



University of West Bohemia in Pilsen
Department of Computer Science and Engineering
Univerzitni 8
30614 Pilsen
Czech Republic

3D Object Classification and Retrieval

State of the Art and Concept of Doctoral Thesis

Tomas Hlavaty

Technical Report No. DCSE/TR-2003-05
March, 2003

Distribution: public

3D Object Classification and Retrieval

Tomas Hlavaty

Abstract

At present, 3D retrieval systems that are content-based become very popular. In these systems similar models from a database are retrieved in accordance with a form of a template model. In an early era text-based retrieval systems were designed. These systems assign keywords to each model and then they retrieve similar models by comparing a similarity of the keywords to a template. There are two problems in this approach. Keywords of each model have to be assigned manually and the second more difficult problem is that each person has a little different perception. Two people can assign two different sets of keywords to the same model. Content-based retrieval system is a solution of these problems. This report deals with problems, which are related to a design of a 3D content-based retrieval system, and with their solutions.

The report consists of three main parts. The first part is interested in an issue of a feature extraction. An aim of the feature extraction is to describe a 3D model by a vector of real values characterizing a form of the model. This (multidimensional) vector may be an adequate representation of each model and therefore it can be used as an entry of a database system. The following two parts are dedicated to a design of that database system. The first part deals with types of queries, which the system may execute, and the second part deals with data structures and issues of multidimensional indexing and updating a 3D model database.

This work was supported by the Ministry of Education of the Czech Republic - project MSM23500005.

Copies of this report are available on
<http://www.kiv.zcu.cz/publications/>
or by surface mail on request sent to the following address:

University of West Bohemia in Pilsen
Department of Computer Science and Engineering
Univerzitni 8
30614 Pilsen
Czech Republic

Abstract

At present, 3D retrieval systems that are content-based become very popular. In these systems similar models from a database are retrieved in accordance with a form of a template model. In an early era text-based retrieval systems were designed. These systems assign keywords to each model and then they retrieve similar models by comparing a similarity of the keywords to a template. There are two problems in this approach. Keywords of each model have to be assigned manually and the second more difficult problem is that each person has a little different perception. Two people can assign two different sets of keywords to the same model. Content-based retrieval system is a solution of these problems. This report deals with problems, which are related to a design of a 3D content-based retrieval system, and with their solutions.

The report consists of three main parts. The first part is interested in an issue of a feature extraction. An aim of the feature extraction is to describe a 3D model by a vector of real values characterizing a form of the model. This (multidimensional) vector may be an adequate representation of each model and therefore it can be used as an entry of a database system. The following two parts are dedicated to a design of that database system. The first part deals with types of queries, which the system may execute, and the second part deals with data structures and issues of multidimensional indexing and updating a 3D model database.

Contents

- Abstract..... 1**
- Contents 2**
- 1. Introduction..... 3**
 - 1.1. Design of a content-based Retrieval System..... 4
- 2. Retrieval machine..... 6**
 - 2.1. Query interface..... 6
 - 2.2. Query processing..... 6
 - 2.3. Similarity Measuring..... 8
 - 2.3.1. Minkowsky Distance 9
 - 2.3.2. Mahalanobis distance..... 10
 - 2.3.3. Hausdorff distance 12
 - 2.3.4. Bottleneck distance 13
 - 2.3.5. Earth Movers Distance..... 13
- 3. Feature extraction 15**
 - 3.1. Directional vectors 17
 - 3.2. Mean and Gaussian curvature 19
 - 3.3. A reflective symmetry descriptor 28
 - 3.4. A spherical harmonic descriptor 33
 - 3.5. Topology matching 35
- 4. Multidimensional indexing..... 39**
 - 4.1. Spatial and point access methods 39
 - 4.1.1. Point access methods 39
 - 4.1.2. Spatial access methods..... 41
 - 4.2. Vector and metric space methods..... 42
 - 4.2.1. Vector space methods 42
 - 4.2.2. Metric space methods 44
- 5. Future work and our goals 45**
- 6. References 46**
- Appendix A – Publications 50**
- Appendix B – Stays and Lectures Abroad 51**

1. Introduction

Everyday, giga-bytes of data are generated and they are sent into all over the world by the internet. It carries large collections of various types of information, which are organized in Database Management Systems (DBMS). That database system would allow efficient browsing, searching, retrieving and updating entries in the database. Therefore research centers started to be interested in this issue and projects called “retrieval system” became very popular.

In early era of the research text-based retrieval systems were designed. It was a powerful tool in the organization of articles and some others text-based information. It was no problem to find texts according to a template of keywords in the database. Unfortunately, this retrieval system did not come in useful for visual-based information. In the late 1970’s the first text-based image retrieval system was designed. Each image was annotated by keywords and then the text-based DBMS could be used (e.g., see [42], [27], [50]). However, there are two disadvantages in this approach. The keywords of each image have to be assigned manually. If a collection of images in database would be very huge the time that a person has to spend by assigning keywords to each image is excessive. The second, more difficult disadvantage is in subjectivity of human perception. If two people would independently try to describe the same picture with a rich content the resultant sets of keywords could be different. This fact has unpleasant influence on the whole DBMS.

A solution of the mentioned disadvantages of the text-based image retrieval system is a content-base image retrieval system. In this system the keywords are substituted by own visual content, for which are used techniques as color histogram, segmentation of the image, texture pattern, edge detection, etc. The first systems were proposed for a collection of images in the early 1990’s. Nowadays many image retrieval systems exist as QBIC [37], PhotoBook [39], WebSeek [48] etc. and many issues connected to these systems have been dedicated (e.g., see [36], [54], [47]).

Perhaps someone could say that we still concern with the image retrieval systems but according to our topic we may concern with 3D retrieval systems. It is true. However, many research studies have been written about the image retrieval systems and because both topics are very similar majority outcomes are transmittable for both systems. The 3D data started to use

later than the images in the computer graphics. Their popularity became large when various programs based on the 3D graphics were proposed such as CAD systems, cartography systems, computer vision and robotics systems etc. Nowadays, they are used in many branches of industry. Their everyday usage carries large 3D data collections, which are faced in issues of the object recognition, classification and retrieval above all. These research areas in spite of differences of their practical usage are very close one to another. This report is dedicated to problems related to 3D retrieval systems, but many parts are shared in all mentioned issues.

1.1. Design of a content-based Retrieval System

The aim of all content-based retrieval systems is to minimize human indispensability during creating and updating the DBMS. The recognition and the classification of the objects in the database would only be based on the visual information.

The architecture of those systems can be proposed in many variations. For all that, the basic structure is still very similar. An example of that architecture is shown in Figure 1.1.1.

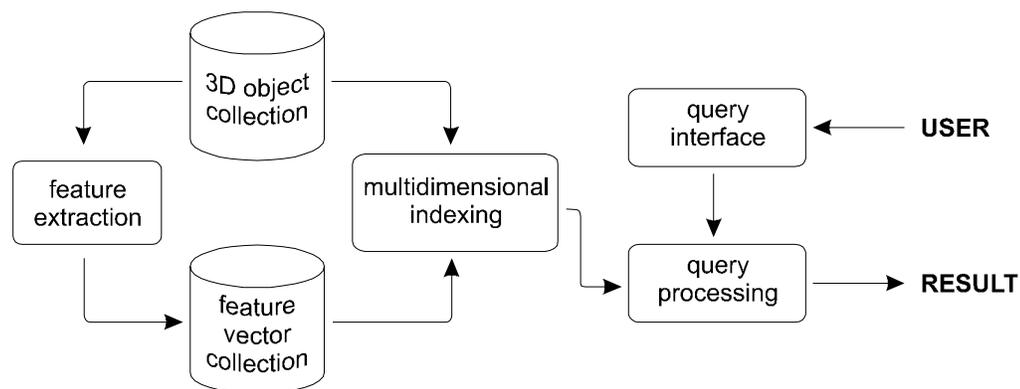


Figure 1.1.1: An architecture of the 3D content-based Retrieval System

In the architecture there are two databases. The first one represents real 3D objects saved in the database and the second one represents the so called feature vector collection, which feature extraction process automatically generates.

If the image and 3D content-based retrieval systems were compared, the feature extraction just would be the system block that would be the most different. The main job of the feature extraction is to describe the content of each entry in the first database by a vector of real values. Then these feature vectors are saved in the second database. Many content-based description techniques exist for image retrieval systems such as using the color histogram, the

texture patterns, shape representation, segmentation etc. and many comprehensive surveys was written about them (e.g., [42], [36], [54], [52]). Unfortunately, their asset in the feature extraction of 3D objects is minimal. For all that, several techniques were proposed and their detail descriptions are possible to find in a following part of this report.

The next important part in the architecture is a retrieval machine. It consists of two blocks: a query interface and a query processing. The task of the query interface is only to transfer a request of a user into an input format for the query processing. After receipt of the required query in the query processing, the query is processed and from the database are selected the fitting entries. The detail survey of this area is described in another following part of the report.

The multidimensional indexing is the last important part in the architecture. It serves as a bridge between the databases and the retrieval machine. The entries of the database (with the feature vectors) have to be organized in an efficient structure that ensures fast and infallible query processing. It is just the job of multidimensional indexing. The survey of the various structures is possible to find in the last part of this report.

2. Retrieval machine

The purpose of the retrieval systems is to retrieve items of the database which are relevant to a piece of a query. The core of that whole system forms the query machine and this is just its main task. The entries are selected from the database according to a query that is defined by a user and they are returned as an answer on the required query. The query machine further can be divided into two independent parts: a query interface and a query processing (see Figure 1.1.1).

2.1. Query interface

The queries have to be set to the system by user friendly way. This is the task of the query interface. It converts user requirements to a format for query processing. The design of the query interface can differ a lot. In some cases only a set of parameters that are regulated manually can be ideal. On the other hand, a sketch or a pattern object can be ideal. It depends on the facts for which exact purposes the retrieval system should serve and which kind of users should handle it.

2.2. Query processing

The second most important part of the retrieval system (after the feature extraction) is the query processing. It has to process all the queries from query interface and to send resultant items from the database to the output as an answer on the queries. The types of the queries are various and it is dependent on the individual usage which the given system supports. An enumeration of the most common types with a short description is following (according to [25], [22],[10]):

- **Exact Match Query (EMQ):** Find all database objects that have exactly the same spatial extent as the spatial query object o .
- **Point Query (PQ):** Find all database objects that overlap the query point p .
- **Window Query (WQ), Range Query:** Find all database objects that have at least one common point with a d -dimensional query window w .
- **Intersection Query (IQ), Region Query, Overlap Query:** Find all database objects that have at least one common point with a query object o .
- **Enclosure Query (EQ):** Find all database objects that enclose a query object o . An object a is said to enclose object b if any point of a is a part of object b .
- **Containment Query (CQ):** Find all database objects that are enclosed by a query object o .
- **Adjacency Query (AQ):** Find all database objects that are adjacent to a query object o . Two objects are said to be adjacent if they have common boundaries, but the one does not enclose the other.
- **k-Nearest Neighbor Query (k-NNQ):** Find k database objects that have a minimum distance from a query object o . Distance between spatial objects is usually defined as the distance between their closest points.
- **Within-distance Query (α -cut):** Find all database objects that have the distance less than α from a query object o . Distance between spatial objects is usually defined as the distance between their closest points.
- **Spatial Join:** Find all (a,b) database pairs that evaluate a spatial predicate θ to true where a database object is from the R database and b database object is from the S database.

Evidently, this is not an exhaustive enumeration. Many other queries that could be ranked into no mentioned group are possible to define. It only depends on the desiderative capabilities of the retrieval machine. Our goal is to design a content-based retrieval machine. Therefore the most important types of queries in this case are the Window query (WQ), the k-Nearest Neighbor Query (k-NNQ) and Within-distance Query (α -cut).

However, the capabilities and quality of the retrieval machine do not depend only on one property. The publication [10] contains a description of the other three properties that characterize the retrieval machine. They are following:

- **Exhaustiveness:** If all the database items that satisfy the queries are retrieved the query processing is exhaustiveness.
- **Correctness:** If all the returned items satisfy the queries the query processing is correctness.
- **Determinism:** If the same results are returned for the same query every time the query processing is deterministic.

Perhaps the definitions of the exhaustiveness and of the correctness seem similar but they have different meaning. If the query processing were not exhaustive, a database item that satisfies the query would not belong to the resultant set. If the query processing were not correct, a database item that does not satisfy the query would belong to the resultant set.

It is obvious that design of the retrieval system satisfying all properties is not easy. For all that we will strive to design a content-based retrieval system that would fit all the mentioned properties.

2.3. Similarity Measuring

In the chapter about query processing a selection of the similar object is still mentioned, but the question how to measure a resemblance of objects or feature vectors is not answered.

In general, the systems rely that the feature vectors can be represented as points in the n -dimensional space. At the worst the feature vectors are represented as n -dimensional objects and then it is better to replace them by some primitive objects such as n -dimensional rectangles or spheres which enclose the origin objects. Those primitive objects can be represented by a centroid (n -dimensional point) and some parameters and this representation is similar to the general cases.

Let us assume that S is a set of all n -dimensional points (or objects). The resemblance of objects can be measured by the function $d: S \times S \rightarrow \mathbf{R}^+ \cup \{0\}$. If the function satisfies following conditions for all $x, y, z \in S$ then it is called *metric* function [51], [52]:

(i.) **identity:** $d(x,x) = 0$,

(ii.) **uniqueness:** $d(\mathbf{x}, \mathbf{y}) = 0$ implies $\mathbf{x} = \mathbf{y}$,

(iii.) **positive:** $d(\mathbf{x}, \mathbf{y}) > 0$,

(iv.) **triangle inequality:** $d(\mathbf{x}, \mathbf{y}) + d(\mathbf{x}, \mathbf{z}) \geq d(\mathbf{y}, \mathbf{z})$.

Sometimes, instead of the last condition its alternative is taken: $d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) \geq d(\mathbf{x}, \mathbf{z})$, but this alternative condition does not imply symmetry. If the symmetry metric function is required, the next condition has to be appended: $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$. Note the symmetry implies from the first and the last condition in the origin conditions.

Any distance functions do not satisfy all mentioned condition. For example a special case is a *semi-metric* function that does not satisfy (iv.) condition of the triangle inequality, or a *pseudo-metric* function that does not satisfy (iii.) condition of the positive.

Often it is also desirable invariance of the distance function under a chosen group of transformations G . It can be defined by the following conditions:

(v.) **invariance:** $d(g(\mathbf{x}), g(\mathbf{y})) = d(\mathbf{x}, \mathbf{y})$ for all transformations $g \in G$.

An extensive theory can be written about distance functions. However, this issue is not the main topic of this report. Therefore, only a short list of the most useful distance function is introduced here. The more information is possible to find for example in [51], [53], [52],[10].

2.3.1. Minkowsky Distance

Perhaps it is the most popular distance measuring. For two points $\mathbf{x}, \mathbf{y} \in \mathbf{R}^d$, the general L^p distance is defined as:

$$L^p(\mathbf{x}, \mathbf{y}) = \left[\sum_{i=1}^d |\mathbf{x}_i - \mathbf{y}_i|^p \right]^{\frac{1}{p}}. \quad (2.3.1)$$

This general equation usually does not use in practice. Ordinarily, the parameter p is fixed on the following values:

- **Manhattan distance** or city-block ($p = 1$):

$$L^1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^d |\mathbf{x}_i - \mathbf{y}_i| \quad (2.3.2)$$

- **Euclidian distance** ($p = 2$):

$$L^2(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^d (\mathbf{x}_i - \mathbf{y}_i)^2} \quad (2.3.3)$$

- **Chebychev distance** ($p = \infty$):

$$L^\infty(\mathbf{x}, \mathbf{y}) = \max_{i=1}^d |\mathbf{x}_i - \mathbf{y}_i| \quad (2.3.4)$$

The difference among the individual types of the distance function is better shown in Figure 2.3.1 (taken from [10]). The unit spheres in the mentioned metric spaces are drawn there.

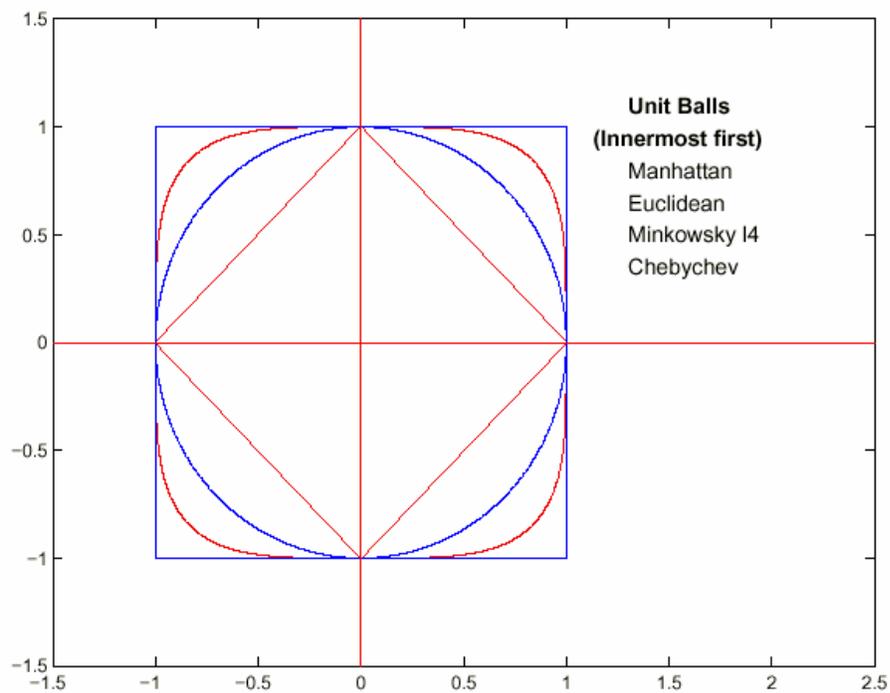


Figure 2.3.1: The unit spheres under Manhattan (L^1), Euclidean (L^2), Minkowsky L^4 and Chebychev (L^∞) distance.

2.3.2. Mahalanobis distance

This distance is rather used for a classification (i.e., a class for a object is selected according to the closest distance of the object to the given class). At first, assume a simple case:

Let $x_i = \{x_{1,i}, x_{2,i}, \dots, x_{N,i}\}$ be a collection of N examples of the feature i and $x_j = \{x_{1,j}, x_{2,j}, \dots, x_{N,j}\}$ be a corresponding collection of N examples of the feature j (it means that $x_{k,i}$ and $x_{k,j}$ are two features of the same pattern).

The features can be characterized by *mean values* (it is calculated as the average of representative values of the given class):

$$m_i = \frac{1}{N} \cdot \left(\sum_{k=1}^N x_{k,i} \right), \quad (2.3.5)$$

and terms $c_{i,j}$ of a so called *covariance matrix* \mathbf{C} that measures a tendency to vary between two features (it is the average of the products of the deviations of representative values from their means):

$$c_{i,j} = \frac{1}{N-1} \cdot \left\{ \sum_{k=1}^N (x_{k,i} - m_i) \cdot (x_{k,j} - m_j) \right\}, \quad (2.3.6)$$

where $x_{k,i}$ and $x_{k,j}$ are representative values of the feature i and j , respectively, and m_i, m_j are their mean values (see the equation (2.3.5)). The physical meaning of the covariance is illustrated in Figure 2.3.2, where the dependence between features x_i and x_j for different values of the covariance is shown.

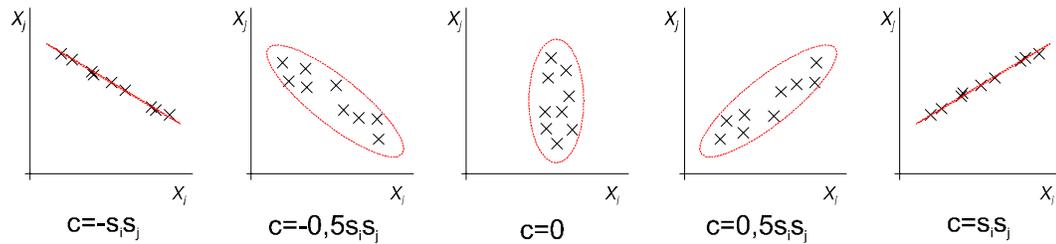


Figure 2.3.2: An illustration of the meaning of the covariance c between two features (s_i and s_j are the standard deviations of the given features).

The value s_i represents a so called *standard deviation* (it is a measure of the size of the cluster) that is defined as:

$$s_i = \sqrt{\frac{1}{N-1} \cdot \left\{ \sum_{k=1}^N (x_{k,i} - m_i)^2 \right\}}. \quad (2.3.7)$$

Suppose now that we have an n -dimensional feature vector \mathbf{x} and a corresponding mean vector \mathbf{m}_X with covariance matrix \mathbf{C}_X of a class X . Then the Mahalanobis distance r of the vector \mathbf{x} from the class X can be computed by the formula:

$$r^2 = (\mathbf{x} - \mathbf{m}_X)^T \mathbf{C}_X^{-1} (\mathbf{x} - \mathbf{m}_X), \quad (2.3.8)$$

where \mathbf{C}_X^{-1} denotes inverse matrix of the covariance matrix and the superscript T denotes vector transposing.

Finally, note this distance can solve problems caused by poorly scaled or highly correlated coefficients of a vector.

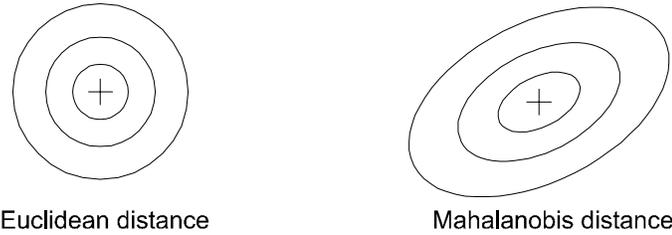


Figure 2.3.3: An illustration of contours with the constant Euclidean distance and Mahalanobis distance (for a given covariance matrix).

2.3.3. Hausdorff distance

Sometimes, the number of points of set of points A and B is not correspondent. In this case the Housdorff distance is commonly used. It is defined as:

$$h(A, B) = \min_{a \in A} \max_{b \in B} d(a, b), \tag{2.3.9}$$

where max and min denotes maximum and minimum over all elements of the given set. The function $d(a, b)$ is an underlying distance function (typically, it is Euclidean distance).

In other words, the Hausdorff distance is defined as the lowest upper bound over all points in A to B , where distance between two points is measured by the $d(a, b)$ function. Unfortunately, this distance has two uncomfortable properties. It is very sensitive to noise and it is not metric distance (the condition of the triangle inequality fails). More detail information is presented for example in [52], [53].

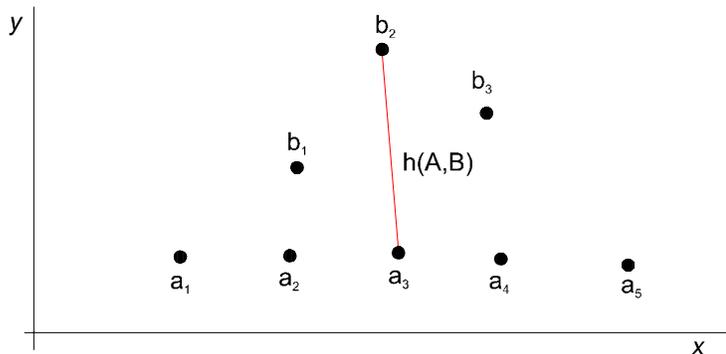


Figure 2.3.4: An simple example of the Hausdorff distance for A and B sets of points in E^2 (as the measuring function $d(a, b)$ is used Euclidean distance).

2.3.4. Bottleneck distance

The disadvantage of Hausdorff distance is in mapping that is defined as an association of points from A to its nearest neighbor in B . It does not always have to be a one-to-one mapping. In the case where this correspondence is needed, i.e., where each point from A is just matched by only one point from B , there is better to use the Bottleneck distance. It is defined as [52], [53]:

Let A and B be two point sets of size N and $d(\mathbf{a}, \mathbf{b})$ a distance between two points. The bottleneck distance is the minimum of the maximum distance $d(\mathbf{a}, f(\mathbf{a}))$ over all one-to-one mapping f between A and B .

2.3.5. Earth Movers Distance

This next type of distance function is based on likening the distance measures to minimal amount of work needed to transform of earth or mass from one position to the other. For example, when we suppose two distribution and we would like to measure their distance by Earth Movers Distance (EMD), then one distribution can be seen as a mass of earth properly spread in space, the other as a collection of holes in that same space. The computation of the EMD is possible to see as a minimal distance that is needed to transport the earth into the holes (a quantity of the mass of the earth and the size of the hole is represented by weight values for the given distribution). Formally, the exact definition is following [23]:

Let $d(\mathbf{x}, \mathbf{y})$ be a ground distance function and $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m\}$, $B = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n\}$ be two weighted point sets such that $\mathbf{a}_i = \{(\mathbf{x}_i, w_i)\}$, $i = 1, \dots, m$ and $\mathbf{b}_j = \{(\mathbf{y}_j, u_j)\}$, $j = 1, \dots, n$, where $\mathbf{x}_i, \mathbf{y}_j \in \mathbf{R}^k$ with $w_i, u_j \in \mathbf{R}^+ \cup \{0\}$ being its corresponding weight. Let also W and U be the total weights of set A and B , respectively:

$$W = \sum_{i=1}^m w_i, U = \sum_{i=1}^n u_i, \quad (2.3.10)$$

Let us denote f_{ij} the elementary flow of the weight (of the earth) from \mathbf{x}_i to \mathbf{y}_j , over the elementary distance $d(\mathbf{x}_i, \mathbf{y}_j)$, then a set of all feasible flows $\bar{F} = [f_{ij}]$ is defined by the following constraints:

- $f_{ij} \geq 0$, $i = 1, \dots, m$, $j = 1, \dots, n$
- $\sum_{j=1}^n f_{ij} \leq w_i$, $i = 1, \dots, m$
- $\sum_{i=1}^m f_{ij} \leq u_j$, $j = 1, \dots, n$

- $\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \min(W, U)$

The Earth Movers Distance $EMD(A, B)$ is defined as the minimum total cost over all possible \bar{F} normalized by the weight of the lighter set:

$$EMD(A, B) = \frac{\min_{\bar{F} \in F} \sum_{i=1}^m \sum_{j=1}^n f_{ij} d_{ij}}{\min(W, U)}, \quad (2.3.11)$$

Generally, that distance function does not obey the conditions of the positivity and the triangle inequality (see above) and it is very computationally expensive. For all that, some nice properties are satisfied when some additional conditions are held. The other detail information with some derived function, such as *Proportional Transportation Distance*, is possible to find for example in [23], [16], [51], [53], [52].

3. Feature extraction

As it was remark above, the feature extraction is the most important part of the retrieval system. This part of the report is dedicated to this issue and several methods that were proposed are described here.

At first, something about the feature extraction from 3D objects would be known. The objects (or models¹) can be described in several representations. Perhaps the most popular are polyhedral meshes or volumetric data that can be obtained from a scanner, or parametric or implicit equations that can be obtained from a mathematic or 3D model system. Note it is not a complete enumeration how objects can be described or defined. Other representations are for examples superquadrics or generalized cylinders (e.g., see [9]).

Several possibilities of the description exist and that we will use it only depends on the way of obtaining data and our decision. Our choice also determines another parameters that can be obtained from the given description, such as normal, gradient, etc. These parameters just can be used in the feature extraction (perhaps the description by mathematical equations seems as the best solution in our case but it is not always possible). The final feature vector and a method of the feature extraction would have the following properties or at least they would be optimized them as good as it is possible:

- quick to compute
- concise to store
- easy to index
- invariant under transforms (translation, rotation, scale)
- insensitive to noise and small features
- independent of 3D representation

¹ The words the object and the models have the same sense here. Generally, it means 3D data describing the items in the database of the retrieval system.

- robust to arbitrary topological degeneracy
- discriminating of shape similarities and differences

Naturally, a method for feature extraction that would be ideal and that would completely describe the surface of a 3D object by a feature vector is impossible to find. For all that several methods exist that are based on different mathematical theories. Tangelder [51] divides them into the following classes:

- **Manufacturing features:** Solid models which were made by a manufacturing process, can be described by features representing the process of the manufacture. More detail description of this approach is presented for example in [45], [11].
- **2D view based features:** At present, the issue of image feature extraction is examined very well. This knowledge can be used in 3D case where similar objects can be searched according some 2D sketches. The 3D models are usually compared with the similarity of their 2D view. The description of that content-based method is possible to find e.g., [35], [14].
- **Histogram based features:** This kind of features is based on comparing histograms or distributions encoding properties of 3D models. The feature extraction can be based on many various properties, such as mapping the surface curvature to the unit sphere [46], using moment-based classifier [17], using reflective symmetry description [31], etc. The individual methods are described later. The similarity is usually evaluated by a metric that measures distances between distributions.
- **Topology based features:** The features based on topological properties of 3D objects can be used in matching very well. This is proofed, e.g., in [24] where the method uses Reeb graph based on geodetic distances to encode the topology of 3D objects.
- **Volume based features:** The volume information is the next property that can serve to description of 3D models. It is possible to find an example of a method in [38] where calculating a volumetric error is used for matching similarity.
- **Deformation based features:** Some image retrieval systems use the measuring of a deformation of 2D shapes in which the amount of deformations to register the shape is measured, e.g., [2], [12]. Unfortunately, these methods are very difficult to apply

for 3D shape matching. For all that, it is good to know that these methods exist at least for 2D case.

The following subsections are dedicated to descriptions of several methods. Some methods from the groups of histogram and topology based features mainly are described. Specially, the methods based on curvatures and Reeb graph are described in more detail. It is because I plan to use them in the proposition of my methods that would be based on their properties. The remainder methods are introduced here to obtain point of view how easily or difficulty a feature vector can be generated.

3.1. Directional vectors²

This method is proposed for models represented as triangle meshes. The own principle of the method is not difficult. At first, the vertices of the triangle mesh are transformed into the form guaranteeing scale, rotate and translate invariance. A modification of the Principal Component Analysis (shortly PCA, e.g., see [55], [51]) is employed to accomplish this aim. Although the PCA exactly does not guarantee the invariance, the resultant form is independent as much as possible. The model is transformed into the PCA coordinate system where the center of the mass (represented by mean vectors computed over all points of the mesh) is in the origin and eigenvectors of the covariance matrix are orthogonal and coincide with the vectors of the new coordinate system (see low). Now let us suppose that we have a set of k directional vectors $\{\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \dots, \hat{\mathbf{u}}_k\}$. Then distances from the origin to the surface of the mesh in accord with the individual directional vectors (in the PCA coordinate system) can be measured and these distance just form a feature description of the 3D model in this case.

Assume that a 3D model is represented by a mesh including a set of N vertices $V = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N\}$, where $\mathbf{v}_i \in \mathbf{R}^3$ and is represented as a column vector. Let the mean vertex \mathbf{m} be defined as:

$$\mathbf{m}^{(0)} = \frac{1}{N} \sum_{k=1}^N w_k \mathbf{v}_k^0. \quad (3.1.1)$$

It is the average over all vertices that are weighted by a weight w_k associated to the given vertex \mathbf{v}_k . The weights of the vertices are computed by the formula:

² Figures and equation in this chapter are taken from [55]. This retrieval system is possible to find on the web page: <http://dbvis.fmi.uni-konstanz.de/research/projects/SimSearch3D/>.

$$w_k = \frac{N \cdot S_k}{3S}, \quad k = 1, \dots, N, \quad (3.1.2)$$

where S is the surface area of the mesh and S_k is the sum of surfaces of all triangles including the vertex \mathbf{v}_k . Note the sum of the weights of all points in the mesh is equal to N (i.e. the number of points).

The modified principal component analysis (PCA) is based on the computation of eigenvalues and eigenvectors of the covariance matrix \mathbf{C} that is defined by the formula:

$$\mathbf{C}^{(0)} = \frac{1}{\sum w_k} \sum_{k=1}^N w_k (\mathbf{v}_k^{(0)} - \mathbf{m}^{(0)}) (\mathbf{v}_k^{(0)} - \mathbf{m}^{(0)})^T. \quad (3.1.3)$$

After finding eigenvalues and eigenvectors a matrix \mathbf{A} can be formed. The columns of the matrix are composed of the normalized eigenvectors that are sorted in the rowing order of the non-negative values on the diagonal of the matrix (this procedure guarantees the unique order of the found eigenvectors). Finally, the all vertices are transformed into a new so called PCA *coordinate system* by the help of the matrix \mathbf{A} and the mean vector \mathbf{m} , where the normalized eigenvectors form the coordinate system (see Figure 3.1.1, Figure 3.1.2). The formula for the transformation is following:

$$\bar{\mathbf{v}}_k = \mathbf{A}(\mathbf{v}_k - \mathbf{m}), \quad k = 1, \dots, N. \quad (3.1.4)$$

That triangle mash in PCA coordinate system is invariant to translation and rotation (as much as possible).

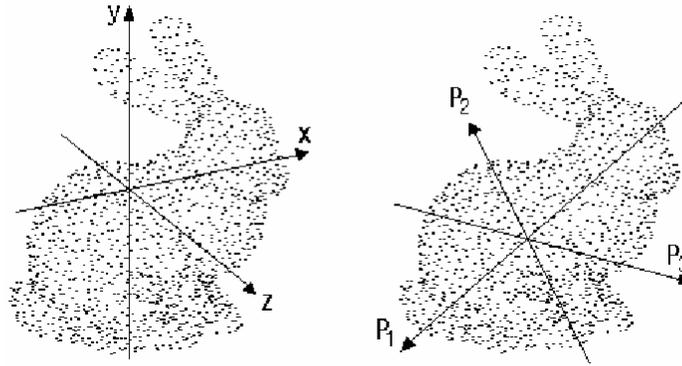


Figure 3.1.1: The Principal Component Analysis

Now we can pass on a feature extraction. Assume that a set of k directional vectors $\{\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \dots, \hat{\mathbf{u}}_k\}$ is defined. Then a feature vector can be represented as the distances (in PCA coordinate system) from the origin to the surface of the model in the direction of the given vectors (see Figure 3.1.2).

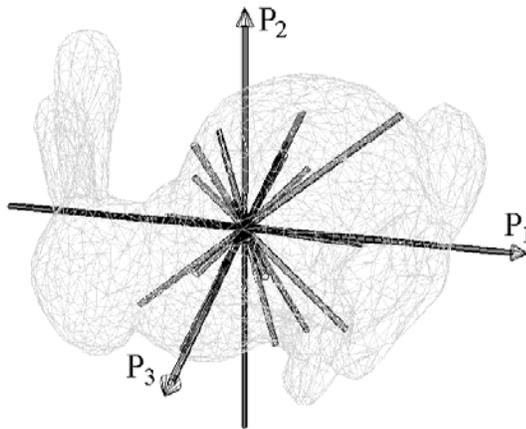


Figure 3.1.2: Extraction of Shape descriptors by a set of rays

However, this feature vector is not invariant to scale, therefore a procedure that this invariance will be guarantee should be defined. In the cited report all the elements of the feature vector are simply divided by the value of the maximal element. That feature vector is finally invariant to scale, translation and rotation and can be used for a description of 3D models.

3.2. Mean and Gaussian curvature

A curve in the space can be described by several different ways. A unique description of the curve is by a curvature 1k and torsion 2k that allow to describe an arbitrary shape curve exhaustively except a position of the curve in the space).³ A similar description for a surface is formulated in this section. However, a theory about surfaces and their curvatures is very wide therefore only basic facts needed to define curvatures are shortly described here. A more detail theory is possible to find in a book dedicated to the differential geometry (e.g. [3], [7], [41]).

Suppose that a formula describing a surface of the model is defined in a parametric form⁴:

Definition 3.2.1: Let parameters u and v be defined in a region $\Omega \subseteq E^2$ (of the type A, see [41]). Then a surface can be described by the parametric formula:

³ A mathematical background of the computation of the curvature and the torsion is possible to find in a mathematical book dedicated to curves, Frenet formulas and a parametrization of the curve.

⁴ Various definitions of the formula describing a surface exist. We only will think about parametrical definition. All definitions and theorems are cited from [3], [7], [41].

$$\mathbf{r}(u, v) = (x(u, v), y(u, v), z(u, v)), [u, v] \in \Omega, \quad (3.2.1)$$

where all the following requirements are fulfilled:

- The functions x, y, z are continual and have piecewise continuous derivatives of the first order in Ω .
- The matrix:

$$M = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial y}{\partial u} & \frac{\partial z}{\partial u} \\ \frac{\partial x}{\partial v} & \frac{\partial y}{\partial v} & \frac{\partial z}{\partial v} \end{pmatrix} \quad (3.2.2)$$

is everywhere (with the exception at most a finite number of points) of rank $h=2$, i.e. at last one of its determinants of order two is non-vanishing.

On that defined surface an arbitrary curve can be described. In the following text the parametric definition of a curve on the surface and definitions of fundamentals forms characterizing the curve on the surface are cited.

Definition 3.2.2: The equations:

$$u = u(t), v = v(t), t \in \omega \quad (3.2.3)$$

express parametrically a curve on a surface provided that the functions defined in an interval ω have the following properties:

- The functions are continual and posses continuous first derivatives which do not vanish simultaneously.
- All points lie for all $t \in \omega$ in the domain Ω of the surface (see **Definition 3.2.1**).
- The elements of the matrix M (defined in **Definition 3.2.1**) vanish simultaneously at a finite number of points at most.

Definition 3.2.3: The square of the differential of the arc of a curve $u = u(t), v = v(t)$ on a surface $\mathbf{r} = \mathbf{r}(u, v)$ is given by formula:

$$ds^2 = E du^2 + 2F du dv + G dv^2 \equiv I. \quad (3.2.4)$$

This quadratic differential form is called the *first fundamental form* of the surface, ds is called the *element of the art* and E, F, G are called the *first fundamental coefficients*.

$$\begin{aligned}
E &= \mathbf{r}_u \cdot \mathbf{r}_u = \left(\frac{\partial x}{\partial u}\right)^2 + \left(\frac{\partial y}{\partial u}\right)^2 + \left(\frac{\partial z}{\partial u}\right)^2, \\
F &= \mathbf{r}_u \cdot \mathbf{r}_v = \frac{\partial x}{\partial u} \cdot \frac{\partial x}{\partial v} + \frac{\partial y}{\partial u} \cdot \frac{\partial y}{\partial v} + \frac{\partial z}{\partial u} \cdot \frac{\partial z}{\partial v}, \\
G &= \mathbf{r}_v \cdot \mathbf{r}_v = \left(\frac{\partial x}{\partial v}\right)^2 + \left(\frac{\partial y}{\partial v}\right)^2 + \left(\frac{\partial z}{\partial v}\right)^2,
\end{aligned} \tag{3.2.5}$$

Definition 3.2.4: Along a curve $u = u(s)$, $v = v(s)$ (s being the arc of the curve) on a surface $\mathbf{r} = \mathbf{r}(u,v)$ the formula:

$$-\mathbf{dr} \cdot \mathbf{dn} = L du^2 + 2M du dv + N dv^2 \equiv II, \tag{3.2.6}$$

where

$$\begin{aligned}
L &= -\mathbf{r}_u \cdot \mathbf{n}_u, \\
2M &= -(\mathbf{r}_u \cdot \mathbf{n}_v + \mathbf{r}_v \cdot \mathbf{n}_u), \\
N &= -\mathbf{r}_v \cdot \mathbf{n}_v,
\end{aligned} \tag{3.2.7}$$

is called the *second fundamental form* of the surface. Coefficients L, M, N are called the *second fundamental coefficients* (\mathbf{n} is unit normal vector of the surface and $\mathbf{n}_u, \mathbf{n}_v, \mathbf{r}_u, \mathbf{r}_v$ denote the partial derivatives $\partial \mathbf{n} / \partial u, \partial \mathbf{n} / \partial v, \partial \mathbf{r} / \partial u, \partial \mathbf{r} / \partial v$, respectively).

Definition 3.2.5: a regular point of surface, in which is:

$$LN - M^2 > 0, LN - M^2 = 0, LN - M^2 < 0, \tag{3.2.8}$$

is called *elliptic point*, *parabolic point* or *hyperbolic point* of the surface, respectively (see Figure 3.2.1).

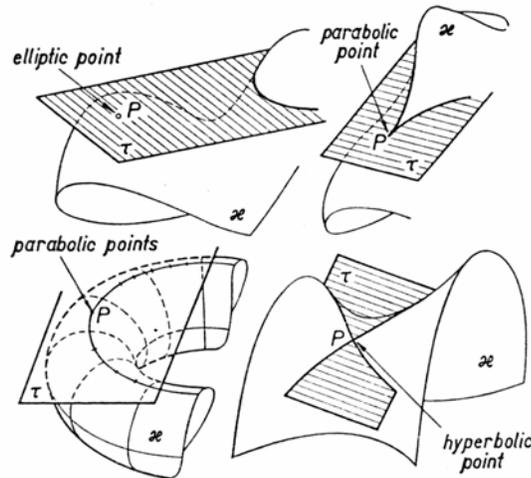


Figure 3.2.1: Examples of elliptic, parabolic and hyperbolic points (taken from [41]).

Now, we can speak about a surface, a curve on the surface and fundamental forms characterizing the surface. From the coefficients of first or second fundamental forms some properties can be derived (note that the first and second fundamental forms define a shape of the surface exhaustively except of its position in the space).

If a property is only given by first fundamental coefficients it is classified as a property describing a so called *inner geometry of the surface*. That property is constant when a surface is spread on another surface (e.g. when a tube surface is spread on a plane). Properties derived from second fundamental coefficients are classified as properties describing a so called *outer geometry of the surface* and the following text is dedicated them.

The following theorems formulate equations calculating the radius of curvature of the curve on the surface and they just serve as a mathematical background for definition of principal radii of curvature and so called Dupin's indicatrix.

Theorem 3.2.1: All curves on a surface which pass through a regular point **P** of the surface and have the same osculating plane at **P** have also the same curvature at **P**. The radius of curvature of the curve of the section cut by a plane passing through a point **P** of the surface $\mathbf{r}(u,v)$ is :

$$r = \frac{E du^2 + 2F du dv + G dv^2}{|L du^2 + 2M du dv + N dv^2|} \cos \vartheta, (II \neq 0). \tag{3.2.9}$$

where ϑ is the angle between the plane of section and the normal to the surface at P (see Figure 3.2.2).

Definition 3.2.6: A curve cut on a surface by a plane that contains a normal to the surface is called a *curve of normal section* and radius of curvature and curvature have prefix *normal*.

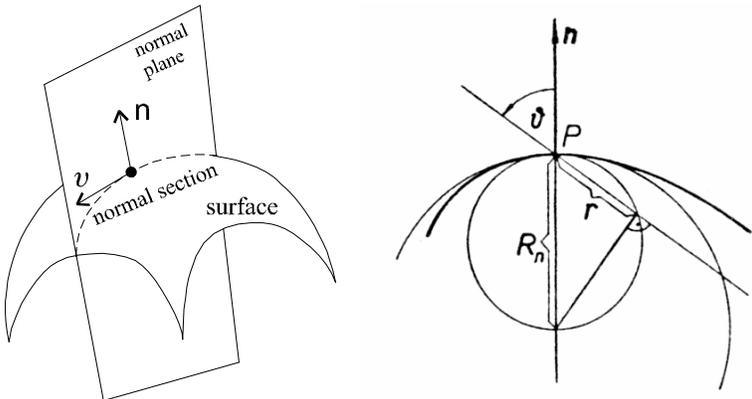


Figure 3.2.2: An illustration of the Meusnier theorem (right figure taken from [41]).

Theorem 3.2.2: (Meusnier's Theorem). A curve of section passing through a regular point \mathbf{P} on a surface, has at \mathbf{P} a radius of curvature which is the orthogonal projection of the radius of curvature R_n of the curve of normal section (into the osculating line at \mathbf{P}):

$$r = R_n \cos \mathcal{G}. \quad (3.2.10)$$

Remark the Meusnier's theorem says if a regular point \mathbf{P} on the surface and a tangent line \mathbf{v} in this point are given, then for all curves on the surface that have the tangent line \mathbf{v} in the point \mathbf{P} the center of the radius of curvature lie on the circle with the radius R_n . The circles of curvature of all these curves lie on the sphere of radius R_n (see Figure 3.2.2).

When we summarize previous theorems the normal curvature in a given direction at a point \mathbf{P} of a surface $\mathbf{r}(u,v)$ can be computed by the equation:

$$\frac{\varepsilon}{R_n} = \frac{L du^2 + 2M du dv + N dv^2}{E du^2 + 2F du dv + G dv^2}, \varepsilon = \pm 1. \quad (3.2.11)$$

Note the sign of ε has a geometrical meaning as will be shown later.

Definition 3.2.7: The curves of normal section of a surface for which the corresponding normal curvatures have extreme values are called the *principal curves of normal section* and their radii of curvature R_1 and R_2 the *principal radii of curvature*, at the point considered on the surface.

Theorem 3.2.3: (Euler's Theorem). The curvature $1/R_n$ of a curve of normal section at a regular point of a surface is given by the formula:

$$\frac{1}{R_n} = \frac{\cos^2 \delta}{\varepsilon R_1} + \frac{\sin^2 \delta}{\varepsilon R_2}, \varepsilon = \pm 1, \quad (3.2.12)$$

where δ is the angle between the plane of the curve of normal section and the plane of the first principal curve of normal section.

An importance of the previous definitions and theorems is introduced in the following text. Suppose a tangent plane at a point \mathbf{P} of a surface is known. On the tangent plane is defined the cartesian coordinate system such that x and y axis is in the tangent line of the first and second principal curve of normal section at \mathbf{P} , respectively. If the point \mathbf{P} is elliptic, hyperbolic or parabolic⁵, let us construct in the tangent plane at \mathbf{P} , the ellipse, two hyperbolas or two parallel straight lines given by the equations:

⁵ In this case, we suppose that second principal curvature is equal to zero.

$$\frac{x^2}{R_1} + \frac{y^2}{R_2} = 1, \quad \pm \frac{x^2}{R_1} \mp \frac{y^2}{R_2} = 1, \quad \frac{x^2}{R_1} = 1, \quad (3.2.13)$$

respectively as is shown in Figure 3.2.3. The length of the radius vector ρ of each point of the ellipse, hyperbolas or the straight lines is the square root of the radius of curvature of the curve of normal section whose plane passes through the radius vector ρ at the point **P**. The angle δ between ρ and the first principal direction is the same as in the equation (3.2.12).

Definition 3.2.8: The ellipse, two hyperbolas or two parallel straight lines (3.2.13) in the tangent plane at a regular point of a surface is called the *Dupin's indicatrix*.

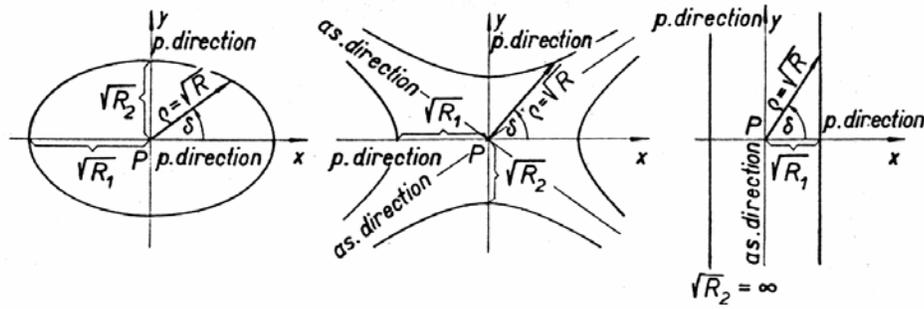


Figure 3.2.3: An illustration of the Dupin's indicatrix (taken from [41]).

The radii of the principal curvatures at a point on a surface are dominant elements in definitions of Gaussian and mean curvatures:

Definition 3.2.9: (Gaussian curvature). The product of principal curvatures at a regular point **P** on the surface is called *Gaussian curvature* K .

Note the Gaussian curvature can be expressed by the coefficients I. And II. fundamental form (see equation (3.2.5), (3.2.7)).

$$K = \pm \frac{1}{R_1} \cdot \frac{1}{R_2} = \frac{LN - M^2}{EG - F^2}, \quad (3.2.14)$$

Definition 3.2.10: (mean curvature). The average of principal curvatures at a regular point **P** on the surface is called *mean curvature* H .

Note the mean curvature also can be expressed by the coefficients I. And II. fundamental form (see equation (3.2.5), (3.2.7)).

$$H = \frac{1}{2} \left(\frac{1}{\varepsilon R_1} + \frac{1}{\varepsilon R_2} \right) = \frac{EN - 2FM + GL}{2(EG - F^2)^2}, \quad \varepsilon = \pm 1. \quad (3.2.15)$$

In this moment, we can stop speaking about theory of differential geometry and we can look at their applications for the feature extraction. In several papers (eg. [6], [56], [40]) it is shown how to classify a shape of the surface in a given point \mathbf{P} according to the values of Gaussian and mean curvatures as is shown in Table 3.2.1:

	$K < 0$	$K = 0$	$K > 0$
$H < 0$	saddle ridge	ridge	peak
$H = 0$	minimal surface	flat	none
$H > 0$	saddle valley	valley	pit

Table 3.2.1: Interpretation of Gaussian and mean curvature on a surface (taken from [6]).

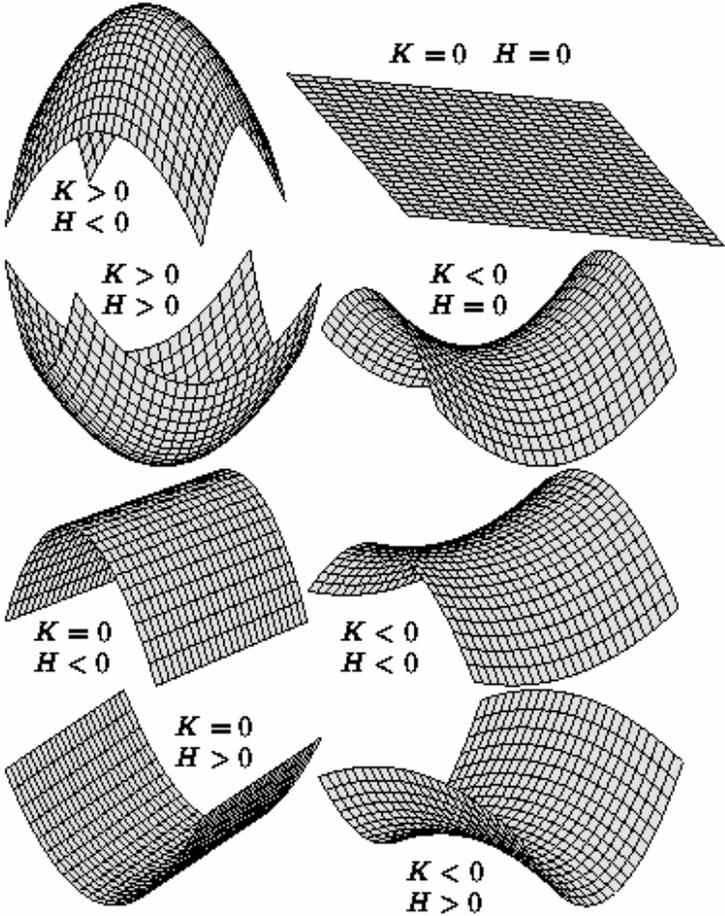


Figure 3.2.4: An outline of the basic shapes of the surface classified by Gaussian curvature K and mean curvature H (taken from [3]).

or according to principal curvatures k_1, k_2 (corresponding to values $1/R_1$ and $1/R_2$, respectively) as is shown in Table 3.2.2.

	$k_1 < 0$	$k_1 = 0$	$k_1 > 0$
$k_2 < 0$	peak	ridge	saddle
$k_2 = 0$	ridge	flat	valley
$k_2 > 0$	saddle	Valley	pit

Table 3.2.2: Interpretation of principal curvatures on a surface (taken from [6]).

The shape of mentioned types of surfaces is possible to see in Figure 3.2.4.

The curvatures (exactly it would be spoken about their estimations) can be computed many ways. Flynn and Jain [18] describe some existing methods of surface curvature estimation and they classify them into numeric and analytic categories.

The numerical methods use collection of curvature estimation in some directions. Many methods exist. Calladine [8] estimates the Gaussian curvature in a point \mathbf{r} by the formula:

$$\kappa(\mathbf{r}) = \frac{\beta(\mathbf{r})}{S(\mathbf{r})}, \quad (3.2.16)$$

where is $\beta(\mathbf{r})$ an angular defect in the vertex that is defined as 2π minus sum of interior angles of triangles meeting at the vertex \mathbf{r} (see Figure 3.2.5) and $S(\mathbf{r})$ is equal to 1/3 of the areas of the triangles meeting in the vertex.

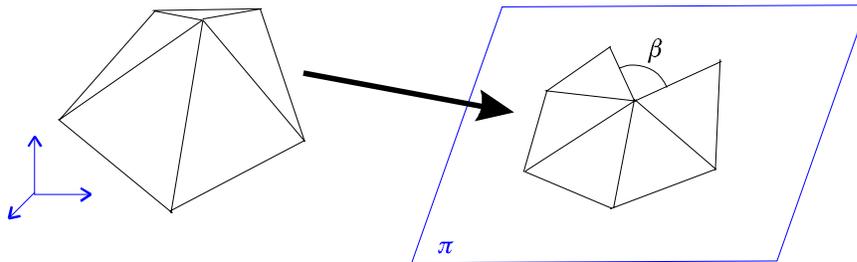


Figure 3.2.5: Illustration of spreading the triangles meeting in a vertex on a plane π due to computation of the angular defect β in the vertex.

Hoffman and Jain [26] estimate the curvature in moving from point \mathbf{p} to point \mathbf{q} on the surface as:

$$\kappa(\mathbf{p}, \mathbf{q}) = \frac{\|\mathbf{n}_p - \mathbf{n}_q\|}{\|\mathbf{p} - \mathbf{q}\|} s(\mathbf{p}, \mathbf{q}), \quad (3.2.17)$$

where $s(\mathbf{p}, \mathbf{q})$ indicates whether the curvature is positive (convex) or negative (concave). The parameters \mathbf{n}_p and \mathbf{n}_q denotes normals in the point \mathbf{p} and \mathbf{q} , respectively (a correlation of the direction of the normals classifies the curvature as positive or negative).

The mentioned numerical methods show that a curvature of a surface can be computed numerically and, of course, many other numerical methods exist (see e.g. [49], [30]).

Second approach of computation of curvatures is based on analytic methods. These methods usually estimate curvatures in a point on the surface based on so called *Monge path* that is defined as:

$$\mathbf{x}(u, v) = (u, v, f(u, v)). \quad (3.2.18)$$

The Gaussian curvature K and mean curvature H are expressed as:

$$K = \frac{f_{uu}f_{vv} - f_{uv}^2}{(1 + f_u^2 + f_v^2)^2}, \quad (3.2.19)$$

$$H = \frac{(1 + f_v^2)f_{uu} - 2f_u f_v f_{uv} + (1 + f_u^2)f_{vv}}{2(1 + f_u^2 + f_v^2)^{\frac{3}{2}}}, \quad (3.2.20)$$

Where the subscripts indicate partial differentiations:

$$\begin{aligned} f_u &= \frac{\delta}{\delta u} f, \quad f_v = \frac{\delta}{\delta v} f, \\ f_{uu} &= \frac{\delta^2}{\delta u \delta u} f, \quad f_{vv} = \frac{\delta^2}{\delta v \delta v} f, \\ f_{uv} &= f_{vu} = \frac{\delta^2}{\delta u \delta v} f = \frac{\delta^2}{\delta v \delta u} f. \end{aligned} \quad (3.2.21)$$

Own function $f(u, v)$ of Monge path can be various. Usually, it depends on the other properties that are required from the function. Besl and Jain [4] use a set of discrete orthogonal polynomials to provide a quadratic surface fit. Flynn and Jain [18] use a bicubic polynomial or Boyer and Srikantiah [6] use biquadratic polynomial. Naturally, many another functions can be chosen. However, the Monge path serves here as a tool for an estimation of curvatures and because the formulas of curvatures need partial derivatives of second order, maximally quadratic function would be sufficient. Note that a coordinate system and values of parameters in polynomial are needed to determine for a calculation of the curvatures. Boyer and Srikantiah [6] use PCA (see chapter 3.1) to determine a coordinate system (eigenvectors are orthogonal, one from them has the same direction as normal vector and odd eigenvectors determine tangential plane in a point). The coefficients of polynomial (Monge path) can be computed for example by least squares (see e.g. [5], [41]).

The main and Gaussian curvatures are used by many ways for the feature extraction of 3D models. First of all, many approaches are based on classification of the surface according of

values of curvatures (see Figure 3.2.6) and then are determined some key points representing given regions.

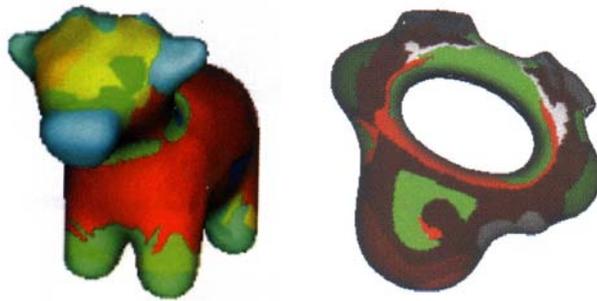


Figure 3.2.6: Examples of surface classification of 3D models based on values of mean and Gaussian curvatures (regions are highlighted by different colors, taken from [6]).

3.3. A reflective symmetry descriptor⁶

Kazhdan et al. [31], [32] presents a new method that can be used for matching or classification of objects. A reflective symmetry descriptor, which is introduced, is based on a measure of reflective symmetry for 3D voxel models. The method can also be used for a triangle mesh or the other 3D model representation. However, first step has to be transformed into the voxel representation. The advantages of this method are following:

- It is defined over a canonical 2D domain (the sphere) and thus it provides a common parameterization for arbitrary 3D models.
- It characterizes the global shape of the object and it is insensitive to noise and the other small perturbation in a 3D model.
- It provides distinguishing shape information for many objects

The complete process of the calculation of the reflective symmetry descriptor is shown in Figure 3.3.1 in a simplified way. It is separated into several individual tasks that have the following function:

- (0) It converts the model from the origin representation onto 3D voxel representation characterizing the model.

⁶ Majority information, images and formulas in this section are taken from [31], [32].

- (1) For every plane passing through the center of mass w compute the reflective symmetry distance of the model with respect to the plane.
- (2) The distances are combined to obtain the reflective symmetry descriptor – it measures how similar it is to its reflection (see later).
- (3) Finally, the similarity between two reflective symmetry descriptors are computed (it is taken their L^∞ distance)

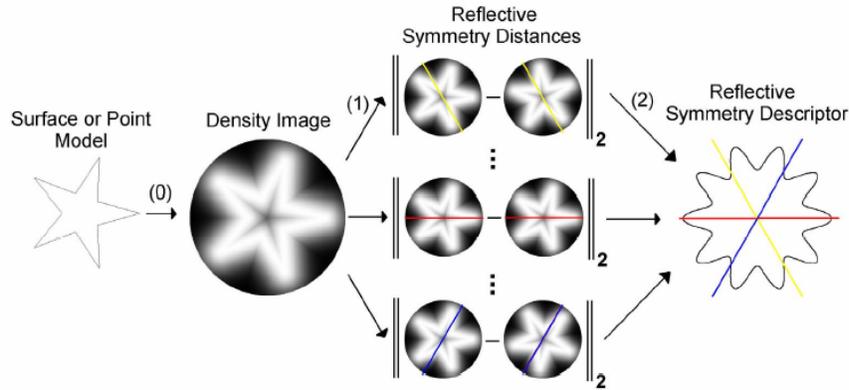


Figure 3.3.1: An outline of the computation reflective symmetry descriptor for a 3D model.

The author uses the following mathematical theory for the definition of the *reflective symmetry descriptor* that is taken from [31] , [32]:

Definition 3.3.1: The L^2 -distance of a function f to the nearest function g that is invariant with respect to the reflection γ is called the *symmetry distance* of the function. It can be formulated into the equation:

$$SD(f, \gamma) = \min_{g|\gamma(g)=g} \|f - g\|. \quad (3.3.1)$$

Theorem 3.3.1: The space of functions is a inner product space and the functions invariant to the reflection γ define a vector subspace. It follows that the nearest invariant function g is precisely the projection of f onto the subspace of invariant functions. If we define π_γ to be the projection onto the space of functions invariant under the action of γ and we define $\pi_{1/\gamma}$ to be the projection onto the orthogonal subspace then:

$$SD(f, \gamma) = \|f - \pi_\gamma(f)\| = \|\pi_{1/\gamma}(f)\|. \quad (3.3.2)$$

Shortly we can say that the symmetry distance of f with respect to reflection γ is the length of the projection of f onto a subspace of function indexed by γ . It was observed that

reflections are orthogonal transformations and therefore the theorem from representation theory [44] can be applied on the symmetry distance.

Theorem 3.3.2: The projection of a vector onto the subspace invariant under the action of an orthogonal group is the average of the vector over the different elements in the group.

Thus in the case of a function f and a reflection γ we get:

$$SD(f, \gamma) = \left\| f - \frac{1}{2}(f + \gamma(f)) \right\| = \left\| \frac{f - \gamma(f)}{2} \right\|. \quad (3.3.3)$$

We can simply say that the symmetry distance is the L^2 difference between the initial function and its reflection.

After definition of the symmetry distance and the citation of its properties, the reflective symmetry descriptor can be defined. Since we are interested in 3D space we only can formulate the definition for this case:

Definition 3.3.2: Given a 3D function, a 2D function on the space of planes through the origin (indexed by their unit normals), describing the proportion of f that is symmetric with respect to reflection about a given plane and the proportion of f that is anti-symmetric:

$$RSD(f, s) = \left(\frac{\|\pi_s(f)\|}{\|f\|}, \frac{\|\pi_{1/s}(f)\|}{\|f\|} \right) = \left(\frac{\sqrt{\|f\|^2 - SD^2(f, s)}}{\|f\|}, \frac{SD(f, s)}{\|f\|} \right). \quad (3.3.4)$$

Where π_s is the projection onto the space of functions invariant under reflection about the plane passing through the origin, perpendicular to s , and $\pi_{1/s}$ is the projection onto the orthogonal complement.

Now we know the needed theory background for the computation of the reflective symmetry descriptor and so we can look at its practical usage for 3D models. Firstly we describe symmetry distance function for a simple example, exactly for a circle.

Assume that function $f(\theta)$ on the circle and a reflection γ_α about the line through the origin with angle α is given. When the fact that this reflection maps a point with angle θ to the point with angle $2\alpha - \theta$ is used (see Figure 3.3.2) we can apply equation (3.3.3). Then the symmetry distance can be formulated as:

$$SD(f, \gamma_\alpha) = \sqrt{\int_0^{2\pi} \left(\frac{f(\theta) - f(2\alpha - \theta)}{2} \right)^2 d\theta}. \quad (3.3.5)$$

Note that the required time for the calculation symmetry distance can be accelerated if the equation (3.3.5) is adjusted in the following form:

$$SD(f, \gamma_\alpha) = \sqrt{\frac{\|f\|^2}{2} - \int_0^{2\pi} \frac{f(\theta)f(2\alpha - \theta)}{2} d\theta}. \quad (3.3.6)$$

The expression in square root consists of two terms. First term represents L^2 norm and the second one represents a convolution term. This convolution term just can be computed by Fast Fourier Transform for all angles α in $O(N \log(N))$ time complexity.

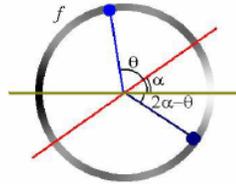


Figure 3.3.2: Reflection about α maps a point with angle θ to the points with angle $2\alpha - \theta$

That definition of the symmetry distance for the circle can be used for the calculation of the symmetry distance of an interior of a disk (by reason of simplicity the disk with unit radius will be thought in the following text). The interior of the disk can be represents as a collection of the circles with the radius from the range from θ to the maximal radius of the disk (see Figure 3.3.3). At first, the function $f(x,y)$ have to be transformed into polar coordinates to get the collection of the function $\{\hat{g}_r\}$:

$$\hat{g}_r(\theta) = f(r \cdot \cos\theta, r \cdot \sin\theta), \quad (3.3.7)$$

where $r \in (0,1]$ and $\theta \in [0,2\pi)$. Then the final equation for the symmetry distance of the interior of the disk can be formulated as:

$$SD(f, \gamma_\alpha) = \sqrt{\int_0^1 SD^2(\hat{g}_r, \gamma_\alpha) \cdot r dr}, \quad (3.3.8)$$

where γ_α is the reflection about a given line through the origin with angle α .

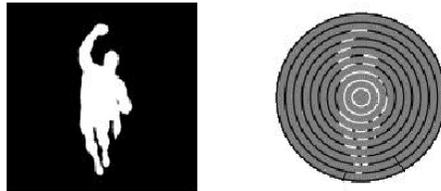


Figure 3.3.3: An illustration of the decomposition of an image into concentric circles.

We can use an analogical procedure on the surface of a unit sphere. For the computation planes passing through origin and a North pole on the sphere are only used. The symmetry distance is computed by braking up to upper and lower hemisphere and by projecting them into

a disk (see Figure 3.3.5 where the process for the upper hemisphere is illustrated). For a parametrization of the sphere in terms of spherical coordinates:

$$\Phi(\phi, \theta) = (\cos \phi, \sin \phi \cos \theta, \sin \phi \sin \theta), \quad (3.3.9)$$

where $\phi \in (0, \pi]$, $\theta \in (0, 2\pi]$, we can write a formula for the function \hat{g}_u of the upper hemisphere ($\phi \in (0, \pi/2]$) with the constant latitude:

$$\hat{g}_u(\phi \cos \theta, \phi \sin \theta) = f(\Phi(\phi, \theta)) \sqrt{\sin \phi / \phi}. \quad (3.3.10)$$

Note that the obtained disk has radius $\pi/2$, the factor of square root of $\sin \phi / \phi$ is necessary for integration below and analogically, the formula for the function \hat{g}_l of the lower hemisphere can be deduced. The final symmetry distance is calculated according to the formula:

$$SD(f, \gamma_\alpha) = \sqrt{SD^2(\hat{g}_u, \gamma_\alpha) + SD^2(\hat{g}_l, \gamma_\alpha)}. \quad (3.3.11)$$

Finally, the symmetry distance function on the surface of the sphere can be used to define the symmetry distance for functions on a voxel grid. The procedure of the deduction is very similar to previous. The function f is decomposed into a collection of function $\{\hat{g}_r\}$, where $\hat{g}_r(v) = f(rv)$. The measure of symmetry of f with respect to a reflection γ is equal:

$$SD(f, \gamma_\alpha) = \sqrt{\int_0^1 SD^2(\hat{g}_r, \gamma) \cdot r^2 dr}. \quad (3.3.12)$$

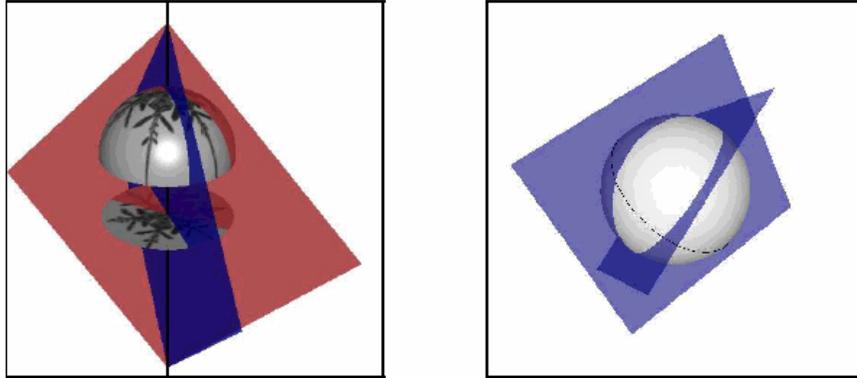


Figure 3.3.4: The left image illustrates a decomposition of the sphere surface (only upper hemisphere) into disk representation. The right image illustrates that all planes through the origin and North pole vary over a great circle.

This symmetry distance function can be used for a calculation of the reflective symmetry descriptor (see Figure 3.3.5) of 3D models described by voxel grid. It has nice properties, such as stability, scale invariance, etc. (see [32]) and therefore it is suitable for feature extraction. More detail information is possible to find in [31], [32].

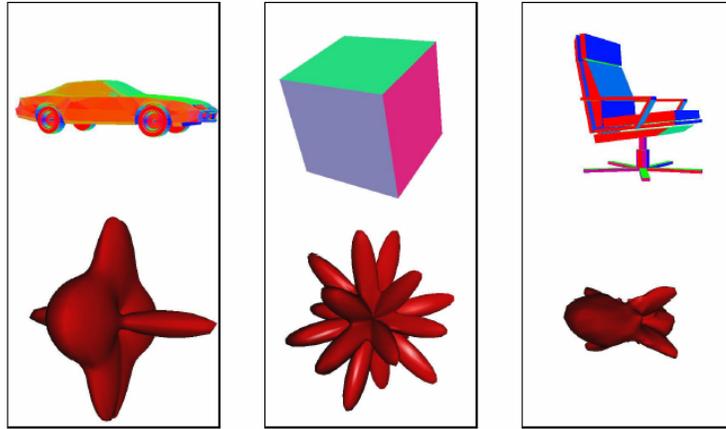


Figure 3.3.5: Examples of the visualization of the reflective symmetry descriptor for a car, a cube and a chair.

3.4. A spherical harmonic descriptor⁷

Another feature extraction similar to the reflection symmetry descriptor (RSD) is a so called *spherical harmonic descriptor*. Such as RSD it is defined on a voxel grid, however, another 3D model representation can also be used. They only have to be transformed into the voxel grid. This descriptor is a component of the search engine for 3D models that is described in [21]. Note the search engine in this report has the other components serving for retrieval of the models on 2D sketch or a text⁸. However, we are only interested in the feature extraction from 3D representation, so these components will be omitted.

Shortly, this method can be described as a decomposition of a 3D model into a collection of functions defined on concentric spheres. The spherical harmonics are used to discard orientation information. The process of the calculation of the spherical harmonic descriptor is outlined in Figure 3.4.1. The individual phase (which are indexed in the figure from value 1 to 5) has following function:

- (1) The polygonal surface is rasterized into a $2R \times 2R \times 2R$ voxel grid. If it is within one voxel width of a polygon surface the value of the voxel is assigned on the value 1, otherwise it is assigned on the value 0. The model is moved so that the center of

⁷ The images and equations in this section are taken from [21].

⁸ This retrieval system is executed on the web page: <http://shape.cs.princeton.edu/search.html>.

mass lies on the point (R, R, R) and one is scaled so that the average distance from the surface into center of mass is $R/2$.

- (2) The grid file is decomposed into spherical coordinates:

$$f(r, \theta, \phi) = \text{Voxel}(r \sin \theta \cos \phi + R, r \cos \theta + R, r \sin \theta \sin \phi + R), \quad (3.4.1)$$

where $r \in (0, R]$, $\theta \in (0, \pi]$, $\phi \in (0, 2\pi]$. It is obtained a collection of spherical function $\{f_0, f_1, \dots, f_R\}$ with:

$$f_r(\theta, \phi) = f(r, \theta, \phi). \quad (3.4.2)$$

- (3) Using spherical harmonics, each function f_r is expressed as a sum of its different frequencies:

$$f_r(\theta, \phi) = \sum_m f_r^m(\theta, \phi), \quad (3.4.3)$$

where the function f_r^m is the projection of the function f_r onto m -th irreducible representation of the rotation group acting on the space of spherical functions (it is described lower in more detail):

$$f_r^m(\theta, \phi) = \sum_{n=-m}^m a_{mn} \sqrt{\frac{2m+1}{4\pi} \cdot \frac{(m-|n|)!}{(m+|n|)!}} \cdot P_{mn}(\cos \theta) e^{in\phi}. \quad (3.4.4)$$

- (4) A rotation invariant signature for f_r is defined as the collection of scalars $\{\|f_r^0\|, \|f_r^1\|, \dots\}$. Note it was observed that the value $\|f_r^m\|$ does not change if the function f_r is rotated.
- (5) The signatures over different radii are combined and by combining them the two-dimensional rotation invariant spherical harmonics descriptor for a 3D model is obtained. The indexes correspond to the length of the m -th frequency of the rasterisation of f to the sphere with radius r .

The key role of the algorithm plays a third point in which spherical harmonic functions are computed. The spherical harmonics occur in a large variety of physical problems, e.g., in the quantic physics. It is a collection of functions of two coordinates ϕ, θ on the surface of the sphere. Mathematically, they are defined as (taken from [15]):

$$Y_{lm}(\theta, \phi) = \sqrt{\frac{2l+1}{4\pi} \cdot \frac{(l-m)!}{(l+m)!}} \cdot P_l^m(\cos \theta) e^{im\phi}. \quad (3.4.5)$$

and:

$$Y_{l,-m}(\theta, \phi) = (-1)^m \cdot Y_{lm}^*(\theta, \phi). \quad (3.4.6)$$

where $-l \leq m \leq l$. The function P_m^l denotes a so called *Legendre polynomials* and asterisk in the equation denotes complex conjugation.

The mathematic interpretation of these equations will not be explained here, but it is possible to find, e.g., in [15], [41], etc. Remark that the authors the voxel grid (that consists of 64x64x64 voxels) decompose into 32 spherical functions by restricting the voxel grid to spheres with radii 1 through 32 (the point (2) in the procedure above). Then each of these functions is decomposed as a sum of its first 16 harmonics components in the point (3) of the procedure (it is analogous to a Fourier decomposition into different frequencies).

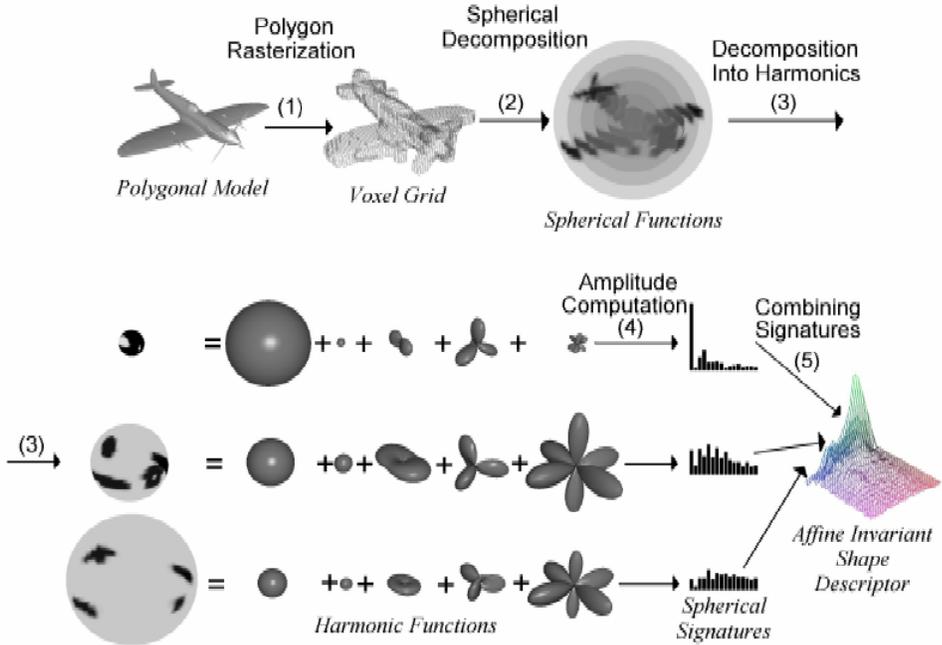


Figure 3.4.1: Outlines of the computation of the spherical harmonic descriptor

Finally, note that that described 3D model can be indexed without registration of 3D models in a canonical coordinate systems and so an influence of the orientation of the model in the space is minimized.

3.5. Topology matching

The topological and skeletal structures are features that could be suitable for description of the form of 3D models. Several studies have been proposed in which different ways of skeleton or topology description are used. However, no each data structure is suitable as a search key in the retrieval system. For example, the medial axis model is well-known skeletal structure (see, e.g., [29], [12]). Unfortunately, it is inapplicable for feature extraction because it

is sensitive to noise and small undulations on the surface of the 3D model. On the other hand, Hilaga etc. [24] proposed a method based on a so called *Reeb graph* that has nice properties from our point of view. It represents topological based features extraction very well. An appropriately definition of the Reeb graph guarantees the data structure that is invariant to translation and rotation, resistant against noise and certain deformations. The Reeb graph is shortly described⁹ in the following text. More detail information is possible to find in [24], [28].

Definition 3.5.1: Let $\mu : C \rightarrow \mathbb{R}$ be a continuous function defined on a object C . The Reeb graph is the quotient space of the graph of μ in $C \times \mathbb{R}$ by the equivalent relation $(X_1, \mu(X_1)) \sim (X_2, \mu(X_2))$, which holds if and only if $\mu(X_1) = \mu(X_2)$ and X_1, X_2 are in the same connected component of $\mu^{-1}(\mu(X_1))$.

Naturally, amount of the μ function can be used. Perhaps the most simplest and very often used function is a height function on a 2D manifold that is defined as:

$$\mu : \mathbf{v}(x, y, z) \rightarrow z, \tag{3.5.1}$$

where \mathbf{v} is a point on the manifold and x, y, z are its axis coordinates. An example of the Reeb graph with the height function is shown in Figure 3.5.1 (it is taken from [24]).

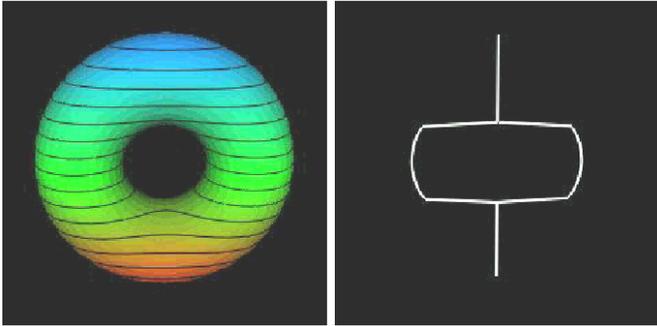


Figure 3.5.1: Torus and its Reeb graph using a height function μ

On the left figure the range color represents change of the values of the height function from minimum (red color) to maximum (blue color) value and the black lines represent the contours which was used for generating Reeb graph. The resultant Reeb graph is on the right figure.

It is evident that the height function is not suitable for feature extraction because it is not invariant to transformations. To obtain translate and rotate invariant function Hilaga defined the function μ at a point \mathbf{v} on a surface S as:

⁹ The retrieval system based on the Reeb graph is possible to find on the web page <http://3dsite.dhs.org/~dynamic>. The description of the implementation of the system is presented in [28].

$$\mu(\mathbf{v}) = \int_{\mathbf{p} \in S} g(\mathbf{v}, \mathbf{p}) dS, \quad (3.5.2)$$

where the function $g(\mathbf{v}, \mathbf{p})$ is geodetic distance between two points \mathbf{v} and \mathbf{p} on the surface S .

Such defined function is not invariant for scaling yet. Therefore it is normalized according to the following equation:

$$\mu_n(\mathbf{v}) = \frac{\mu(\mathbf{v}) - \min_{\mathbf{p} \in S} \mu(\mathbf{p})}{\max_{\mathbf{p} \in S} \mu(\mathbf{p})}. \quad (3.5.3)$$

It is normalization from the range $\langle \min_{\mathbf{p} \in S} \mu(\mathbf{p}), \max_{\mathbf{p} \in S} \mu(\mathbf{p}) \rangle$ into the range $\langle 0, 1 \rangle$. Let us note that the denominator of the equation (3.5.3) is not typical for normalization. Correctly it should equal to the value $range(S) = \max_{\mathbf{p} \in S} \mu(\mathbf{p}) - \min_{\mathbf{p} \in S} \mu(\mathbf{p})$. However, in the cases where the value $range(S)$ is small (see the sphere and the cube in Figure 3.5.2) it would amplify errors. Therefore it is only employed maximal value of the function μ in the denominator. For all that the character of that function is efficient and invariant to the rotation, translation, scaling and also a small deformation of the object. Several examples of the distribution values of the function is shown in Figure 3.5.2.

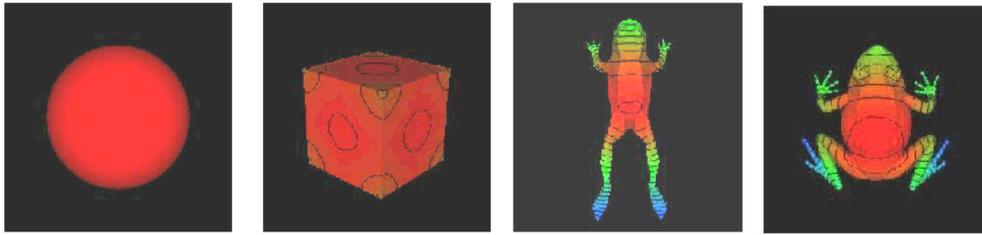


Figure 3.5.2: Examples of the distribution of the function μ that is defined as an integral of the geodetic distances (red and blue color represents min. and max. value respectively).

The figure illustrates the quality of the selected function, however, the question similarity comparing models is still open. The methods of the similarity measuring described in the 2.3 section mostly rely on the distance measuring points in n -dimensional space but they do not allow to calculate with the topology or the skeleton of the objects. Therefore, Hilaga uses special matching algorithms. The basic point of the algorithm is the construction of the multiresolutional Reeb graph. It is a set of the Reeb graphs for the same 3D model that are generated with smaller and smaller splitting of the space, such as shown in Figure 3.5.3 for height function.

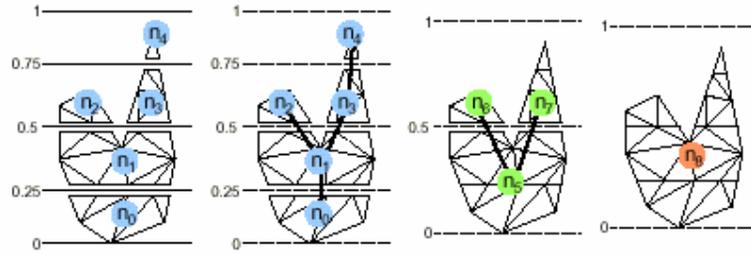


Figure 3.5.3: Construction of a Multiresolutional Reeb graph.

The own matching algorithm is based on coarse-to-fine strategy. The nodes of the Reeb graph with the lowest resolution are taken as a start point and by comparing individual nodes in the graphs are created pairs of the nodes which are compared in more detail (the Reeb graph with bigger resolution is used). Detailed description of the procedure is described in the mentioned paper.

4. Multidimensional indexing

In practice, databases are very large. They can contain thousands and more objects. Naturally, growing number of the objects leads to longer time needed for searching in the database. Unfortunately, some standard indexing methods that accelerate searching in the database cannot be used because they are only proposed for one-dimensional case and the feature vector (in our case) generally represents an object in n -dimensional space. A solution is used a so called *multidimensional access methods*. At present, many methods exist. Detail description of methods would be very extensive¹⁰ therefore an overview and primary classification of the method are only mentioned here.

4.1. Spatial and point access methods

Geade [22], Seeger and Kriegel [34], [43] present a classification of multidimensional access methods which are divided into two fundamental groups: *point access methods* (PAMs) and *spatial access methods* (SAMs). PAM stores only multidimensional points and SAM can store objects that represent a shape, such as lines, polygons, etc. A difference between points of view of on the methods is presented in the next subsection.

4.1.1. Point access methods

Geade [22] looks at this kind of methods according to their implementation. He reports the following classification of them:

¹⁰ Detail description of individual methods that are mentioned in this section is presented in many publications. A survey of methods can be found e.g. in [22], [10], [25].

- **Hashing access methods:** they use one-dimensional hashing technique to index multidimensional points. The main problem of the methods is to ensure proximity of close objects after a projection from multidimensional space. They use heuristic techniques and for example, *grid file*, *EXCELL* belong to this group.
- **Hierarchical access methods:** they use hierarchical data structures to manage point data. Many methods belong to this group, such as *quadtree*, *k-d-B Tree*, etc. Note hybrid methods that are combination of both hierarchical and hashing techniques also exist (e.g. *Buddy tree*, *BANG file*) and they are classified as hierarchical.

Seeger and Kriegel [43], [25] classify PAMs according to properties of regions of the multidimensional space in which points are managed. These properties are:

- **Intervals:** the regions have shape of intervals (it means n-dimensional box) or some arbitrary polyhedron.
- **Complete:** the regions cover complete space or only some part including of points.
- **Disjoint:** the regions are pairwise disjoint or they may have mutual overlaps.

Property			PAM
intervals	complete	disjoint	
X	X	X	quadtree (<i>Finkel and Bentley 1974</i> , <i>Samet 1984</i>), k-d-B tree (<i>Robinson 1981</i>), EXCELL (<i>Tamminen 1982</i>), interpolation hashing (<i>Burkhard 1983</i>), multidimensional extendible hashing (<i>Otoo 1984</i>), grid file (<i>Nievergelt, Hinterberger and Sevcik 1984</i>), balanced multidimensional two-level grid file (<i>Hinrichs 1985</i>), interpolation-based grid file (<i>Ouksel 1985</i>), extendible hash tree (<i>Otoo 1986</i>), MOLHPE (<i>Kriegel and Seeger 1986</i>), PLOP-hashing (<i>Kriegel and Seeger 1988</i>), quantile hashing (<i>Kriegel and Seeger 1989</i>), LSD-tree (<i>Henrich, Six and Widmayer 1989</i>)
X	X		twin grid file (<i>Hutflesz, Six and Widmayer 1988</i>)
X		X	Multilevel grid file (<i>Whang and Krishnamurthy 1985</i>), buddy tree (<i>Seeger and Kriegel 1990</i>)
	X	X	BSP-tree (<i>Fuchs, Kedem and Naylor 1980</i>), BD-tree (<i>Ohsawa and Sakauchi 1983</i>), BANG file (<i>Freeston 1987</i>), hB-tree (<i>Lomet and Salzberg 1989</i>)

Table 4.1.1: A classification of PAMs according Seeger and Kriegel (1990).

The table that authors published shows overview of methods that are classified according these properties.

4.1.2. Spatial access methods

As it was mentioned spatial access methods serve to manage objects with a spatial extension (e.g. line, polygons, polyhedrons, etc.) and they are often an extension of PAMs. Seeger and Kriegel [43], [25] classify these methods according to a technique (see below) for that is used for modification of PAMs (in 1988). Later on, Kriegel et al. (in 1991) added another factor for classification – base type (i.e., spatial data types that a method supports primarily). An overview of method that are classified according to the mentioned factors is shown in Table 4.1.2 (taken from [22]).

technique	base type			
	grid cell	interval (box)	sphere	polyhedron
transformation	ZkDB+ tree (Orenstein 1986), BANG file (Freeston 1987), hB tree (Lomet and Salzberg 1989)	all PAMs except of the BANG file and the hB tree		P tree (Jagadish 1990)
overlapping regions		R tree (Guttman 1984), R* tree (Backmann et al. 1990), skd tree (Ooi et al. 1987), GBD tree (Ohsawa and Sakauchi 1990), Hilbert R-tree (Kamel and Faloutsos 1994), buddy tree with overlapping (Seeger 1991)	sphere tree (Oosterom 1990)	P tree (Schiewietz 1993), KD2B tree (Oosterom 1990)
clipping		EXCELL (Tamminen 1982), extended k-d tree (Matsuyana et al. 1984), R+ tree (Sellis et al. 1987), buddy tree with clipping (Seeger 1991)		cell tree (Günther 1988)
multiple layers		multi-layer grid file (Six and Widmayer 1988), R-file (Hutflesz et al. 1990)		

Table 4.1.2: A classification of SAMs according Kriegel et al. (1991).

In the table, four techniques that can be used to modify PAMs are mentioned. They have following meaning [22], [25]:

- **Transformation:** geometric objects are mapped into points in a higher-dimensional space (e.g. a rectangle in E^2 can be mapped as a point in E^4) then the classical PAMs can be used.
- **Overlapping regions:** it is based on decomposition of the space into a hierarchical structure. Usually, objects are stored in the leaves of the hierarchical structure and intermediate nodes serve to facilitate searching.
- **Clipping:** it is use also hierarchical data structure as overlapping regions technique. However, it ensures by clipping that intermediate nodes are non-overlapping, i.e., if an object is clipped then its parts are stored in several nodes.
- **Multiple layers:** it partitions the space into more independent parts that are referred as layers. The layers are managed in a hierarchical structure and each one may use a different algorithm to partition of the space.

Detail description of the individual techniques is presented in Geade [22]. The basic principles of the methods (PAMs and SAMs) are also presented there and it can serve as a primary report for introduction into multidimensional indexing issue.

4.2. Vector and metric space methods

Castelli [10] presents another point of view on multidimensional access methods. He divides the methods on *vector space methods* and *metric space methods* where the criterion is approach of indexing. The following subsections describe both categories in more detail.

4.2.1. Vector space methods

The methods that index individual items of database are called vector space methods. They are further subdivided into four following categories (taken from [10]):

- **Non-hierarchical methods:** two classes of the methods belong to this subcategory. The first class is based on mapping multidimensional space onto the real line be

means of a space-filling curve (e.g., *Peano curve*, *z-order*, *Hilbert curve*) and on indexing the mapped records by a one-dimensional indexing structure. The second class is based on partitioning the space into a predefined number of non-overlapping fixed-size regions (e.g., *grid file*).

- **Recursive partitioning methods:** they recursively divide the space into smaller regions. The resultant hierarchical structure is usually represented as a tree (e.g., *R tree*, *SS tree*).
- **Projection-based methods:** they are indexing structures that support approximate nearest-neighbor queries. They can be subdivided into two further classes where the first class supports *fixed-radius queries*, i.e., the queries that returns only results with distance smaller than the fixed radius from a query point (e.g., see [20]), and the second one supports $(1+\epsilon)$ queries, i.e., the queries that return results whose distance is guaranteed to be less than $1+\epsilon$ times the distance of the exact result (e.g., see [33]).
- **Miscellaneous partitioning methods:** These methods belong to vector space methods but they cannot be categorized into previous categories, such as *CSVD method*, *Onion method*, or *Pyramid method* (see [10]).

Each subcategory of the vector space methods has some advantage and some disadvantage. Castelli [10] presents interesting table in which he compares suitability of the methods for retrieval systems according to two factors: type of the query and dimensionality of the feature vector (see Figure 4.2.1).

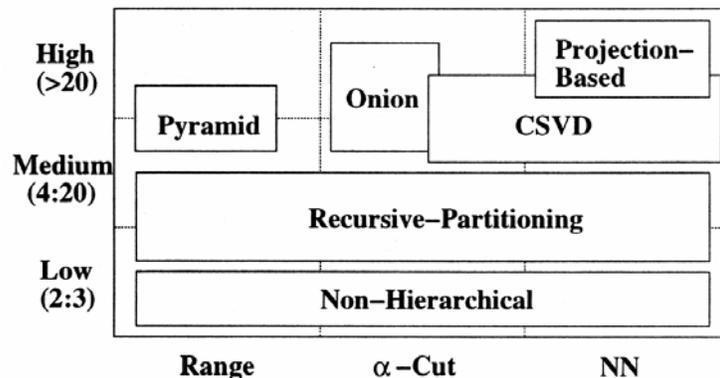


Figure 4.2.1: An illustration of a usage of vector space methods in relation to dimensionality of the feature vector and the type of query (taken from [10]).

4.2.2. Metric space methods

Matrix space methods are based on indexing the distances between database items. They can be used e.g. where the selected metric is very computationally expensive or where the distances are provided with data set. Such as vector space method, they are subdivided into two basic categories (taken from [10]):

- **Methods indexing metric structure:** these methods are based on indexing metric structure of the space. Two main approaches exist. First approach uses so called *Voronoi regions* (each point of feature space can be associated with the closest database item, this collection of associations is called Voronoi region) and for example, *cell* methods, *M tree*, or *X tree* belong to this group. The second one uses an *ordering list*. It is the list of all the pairwise distances between database items that is sorted in ascending order of distance.
- **Vantage-point methods:** they use a tree structure for searching in the space where the tree is generated with the help a *vantage points*. Typical example of this group is e.g., *vp tree*.

These types of methods also can be compared according to their suitability for retrieval systems, such as the vector space method. Figure 4.2.2 shows dependence of the methods on the type of queries and dimensionality of the feature vector.

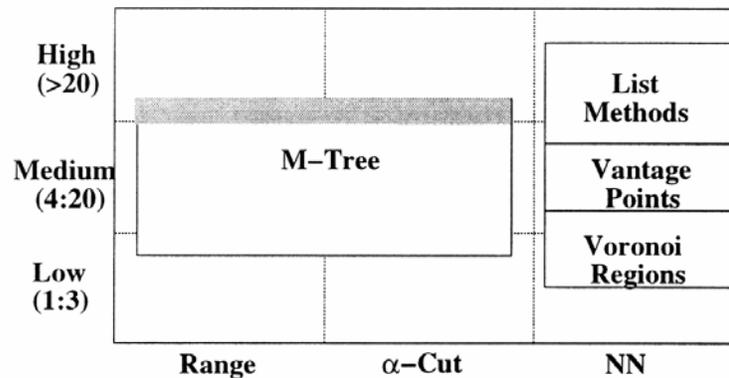


Figure 4.2.2: An illustration of a usage of metric space methods in relation to dimensionality of the feature vector and the type of query (taken from [10]).

5. Future work and our goals

3D data feature extraction is introduced in third chapter. It was shown that it is not simple issue and it can be solved by different approaches. In the next period I will go more deeply in the field 3D data feature extraction method and I would like to develop a novel technique in which in which would be used my knowledge from computer graphics, mathematics and another fields with which I have met during my studies.

My existing work up to the present has been pointed on generating triangulations by brutal force. Perhaps, it could seem a little bit different from this topic, however, my last report describes a method in which a technique comparing a similarity of subtrees in a generated tree are used (see [ii], [iii]). The given knowledge and experience with brutal force will be used in my future work as well.

My main goal is to propose a method that would combine properties of methods based on a histogram and a topology features. More exactly, two methods based on curvatures and Reeb graphs (see chapters 3.2 and 3.5) that just represent both kinds of features are introduced in this report. I plan to use them as fundamental approaches for proposing a new method. In the first period, it will be supposed that 3D models are represented by triangular meshes without singularities. Of course, this condition restricts the use of the method only for a small typical collection of data. However, if the properties of our method are good then we will try to expand the methods for more general 3D data sets. It is expected that the final methods will be convenient for free-form data.

6. References

- [1] Barros J., French J., Martin W., Kelly P., Cannon M.: Using the triangle inequality to reduce the number of comparisons required for similarity-based retrieval. Proc. of SPIE, vol. 2670, 392-403, 1996.
- [2] Basri R., Costa L., Geiger D., Jacobs D.: Determining the similarity of deformable shapes. Vision Research, 1998.
- [3] Belyaev A.: Surfaces. Surface Curvatures. Curves on Surface. Geometric Modeling course, <http://www.mpi-sb.mpg.de/~belyaev/Math4CG/Math4CG.html>.
- [4] Besl P. J., Jain R. C.: Invariant surface characteristics for 3D object recognition in range images. Computer Vision Graphics Image Process., 1986.
- [5] Boehm W., Prautzsch H.: Geometric Concepts for Geometric Design. A K Peters, Welleslay, Massachusetts, 1994.
- [6] Boyer K. L., Srikantiah R., Flynn P. J.: Saliency Sequential Surface Organization for Free-Form Object Recognition, Computer Vision and Image Understanding, Volume 88, Number 3, December 2002.
- [7] Budinský B.: Analytická a diferenciální geometrie. Praha, SNTL 1983.
- [8] Calladine C. R.: Gaussian Curvatures and Shell Structures, In The Mathematics of Surfaces. Clarendon Press, Oxford, 1985.
- [9] Campbell R. J., Flynn P. J.: A Survey of Free-Form Object Representation and Recognition Techniques. Computer Vision and Image Understanding 81, 2001.
- [10] Castelli V.: Multidimensional Indexing Structures for Content-based retrieval. IBM research Report, RC 22208(98723), 2001.
- [11] Cicirello V., Regli W. C.: Machining Feature-based Comparisons of Mechanical Parts. International Conference on Shape Modeling and Applications (SMI 2001), 176-185, 2001.
- [12] Cohen I., Ayache N., Sulger P.: Tracking Points on Deformable Objects Using Curvature Estimation. ECCV '92, Lecture notes in Computer Science 588, 1992.
- [13] Culver T., Keyser J., Manocha D.: Accurate Computation of the Medial Axis of a Polyhedron. Proc. Symp. Solid Modeling. pp. 179-190, 1999.

- [14] Cyr Ch. M., Kimia B. B.: 3D Object Recognition Using Shape Similarity-Based Aspect Graph. International Conference on Computer Vision, 2001.
- [15] Eberly D.: Spherical Harmonics. In Numerical Recipes in C: The Art of Scientific Computing, ISBN 0-521-43108-5, 1992.
- [16] Efrat A., Itai A., Katz M. J.: Improvements on Bottleneck Matching and Related Problem Using Geometry, Proceedings 12 Annual Symposium on Computational Geometry, 1996.
- [17] Elad M., Tal A., Ar S.: Content based Retrieval of VRML Objects - An Iterative and Interactive Approach. Eurographics Multimedia Workshop, 97-108, 2001.
- [18] Flynn P. J., Jain A. K.: Surface classification: Hypothesis testing and parameter estimation. In Proc. Conf. Computer Vision nad Pattern Recog., June 1988.
- [19] Flynn P. J., Jain A. K.: On reliable curvature estimation. In Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 110-116, June 1989.
- [20] Friedman J. H., Baskett F., Shustek L. J.: An algorithm for finding nearest neighbors IEEE Trans. Computer, October 1975.
- [21] Funkhouser T., Min P., Kazhdan M., Chen J., Halderman A., Dobkin D.: A Search Engine for 3D Models. ACM Transactions on Graphics, Volume 22, Issue 1, pp 83 – 105, 2003.
- [22] Gaede V., Gunther O.: Multidimensional Access Methods. ACM Computing Surveys, Volume 30 , Issue 2, pp. 170 – 231, 1998.
- [23] Giannopoulos P., Veltkamp R. C.: A pseudo-metric for weighted point sets. To appear in European Conference on Computer Vision, 2002.
- [24] Hilaga M., Shinagawa Y., Kohmura T., Kunii T. L.: Topology Matching for Fully Automatic Similarity Estimation of 3D shapes. SIGGRAPH 2001, 203-212, 2001.
- [25] Hee-Kap A., Mamoulis N., Wong H. M.: A Survey on Multidimensional Access Methods, Technical Report, UU-CS-2001-14, Institute of Information and Computing Sciences, Utrecht University, May 2001.
- [26] Hoffman R. L. and Jain A. K.: Segmentation and classification of range images, IEEE Trans. Pattern Anal. Mach. Intell, 1987.
- [27] Chang S., Hsu A.: Image information systems: Where do we go from here. IEEE Trans. on Knowledge and Data Eng., 4(5):431-442, 1992.
- [28] Chen D.-Y., Ouhyoung M.: A 3D Object Retrieval System Based on Multi-Resolution Reeb Graph, Proc. of Computer Graphics Workshop, pp.16, Tainan, Taiwan, June 2002.
- [29] Chuang J.-H., Tsai C.-H., Ko, M.-C.: Skeletonization of Three-dimensional Object Using Generalized Potencial Field. IEEE Trans. PAMI, Vol. 22, pp.1241-1251, 2000.
- [30] Ittner D. J., Jain A. K.: 3-d surface discrimination from local curvature measures. Conf. Computer Vision and Pattern Recognition, May 1985.
- [31] Kazhdan M., Chazelle B., Dobkin D., Finkelstein A., Funkhouser T.: A Reflective Symmetry Description. To appear in European Conf. on Computer Vision, 2002.

- [32] Kazhdan M., Chazelle B., Dobkin D., Finkelstein A., Funkhouser T.: A Reflective Symmetry Description for 3D Models. To appear in *Algorithmica*, 2003.
- [33] Kleinberg J. M.: Two algorithms for nearest-neighbor search in high dimensions. In Proc. 29th ACM Symp. Theory of Computing, Texas, USA, 1997.
- [34] Kriegel H. P., Schiwietz M., Schneider R., Seeger B.: Performance comparison of point and spatial access methods. In Proc. 1st Symp. Design Large Spatial Databases, 1989.
- [35] Löffler J.: Content-based Retrieval of 3D models in Distributed Web Databases by Visual Shape Information. Int. Conf. on Information Visualisation, 2000.
- [36] Naramsihalu A. D.: Special section on content-based retrieval. *ACM Multimedia Sys.*, 3(1):1-41, 1995.
- [37] Niblack W., Barber R. and et al.: The QBIC project: Querying images by content using color, texture and shape. In Proc. SPIE Storage and Retrieval for image and Video databases, 1994.
- [38] Novotni M., Klein R.: A Geometric Approach to 3D Object Comparison. Int. Conf. on Shape Modeling and Applications (SMI 2001), 154-166, 2001.
- [39] Pentland A., Picard R. W., Sclaroff S.: Photobook: Content-based manipulation of image databases. *Int. J. Comput. Vis.*, 18(3):233-254, 1996.
- [40] Quek K. H., Yarger R. W. I., Kirbas C.: Surface Parameterization in Volumetric Images for Curvature-based Feature Classification. *IEEE Transactions on Systems, Man and Cybernetics*. 2001.
- [41] Rektorys K. and at al.: *Přehled užití matematiky*. Nakladatelství Prometheus, Praha, ISBN 80-85849-72-0, 1995
- [42] Rui Y., Huang S. T., Chang S.: Image Retrieval: Current Technologies, Promising Directions and Open Issues. *Journal of Visual Communication and Image Representation*, 10(1):39-32, 1999.
- [43] Seeger B., Kriegel H. P.: Techniques for design and implementation of efficient spatial access methods. In Proc. 14th Int. Conf. On Very Large Data Bases VLDB'97, Los Angeles, CA, USA, 1997
- [44] Serre J.: *Linear representations of Finite Groups*. Springer-Verlag. New York. 1977.
- [45] Shikhare D.: Feature-based Techniques for Handling Geometric Models. <http://angelfire.lycos.com/space2/dineshshikhare/writings/features.pdf>.
- [46] Shum H.-Y., Hebert M., Ikeuchi K.: On 3D Shape Similarity. *Proc IEEE Computer Vision and Pattern Recognition*, 526-531, 1996.
- [47] Smeulders A.W.M., Worring M., Santini S., Gupta A., Jain R.: Content based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12): 1349-1380, 2000.

- [48] Smith A. D.: The folding of the human brain: From shape to function. PhD Thesis, kings Collag, London, 1999.
- [49] Smith J. R., Chang S.-F.: Visually searching the web for content. IEEE Multimedia Mag., 4(3):12-20,1997.
- [50] Tamara H., Yokoya N.: Image database systems: A survey. Pattern Recognition, 17(1):29-43, 1984.
- [51] Tangelder J. W. H., Veltkamp R. C.: Polyhedral Model Retrieval Using Weighted Point Sets. International Journal of Image and Graphics, Vol. 3, No. 1: 209-229, 2003.
- [52] Veltkamp R. C.: Shape Matching: Similarity Measures and Algorithms. Technical Report UU-CS-2001-03, Utrecht University, 2001.
- [53] Veltkamp R. C., Hagedoorn M.: State-of-the-Art in Shape Matching. Technical Report UU-CS-1999-27, Utrecht University, 1999.
- [54] Veltkamp R. C., Tanase M.: Content-based image retrieval systems: A survey. Technical Report UU-CS-2000-34, Utrecht University, 2001.
- [55] Vranic D. J., Saupe D.: 3D Model Retrieval. Proceedings of the Spring Conference on Computer Graphics and its Applications (SCCG2000), Budmerice, Slovakia, pp. 89-93, May 2000.
- [56] Yarger R. W. I., Quek K. H.: Surface Parameterization in Volumetric Images for Feature Classification. IEEE Inter. Symposium of Bio-Eng., BIBE 2000, pp. 297-303, 2000.

Appendix A – Publications

- [i] Hlavaty T.: Generator of triangular meshes with required properties, MSc Thesis (in Czech), supervisor: Skala V., University of West Bohemia, Plzen, 2001.
- [ii] Hlavaty T.: Brute-force generator of triangular meshes with required properties, Technical report of Department of Computer Science and Engineering, (in Czech), University of West Bohemia, Czech Republic, DCSE/TR-2002-09, 2002.
- [iii] Hlavaty T., Skala V.: The Brute-force Generator of Triangulations with Required Properties, ICCVG 2002, Zakopane, Poland, ISBN 839176830-9, September 25-29, 2002.
- [iv] Hlavaty T., Skala V.: A Survey of Methods for 3D Model Feature Extraction. Appeal to submission in Szczyrk 2003.

Appendix B – Stays and Lectures Abroad

Stays:

February– May 1999	University of Ioannina, Greece.
12.6.2002 – 23.6.2002	University of Limoges, France.

Lectures:

19.6.2002	Triangular Mesh Generation with Required Properties by The Brute-force Approach, University of Limoges, France.
-----------	--

Conferences:

25.9.2002 – 29.9.2002	ICCVG 2002, Zakopane, Poland.
-----------------------	-------------------------------

Appendix C – MPEG-7 Standard¹

The Moving Picture Coding Group (MPEG) is a working group of the Geneva-based ISO/IEC standard organization (International Standards Organization/International Electro-technical Committee). Its main goal is to develop international standards for compression, decompression, processing, and coded representation of moving pictures and audio.

Nowadays five standards exist (and/or develop) and MPEG-7 standard just belongs to them. Formally, it is named *Multimedia Content Description Iner-face*. Note that it does not standardize the extraction of audiovisual descriptions and it does not specify any program that should be used. It only describes multimedia content so users can search, browse, and retrieve that content efficiently (see Figure C.1).

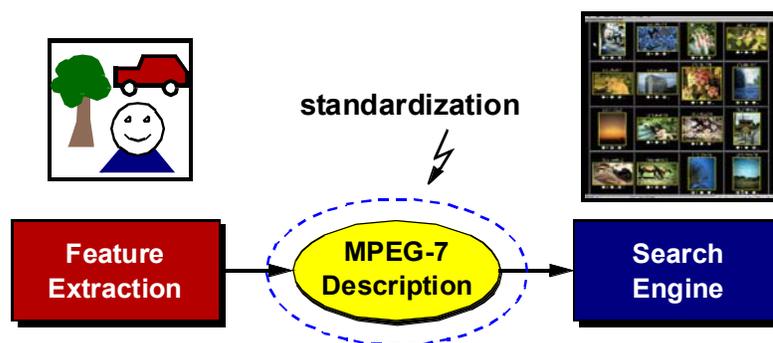


Figure C.1: The Scope of MPEG-7 (taken from [60]).

Main goal is to standardize [60], [61]:

- **A set of descriptors:** A descriptor (D) is a representation of a feature that defines the syntax and semantics of the feature representation.
- **A set of description schemes:** A description scheme (DS) specifies the structure and semantics of the relationships between its components, which may be both descriptors and description schemes.
- **The Description Definition Language (DDL):** It is a language that specifies description schemes. MPEG-7 uses XML Schema Language for content description

¹ Detail description of the MPEG-7 standard is possible to find on homepages: <http://www.mpeg-industry.com>, or <http://mpeg.telecomitalialab.com/standards.htm>.

(MPEG-7 DDL). Note the DDL requires some specific extensions to XML Schema Language to satisfy all the requirements of MPEG-7.

- **One or more way to encode descriptions:** A coded description is a description that's been encoded to fulfill relevant requirements such as compression efficiency, error resilience, and random access.

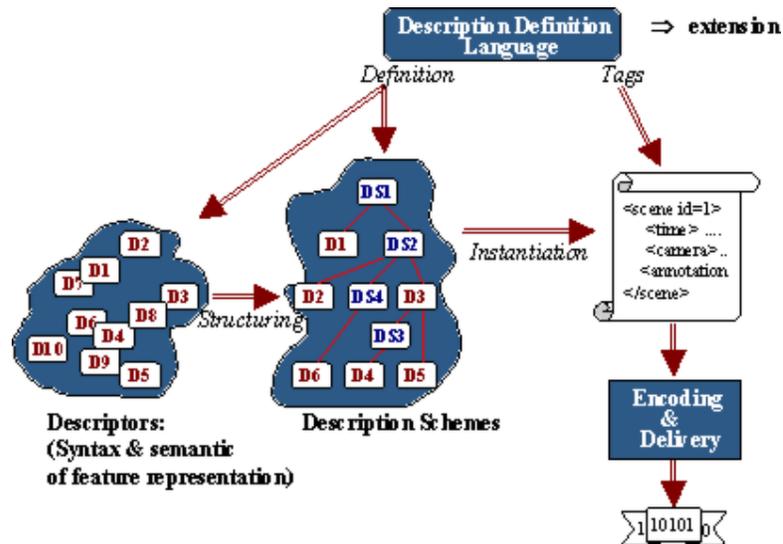


Figure C.2: The relationship among MPEG-7 main elements (taken from [60]).

Currently, MPEG-7 concentrates on the specification of description tools (i.e., Descriptors and Description Schemes) together with the development of the MPEG-7 reference software that is known as XM (eXperimentation Model). Several working groups exist that are interested in different parts of the MPEG-7 standard. The MPEG-7 Visual group belongs to them. It develops description tools as Color, Texture, Shape, Motion, Localization, and Face recognition, where each category consists of next elementary descriptors. From this set of tools the Shape description tool is can be used for describing of 3D models, therefore, the next text is only dedicated to it.

As it was mentioned before, the Shape description tool is proposed to the description of the shape of objects. It further consists of three shape descriptors: Contour Shape, Region Shape, and Shape 3D.

The Contour Shape descriptor captures characteristic shape features of an object or region based on its contour. It uses so-called Curvature Scale-Space representation, which captures perceptually meaningful features of the shape.

The Region Shape descriptor can describe diverse shapes efficiently. It may consist of either a single region or a set of regions that also can describe an object with some holes. Some examples of those objects are illustrated in the Figure C.3.



Figure C.3: Some examples of shapes (taken from [60]).

The 3D Shape Descriptor considers that 3D objects are represented as polygonal meshes. Therefore, it provides an intrinsic shape description of 3D mesh models. It exploits some local attributes of the 3D surface.

Appendix D – Reeb graph

Topological description of 3D models belongs to interesting problems of computer graphics. One of many solutions is to use so called Reeb graph that can represent the skeleton of 3D models. However, some terms have to be introduced before describing the Reeb graph.

In the 1930's M. Morse proved so called Morse theory. Originally, Morse theory was applied on a class of mathematical objects called smooth manifolds (such as a plane, a circle, the surface of a sphere, etc.) and provides a tool for understanding the topology of objects with limited information. Briefly, this theory is defined in the following definition (taken from [57], [58], [64]).

Definition D.1: Let $f(\mathbf{x})$ be a real function defined on a manifold M and \mathbf{x} a point of M . For any point \mathbf{x} near a given point \mathbf{x}_0 the value of function $f(\mathbf{x})$ with infinitely many derivatives at \mathbf{x}_0 can be expressed by the Taylor series:

$$f(\mathbf{x}) = c_0(\mathbf{x}_0) + c_1(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0) + c_2(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0)^2 + \dots, \quad (D.1)$$

where

$$c_k(\mathbf{x}_0) = \frac{f^{(k)}(\mathbf{x}_0)}{k!}. \quad (D.2)$$

Suppose $\mathbf{x}_0 = \mathbf{0}$ (i.e., it represents the origin), then the equation can be simplified:

$$f(\mathbf{x}) = c_0(\mathbf{x}_0) + c_1(\mathbf{x}_0) \cdot \mathbf{x} + c_2(\mathbf{x}_0) \cdot \mathbf{x}^2 + \dots. \quad (D.3)$$

If $c_0(\mathbf{0}) = 0$ and $c_1(\mathbf{0})$ is different from zero, then for \mathbf{x} close to origin the equation (D.3) can be approximated as:

$$f(\mathbf{x}) \approx c_1(\mathbf{x}_0) \cdot \mathbf{x}. \quad (D.4)$$

That point is called *regular point* of $f(x)$.

If $c_0(\mathbf{0}) = 0$, $c_1(\mathbf{0}) = 0$, and $c_2(\mathbf{0})$ is different from zero, then for \mathbf{x} close to origin the equation (D.3) can be approximated as:

$$f(\mathbf{x}) \approx c_2(\mathbf{x}_0) \cdot \mathbf{x}^2. \quad (D.5)$$

That point is called *non-degenerate critical point* of $f(x)$.

If $c_0(\mathbf{0}) = 0$, $c_1(\mathbf{0}) = 0$, $c_2(\mathbf{0}) = 0$, and $c_3(\mathbf{0})$ is different from zero, then for \mathbf{x} close to origin the equation (D.3) can be approximated as:

$$f(\mathbf{x}) \approx c_3(\mathbf{x}_0) \cdot \mathbf{x}^3. \quad (D.6)$$

That point is called *degenerate critical point* of $f(\mathbf{x})$. Note that degenerate or non-degenerate critical point is also called a *singularity* of the function $f(\mathbf{x})$.

Definition D.2: (Morse function) Let $f(\mathbf{x})$ be a real smooth function defined on a smooth manifold M . The function is called the Morse function iff all of its critical points are non-degenerate. A critical point is non-degenerate if the Hessian matrix² \mathbf{H} of $f(\mathbf{x})$ is non-singular (i.e., $\det(\mathbf{H}) \neq 0$) at that point.

Theorem D.1: (Morse's lemma) If \mathbf{x}_0 is a non-degenerate critical point of a function f on a manifold M , there is some open neighborhood of \mathbf{x}_0 in M and a set of local coordinates $\mathbf{x}_0 = [x_1, \dots, x_n]^T$ such that, in these coordinates, the function has the form $f(\mathbf{x}) = f(\mathbf{x}_0) - (x_1)^2 - \dots - (x_k)^2 + (x_{k+1})^2 + \dots + (x_n)^2$ where k is called the index of \mathbf{x}_0 (the prove of this theorem can be found in [59]).

Briefly, if f is a function of more than one variable its local geometry near a non-degenerate critical point looks like a saddle, since the graph of the function is a surface that may bend in different directions at a given point. Since the critical point is non-degenerate, this saddle must curve downward in k coordinate directions and upward in the remaining directions.

Morse theory guarantees that a space topologically equivalent to the manifold can be constructed by attaching a finite number of primitive cells where each cell is associated to a type of critical point. However, these cells do not describe surface of 3D object completely.

Reeb [62], [63] tries to expand the Morse theory and he proposed so called Reeb graph that codes information about the evaluation and the structure. This graph can be defined as follows (taken from [24], [57], [58]).

Definition D.3: Let $\mu : C \rightarrow \mathbb{R}$ be a continuous function defined on an object C . The Reeb graph is the quotient space of the graph of μ in $C \times \mathbb{R}$ by the equivalent relation $(X_1, \mu(X_1)) \sim (X_2, \mu(X_2))$, which holds if and only if $\mu(X_1) = \mu(X_2)$ and X_1, X_2 are in the same connected component of $\mu^{-1}(\mu(X_1))$.

Practical application of the Reeb graph for description of the topology of a 3D object is shown in the Figure D.1. The function μ represents the height h of the object, where the function has a minimal value at the bottom of the figure and a maximal value in the top of the figure (see the arrow in the figure). The critical points represent nodes in which the graph is bifurcated or connected. The structure of the branches can be obtained by calculation of the components (contours) between critical points such as shown in the figure.

² The elements of Hessian matrix are second order partial derivatives $f(\mathbf{x})$, i.e. $h_{ij} = \partial^2 f(\mathbf{x}) / \partial x_i \partial x_j$.

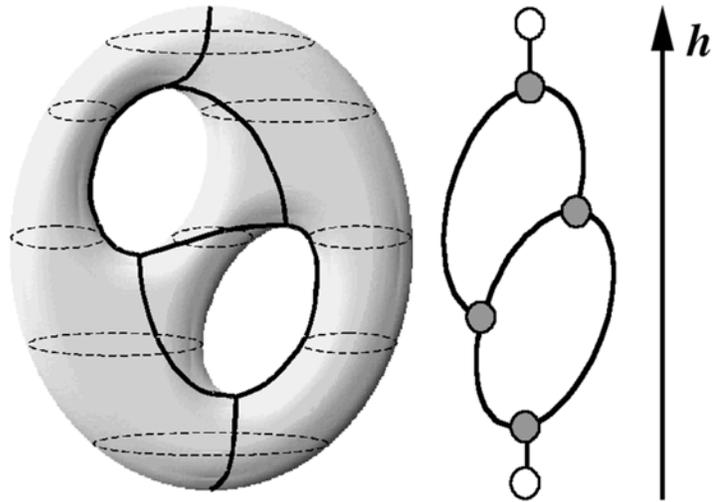


Figure D.1: An example of the Reeb graph (taken from [57]).

Of course, the height function is not ideal for feature extraction because it has not any desired properties (e.g., invariance to transformation, etc.). It has been only taken for illustration of behavior of the Reeb graph. Generally, more complex functions are needed that satisfy all desired properties. A good example is the function based on calculation of geodesic distance that Hilaga [24] used in his work (see 3.5 section for more detail).

Appendix E – My future work

The fifth section of the work is dedicated to short description of my future work. As it was mentioned there, our goal is to contribute to the development of a new method for feature extraction of 3D objects.

Nowadays, many methods exist. However, an ideal method has not been developed yet. The existing methods try to describe 3D objects by a feature vector that would have ideal properties as invariance to transformations, accuracy of description, quick to compute, etc. (see the introduction of the third section of my work). The professor Thomas A. Funkhouser from the Princeton University belongs to the most respected expert working in this field. His group has developed several methods for feature extraction of 3D objects that are based on different mathematical background (the methods that are described in the 3.4 and 3.5 section belong to them). All methods represent the feature vector by a histogram of some measured values. Therefore, according to classification of the methods (see the introduction of the third section) they belong to the group of histogram based methods.

An advantage of histogram based methods is that feature vectors can be quickly compared mutually. However, on the other hand, these methods cannot estimate local features of the surface of the 3D object. A solution of this drawback can be in use of mean and Gaussian curvatures for describing 3D objects. The values of curvatures can classify the local features of the surface and so they can describe the object in more detail (see the 3.2 section). Perhaps it can seem as an ideal method, however, it has also any disadvantages. One of them is higher sensitivity to noise. During scanning of a 3D object some error can occur and this inaccuracy exhibits in calculation of the curvatures. In this moment, a topological based methods offer as a solution. These methods describe the topological structure of 3D objects and selection of a suitable method from this group can lead to smaller sensitivity to noise. A good example of this method is the Reeb graph (or his modification) that can describe the skeleton of 3D objects (see 3.3 section and appendix D). However, this approach has also some drawbacks such as time complexity of the methods and of the comparing similarity of the graphs, etc.

When we sum previous paragraph each method has some advantages and disadvantages. Our goal is to propose a new method that would combine advantages of the object extraction computed by the curvatures and by a topological based method (as well Reeb graph, or possibly another representation). This method should be fast, invariant to transformations (such as rotation, translation, scale) and less sensitive to a noise. In the first step, it is supposed that objects are represented as triangular meshes without singularities (i.e., manifold objects). This condition is very restrictive to collections of 3D data that are accessible at present. However, if our method has good properties for this restricted group of objects we

could continue to expand the method for more general 3D data sets. It is expected that the final method will be convenient for free-form data, too.

It can be seen that the design of the completed retrieval system is not easy and the problem of feature extraction is not the only problem that has to be solved. Practically, it means that a team of people that are interested in individual issues would be needed. Fortunately, another alternative exists. It is based on use of MPEG-7 standard (see appendix D). In the word, many research centers deal with individual problems (such as query processing, multidimensional indexing, etc.) and the MPEG-7 can serve as a bridge that standardizes interface among them. In this moment, the design of the retrieval system can be seen as a connection of those parts where one of them just can present our method. The MPEG-7 is the powerful tool and its employment offers the chance to propose a retrieval system in which we only could be interested in the issue of 3D object feature extraction. It is just our goal that we would like to reach.

Appendix F – Publications

- [57] Biasotti S.: Topological techniques for shape understanding. CESC2001, Budmerice, Slovakia, May 2001.
- [58] Biasotti S., Mortara M., Spagnuolo M.: Surface compression and reconstruction using Reeb graphs and shape analysis. In Proc. of Spring Conference on Computer Graphics, pp. 175-184, Bratislava, May 2000.
- [59] Milnor J.: Morse Theory. Princeton university press, New Jersey, 1963.
- [60] MPEG Requirements Group: MPEG-7 Overview. Doc. ISO/IEC JTC1/SC29/WG11 N4980, Klagenfurt, July 2002.
- [61] MPEG Requirements Group: MPEG-7 Requirements Document V.10. Doc. ISO/MPEG N2996, Melbourne, October 1999.
- [62] Reeb, G.: Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. in Comptes Rendus Acad. Sciences, Vol. 222, pp. 847-849, Paris, 1946.
- [63] Shinagawa Y., Kunii T. L.: Construction a Reeb graph automatically from cross section. IEEE Computer Graphics and Applications, Vol. 11, No. 6, Nov./Dec. 1991.
- [64] Shinagawa Y., Kunii T. L., Kergosien Y. L.: Surface Coding Based on Morse Theory. IEEE Computer Graphics and Applications, Vol. 11, No. 5, Sept./Oct.1991.