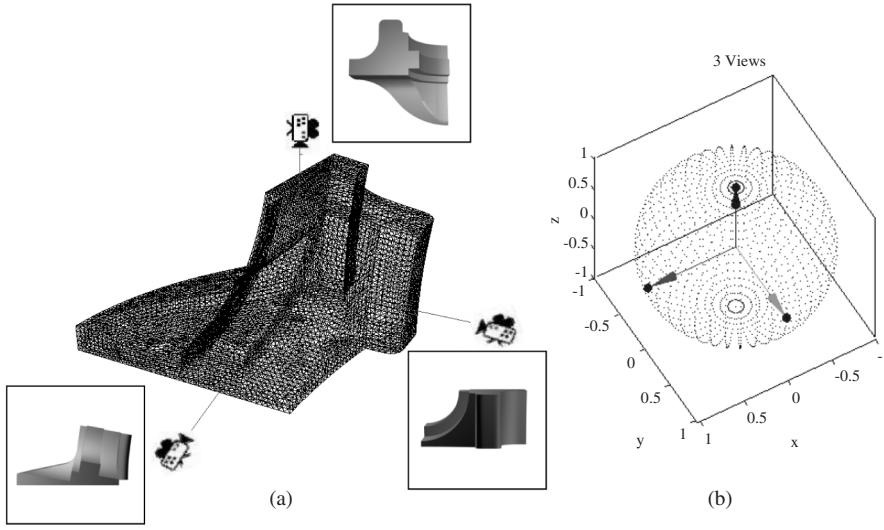

Figure 3.32 Three views created around the 3D model corresponding to three principal axes. Courtesy of J Ricard

The process of comparing two 3D models using their views can be separated in to two main steps:

- **3D model indexing**: Each 3D model is characterized by a set of 2D views. For each view, a 2D descriptor is calculated.
- **3D model retrieval**: The same descriptor is applied to the 2D query views. A matching process is then started to match the request 2D views to the views representing the 3D models of the database.

The main idea of this type of method is to represent a 3D model by a set of characteristic views. For that, the space of views of the 3D model is discretized to N points of view distributed around the 3D model. For each point, a view (2D image) of the 3D model is taken. Figure 3.32 shows three views created around the 3D model by placing the cameras on the three principal axes.

Each view is indexed using a method of 2D image analysis. In the retrieval process, each view of the 3D model request is indexed and compared to the N views of each 3D model. The two main points discussed in the 3D

retrieval literature are the selection of the characteristic views and the choice of the descriptor. The choice of the view descriptor will not modify the principle of the view-based approach, but can modify the performance and the effectiveness of it.

The main idea of view-based similarity methods is that two 3D models are similar, if they look similar from all viewpoints. This paradigm leads to the implementation of query interfaces based on defining a query by one or more views, sketches, photos showing the query from different points of view. Figure 3.30 shows an overview of the characteristic view selection algorithm. The main difficulty in view-based 3D retrieval is to characterize a 3D model by a set of 2D views.

One point which will have an impact on the effectiveness and the speed of the view-based methods is the number and position of views used to describe a 3D model. This number is directly related to the performance of the system. Two approaches were used to choose the number of characteristic views:

- Using a fixed and restricted number of views.
- Using a dynamic number of views.

3.5.1 Methods with a Fixed Number of Views

To guarantee a low number of views and consequently quick retrieval, a class of techniques can create a restricted number of views with fixed positions, independently of the 3D model to be indexed. These methods select a fixed number of views on a sphere surrounding the object and create images starting from these points of view. To increase the similarity between views, the 3D models are aligned and scaled. For this, the principal axes of the objects are calculated by analyzing the principal component (PCA), for example, or by the calculation of the matrix of covariances. Figure 3.33 shows four examples of views distributed on the viewing sphere.

Light-field descriptor

Chen *et al.* (2003) proposed a descriptor used for comparing the similarity among 3D models extracted from 4D light fields, which are representations of a 3D model. The light fields describe the radiometric properties of light in space. A light field (or plenoptic function) is traditionally used in image-based rendering and is defined as a 5D function that represents the radiance at a given 3D point in a given direction. For a 3D model, the representation is the same along a ray, so the dimension of the light fields around an object

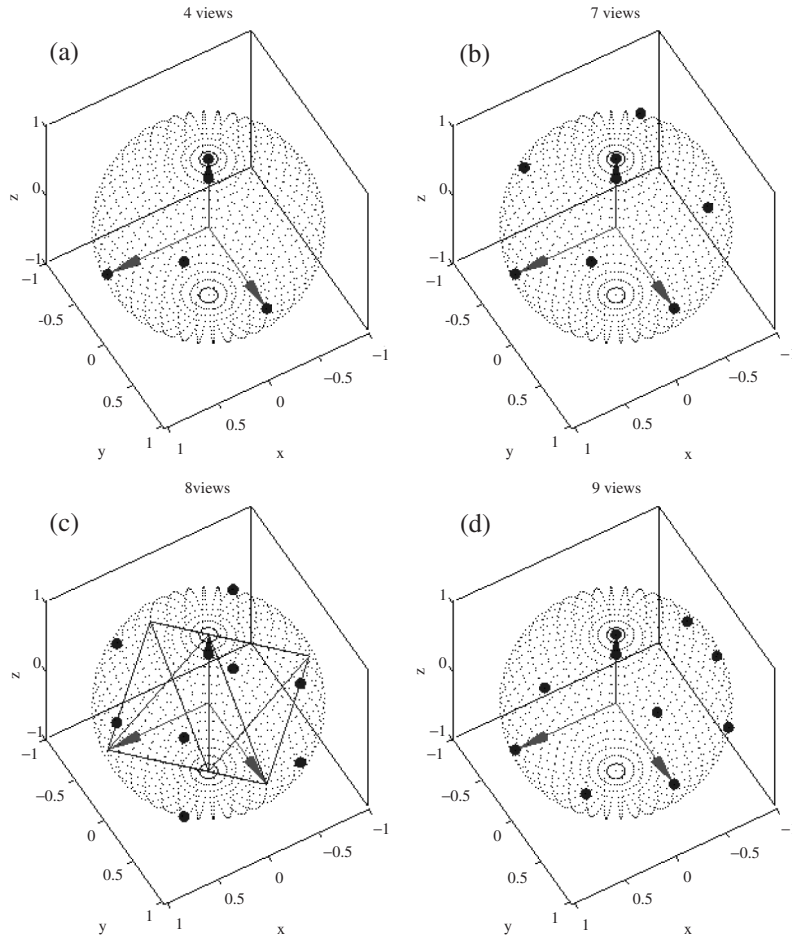

Figure 3.33 Example of views distributed on the viewing sphere

can be reduced to four. Each 4D light field of a 3D model is represented by a collection of 2D images, which are rendered from a 2D array of cameras. The camera positions of one light field can be put either on a flat surface or on a sphere in the 3D world. The light-field representation has not only been used in image-based rendering, but also in image-driven simplification to decide which portions of an object to simplify.

The main idea comes from the following statement: 'If two 3D models are similar, they also look similar from all viewing angles.' Accordingly,

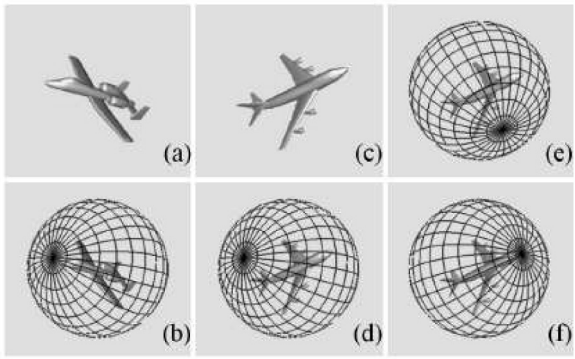

Figure 3.34 The main ideas for measuring similarity between two 3D models using a Light-field descriptor. Reproduced from D Chen *et al*, 2003

the similarity between two 3D models can be measured by summing the similarity from all corresponding images of a light field. However, what must be considered is the transformation, including translation, rotation and scaling.

The camera system surrounding each model is rotated until the highest overall similarity (cross-correlation) between the two models from all viewing angles is reached. Take Figure 3.34 as an example, where (a) and (c) are two different aircraft with inconsistent rotations. Firstly, for the aircraft in (a), the cameras of a light field are placed on a sphere, as shown in (b), they are put on the intersection points of the sphere. Then, the cameras of this light field can be applied, at the same positions, to the aircraft in (c), as shown in (d). By summing the similarities of all pairs of corresponding images in (b) and (d), the overall similarity between the two 3D models is obtained. Next, the camera system in (d) can be rotated to a different orientation, such as in (e), which leads to another similarity value between the two models. After evaluating similarity values, the correct corresponding orientation, in which the two models look most similar from all corresponding viewing angles, can be found, such as in (f). The similarity between the two models is defined by summing the similarity from all corresponding images between (b) and (f):

1. Since each model has its own coordinate system, translation and scaling are applied first in order to ensure that a model is entirely contained in each rendered image. The input 3D model is translated from the center of

the model to the origin of the world coordinate system, and then scales the axis of maximum length to be 1. The translation $T = (T_x; T_y; T_z)$ assigns the middle point of the whole model to be the new origin:

$$T_i = \frac{MaxCoor_i + MinCoor_i}{2}, \quad i = x, y, z$$

where $MaxCoor_i$ and $MinCoor_i$ are the maximum and minimum coordinate values of the i axis, respectively. The scaling is isotropic, and normalizes according to the maximum distance from the x, y and z axes of the whole model:

$$S = \frac{1}{\min_{i=x,y,z}(MaxCoor_i + MinCoor_i)}$$

2. Images are rendered from the camera positions of the light fields, which are on the surface of a larger sphere. There are 10 light fields for each 3D model, and the camera positions of each light field are set at the 20 vertices of a regular dodecahedron. The camera at each viewpoint is directed toward the center of the sphere, and the up-vectors of the cameras are placed uniformly. If two 3D models are of different orientations, their proper corresponding viewing images will have different rotational angles. This does not matter, since the image metric of the approach is also robust against rotations.
3. An orthogonal projection is used in order to reduce the size of descriptors. Therefore, in a descriptor of a light field, there are 10 images represented from 20 viewpoints. For a 3D model, 10 descriptors of light fields are created, so there are 100 images that should be rendered and extracted for features.
4. The Zernike moment and Fourier descriptor are extracted from each image. Descriptors for a 3D model are those features from the 100 images.

Depth buffer-based methods

Ohbuchi *et al*. (2003b) proposed a method which works on polygon soup 3D models (MODFD). The 3D 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 sets 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 M_1 and M_2 be two 3D models to be compared. The similarity measure is defined as follows:

$$D(M_1, M_2) = \frac{1}{N} \sum_{i=1}^{N} \min_{j \in 1 \ldots N} d(M_{1i}, M_{2j})$$

where M_{1i}, M_{2j} are the computed 2D descriptors and the distance d is the 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.

Vranic (2003) proposed a descriptor based on the *buffer-based descriptor* (BBD) introduced by Heczko *et al*. (2002). To make the descriptor invariant to rotation and scaling, each 3D model is oriented using continuous PCA and normalized for scaling.

Using projections on the bounding box, six depth buffer images are used. Each image is described using a 2D fast Fourrier transform (2D-FFT). The 3D model descriptor is based on the low-frequencies features of the 2D-FFT of each depth buffer image.

Chaouch and Verroust-Blondet (2006) developed a new approach called the *enhanced depth buffer-based descriptor*, based on the depth buffer images and the continuous PCA to be geometrically invariant. This method uses the same descriptor as Vranic (2003), but weights the buffer images used in the descriptor, based on informational criteria.

The approach suggested by Vajramushti *et al*. (2004) also consists of describing the 3D model using depth images. The surface and the volume calculated from these projections constitute the characteristic features vector of the 3D model. These vectors are then used in an iterative algorithm to estimate the errors between the depth images while varying the six degrees of freedom characterizing the 3D model position vector. This approach allows us to find the best corresponding 2D depth images to overcome the PCA problems.

Other methods

We can cite, for example, the work of Abbasi and Mokhtarian (2001), which fixe the number of views associated with each 3D model to nine views: four views corresponding to the top, side, front and back views, and five intermediate views (Figure 3.33(d)). Chen and Stockman (1998) set the number of views to eight views equally distributed on the viewing sphere of the

3D model. The camera is placed on each viewpoint to obtain the views (Figure 3.33(c)).

Nayar *et al.* (1996), as well as Mahmoudi and Daoudi (2007), considered each 3D model as a cloud of points. Principal axes of each 3D model are calculated using the eigenvalues of the matrix of covariance. The 3D model cloud of points is projected in 2D from according to the three principal directions thus calculated. Seven characteristic views are created: three principal views along the principal axes and four views according to the four directions corresponding to 45° views between the principal views (Figure 3.33(b)).

3.5.2 Methods with a Dynamic Number of Views

Conventional multi-view representations are based on a large number of views and cannot be used in many applications such as retrieval from large databases. Multi-view representations have to deal with the following issues:

1. What is the optimal number of views?
2. How to select the optimal views?

To solve these problems some methods have been proposed for automatic selection of optimal views of an object. In order to represent an object efficiently, these methods eliminate similar views and select a relatively small number of views using an optimization algorithm. This number varies depending on the complexity of the object and the measure of expected accuracy.

Aspect graphs

In aspect-graph-based methods, the main idea is that 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 (1990). 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.

Using aspect graphs, Cyr and Kimia (2001) specified a query by a view of 3D models. A descriptor of a 3D model consists of a set of views. The number of views is kept small by clustering them and by representing each cluster by one view, which is described by a shock graph. Schiffenbauer (2001) presented a complete survey of aspect graph methods. Using shock matching, Macrini *et al*. (2002) applied indexing using topological signature vectors to implement view-based similarity matching more efficiently.

Adaptive views clustering

In the adaptive views clustering (AVC) presented by Filali Ansary *et al*. (2007), the main idea is to generate an initial set of views from the 3D model, and then reduce this set to only those that best characterize this 3D model. This idea comes from the fact that not all the views of a 3D model are of equal importance: there are some views that contain more information than others. For example, one view is sufficient to represent a sphere because it looks the same from all angles, but more than one view is needed to represent a more complex 3D model such as an aircraft.

In the AVC method, every 2D view is represented by 49 Zernike moment coefficients. To choose X characteristic views which best represent a set of $N = 320$ initial views, the authors presented a derivative of the X-means (Pelleg and Moore 2000) clustering algorithm, where, instead of using a fixed number of clusters, they used the range $[1, \ldots, 40]$, in which they chose the 'optimal' number of clusters.

In essence, the algorithm starts with one characteristic view (K equal to 1), and then adds characteristic views where they are needed. The authors took the global K-means on the data starting with characteristic views as cluster centers. They continued alternating between adding characteristic views and taking global K-means until the upper bound for the characteristic view number (40) was reached. During this process, for each K, they saved the characteristic view set.

To add new characteristic views, they used the idea presented in the X-means clustering method by Pelleg and Moore (2000). Firstly, for every cluster of views represented by a characteristic view, they selected two views that have the maximum distance in this cluster. Next, in each cluster of views, they ran local K-means (with $K = 2$) for each pair of selected views. By *local*, we mean that only the views that are in the cluster are used in this local clustering (Figure 3.35). At this point, a question arises: 'Are the two new characteristic views giving more information on the region than the original

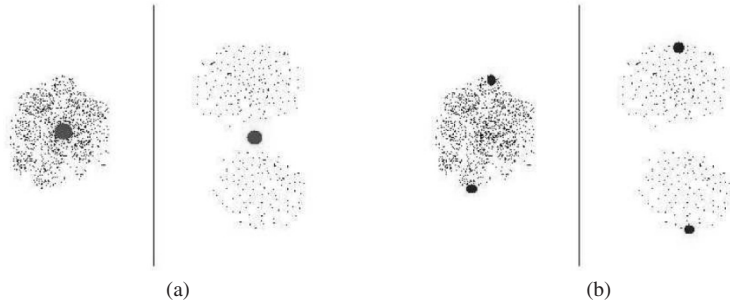

Figure 3.35 Local K-means on each part of the views cluster with $K = 2$

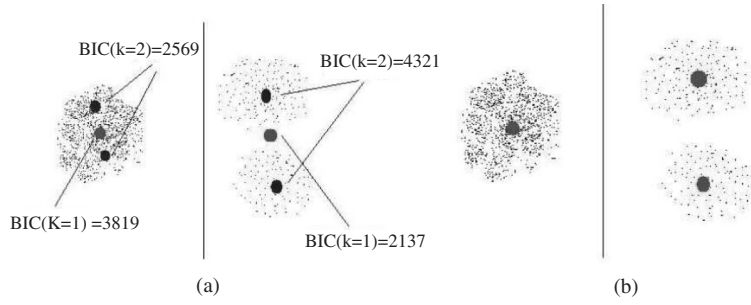

Figure 3.36 Selecting the representations (with one or two characteristic views) that have the higher Bayesian information criterion (BIC) score

characteristic view?' To answer this question, the authors used a Bayesian information criterion (BIC) (Schwarz 1978), which scores how likely the representation model (using one or two characteristic views) fits the data. Note that Figures 3.35 and 3.36 are just schematic examples, as we represent a view in a 2D space.

3.6 Normative Aspect: MPEG-7

The MPEG-7 standard, also known as the *Multimedia Content Description Interface,* aims at providing standardized core technologies allowing the description of audiovisual data content in multimedia environments (Salembier and Sikora 2002).

This is a challenging task given the broad spectrum of requirements and targeted multimedia applications, and the broad number of audiovisual features

of importance in such contexts. In order to achieve this goal, MPEG-7 will standardize:

- Descriptors (D), representations of features that define the syntax and the semantics of each feature representation.
- Description schemes (DS), schemes that specify the structure and semantics of the relationships between their components, which may be both D and DS.
- A description definition language (DDL), to allow the creation of new DS and, possibly, D and to allow the extension and modification of existing DS.
- System tools, to support multiplexing of description, synchronization issues, transmission mechanisms, file formats, etc.

The description may include still pictures, video, graphics, audio, speech, 3D models and the information about how these elements are put together in a multimedia representation. In the case of 3D data, the 3D shape spectrum descriptor has been proposed as a description of 3D shape. A multiple-views descriptor has also been developed to combine 2D descriptors representing a visual feature of a 3D model seen from different view angles. The descriptor shapes a complete 3D view-based representation of the object. Any 2D visual descriptor, such as contour shape, region shape, color or texture, can be used. The 2D/3D descriptor supports integration of the 2D descriptors used in the image plane to describe features of the 3D (real-world) objects.

3.7 Summary

The use of 3D image and model databases throughout the Internet is growing in both number and size. Exploiting the information content of digital collections poses several problems. In order to create value out of these collections, users need to find information that matches certain expectations, a notoriously hard problem due to the inherent difficulties of describing visual information content.

In recent years, as a solution to these problems, many systems have been proposed that enable effective information retrieval from digital collections of images and videos. However, the solutions proposed so far are not always effective in application contexts where the information is intrinsically 3D.

An indexing algorithm is generally a two-process mechanism:

1. The *off line process* which consists of finding invariant – to certain transformations – and robust – to noise, for example – descriptors for each 3D model in the collection.
2. The *on line process* experimented by the end-user who is querying the search engine to find a relevant 3D model in the collection.

Hence, the two key problems for 3D model indexing are to find performance descriptors and to define a similarity metric in order to compute the visual similarity between 3D models, given their descriptions.

Several approaches can be used to describe a 3D model and were presented in this chapter:

- Statistical approaches focus on the distribution of local and global descriptions of the 3D model surface: local curvatures, global distances, etc.
- Structural approaches aim at finding a high-level structure, such as a graph, from a 3D mesh.
- Transform-based approaches use transformations of the 3D shape from a 3D Euclidean space representation to another space representation.
- View-based approaches defend the intuitive idea that a 3D model is well represented by a set of characteristic views, and use well-known 2D descriptors on these views.

But before presenting these approaches, some comparison and evaluation criteria, so as to compare and evaluate the different 3D indexing methods, have been presented and discussed.

Lastly, the normative aspects were briefly explored for the MPEG-7 standard, also known as the Multimedia Content Description Interface, and how the 3D model domain is integrated in this standard for new multimedia environments.

References

Abbasi S and Mokhtarian F 2001 Affine-similar shape retrieval: application to multi-view 3D object recognition. *IEEE Transactions on Image Processing* **10**, 131–139.

Ankerst M, Kastenmuller G, Kriegel HP and Seidl T 1999 Nearest neighbor classification in 3D protein databases. *7th International Conference on Intelligent Systems for Molecular Biology (ISMB'99)*.

Antini G, Berretti S, Del Bimbo A and Pala P 2005 Retrieval of 3D objects using curvature correlograms. *Proceedings of International Conference on Multimedia & Expo*, Amsterdam.

Assfalg J, Bertini M, Del Bimbo A and Pala P 2007 Content-based retrieval of 3D objects using spin image signatures. *IEEE Transactions on Multimedia* **9**(3), 589–599.

Attene M, Biasotti S and Spagnuolo M 2003 Shape understanding by contour-driven retiling. *The Visual Computer* **19**, 127–138.

Bremer PT, Edelsbrunner H, Hamann B and Pascucci V 2004 Topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics* **10**, 385–396.

Bustos B, Kein DA, Saupe D, Schreck T and Vranić DV 2005 Feature-based similarity search in 3D object databases. *ACM Comput Surv*. **37**(4), 345–387.

Carr H, Snoeyink J and de Panne MV 2004 Simplifying flexible isosurfaces using local geometric measures. *IEEE Visualization*, pp. 497–504.

Chaouch M and Verroust-Blondet A 2006 Enhanced 2D/3D approaches based on relevance index for 3D-shape retrieval. *Shape Modeling International (SMI 2006)*, Matsushima, Japan.

Chen D, Tian X, Shen Y and Ouhyoung M 2003 On visual similarity based 3D model retrieval. *Eurographics*, pp. 223–232, Granada, Spain.

Chen JL and Stockman G 1998 3D free-form object recognition using indexing by contour feature. *Computer Vision and Image Understanding* **71**(3), 334–355.

Cole-McLaughlin K, Edelsbrunner H, Harer J, Natarajan V and Pascucci V 2003 Loops in Reeb graphs of 2-manifolds. *Symposium on Computational Geometry*, pp. 344–350.

Cyr CM and Kimia B 2001 3D object recognition using shape similarity-based aspect graph. *IEEE International Conference on Computer Vision*, pp. 254–261.

Del Bimbo A 1999 *Visual Information Retrieval*. Mogan Kaufmann Publisher, Inc.

do Carmo MP 1976 *Differential Geometry of Curves and Surfaces*. Prentice Hall.

Edelsbrunner H and Mücke EP 1990 Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Transactions on Graphics* **9**, 66–104.

Filali Ansary T, Daoudi M and Vandeborre JP 2007 A Bayesian 3D search engine using adaptive views clustering. *IEEE Transactions on Multimedia* **9**(1), 78–88.

Funkhouser T, Kazhdan M, Shilane P, Min P, Kiefer W, Tal A, Rusinkiewicz S and Dobkin D 2004 Modeling by example. *ACM Transactions on Graphics* **23**, 652–663.

Heczko M, Keim D, Saupe D and Vranic D 2002 Methods for similarity search on 3D databases. *Datenbank-Spektrum*, pp. 54–63.

Hétroy F 2005 Constriction computation using surface curvature. *Eurographics*, pp. 1–4.

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

Ip CY, Lapadat D, Sieger L and Regli WC 2002 Using shape distributions to compare solid models. *ACM Symposium on Solid Modeling and Applications*, pp. 273–280.

Johnson AE and Hebert M 1999 Using spin-images for efficient multiple model recognition in cluttered 3-D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(5), 433–449.

Kendall DG 1984 Shape manifolds, procrustean metrics and complex projective spaces. *Bulletin of London Mathematical Society* **16**, 81–121.

Koenderink JJ 1990 *Solid Shape*. The MIT Press.

Koenderink JJ and van Doorn AJ 1992 Surface shape and curvature scales. *Image and Vision Computing* **10**(8), 557–565.

Leifman G, Katz S, Tal A and Mei R 2003 Signatures of 3D models for retrieval. *4th Israel–Korea Bi-National Conference on Geometric Modeling and Computer Graphics*, pp. 159–163.

Lo CH and Don HS 1989 3-D moments forms: Their construction and application to object identification and positioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11**(10), 1053–1064.

Macrini D, Shokoufandeh A, Dickenson S, Siddiqi K and Zucker S 2002 View based 3-D object recognition using shock graphs. *IEEE International Conference on Pattern Recognition* **3**, 24–28.

Mahmoudi S and Daoudi M 2007 A probabilistic approach for 3D shape retrieval by characteristic views. *Pattern Recognition Letters* **28**(13), 1705–1718.

Nayar S, Nene S and Murase H 1996 Real-time object recognition system. *International Conference on Robotics and Automation*.

Ni X, Garland M and Hart J 2004 Fair Morse functions for extracting the topological structure of a surface mesh. *ACM Transactions on Graphics* **23**, 613–622.

Ohbuchi R, Minamitani T and Takei T 2003a Shape similarity search of 3D models by using enhanced shape functions. *Theory and Practice of Computer Graphics*, pp. 97–104.

Ohbuchi R, Nakazawa M and Takei T 2003b Retrieving 3D shapes based on their appearance. *ACM SIGMM Workshop on Multimedia Information Retrieval*, Berkeley, California, pp. 39–46.

Osada R, Funkhouser T, Chazells B and Dobkin D 2001 Matching 3D models with shape distributions. *Shape Modeling International*.

Paquet E and Rioux M 1997 Nefertiti: a query by content software for three-dimensional models databases management. *International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pp. 345–352.

Paquet E, Murching A, Naveen T, Tabatabai A and Rious M 2000 Description of shape information for 2-D and 3-D objects. *Signal Processing: Image Communication* **16**, 103–122.

Pelleg D and Moore A 2000 X-means: extending k-means with efficient estimation of the number of clusters. *International Conference on Machine Learning*, pp. 727–734.

Reeb G 1946 Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *Comptes-rendus de l'Académie des Sciences* **222**, 847–849.

Riesenhuber M and Poggio T 2000 Models of object recognition. *Nature Neuroscience* **3**, 50–89.

Rossl C, Kobbelt L and Seidel HP 2000 Extraction of feature lines on triangulated surfaces using morphological operators. *Smart Graphics, Proceedings of the 2000 AAAI Spring Symposium.* Stanford, California.

Sadjadi F and Hall E 1980 Three-dimensional moment invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2**(2), 127–136.

Salembier P and Sikora T 2002 *Introduction to MPEG-7: Multimedia Content Description Interface*. John Wiley & Sons, Inc.

Sander PT and Zucker SW 1990 Inferring surface trace and differential structure from 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**(9), 833–854.

Schiffenbauer RD 2001 A survey of aspect graphs. Technical Report TR-CIS-2001-01, CIS.

Schwarz G 1978 Estimating the dimension of a model. *Annals of Statistics* **6**, 461–464.

Stamou G and Kollias S 2005 *Multimedia Content and the Semantic Web*. John Wiley & Sons, Ltd.

Stokely EM and Wu SY 1992 Surface parametrization and curvature measurement of arbitrary 3D objects: five practical methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(8), 833–840.

Takahashi S, Ikeda T, Shinagawa Y, Kunii TL and Ueda M 1995 Algorithms for extracting correct critical points and constructing topological graphs from discrete geographical elevation data. *Computer Graphics Forum* **14**, 181–192.

Tam GKL and Lau RWH 2007 Deformable model retrieval based on topological and geometric signatures. *IEEE Transactions on Visualization and Computer Graphics* **13**, 470–482.

Tierny J, Vandeborre JP and Daoudi M 2007a Reeb chart unfolding based 3D shape signatures. *Eurographics*, Prague.

Tierny J, Vandeborre JP and Daoudi M 2007b Topology driven 3D mesh hierarchical segmentation. *IEEE International Conference on Shape Modeling and Applications (SMI'2007)*, Lyons.

Vajramushti N, Kakadiaris I, Theoharis T and Papaioanno G 2004 Efficient 3D retrieval using depth images. *6th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR'04)*, pp. 189–196.

Vandeborre JP, Couillet V and Daoudi M 2002 A practical approach for 3D model indexing by combining local and global invariants. *IEEE International Symposium on 3D Data Processing, Visualization and Transmission*, pp. 644–647.

Vranic DV 2003 An improvement of rotation invariant 3D shape descriptor based on functions on concentric spheres. *IEEE International Conference on Image Processing*, pp. 757–760.

Zaharia T and Prêteux F 2000 New content for the 3D shape core experiment: the 3D CAF data set. MPEG-7 ISO/IEC JTC1/SC29/WG11 MPEG00/M5915, Noordwijkerhout.

Zaharia T and Prêteux F 2001 3D-shape-based retrieval within the MPEG-7 framework. *SPIE Conference on Nonlinear Image Processing and Pattern Analysis XII*, vol. 4304, pp. 133–145.

Zhang C and Chen T 2001 Efficient feature extraction for 2D/3D objects in mesh representation. *International Conference on Image Processing*.