INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Im-ORet : Immersive 3D Object Retriever

Pedro Miguel Bacalhau Pascoal

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática e de Computadores

Júri

Presidente: Professor Doutor João Emilio Segurado Pavão Martins
Orientador: Professor Doutor Alfredo Manuel dos Santos Ferreira Junior
Vogal: Professora Doutora Maria Teresa Caeiro Chambel

Outubro 2011

Agradecimentos

Aos meus pais Mariana Pascoal e José Pascoal, que foram responsáveis pela formação do meu caractér e personalidade;

À minha irmã, Joana Pascoal, pelo carinho, paciência e incentivo;

À minha avó, Natividade Noronha, pelo apoio moral;

Aos meus colegas, pelo apoio e colaboração;

Aos meus amigos que ajudaram e participaram deste trabalho;

A todos os utilizadores que se disponibilizaram participar na avaliação deste trabalho, o meu reconhecimento e gratidão;

À Fundação Portuguesa para Ciência e Tecnologia (FCT), que financiou este trabalho através do projecto 3DORuS, referência PTDC/EIA-EIA/102930/2008;

Ao Bruno Araújo pelo apoio e assistência, no uso da *framework* OpenIVI; Assim como, na sua disponibilidade e ajuda na integração do protótipo com o equipamento do Laboratório *João Lourenço Fernandes*, do Taguspark;

De modo muito particular, ao Prof. Doutor Alfredo Ferreira, por toda a dedicação, disponibilidade e orientação que me concedeu ao longo do trabalho.

Lisboa, Outubro 2011

Pedro Pascoal

Resumo

O rápido aumento de modelos 3D agravou o problema de procura em grandes coleções de modelos. Apesar do grande numero de ferramentas de pesquisa de objetos 3D, estas ferramentas não tiram partido das novas tecnologias. Os resultados das pesquisas são amostrados por vistas 2D, o que dificulta a sua interpretação, com pouca ou nenhuma manipulação possível. Com a disseminação da visão estereoscópica, e de dispositivos de interação de última geração fora do ambiente de laboratório, é possível serem apresentados resultados mais ricos, em vez de uma lista de vistas 2D.

Esta tese descreve uma nova abordagem para visualização de resultados de uma pesquisa de modelos 3D, que tira partido das vantagens de ambientes de realidade virtual. Desenvolvemos um sistema onde os utilizadores podem navegar pelos resultados de uma pesquisa num ambiente imersivo de realidade virtual, tirando partido de uma interface post-WIMP. O utilizador pode explorar os resultados, navegar no espaço 3D, e manipular os objetos nele distribuídos. O uso conjunto de ambiente virtual e dispositivos com seis graus de liberdade, oferecem um visualização dos modelos mais completa e uma interação mais natural, com manipulação direta. Para validar o nosso prototipo foram realizados testes com utilizador.

A finalidade e os objetivos desta dissertação foram atingidos, revelando-se um trabalho muito promissor. Com a validação da nossa abordagem abrimos caminho para a investigação, e resolução de novos desafios no contexto de recuperação de objetos 3D em ambientes imersivo.

Abstract

The rapid growth of 3D models has urged the problem of searching in large collections of models. Although there are a large number of tools for 3D object retrieval, these tools do not take advantage of recent technologies. The results are presented as thumbnails, which hinders the interpretation of results, and allow little or no manipulation at all. With the spreading of stereoscopic viewing and last generation interaction devices outside lab environment, richer results could be presented instead of a list of thumbnails.

This thesis describes a new approach for the visualization of query results for 3D object retrieval, which takes advantage of virtual reality immersive environment. We devised a system where users can navigate through the results of an query on an immersive virtual reality environment by taking advantage of a post-WIMP interface. The user can explore the results, navigate in the three-dimensional space and manipulate the scattered objects. The combined use of virtual environments and six degree-of-freedom devices, provides a more complete visualization of models and makes interaction more natural, with direct manipulation. To validate our prototype we performed a evaluation with users.

The goal and objectives of this thesis were achieved, proving it to be a very promising work. With the validation of our approach we opened the way to research and solve new challenges in the context of 3DOR using immersive environments.

Palavras-Chave

Keywords

Palavras-Chave

Im-O-Ret
Objectos Tridimensionais
Recuperação de Objectos 3D
Realidade Virtual
Ambientes Imersivos
Interface Multimodal Inteligente

Keywords

Im-O-Ret
Three-dimensional Objects
3D Object Retrieval
Virtual Reality
Immersive environment
Intelligent Multimodal Interface

Contents

List of Figures

List of Tables

List of Abbreviations

3DOR 3D Object Retrieval

ACM Association for Computing Machinery

CAH Cord and Angle Histogram

CBIR Content-Based Image Retrieval

DoF Degrees of Freedom

HCI Human Computer Interaction

HMD Head Mounted Display

LFD Light-Field Descriptor

MIR Multimedia Information Retrieval

VE Virtual Environment

VR Virtual Reality

WIMP Window Icon Menu Pointer

Chapter 1

Introduction

Over the last decades, the amount of digital multimedia information (e.g. audio, images, video) has been growing. This rapid growth has urged the problem of searching for specific information in large collections. To aid us in this search, there are many search engines exist that index large amounts of information. These search engines are classified by the type of information they provide. Popular examples of text-based search engine are **Google** [BP98] and **Bing** [MVM09]. They provide a simple interface for textual input after which web pages containing these keywords are returned. Textual search engines are maturely developed and its widespread use makes them familiar to most users.

The current scenario in Multimedia Information Retrieval (MIR), is quite different. Current solutions still face major drawbacks and challenges. Among others, extensively identified in Datta's survey [DJLW08], we highlight two. First, queries rely mostly on meta-information, often keyword-based. This means that, in a closer analysis, searches can be reduced to text information retrieval of multimedia objects. Second, the result visualization follows the traditional paradigm, where the results are presented as a list of items on a screen. These items are usually thumbnails, but can be just filenames or metadata. These drawbacks are also applied to 3D search engines.

Current 3D search engines, such as the Princeton 3D model search engine [FMK+03, FKMS05], display their query results as a list of model thumbnails, which greatly hinders the interpretation of query results on collections of 3D objects. Such images, may not provide enough information for the user to identify which model is being presented in the image.

For instance, Dutagaci [DCG10], using the preferred best views selected by 26 users as ground truth, evaluates a set of seven algorithms of best view selection. One conclusion, was that none of the analysed methods did consistently well for all the objects tested, some providing images that could not identify the represented model.

1.1 Proposed Approach

With the spreading of stereoscopic viewing and last generation interaction devices outside lab environment and into our everyday lives, we believe that, in a short time, users will expect richer results, from multimedia search engines, than just a list of thumbnails. As such, to overcome the problems identified in the previous Section, we propose the use of virtual reality environments as a promising solution to those problem. Instead of presenting the query results as thumbnails, we present them as 3D objects scattered in a 3D space, where the user can literally navigate and manipulate the results. The object are distributed according to their geometric similarity, using matching algorithms for each axis of the coordinate axes. Furthermore, with the appearance of new off-the-shelf devices, it is required that our approach also aims to be modular and scalable, in order to add new devices for both visualization and interaction.

Taking into consideration the requirements described above, we built a prototype for 3D model search where query results are presented in a 3D virtual space. Our system provides a post-WIMP interface, using new off-the-shelf devices, for both the visualization and interaction. Using a modular architecture, we were also able to combine different sets of devices, as well as, make it easy to add new devices.

In order to validate our approach, an user test evaluation was conducted, in which our prototype was compared to a traditional 3D object retrieval system. Our approach proved to provide a better interpretation of results, resulting in less errors and number of steps required in order to find a specific object.

1.2 Contribution

With our approach, we had the following contribution:

- we merged the benefits of 3D object retrieval with virtual reality environments.
- we conducted an evaluation in order to validate our proposal. For this, we conducted three search task in both our prototype and a traditional 3D model search system.
- we evaluated the usability of our system, using distinct interaction scenarios with off-the-shelf devices, for both visualization and interaction.

Also, throughout the timespan of this work we published a paper at "The 1st European Workshop on Human-Computer Interaction and Information Retrieval" (**EuroHCIR2011**). In this paper, we presented the potential of using immersive VEs for 3DOR result visualization. The reference of this contribution is as follows:

- Pedro B. Pascoal Alfredo Ferreira and Manuel J. Fonseca. Back to mars: The unexplored possibilities in query result visualization. Proceedings of 1st European Workshop on Human-Computer Information Retrieval, July 2011

1.3 Outline

This dissertation is structured in six chapters. Next chapter provides the overview of some basic concepts for 3D object retrieval, namely an overhaul on the state-of-art of 3D shape description. Throughout the third chapter we discuss related work that is relevant to our proposal. In chapter four we describe our proposed architecture in detail. We follow it with an user evaluation, presented in the fifth chapter, where we draw a critical comparison of our proposal against standard solutions, using data collected from test sessions. Finally, in chapter six we present an overall discussion of our work, delineating conclusions and introducing perspectives for future work.

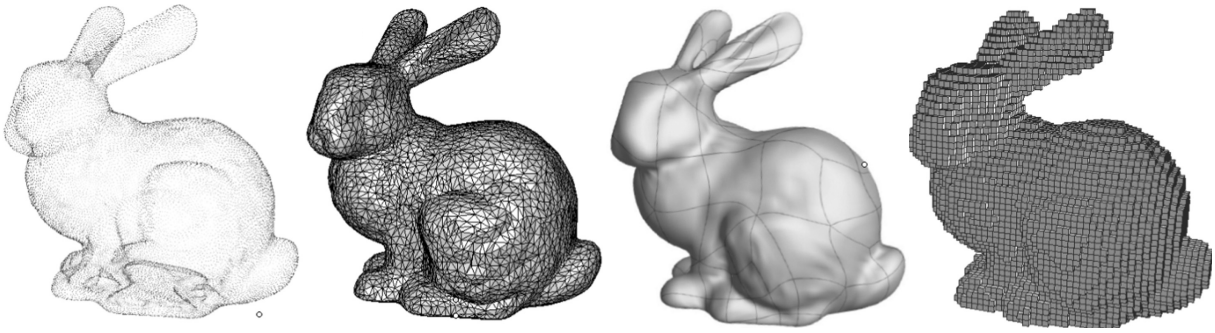
Chapter 2

Background

In this chapter we present some fundamental concepts necessary for an easier understanding of our work. The use of concepts and techniques for 3D Object Retrieval (3DOR) in our work, makes it crucial to provide a description of each step involved in the 3DOR process. Also, since shape descriptor are a central part of the process it is also required to give a brief explanation of what they are and introduce some examples of their different types.

2.1 3D Object Representation

Three-dimensional models are multimedia data very complex to handle cause they have different types of representation. The same object in different domains will have different representations. There are three main representations schemes: point-based representation (Figure 2.1a), surface-based representation (Figures 2.1b,2.1c), and volumetric representation (Figure 2.1d).



(a) Cloud of points (b) Triangular mesh (c) Set of parametric surfaces (d) Set of voxels

Figure 2.1: Different representations of *Bunny* [DBD08]. ©2008 John Wiley & Sons

The point-based models are defined by a set of points in 3D space, which reproduces a real-world object, that was captured using 3D scanners. Since this cloud of points is not efficient for computing physical or geometrical properties, it is often converted to polygonal meshes, more particularly triangular meshes.

The surface-based models, such as parametric, implicit and subdivision surfaces, represent the 3D object by its boundary, using one or more pieces of surface. The parametric surfaces are define using mathematically exact surfaces (Figure 2.1c). Using a set of control points, they are mapped into smooth-continuous surfaces. Contrary to the parametric surfaces, the polygonal meshes represent only an approximation.

The polygonal mesh, depicted in Figure 2.1b, also represent the object by its boundary surface, but is composed by a set of planar faces. It contains two kinds of information: geometry, position of the vertices in the 3D space; and connectivity, which vertices are linked by edges to form a set of polygonal faces.

The volumetric models decompose a complex 3D object into a set of simple primitives. These group can be divided into: primitive-based models, constructive solid geometry (CSG), and discrete model. In the discrete representation, the 3D object is decomposed into volume elements, called voxels, and represented in a 3D grid similar to the 2D image representation (Figure 2.1d).

Despite the variety of object representations, none is appropriate to efficiently compare two model. Indeed, some geometric measures exist in order to check the difference of 3D objects, but these geometric measures are not well matched with the visual human perception. As such, in order to compare two models, a similarity metric has to be define to compute the visual similarity. For this are generally used shape descriptors.

2.2 Shape Descriptors

Usually, shape matching algorithms rely on feature vectors, which are the standard approach for multimedia retrieval [FSZ03]. A feature vector, also referred as shape descriptor, is a way to describe characteristics of an object by using a one-dimensional function. However, a descriptor is not universal, since the information of a descriptor will vary depending on the type of comparison method used.

There is no universally accepted taxonomy of 3D shape descriptors, due to the variety of distinct approachs. Bustos et al. [BKS+05] propose a taxonomy where the shape descriptor are divided into five categories, resulting of the technique used to construct the numerical representation of the shape. Iyer et al. published another state-of-the-art review [IJL+05] where they classify and compare several 3D shape searching techniques from a CAD/CAM perspective and suggest future trends.

However, we will follow a classification scheme proposed by Tangelder and Veltkamp [TV08]. The matching methods are divided into three categories, based on the representation of the shape descriptor. These three categories are: feature based, graph based and geometry based. For each of the three categories we will present one or two descriptors we consider relevant, since they were implemented for the developed of work.

Feature Based

In the context of comparison, feature based methods denote topological and geometric properties of 3D shapes. Can be sub-divided into four categories according to the type of shape feature used: global features, global feature distributions, spacial maps, and local feature.

Global features describe the overall shape of the 3D model. Examples of such features are statistical moments of the border or the volume model, the volume ratio of the surface, the Fourier transform of the volume or the boundary of the shape. As global features are only used to describe the overall shape of the object, these methods are not very descriptive about the details of the objects, however, its implementation is simple.

One example is the Cord and Angle Histogram presented by Paquet et al. in [PMN+00, PR97a]. 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.

A refinement of the global features are the distributions of global features, which measure properties based on distance, angle, area and volume calculated from random points on the surface. Osada et al. [OFCD02] introduces the D2 shape distribution, which measures distances of random surface points. As we can see in Figure 2.2, the distributions are so good distinction of the models by broad categories: cars, people, animals, and so on. However, they have poor performance when used to distinguish between shapes that have similar overall shape properties but different detailed shape properties.

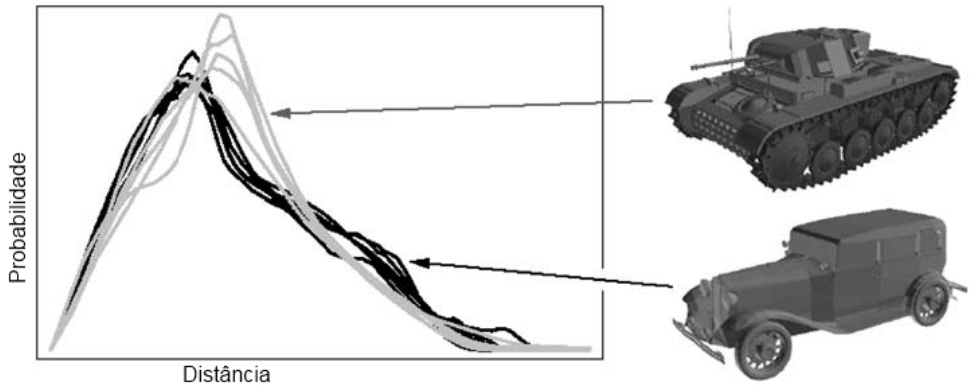


Figure 2.2: D2 shape distributions of five tanks (gray curves) and six cars (black curves) [OFCD02]. ©2002 ACM

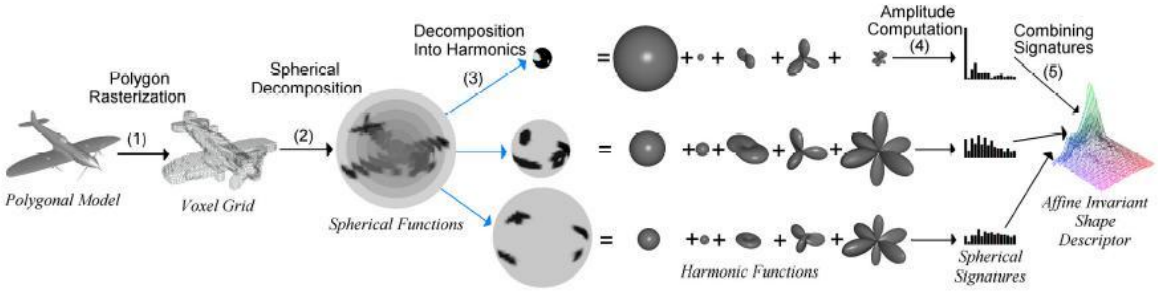


Figure 2.3: Computing spherical harmonics shape descriptors [FKMS05]. ©2005 ACM

Spacial maps are representations that capture the spatial location of an object. The map entries correspond to physical locations or sections of the object, and are organized in a way that preserves the relative position of features. Spatial maps are generally not invariant to rotations. Therefore, a normalization of the orientation of the model is usually done first. To overcome this limitation, Kazhdan et al. [KFR03] presents a new approach based on spherical harmonics to obtain the rotation invariant representation of three-dimensional objects. The main idea is to decompose a 3D model into a collection of functions defined on concentric spheres and to use spherical harmonics to discard orientation information (phase) for each one. This involves several steps, that are illustrated in Figure 2.3.

In contrast to the methods of the first three categories, the methods of local features describe the 3D shape around a number of points on the surface. To this end, a descriptor is used for each point on the surface instead of a single descriptor. Compared to the methods described previously, matching 3D shape contexts is less efficient, efficient indexing is not straightforward, and the obtained dissimilarity measure does not obey the triangle inequality.

Graph Based

Unlike shape descriptors based on features who only takes the geometric shape into account, the graph based try to extract meaning from the geometry using a graph to indicate how the components are connected. Using graphs to represent relations between components of the object, these approaches focus on the topological of the model. This category is further sub-divided into three categories according to the type of graph used: model graphs, reeb graphs, and skeletons.

Sundar et al. [SSG+03] use as a shape descriptor a skeletal graph that encodes geometric and topological information. This example is depicted in Figure 2.4. Each node in the graph represents a segment of the original skeleton. With each node a geometrical signature vector is associated. Then, two shape are matched by approximate comparison of their hierarchical skeletal graphs.

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

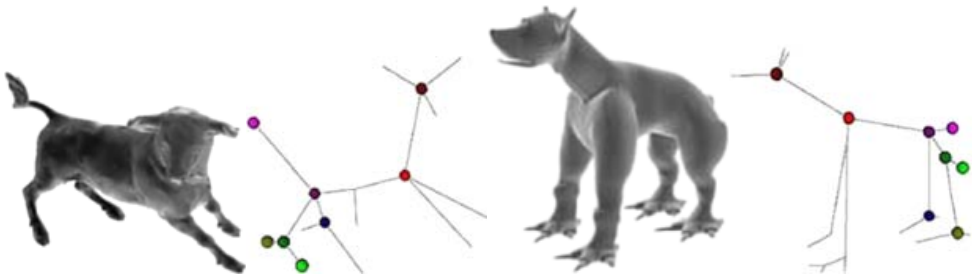


Figure 2.4: Skeletal graph matching with colors showing the node-to-node correspondence based upon the topological and radial distance about the edge.[SSG+03] ©2003 IEEE Computer Society

of each approach scope to a particular object type. Moreover, feature based have the advantage of being able to find similarities between different parts connected, where the graph based are not.

Geometry Based

Finally, we discuss the geometry based shape descriptors. The geometry based methods are sub-divided into four categories: view based, volumetric error based, weight point set based and deformation based. The view-based, which we will focus, is based on the idea that two 3D models are similar, if they look alike. This results that often are used 2D sketches as query, being later used image matching techniques for the comparison with the descriptors from shapes in the database. Chen et al. proposed a descriptor based on silhouettes from several different viewing directions [CTtSO03], named Light-Field Descriptor (LFD). They consider that *"if two 3D models are similar, they also look from all viewing angles"*. The general idea is to compare 2D silhouettes of the 3D shape, obtained from different viewing angles equally distributed around the viewing sphere, as Figure 2.5 illustrates.

First, it is required to normalize the model, in which translation and scale are applied to ensure that the model is entirely contained in rendered images. Then multiple images are processed, each corresponding to a silhouette. Finally, for each silhouette are extracted the feature vectors. To produce these feature vectors it is used the Zernike moments for 2D shape and the Fourier descriptors for the contour.

To improve the robustness invariance a set of ten light field descriptors is applied to each 3D model,

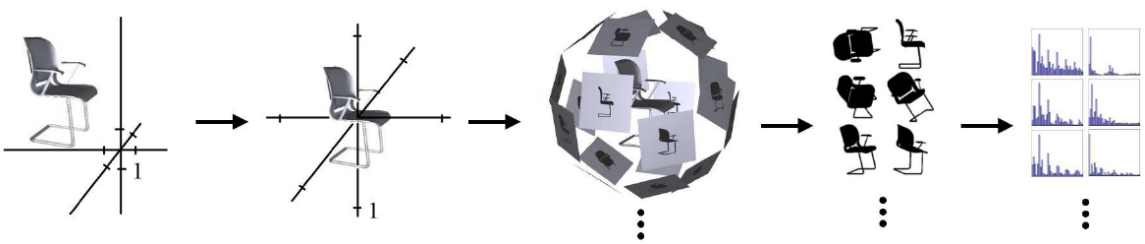


Figure 2.5: Steps of extracting the Light-Field Descriptors for a 3D model.[tSCTO03] ©2003 Eurographics

created from different orientation systems of the camera. This leads to the processing of one hundred silhouettes. Thus, the dissimilarity between two models will result in the minimum difference between all combinations of LFD.

Shilane et al. [SMKF04] applied the Princeton Shape Benchmark to compare twelve shape descriptors. In their study, the lightfield descriptor was found as the most discriminating between the twelve shape descriptors tested, but at higher storage and computer cost than most other descriptors.

Beside the shape descriptors studied here there are many other, however, since an exhaustive study of 3D descriptors is not focus of this work, they will not be discussed here. A more complete study on shape descriptors can be found at [BKS+05, DBD08, IJL+05, TV08]

2.3 Similarity Measures

Similarity measures are essential in order to measure how similar two objects are. To measure the similarity between two objects, it is first necessary to compute the distances between the shape descriptors. Following the terminology used by Tangelder and Veltkamp [TV08] we will use the term dissimilarity to measure how similar two objects are. Although the term similarity is most often used, dissimilarity better corresponds to the notion of distance, since small distance means small dissimilarity, therefore, very similar.

There are various approaches to measure the distances between two points in space, however we will only focus on the Minkowski distances, which are the most used in multimedia information retrieval. A more extensive study of similarity distances is presented by Feng et al. [FSZ03] and Chan [Cha07], which can be seen to a better understanding of this topic. The Minkowski distances [FSZ03, Cha07] are a set of three formulas, $L_1, L_2 e L_\infty$, generalized by Hermann Minkowski and depicted in Table 2.1.

Minkowski	$L_p = \sqrt[p]{\sum_{i=1}^{d} \lvert P_i - Q_i \rvert^p}$,	$p = 1, 2, \infty$
Manhattan	$L_1 = \sum_{i=1}^{d} \lvert P_i - Q_i \rvert$,	$p = 1$
Euclidean	$L_2 = \sqrt[2]{\sum_{i=1}^{d} \lvert P_i - Q_i \rvert^2}$,	$p = 2$
Chebyshev	$L_\infty = max_i \lvert P_i - Q_i \rvert$,	$p = \infty$

Table 2.1: Minkowski generalized formulas

When $p = 1$, $L_1 = \sum_{i=1}^{d} |P_i - Q_i|$, the formula corresponds to the Manhattan distance, also introduced by Hermann Minkowski [Cha07]. The Manhattan distance between two points is measured by the sum of the lengths of projections on the coordinate axis, of the line segment that connects the points.

With $p = 2$, $L_2 = \sqrt[2]{\sum_{i=1}^{d} |P_i - Q_i|^2}$, we have the Euclidean formula. This formula was created by

Euclid over a thousand years to determine the distance between two points by calculating the length of the straight line formed between them. This formula can be proved by applying the Pythagorean theorem.

Finally, for $p = \infty$, $L_\infty = max_i|P_i - Q_i|$, we have Chebyshev, named after Pafnuty Lvovich Chebyshev. The distance between two points is measured as the biggest difference between them, on one of the coordinate axis. It is also the chessboard distance in 2D, the minimax approximation.

The similarity measures presented here, are some of the simplest and well known approaches to measure distances between two points. A more extensive study is presented at [FSZ03, Cha07].

Although, with the similarity measures presented and shape descriptors, described in the previous Section, it is already possible to determine the most similar models by sequentially comparing each two models, using such method in large collections, would not be practical task. As such, retrieval system normally resort in using indexes in order to speed up the search process and reduce the number of element compared.

2.4 Indexing and Retrieval

In overview, a retrieval system has two phases: indexing and retrieval [DBD08, FSZ03, TV08]. Figure 2.6 briefly illustrates the various steps conducted on our prototype implementation of the indexing procedure. Thus, it is necessary to note that other systems may have a slightly different implementation of those described here.

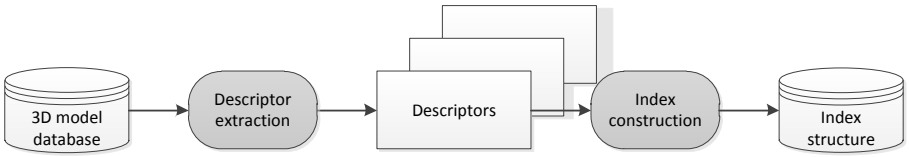


Figure 2.6: Conceptual overview of the indexing procedure.

The main purpose of indexing would be to reduce the number of elements to be compared, in order to facilitate fast content-based retrieval of multimedia objects in large databases [Wu97]. The descriptors are extracted for each model from the database and an indexing structure built with these descriptors. Normally it is necessary to perform a preprocessing, which consists in the normalization of the 3D object. This is required because the measures of dissimilarity are invariant regarding scale, translation and rotation of the models, and these may have arbitrary scale, position and rotation in the 3D space. After the extraction of the descriptors of all the 3D models in the database, it is then created our index, that shall be used by the retrieval process to accelerate in the searching of specific data.

In the retrieval process, illustrated in Figure 2.7, the extraction process will be similar to that of the indexing. The shape descriptor extracted from the query model, will be compared with the descriptors

stored in the index. The index consists of a collection of entries, each containing the key for an item, and a reference pointer which allows immediate access to that item. Then, the identifiers returned, are mapped to the identifiers of the models in the database. Finally, the corresponding models are returned for visualization. In our current prototype, only the query by example [Zlo77], using a sample model was implemented.

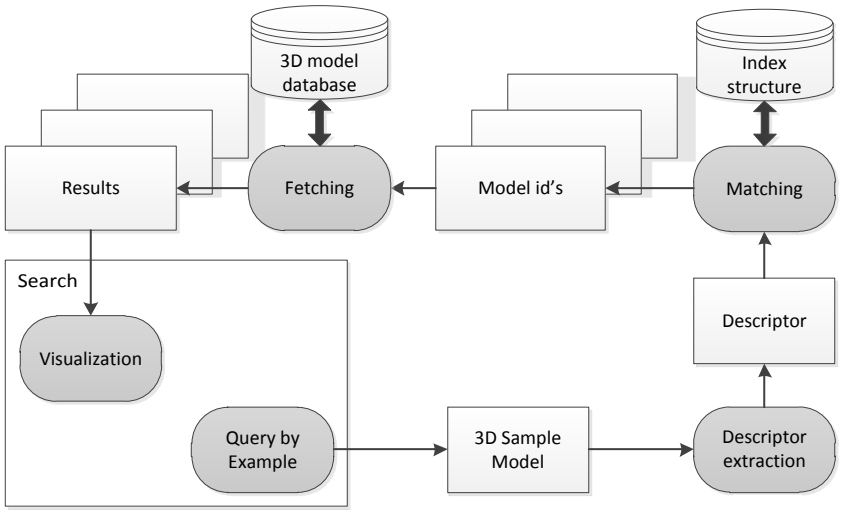
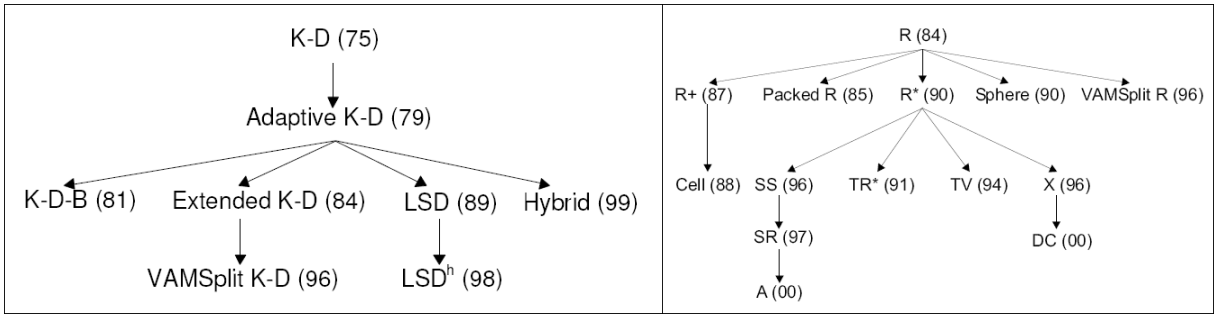


Figure 2.7: Conceptual overview of the retrieval procedure.

To accelerate the searching of specific data items in the index, most database systems use a tree indexing mechanism [FJ03, Gut84, Ben75]. Figure 2.8, shows the most common indexing structures. Since the KD-Tree[Ben75] and R-Tree[Gut84] are the root of most of the indexing structures, we shall overview how they structure the items. Also, a description of the NB-Tree[FJ03] is required since it was used as index mechanism for our prototype.



(a) KD-Tree

(b) R-Tree

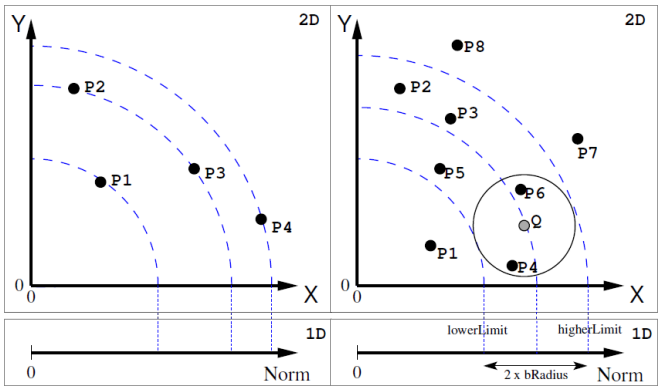
Figure 2.8: Family trees of most common indexing structures.[FJ03]

The KD-Tree introduced by Bentley [Ben75], is a binary tree where each node is a point of size k. The space is partitioned into regions of k-dimension, where each non-leaf node serves as a partition plan. The resulting regions are mutually disjoint, and their union forms the whole space. The division of space is made independently of the data. Contrary to what happens in the KD-Tree, the R-Tree presented by Guttman [Gut84], partitions the space according to the data distribution. The data is divided hierarchically using minimum bounding boxes. Each entry of a non-leaf node will have stored two types of information: a way of identifying a child node and bounding box with all the entries in this child node. Both these structures allow to query the stored elements. Table 2.2 shows some of the most common types of queries conducted index trees.

Exact match	Find objects equal to the query.
Range search	Find objects within a range.
k-Nearest Neighbours	Find the k most similar objects.
Within distance	Find all objects within a similar degree to the query.

Table 2.2: Tree query types

Finally, the NB-Tree is an indexing system proposed by Fonseca and Jorge [FJ03], which aims to offer a compact and simple solution for indexing large volumes of multimedia data. For the index is then used a $B^+ - Tree$, and as key value it uses the Euclidean norm. This method, first maps $R^D \rightarrow R^1$, as shown in Figure 2.9a. This will help reduce the cost of sequential scan of the elements of $B^+ - Tree$. When querying, the system will only consider points whose norm is in the neighbourhood of norm of the searched point, as illustrates Figure 2.9b.



(a) Dimension Reduction. (b) 2D Range query example.

Figure 2.9: *NB-Tree* [FJ03]

2.5 Summary

Since most of work is around the retrieval and use of 3D object, we presented a brief introduction on various topics related to object representation, shape descriptors, similarity measures, indexing and retrieval of 3D objects.

As we seen, in order to compare two 3D objects are used feature vectors, also referred as shape descriptor, as a way to describe characteristics of an object by using a one-dimensional function. These shape descriptors can be divided into three categories, based on the representation of the shape descriptor. These three categories are: feature based, graph based and geometry based.

Then it was discussed the steps involved in the indexing 3D object. The descriptors of all model are extracted and then indexed, creating the index, that is then used by the retrieval process to accelerate in the searching of specific data. The retrieval process, in the event of query, will extract its the shape descriptor, and compare it to the one stored in the previously created index. Finally, the most similar models are returned for visualization.

In our work, we used the Princeton Shape Benchmark [SMKF04] model database, and three shape descriptors: the Light-Field Descriptors [CTtSO03], the Cord and Angle Histogram [PR97a], and the Spherical Harmonics Descriptor [KFR03]. Using each of the three shape descriptors, we extracted the feature vectors from all models, and indexed the feature vectors using a NB-Tree [FJ03]. Then, for query it used the k-Nearest Neighbours available in the NB-Tree.

Chapter 3

Related Work

In the present chapter we start by looking at some 3D-model search engines, both academic and commercial, which represent the state-of-art regarding 3DOR. Next, we describe the 3D MARS proposed by Nakazato [NH01a]. The concepts behind the 3D MARS are very important since, in regards to the interface, they resemble our proposed work. To the extent of our knowledge, there has not been presented any research or new solution that take advantage of immersive virtual environments for information retrieval.

Following the study of the 3D MARS, we present some techniques for visualization, navigation and interaction in 3D virtual environments. In this section, we make a comparison of different methods, pointing advantages and disadvantages of each approach. Finally, we give an overview of different post-WIMP user interfaces. During this study we present examples of systems which, similar to us, used a post-WIMP user interface to gain a more comprehensible, predictable and controllable interaction.

3.1 3DOR Search Engines

Last decades, several 3D shape search engines have been presented. In 1997, Paquet and Rioux presented the Nefertiti[PR97b], a query by content software for three-dimensional model databases. It incorporates a set of retrieval algorithms that allows database searches by scale, shape, color or any combination of these parameters.

Later in 2001, the Princeton 3D model search engine is introduced by Thomas Funkhouser et al. [FMK+03, FKMS05]. This work provides content-based retrieval of 3D models from a collection of more than 36000 objects. For query specification it has four options available: text based; by example; by 2D sketch(Figure 3.1a); and by 3D sketch (Figure 3.1b). The results of this queries are presented as an array of model thumbnails, has depicted in Figure 3.1. After a search, it is also possible to choose a result as query-by-example for a new search.

(a) Search by 2D Sketch. (b) Search by 3D Sketch.

Figure 3.1: Princeton 3D Model Search Engine.

Later in 2003, Shen et al. presents a retrieval system [tSCTO03] based on the light field descriptor [CTtSO03]. This descriptor, as well as the processes of extraction, indexing and retrieval of 3D models, were explained in chapter 2, where we gave a brief overview of all the processes involved in 3DOR. This system, contains a database of 10,911 3D models. For query interface, there are available by text and by 2D silhouette drawing.

Additionally to queries by example and sketch-based queries, Ansary et al. proposes the FOX-MIIRE search engine[AVD07], which introduces the query by photo. This tool, shown in Figure 3.2, was the first capable to retrieve a 3D model from a similar photograph. Also, this system has both standard and mobile device interfaces available, incorporating the idea of a 3D model retrieval system in a mobile device proposed by Suzuki et al. [SYS03]. However, and similarly to previous engines, the results are displayed as a thumbnail list.

More recently, in 2008, Bonhomme et al. deployed MyMultimediaWorld.com [BMC+08, LBPP08], an online platform for sharing various types of media, including video, image, audio and 3D objects. The similarity search presents the most relevant similar objects and some evaluation measures for all descrip-

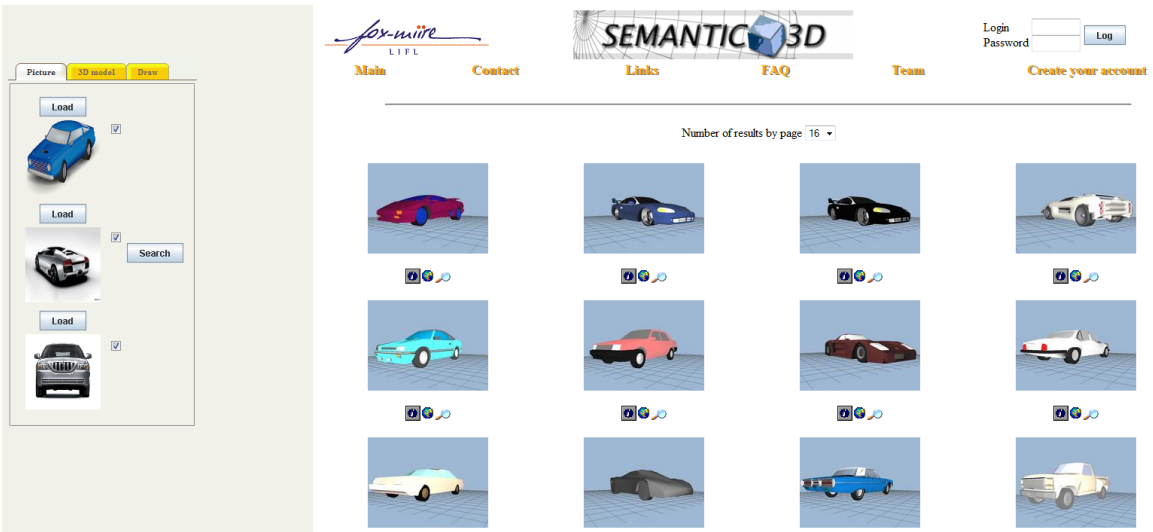


Figure 3.2: FOX-MIIRE: Query by photo.

tors. This platform follows the MPEG standards, using MPEG-4 for the representation of media and MPEG-7 for its description. It also gives the possibility to see benchmark results.

Outside the research field, **Google 3D Warehouse** [War] offers a text-based search engine. This online repository contains a very large number of different models, from monuments to cars and furniture, humans and spaceships. However, searching for models in this collection is limited by textual queries or, when models represent real objects, by its geographic reference. On the other hand, the results are displayed by model images in a list, with the opportunity to manipulate a 3D view of a selected model 3.3.

Generally, the query specification and visualization of results of other commercial tools for 3D object retrieval, usually associated with 3D model online selling sites, did not differ much from those presented above. The query is specified through keywords and results are presented as a list of model thumbnails.

Despite the growing of current hardware and software, these approaches to query specification and result visualization do not take advantage of latest advances of neither computer graphics or interaction paradigms. Indeed, to the extent of our knowledge, there has not been presented any research or new solution that take advantage of immersive virtual environments for information retrieval since Nakazato's 3D MARS [NH01a].

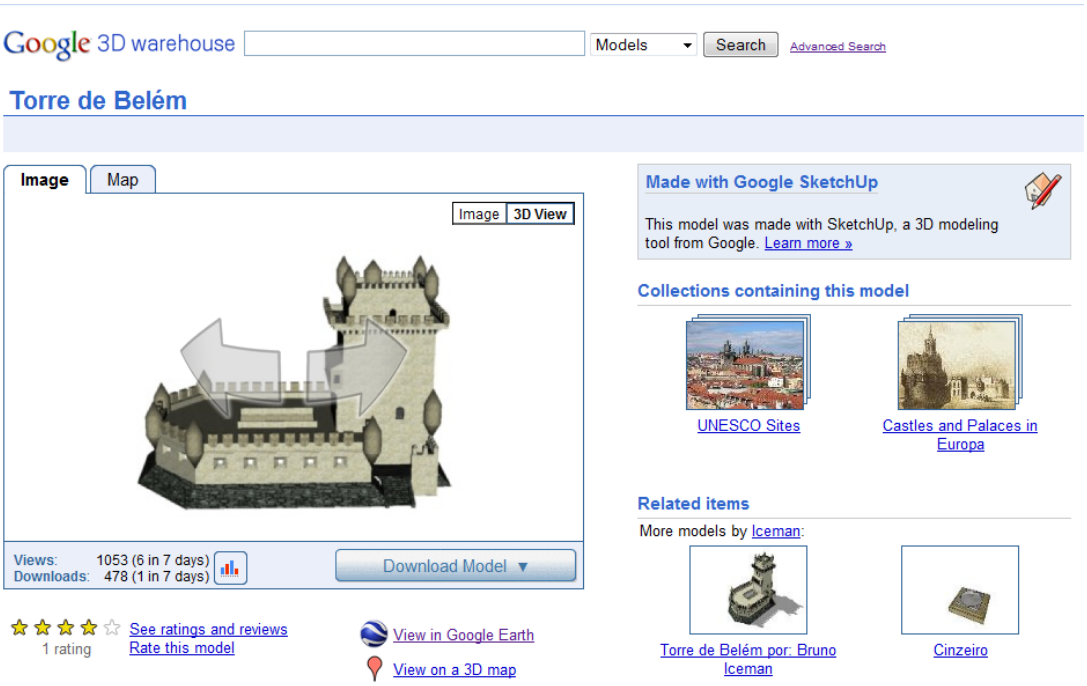


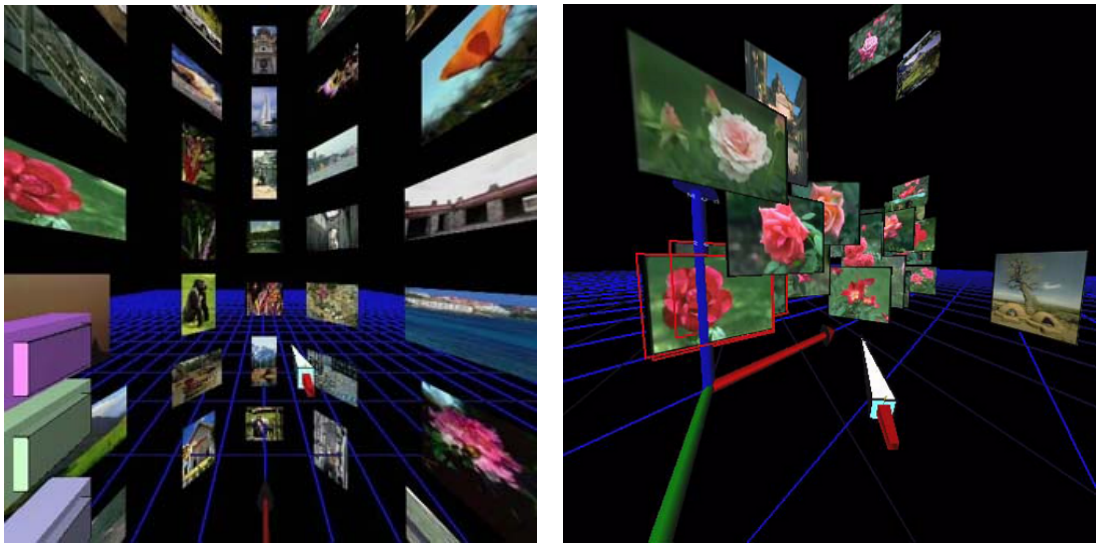
Figure 3.3: Google 3D Warehouse: 3D view of a selected model.

3.2 3D Mars

A decade ago, Nakazato proposed 3D MARS [NH01a, NH01b], an immersive virtual reality environment for content-based image retrieval. The user selects one image from a list and the system retrieves and displays the most similar images from the image database in a 3D virtual space, as shown in Figure 3.4. The image location on this space is determined by its distance to the query image, where more similar images are closer to the origin of the space. The distance in each coordinate axis depend on a pre-defined set of features. The X-axis, Y-axis and Z-axis represent color, texture and structure of images respectively. Nakazato focused his work on query result visualization. Thus 3D MARS supports only query-by-example mechanism to specify the search.

The 3D MARS system demonstrates that the use of 3D visualization in multimedia retrieval has two main benefit. First, more content can be displayed at the same time without occluding one another. Second, by assigning different meanings to each axis, the user can determine which features are important as well as examine the query result with respect to three different criteria at the same time.

The interaction with the query results was done on the NCSA CAVE [CNSD93], which provided fully immersive experience, as depicted in Figure 3.4, and later on desktop VR systems. By wearing shutter glasses, the user can see a stereoscopic view of the world. In such solution, visualizing query results goes far beyond scrolling on a list of thumbnails.



(a) Initial Configuration. Images are selected at random.

(b) The result after a query for a flower image.

Figure 3.4: Interface of 3D MARS. [NH, NH01a]

Figure 3.5: 3D MARS user interaction on CAVE Environment. [NH]

3.3 Exploring 3D Environments

Immersive 3D environments are often associated with virtual reality environments. Virtual Reality (VR) is defined by Aukstakalnis Blatner [AB92] as "a way for people to visualize, manipulate and interact with computers and extremely complex data". Based on this definition and the fact that 3D models are multimedia data that is very complex to handle [DBD08], we believe that the use of virtual reality environments is the ideal for our work.

Tan et al. [TRC01] extends this definition, characterizing Virtual Environment (VE) as a larger space than can be displayed. It should thus be possible for the user to see different views of the virtual world. As such, it is also required an efficiently and convenient way to move between location. It is also essential for the navigation to be quick and natural, so that the user can focus on the tasks that are really important.

Due to the dimensions of VE, Fukatsu et al. [FKMK98] concludes that despite the diversity of approaches to navigation in virtual worlds, there is a natural tendency of the user to lose his position in the virtual world. Therefore, it is often necessary to use the navigation aids, that would increase the perception. The same Fukatsu et al. proposes the use of the birds eye to assist the navigation.

Since the notions of distance and movement that are present in the real world, as well as limitations to the field of view, are not present in the virtual environment. These extra degrees of freedom can sometimes make it difficult for users to understand the system. Thus, extra support should be given to users (e.g. increasing the field of view, realistic motion, sound, maps).

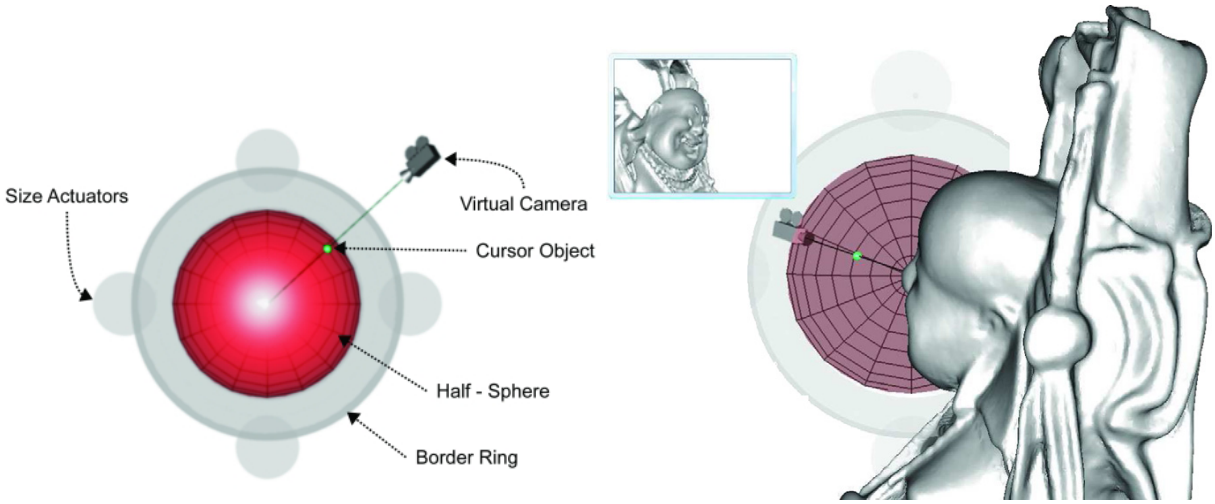
3.3.1 Navigation

There is a large set of parameters to take into account and that obviously depend on the metaphor used for navigation. Bowman et al. [BDHB99] proposes the division into two types of navigation based on their movement as: *wayfinding* and *travel*.

The *wayfinding* navigation must have prior knowledge of the space, in order to create a cognitive map of the environment. The movement is done based on a coordinated that specifies the position where the camera is moved to. Mackinlay et al. [MCR90] proposed the navigation by points of interest. A point of interest is defined as a desired location on a particular object of the scene. Chosen a point of interest it is calculated the trajectory so that there is an approximation of it. This method is not only used for calculating the translation point of view, but also for calculating the correct orientation using the information for that of the normal vector point of interest chosen.

The navigation to points of interest is however limited in very dense worlds (with many objects), since many objects are naturally occluded, thus making navigation limited. Hachet et al. proposes the Navidget [HDKG08, KHG08], which extends the techniques of points of interest. As Figure 3.6 illustrates, the technique uses half-sphere on which the user can define and adjust the direction of view, controlling the cursor over the ball you set the camera position. After defining this position and a smooth trajectory calculated from the current position and target position.

Compared with the technique of point of interest where the user simply sets the target, this new technique offers the possibility to control the distance to the target as well as direction for the visualization of it. In the first face, the user is able to position the focus area by directly moving the virtual ray in the scene. If the ray intersects an object during the dragging movement the half-sphere is placed at the intersected



(a) The Naviget.

(b) The Naviget preview window.

Figure 3.6: The *Naviget*. [HDKG08, KHG08]

position, else the sphere sticks with the ray within the distance of the last intersection. The second face relates to the positioning of the camera. In this step the user can define the radius of the semi-sphere that defines the distance to objects, as well as moving the cursor over the hemisphere in order to define the point of view camera.

The use of this technique for distant objects produces weaker results. As solution they proposes the addition of a preview window that is used to predict the camera position and thereby assist the proper positioning of this as depicted in Figure 3.6b. A later study by the same authors [HDKG09] extends this technique for use on multiple platforms including mobile phones, PDA and TablePC for interaction with environments, large-scale visualization.

Contrary to the *wayfinding*, the *travel* navigation requires no prior knowledge of the space. An example of *travel*, was the *Avatar* proposed by Tan et al. [TRC01]. This technique consists, in general, putting the camera in eye position of the object will move. It is traditionally used in 3D games, where the main idea is based on the user impersonating a scene object and be able to see the world through this same object.

Devices like the Head Mounted Display (HMD), are another example of the use travel of navigation. Typically, these devices have the guidance of the *viewport* being controlled by *"head tracking"*. A pioneer in this type of interaction was Sutherland et al. [Sut68], with motion sensors that could detect the correct rotation of the head, which is used in camera orientation. This system only considers techniques of *viewport* orientation.

Later Fisher et al.[FMHR87] extended this approach, by using devices such as gloves with which was possible to detect pre-defined movements that allowed the navigation and interaction of the scene. An example of these type of devices we have the *5DT Data Glove* [Min04] used in our prototype. The *5DT Data Glove* is a glove with flexible fibre optic sensors that capture gestures and hand movements, as depicted in Figure 3.7a. It has six sensors distributed as shown in Figure 3.7b, and detects finger bending, as well as rotation and closing the hand.

3.3.2 Visualization

The visualization of virtual reality environments can be broadly divided into two types: non-immersive and immersive. The immersive is based on the use of helmet or projection rooms (eg, HMDs [SSDP$^+$09], CAVEs [CNSD93, DAA$^+$11]) while the non-immersive virtual reality based on the use of monitors. The notion of immersion, is based on the idea of the user being inside the environment.

HMDs have two LCD screens in a relative position of each eye, which draws the virtual world depending on the orientation of the user's head via a tracking system. The Figure 3.8a illustrates an example of an HMD. HMD's can display the same picture in both eyes or be stereoscopic by combining separate images

from two offset sources. Both of the offset images are then combined in the brain to give the perception of 3D depth.

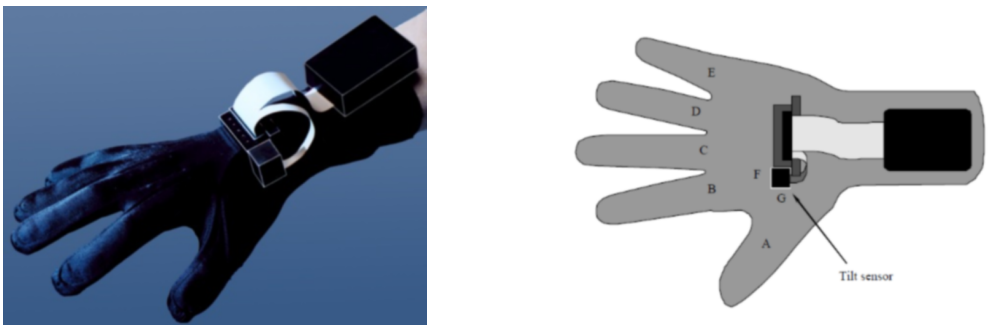
In the 3D MARS [NH01a] the images are dispersed in an immersive space of the NCSA CAVE [CNSD93]. The CAVE is an environment surrounded by views and sounds, where through a stereoscopic projection, it gives the illusion of being completely within a virtual environment.

VR may also be mixed with reality to create an augmented reality environment. In augmented environments, the vision of the real environment is overlaid with information from the virtual environment. In this type of systems the glasses or helmets usually have a semi-transparent display so as to allow a mix of real and virtual. The Figure 3.8b illustrates an example of a 3D viewer that contains liquid crystal. Liquid Crystal shutter glasses contains liquid crystal that will block or let light through in synchronization with the refresh rate of the screen. The display alternately displays different perspectives for each eye, using the concept of alternate-frame sequencing.

VE as a large space enables the display of more content at the same. Also the use of stereoscopic view of the world, gives a visualization of the world that goes far beyond 2D view of the screens. Additionally, with the growingly and common use of new human-computer interaction (HCI) paradigms and devices brought new possibilities for multi-modal systems.

3.4 Post-WIMP Interaction

Generally, post-WIMP approaches abandon the traditional mouse and keyboard combination, favouring devices with six Degrees of Freedom (DoF). Unlike traditional Window Icon Menu Pointer (WIMP) interaction style, where it is necessary to map the inputs from a 2D interaction space to a 3D visualization space, six DoF devices allow straightforward direct mapping between device movements and rotations and corresponding effects on the three-dimensional space.



(a) 5DT Data Glove. (b) Positions of the sensors.

Figure 3.7: 5DT Data Glove [Min04]. ©2004 5DT

(a) *Z800 3DVisor* (b) *CrystalEyes 3*

Figure 3.8: *Head-Mounted Display and Shutter Glasses*

Balakrishnan et al. [BBKF97] proposed a classification where interaction techniques can be broadly classified into two categories: those based on three or more degree-of-freedom input devices, and those which rely on the ubiquitous mouse coupled with a variety of schemes for mapping 2D input to 3D control. The biggest difference between these two categories will be, as you can see, whether or not there is direct mapping.

This represents an huge leap to the concept of direct manipulation, which, according to Shneiderman [Shn97], rapidly increments operations and allows the immediate visualization of effects on an manipulated object. This helps making the interaction more comprehensible, predictable and controllable.

The recent dissemination among common users of new Human Computer Interaction (HCI) paradigms and devices (e.g. Nintendo Wiimote[Nin06, Lee08] or Microsoft Kinect[Mic10]) brought new possibilities for multi-modal systems. For decades, the WIMP interaction style prevailed outside the research field, while post-WIMP interfaces were being devised and explored [vD97], but without major impact in everyday use of computer systems.

Particularly, the use of gestures to interact with system has been part of the interface scene since the very early days. In 1980, Bolt wrote ”Put-that-there”[Bol80], a pioneering multi-modal application where the use of natural language helps direct manipulation to describe arguments to select actions. In “Put-that-there”, the user commands simple shapes on a large-screen graphics display surface. This approach combined gestures and voice commands to interact with the system.

In 1994, Koons et al. introduces the ICONIC [KS94].With the ICONIC, they present a new form of interaction based on the detection of hand gestures. This aims to better represent some operations allowed to the user. This way facilitates some commands that, so far, were unnatural. A few years later, Sharma et al. [SZP+00], describes a system with bi-modal interface, voice and gestures, integrated into a 3D visualization system, where users model and manipulate 3D objects, in this case, models of biomolecules.

Billinghurst [Bil98] in his survey of several work done between 1980 and 1998, points the fact that natural language and gestures individually are insufficient. *Gesture recognition is not as efficient as speech* and *A spoken vocabulary has a more standard interpretation than gesture.* However, when used together, both hand gesturing and speech complement each other, providing a more natural and free interaction to the user.

Recently such interaction paradigms have been introduced in off-the-shelf commodity products (Figure 3.9). Now, with limited resources, novel and more natural HCI can be developed and explored. For instance, Lee [Lee08] used a Wiimote and took advantage of its high resolution infra-red camera to implement multipoint interactive whiteboard, finger tracking and head tracking for desktop virtual reality displays.

The **Wii Remote** [Nin06, Lee08], illustrated in Figure 3.9a, is equipped with a infra-red camera and a three-axis accelerometer and gyroscope, which gives six DoF. Cochard and Pham [1], showed how the accelerometers and optical sensor of the wiimote, by allowing the user to move 3D objects directly with the movements and gestures of their hands, can enhance the user experience within a 3D environment.

Similar to the wiimote, the **PS Move** [Ent10] also possess a three-axis accelerometer and gyroscope. Although, thanks to the use of a magnetometer it provides a better precision and greater control of movement. Also, unlike the wiimote that has built-in a camera and transmits its targeting position according to the location of infra-red censor bar, the psmove uses the *PlayStation Eye* camera to discovers his position of thanks to a luminous sphere on his head. It calculate the distance of the command to the camera, thus, enabling the tracking of the depth position of controller.

The **Kinect** [Mic10], displayed in Figure 3.9c, is a device sold by Microsoft, capable of 3D motion cap-

(a) Wii Remote [Nin06]. ©Nintendo (b) PS Move [Ent10]. ©Sony (c) Kinect [Mic10]. ©Microsoft

Figure 3.9: Set of commercial devices.

[1] Navigation in a 3D environment: Using the Wiimote to Map Natural Gestures. http://video.davidcochard.fr/projects/CS6456/Cochard_Pham_report.pdf, accessed on 6/10/2011

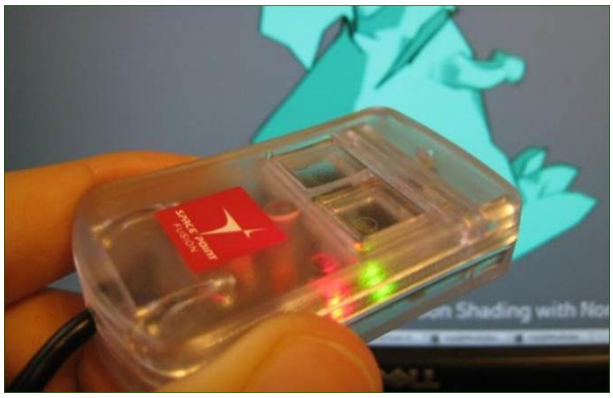


Figure 3.10: SpacePoint Fusion [Cor10]. ©PNI Sensor Corporation

ture, voice recognition and body. To this end, it comes equipped with an RGB camera, depth sensor and multi-microphone array. The camera helps in facial recognition and color detection. The depth sensor consists of an infra-red spotlight and a monochrome CMOS sensor, which combined can offer a 3D view of the room, under any lighting conditions. The microphone consists of a series of four microphones that can isolate the voices of the players from ambient noise.

In January 2011, during two weekends in Barcelona, the blablabLAB installed a system, named Be Your Own Souvenir[2], which printed a small three-dimensionally object from a volumetric reconstruction generated by the use of three Kinects. In the context of 3DOR, Holz and Wilson [HW11], present a system which allows users to describe spatial object through gestures. The system capture gestures with a Kinect camera and then finds the most closely matching object in a database of physical objects. This system represents a good example of the use of new interaction paradigms in 3DOR, which help give the user a more natural way of interaction.

Another device which provides a very accurate targeting precision and greater control of movement is the **SpacePoint Fusion** [Cor10] developed by *PNI Sensor Corporation*, illustrated in Figure 3.10. With the use of a magnetometer, a gyroscope and an accelerometer, all of three axes it self-calibrates and maintains pinpoint accuracy, thus allowing a better immersion experience.

Also, by combining these new devices with stereoscopy, it is possible to make a multi-modal immersive interaction with direct and natural manipulation of objects shapes within virtual environments. On the other hand, the way users navigate and interact with objects in an immersive environment and interact with it, is still an open issue. Norman[Nor10] stated that gesturing is a natural, automatic behaviour, but the unintended interpretations of gestures can create undesirable states. Having this in mind, it is important to aim for an interface that is both predictable and easy to learn. In our current protoype we used as Wii Remote and the Spacepoint Fusion, which were tested through various metaphors for navigation and interaction.

[2]Be Your Own Souvenir. http://www.blablablab.org/ accessed on 6/10/2011

3.5 Summary

As seen, most of the search engines only present the results as a list of thumbnails, which hinders the interpretation of query results of 3D objects. Also, looking back to 3D MARS, we realize it was a valid idea that fell almost into obliviousness. The 3DMARS [NH01a] was an immersive virtual reality (VR) environment to perform image retrieval.

Following the idea of using a VR environment it was required to identify mean to visualize and navigate in such environments. Considering the latest low-cost, off-the-shelf hardware for visualization and interaction, we then extended our study to post-WIMP systems. Generally, post-WIMP approaches abandoned the traditional mouse and keyboard combination, favouring devices with six degrees of freedom (DoF). Unlike traditional WIMP interaction style, it allow straightforward direct mapping between device movements and rotations with corresponding effects on the three-dimensional space.

As such, by merging each of the major topics discussed in this Chapter, we intend to create an immersive VR system for 3D object retrieval (**Im-O-Ret**) which will extend benefits of VR environments and post-WIMP approach, to the 3D object retrieval.

Chapter 4

Immersive 3D Object Retrieval

In this Chapter we propose an immersive VR system for 3D object retrieval, (**Im-O-Ret**), in order to overcome the limitations of the current 3DOR search engines. We aimed to make use of display devices for viewing and interaction of our results in immersive virtual reality environment, and existing 3DOR techniques to implement our retrieval system, not to create or enhance new methods for retrieval of 3D objects. For the end result we also aim to make our prototype accessible to anyone. Therefore, we used commercial devices for both viewing and for interaction.

4.1 Overview

With our work, we wanted a system that would allow the user to browse, navigation and manipulate, query results of a search in an immersive VR environment. Based on this propose, we conceptualized our architecture to fulfil a the set of technical requirements. As such, our system was divided into two major modules, the **3D!** (**3D!**) module and the Object Retrieval (OR) module.

The OR module, would be responsible for indexing and retrieving 3D models. This should ultimately receive a model, and return a list of similar models ordered by the degree of similarity between these and the query model. Also, similar to what is done in the 3D MARS [NH01a, NH01b], were for each coordinate axis is pre-defined set of features, we intend to assign an different 3DOR algorithm for each axis. For the propose of studying and comparing different retrieval methods, it should be easy to integrate or replace these algorithms.

In the **3D!** module, we would create the VE were we display the results of a search. Since we aimed to make our prototype accessible to anyone, thus using of a set of different commercial devices for both the visualization and the interaction, it is required that our system is both modular and scalable, in order for an easy integration of new devices. As such, we divided this module in three sub-modules: **Core**, **View**, and **Controller**. For design architecture of our system we based it on the standard *Model-View-*

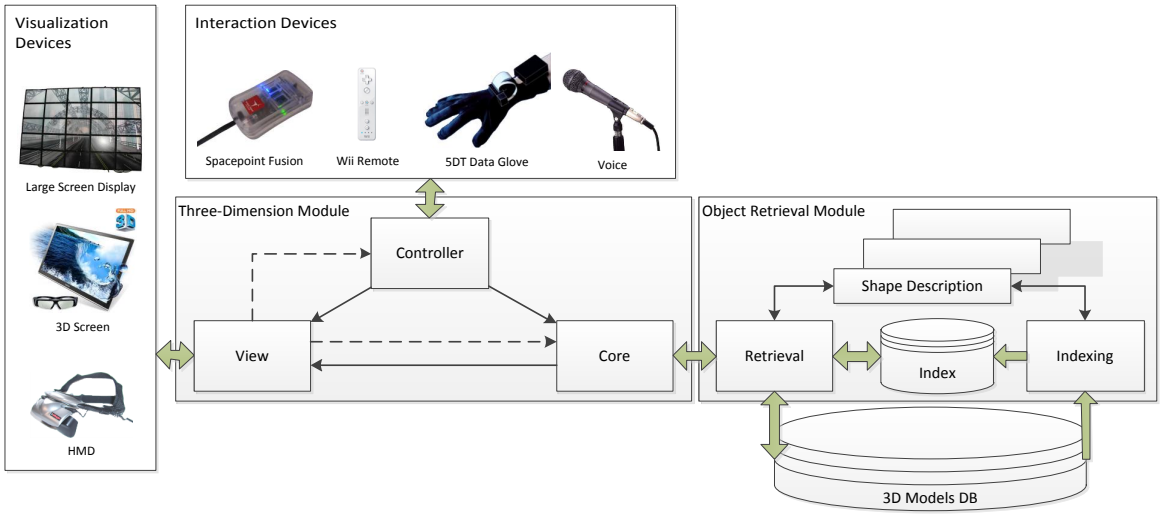


Figure 4.1: Architecture overview

Controller[1], as depicted in Figure 4.1. The MVC was originally described by Trygve Reenskaug in 1979, and consists of three components: *Model*, *View* and *Controller*.

The module **Core** is responsible for managing the entire logic of the system. It is where we implemented all the tasks of management of the objects. It will also handle the communication with the OR module for retrieval tasks. In the **View** we design and represent the objects. It is responsible for the presentation of the data handled by *Model*. Lastly, the **Controller**, would manage the inputs entered by the user and the changes that are needed as a result in the **Model** and **View** modules. It should also be responsible for receiving inputs from different devices to interact and map them to the system.

4.2 Architecture

For the implementation of our system we required a framework that would provide easy usage of multiple input devices. As such, we used the OpenIVI framework [AGFJ09], which provides a flexible event loop management to fit any VR/MR/AR setup and to support multi-input configurations. The OpenIVI framework is an Open Source Initiative launched by INESC-ID to support the fast prototyping of new applications with advanced 3D visualization techniques combined with innovative interaction techniques.

The OpenIVI framework relies on both OpenSG and OpenTracker technologies to provide an integrated solution with a powerful 3D scenegraph and support for tracking devices. Thanks to flexible XML based configuration files, it enables customized prototyping and fast integration mechanisms with a complete separation of inputs, interaction and functionality. As such, it enables the usage of customized visualization devices such as Head Mounted Displays, Tiled display systems or traditional Desktop monitors.

[1]http://heim.ifi.uio.no/ trygver/themes/mvc/mvc-index.html, accessed on 6/10/2011

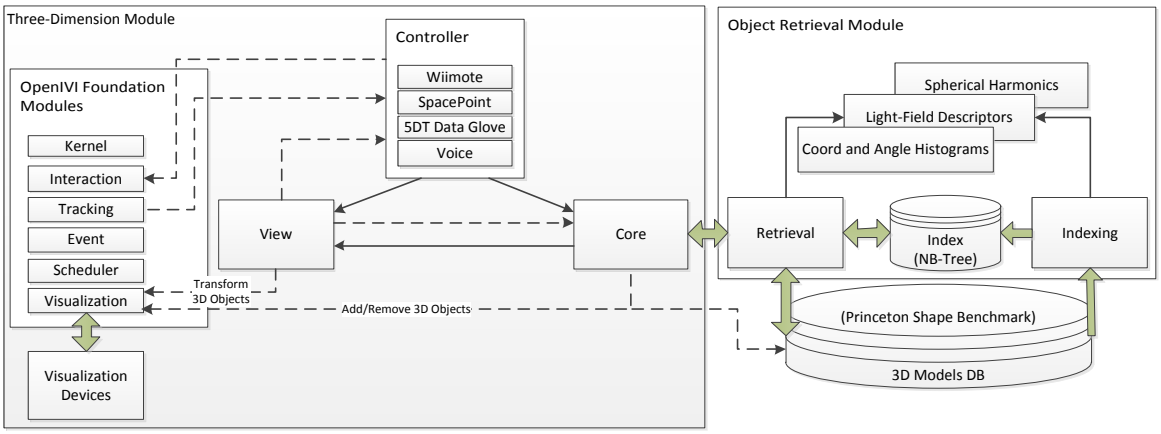


Figure 4.2: Architecture

The architecture of our prototype integrated with the OpenIVI is depicted in Figure 4.2.

4.2.1 Object Retrieval Module

The OR module, responsible for indexing and retrieving 3D models. For the models we used the Princeton Shape Benchmark [SMKF04] model database. This benchmark contains a collection of 1,814 polygonal models, all collected from various sources in the World Wide Web. Each model, has a JPEG image file with a thumbnail view of the model, which we used for visualization in our traditional system. This traditional system was created for the purpose of compare our proposal with current traditional systems. A more detailed description of this system is available in the Appendix A.

For matching algorithms we used the Light-Field Descriptors [CTtSO03] (LFD), the Cord and Angle Histogram [PR97a] (CAH), the Spherical Harmonics Descriptor [KFR03] (SHA). The executables of the LFD and CAH, we implemented both in C++, based on the description provided by their respective authors. For the LFD, we also used OpenGL, for the drawing of the 2D silhouettes. For the executable of the SHA, was used a created by one of the authors[2]. A detailed explanation of these algorithms was presented in Chapter 2.

We used these shape matching algorithms for three reasons. First, because each targets a different set of features [TV08], as we described in Chapter 2. Also, by using shape descriptors from different categories we can increase the diversity of query results. With the combined used of different matching methods, we aimed to obtain a best overall performance.

[2]Misha Kazhdan's Home Page. http://www.cs.jhu.edu/~misha/, accessed on 6/10/2011

Using each of the three shape descriptors, we extracted the feature vectors from all models. The extracted feature vectors extracted from each shape descriptor, were then indexed into three separated NB-Trees[FJ03]. Each indexed feature vector, would have a corresponding identifier, in order to match to the corresponding model. When querying, we extract the feature vectors of the query model, for each of shape descriptors. Then using the query feature vectors, in the corresponding NB-Tree, we receive the identifiers corresponding to the model which are more similar to the query model. For querying the NB-Tree we used its k-Nearest Neighbours query.

4.2.2 Three-Dimension Module

The 3D module, will use the OpenIVI framework in order to create the VE for the display of the query results. Since OpenIVI already enables the usage of customized visualization devices, we used it's visualization module for the management of the visualization device. For the interaction devices, we implemented our modules, using the tracking module only for the management of the NaturalPoint TrackIR cameras and the interaction module for the process of input events of the keyboard and mouse.

The module **Core**, which manages all the system logic will handle all the communication with the OR module for retrieval tasks. In Table 4.1, are depicted the arguments required in the configuration file for this module, and in Figure 4.3 our XML configuration file. In the current version of **Im-O-Ret** we assigned, the following matching algorithms: the Light-Field Descriptors [CTtSO03] on the X-axis; the Cord and Angle Histogram [PR97a] for the Y-axis; the Spherical Harmonics Descriptor [KFR03] for the Z-axis.

<IORCoreConfig>	
<Models/>	Information of models.
mode =	The arrangement of the models in the 3D Space.
location =	Directory of models.
<Searcher/>	Information of shape matching algorithm.
id =	Parameter identifier.
axis =	The axis which the algorithm is assigned.
location =	Directory of the executable.
search =	The executable file.
result =	The result file of a search.
</IORCoreConfig>	

Table 4.1: Specification of the XML configuration file - *Core* Module

```
▼<IORCoreModule>
  ▼<IORCoreConfig>
     <Models mode="0" location="../data/models/"/>
     <Searcher id="lightfields" axis="x" location="../search/LFD/" search="LFD-Retrieval.exe" result="search.res"/>
     <Searcher id="sharmonics" axis="z" location="../search/SH/" search="SH-Retrieval.exe" result="search.res"/>
     <Searcher id="cahistograms" axis="y" location="../search/CAH/" search="CAH-Retrieval.exe" result="search.res"/>
   </IORCoreConfig>
 </IORCoreModule>
```

Figure 4.3: XML configuration file - *Core* Module

In the event of a search the **Core** module will communicate with the 3D module, receiving as result, a list of objects similar to the query. Then the **Core** module will update his list of present objects. For new objects, it will add the object to the display and issue an order to scale it, from 0% to 100% of his dimension, to the **View** module. If the object already exists on the list, it will issue an order to the **View** module, to move the object to a new position in the 3D space. For object that are displayed, but are not present in the new result list, it will issue an order to scale it, from 100% to 0% of his dimension, to the **View** module. When the dimension of the object reaches 0%, it is removed from the display.

The purpose of the scale and relocation of models being done gradually and periodically, until reaching the goal, is to ease the load time required for the addition of all the objects in the list of result. By adding each, model separately, and adding animation to each added model, we lessen the system load and captures the user attention.

The management of input devices will be handled by the *Controller* module. In Table 4.2, is depicted the specification for this module configuration, and in Figure 4.2 our corresponding XML configuration file. Each sub-module is identified by a name identifier, and the name of the library corresponding to the DLL filename without the extension. For example the library name "WiiModule" will correspond to the WiiModule.dll file.

<IORControllerConfig>	
<SubModule/>	Information of Sub-modules.
id =	Parameter identifier.
library =	The library filename.
</IORControllerConfig>	

Table 4.2: Specification of the XML configuration file - *Controller* Module

```
▼<IORControllerModule>
  ▼<IORControllerConfig>
     <SubModule id="OTWii" model="WiiModule"/>
     <SubModule id="OTSpacepoint" model="SpacepointModule"/>
     <SubModule id="OTGlove" model="GloveModule"/>
   </IORControllerConfig>
 </IORControllerModule>
```

Figure 4.4: XML configuration file - *Controller* Module

Each sub-module will interact with existing modules and implement a predefined behaviour or functionality. This implementation, allows easy to extension and creation of new sub-module with specific interactive behaviour. With minimal effort is possible to have the system working in a context using other input devices, such as the Kinect.

In the occurrence of an event in a sub-module, it is published by the **Controller** module in the event sink. The published event will contain the action to be performed, and which module should handle the

corresponding action. For example, the voice command used to start a new search, will be addressed to the *Core* module, which communicates with the OR module, while the grabbing and direct manipulation of an object will be handled by the *View* module. The **View** module will, not only, handle the presentation of the data handled by *Model*, but also the transformations to the objects position, scale and rotation.

4.3 System features

Taking into consideration the requirements described above, we built a prototype for 3D model search where query results are presented in a 3D virtual space. In our prototype, (**Im-O-Ret**), the results of a query are displayed and distributed in the three-dimensional space as 3D objects, instead of the traditional list of thumbnails. As such, the user can explore the results, navigate and manipulate the scattered objects in the three-dimensional space.

For the interaction in our system we used a post-WIMP interface, using devices with six DoF, which afford direct object manipulation. We complemented the direct manipulation, using voice commands for the selection of actions, providing a more natural and free interaction to the user.

Finally, thanks to our modular architecture, we are able to combine different sets of devices, and easily add new devices to our system. With the combined use of virtual environments and new of the self devices, we provide a more complete visualization of models and enable a more natural interaction.

Figure 4.5: Im-O-Ret: Using a commercial tv and the wiimote

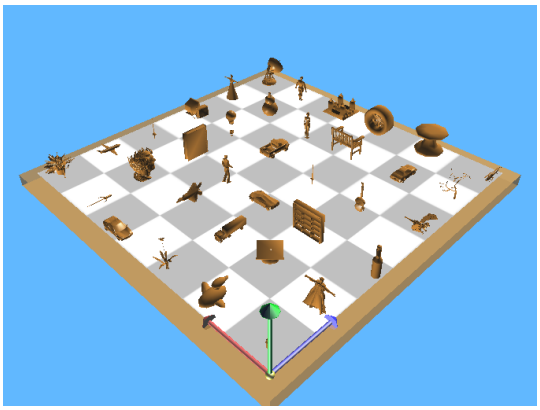


Figure 4.6: Im-O-Ret: Query specification

4.3.1 Query Specification

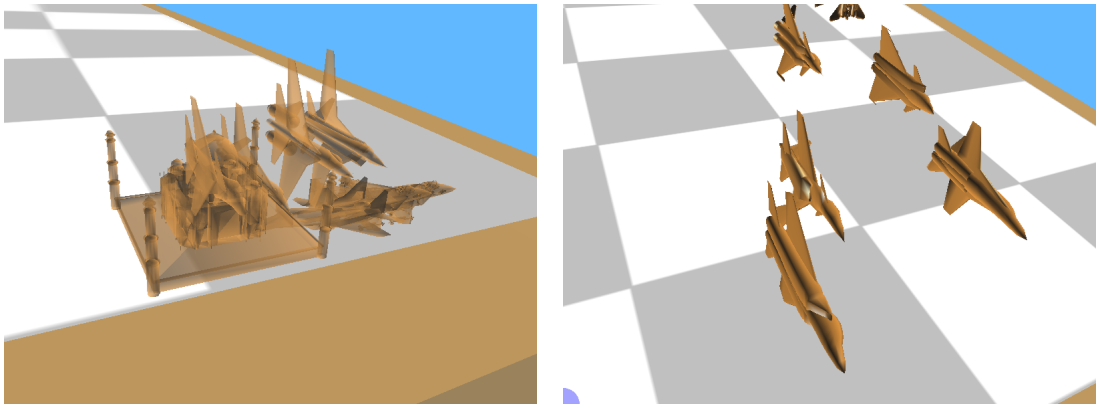
Since query specification was outside the scope of this thesis, little work was done regarding it. As such, for query specification, only a simple query-by-example was implemented. When starting a search, the user is presented with 36 models, randomly selected, that are placed in the plane xOz, equally distant from one another, as depicted in Figure 4.6.

To perform a search the user selects one of the initial presented models, for a search for similar objects. Then, the user can either perform a new search, in which the system will present a new set of 36 models or another search using one of the query result models as query.

4.3.2 Spacial Distribution of Results

The query results are distributed in the virtual 3D space according to their similarity. To each axis it can be assigned a different shape matching algorithm. The coordinate value is determined by the similarity to the query given by the corresponding algorithm. As such, when performing a search, the query model is used to find similar models, using each algorithm. The results are then merged, giving a 3D position for each similar model retrieved. The query mechanism can be adapted to specific domains, producing more precise results.

There are two modes for the distribution the 3D objects. In the first, the objects are distributed in the space with equal distance from each other. This allows to view the objects individually without occlusion. In this mode, although the more similar are closer to the origin of the 3D space, there is no information of the different of similarity between two objects. In the second mode, the objects position is the ground-truth value retrieved from each algorithm. This cause the creation of clusters, when the retrieved values of different models are very similar, which provides a better study and analyse of algorithms and query-results.



(a) equally-distant mode. (b) ground truth result mode.

Figure 4.7: Im-O-Ret: Spacial distribution modes.

4.3.3 Exploration of Query Results

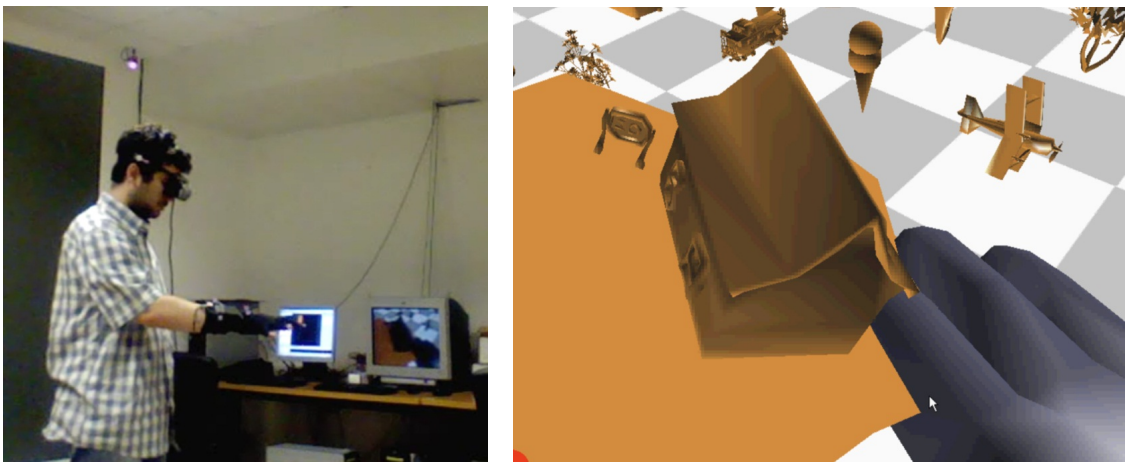
In order to view the scattered models of a search result, the user can explore the results, by navigating in that space and directly manipulating the objects. For the navigation we considered the division proposed by Bowman et al. [BDHB99], were he divides the navigation into two types based on their movement. As such we created both a *travel* and a *wayfinding*.

For the *travel* it consists in exploratory movements were the viewpoint is moved from one location to another, by using either the arrows in the keyboard, the wiimote nunchuck, or tracking tools, similar to the process of walking. The *wayfinding* was described by using points of interest, where the movement is done by selecting an object that specifies the position where the camera is moved to. For the *wayfinding*, we used a combination of pointing and voice commands.

In Figure 4.8 is illustrated the use of tracking tools for the exploration of the 3D world. This interaction does not require neither much teaching or previous knowledge in order to use. Also, the use of shutter glasses, provides the user with a stereoscopic view of the world, which enables a more complete immersive experience. As seen, the use of stereoscopic view and multi-modal ways of interaction, provide a visualization of query results that goes far beyond the traditional scrolling on a list of thumbnails. In this interaction scenarios, the user literally navigates among the results in the three-dimensional space.

4.3.4 Multimodal Interaction

The use of voice commands, complements the pointing in describing arguments and selecting actions. To effectively use the information from user input through words, direct manipulation helps users know which concept to cover their actions. For instance, to use an object as a query, user can point to it and say 'search similar to this', thus triggering a new query. The combined use of virtual environments, devices with six DoF and voice commands provides effective result visualization and makes interaction natural, comprehensible and predictable.



(a) exploring the 3D space with HMD and data gloves.

(b) user's view.

Figure 4.8: Im-O-Ret: using the HMD and 5DT Data Gloves

For each action there are some voice commands with few small variations. Most of this variations consist in the order or addition of some words, which gives the user less need of learning the commands. In Appendix C is depicted the full list of voice commands available in our system, as well as, an explanation of the context where each is used.

The proposed modular architecture, makes the system able to be configured to work with a wide range of different interaction devices and displays. Combining different visualization and interaction devices, allows us to create multiple interaction paradigms for our system. With minimal effort is possible to have the system working in a context using other input devices, such as the Kinect. This way we could combine different visualization and interaction devices, and create multiple interaction paradigms for our system.

4.4 Summary

In this Chapter, we have presented the solution we developed in order to test our approach. We started by identifying the main requirement that we needed to take into account in order to build such system. Then, following these requirements, we discussed how we created our modular architecture. In it, we showed how we are able to combine different sets of devices, and easily add new devices to our system. With the combined use of virtual environments and new of the self devices, we provide a more complete visualization of models and enable a more natural interaction. Finally, we described our prototype for 3D object retrieval, the **Im-O-Ret**. Has seen, it uses a post-WIMP interface to interact with the query-result that are distributed in the three-dimensional space. Also, using voice commands for action selection, and direct manipulation for spacial reference, we provide a more natural and free interaction to the user. In next Chapter, we validate our approach by using a set of user tests.

Chapter 5

Evaluation

In order to validate our proposal, it was necessary to conduct a set of evaluation tests. The tests were structured into three stages: a set of preliminary tests, conducted by a small group of users which followed the project development; three search task, using our prototype and a traditional system; and, finally, using a set of different interaction paradigms, followed by a post-questionnaire to rate the prototype's ease of usage with each of the scenarios.

All tests were conducted in a closed environment. The interaction paradigms evaluation, using various scenarios with different visualization and interaction devices, was conducted in laboratory *João Lourenço Fernandes*[1] on the Taguspark campus. This laboratory consists on a large multimedia room, with a large screen, the LEMe wall [AGC+05], and tracking system with ten NaturalPoint TrackIR cameras. Beside the LEMe Wall and the tracking tools, we also used a HMD, the Z800 3DVisor, and a SpacePoint device. Both this devices are shown and discussed at Chapter 3. Figure 5.1 illustrates the HMD setup used in the interaction paradigms evaluation.

Figure 5.1: HMD setup prepared for final user testing.

[1] http://lab-jlf.ist.utl.pt, accessed on 6/10/2011

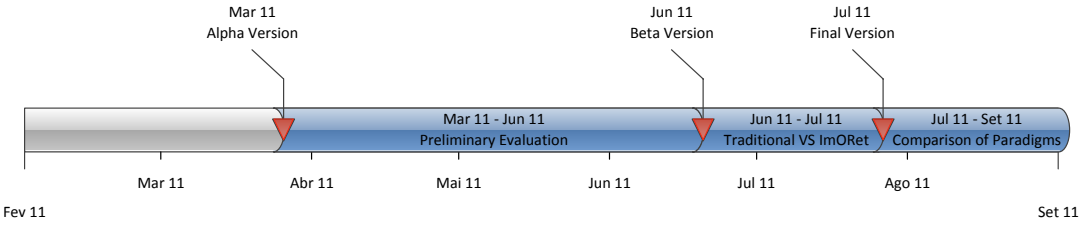


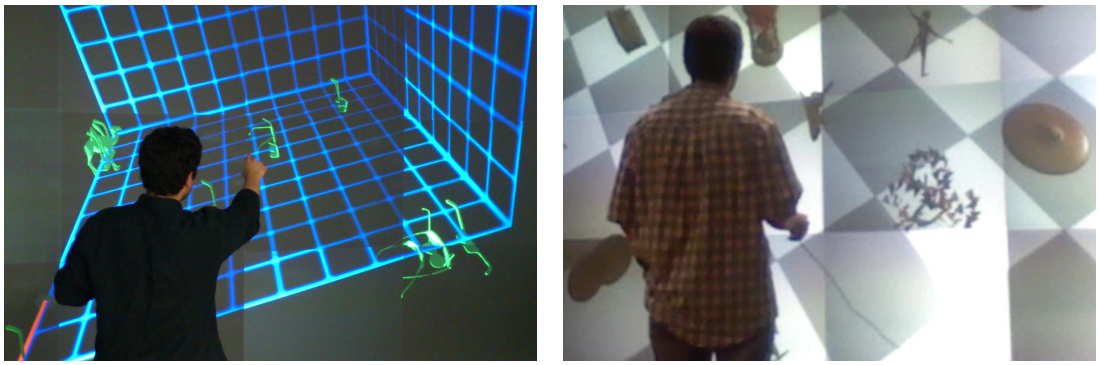
Figure 5.2: The time-line in which each test was conducted.

5.1 Preliminary Evaluation

The first stage of test was composed by informal tests conducted during the development of the prototype. It was carried out using a panel of six users, all of them with knowledge on both retrieval and virtual environments. This test where mainly focused in two main features: interface and interaction.

In respect of interaction it was tested the ease of usage of devices and voice commands. While for the devices we focused on the usage of direct manipulation, the selection of words for the voices commands required more accuracy. As such, it was tested the percentage of success for each of the voice command. Only when the voice command was able to achieve a percentage of 70% of success was the voice command accepted. For this, we constantly changed the words of the voice commands, in order to find those that would be easier for use, taking into consideration the advices and recommendations given by the participants.

For the interface, we tested not only the usage of color, but also the adding of effects, such as glow or special colours, to the results, order to convey additional information. For instance, in the event of selecting an object, the selected object will have its colour inverted. Also, the application of transparency to results, in the cluster distribution mode, making it possible to view overlaid objects.
Thanks to this preliminary evaluation, it we were able to constantly analyse and refine each of our solution



(a) Preliminary version.

(b) Final version.

Figure 5.3: Im-O-Ret using the LEMe Wall and SpacePoint Fusion.

versions. In Figure 5.3, we show an example of an early version and the current final version, using both the LEMe Wall and the SpacePoint.

5.2 Traditional vs Im-O-Ret

In order to validate our system it was required to compare it with the traditional search engines. As such, we created a web-page where users could perform a traditional 3DOR search, where the query result are presented as a list of thumbnails. We named this system Thumbnail Object Retrieval (**THOR**). A detailed description of this system is presented in Appendix A.

5.2.1 Test Description

In this test, the participants would perform three searches of increasing difficulty using **THOR**. The same three searches, would then be perform using **Im-O-Ret**, as depicted in Figure 5.4. In order to make the starting point in both systems be the same, we used a static group of 36 models as queries for a new search. These 36 models are presented in Appendix B.

We counted with twelve participants, from which we noted the number of steps, errors, and time required to find a specific object. We considered as an error each time the user either rolled back to a previous search result, or re-started the search. For this test, since we intended to only test the advantages of using the 3D models instead of thumbnails. As such, we used a simple computer screen with the mouse as pointing device. This test was followed by a post-questionnaire to rate the prototype's usability.

(a) Using THOR (b) Using Im-O-Ret

Figure 5.4: Search Efficiency Evaluation: Test Environment

5.2.2 Results and Discussion

This evaluation allowed us to compare our approach with a traditional 3DOR system. In Figure 5.8a we present a comparison of the number of steps taken from both THOR and Im-O-Ret. In the first search task, there was no meaningful difference in the number of steps. However, the more challenging the search task, the fewer the steps using the Im-O-Ret in comparison with THOR.

The same observation can be seen in the number of error, depicted in Figure 5.8b. The average of both systems had very close values, for the first search task. However, for the more complex second and third search tasks, our approach required less steps and did errors, when comparing with THOR. These results, allow us to conclude that Im-O-Ret provides better visualization and interpretation of the query results.

However, the same cannot be said of the time required to perform the search tasks. The average of time required, depicted in Figure 5.7, shows that the participants wasted more time in the tasks performed using the Im-O-Ret. During the test we observed two main causes. Since used it used a 3D environment, the users wasted more time navigating and viewing the query results, instead of performing new searches. The second cause, was the spatial distribution of the 3D objects. Since we presented the 36 query models

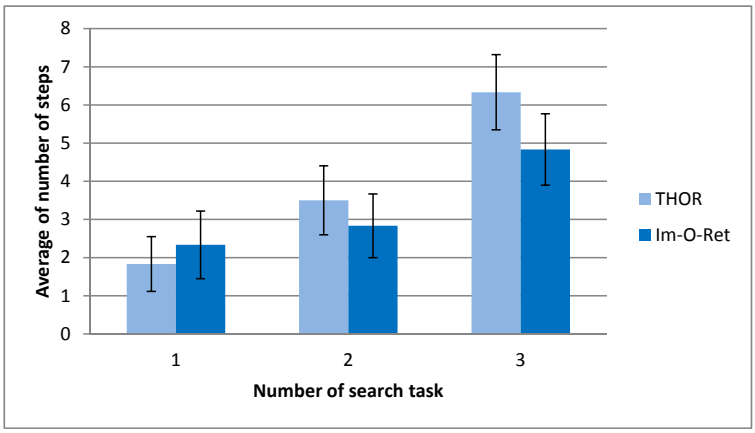


Figure 5.5: Average number of steps for each task.

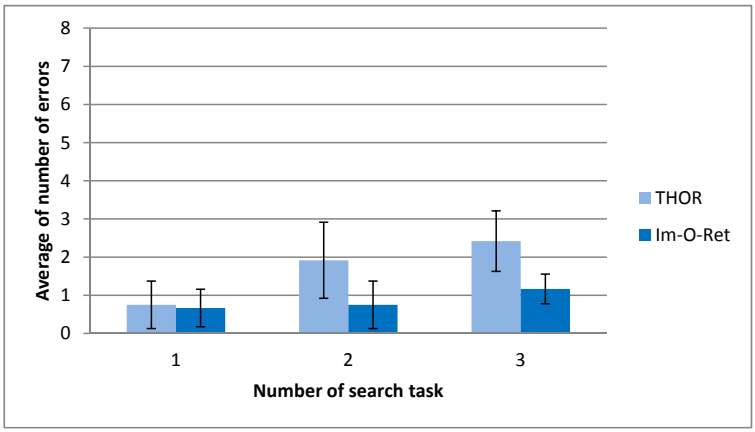


Figure 5.6: Average number of errors for each task.

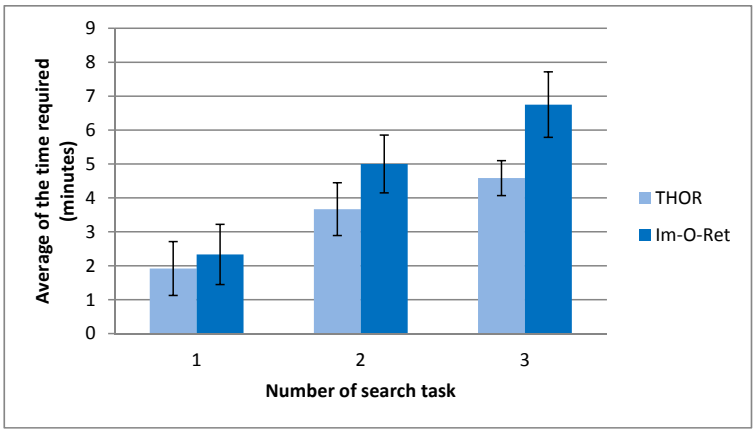
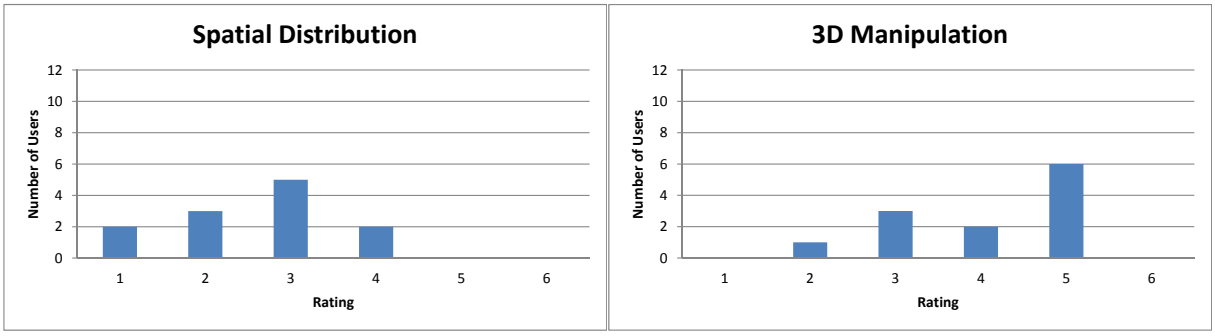


Figure 5.7: The average time required, in minutes, for each task.

in a board, most users expected the models to be scattered only in the board, and did not like different height to each object, caused by the shape descriptor assigned to the Y-axis. Since our systems is easily configurable, we removed the shape descriptor of the Y-axis, making the objects being of scattered only using the shape descriptors assigned to the X-axis and Z-axis. After some quick preliminary test the times required to perform search greatly improved. Our analyse, can be matched with the information retrieved from the post-questionnaire to rating, depicted in Figure 5.8.

5.3 Comparison of Paradigms for Result Exploration

In this evaluation, we tested the usability of our system, using distinct interaction scenarios with off-the-shelf devices, for both visualization and interaction. These would allows us to distinguished the benefits and advantages of each device in our prototype. This test were followed by a post-questionnaire to rate the prototype's ease of usage in each of the interaction scenarios. We used four different interaction paradigms.



(a) Spacial distribution.

(b) Manipulation of 3D objects.

Figure 5.8: Chart illustrating the user classification of Im-O-Ret features. (1:Bad; 6:Good)

5.3.1 Test Description

For the first scenario we used a computer screen with the mouse as pointing device. The second scenario, was performed using a commercial TV screen and a Wiimote. As third scenario, we used a large-screen display, the LEMe Wall [AGC+05], with a six DoF interaction device, the SpacePoint Fusion. Finally, for the fourth scenario used a HMD, the *Z800 3DVisor*, the 5DT Data Glove, and the tracking system with ten NaturalPoint TrackIR cameras, for the tracking of the head and hands positions.

This evaluation was conducted in laboratory *João Lourenço Fernandes*[2] on the Taguspark campus, and counted with the participation of ten users. The users were asked to conduct simple searches and then navigate and manipulate the query results. In the end, we asked the participants to fill answered a questionnaire that aimed to observe the ease of using each of this interaction paradigms.

5.3.2 Result and Discussion

This evaluation allowed us to gauge the ease of learning and using different interaction paradigms with our prototype. From the questionnaires we observed that for the easiest interaction paradigm to use, the participant were divided between Screen + Mouse and the LEMe Wall + Spacepoint (Figure 5.9). In the case of the Screen + Mouse, we concluded that is due being an interaction in the daily use of the computer, and most participants already being used to such interface.

The selection of the LEMe Wall + Spacepoint, was mainly thanks to the motion-tracking system of the SpacePoint device which auto-calibrates and maintains a precise accuracy of the location pointed by the user. The fact that the object were displayed in a large screen, also provided the user with a better view of the query result, without need of navigating to a closer position of the object. The usability rating of this interaction scenario is depicted in Figure 5.12.

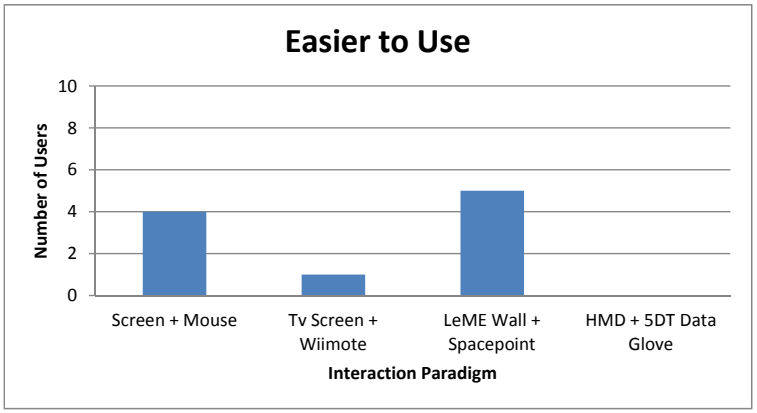


Figure 5.9: Chart illustrating the user selection as the of the easiest device to use.

[2]http://lab-jlf.ist.utl.pt, accessed on 6/10/2011

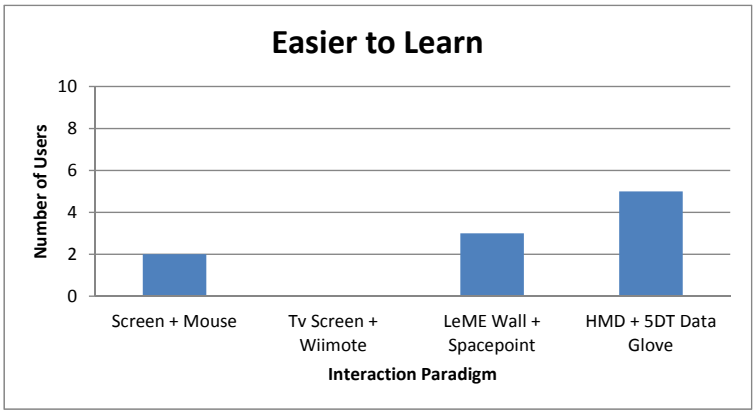


Figure 5.10: Chart illustrating the user selection as the of the easiest device to learn.

For the easiest interaction paradigm to learn, most participants chose the HMD + 5DT Data Glove and a smaller number divided themselves between the use Screen + Mouse and the LEMe Wall + Spacepoint.

The fact that the HMD + 5DT Data Glove takes advantage of the user's body movements, such as walking and rotating the head to set the virtual camera position, as well as the use of simple gestures to manipulate the objects in the 3D world, made it very easy to learn. However, it was pointed out by the participants, that the great number of wires used by these devices hindered their movement and interaction, and forcing the constant awareness in order to avoiding the wires, which justify the result depicted in Figure 5.11 regarding its usability.

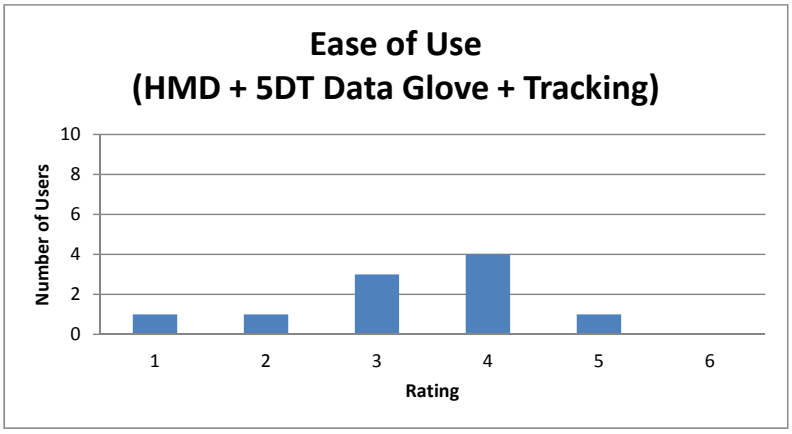


Figure 5.11: Chart illustrating the user classification of the ease of using the HMD with the 5DT data glove.

The usability rating of TV Screen + Wiimote, depicted in Figure 5.13, faded in comparison with the LEMe Wall + Spacepoint. There were two main reason for this. First, the LEMe Wall provided a more wide view of the virtual world than the TV Screen. Second, the Wiimote use of the accelerometer and gyroscope, is limited in comparison to the SpacePoint who additionally has a magnetometer, that provides a more direct mapping.

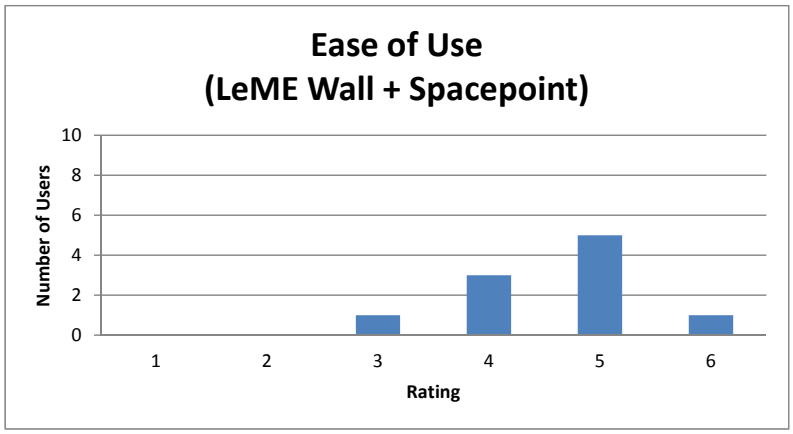


Figure 5.12: Chart illustrating the user classification of the ease of using the LEMe Wall with the SpacePoint Fusion.

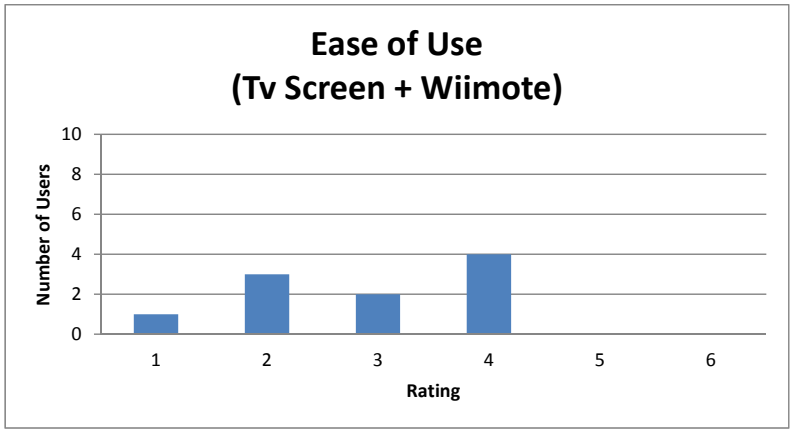


Figure 5.13: Chart illustrating the user classification of the ease of using a commercial TV with the Wiimote.

5.4 Summary

On a first stage we develop our proposal, performing along side a set of preliminary test that helped to improve our prototype's interaction. The average of both interaction paradigms had very close values, in the first search, although for the more complex second and third searches, our system had less steps and errors, but required a little more time to perform the task. This was due the user wasting more time navigating and viewing the results.

Then, on a second stage, we evaluated a traditional 3D search engine against our 3D retrieval approach. The results gathered allow us to conclude that Im-O-Ret provides better visualization and interpretation of the query results. However it requires more time to perform search tasks, mainly due to overhead of navigation through the results.

Finally, we tested the usability of our system using different devices for visualization and interaction, followed by a post-questionnaire to rate the prototype's ease of usage with each of the scenarios. This allowed us to observe the benefits and advantages of using each device in the context of 3DOR in immersive environments.

With the evaluation, we have validated our approach, although there still is a way to go until we can clearly surpass the traditional 3DOR systems.

Chapter 6

Conclusions and Future Work

This thesis was executed fairly well, reaching the objectives that were planned. We started by identifying that the use of lists of thumbnails for the query result visualization in current 3D search engines, greatly hinders the interpretation of query results on collections of 3D objects. As such, we proposed a novel approach for the query result visualization of 3D object retrieval. In this context we studied 3D object retrieval, virtual environments, and post-WIMP interaction. Taking advantage of each fields benefits, we developed a system which merges 3D object retrieval techniques with virtual reality environments, using new off-the-shelf devices.

The Im-O-Ret, presents the query results as 3D objects, instead of the list of thumbnails presented in current traditional systems. The query results are distributed in a 3D space according to their similarity. Each coordinate value will be determined by an assigned shape matching algorithm to that axis. These matching algorithms can be replaced, in order to produce more precise results for specific domains.

The Im-O-Ret uses a post-WIMP interface, which can use a set of different interaction devices and displays. For instance, Im-O-Ret supports combining hand gesturing and speech commands to provide a more natural interaction. From a practical point-of-view, thanks to our modular architecture, it is possible to add new input devices with minimal effort.

To validate our solution we conducted a set of user tests, where a panel of users, evaluated a traditional 3D search engine against our 3D retrieval approach. The results gathered allow us to conclude that Im-O-Ret can potentially provide better visualization and interpretation of the query results. However it requires more time to perform search tasks. This is mainly due to overhead of navigation through the results. Nevertheless, thanks to the better visualization and interpretation of the query results provided, with the Im-O-Ret the search tasks are done in less steps and with fewer errors. In a final evaluation we tested the usability of our system with multiple interaction scenarios, using different sets of visualization and interaction devices.

With the validation of our approach, we raises new challenges for 3DOR using immersive virtual environments. For instance, the usage of different shape matching algorithms can generate different results, specific to certain domains. Also, the spacial organization of results in our approach, was implemented as a simple solution by assigning a different shape descriptor to each axis. However, we concluded that such solution not always provide a distribution of query results that is meaningful to the user, being required more research in this area.

In the navigation, we observed that the use of a simple *wayfinding* technique, was limited and allowed little control. The Navidget technique [HDKG08, KHG08], which extends the techniques by points of interest, and offers the possibility to control the distance to the target as well as direction for its visualization, illustrates an example that would improve the exploration in our system.

Another major challenge that should be tackled is the query specification. For our prototype only a simple query-by-example was implemented, but similar to the Princeton 3D model search engine [FMK+03], with the 2D and 3D sketches, more natural query interfaces are required. For instance, the integration of gesture-based query specification, such as the one proposed by Holz and Wilson [HW11]. There is zlso, a multitouch version of ShapeShop [LMAJ11], which creates 3D shapes using bi-manual gestures to navigate and a pen to sketch.

These wide range of different features could provide a better visualization, navigation and interaction to our prototype, which alone, has given a huge step regarding 3DOR in immersive environments. With the knowledge and achievements accumulated in our work, we hope that it may be a starting point for future research on these subjects.

Bibliography

[AB92] Steve Aukstakalnis and David Blatner. *Silicon Mirage; The Art and Science of Virtual Reality*. Peachpit Press, Berkeley, CA, USA, 1992.

[AGC+05] Bruno Araujo, Tiago Guerreiro, Ricardo Costa, Joaquim Jorge, and Joao M. Pereira. Leme wall: Desenvolvendo um sistema de multi-projecção. 13o Encontro Portugues de Computação Grafica, Vila Real, Portugal, 2005.

[AGFJ09] Bruno R. De Araujo, Tiago Guerreiro, Manuel J. Fonseca, and Joaquim A. Jorge. Openivi. http://open5.sourceforge.net/, 2009.

[AVD07] Tarik Filali Ansary, Jean-Phillipe Vandeborre, and Mohamed Daoudi. 3d-model search engine from photos. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, CIVR '07, pages 89–92, New York, NY, USA, 2007. ACM.

[BBKF97] Ravin Balakrishnan, Thomas Baudel, Gordon Kurtenbach, and George Fitzmaurice. The rockin'mouse: integral 3d manipulation on a plane. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '97, pages 311–318, New York, NY, USA, 1997. ACM.

[BDHB99] Doug A. Bowman, Elizabeth T. Davis, Larry F. Hodges, and Albert N. Badre. Maintaining spatial orientation during travel in an immersive virtual environment. In *Presence: Teleoperators and Virtual Environments*, pages 618–631, 1999.

[Ben75] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18:509–517, September 1975.

[Bil98] Mark Billinghurst. Put that where? voice and gesture at the graphics interface. *SIGGRAPH Comput. Graph.*, 32:60–63, November 1998.

[BKS+05] Benjamin Bustos, Daniel A. Keim, Dietmar Saupe, Tobias Schreck, and Dejan V. Vranić. Feature-based similarity search in 3d object databases. *ACM Computing Surveys*, 37:2005, 2005.

[BMC+08] Benoît Le Bonhomme, B. Mustafa, Sasko Celakovsky, Marius Preda, Françoise J. Prêteux, and D. Davcev. On-line and open platform for 3d object retrieval. In *3DOR'08*, pages 73–79, 2008.

[Bol80] Richard A. Bolt. Put-that-there: Voice and gesture at the graphics interface. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '80, pages 262–270, New York, NY, USA, 1980. ACM.

[BP98] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web 7*, WWW7, pages 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[Cha07] Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions, 2007.

[CNSD93] Carolina Cruz-Neira, Daniel J. Sandin, and Thomas A. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the cave. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, pages 135–142, New York, NY, USA, 1993. ACM.

[Cor10] PNI Sensor Corporation. Spacepoint fusion. http://www.pnicorp.com/products/spacepoint-gaming, 2010.

[CTtSO03] Ding-Yun Chen, Xiao-Pei Tian, Yu te Shen, and Ming Ouhyoung. On visual similarity based 3d model retrieval. volume 22 of *EUROGRAPHICS 2003 Proceedings*, pages 223–232, 2003.

[DAA+11] Thomas DeFanti, Daniel Acevedo, Richard Ainsworth, Maxine Brown, Steven Cutchin, Gregory Dawe, Kai-Uwe Doerr, Andrew Johnson, Chris Knox, Robert Kooima, Falko Kuester, Jason Leigh, Lance Long, Peter Otto, Vid Petrovic, Kevin Ponto, Andrew Prudhomme, Ramesh Rao, Luc Renambot, Daniel Sandin, Jurgen Schulze, Larry Smarr, Madhu Srinivasan, Philip Weber, and Gregory Wickham. The future of the cave. *Central European Journal of Engineering*, 1:16–37, 2011. 10.2478/s13531-010-0002-5.

[DBD08] Jean-Luc Dugelay, Atilla Baskurt, and Mohamed Daoudi. *3D Object Processing: Compression, Indexing and Watermarking*. Wiley Publishing, 2008.

[DCG10] Helin Dutagaci, Chun Pan Cheung, and Afzal Godil. A benchmark for best view selection of 3d objects. In *Proceedings of the ACM workshop on 3D object retrieval*, 3DOR '10, pages 45–50, New York, NY, USA, 2010. ACM.

[DJLW08] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40:5:1–5:60, May 2008.

[Ent10] Sony Computer Entertainment. Playstation®move. http://us.playstation.com/ps3/playstation-move/product-information/, September 2010.

[FJ03] Manuel J. Fonseca and Joaquim A. Jorge. Nb-tree: An indexing structure for content-based retrieval in large databases. Technical report, Instituto Superior Técnico, 2003.

[FKMK98] Shinji Fukatsu, Yoshifumi Kitamura, Toshihiro Masaki, and Fumio Kishino. Intuitive control of "bird's eye" overview images for navigation in an enormous virtual environment. In *Proceedings of the ACM symposium on Virtual reality software and technology*, VRST '98, pages 67–76, New York, NY, USA, 1998. ACM.

[FKMS05] Thomas Funkhouser, Michael Kazhdan, Patrick Min, and Philip Shilane. Shape-based retrieval and analysis of 3d models. *Commun. ACM*, 48:58–64, June 2005.

[FMHR87] S. S. Fisher, M. McGreevy, J. Humphries, and W. Robinett. Virtual environment display system. In *Proceedings of the 1986 workshop on Interactive 3D graphics*, I3D '86, pages 77–87, New York, NY, USA, 1987. ACM.

[FMK+03] Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. A search engine for 3d models. *ACM Trans. Graph.*, 22:83–105, January 2003.

[FSZ03] David Feng, W.C. Siu, and Hong Jiang Zhang. *Multimedia Information Retrieval and Management: Technological Fundamentals and Applications Series*. Springer, 2003.

[Gut84] Antonin Guttman. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, SIGMOD '84, pages 47–57, New York, NY, USA, 1984. ACM.

[HDKG08] Martin Hachet, Fabrice Decle, Sebastian Knödel, and Pascal Guitton. Navidget for easy 3d camera positioning from 2d inputs, 2008. best paper award.

[HDKG09] Martin Hachet, Fabrice Decle, Sebastian Knödel, and Pascal Guitton. Navidget for 3d interaction: Camera positioning and further uses. *Int. J. Hum.-Comput. Stud.*, 67:225–236, March 2009.

[HW11] Christian Holz and Andrew Wilson. Data miming: inferring spatial object descriptions from human gesture. In *Proc. of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 811–820. ACM, 2011.

[IJL+05] Natraj Iyer, Subramaniam Jayanti, Kuiyang Lou, Yagnanarayanan Kalyanaraman, and Karthik Ramani. Three-dimensional shape searching: state-of-the-art review and future trends. *Comput. Aided Des.*, 37:509–530, April 2005.

[KFR03] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, SGP '03, pages 156–164, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[KHG08] Sebastian Knödel, Martin Hachet, and Pascal Guitton. Navidget for immersive virtual environments. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, VRST '08, pages 47–50, New York, NY, USA, 2008. ACM.

[KS94] David B. Koons and Carlton J. Sparrell. Iconic: speech and depictive gestures at the human-machine interface. In *Conference companion on Human factors in computing systems*, CHI '94, pages 453–454, New York, NY, USA, 1994. ACM.

[LBPP08] B. Le Bonhomme, M. Preda, and F. Preteux. Mymultimediaworld.com: A benchmark platform for 3d compression algorithms. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 2700 –2703, oct. 2008.

[Lee08] J.C. Lee. Hacking the nintendo wii remote. *Pervasive Computing, IEEE*, 7(3):39 –45, july-sept. 2008.

[LMAJ11] Pedro Lopes, Daniel Mendes, Bruno Araújo, and Joaquim A. Jorge. Combining bimanual manipulation and pen-based input for 3d modelling. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, SBIM '11, pages 15–22, New York, NY, USA, 2011. ACM.

[MCR90] Jock D. Mackinlay, Stuart K. Card, and George G. Robertson. Rapid controlled movement through a virtual 3d workspace. pages 171–176, 1990.

[Mic10] Microsoft. Kinect. http://www.xbox.com/en-US/kinect, November 2010.

[Min04] Mindflux. 5dt data glove ultra series users manual. http://www.5dt.com/, October 2004.

[MVM09] Frederic P. Miller, Agnes F. Vandome, and John McBrewster. *Bing (Search Engine): Steve Ballmer, Powerset (company), Windows Live SkyDrive, Facebook, E- mail, Yahoo!, Yahoo! Search, Windows Live, Google search, ... Zune HD, Web search engine, Microsoft.* Alpha Press, 2009.

[NH] Munehiro Nakazato and Thomas S. Huang. 3d mars. http://www.ifp.illinois.edu/~nakazato/3dmars/.

[NH01a] Munehiro Nakazato and Thomas S. Huang. 3d mars: Immersive virtual reality for content-based image retrieval. In *Proceedings of 2001 IEEE International Conference on Multimedia and Expo (ICME2001)*, 2001.

[NH01b] Munehiro Nakazato and Thomas S. Huang. An interactive 3d visualization for content-based image retrieval. In *http://www.ifp.illinois.edu/~nakazato/publications/PDF/icme2001ext.pdf*, 2001.

[Nin06] Nintendo. Wii remote. http://www.nintendo.com/wii/console/controllers, October 2006.

[Nor10] Donald A. Norman. Natural user interfaces are not natural. *interactions*, 17:6–10, May 2010.

[OFCD02] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21:807–832, 2002.

[PMN+00] E. Paquet, A. Murching, T. Naveen, A. Tabatabai, and M. Rioux. Description of shape information for 2-d and 3-d objects. *Signal Process Image Commun*, 16:103–122, September 2000.

[PR97a] E. Paquet and M. Rioux. Nefertiti: a query by content software for three-dimensional models databases management. In *Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, NRC '97, pages 345–, Washington, DC, USA, 1997. IEEE Computer Society.

[PR97b] E. Paquet and M. Rioux. Nefertiti: a query by content software for three-dimensional models databases management. In *3-D Digital Imaging and Modeling, 1997. Proceedings., International Conference on Recent Advances in*, pages 345 –352, may 1997.

[Shn97] Ben Shneiderman. Direct manipulation for comprehensible, predictable and controllable user interfaces. In *Proceedings of the 2nd international conference on Intelligent user interfaces*, IUI '97, pages 33–39, New York, NY, USA, 1997. ACM.

[SMKF04] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. The princeton shape benchmark. *Shape Modeling and Applications, International Conference on*, 0:167–178, 2004.

[SSDP+09] Beatriz Sousa Santos, Paulo Dias, Angela Pimentel, Jan-Willem Baggerman, Carlos Ferreira, Samuel Silva, and Joaquim Madeira. Head-mounted display versus desktop for 3d navigation in virtual reality: a user study. *Multimedia Tools Appl.*, 41:161–181, January 2009.

[SSG+03] H. Sundar, D. Silver, N. Gagvani, S. Dickinson, and D. Silver Ý. Skeleton based shape matching and retrieval, 2003.

[Sut68] Ivan E. Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, AFIPS '68 (Fall, part I), pages 757–764, New York, NY, USA, 1968. ACM.

[SYS03] M.T. Suzuki, Y. Yaginuma, and Y.Y. Sugimoto. A 3d model retrieval system for cellular phones. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 4, pages 3846 – 3851 vol.4, oct. 2003.

[SZP+00] Rajeev Sharma, Michael Zeller, Vladimir I. Pavlovic, Thomas S. Huang, Zion Lo, Stephen Chu, Yunxin Zhao, James C. Phillips, and Klaus Schulten. Speech/gesture interface to a visual-computing environment. *IEEE Comput. Graph. Appl.*, 20:29–37, March 2000.

[TRC01] Desney S. Tan, George G. Robertson, and Mary Czerwinski. Exploring 3d navigation: Combining speed-coupled flying with orbiting. pages 418–425, 2001.

[tSCTO03] Yu te Shen, Ding-Yun Chen, Xiao-Pei Tian, and Ming Ouhyoung. 3d model search engine based on lightfield descriptors. EUROGRAPHICS 2003 Proceedings, 2003.

[TV08] Johan W. Tangelder and Remco C. Veltkamp. A survey of content based 3d shape retrieval methods. *Multimedia Tools Appl.*, 39:441–471, September 2008.

[vD97] Andries van Dam. Post-wimp user interfaces. *Commun. ACM*, 40:63–67, February 1997.

[War] Google 3D Warehouse. http://sketchup.google.com/3dwarehouse/.

[Wu97] Jian-Kang Wu. Content-based indexing of multimedia databases. *Knowledge and Data Engineering, IEEE Transactions on*, 9(6):978 –989, nov/dec 1997.

[Zlo77] M. M. Zloof. Query-by-example: A data base language. *IBM Systems Journal*, 16(4):324 –343, 1977.

Appendix A

Traditional 3D model search system

As we described in previous Chapters, in order to validate our approach of using the 3D models instead of thumbnails, we developed a simple 3D search system. Because of the use of thumbnails, we named this system Thumbnail Object Retrieval (THOR). In this appendix, be give a more detailed explanation of this system and its implementation.

A.1 Architecture

Similar to what had be done in **Im-O-Ret**, we divided this system into two modules: the Thumbnail module and the Object Retrieval (OR) module. Figure A.1 illustrates the architecture of this system.

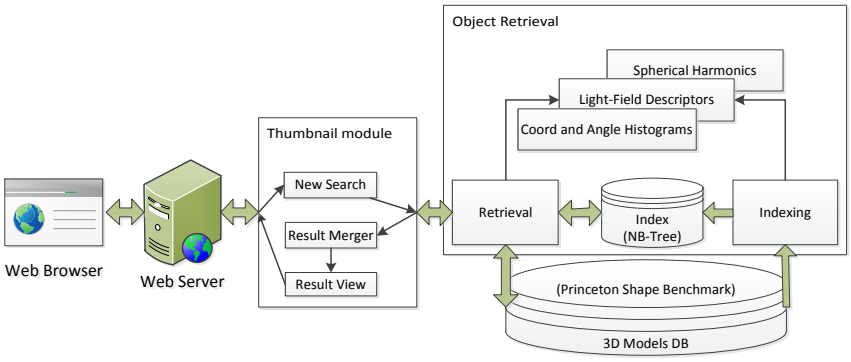


Figure A.1: THOR: Architecture

The OR module, was the same used in **Im-O-Ret**. This was done in order to make the search results provided by both systems the same, so that the only query-result distribution and visualization would differ, allowing us to test our proposal. A detailed explanation of this module functionality was already given in Chapter 4.

The Thumbnail module was divided into three sub-module, according to the task it performs: the New Search module, the Result Merger module, and the Result View module. The New Search module will receive a query model, and communicate with the 3D module to conduct the search, similar to what is done by the Core module of our 3D prototype. The Result Merger module, will read the results of each of the matching algorithms, and create a merged result, where the modules are order by the Euclidean distance. The Euclidean formula is described in Chapter 2. Finally, the Result View module, will read the merged results, are display the thumbnail of each of the retrieved models.

A.2 Description

The main purpose of this system was allowing the realization of simple 3D object retrievals, where the results were displayed as a list of thumbnails, giving the participants in our user test, an overview of the traditional 3D object search systems. As such, little work was done regarding the interface and query specification. Figure A.2 illustrates the interface of this system.

Figure A.2: THOR: Query-results of a search for similar to model m1529.

For query specification, there are two available: query-by-example, and query-similar. In the query-by-example, the user uploads a model, that will be used as query to search for similar, while on the query-similar, which in practice is also a query-by-example, the user selects one model as query, from a panel of 36 pre-selected models. The query-similar, is in practice also a query-by-example. In our evaluation, the 36 pre-selected models provided for new searchs in both **Im-O-Ret** and **THOR** where the same.

Appendix B

Query Models

For evaluation purposes, we used a group of 36 models as starting point for a new search, that could be used by both THOR and Im-O-Ret. These 36 models are illustrated in Figure B.1.

Figure B.1: The 36 query models used for the search efficiency evaluation.

Appendix C

Voice Commands

In Table C.1 are depicted the voice commands available in our system. From one to ten, they are mainly used for search and query-result visualization. The commands nine and ten, allow the user to navigate through the search history, moving back and forward the query-results. The voice commands used for combination with the pointing for the *wayfinding* navigation are listed from eleven to fourteen. Finally, the commands from fifteen to seventeen when used give the user information of how to interact with our system and which commands are available.

Action	**Voice Command**
1. Initiate a new search	(Start) new search
2. Clean the displayed search results	Clean (search) results
3. Search similar objects	Search/Find/Locate (objects) similar to this/that
4. View in detail a single object	View this/that (object)
5. Select object	Select this/that (object)
6. Search similar to selected object	Search/Find/Locate (objects) similar (to selected)
7. View in detail the selected object	View selected (object)
8. Rotates a specific object	Rotate/Turn/Spin this/that (object)
9. Show previous query-results	Show/View/Display previous/former (search/query) results
10. Show next query-results	Show/View/Display next/following (search/query) results
11. Move camera to the start position	Go/Move (to) start position
12. Move camera closer to a specific object	Go/Move here/there Go/Move (to) this/that (object/position)
13. Camera zoom in	Zoom in Get/Move closer
14. Camera zoom out	Zoom out Get/Move further
15. Help	Help What (words) can i say What (words) can be said
16.	How to search (similar) (objects)
17.	How to navigate

Table C.1: List of voice commands

Appendix D

User tests specification

D.1 Traditional vs Im-O-Ret task specification

For the comparison of the Im-O-Ret with current traditional systems, we implemented a group of test tasks of increased difficulty, where each task would require more steps/time in order to complete. For the selection of these tasks, we conducted a preliminary tests, with the panel of six users in the preliminary stage. To each user we presented an explanation of each system usage. After the explanation, we showed a picture of a real life object and asked the user to find a similar object using each system. The three images used for each task are illustrated in Figure D.1.

First, the user would use the THOR, described in appendix A, and attempt to find a similar object to the picture, using the starting objects presented in appendix B. Then, the user was asked to conduct the same search using the Im-O-Ret. For both system the query objects presented where the same (Appendix B) as well as the query results, since both used the same engine, as explained in appendix A. The only difference of both system, which was the purpose of this test, was the query result visualization and distribution.

(a) Task 1 (b) Task 2 (c) Task 3

Figure D.1: Images used for search tasks in the system comparison.

D.2 Comparison of Paradigms task specification

For comparison of the use of different devices for result exploration, we also used three different tasks of increased difficulty. In these tests, the user were asked to conduct random searches, followed by the navigation and manipulation of query results. These tasks were used for all the interaction paradigms used.

For the first task, the user was asked to conduct a simple search, in order to get used interaction paradigm. In this task, we explained how the system worked and that the results were scattered according to the similarity (the closer the more similar). Then in the second task, we asked the user to create a new search, followed by a navigation in the query results. Finally, in the third task, in addition to the search and navigation, we asked that the users select and manipulate the query results.

Appendix E

Implementation Details

As presented in previous Chapters, we used various devices in order to provide spacial reference. Then with combination with the voice commands described in Appendix C, we intended to provide a more natural way to interact with our system. In this appendix we will describe the mapping used with each device and in which context they were used.

E.1 Wii-Remote

For the mapping of the wiimote, depicted in Figure E.1, we used the pitch value to determine the cursors y-position, and the roll value for the x-position of the cursor, while the buttons only the **A** and the **B** buttons were used. The **A** button was used to center the cursor to the center position of the viewport. This was require in order to recalibrate the cursor position at the start of the system. The **B** button when pressed, would grab the object pointed by the cursor, and map the rotation values of the pitch and roll, to the object. Additionally, by using the nunchuck, it is possible to simulate the avatar navigation.

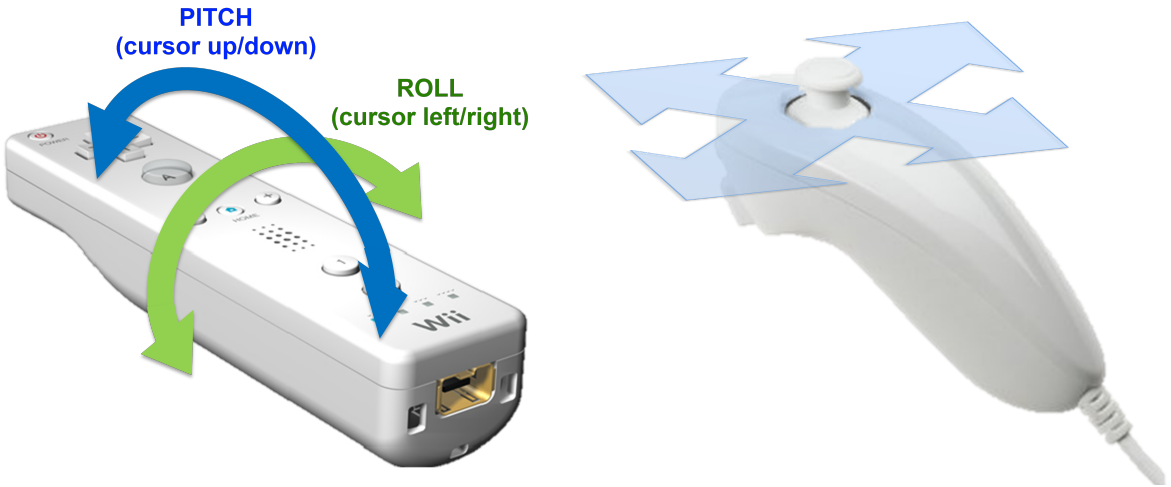


Figure E.1: Wii-Remote input mapping.

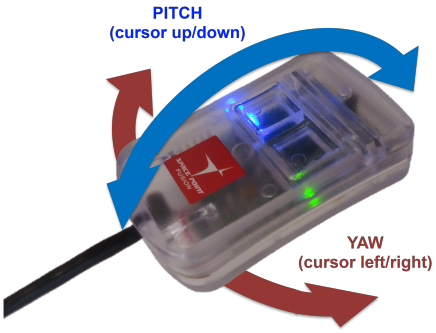


Figure E.2: SpacePoint input mapping.

E.2 SpacePoint Fusion

For the SpacePoint, we used the pitch value to determine the cursors y-position, and the yaw value for the x-position of the cursor, as showed in Figure E.2. Similar to the wiimote, we used the left button for object grabbing, and the right button for position calibration.

E.3 HMD Z800 and 5DT Data Glove

For the set HMD Z800 and 5DT Data Glove we used the tracking tools available in laboratory *João Lourenço Fernandes*[1] on the Taguspark campus. We assign three rigid bodies, as depicted in Figure E.3, and are used for head-tracking and spacial reference of the hands. Also, using the gestures captured by the 5DT Data Glove, we assigned some actions. The closed hand would allow to grab objects in a similar way to what was done with the wiimote and spacepoint, and the index pointing would move the cursor to the index's position of the model in the viewport.

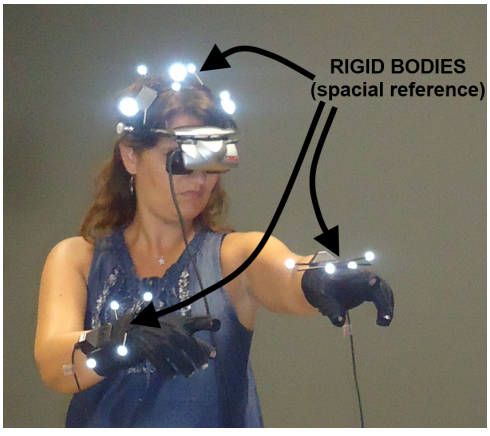


Figure E.3: HMD and 5DT Data Glove input mapping.

[1] http://lab-jlf.ist.utl.pt, accessed on 6/10/2011