INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

ShaDe Workbench: 3D Shape Descriptor Workbench

Tiago André Cardoso Lourenço

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática e de Computadores

Júri

Presidente:	Professor Doutor João António Madeiras Pereira
Orientador:	Professor Doutor Alfredo Manuel dos Santos Ferreira Júnior
Co-Orientador:	Professor Doutor Manuel João Caneira Monteiro da Fonseca
Vogais:	Professor Doutor Pável Pereira Calado

Novembro 2010

ShaDe Workbench: 3D Shape Descriptor Workbench

Tiago André Cardoso Lourenço

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática e de Computadores

Acknowledgements

First, I would like to express my gratitude to my adviser Professor Alfredo Ferreira and co-adviser Professor Manuel J. Fonseca for their support during the course of this thesis. I would like to thank in particular to the first one, for the constant and proper guidance given. Thank you for always believing in me and for all the confidence demonstrated.

I also want express my appreciation to the members of the Shape Modelling group of the CNR IMATI-Ge, who welcomed me so well in Genoa. Bianca Falcidieno, Michela Spagnuolo, Simone Marini, Michela Mortara, Francesco Robbiano, Marco Attene, Daniela Giorgi, Chiara Catalano, Riccardo Albertoni and Silvia Biasotti were phenomenal during my short stay with them, both in a personal and professional level.

Next, I want to thank to my colleagues and friends, Nelson Marques and Paulo Gomes, whom helped me throughout the degree. Also, I like to acknowledge Raquel Costa for the work we have done together.

Finally, would I like to thank to my family for the love shown over the years. In particular, to my parents, for all the unconditional support and the sacrifices they had to do during these last years.

The presented work was supported in part by the Portuguese Foundation for Science and Technology (FCT) through project 3DORuS, with reference PTDC/EIA-EIA/102930/2008.

Lisboa, November 22, 2010
Tiago Lourenço

All you need in this life is ignorance and
confidence, and then success is sure.
-Mark Twain

Resumo

O crescente número de objectos tridimensionais disponíveis em formato digital desencadeou um notável interesse de investigação nesta área. Actualmente é cada vez mais importante ter algoritmos de recuperação eficazes, uma vez que a complexidade associada à indexação e pesquisa de modelos tridimensionais numa base de dados é elevada. Encontrar métodos de análise, comparação e recuperação de modelos 3D eficientes tornou-se uma tarefa importante.

No entanto, apesar da existência de alguns *benchmarks* na área de recuperação de objectos, actualmente não existe uma ferramenta globalmente aceite para a análise e estudo das técnicas algorítmicas existentes. Normalmente, cada grupo de investigação usa o seu próprio sistema, com as suas próprias métricas e não há possibilidade de comparar os seus resultados com outros grupos. Somando a isso, quase não existem soluções que operem online. Quase todas as soluções de *benchmarking* têm de ser obtidas e executadas localmente. Existe, no entanto, uma ferramenta online, mas é parte de uma plataforma multimédia maior, não apenas para objectos 3D mas para vários tipos de informação, sendo complexa e não explorando plenamente o campo da análise de formas 3D.

Nesta dissertação, apresentamos uma solução simples e dedicada a objectos 3D. Propomos um sistema modular e escalável baseado na *web* que permita a adição de novos componentes, como descritores de formas ou algoritmos de segmentação, com esforço reduzido por parte dos investigadores que os desenvolveram. Acreditamos que virá a ser relevante para os investigadores que trabalham na análise, comparação e recuperação de objectos tridimensionais.

Abstract

The increasing number of three-dimensional objects available on digital format triggered a great interest in research in this domain. Today, it is increasingly important to have effective retrieval algorithms, since the complexity of indexing and searching of three-dimensional models in a database is high. Finding efficient methods of analysis, comparison and retrieval of 3D models has become an important task.

However, despite the existence of some benchmarks in the object retrieval area, currently there is no globally accepted tool for the analysis and study of existing algorithms techniques. Typically, each research group uses its own system, with their own metrics and there is no possibility to compare their results with other groups. Adding to that, there are hardly any solutions which operate online. Almost every benchmarking solution must be obtained and run locally. There is an online tool, but is part of a larger multimedia platform, not only for three-dimensional objects, but for various types of information, being complex and not fully exploiting the 3D shape analysis field.

In this thesis, we present a simple solution and dedicated to 3D objects. We propose a modular and scalable web-based system that allows the addition of new components, like shape descriptors or segmentation algorithms, with minor effort by researchers who developed them. We believe that will be relevant to researchers working on the analysis, comparison and retrieval of three-dimensional objects.

Palavras Chave
Keywords

Palavras Chave

Benchmarking
Framework
Análise de Formas
Descritores de Formas
Objectos Tridimensionais

Keywords

Benchmarking
Framework
Shape Analysis
Shape Descriptors
Three-dimensional Objects

Contents

List of Figures

List of Tables

List of Abbreviations

AP	Average Precision
API	Application Programming Interface
CaS	Collection-aware Segmentation
CBIR	Content-based Image Retrieval
CNR	Consiglio Nazionale delle Ricerche
DCG	Discounted Cumulative Gain
DSW	Digital Shape Workbench
ESB	Engineering Shape Benchmark
F-M	F-Measure
HFP	Hierarchical Fitting Primitives
IMATI-Ge	Istituto di Matematica Applicata e Tecnologie Informatiche, from Genoa
INEX	INitiative for the Evaluation of XML Retrieval
IR	Information Retrieval
MAP	Mean Average Precision
MIREX	Music Information Retrieval Evaluation eXchange
NN	Nearest Neighbor
PSB	Princeton Shape Benchmark
SHA	Spherical Harmonics
ShaDe WB	3D Shape Descriptor Workbench
SHREC	3D Shape Retrieval Contest
TREC	Text REtrieval Conference
XML	eXtensible Markup Language

Chapter 1

Introduction

Over the last decades, the amount of digital multimedia information (*e.g.* audio, images, video) has been growing. More recently, three-dimensional models gained a bigger importance in multimedia data. 3D models constitute a new emerging multimedia content due to recent advances on modeling, digitizing and visualizing techniques, like 3D scanner acquisition or 3D graphics rendering technologies. As a result of these advances, three-dimensional models are now widely used in several different domains, such as computer vision, virtual reality, medicine, molecular biology, mechanical computer-aided design, archaeology or video games.

However, 3D objects are more complex than other media, since there are several representations for these objects, the most common being the polygonal mesh. The object is represented by its boundary surface which is composed of a set of planar faces.

The increasing number of three-dimensional objects stored in collections triggered the research on three-dimensional model analysis, comparison and retrieval methods, because it was necessary to have algorithms that perform effective 3D object retrieval, since looking for three-dimensional objects in a database is not a trivial task. During the last few years, researchers have been developing several algorithms related to shape description or segmentation of 3D objects, for example.

The following section describes the motivation behind our work. Next, we enumerate the key objectives of our work. Following we set forth our approach. Then, we summarize our main contributions of this research. Finally, we give an overview of this document.

1.1 Motivation

Despite the augmented interest in shape-based retrieval methods and matching algorithms there is no simple and dedicated system which makes possible to researchers working in this area a central study and comparison of algorithms and techniques developed within this context.

The researchers might use some existing 3D benchmarking solutions, like the Princeton Shape Benchmark [47], the AIM@SHAPE Shape Repository[1], the Engineering Shape Benchmark [26] or the SHREC[2], but currently there is no generally accepted tool for the analysis and study of the 3D shape descriptors. Normally, each research group uses their own system, with their own metrics and there is no possibility for comparing their results with others. Adding to this, there are hardly any solutions, which operate online. Almost every benchmark solution must be obtained and run locally. This might lead to different results depending of the local machines where the 3D shape descriptors are tested. However, there is a benchmark solution which works online, the MyMultimediaWorld.com [12, 32], but is part of a bigger

[1] AIM@SHAPE Shape Repository at http://shapes.aim-at-shape.net
[2] http://www.aimatshape.net/event/SHREC

multimedia platform, not just for 3D objects, being more complex and does not fully explore the 3D shape analysis field.

1.2 Objectives

Indeed, despite the large size of the research community working on the analysis, indexing and retrieval of three-dimensional objects, several issues remain unsolved on this topic. No existing approach offers a global satisfactory solution. The reduced number and availability of tools to support researchers' work on this area is a major problem that we aim to solve.

The work developed under this dissertation focuses on the development of a tool that will provide a simple and centralized workbench for 3D shape analysis, classification and retrieval. Despite its simplicity, this workbench covers a wide range of topics in the field, such as object segmentation, best view selection, shape description or mesh reconstruction. This innovative system should allow researchers, who work in the area of analysis, classification and retrieval of 3D objects, to study and compare available techniques with minor effort. It should also allow researchers to provide their new algorithms to be added to the system. The solution will be based on a client-server architecture working over the internet, thus easily available.

1.3 Our Approach

To meet the objectives described earlier in this work, we developed the 3D Shape Descriptor Workbench (ShaDe WB). However, during the initial study, we found it important to embrace a wider range of topics that were not initially seen as targets, such as the segmentation of three-dimensional objects, the selection of their best views and the repair of their meshes. This web-based system, *i.e.*, an application that is accessed through a Web browser over the internet, makes the solution available almost everywhere and all researchers are subject to the same conditions, thus allowing an easier comparison of results.

The system it is composed by two main components: a front-end web page and a back-end site. The front-end web page allows users to test and compare the various descriptors of 3D objects or the 3D object itself. In turn, the back-end site provides the necessary services for the front-end to work. The back-end also has an API based on XML, which allows the easy addition of new techniques for shape description, similarity computation, object segmentation, retrieval evaluation, mesh repairing and best view computation. Thus, this system will provide to researchers working in 3D object domain an easy way to perform their studies.

Therefore, to achieve the full success, we must involve researchers working in the field into our project. To that end we worked with the Shape Modelling group of CNR IMATI-Ge. These researchers provided some of their algorithms to be tested and evaluated the system. These evaluations were conducted through a questionnaire about the system and informal suggestions for improvement by the users themselves.

1.4 Contribution

The major contribution of our work to the scientific community was the development of a framework for shape analysis. This framework is flexible and modular, supporting various 3D shape descriptors and providing a simple way of analysing three-dimensional models and algorithms such as, segmentation, selection of best view of an object and reconstruction of meshes.

This work also served as a framework for Raquel Costa, in her thesis work "CaS: Collection-aware Segmentation". CaS is an algorithm to segment sets of objects based on their individuality. The purpose of her work consisted in the separation phase of the segmentation algorithm that uses the CaS which is integrated into a solution of indexing and retrieval of three-dimensional shapes of objects and validate the applicability of this approach for the segmentation of objects in a collection. Using ShaDe WB has been possible to perform tests with users in order to validate the applicability of their approach.

1.5 Dissertation Overview

The remaining of this dissertation is organized in five chapters. The next chapter presents the research background related with our work. We give an overview of some basic concepts for 3D object retrieval, key to understanding the purpose of our work, namely an overhaul on the state-of-art of 3D shape description, similarity measurement between three-dimensional models, shape segmentation, selection of the best view of a model, reparation of 3D shapes of objects and evaluation measurement of shape descriptors in content-based retrieval systems.

In Chapter 3 we describe the state-of-art regarding multimedia information retrieval benchmarking. We introduce briefly some benchmarks of several multimedia retrieval domains and then we depict in more detail the 3D shape benchmarking solutions. Chapter 4 describes in detail the framework for shape analysis that we purpose in this dissertation. The ShaDe WB is an innovative web-based system, composed by a front-end web and a back-end site that allows researchers working in 3D object domain an easy way to perform their studies.

Chapter 5 presents the methodologies adopted for the evaluation of our work, particularly with regard to the API designed and the user interface through user testing and heuristic evaluation. Finally, in Chapter 6 we present an overall discussion of our work, delineating conclusions and introducing perspectives for future work.

Chapter 2

Background

In this chapter we present some fundamental concepts necessary for an easier understanding of our work. Due to the importance of shape descriptors in our work, we will give a brief explanation of what they are and introduce some examples of their different types. Indeed, shape descriptor analysis and benchmarking was the primary focus of this thesis. However, during the initial study we understood that a wider range of topics must be addressed to fulfil the users needs. Therefore, we broadcast the focus of our work to cover other topics like similarity measures to check whether three-dimensional objects are similar, segmentation, best view selection and repair of 3D shapes of objects and, finally, evaluation measures commonly used to measure the performance of shape descriptors in content-based retrieval systems.

2.1 Shape Signatures

Three-dimensional models are used in different fields, which leads to a large number of domain-specific collections with several forms of model representation [14]. Despite these different model representations, they can be converted or approximated to a more generic one, such as a polygonal mesh, which can be interpreted by classification and retrieval algorithms.

Usually, these algorithms rely on a numerical representation to describe both models and queries. Three-dimensional shapes of objects are normally numerical represented by feature vectors, which are the standard approach for multimedia retrieval [20]. A feature vector, also referred as descriptor or object signature, is a set of values extracted from a multimedia object that describe it in a multi-dimensional space, making easier to look for similar objects.

To preserve the maximum shape information on a feature vector with the lower dimensionality possible is an important goal of any 3D shape description approach and is considered as the primary challenge in building a shape based retrieval system [23].

Due to the variety of distinct approaches, there is no universally accepted taxonomy of 3D shape descriptors. However, we will follow a classification scheme, illustrated in Figure 2.1, which was suggested by Akgül [3] and is similar to one proposed by Bustos *et al.* [13].

The taxonomy that we adopted in this report is divided into five categories, resulting of the technique used to construct the numerical representation of the shape. These five categories are: histogram-based, transform-based, graph-based, image-based and other shape descriptors. In the following sections we will present one or two descriptors we consider relevant of each of the first four categories. The current version of our framework supports these descriptors and any other that follows the same approach.

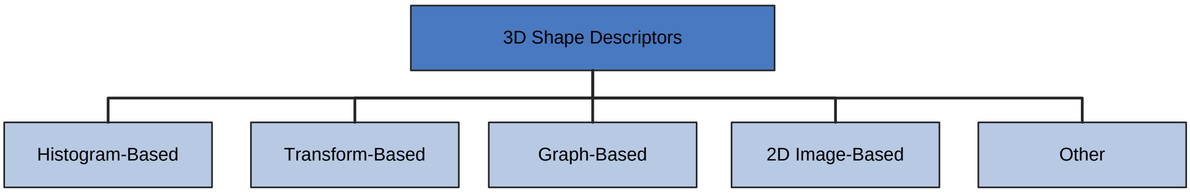

Figure 2.1: Taxonomy of 3D shape descriptors.

2.1.1 Histogram-based Descriptors

An histogram is a standard statistical description of a distribution in terms of occurrence frequencies of different event classes. Widely used in computer graphics, a color histogram is a representation of the distribution of colors in an image, derived by counting the number of pixels of each of given set of color ranges in a typically two-dimensional or three-dimensional color space. In 3D shape description domain, an histogram is normally seen as an accumulator of numerical values of certain features calculated from the shape to represent. Based in this free definition, many 3D shape descriptors can be considered as histogram-based methods, although they are not based on histograms in strict statistical sense.

Cord and Angle Histograms

Paquet *et al.* in [43, 42] presented the use of cord and angle histograms for 3D shape descriptors. They defined a cord as a vector that goes from the center of mass of the object to the center of mass of a bounded region on the surface of the object. Then, the descriptor is computed based on a set of three histograms. The first two represent the distribution of the angles between the cords and the first and second reference axes, while the third histogram shows the distribution of the radius.

This approach simplifies the triangles to their centers and does not consider the size and shape of the mesh of triangles. This allows that its implementation is simple and direct, because this method is not very discriminating regarding the details of objects, once global features are used to characterize the shape of objects.

Cord and angle histograms are commonly used in the retrieval of objects as an active filter, through which can be made more detailed comparisons or can be used in conjunction with other methods for improving shape descriptors.

Density-based Shape Descriptors

Another example of histogram-based descriptors are the density-based shape descriptors. These descriptors are defined as the sampled probability function (PDF) of some surface feature. The feature is local to the surface patch and treated as a random variable. Akgül and Sankur [16, 17] proposed this approach with the aim of extracting three-dimensional shape descriptors from local surface features characterizing the geometry of an object. In particular, the authors suggested [4] the use of the radial distance, the radial direction, the normal direction, the radial-normal alignment and the tangent-plane distance as features.

The features used in this 3D shape descriptor are not scalar or rotation-invariant and, during the pre-processing, pose normalization must be achieved. Nevertheless, this descriptor has performed well compared with several other histogram-based methods [17].

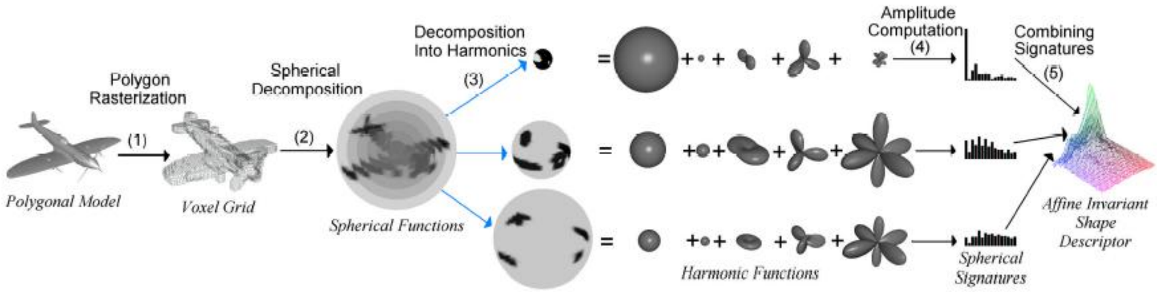

Figure 2.2: Princeton methodology for computing spherical harmonics shape descriptors (Figure taken from [24] ©2003 ACM).

2.1.2 Transform-based Descriptors

Transform-based descriptors rely on mathematical transformations to change from the spatial domain to a more suitable one and compute from there a shape descriptor. Although some recent approaches use different spaces, usually these new spaces are the frequency domain.

Rotation Invariant Spherical Harmonics

Kazhdan *et al.* proposed [30] a new method based on spherical harmonics to obtain the representation of the rotation invariant of three-dimensional objects. This method aims to overcome some limitations presented in other approaches and that were related with the canonical alignment used.

This new approach involves several steps. First, the polygon mesh is voxelized into a grid. Next, the object is intersected with a set of concentric spheres and a spherical function is constructed from the voxel values for each sphere. Then, the decomposition of the frequency of each of these functions is calculated, and the norms of each frequency of each component at each ray. The result is a 2D grid indexed by radius and frequency.

Following this idea, the Princeton group derived a practical approach [24], illustrated in Figure 2.2, to calculate the rotation invariant descriptor by using spherical harmonics and, more recently, Papadakis *et al.* also presented a methodology for 3D shape retrieval based on spherical harmonics [41].

2.1.3 Graph-based Descriptors

Graph-based approaches differ from previous approaches, because they extract not only the geometry but also the topology of three-dimensional objects. Indeed, these approaches focus on the topological relations between components of the object, using graphs to represent these relationships.

However, the graph-based approaches are not effective for most retrieval applications, since they are more complex than the previous approaches and can not be generalized for any 3D shape, forcing a restriction of each approach scope to a particular object type. Moreover, the complexity of the graph matching involves the use of alternatives to graph isomorphism, such as the application of techniques from spectral graph theory for the conversion of graphs into numerical descriptors [22, 48].

However, due to the potential of descriptors based on graphs, several researchers have focused attention on this approach in recent years.

Size Graphs

Biasotti *et al.* followed what had been done in other approaches based on graphs. They used the Reeb graph [44] to build the centerline skeleton of a 3D model and applied a size function to create a graph size [10]. Their idea is to associate with a three-dimensional object a graph (G^f, ϕ), being G^f the centerline skeleton calculated using the quotient function f and ϕ a measuring function labelling each node of the graph with local geometric properties of the model.

More specifically, Biasotti *et al.* consider four mapping functions *f*. These functions are the distance from the barycenter, the distance from the center of the bounding shape, the integral geodetic distance and the topological distance from the curvature extrema [36], and allow the extraction from the original model of the centerline skeleton. Then, for each node of the skeleton, the function ϕ value has to be computed to obtain the size graph. The authors also suggest measuring a set of features of the corresponding region on the model, such as the area of the region or the minimum, average and maximum distance of the barycenter of the region to vertices region. To compare models, Biasotti *et al.* use he matching between their size functions, as d'Amico *et al.* presented in [19].

2.1.4 Image-based Descriptors

This type of approach represents a three-dimensional model through a set of 2D spaces. The basic idea behind the image-based descriptors is that when images of 3D objects captured from the same viewpoints are similar, then the three-dimensional objects are as well. Thus, several researchers have managed to reduce the problem of comparison of shapes of 3D objects to a problem of image matching, which is a well known problem and with good retrieval results.

Spin Images

Johnson and Herbert proposed a recognition system of three-dimensional objects [28] based on matching surfaces using a representation of spin images. They use oriented points on the surface of the model, *i.e.*, points associated with the normal to the surface at that point, to produce the spin images. Each oriented point corresponds to a spin image and sets the local system of coordinates using the tangent plane.

They use a 2D accumulator indexed by two coordinates defined with respect to the oriented point to create the spin image and where the dark areas correspond to bins that contain many points projected

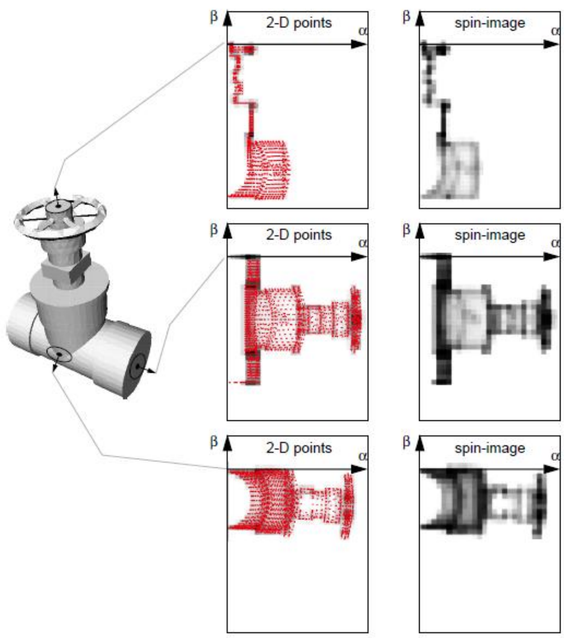

Figure 2.3: Spin-images for three oriented points on the surface of a model of a valve (Figure taken from [27]).

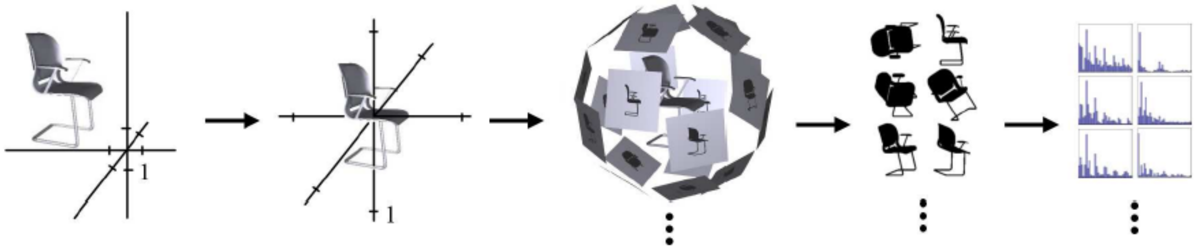

Figure 2.4: Steps of extracting the LightField Descriptors for a 3D model.

(see Figure 2.3). For 3D object matching, spin images can be constructed for each vertex of the surface mesh, producing a set of 2D histograms representing the geometry of object.

This approach was later improved for 3D shape retrieval by de Alarcón *et al.* [5]. de Alarcón *et al.* suggested data reduction through clustering of the set of spin images using a self organizing map algorithm to group similar spin images, followed by a clustering algorithm. More recently, Assfalg *et al.* presented a method to use spin images for retrieval by content of 3D objects based on global object features [6].

Light Field Descriptor

Chen *et al.* proposed a descriptor based on silhouettes from several different viewing directions [18]. The Light Field Descriptor method represents a three-dimensional model as a collection of 2D images rendered from uniformly sampled positions on a viewing sphere located around the model (see Figure 2.4). Each viewing position yields a 2D image representing the silhouette of the 3D object. The light field descriptor encodes one hundred orthogonal projections of an object using Zernike moments for the filled contour and the Fourier descriptors for the contour of the 2D views, to produce feature vectors that describe the object.

To improve the robustness invariance a set of ten light field descriptors is applied to each three-dimensional model, which leads to the one hundred orthogonal projections. These ten descriptors are created from different orientation systems of the camera. Thus, the dissimilarity between two models is the minimum difference between all combinations of light fields.

2.1.5 Other Approaches

In the previous sections, we described some methods to represent 3D shapes according to the technique behind the descriptor computation. However, there are some algorithms that do not fit into any of the categories previously introduced, either because they do not use any of these techniques or just because they combine several methods to achieve a distinct result, *e.g.*, the 3D Zernike Moments [40] or the Generalized Shape Distributions [33].

Our framework is prepared to support most of these shape descriptors. To do this they must be represented by a multidimensional feature vector.

The descriptors themselves are used to represent numerically the object, thus allowing comparisons of that object with others. To perform these comparisons some similarity measures can be used.

2.2 Similarity Measures

Since it is not trivial to compare three-dimensional representations of objects, to measure the similarity between objects, is measured the distance between descriptors, which are nothing more than points in a multi-dimensional space.

There are various approaches to measure the distances between two points in space, however we will only present the Euclidean, the Manhattan and Chebychev distances, which are particular cases of the Minkowsky distances, and the Mahalanobis distance. A more extensive explanation of similarity measures often used in content based retrieval solutions published by Castelli [15] or a theoretical description of distance functions presented by Hervé Abdi [1], can be seen to a better comprehension of this topic.

Minkowsky distance

The Minkowsky distance of degree p, also called p-distance, between two points in a n-dimensional space is given by:

$$d_p = (\sum_{i=0}^{n}(x_i - y_i)^p)^{\frac{1}{p}}$$

However, this equation is not applied in practice. Normally, the parameter p is fixed in values commonly used. For example, Manhattan distance is the Minkowsky distance with $p = 1$, Euclidean distance to the Minkowsky distance with $p = 2$ or Chebychev distance to the Minkowsky distance with p $= \infty$. The corresponding distance functions are defined as:

- **Manhattan distance:** $d_1(x, y) = \sum_{i=0}^{n}(x_i - y_i)$;
- **Euclidean distance:** $d_2(x, y) = \sqrt{\sum_{i=0}^{n}(x_i - y_i)^2}$;
- **Chebychev distance:** $d_\infty(x, y) = max_{i=0..n}(x_i - y_i)$.

Manhattan, Euclidean and Chebychev are simple and fast to compute, being normally used. However, in some cases the results obtained by these measurements do not fulfil the needs of retrieval solutions. Thus, to solve problems caused by poorly scaled or highly correlated coefficients of a descriptor, is often used the Mahalanobis [34] distance.

Mahalanobis distance

Normally used in cluster analysis and other classification techniques to measure the distance between probability distributions, the Mahalanobis distance is a computationally expensive generalisation of the Euclidean distance. It differs from Euclidean distance in that it takes into account the correlations of the data set and is scale-invariant, *i.e.*, not dependent on the scale of measurements. This measurement is based on correlations between variables by which different patterns can be identified and analysed. It is a useful way of determining similarity of an unknown sample set to a known one. The Mahalanobis distance is defined in terms of a covariance matrix S, which measures a tendency to vary between two features, as given by the function d_M:

$$d_M = det|S|^{\frac{1}{2}}(x - y)^T S^{-1}(x - y).$$

These measures of similarity which have been presented are some of the simplest and well known approaches to measure distances between two points in space and, therefore, used to compare three-dimensional models. However, to date, has not been possible to have none of them integrated in the current prototype.

2.3 Segmentation

The segmentation is also a very interesting domain in the three-dimensional models domain and is used in various applications. To segment an object, a human takes into account various factors, which could produce a segmentation that differ, depending on the domain and even application, from person to person. As such, in computer graphics, there are several approaches that try to generate segmentations that are close to the segmentation generated by humans. These approaches range from the clustering of mesh faces to the use of the properties of the mesh or the object's skeleton.

The final results of the segmentation approaches may differ, depending on the factors used in the segmentation or the expected result. According to the type of segmentation carried out, the segmentation techniques can be classified into two distinct categories [46, 2]: part based or surface based. The part based categories are based on the curvature of the object and can assign semantics to segment, decomposing the object according to human perception of the object. The surface based categories are based on criteria related to the surface, segmenting the object according to the geometric properties of its surface.

As mentioned above, there are several approaches for the segmentation of three-dimensional models. We will then explain a little better just two of them, the hierarchical clustering and the multiscale shape descriptor.

Hierarchical Clustering

Hierarchical clustering is based on a tree that is created and whose root node is the object itself, the leaves are the triangles of the mesh and the intermediate nodes are the various segments obtained over the course of construction of the tree, by grouping each pair of adjacent clusters with lower cost. In recent solutions presented, using the clustering of faces, it is initially set up a dual graph which aims to support segmentation. Together is created a tree (that is built during the execution of the algorithm) whose leaves are the clusters that represent the faces. Whenever two nodes of the dual graph are merged it is created a parent node in the tree and the algorithm proceeds as explained above. Each pair of nodes is assigned a cut value, being chosen the edge that has a lower cut value in order to be contracted. The value of the cut is calculated based on a primitive, *i.e.* a given cluster is approximated, in this case, to a plan. Attene *et al.*, in the Hierarchical Fitting Primitives [8], besides using plans, are also used spheres and cylinders to calculate the cost of merging clusters. In more recent work of Attene *et al.* [9], emphasis is given to the convexity of the object. As such, it is created a hierarchy similar to that described above where the cost of fusion is calculated based on the convexity of the cluster.

Multiscale Shape Descriptor

The surface-based segmentation methods use descriptors, which should capture the main features of a given surface and should be based on the shape of the object. One of these methods, called Tailor, was proposed by Mortara *et al.* [37] and it uses the paradigm of blowing bubbles: a set of spheres of increasing radius r is drawn, whose centers are at each vertex of the mesh, and whose radius represents the scale at which the shape is analyzed. With this information it is possible to extract some information from the object. If there is only one intersection between the model and the sphere, the surface around the point is considered equivalent to a disk and can be in the presence of a sharp or round area or in the presence of blended vertices. If there are two intersections, it is a cylindrical or tapered area, depending on the threshold between the two rays of the cuts. If the number of intersections is more than two we are in the presence of a branch of the object.

In this section we made a small presentation of some segmentation algorithms which we incorporated into our system. However, many more approaches to segment 3D objects exist and are used in different applications. The segmentation of a three-dimensional object is often used to calculate the most representative view of that 3D object.

2.4 Best View Selection

Commonly, 3D objects are represented in two dimensions, as in computer monitors or printed documents. During the process of converting 3D information into 2D information, the amount of 3D information that can be communicated largely depends on the viewpoints selected. Although it is possible to create projections of several points of view, it is important to identify the most ideal viewpoint, because first impressions have a great influence on humans.

According to a psychological experiment of viewpoint selection by Blanz *et al.* [11] (see Figure 2.5), humans believe that the most optimal viewpoint is the one that satisfy the following conditions:

- Many important features are included in each projection.
- Stability is preserved if the viewpoint changes slightly.
- Occluded parts are few.

The problem of automatically selecting the pose of a 3D object that corresponds to the most informative and intuitive view of the shape is known as the best view problem. There are several situations where 3D models should be shown in the form of few representative views, called salient or best 2D views. The saliency of a 2D view of a 3D object can be defined as a function of some view-dependent shape properties. The viewpoint that maximizes this scoring function is the one that defines the best view.

Mortara *et al.* [38] proposed that the best view should be evaluated taking into consideration the meaningful features of a 3D object, in order to maximize the visibility of salient components from the

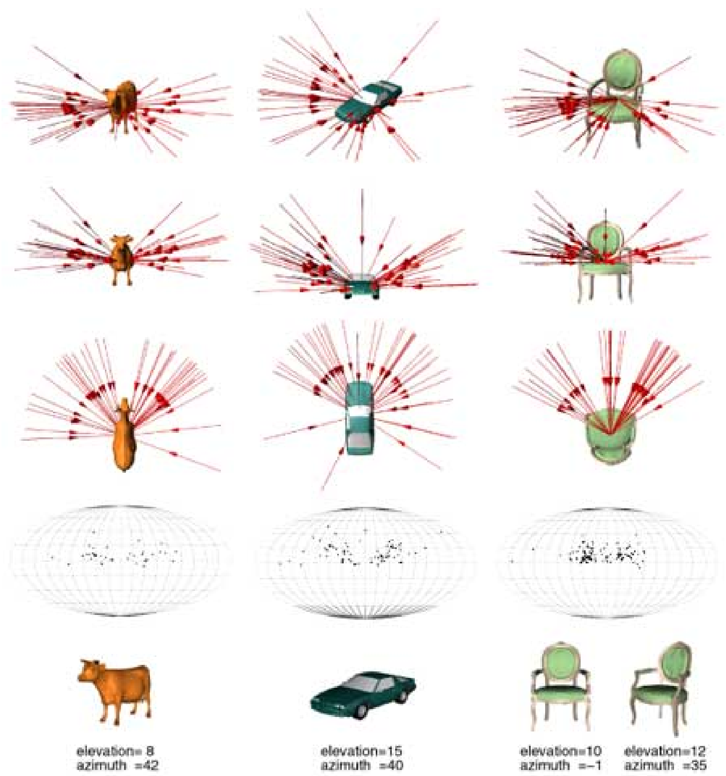

Figure 2.5: Psychological experiment about selection of optimal viewpoints (Figure taken from [11]).

context or from the application point of view. This means that the quality of a view should be related to the semantics of the displayed features. In particular, the authors propose a new scoring function for selecting the optimal view point which blends visibility criteria with reasoning on the proportion of relevant features which are visible in a given direction. The salient features may be given by a previous step of annotation or automatically extracted by means of semantic-driven segmentation algorithms, like the Fitting Primitives, the Reeb graph, the Tailor or the Plumber.

In a different work [31], Hamid Laga defined the best views of a 3D object as the views that allow to discriminate a object from other objects in a database. The solution the author proposes is based on the assumption that 3D models belonging to the same class of shapes share the same salient features that discriminate them from other classes of shapes. The major difference of this approach is that they consider a 3D model within a context, the data collection to which the 3D model belongs to. The model is described with a set of view-based descriptors computed from different viewpoints. Then a a classifier is trained, in a supervised manner, on a collection of 3D models belonging to several shape categories. Finally, the classifier learns the set of 2D views that maximize the similarity between shapes of the same class and also the views that discriminate shapes of different classes.

In our framework we have integrated the Mortara *et al.* [38] work. So, selecting a model that has previously been segmented by Fitting Primitives, Reeb graph, Tailor or Plumber algorithms and choosing the corresponding algorithm is possible to see the best view of that model. Thus, for example, it can be realizable to create thumbnails of the object's more relevant view.

2.5 Shape Repair

Sometimes, in the three-dimensional scanning process, the meshes generated may contain degenerate elements, overlapping or self-intersecting parts, surface holes or other types of defects. However, when researchers develop algorithms for geometry processing and analysis assume that the polygonal meshes do not contain these types of defects. Therefore, it is very important that there are algorithms that allow the enable the fix/repair of these defective polygonal meshes.

Algorithms for mesh repairing can be classified into two main categories: surface-based and volume-based methods. Surface-based algorithms try to remove the defects by modifying the input only locally. These methods are not invasive, as they act only where necessary, but unfortunately they typically fail on

Figure 2.6: The raw model of a chair with defects (left). The same model repaired with the Attene's algorithm (right) (Figure taken from [7]).

complex configurations or require the user to interact to resolve ambiguities. Volume-based algorithms use the input to define a new (implicit) surface which is eventually tessellated to produce the output mesh. These methods are typically much more robust and can treat a wide range of configurations, but unfortunately the introduce modifications in all the parts of the surface, regardless of the presence or absence of defects.

In [7], Attene presented an automatic algorithm that strives to convert a low-quality digitized polygon mesh into a single manifold and watertight triangle mesh without degenerate or intersecting elements (see Figure 2.6). This approach differs from most others which globally resample the model to produce a fixed version. The algorithm tries to modify the input mesh only locally within the neighbourhood of undesired configurations. It strives to modify the mesh as little as possible, which makes the algorithm less aggressive than typical volume-based approaches. It is also tailored to treat a specific class of meshes. After having converted the input to a single combinatorial manifold, the algorithm proceeds iteratively by removing growing neighbourhoods of undesired elements and by patching the resulting surface until all the defects are removed. This repairing approach has as main domain raw digitized solid objects. With this it is possible to assume that the resulting mesh should be a single connected manifold bounding a polyhedron and the sampling density should not vary significantly from one part of the mesh to another. Attene's algorithm has proved to be more computationally efficient and produce more accurate results while using fewer triangles than similar existing algorithms.

2.6 Retrieval Evaluation

Information Retrieval (IR) consists in finding material of an unstructured nature (text, video, audio, 3D objects, etc.) that satisfies an information need from within large collections, as defined by Manning in [35]. However, in Information Retrieval systems the query result, normally, does not identify a single object in the collection, because the probability of finding two identical objects is virtually nil (unless the objects were copies of each other). Instead, several objects might match the query (perhaps with different degrees of relevancy).

Since the Information Retrieval does not produce single results, but a set of the most similar, it is necessary to measure the similarity of these results. In this section, we summarize the most commonly used statistics for measuring the performance of a shape descriptor in a content-based retrieval system [50].

Precision-Recall curve

Two well known effectiveness measures are precision and recall. Precision is the fraction of the retrieved objects which is relevant to a given query and recall is the fraction of the relevant objects which has been retrieved from the database. Precision-Recall curves describe the relationship between precision and recall for an information retrieval method. A perfect retrieval result produces a horizontal line at precision=1.0, indicating that all the models within the query object's class are returned as the top ranked matches. Otherwise, curves that appear shifted up represent superior retrieval results.

Average Precision (AP) and Mean Average Precision (MAP)

Average precision emphasizes returning more relevant objects earlier. It is average of precisions computed after truncating the list after each of the relevant documents in turn. The mean average precision is the mean value of the average precisions computed for all queries.

E-Measure

The E-measure [50] provides a single value which describes the performance of the retrieval for a given retrieval size. The E-measures is a composition of both precision and recall, computed for a fixed number of top k matches:

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

The maximum score is 1.0 and higher values indicate better results.

Discounted Cumulative Gain (DCG)

A statistic that weights correct results near the front of the list more than correct results later in the ranked list under the assumption that a user is less likely to consider elements near the end of the list. To calculate this measure, the ranked list R of retrieved objects is converted to a list G, where element G_i has value 1 if element R_i is in the correct class and value 0 otherwise. Discounted cumulative gain is then defined as:

$$DCG_i = \begin{cases} G_i, & \text{i=1} \\ DCG_{i-1} + \frac{G_i}{\lg_2(i)}, & \text{otherwise} \end{cases}$$

To achieve the final score, this result is then divided by the maximum possible DCG:

$$DCG = \frac{DCG_k}{1 + \sum_{j=2}^{|C|} \frac{1}{\lg_2(j)}}$$

Normalized DCG

This is a very useful statistic based on averaging DCG values of a set of algorithms on a particular database. NDCG gives the relative performance of an algorithm with respect to the others tested under similar circumstances. A negative value means that the performance is below the average; similarly a positive value indicates an above-the-average performance. Being DCG_A the DCG of a certain algorithm A and $AvgDCG$ the average DCG value for all algorithms being compared in the same experiment, then the Normalized DCG for the algorithm A is defined as

$$NormalizedDCG_A = \frac{DCG_A}{AvgDCG} - 1$$

Nearest Neighbor (NN)

Nearest Neighbor is defined as the percentage of the closest matches that belong to the same class as the query. This statistic provides an indication of how well a nearest neighbor classifier would perform. Obviously, an ideal score is 100% and higher NN scores represent the potential of the algorithm in a classification application.

First-Tier (Tier1) and Second-Tier (Tier2)

First tier is defined as the precision for retrieving N objects, where N is equal to the number of relevant objects to the query stored in the database. The second-tier is similar, but is defined as the recall for retrieving $2N$ objects from the database.

F-Measure (F-M)

F-Measure is the harmonic mean of precision and recall and can be interpreted as a weighted average of the precision and recall. The F-Measure can be defined as

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

These measures provide an efficient way to assess the performance of a shape descriptor in a content-based retrieval system. Currently, the module associated with the retrieval of three-dimensional objects is not yet integrated in the current prototype, however is under development.

2.7 Summary

During the initial study carried out, we realized that it was important for users to have a tool to do more than analysis and benchmarking of shape descriptors and which could also cover other topics. As such, we decided to make a brief introduction on various topics related to the analysis of shapes, such as shape signatures, similarity measures, segmentation, best view selection, shape repair and retrieval evaluation.

As we have seen, the three-dimensional models are represented by a set of values which represent them in a multi-dimensional space, the shape signatures or shape descriptors. We divided the 3D shape descriptions algorithms into five different categories and explained some of the approaches within these categories that the current version of our framework supports.

Although our system has not yet integrated measures of similarity to allow comparisons of the representations of 3D objects, we found it also necessary to present some approaches to measure the distances between descriptors, such as the Minkowsky and Mahalanobis distances.

Being also the segmentation a very interesting field for the shape analysis, we presented two of several existing approaches to segment three-dimensional objects, the hierarchical clustering and the multiscale shape descriptor. These two approaches were chosen because they are related to segmentation algorithms that were added to our framework.

There are several situations in which the researchers want to have available the best view of a 3D model. Mortara *et al.* [38] proposed that the best view should be evaluated taking into consideration the meaningful features of a 3D object, in order to maximize the visibility of salient components from the context or from the application point of view. In turn, Hamid Laga [31] defined the best views of a 3D object as the views that allow to discriminate a object from other objects in a database.

The repair of the mesh of 3D objects is also an important topic in shape analysis, since it is not recommended to implement algorithms on defective shapes due to erroneous results that will be produced. We presented an algorithm developed by Marco Attene [7] which strives to convert a low-quality digitized polygon mesh into a single manifold and watertight triangle mesh without degenerate or intersecting elements, in an automated way.

Finally, some evaluation measures that provide an efficient way to check the effectiveness of information retrieval systems have been introduced. However, they alone are not enough to evaluate and compare different systems. This is done using benchmarks. The benchmarks are used not only for text (which is the most common), but also for multimedia, like audio, video or three-dimensional objects.

Chapter 3

Related Work

In computer science, the use of benchmarking solutions is very important, because it aims to assess the relative performance of an object, usually performing a series of standard tests and trials against it. In particular, the usage of such benchmarking solutions in multimedia retrieval helps the researchers working on that domain to compare, test and develop better algorithms.

In this chapter we present the state-of-art regarding multimedia information retrieval benchmarking, since the initial scope of our work was to analyze the descriptors used in retrieval. We start by introducing some examples of benchmarks for several domains of multimedia retrieval. Then, we give a special attention to the 3D shape benchmarks solutions, which are more related with the purpose of our work.

3.1 Information Retrieval Benchmarks

In recent years several multimedia retrieval benchmarks with a varying focus have been created and run. Below we present some of the most relevant, starting by the best known in text retrieval.

TREC

The Text REtrieval Conference (TREC)[1], co-sponsored by the National Institute of Standards and Technology (NIST) and U.S. Department of Defense, was started in 1992 as part of the TIPSTER Text program. Its aim was to support research within the community of information retrieval by providing the infrastructure necessary for large-scale evaluation of methodologies for text retrieval. For each TREC, participants run their own retrieval systems on a test set of documents and questions provided by NIST. Then, NIST receives a list of the retrieved top-ranked documents and pools, judges and evaluates the results. Finally, participants can share their experiences in a workshop. The TREC test collections and evaluation software are available to retrieval research community, allowing each research group to evaluate their own retrieval systems at any time. TREC has successfully improved the state of the art in information retrieval, as well as facilitating the technology transfer. Today, TREC is in the eighteenth edition.

MIREX

Music Information Retrieval Evaluation eXchange (MIREX)[2] is an annual evaluation of Music Information Retrieval algorithms that takes place in conjunction with the International Symposium on Music Information Retrieval (ISMIR). The goal of this contest is to compare state-of-the-art algorithms and systems relevant for Music Information Retrieval.

[1] The TREC Website. Available at http://trec.nist.gov Last visited October 2010.

[2] MIREX Website. Available at http://www.music-ir.org/mirex/wiki/2011:MIREX_HOME Last visited October 2010.

INEX

The widespread use of XML prompted the development of appropriate searching and browsing methods for XML documents. This explosion of XML retrieval tools requires the development of appropriate testbeds and evaluation methods. As part of a large-scale effort to improve the efficiency of research in information retrieval and digital libraries, the INitiative for the Evaluation of XML Retrieval (INEX) organises an international, coordinated effort to promote evaluation procedures for content-based XML retrieval [29]. The Project provides an opportunity for participants to evaluate their retrieval methods using uniform scoring procedures and a forum for participating organisations to compare their results.

TRECVid

TRECVid started as a task in TREC but has since become an independent workshop on the evaluation of video retrieval systems [25, 49]. The goal of the workshop is to encourage research in information retrieval by providing a large test collection, uniform scoring procedures and a forum for organizations interested in comparing their results. The strong participation has also made this benchmark important for image retrieval where evaluation can be performed on extracted video key frames.

ImagEVAL and ImageCLEF

Another initiative is ImagEVAL[3], financed by the French research foundation and with participants mainly from the French research community and mainly on visual retrieval of images and image classification. The ImagEVAL project relates to the evaluation of technologies of image filtering, content-based image retrieval (CBIR) and automatic description of images in large-scale image databases.

ImageCLEF[4] is the cross-language image retrieval track which is run as part of the Cross Language Evaluation Forum (CLEF). This event focuses on image retrieval from multilingual repositories and combines visual and textual features for multi-modal retrieval.

Most of these information retrieval benchmarks are based on the organization of conferences or workshops. The organizations give the necessary infrastructure for the research communities to perform their studies. They provide test collections, uniform scoring procedures and means for participants share their results. As a result, the state of the art in information retrieval has successfully been improved.

In the information retrieval field, the several existing benchmarks are relatively stable and used in a peaceful manner, the same is not true with benchmarks related to 3D shapes. In that area, there are several solutions, but there is no absolute consensus on them.

3.2 3D Shape Benchmarks Solutions

Three-dimensional objects may be very complex, both in terms of the data structures and methods used to represent such objects, as well as in terms of the topological and geometric structures of the objects themselves. The primary goal in the 3D, as well as in other retrieval domains, is to design algorithms with the ability to effectively and efficiently execute similarity queries.

Like in previous section for multimedia information retrieval, solutions for effective testing and comparison related to 3D shape retrieval were developed. Some of the most relevant benchmarks are presented beneath.

[3]The ImageEVAL Website. Available at http://www.imageval.org/

[4]The ImageCLEF Website. Available at http://www.imageclef.org/

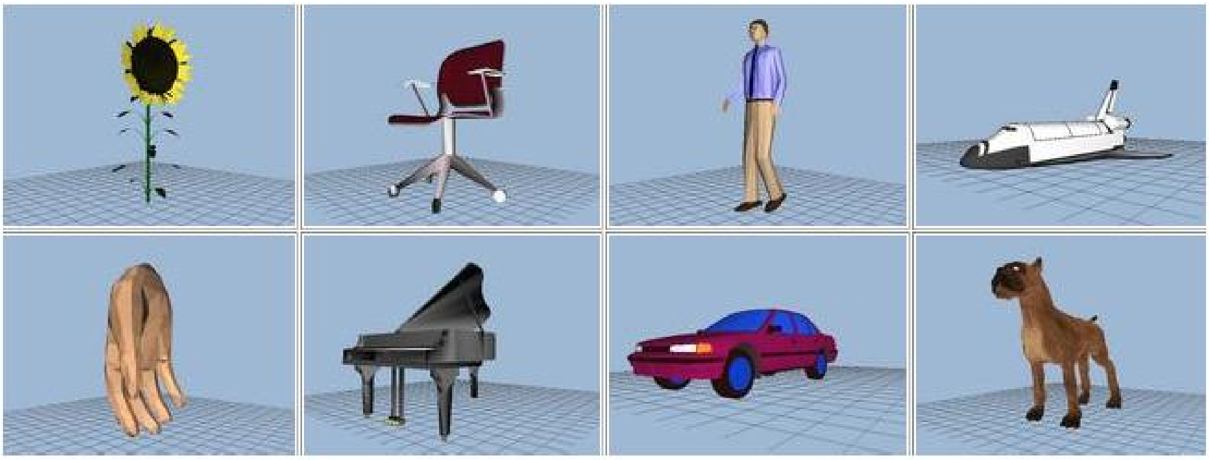

Figure 3.1: An example set of objects from the Princeton Shape Benchmark.

3.2.1 Princeton Shape Benchmark

The Princeton Shape Benchmark (PSB) [47], deployed by Thomas Funkhouser team, is a publicly available database of 3D models, software tools and a standardized set of experiments for evaluating and comparing various representation methods and 3D shape retrieval techniques.

The PSB contains a compiled set of 1,814 classified 3D models (of about 36,000 models) collected from World Wide Web. The models represent real world objects (see Figure 3.1) and are classified according to function and shape on multiple levels of abstraction. The benchmark includes a set of hierarchical classifications and a separate training and test sets. Algorithms should be trained on the training database (without influence of the test database). Then, after all exploration has been completed and all algorithmic parameters have been frozen, results should be reported for the test database. It also provides mechanisms to define multiple classifications and queries sets which show the differences between different shape matching algorithms, by evaluating retrieval results.

This system, for the shape matching results analysis, provides tools which allow the evaluation and visualization of these same results. Thus, it contains tools for parsing classification files, creating overview web pages, viewing and parsing .off files, creating Precision-Recall plots, calculating retrieval statistics (Nearest Neighbor, First Tier, Second Tier, E-Measure and Discounted Cumulative Gain) for a given classification and distance matrix, reading a distance matrix and printing it to the screen, viewing retrieval result and producing distance and tier images. In short, it gives a standard set of tools with which researchers compare the independently run tests in a consistent manner.

An interesting feature of the benchmark is that it provides mechanisms to define multiple classifications and query sets that can be used to differentiate properties of shape matching algorithms. It also includes several query sets intended to differentiate how matching algorithms work on models with specific properties. By evaluating retrieval results with these different classifications and queries, it is possible to expose the differences between different shape matching algorithms. In summary, this benchmarking solution provides a flexible framework for comparing shape matching algorithms.

The Princeton Shape Benchmark is now widely accepted as the major benchmarking tool within 3D shape retrieval and classification.

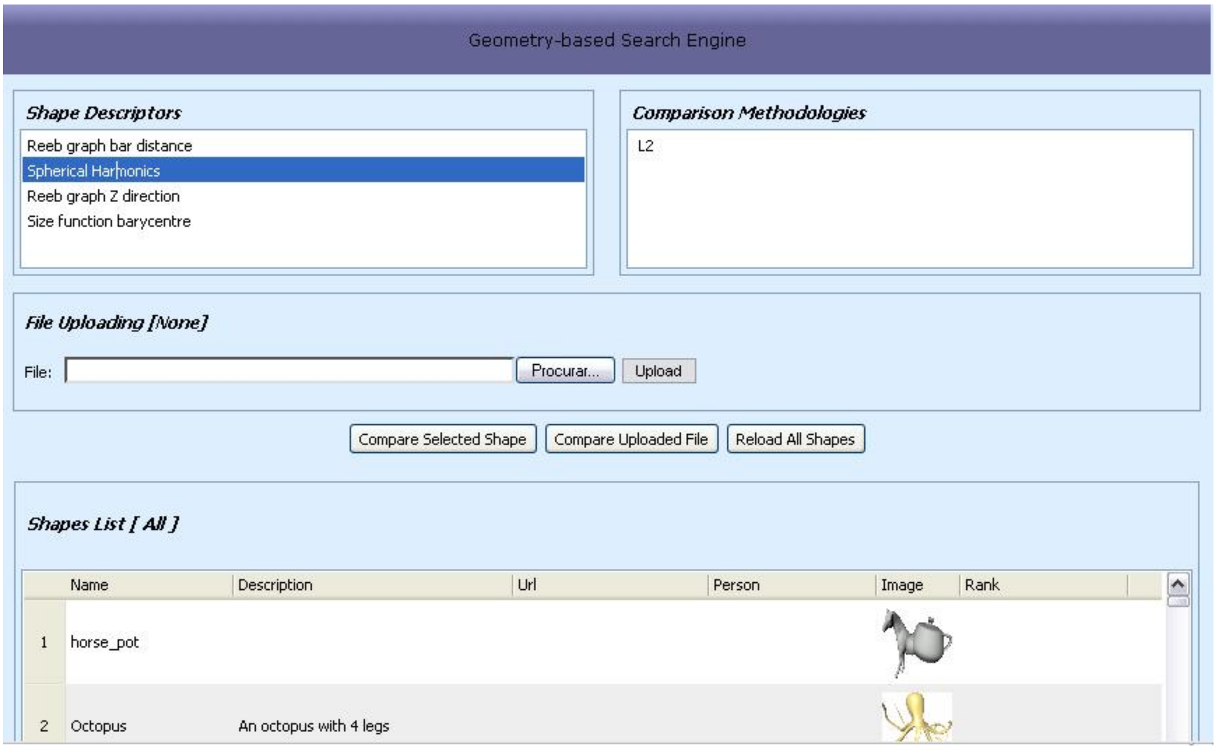

Figure 3.2: AIM@SHAPE Geometry-based Search Engine.

3.2.2 AIM@SHAPE Shape Repository

The Advanced and Innovative Models And Tools for the development of Semantic-based systems for Handling, Acquiring, and Processing knowledge Embedded in multidimensional digital objects (AIM@SHAPE[5]) is a sixth framework program project that promotes the development of new methodologies for modelling and processing the knowledge related to digital shapes.

The AIM@SHAPE pursues lasting integration both at the foundational level, by initiating a new Theory of Digital Shapes, and at the component level, by developing a Digital Shape Workbench as a common platform for shape models and software tools.

The Digital Shape Workbench (DSW) is a infrastructure were new and existing software tools and shape databases are integrated. The Digital Shape Workbench is composed by the Shape Repository, the Tool Repository and the Ontology and Metadata Repository, together with an advanced Search Framework (see Figure 3.2).

The AIM@SHAPE Shape Repository[6] is a shared repository filled with a collection of digital shapes. Its primary goal is to provide a shared collection of standard test cases and benchmarks in order to enable efficient prototyping as well as practical evaluation on real-world and large-scale shape models. Its distinctive feature is a full documentation of the most interesting geometric properties provided by detailed metadata of the common shape ontology. Additionally, the Shape Repository offers the unique possibility to download models for some shape categories in multi-resolution format at the desired Level of Detail (LOD).

[5] AIM@SHAPE website at http://aim-at-shape.net

[6] AIM@SHAPE Shape Repository at http://shapes.aim-at-shape.net

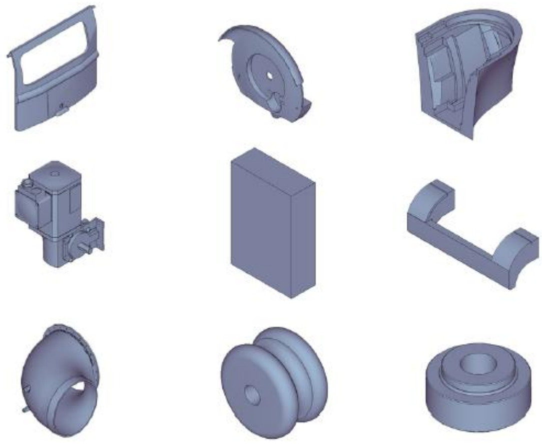

Figure 3.3: An example set of objects from the Engineering Shape Benchmark.

3.2.3 Engineering Shape Benchmark

In the generic models domain, the Princeton Shape Benchmark [47] has become the standard and has been used to evaluate various shape representations. The repository of forms at AIM@SHAPE also allows researchers to compare different algorithms. However, the Princeton Shape Benchmark and other datasets are not appropriate for the engineering domain, since engineering shapes often have high genus, but also because 3D models on datasets like PSB have an high level of abstraction and classify models primarily based on function and secondarily on shape [47]. Therefore, it was necessary to develop a system benchmarking for engineering artefacts, because shape descriptors might perform differently on an engineering database compared to the Princeton Shape Benchmark, for example.

Engineering Shape Benchmark (ESB) [26] was designed for the engineering domain, especially for CAD models. It is composed by 867 3D models (see Figure 3.3) collected from multiple sources like National Design Repository [45], World Wide Web, industry and CAD models created by undergraduate design students at Purdue University. These models are partitioned in three super-classes and, within each super class, are classified into clusters of similar shape. In terms of evaluation methodology, it was used Precision-Recall curves and E-Measure. Additionally to the standard measures, was also quantified the performance of shape representation methods in relation to a base method (3D shape distributions) as an Average of Differences (AOD).

3.2.4 SHREC

The 3D Shape Retrieval Contest (SHREC[7]) is an annual contest where the effectiveness of 3D shape retrieval algorithms can be evaluated. This contest has been organized since 2006 and every year its tracks change. Today, SHREC is in the fourth edition of the contest and since 2009 it is organized in conjunction with the Eurographics Workshop on 3D Object Retrieval.

This contest is composed by different tracks that use specific collections. Some of these collections are extracted from benchmarks and repositories we described above.

The 2009 contest consisted of four tracks, but we are going to focus on just two of them. One of the tracks consisted in the evaluation of the performance of three-dimensional shape retrieval approaches on a new generic 3D shape benchmark. It was expected that the contestants, in response to a given set of queries, returned an ordered ranked list together with the similarity scores between the target models and each query. The dataset used for the generic shape retrieval contest was from the NIST

[7] http://www.aimatshape.net/event/SHREC

generic shape benchmark [21] and consisted of 720 3D models classified in 40 categories (18 models in each category). The evaluation of the shape retrieval contest was based on standard measures: Precision-Recall curve; Average Precision and Mean Average Precision; E-Measure; Discounted Cumulative Gain; Nearest Neighbor, First-Tier and Second-Tier.

Another interesting track at the 2009 contest was related with the retrieval of 3D models which exhibit a relevant similarity in the shape structure. The contestants of this track were supposed to retrieve a ranked list of models given a query for each class, similar to what was done in the previous track. However, the retrieval method should only depend on the characteristics of the shape, since the models did not contain any other information besides its mesh. The track featured a new set of 3D models, without any metadata and color, which contained structural differences between them. The dataset contained 200 models, classified in 10 main classes and each main class was divided in 2 subclasses with 10 models each. Matching models from inside a class are highly relevant and matching results between models the two pair class are marginally relevant. The performance of this shape retrieval task was measured using First and Second Tier, Precision-Recall and presented as the F-measure.

3.2.5 MyMultimediaWorld.com

All the previous benchmarking solutions only provide offline solutions. The databases are obtained and each research group makes its own analysis of 3D shape descriptors locally, having little contact with the analysis made by other groups. In turn, Bonhomme *et al.* deployed MyMultimediaWorld.com [12, 32] (MMW.com), an online platform for sharing various types of media, including video, image, audio and 3D objects. This platform follows the MPEG standards, using MPEG-4 for the representation of media and MPEG-7 for its description.

The MMW.com database contains various types of media, but we only focus on the 3D model related that is more relevant to our work. The database of 3D objects is organized in a hierarchical tree of categories for the objects classification. This classification was created from three existing classifications: the classification of the Princeton 3D database (over 1,800 objects and 90 hierarchical categories), the MPEG 3D database (containing about 1,400 objects classified into 21 categories) and 3D object database published by Taschen (with 1,000 objects classified into 20 categories).

This system presents, in a web page, some information about the three-dimensional object chosen, like title, upload data, owner, category, the url pointing to a detailed description, tags and free text description. Also, it gives the possibility to perform similarity searches (see Figure 3.4) and to see benchmark results (see Figure 3.5). The similarity search presents the most relevant similar objects and some evaluation measures (true positive, First-Tier, Second-Tier, Precision and Recall) for all descriptors. The benchmark with the 3D API compression contains object properties (number of components, triangles and vertices, and the size of the file), graphic with the dependency between distortion and bit stream size, the compression gain of each method, and the average decoding and encoding times.

The great advantage of MMW.com is to have an API that allows the easy integration of new descriptor extraction algorithms as plug-in libraries. This API is designed to integrate 3D object retrieval technology from third-party researchers independent of the complexity of the global system.

When new content is added, the benchmark is automatically executed and the results are shown in real time. Similarly, when a new algorithm is integrated, it is executed for all files in the database and the benchmark is updated automatically.

Figure 3.4: Example of 3D object retrieval benchmarking.

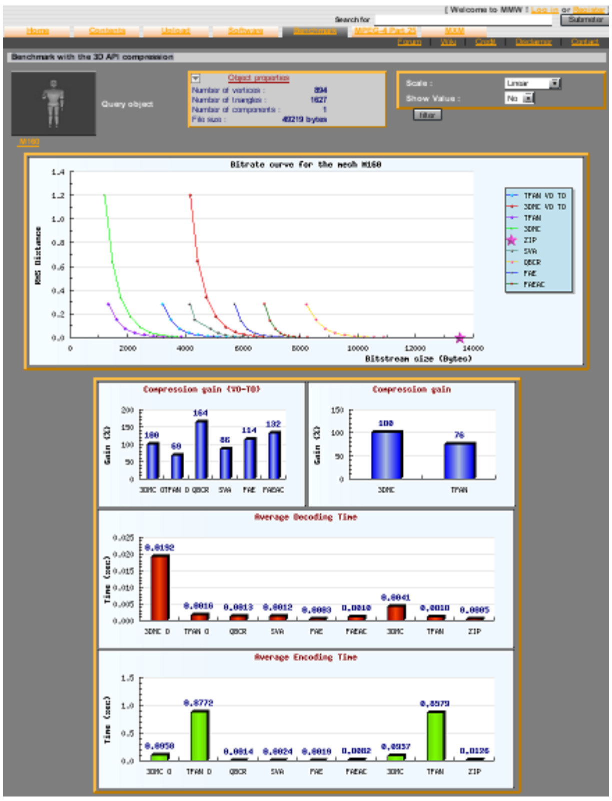

Figure 3.5: Global 3D compression benchmarking result per-object.

3.3 Discussion on 3D Shape Benchmarks

As we saw above, there are some 3D shape benchmark solutions and each of which has its characteristics. Next, we will make a brief comparison between the different 3D benchmark solutions and which can be summarized by Table 3.1.

The Princeton Shape Benchmark (PSB) is accepted as the major benchmarking tool within 3D shape retrieval and classification. The PSB contains generic 1,814 models and provides tools which allow the evaluation and visualization of the shape matching results. Thus, it contains tools for parsing classification files, creating overview web pages, viewing and parsing .off files, creating precision-recall plots, calculating retrieval statistics for a given classification and distance matrix, reading a distance matrix and printing it to the screen, viewing retrieval result and producing distance and tier images. The evaluation measures used to evaluate the different shape descriptors can be seen at Table 3.1.

The AIM@SHAPE Shape Repository (A@S SR) is a shared repository with a collection of 505 models, containing a total of 1152 shapes. Its most striking feature is a full documentation of the most interesting geometric properties provided by detailed metadata of the common shape ontology.

Unlike the previous two benchmarks, the Engineering Shape Benchmark (ESB), as its name suggests, is suitable for the field of engineering, especially for CAD models. This benchmark database consists of 867 models, divided into three super-classes, which are categorized into 45 classes. Apart the use of some standard measures, like precision-recall curves and E-measure, for evaluating the shape representations, it was also used the Average of Differences (AOD) to quantify the performance of shape representations methods with the 3D shape distributions.

The 3D Shape Retrieval Contest (SHREC) is an annual contest where the effectiveness of 3D shape retrieval algorithms is evaluated. The 2009 edition contained four tracks, however only two of them are particularly relevant in the context of our work: the generic retrieval on new benchmark and the structural shape retrieval tracks. The generic retrieval on new benchmark track (called Track 1 in the Table 3.1) was composed by a dataset with 720 generic 3D models categorized in 40 classes. To evaluate similarity scores with the target models and return an ordered ranked list along with the similarity scores for each query where used the similarity measures shown on Table 3.1. The structural shape retrieval track (called Track 2 in the table 3.1) required the participants to retrieve a ranked list of models given a query for each class, using only the object mesh. The dataset was composed by 200 3D objects, classified

		PSB	A@S SR	ESB	SHREC'09		MMW.com
					Track 1	Track 2	
Collection Domain		Generic	Generic	CAD	Generic	Generic	Generic
Collection Size		1,814	505	867	720	200	2,770
Evaluation Measures	PR	Yes	No	Yes	Yes	Yes	Yes
	AP	No	No	No	Yes	No	No
	MAP	No	No	No	Yes	No	No
	E-M	Yes	No	Yes	Yes	No	No
	DCG	Yes	No	No	Yes	No	No
	NDCG	Yes	No	No	No	No	No
	NN	Yes	No	No	Yes	No	No
	Tier1	Yes	No	No	Yes	Yes	Yes
	Tier2	Yes	No	No	Yes	Yes	Yes
	F-M	No	No	No	No	Yes	No
Provide Evaluation Tools		Yes	No	No	No	No	No
Model Information		No	Yes	No	No	No	Yes
Online		No	No	No	No	No	Yes

Table 3.1: Comparison of 3D Benchmark Solutions.

in 10 classes and each class was then subdivided in two subclasses with 10 models each. This track was measured using precision-recall (PR), first-tier (Tier1), second-tier (Tier2) and F-Measure (F-M).

All these benchmarks only provide offline solutions. The MMW.com, in turn, is an online platform and does not have this limitation. Additionally, it also has a modular architecture that allows easy addition of new descriptors.

3.4 Summary

As seen, most benchmarks only provide offline solutions. The databases are obtained and each research group makes its own analysis of 3D shape descriptors locally, having little contact with the analysis that other groups do or did. Only one solution is available online (the MMW.com), but this benchmark solution for 3D retrieval is part of a larger and complex platform for multimedia analysis.

During the development of our work, we identified a gap in the area that went far beyond the retrieval benchmarking. There are no tools that allow comparison across most of the topics studied in the area of 3D object retrieval. Thus, our work has evolved from a workbench for shape descriptors, the original purpose, to a framework for shape analysis.

Therefore, with our work, we intend to solve these problems by creating a dedicated and simple web-based system that allows researchers working in the area of analysis, classification and retrieval of 3D objects, to study and compare objects description techniques. Our solution is modular and scalable, allowing the easy addition of new algorithms and collections.

Chapter 4

Framework for Shape Analysis

Now that we have briefly analysed the background and state-of-the-art on multimedia information retrieval benchmarks (in particular 3D shape benchmarks) we focus on the motivation behind our work and the proposed solution.

As we have seen before, there are some benchmarks that allow the comparison of shape matching methods using different evaluation measures. However, nowadays, there is no framework which is modular, supports various 3D shape descriptors and evaluation measures, and provides a simply way of analysing three-dimensional models on diverse topics such as, segmentation, selection of the best views or reconstruction of meshes. Therefore, our work evolved from a workbench for shape descriptors into a shape analysis framework that will address those limitations.

In this chapter we will present the approach that we followed to overcome the limitations previously mentioned.

4.1 Overview

We created a modular and scalable system which allowed the analysis, comparison and retrieval methods of three-dimensional objects. This system should enable the easy integration of new components, *e.g.*, 3D shape descriptors or evaluation measures. Additionally, it should be available online for the entire scientific community that works in this domain.

To achieve these objectives, we decided to follow a client-server architecture for our system, as depicted in Figure 4.1. We provide a web page that allows users to connect to our server via a web browser and

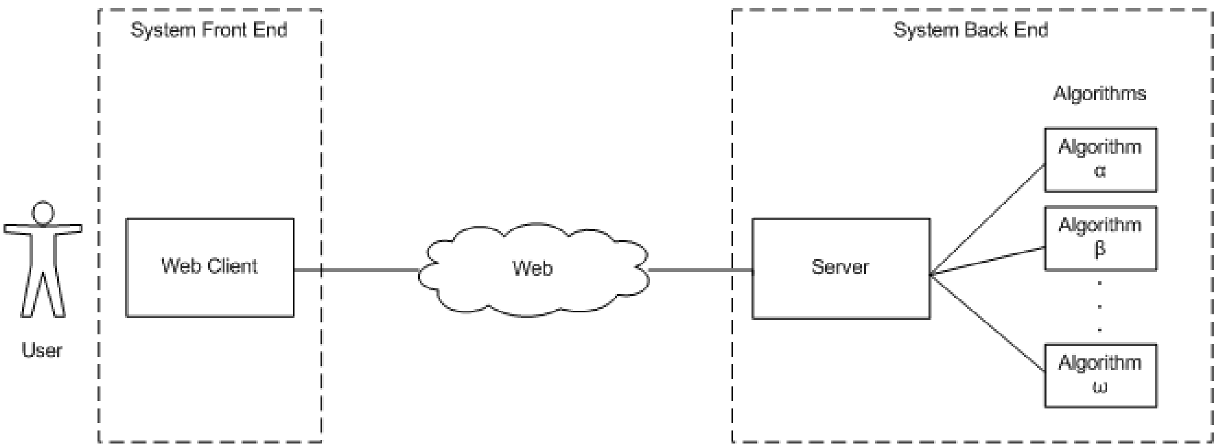

Figure 4.1: Approach overview.

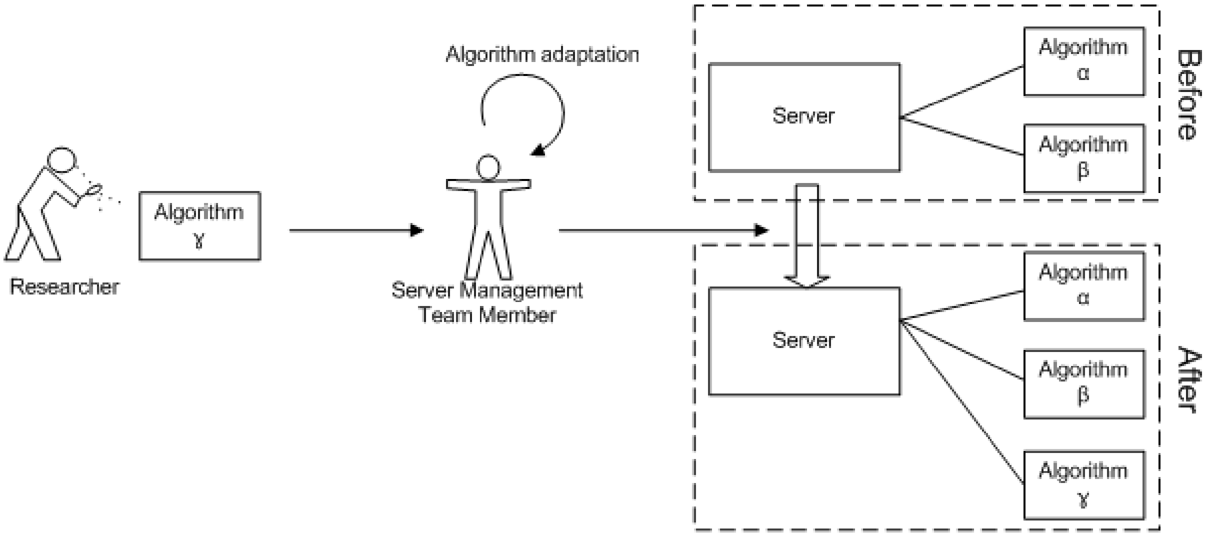

Figure 4.2: Adding algorithms to the system.

thus use all the features already integrated into our system, whether algorithms or collections of objects.

The system relies primarily on two main components: a front-end web page and a back-end site. The front-end consists on a browser navigating in web pages that serves as an interface for users to interact with the system, offering the ability to test and compare the various descriptors of 3D shapes of objects. The back-end has the core of the application and responds to requests sent by the front-end, providing the necessary services for the effective functioning of the system.

As already mentioned, the system allows the addition of new algorithms or collections of three-dimensional objects. This process of adding new algorithms is illustrated in Figure 4.2

The researchers provide us the algorithm they want to be added to the system. Then, we make the adjustments that are necessary for the algorithm to be properly integrated into the system. Finally, the algorithm is integrated into the system and becomes available to be used by other researchers. The process of adding new collections is similar to the integration of new algorithms.

Currently, some algorithms have been integrated into the system, Kazhdan *et al.* Spherical Harmonics descriptor [30], Attene's Hierarchical Fitting Primitives segmentation algorithm [8] and Mesh Fix shape repair algorithm [7], Mortara *et al.* Tailor segmentation algorithm [37] and Best View Selection algorithm [38].

4.2 Architecture

As already mentioned, our solution is based on a simple client-server architecture, depicted in Figure 4.3. The clients are the web browsers which interact with the server. In this kind of architecture, the server is the main component and responsible for the system itself. It is composed by a front-end web, which acts as the user interface, and a back-end site, which contain the core of the application. The back-end is implemented in C++ and the front-end in Flex[1]. These two components communicate through the exchange of XML messages sent over sockets.

The core of the application is divided into different modules to facilitate the addition of new features. We identified seven main components: one responsible for the shape information, other for the calculation of the shape descriptors, another for the similarity estimation, one for the selection of the best view of a model, other for the retrieval performance analysis, another for the segmentation and, finally, one for

[1]http://www.adobe.com/products/flex/

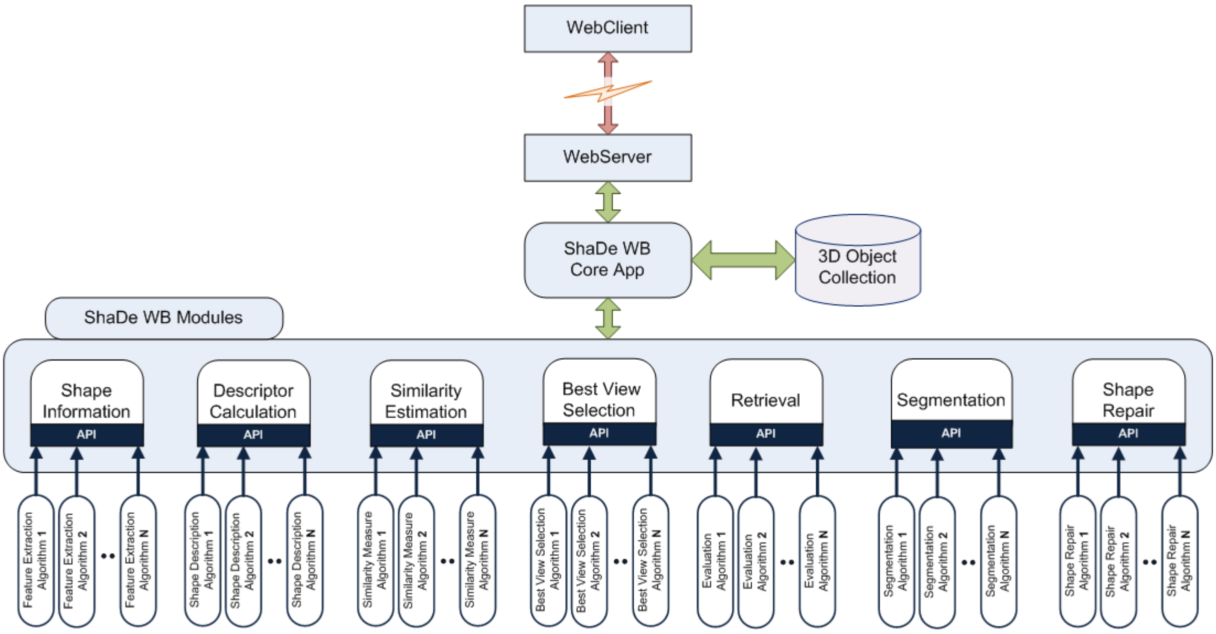

Figure 4.3: Architecture of the proposed solution.

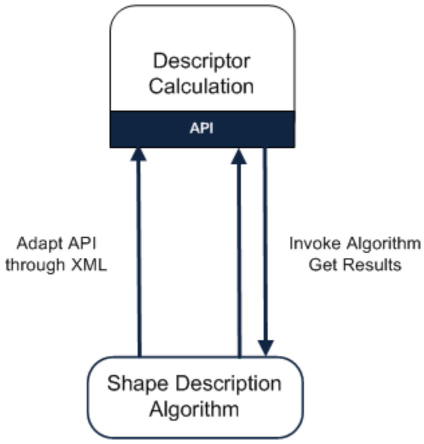

Figure 4.4: API adaptation process.

the reparation of the shapes[2].

As we can see in the architecture sketch (Figure 4.3), we then subdivided each module in sub-modules for each particular task. This kind of modular and scalable architecture, as we already said, will make easier, for example, the addition of a new shape descriptor. For that, we simply add a new sub-module corresponding to the new shape descriptor to the descriptor calculation module. This type of behaviour can be easily extended to other modules or to new modules. This happens because each module has an API based on XML that provides the interface of the services and facilitates the addition of the new sub-modules. These algorithms that are added are no more than simple executable files (which for now only run on Windows). So, when someone wants to run a particular algorithm, the system will run the corresponding executable file and show the results produced (see Figure 4.4).

Additionally, it was also necessary a database with a 3D objects collection that is connected to application core. In this phase, we added SHREC 2007 Watertight collection to the system, making possible to browse in this collection and use these models to perform the algorithmic analysis. However,

[2]This module was added during a visit we made to the CNR IMATI-Ge where we had the opportunity to integrate a shape repair algorithm

like the other components in the proposed architecture, it will be relatively easy to add other collections of three-dimensional models taking into account that this operation is not performed regularly, since the classification and integration is a manual process.

4.3 ShaDe WB

After we have presented the general architecture of the system we will now explain in more detail its two main components, the back-end and the front-end.

4.3.1 Back-End

The back-end has the core of the application. Basically is composed by a server implemented in C++ that responds to the requests of the front-end by executing some algorithm and returns the results back to the front-end. This communication is accomplished through the exchange of XML messages between the two components.

Previously, we have demonstrated a more global view of the architecture of the whole system. Now, through the Figure 4.5, it is possible to see in more detail the modules that comprise the system back-end.

The back-end consists of a communication module that has the job of sending/receiving messages to/from the front-end. In case of receipt of messages, then it sends them to the Interpreter that will check the message and forward it to the right place within the Core. The Core, in turn, if is needed some kind of interaction with the collections of 3D objects in the system, will call the Collection Manager for that purpose. Then the core will call the External Handler Algorithm which is responsible for executing the binary files of algorithms and receive the results generated. Finally, the results generated are passed back to the Core that will build the XML message to send through the Communication module to the front-end.

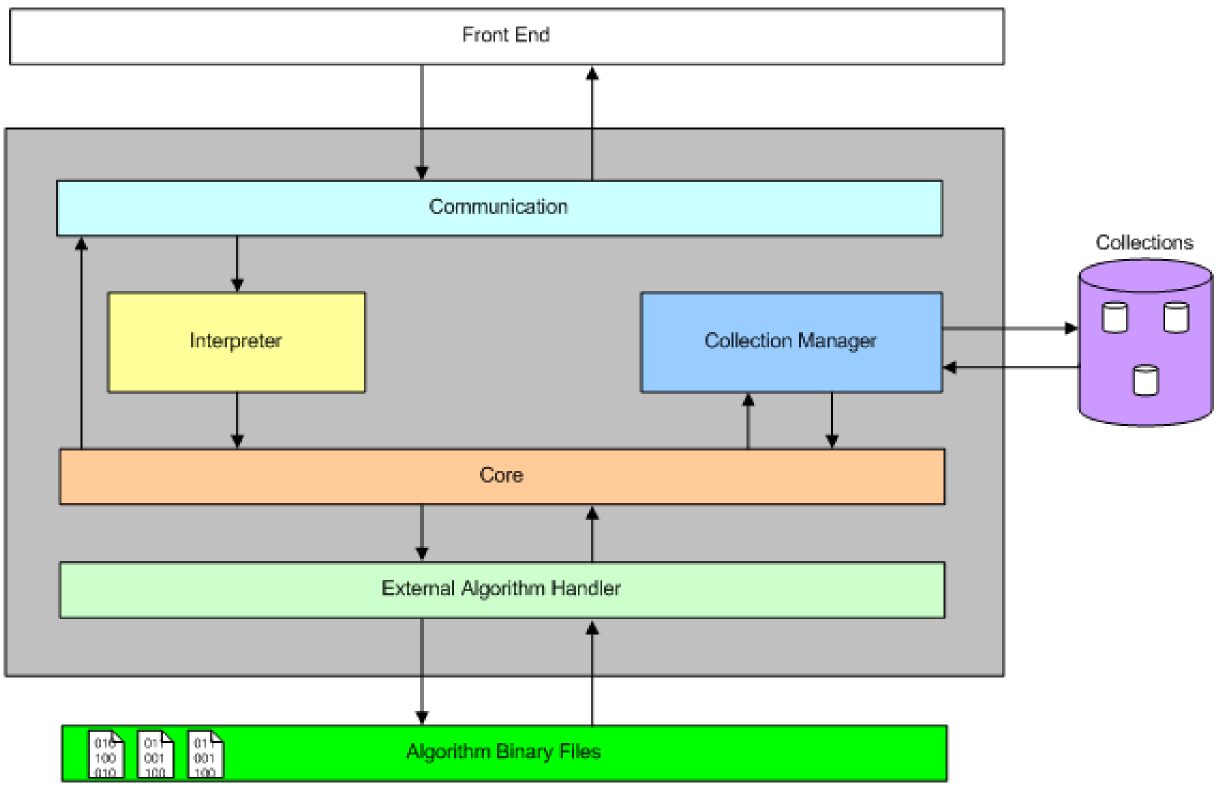

Figure 4.5: Architecture of the system's Back-End.

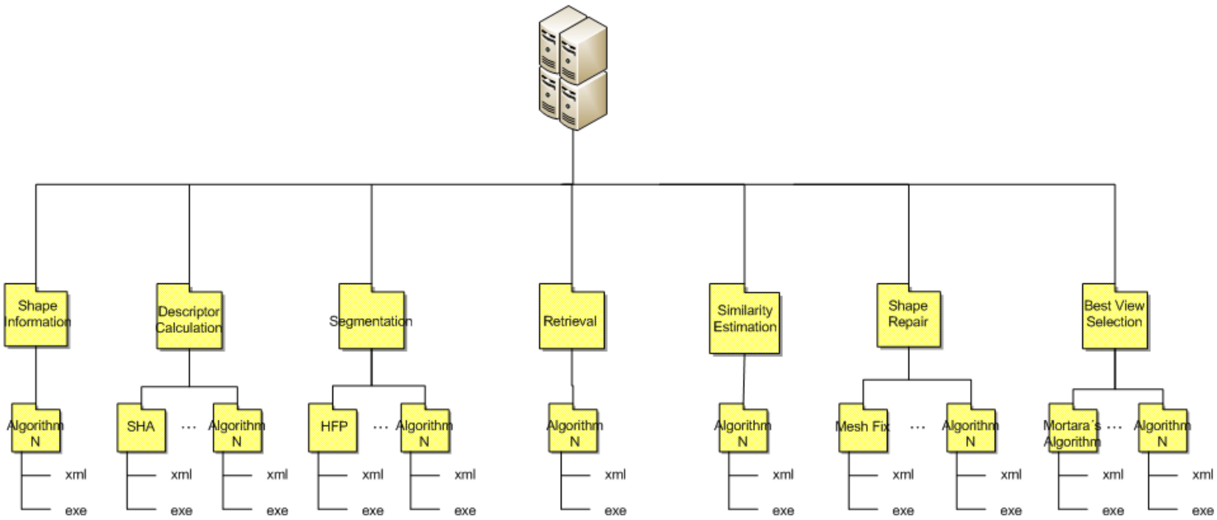

Figure 4.6: General view of the ShaDe WB system folders.

The back-end is divided into different folders corresponding to each of the modules identified in the system's architecture. Then, each of these folders is divided into sub-folders for the algorithms that were integrated and correspond to that module. Each algorithm's folder is composed by an executable file (the algorithm itself), an XML configuration file and any other auxiliary data that might be needed for the algorithm to work (as depicted in Figure 4.6).

As we said before, the core of the application is composed of different modules that have an API which will facilitate the addition of new algorithms. The API is adapted through the XML configuration file that we talked above. This adaptation is made in two phases. In the first, the algorithm introduces itself, through some basic information about itself, like the module it should be integrated, its long name, its short name and some references about it. In the second phase, the XML file configures how it interacts with the API, featuring the executable file name, the parameters necessary for the algorithm to work or the output. In summary, the XML configuration file prepares the API to deal with the executable and results.

The specification of the XML configuration file is as follows (Table 4.1 and Table 4.2) and an example of its use can be seen in Figure 4.7.

The API formalization is not concluded, because it is expected that the API will evolve as more algorithms are added. However, we expect that the API will converge to a stable version soon. For now, some programming effort is needed to adapt, but mostly from our side. Researchers have to make minimal changes to the code of the algorithms.

During the development of our system we had the opportunity to integrate some algorithms. Following the process described above, in an initial phase, we added to the Shape Signature module the Spherical

<**info**> - Relevant information about the algorithm.		
	<**module**> - Module name to which the algorithm will be connected.	
	<**long_name**> - Algorithm's full name.	
	<**short_name**> - Abbreviation of the name of the algorithm.	
	<**reference**> - Reference where the algorithm has been published.	
		<**authors**> - Authors of the article.
		<**title**> - Title of the article.
		<**source**> - Source where it was published the article.

Table 4.1: Specification of the XML configuration file - <info>.

<**config**> - Information on configuring the algorithm.			
	<**exe_name**> - Executable name.		
	<**parameters**> - Information of the parameters for the executable to run.		
		<**parameter**> - Information of the parameter.	
			<**name**> - Parameter name.
			<**type**> - Type of parameter expected.
			<**arg**> - Argument value that is passed along with the executable.
			<**default**> - Default value of the parameter.
			<**max**> - Maximum value that the parameter can have.
			<**min**> - Minimum value that the parameter can have.
			<**option**> - Optional value that the parameter can have.
	<**output**> - Indicates the output of the executable.		
		<**file**> - File that is created.	
			<**type**> - Type of file that is created as output.

Table 4.2: Specification of the XML configuration file - <config>.

```xml
<algorithm>
  <info>
    <module>Segmentation</module>
    <long_name>Tailor</long_name>
    <short_name>Tailor</short_name>
    <references>
      <reference>
        <authors>M. Mortara, G. Patanè, M. Spagnuolo, B. Falcidieno, J. Rossignac</authors>
        <title>Blowing Bubbles for Multi-Scale Analysis and Decomposition of Triangle Meshes.</title>
        <source>Algorithmica 38 (1): 227-248 (2003)</source>
      </reference>
    </references>
  </info>
  <config>
    <exe_name>Tailor_32.exe</exe_name>
    <parameters>
      <parameter name="source_name" type="text" format="off" >
            <option>wrl</option>
            <option>tri</option>
      </parameter>
      <parameter name="scale" type="int" min="1" max="50" />
    </parameters>
    <output>
      <file type="tailor_out.tlr">
    </output>
  </config>
</algorithm>
```

Figure 4.7: XML configuration file of the Tailor algorithm.

Harmonics algorithm [30]. This algorithm, given a model, generates a binary file corresponding to the signature of that model. We also added an auxiliary executable which produces an image file based on a signature file, so it could be possible to display a visual representation of the signature on the front-end.

In a second phase, we integrated the Hierarchical Fitting Primitives algorithm [8] in the Segmentation module. This algorithm produces a segmented model from the original model.

Finally, we fully integrate Marco Attene's shape repair algorithm [7], the Mesh Fix. The Mesh Fix is an automatic algorithm that strives to convert a low-quality digitized polygon mesh into a single manifold and watertight triangle mesh without degenerate or intersecting elements. Also, the Tailor [37] and Mortara's Best View Selection [38] algorithms were integrated into ShaDe WB in the Segmentation and Best View Selection modules, respectively.

Almost all these algorithms which have been integrated are exclusively composed of only a binary

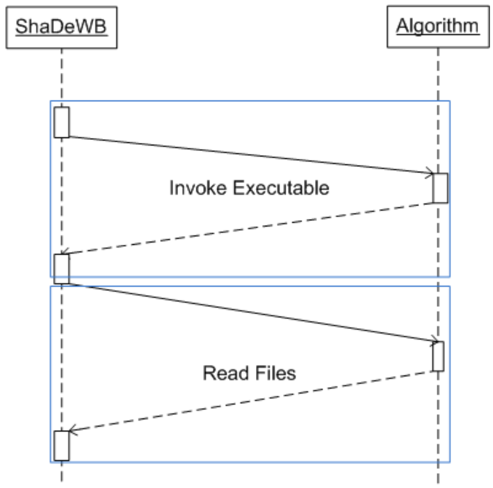

Figure 4.8: Sequence diagram of a generic algorithm's execution.

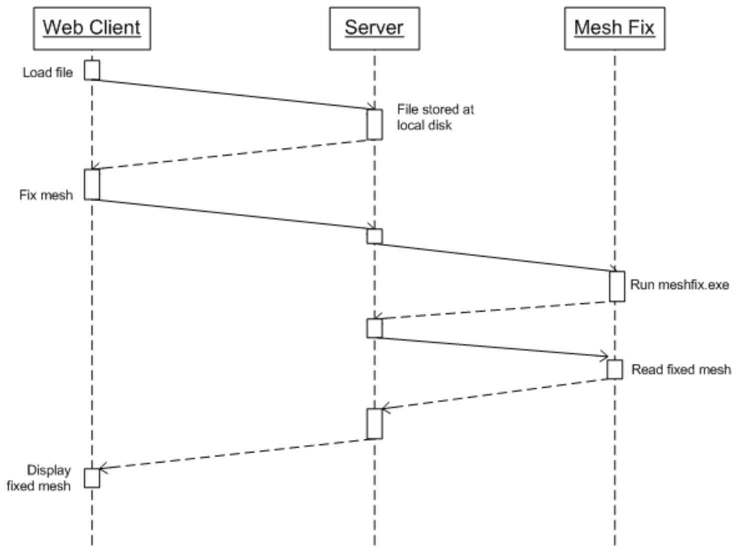

Figure 4.9: Sequence diagram of a simple use case of the algorithm Mesh Fix.

executable file (the Best View and Tailor algorithms need one or two auxiliary files more). Then, the implementation process of the algorithm is similar to all of them. There is a first step in which the executable file is invoked and then the file(s) generated are read, as can be visualized by Figure 4.8 (for a generic case) and Figure 4.9 (for the case of the algorithm Mesh Fix).

4.3.2 Front-End

As referred in Section 4.2., our system is composed by a front-end web, which is the interface where users can analyse models, shape descriptors, etc. It presents different views of the models, graphical representations of the shape signatures or charts illustrating the results. This component was implemented in Flex[3], together with the 3D engine Papervision3D[4].

Similarly to what happened in the previous section with the back end, so now we will show in more detail the operation of front-end.

As can be seen in Figure 4.10, the front-end also consists of different modules. The front-end consists

[3]http://www.adobe.com/products/flex/
[4]http://blog.papervision3d.org/

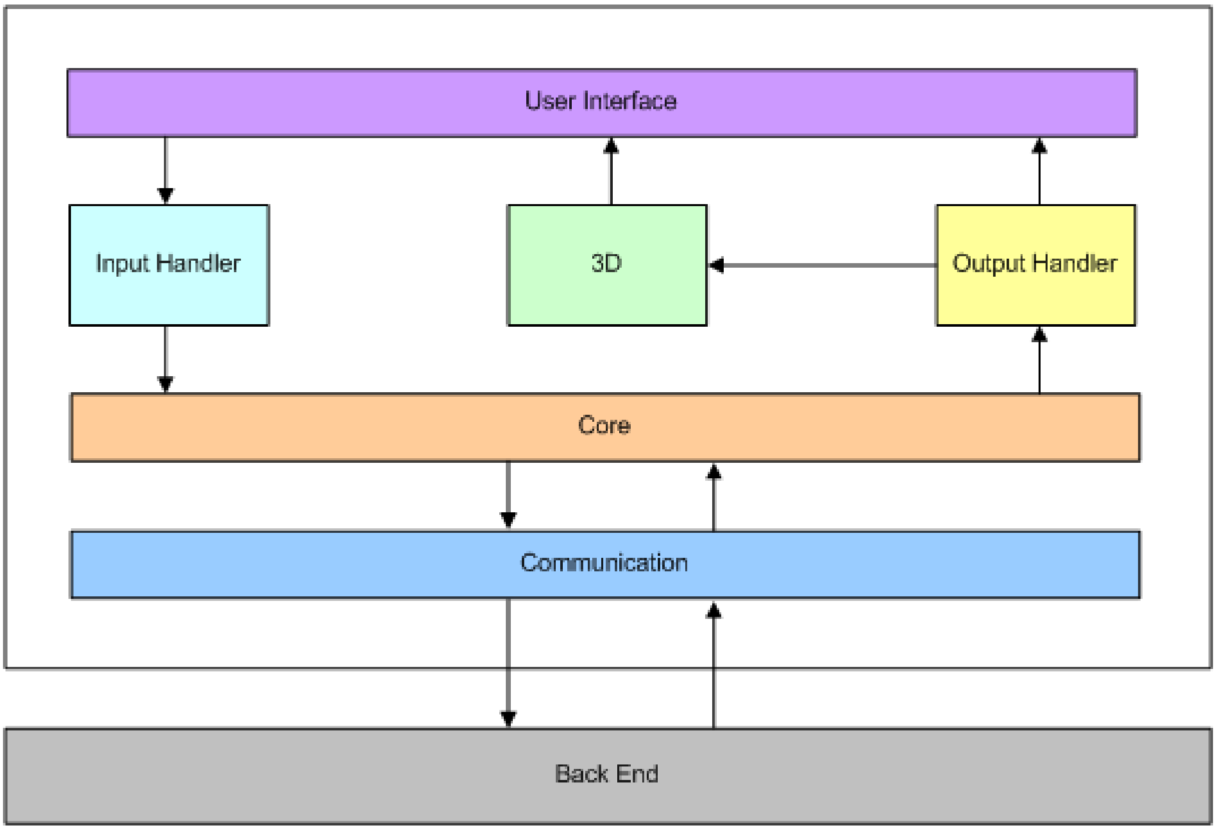

Figure 4.10: Architecture of the Front-End of the system.

of an interface through which the user interacts. When interacting with this interface will be triggered events that will be treated by the Input Handler module. If these events are simple manipulations in the interface, not involving the use of the back-end, this information is treated in the Core and then passed to the Output Handler that will show in the interface, or if necessary, passing through the 3D module before. Otherwise, if it is needed interaction with the back-end, due to be operations with the algorithms integrated, the Core module builds the XML message that is sent via Communication module for the back-end (through XML sockets). When a message is received from the back-end through the Communication module, the message is interpreted in the Core and the process that occurs is the same as previously described.

The design of the user interface was done by dividing the screen into five parts (see Figure 4.11, Figure 4.12, Figure 4.13, Figure 4.14 and Figure 4.15). On top of everything, is located a bar indicating the location of the user in the system. In the top left, is the main view where is shown the 3D model selected. In the upper right, is the secondary view where are displayed the control buttons of the main view or or any ancillary information that is not related to the 3D representation. In the lower right corner, is the menu control of the system and algorithms. Finally, in the lower left corner, is the area for the presentation of information. This area was sub-divided into two parts, one for generic information of the loaded model and other for the information of the algorithm. In the shape information we have the model name, number of triangles, number of vertices, number of segments that the model is divided, the amount of rotation applied to the model in x and y axes, the scaling applied to the model and size ratio between the model and the axes. The algorithm information is composed by the module name where the algorithm has been incorporated, the name of the algorithm itself, the reference of where the article was published on the algorithm and some more information on the algorithm that is requested by the researchers.

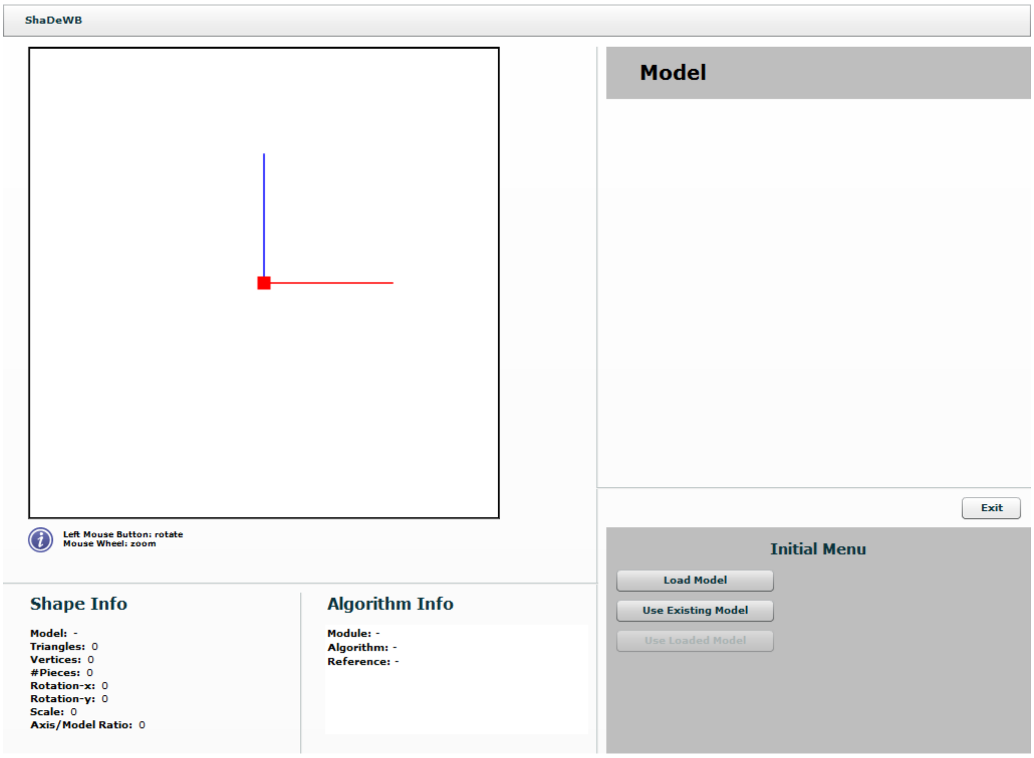

Figure 4.11: Initial screen of the ShaDe WB.

The system interface begins with the initial screen, shown in Figure 4.11. From this screen it is possible to load an external model or use a model of any of the existing collections, which were previously added to our system.

For now, if the user chooses to use one of the existing collections, he just can select the SHREC 2007 Watertight collection. After selecting this collection of objects, it is presented the collection of 3D models, and the user can choose one of these models to load, as is visible in Figure 4.12.

After the user have loaded a model, whether an external model or a model of the existing collections, is shown the main screen of the system (see Figure 4.13). On this screen, the user can now see the three-dimensional model loaded. The user may exercise control over the model, being possible, for example, to change the size of the model, to see the model with or without the wireframe, to change the color of the model, wireframe or background of the display window, between other things. It is also possible to choose a module that contains algorithms (currently, only the modules of Descriptor Calculation, Best View Selection, Segmentation and Shape Repair have algorithms integrated).

The user selecting the button Descriptor Calculation can choose the descriptor that wants to calculate to the model loaded. Currently, only SHA (Spherical Harmonics) was integrated into the Shade WB. As can be seen in Figure 4.14, after the choice of this algorithm, the user can compute the signature of the model. Then is shown the visual representation of this descriptor, and the user can download the binary signature file generated, and the visual representation as well.

If the user chooses the Shape Repair module it can use the algorithm Mesh Fix. By using this algorithm, is presented in the window the 3D object repaired.

By selecting the Segmentation, it is possible to choose two algorithms, HFP (Hierarchical Fitting Primitives) and Tailor. When choosing the algorithm HFP, the user can choose, through a slider, the number of clusters in which he wants to see the object segmented. As a result is displayed in the window the segmented model (see Figure 4.15). If it is chosen the Tailor algorithm, the user also has a slider to choose the scale value to apply. However, despite the request being processed correctly on the back-end's

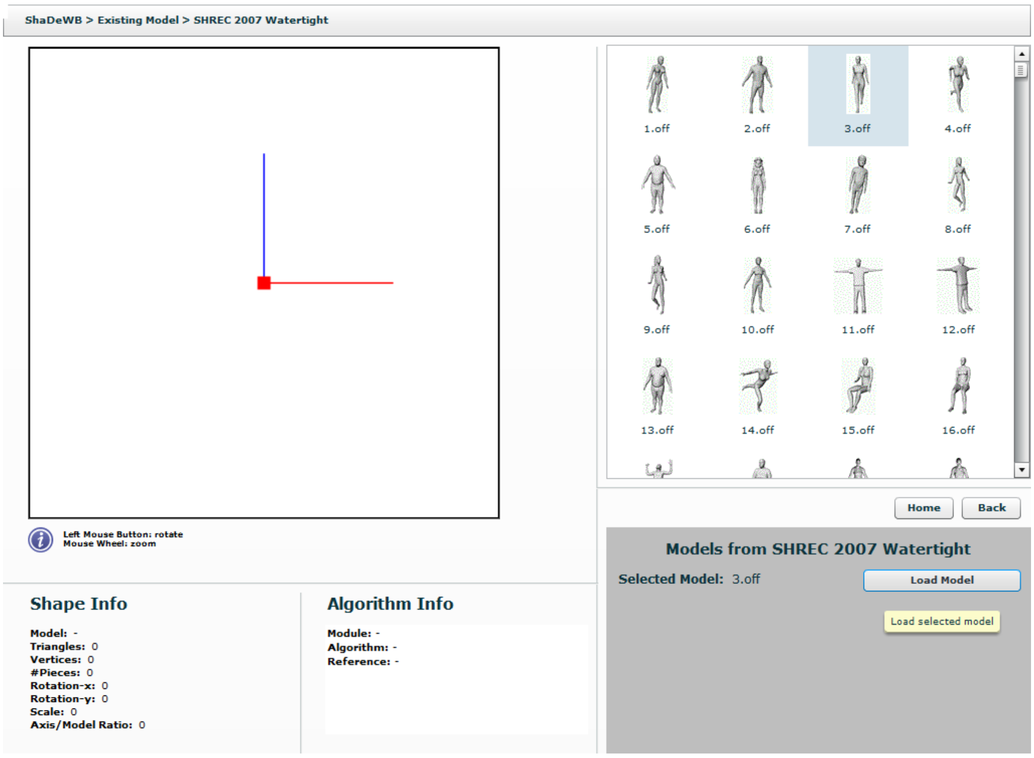

Figure 4.12: Screen displaying the SHREC 2007 Watertight collection.

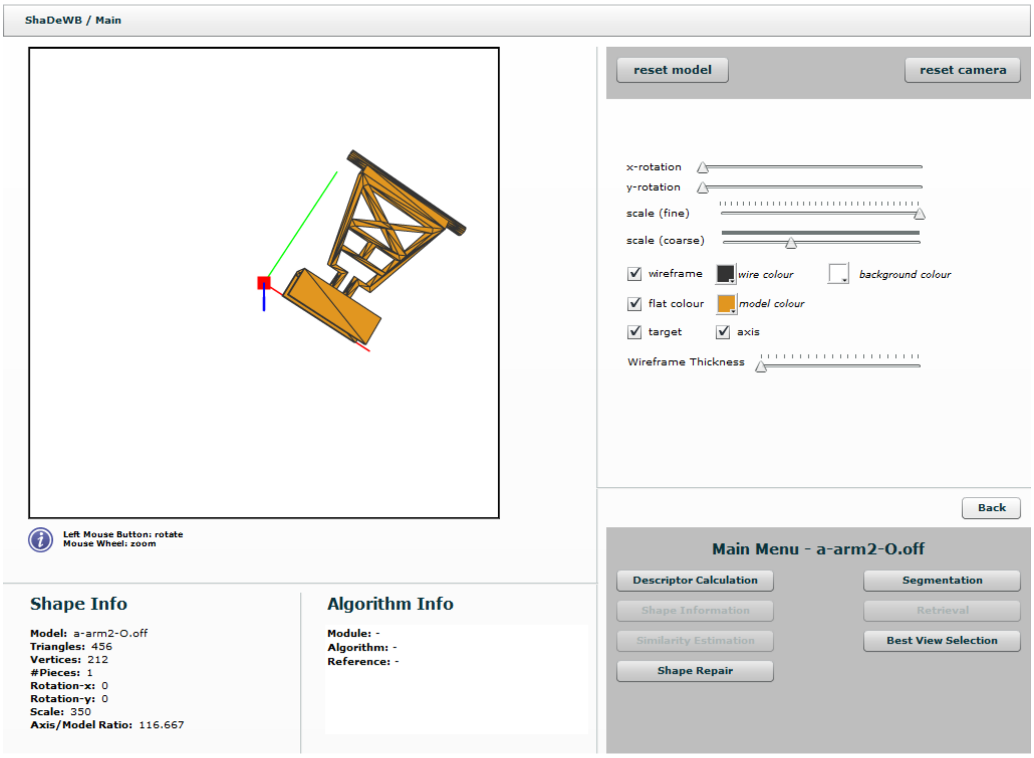

Figure 4.13: Main screen of the ShaDe WB system.

side, the output of the algorithm is not displayed in the system, since the output is not compatible with the system, being necessary to convert the output to a Collada file format.

By following the Best View Selection in the main screen, the user can select the algorithm with which the model was segmented, *i.e.* Fitting Primitives, Plumber, Reeb Graph or Tailor, and calculate the best

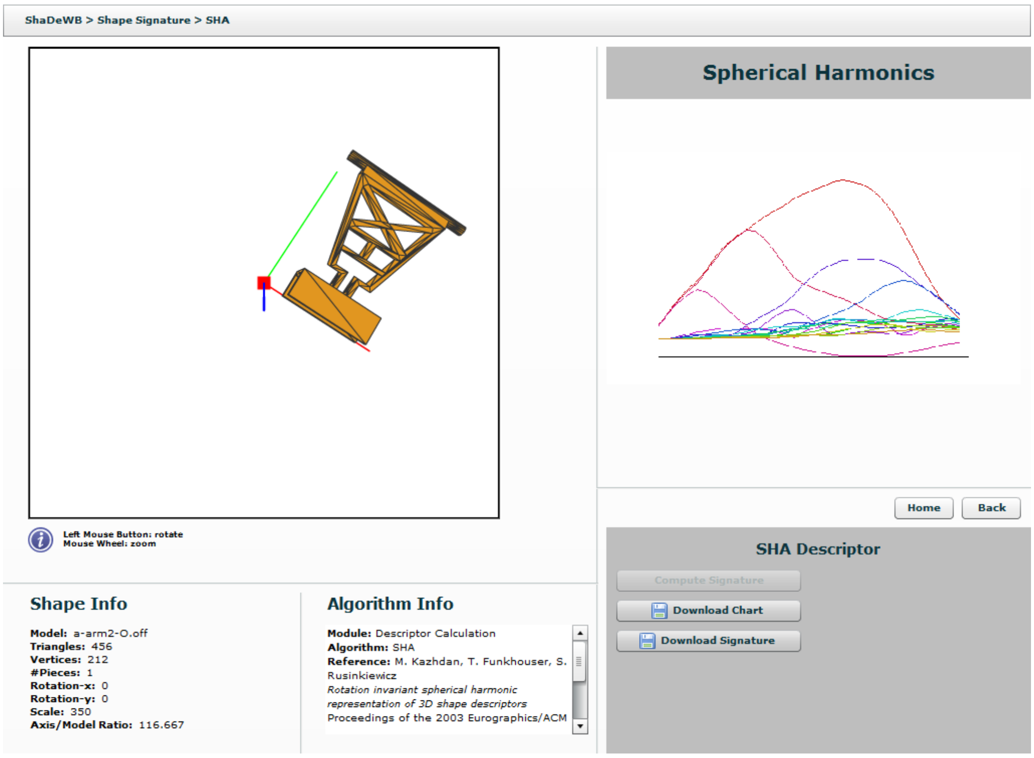

Figure 4.14: Screen of the calculation of the Spherical Harmonics.

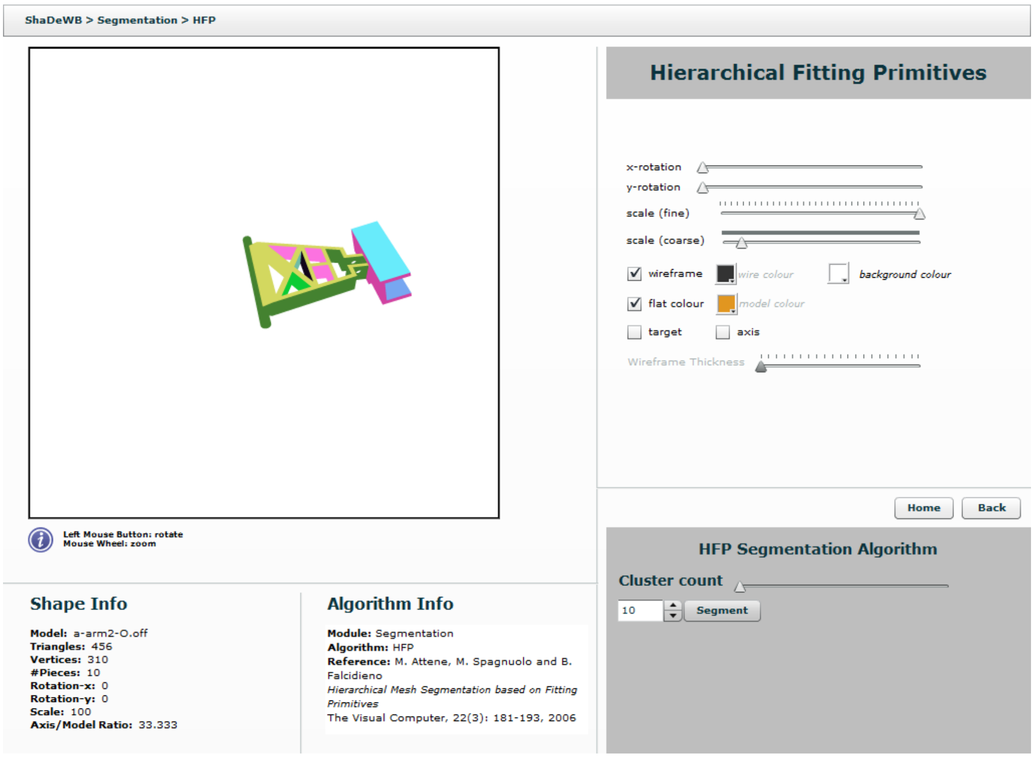

Figure 4.15: Hierarchical Fitting Primitives algorithm screen.

view for this model. Then, the result is displayed in the preview window, *i.e.*, the camera is positioned to present the best view of the object.

4.4 Summary

The framework ShaDe WB: 3D Shape Descriptor Workbench is based on a simple client-server architecture. It is composed by a front-end web, which acts as the user interface, and a back-end site, which acts as the core of the application. The communication between the front-end and the back-end components is done through the exchange of XML messages.

The core of the application is divided into different modules to facilitate the addition of new features. We identified seven main components: one responsible for the shape information, other for the calculation of the shape descriptors, another for the similarity estimation, one for the selection of the best view of a model, other for the retrieval performance analysis, another for the segmentation and, finally, one for there reparation of the shapes. We subdivide each module in sub-modules for each particular feature. This modular and scalable architecture makes easier the addition of features, such as a new shape descriptor. For that, we simply add a new sub-module that implements the shape descriptor to the descriptor calculation module. This type of behaviour is similar in other modules, since each sub-module has an API that provides the interface of the services and facilitates the addition of new sub-modules. The API is adapted through a XML configuration file that each algorithm has to follow.

We have successfully integrated five algorithms in the ShaDe WB, Kazhdan *et al.* Spherical harmonics, Attene's Hierarchical Fitting Primitives and Mesh Fix and Mortara's Tailor and Best View Selection algorithms. We also added SHREC 2007 Watertight collection to ShaDe WB, making possible to browse in this collection and use these models to perform algorithmic analysis.

Given that the front-end is the user interface, through it, users can analyse models, shape descriptors, segment models and fix the mesh of the models. It presents different views of the models, graphical representations of the shape signatures or charts illustrating the results. Athwart it is feasible to perform in the ShaDe WB the actions illustrated in the use case diagram that appear in Figure 4.16. The components in the diagram which are gray and with a dashed link, correspond to some algorithms/modules that are not yet integrated, but that our system is prepared to receive in future.

This system also served as a framework for Raquel Costa, in her thesis work "CaS: Collection-aware Segmentation". The CaS is an algorithm to segment sets of objects based on their individuality. The

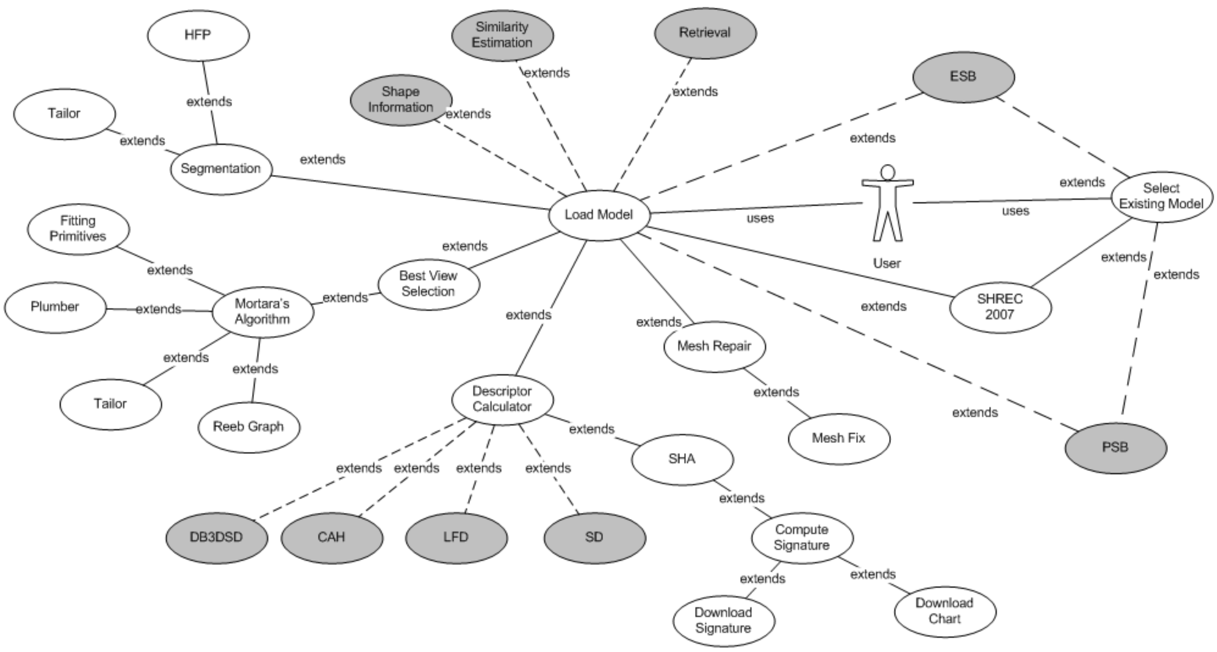

Figure 4.16: Use case diagram of the ShaDe WB.

purpose of her work consisted in the separation phase of the segmentation algorithm that uses the CaS which is integrated into a solution of indexing and retrieval of three-dimensional shapes of objects and validate the applicability of this approach for the segmentation of objects in a collection. We adapted our system and it was possible to perform tests with users in order to validate the applicability of their approach. However, the final version of the ShaDe WB's interface does not include the changes made specifically for this purpose, since it did not fit directly on the objective of our work.

Chapter 5

Evaluation

Our system was designed to meet the needs of users, specifically, researchers working in the area of analysis and retrieval of three-dimensional objects. With this in mind, the most obvious to meet those needs was to perform tests with users, because we can only tell if the system is good only if it is used and pleasing to the user.

We made a two-week long visit to the Shape Modelling Group (SMG) at the Institute of Applied Mathematics and Information Technology (IMATI) of the Italian National Council of Research (CNR) from Genoa, in July 2010. Apart from involving these potential users in the development stage, namely by encouraging them to incorporate their algorithms into the framework, the major goal of this visit was to present the existing system to experienced researchers on 3D analysis and retrieval and obtain valuable feedback.

The evaluation of the ShaDe WB involved the presentation of the system to typical users, the researchers, and to collect their views through questionnaires and informal discussions on two major points: the API and the user interface.

It has also sought the assistance of usability experts to conduct a heuristic evaluation to the user interface of the ShaDe WB, based on Nielsen's heuristics.

5.1 API

One key point that was critical to have some way to evaluate was the API of the system. Given that the API is somehow abstract, it was important to have the opinion of researchers on the system itself and its API. For instance, how easy they find to adapt their algorithms to integrate them in our system.

During our stay with the Shape Modelling Group at the IMATI, we had the opportunity to integrate new algorithms and, at the same time, discuss some interesting points about the system.

We had the chance to do some presentations of our work and gather their advices and thinking about it. In these presentations, they were very open about the system, presenting a few opinions.

They suggested, for the future, the use of databases that contain the three-dimensional objects. Currently, the collections of 3D models are integrated into the system through folders. Each collection has a folder with its name and containing the respective three-dimensional models. They suggested that, in future, this way of storing collections of objects were replaced by the use of databases.

During the algorithms' integration process into our system, we had the opportunity to add a shape repair algorithm developed by Marco Attene [7]. The addition of this new algorithm made us realize the importance of adding a new main component to the application's core, the shape repair module.

We also discussed the XML files for configuring the algorithms that are used in the API. They helped defining some of the parameters used in the configurations files.

Finally, given that we incorporated three algorithms during the stay at CNR IMATI-Ge, we observed the difficulty and the time spent by researchers in the adaptation of algorithms to be added to the system. For the algorithm of mesh repair developed by Marco Attene, the Mesh Fix, it did not require changes, having been integrated exactly as it was.

For Best View Selection and Tailor algorithms developed by Michela Mortara some changes were necessary. These changes were relatively fast, taking about a day or two, but the greater complexity of these changes were due to the conversion of executable algorithms from Linux to Windows. From what we have been transmitted, the corresponding part of the adjustments made specifically to fit our system proved to have small complexity and took little time.

Despite the few algorithms integrated, with the information gathered, it can be concluded that was rather easy and low time consuming for researchers to adapt their algorithms to be integrated in the Shade WB. It was also possible to us realize that, in general, the researchers have shown receptive to the system, partly due to its API. Nevertheless, this evaluation of the API should be targeted for further development as more algorithms are integrated.

5.2 User Interface

In order to test the usability and functionality of a system is necessary to conduct a usability evaluation of it. Therefore, it is possible to use analytical models such as heuristic evaluation, and/or empirical methods, such as the evaluation with users.

In the case of our system, the ShaDe WB, the interface evaluation was performed using both methods. We made a heuristic evaluation with experts in usability, as well as user testing. More specifically, the user evaluation was conducted with the use of questionnaires, through which users gave suggestions of things they would like to see changed in the interface and their general opinion of some points of the system's interface.

5.2.1 Heuristic Evaluation

In order to evaluate the user interface of the ShaDe WB we performed a heuristic evaluation. A heuristic evaluation is a form of usability inspection where usability specialists judge whether each element of a user interface follows a list of established usability heuristics, the Nielsen's heuristics [39].

With this in mind, we asked three usability evaluators to use the system freely and pointing out errors that did not comply with Nielsen's usability heuristics. With this assessment, some usability errors were identified.

One of the main mistakes found consisted of no indication of the loading status or the execution of the tasks on most screens. Another of the more severe errors that was found was, in the HFP segmentation algorithm, the method used to choose the number of clusters was not very user friendly, being difficult to use only the slider to perform this action (see Figure 5.1).

Were also found other errors a little less relevant to the usability of the user, but must be corrected as well, such as, in the main menu is no longer possible to re-use a previous loaded model, the main menu and the buttons "Back" and "Home" are barely visible and there are two sliders to control the same greatness (scale of a model).

Finally, there were other situations, a little more minor, in violation of usability heuristics. From these cases stand out that when is made the reset of a model the applied scale is not the initial one, there are no tooltips on the main menu, when selecting an existing model from the system, the models are not loaded by double-clinking, the slider of wireframe thickness is active even when the wireframe is

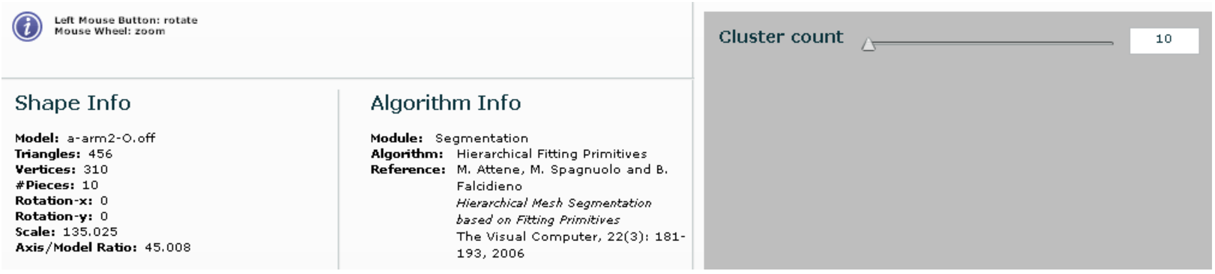

Figure 5.1: Slider to select the number of clusters.

deactivated, the model name on the initial screen is editable implying that it is possible to write the path to the model and the information about the number of clusters appears to be editable despite not being.

For these errors, the usability evaluators also made suggestions for changes to improve the user interface. Next, we present only some of the suggestions for the two most serious mistakes. A more complete version of the usability errors found, their severity, heuristics broken and their corrective actions can be seen at Appendix A.

For the first error identified, the solution is simple and consists of the presentation of messages of "Loadind..." or "Executing..." during the execution of tasks, in order to give the user the information that there is data processing. In the other, should be possible to choose the number of clusters through keyboard and/or using a numeric step.

The correction of the other situations mentioned, in general, are based on small repairs on the user interface, such as the decommissioning or relocation of certain buttons.

This type of assessment allowed us to find some problems with the interface and collecting measures to address them, thereby allowing a better user interface and, in general, making the system more visually appealing.

5.2.2 User Tests

For the evaluation of the interface we also have counted with the participation of users. For this we relied on the valuable contribution of some researchers from the CNR IMATI-Ge.

We had the participation of seven researchers who used the system freely and counted with our help only when they requested it. In the end, they answered a questionnaire that aimed to gauge the state of satisfaction of the current prototype and to improve the system user interface.

This questionnaire was based on the rating of six characteristics of the user interface of the system: spatial organization, color scheme, ease of use, buttons expressiveness, 3D manipulation and general opinion of the system.

Spatial Organization

As seen through Figure 5.2, the system spatial organization can be seen as positive, since five of users qualified it in the positive side of the chart. However we also collected a few comments on this point. Some information exceeds the page and is necessary to scroll the page to see them, being better to fit the most important info and buttons in the first configuration. So a better spatial rearrangement would be opportune.

The "Browse" and "Use existing model" buttons should be one close to the other, in the same portion of the screen. Finally, the information about shapes should be clearly separated by information about tools used. In the current version, on the left is shape representation, on the right is the shape console to navigate and interact with it, on the bottom left is shape info, on the top right is

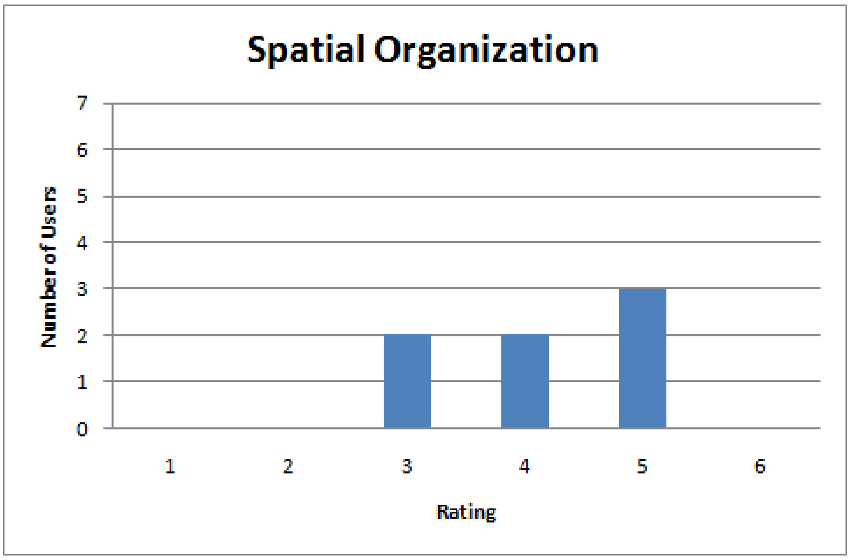

Figure 5.2: Chart illustrating the satisfaction with the spatial organization of the system (1: Bad; 6: Good).

tool title, on the bottom right is tool info. It looked a bit too messy. On the top there should be all the information about shape and on the bottom all the information about the current tool.

Color Scheme

The color scheme is very positive point of the system, since all users rated this aspect in the two highest values, as is visible in Figure 5.3. One consequence of the good color schemes chosen for the system is that there was no comment for amendment of that scheme.

Ease of Use

This feature of the system was difficult to assess because almost all users have selected a different value. However, it is possible to check via the Figure 5.4 that there was a higher incidence (three users) in the average down. Also some suggestions were presented on this topic.

It is advisable to put a few notes on the input/output shape needed for any tools in their short

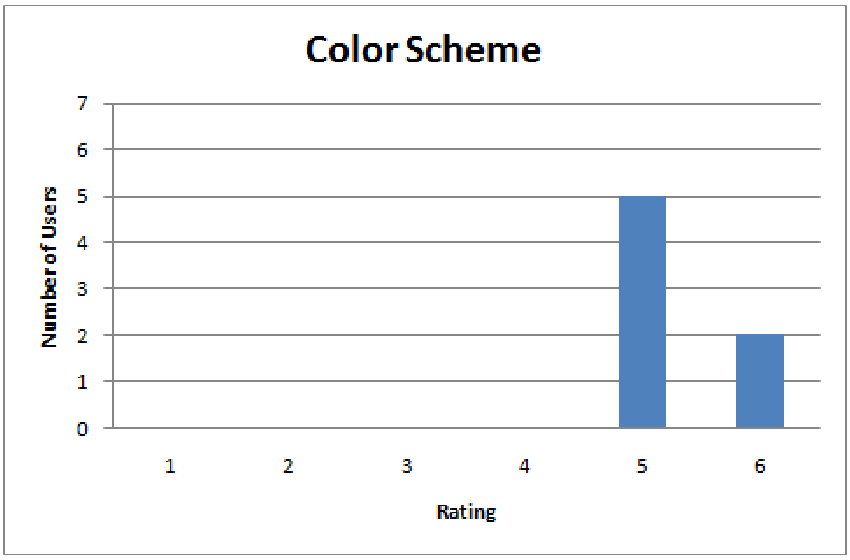

Figure 5.3: Chart illustrating the satisfaction with the system's color scheme (1: Bad; 6: Good).

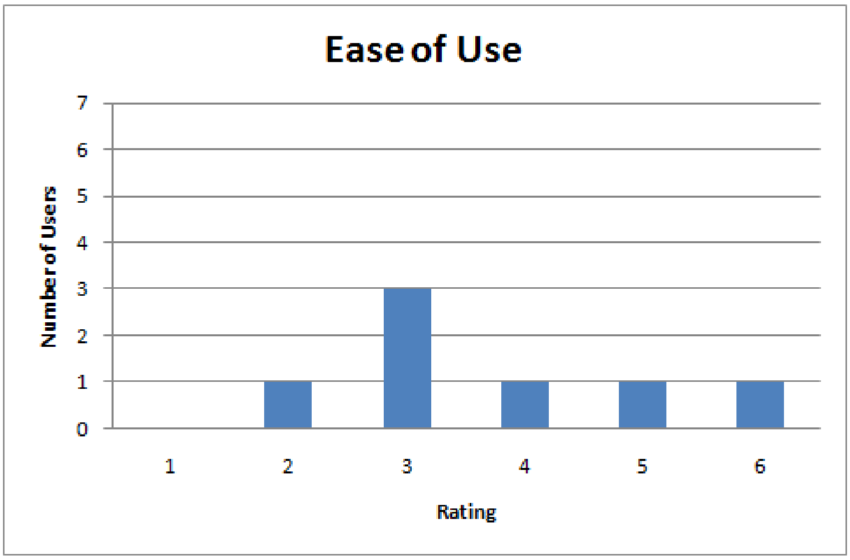

Figure 5.4: Chart illustrating the satisfaction with the ease of use of the system (1: Difficult; 6: Easy).

description. Also, sometimes it is not clear whether the system is processing data or it is stuck. There could be a sort of status message telling "Processing", "Please wait" or "Shape xyz loaded" for example, so that is possible to follow what is going on. Also in the algorithm info, when no algorithm is selected, there should be a message "NO ALGORITHM SELECTED" or something like it.

Only buttons that are compliant with the current shape should be enabled and the other buttons should not be clicked at all. The information about the current tool is displayed only after the tool is selected and the user would like to know something before choosing (a preview, some information), in order to ease its choice.

Two more suggestions were presented. The slider in the HFP segmentation should be coupled also with a numerical choice. It also could be coupled with a <OK> button, so that is possible to move the slider as many times the user wants, set the parameter(s), and only when the user is satisfied with the parameters, click OK and trigger the next event. In the case there is more than one parameter to be set this is very important. Finally, the "Back" button can be wrongly interpreted as "Undo", and sometimes it must be clicked several times. The "Home" button should be always present.

Buttons Expressiveness

As can be seen in Figure 5.5, the expressiveness of the buttons was not consensual. In general, users shared their views in a median rating of this feature and made a few comments about this aspect.

In the Tailor segmentation algorithm, the names of the algorithms should be changed by their full names instead of being with abbreviations. For algorithms which require an input parameter (such as umber of segments and suchlike), a "Do It" button should be, rather than executing the algorithm when the user releases the cursor on the slider. It should also be explained somewhere that leaving the pointer on a widget pops help messages.

The "Back" button can be misleading; the "Home" recalls the OpenInventor[1] viewer *home* functionality which puts the object in the default position inside the canvas. Also the *reset* names are

[1]http://oss.sgi.com/projects/inventor/

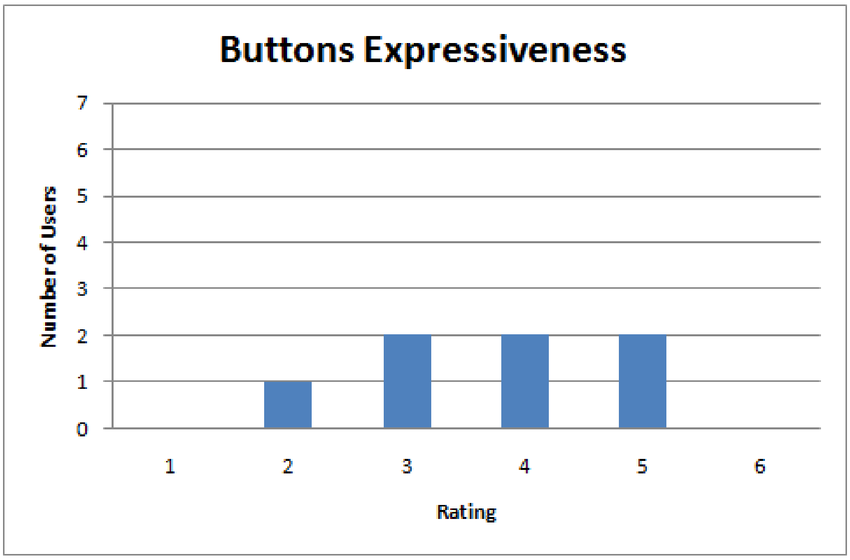

Figure 5.5: Chart illustrating the satisfaction with the buttons expressiveness of the system (1: Bad; 6: Good).

not completely friendly, since they recall a kind of delete function. Finally, each button should be endowed with a short explanation of the corresponding functionality, for example a tooltip.

3D Manipulation

The users of the system at this point are also divided in their opinions. As is depicted in Figure 5.6, three users have given a negative evaluation, and four gave a positive evaluation, finding that manipulation of 3D objects was the normal and traditional. These negative views were based on the difficulty of rotation, in the absence of panning and application of scale in the object is not very intuitive, as shown by the opinion they expressed.

The users who gave a poor rating felt that the scaling was ok with the mouse but, at the beginning, it was unclear and counterintuitive how to use the “fine” and “coarse” scales. These scales were just misleading.

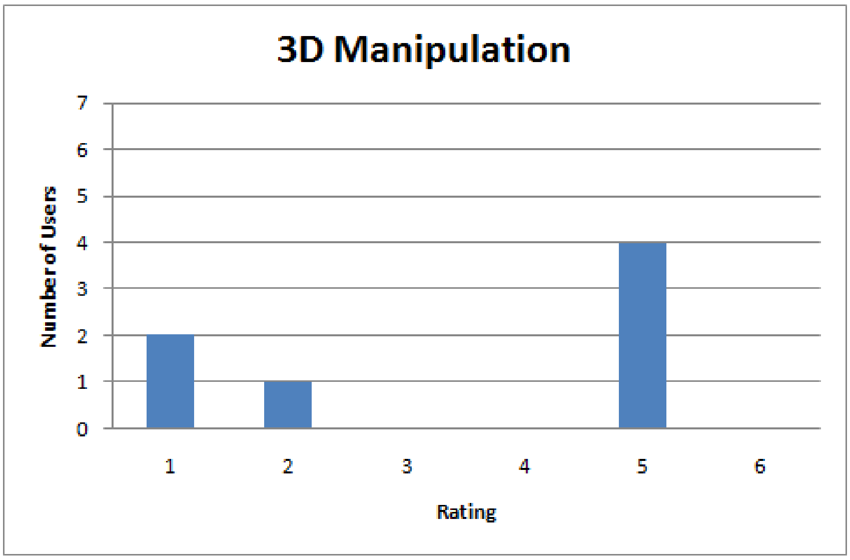

Figure 5.6: Chart illustrating the satisfaction with the system’s 3D manipulation (1: Difficult; 6: Easy).

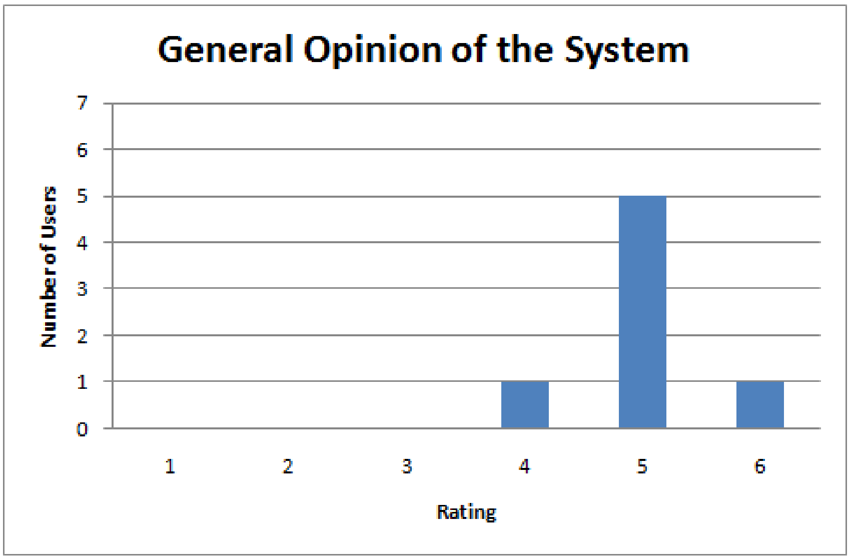

Figure 5.7: Chart illustrating the general opinion of the system (1: Bad; 6: Good).

Also, that it was very difficult to interact with the object. There is no panning option (translating the object), therefore rotation is only possible along the original axes. The rotation around a corner seemed odd and maybe translating the shape in the origin (so that the barycentre coincides with the origin) would improve the feeling.

General Opinion of the System

Despite differing opinions on the characteristics of the system outlined above, in general, the opinion about the ShaDe WB was good, and all users have given an overall positive view of the system, as it can be seen in Figure 5.7. There were also requested additional comments that had not fit in the previous categories, which we briefly summarize.

To improve the system, was suggested the addition of some introductory notes, a sort of short tutorial, to help the new user to use more easily the interface, *e.g.*, which are the constrains about the input files, what is the meaning of the input parameters, etc (this information should be provided by the researchers). Also, to be consistent with name chosen for buttons and properties: for example, the segments of a segmentation are named "pieces" instead of "segments"; there is a shape info coming directly from the basic tool to build the UI (giving the position of the model in the reference system) which corresponds to the button "Shape Information", which will collect different tools to calculate some more advanced properties of the shape.

The users would like to have the possibility to download the output of the segmentation as it is possible to download the output of the shape descriptors and also the opportunity of loading with the Best View the output provided by Tailor. About the visualization of spherical harmonics, as they depend on two parameters, namely frequency and sphere radius, would it be better to visualize them with a bi-dimensional plot (a 2-dim histogram, a surface, ...), rather than with a one-dimensional plot of superimposing graphs.

As it was possible to conclude, in particular by the last graph, the general opinion of the system by the actual users, the researchers in the analysis, comparison and retrieval of three-dimensional objects domain, was positive. Although usability has to be stressed, they thought it was a nice platform, with a core target that looks interesting and well developed.

The users allowed the identification of several errors or situations that should be targeted for improvement, pointing suggestions for changes. This evaluation allowed a glimpse of the work that needs to be done in the future so that the user interface will respond better to what the researchers want.

Still on the user interface, it was not possible to perform any more type of evaluation with users (*e.g.*, measure some usability metrics like, time or number of errors to complete a task) since the time we had with the researchers of the Shape Modelling group from the CNR IMATI-Ge was short. The completion of this evaluation with other users who were not within the context of analysis, comparison and retrieval of three-dimensional objects did not seem relevant, since they would not know what was being treated with the use of our system.

5.3 Summary

To assess the quality of our system we focused ourselves on two points, the API and the user interface. For this, we relied on the cooperation of researchers from the Shape Modelling group of the CNR IMATI-Ge. The evaluation of the interface also included the use of usability experts.

Regarding the evaluation of the API, it was possible to gather the views of researchers about it. We also found that the adaptation of the algorithms by researchers in order to be adapted to our system was quite simple and took little time. They also helped to define some parameters used in the XML files for configuring the algorithms that are used in the API.

In assessing the user interface, both through the evaluation based on Nielsen's usability heuristics as well as through the assessment made by users, it was possible to identify some situations that could be improved. From these situations highlights the little feedback that is given to the user about the state system, whether on the implementation of tasks as well as some additional information that should be provided about the existing algorithms.

We ask users to respond to a questionnaire based on six characteristics of the user interface of the system: spatial organization, color scheme, ease of use, buttons expressiveness, 3D manipulation and general opinion of the system. Despite a certain dispersion by users in evaluating these characteristics, the general opinion was very positive.

In a more global view of the system, although the interface needs to be improved, the researchers thought it was a good framework with a core target that looks interesting and well developed. To this opinion contributed greatly the API. They have also demonstrated their availability to contribute with more algorithms to be integrated into the Shade WB.

In conclusion, given that this work will be further developed in the future under the project 3DROuS, funded by Foundation for Science and Technology (ref. PTDC/EIA-EIA/102930/2008), this evaluation allowed the identification of some situations that should be improved, particularly in the user interface. However, it will be necessary to perform more evaluations in the future, since there were few users who have had the opportunity to check the functioning of the Shade WB. Thus, the system should be targeted for further development as more algorithms are integrated.

Chapter 6

Conclusions and Future Work

In this chapter we present our conclusions about the work developed within this dissertation and point out future work to perform, since this system will continue to be developed beyond the scope of this thesis.

6.1 Conclusions

There was an increasing interest in methods of analysis, comparison and retrieval of three-dimensional models, resulting from the spreading of research on several fields related to 3D objects. Many techniques have been developed for information retrieval, but one problem is that most approaches are very difficult to compare with each other due to the use of different databases, performance measures and methodologies. So, there is not a single and simple standard system that enables researchers working in 3D object analysis domain a central study of the shape descriptors. We intended to develop a benchmark solution, that was web-based, modular, scalable, straightforward and will allow the study and comparison of object description techniques with minor effort.

We have seen that shape descriptors are a set of values extracted from the 3D models, which describe the models in a high-dimensional space. Then, we described some of the most commonly used similarity measures used to calculate the distance between the objects representations in the multidimensional space and that measure the similarity between the objects. Following that, we introduced some approaches for the segmentation of three-dimensional models. We have presented some algorithms for selecting the best view of an object and for the repair of defected meshes. We also brought forward and explained some statistics for measuring the performance of a shape descriptor in a content-based retrieval system.

We have studied all these topics, because these topics are fundamental to the comprehension of a shape analysis solution. We highlighted some of the most relevant multimedia information retrieval benchmarks, because the initial scope of our work was to analyze the descriptors used in retrieval. In particular, given the nature of our work, we gave greater emphasis to the 3D object related, presenting and comparing the most well known 3D shape benchmark solutions, like the Princeton Shape Benchmark, the AIM@SHAPE Shape Repository, Engineering Shape Benchmark, the SHREC - Shape Retrieval Contest and the MyMultimediaWorld.com.

However, during development work, we identified a gap in the area that went far beyond the retrieval benchmarking. There are no tools that allow comparison across most of the topics studied in the area of 3D object retrieval. Thus, our work evolved from a workbench to descriptors to a framework for shape analysis, the ShaDe Workbench. This framework is based in a client-server architecture that works over the internet. The system is composed by a front-end web that behaves as the user interface and gives the users the possibility to test and compare different 3D shape techniques. It also consist of a back-end site which has the core of the application and that is developed in a modular and scalable way to allow the addition of new shape descriptors, segmentation algorithms, etc.

The back-end is composed by seven modules (Descriptor Calculation, Shape Information, Best View Selection, Segmentation, Mesh Repair, Similarity Estimation and Retrieval) and each of which has an API that allows the addition of the algorithms to the system. This API is based on an XML configuration file that is sent together with the algorithm to add. So far, we have integrated five algorithms in the ShaDe WB, Kazhdan *et al.* Spherical harmonics, Attene's Hierarchical Fitting Primitives and Mesh Fix and Mortara's Tailor and Best View Selection algorithms. We also added SHREC 2007 Watertight collection to ShaDe WB, making possible to browse in this collection and use these models to perform algorithmic analysis.

We involved real users to evaluate our approach. Since the proposed solution will be used in a very specific field, we performed tests with researchers working on 3D model analysis, comparison and retrieval methods. These tests were made to evaluate the interface and the API and were done using a combination of techniques like questionnaires and informal interviews. We also involved three usability experts to evaluate the user interface based on Nielsen's heuristics. After several evaluations, there were several points to improve in the system's user interface, but overall, the feedback from users was positive about the Shade WB.

This work also served as a framework for Raquel Costa's thesis work, "CaS: Segmentation Collection-aware" (which has no published articles yet). Through the adaptation of our system specifically for this purpose, it was possible to conduct user testing to validate the applicability of their algorithm.

6.2 Future Work

This work is integrated in the project 3DORuS and as such, is not terminated with the completion of this thesis. Thus, there is still much work to be done hereafter. The future work to accomplish can be divided into short and long term tasks.

As short-term tasks, we have the integration of the CaS algorithm into the segmentation module of the ShaDe WB and the development of a converter for the output produced by the Tailor algorithm to a Collada format, so it is possible to display the result in the interface. We also must improve the user interface according to the heuristic and user evaluations which were conducted, put the system online permanently, and we want to be able to add algorithms' executables that run on Linux, similarly to what already happens to those developed on Windows.

For long-term tasks, we intend to adopt further contacts with other researchers in this area, trying to awaken their interest in our solution, thus providing their algorithms to be integrated in the ShaDe WB. Also related to this task should come up a further development of the API of the system as more algorithms are added.

We believe that our work will be an important contribution for the 3D object analysis, comparison and retrieval researchers. We expect that our solution will be widely used by this community.

Bibliography

[1] H. Abdi. Distance. In *Encyclopedia of Measurement and Statistics*, volume 2. SAGE Publications, Inc, Thousand Oaks, California, 2007.

[2] E. Agathos, I. Pratikakis, S. Perantonis, N. Sapidis, and P. Azariadis. 3d mesh segmentation methodologies for cad applications. 2008.

[3] C. B. Akgül. *3d Shape Descriptors and Similarity Learning.* PhD thesis, Boḡaziçi University, June 2006.

[4] C. B. Akgul, B. Sankur, F. Schmitt, and Y. Yemez. Multivariate density-based 3d shape descriptors. In *SMI '07: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2007*, pages 3–12, Washington, DC, USA, 2007. IEEE Computer Society.

[5] P. A. d. Alarcón, A. D. Pascual-Montano, and J. M. Carazo. Spin images and neural networks for efficient content-based retrieval in 3d object databases. In *CIVR '02: Proceedings of the International Conference on Image and Video Retrieval*, pages 225–234, London, UK, 2002. Springer-Verlag.

[6] J. Assfalg, A. Del Bimbo, and P. Pala. Retrieval of 3d objects by visual similarity. In *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 77–83, New York, NY, USA, 2004. ACM.

[7] M. Attene. A lightweight approach to repairing digitized polygon meshes. *The Visual Computer*, 2010.

[8] M. Attene, B. Falcidieno, and M. Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *Vis. Comput.*, 22(3):181–193, 2006.

[9] M. Attene, M. Mortara, M. Spagnuolo, and B. Falcidieno. Hierarchical convex approximation of 3d shapes for fast region selection. In *SGP '08: Proceedings of the Symposium on Geometry Processing*, pages 1323–1332, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.

[10] S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno. Size functions for 3d shape retrieval. In *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 239–242, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[11] V. Blanz, M. J. Tarr, and H. H. Blthoff. What object attributes determine canonical views? *Perception*, 28(5):575–600, 1999.

[12] B. L. Bonhomme, B. Mustafa, S. Celakovsky, M. Preda, F. J. Prêteux, and D. Davcev. On-line and open platform for 3d object retrieval. In *3DOR*, pages 73–79, 2008.

[13] B. Bustos, D. A. Keim, D. Saupe, T. Schreck, and D. V. Vranić. Feature-based similarity search in 3d object databases. *ACM Comput. Surv.*, 37(4):345–387, 2005.

[14] R. J. Campbell and P. J. Flynn. A survey of free-form object representation and recognition techniques. *Comput. Vis. Image Underst.*, 81(2):166–210, 2001.

[15] V. Castelli. Multidimensional indexing structures for content-based retrieval. Technical report, IBM, 2001.

[16] Y. Y. Ceyhun Burak Akgül, Bülent Sankur and F. Schmitt. A framework for histogram-induced 3d descriptors. In *Proceedings of the 14th. European Signal Processing Conference (EUSIPCO06)*, Florence, Italy, September 2006.

[17] Y. Y. Ceyhun Burak Akgül, Bülent Sankur and F. Schmitt. Density-based 3d shape descriptors. *EURASIP Journal on Advances in Signal Processing*, 2007(4), 2007.

[18] D.-Y. Chen, X.-P. T. Tian, and Y.-T. Shen. On visual similarity based 3d model retrieval. *Computer Graphics Forum*, 22(3):223–232, 2003.

[19] M. d'Amico, P. Frosini, and C. Landi. Using matching distance in size theory: A survey. 16(5):154–161, 2006.

[20] C. Faloutsos. *Searching Multimedia Databases by Content.* Kluwer Academic Publishers, Norwell, MA, USA, 1996.

[21] R. Fang, A. Godil, X. Li, and A. Wagan. A new shape benchmark for 3d object retrieval. In *ISVC '08: Proceedings of the 4th International Symposium on Advances in Visual Computing*, pages 381–392, Berlin, Heidelberg, 2008. Springer-Verlag.

[22] M. J. Fonseca. *Sketch-Based Retrieval in Large Sets of Drawings.* PhD thesis, Instituto Superior Técnico, July 2004.

[23] T. Funkhouser, M. Kazhdan, P. Min, and P. Shilane. Shape-based retrieval and analysis of 3d models. *Commun. ACM*, 48(6):58–64, 2005.

[24] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A search engine for 3d models. *ACM Trans. Graph.*, 22(1):83–105, 2003.

[25] A. G. Hauptmann and M. G. Christel. Successful approaches in the trec video retrieval evaluations. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 668–675, New York, NY, USA, 2004. ACM.

[26] N. Iyer, S. Jayanti, and K. Ramani. An engineering shape benchmark for 3d models. In *Proceedings of ASME IDETC/CIE*, pages 24–28, 2005.

[27] A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching.* PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.

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

[29] G. Kazai, M. Lalmas, and A. P. de Vries. The overlap problem in content-oriented xml retrieval evaluation. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 72–79, New York, NY, USA, 2004. ACM.

[30] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 156–164, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[31] H. Laga. Semantics-driven approach for automatic selection of best views of 3d shapes. In *3DOR2010: Proceedings of the 3rd Eurographics Workshop on 3D Object Retrieval.* Eurographics Association, 2010.

[32] B. le Bonhomme, M. Preda, and F. Preteux. Mymultimediaworld.com: A benchmark platform for 3d compression algorithms. pages 2700–2703, 2008.

[33] Y. Liu, H. Zha, and H. Qin. The generalized shape distributions for shape matching and analysis. In *SMI '06: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006*, page 16, Washington, DC, USA, 2006. IEEE Computer Society.

[34] P. Mahalanobis. On the generalized distance in statistics. *Natl. Inst. Science*, 12:49–55, 1936.

[35] C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA, 2008.

[36] M. Mortara and G. Patanè. Shape-covering for skeleton extraction. *International Journal of Shape Modeling*, 8(2):139–158, 2002.

[37] M. Mortara, G. Patanè, M. Spagnuolo, B. Falcidieno, and J. Rossignac. Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. *Algorithmica*, 38(1):227–248, 2003.

[38] M. Mortara and M. Spagnuolo. Semantics-driven best view of 3d shapes. *Comput. Graph.*, 33(3):280–290, 2009. IEEE International Conference on Shape Modelling and Applications 2009.

[39] J. Nielsen. *Heuristic evaluation*, pages 25–62. John Wiley & Sons, Inc., New York, NY, USA, 1994.

[40] M. Novotni and R. Klein. 3d zernike descriptors for content based shape retrieval. In *SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications*, pages 216–225, New York, NY, USA, 2003. ACM.

[41] P. Papadakis, I. Pratikakis, S. Perantonis, and T. Theoharis. Efficient 3d shape matching and retrieval using a concrete radialized spherical projection representation. *Pattern Recogn.*, 40(9):2437–2452, 2007.

[42] E. Paquet. Description of shape information for 2-d and 3-d objects. *Sig. Proc.: Image Comm.*, 16:103–122, September 2000.

[43] E. Paquet and M. Rioux. Nefertiti: a query by content software for three-dimensional models databases management. In *NRC '97: Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, page 345, Washington, DC, USA, 1997. IEEE Computer Society.

[44] G. Reeb. Sur les points singuliers d'une forme de Pfaff complétement intégrable ou d'une fonction numérique. *Comptes Rendus de L'Académie ses Sciences*, pages 847–849, 1946.

[45] W. C. Regli, C. Foster, E. Hayes, C. Y. Ip, D. Mcwherter, M. Peabody, Y. Shapirsteyn, and V. Zaychik. National design repository project: A status report, 2001.

[46] A. Shamir. A survey on mesh segmentation techniques. *Computer Graphics Forum*, 2008.

[47] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. In *SMI '04: Proceedings of the Shape Modeling International 2004*, pages 167–178, Washington, DC, USA, 2004. IEEE Computer Society.

[48] A. Shokoufandeh, S. J. Dickinson, K. Siddiqi, and S. W. Zucker. Indexing using a spectral encoding of topological structure. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:2491, 1999.

[49] A. F. Smeaton, W. Kraaij, and P. Over. The trec video retrieval evaluation (trecvid): A case study and status report, 2003.

[50] C. J. van Rijsbergen. *Information Retrieval.* Butterworths, London, 1975.

Appendix A

Heuristic Evaluation

In this appendix we present a consolidated report of the heuristic evaluation performed by the three usability experts to the user interface of the ShaDe WB, thus having a more detailed view of it. Therefore, in this report, we produce a more complete version of the usability errors found, their severity, heuristics broken and their corrective actions. The consolidated report of the heuristic evaluation is the following:

1. **No indication of the loading status or the execution of the tasks.**
   - **H2-1:** Visibility of system status
   - **Description:** Except in the initial screen, there is no indication of the loading status or the execution of the tasks on most of the screens
   - **Correction:** Presentation of messages of "Loading ..." or "Executing ..." during the execution of tasks, in order to give the user the information that there is data processing
   - **Severity:** 3
2. **Inability to re-use a template loaded.**
   - **H2-7:** Flexibility and efficiency of use
   - **Description:** In the start menu, if the user has mistaken, is no longer possible to re-use the model that had been loaded
   - **Correction:** Place a button on the initial menu that allows the use of the currently loaded model
   - **Severity:** 2
3. **The main menu and the buttons Back/Home are barely visible.**
   - **H2-4:** Consistency and standards
H2-5: Error prevention
H2-7: Flexibility and efficiency of use
H2-8: Aesthetic and minimalist design
   - **Description:** The main menu and the buttons Back/Home are barely visible in the screen of the application
   - **Correction:** Place the "Back" and "Home" buttons so that they are always visible in the application.
   - **Severity:** 2
4. **Unitary scale applied when resetting a model.**
   - **H2-5:** Error prevention
H2-7: Flexibility and efficiency of use
H2-9: Help users recognize, diagnose, and recover from errors
   - **Description:** In the reset of the model, the applied scale should not be one
   - **Correction:** When resetting a model, application of a scale equal to that applied when the model is originally loaded

- **Severity:** 2

5. **No tooltips on the main menu.**
   - **H2-10:** Help and documentation
   - **Description:** There are no tooltips on the main menu buttons
   - **Correction:** Implement tooltips on the main menu, so that the buttons operations are explained
   - **Severity:** 1

6. **Double-click a model should load it.**
   - **H2-7:** Flexibility and efficiency of use
   - **Description:** When selecting an existing model from the system, the models are not loaded by double-clicking
   - **Correction:** When selecting an existing model from the collections of the system, it should be possible to load models by double-clicking
   - **Severity:** 1

7. **Difficulty choosing the number of clusters.**
   - **H2-3:** User control and freedom
     H2-5: Error prevention
     H2-7: Flexibility and efficiency of use
   - **Description:** In the HFP segmentation algorithm, is difficult to choose the number of clusters through the slider
   - **Correction:** In the HFP segmentation algorithm, it should also be possible to choose the number of clusters via the keyboard. The use of a numeric step also can be handy
   - **Severity:** 3

8. **Two sliders to control the scale.**
   - **H2-5:** Error prevention
   - **Description:** There are two slider to control the same greatness (scale of a model)
   - **Correction:** Adjust the sliders so that one has more granularity than the other
   - **Severity:** 1

9. **Slider of wireframe thickness active even when wireframe is deactivated.**
   - **H2-5:** Error prevention
   - **Description:** The wireframe thickness slider is active even when the wireframe button is deactivated
   - **Correction:** Disable the thickness wireframe slider when the model wireframe is also disabled
   - **Severity:** 1

10. **Model name editable.**
    - **H2-5:** Error prevention
    - **Description:** Model name is editable on the initial screen, implying that it is possible to write the path to the model
    - **Correction:** The model name should not be editable
    - **Severity:** 1