Institute of Business Information Systems

iig.hevs.ch

A 3D OBJECT RETRIEVAL SYSTEM USING TEXT AND SIMPLE VISUAL INFORMATION

G. Rudaz, J. Hennebert, H. Müller

Internal Technical Report no 10-01
July 2010

Hes·so // VALAIS WALLIS
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences Western Switzerland

A 3D Object Retrieval System Using Text and Simple Visual Information

IIG-TR 2010.01

Gilles Rudaz[1] Jean Hennebert[2] Henning Müller[3]

July 7, 2010

Business Information Systems Technical Report

Business Information Systems Institute • HES-SO // Wallis
University of Applied Sciences Western Switzerland
TechnoArk 3 • 3960 Sierre • Switzerland

phone +41 (27) 606 90 01 fax +41 (27) 606 90 00 iig@hevs.ch http://iig.hevs.ch

[1]IIG, HES-SO // Wallis, Sierre, Switzerland, rudagill@gmail.com
[2]IIG, HES-SO // Wallis, Sierre, Switzerland, jean.hennebert@hevs.ch
[3]IIG, HES-SO // Wallis, Sierre, Switzerland, henning.muller@hevs.ch

Abstract

3D objects are being increasingly produced and used by a broad public. Tools are thus required to manage collections of 3D objects in a similar way to the management of image collections including the possibility to search using text queries. The system we are presenting in this paper goes one step further allowing to search for 3D objects not only using text queries but also using very simple similarity metrics inferred from the 3D spatial description. The multimodal search is stepwise. First, the user searches for a set of relevant objects using a classical text–based query engine. Second, the user selects a subset of the returned objects to perform a new query in the database according to a relevance feedback method. The relevance is computed using different geometric criteria that can be activated or deactivated. External objects can also be submitted directly for a similarity query. The current visual features are very simple but are planned to be extended in the future.

Keywords: technical report, IIG, HES-SO // Wallis

1 Introduction

In the context of the Internet the amount of visual information available and being used has risen strongly. In the beginning this was limited mainly to images. Millions of images have been made available via the Internet often on image sharing sites such as FlickR[1]. Most of the image search engines are based on keywords as Internet users have most experience in expressing their information needs in this way [1]. Starting in the 1990s visual information retrieval has become an important research domain with the goal to automatically extract visual features from images and then search for images also by visual similarity [7]. This allows to search for documents or images also when no annotation is available, which is often the case in large databases.

Sparked by the launch of web sites such as YouTube[2] , the load of videos has also known a sharp increase. Systems for visual and textual retrieval of videos have then been made available in a similar way to textual and visual image retrieval [8, 4]. Compared to image retrieval, an advantage for video retrieval is in the semantic information that can be eventually recovered from the audio track using speech to text translation systems. An interesting field of information retrieval is also in all the multimodal applications such as conference or meeting archives retrieval where a mix of speech, text document, images and video are available [3].

Compared to images and video, the amount of 3–dim–ensional (3D) objects available on the Internet is still relatively small. 3D information has been an extremely important aspect of industrial or architectural prototyping based on the use of expensive tools such as Catia or AutoCAD. The car industry has been and still probably is one of the main consumers of professional 3D modeling systems that are used for rapid prototyping. On personal computers, the gaming industry has also made 3D tools popular. The creation of 3D games is also becoming simplified with the releases of 3D game engines such as CryENGINE[3]. On the Internet, 3D objects are also increasingly used in applications such as SecondLife[4] or GoogleEarth[5]. To answer the growing demand for handling 3D objects and reusing them, the first market places of 3D objects are currently appearing on the Internet (see for example TurboSquid[6]). In a similar way as for images and videos, it has become important to develop tools for the management and retrieval of 3D objects. In this area, text based retrieval of tagged and classified 3D objects is prominently used. Research is now becoming active to analyze retrieval systems based on text information retrieval as well as on visual object features [5]. Several prototypes have been developed [10] and benchmarking data sets exist to compare tools and systems [11, 6]. The work we present in this paper has two objectives. First we investigate the use of a multimodal 3D object search engine based on text input and on visual similarity metrics inferred from the 3D spatial description. In this regard, we are proposing a stepwise simple relevance feedback procedure to reorder the returned set of 3D objects. Second we propose, from a software engineering point of view, an evolvable framework that can serve to

[1] `http://www.flickr.com/`
[2] `http://www.youtube.com/`
[3] `http://www.crytech.com/`
[4] `http://www.secondife.com/`
[5] `http://earth.google.com/`
[6] `http://www.turbosquid.com/`

develop further 3D object search engines. The text–based retrieval is based on the robust Lucene search engine that has been evaluated for text retrieval many times. Our visual features are kept very simple to show the feasibility of our approach. However, no quantitative evaluation of the retrieval quality has been performed so far. Another constraint of our system was that it had to be web-based.

The paper is organized as follows. In Section 2 we describe the database that was used to calibrate our prototype, as well as the main technologies used for the implementation. Section 3 gives a more detailed overview of our system as well as an explication of the basic algorithms. Section 4 concludes our work and explains some of the current limitations.

2 Methods

In this section, we provide more details on the data and tools used for the work described in this article. We also describe our main motivations in the different choices we have made.

2.1 Database

We have selected the Princeton Shape Benchmark (PSB) database that has been extensively used for testing shape retrieval algorithms [6]. Our choice is motivated by the fact that the PSB database contains 1814 polygonal models collected from the web and classified by humans into a small semantic hierarchy. The PSB database provides a set of hierarchical classifications and annotations for each model according to its primary function and its shape. This meta information is used as–is to build the index with Lucene. As shown in Figure 1, the PSB database stores each 3D object as a set of files in a directory called m<model id>. A directory contains the model itself, a textual description of its geometrical features and a 2D thumbnail.

2.2 Technologies

The goal was not to re–implement all mechanisms needed to construct a search engine. So we choose to reuse a set of well–known technologies and concepts.

2.2.1 Lucene

Lucene[7], as part of the Apache project, is a text indexing system providing powerful indexing and searching capabilities [2]. Lucene can automatically index free text documents and also retains part of structured information as in our case with author, title, and full textual description. Lucene then offers a querying language that allows detailed queries over the index. A Lucene index can easily be queried using Java, and this is how it was integrated in the system described in this paper. We do not use the full functionality of Lucene but only a subset detailed further down in the article.

2.2.2 Java 3D and the OffLoader library

Java 3D represents the Java community's technology for creating, displaying, and manipulating 3D scenes [9]. To load a 3D object into a 3D scene, Java3D uses a loader. Natively, only the WaveFront format (OBJ) is supported. Specific loaders can easily be added to Java3D to handle most of the commonly used 3D description formats. The OffLoader library was added in our case to Java3D to handle the Object File Format (OFF) used by the PSB database.

2.2.3 Java Architecture for XML Binding (JAXB)

The PSB database uses the text file format to describe each 3D object but XML has been chosen for the current Web application because its database is entierly file-based and a unique file format was needed to store all data types such as objects' descriptions, system access permissions, and so

[7] `http://lucene.apache.org/`

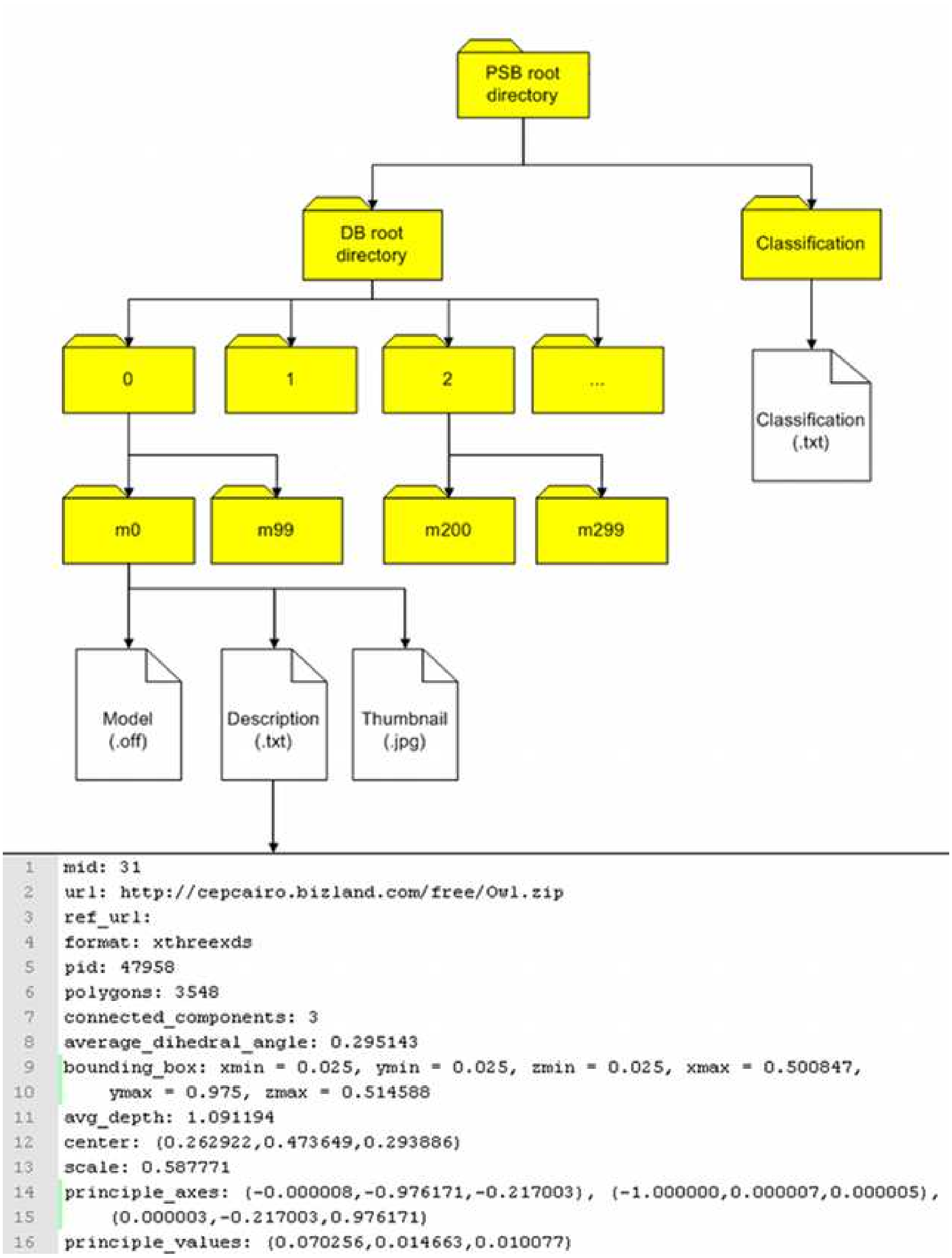

Figure 1: PSB Database Structure including the 3D objects, the thumbnails, and the meta data.

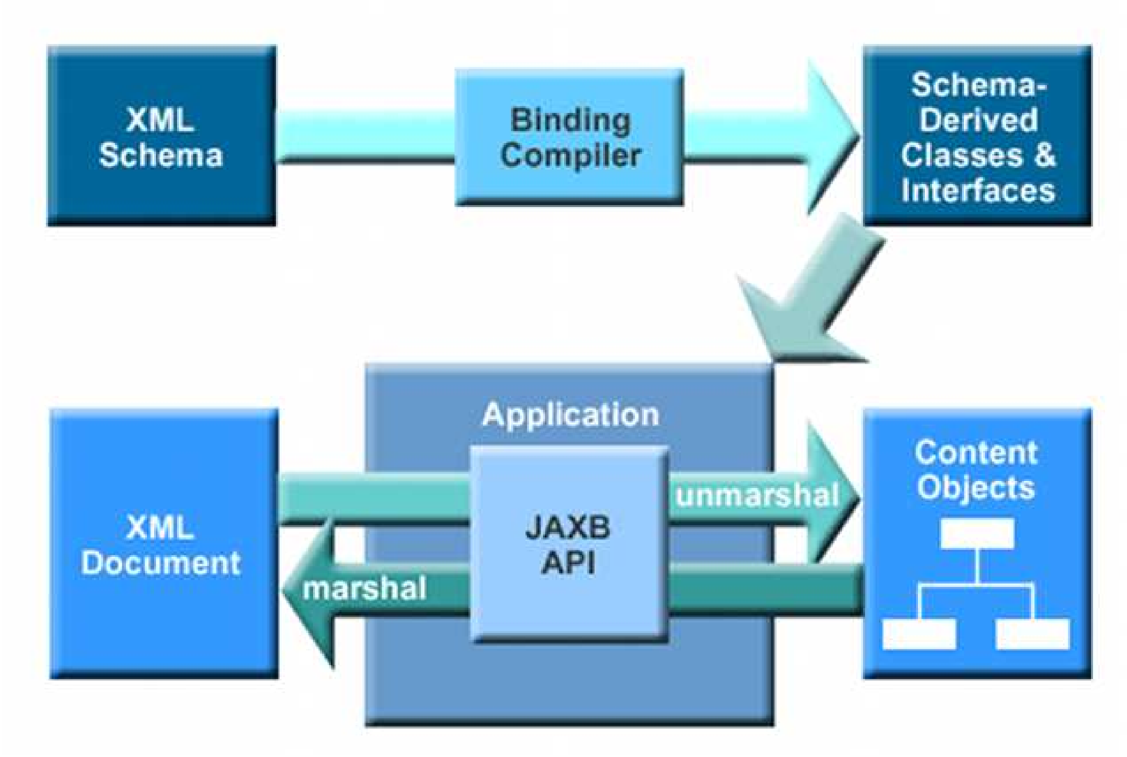

Figure 2: Mechanisms of JAXB to convert between XML and Java objects and vice versa.

on. This decision was also motivated by the XML's ability to allow a clearer structure into the file (hierarchy, collections and properties).

For this reason, we have chosen the JAXB library that allows to directly convert an XML file to a Java object and vice-versa. Rather than using SAX (Simple API for XML) or DOM (Document Object Model) libraries, JAXB was chosen because of its ease of use. As shown in Figure 2[8] , only an XSD (XML Schema Description) file is needed to generate Java classes that will be used to marshal and un-marshal a Java object. In our approach we do not use a separate database, so all information on the objects is read directly from it.

2.2.4 Asynchronous Javascript and XML (AJAX)

AJAX is used in the context of the described prototype to make the interface reactive to the user allowing the feel of a desktop application. Part of the communication is asynchronous and thus no reloading of the web page is necessary at all times.

3 Results

This section describes the main results of our prototype.

3.1 Functionalities Implemented

Figure 3 illustrates the entire system with its components that are organized in a classic 3–tiers client–server architecture. On the presentation layer, the light client Web application uses any Internet browser, AJAX, the Java Virtual Machine plus the Java3D and OffLoader libraries. The business logic is located in a Glassfish server that processes client requests by using the libraries Lucene, JAXB, Java3D and OffLoader to retrieve and organize the information from and to the

[8] http://java.sun.com/developer/technicalArticles/WebServices/jaxb/

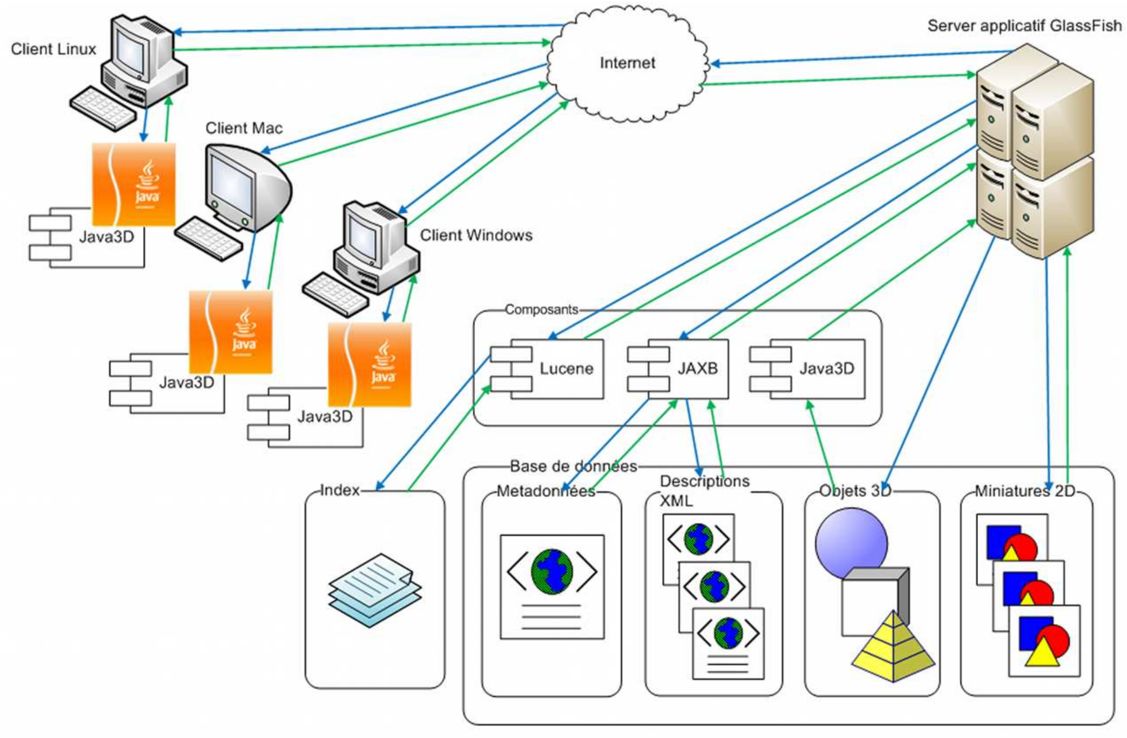

Figure 3: Global architecture of the developed application including the various base data and application used to provide the functionality for textual and visual retrieval using a web interface.

database layer composed by the data set itself (inspired by the PSB model and organized in a directory structure) and an index of the Lucene text retrieval system. The system can manage several user roles, for example to allow only for searching models or allowing to upload and/or delete models.

3.2 User Interfaces and Search Options

Figure 4 illustrates the different parts of the user interface. The user can add, edit, or remove 3D objects in the database management part according to her or his privilege.

The text search provides 3 search possibilities: random, simple and advanced. Random search displays a random selection of objects, simple search allows for text search in all fields of the database and advanced search allows several criteria such as not, and, or that are detailed below. The results of a user query are displayed in the results box. Finally, on the right hand side, the geometric search box allows the user to select targets from the *results box* and to query for similar objects based on geometric criteria.

3.2.1 Text–based search

The random search simply retrieves a given number of 3D objects selected randomly from the database. The simple textual search allows the user to perform a query using a unique field in which either terms (words) or/and exact expressions can be used. The advanced textual search is somehow similar to a Google–like advanced search and is illustrated in Figure 5 showing the options to retrieve documents with all of a set of words, not containing other words or containing either one of a set of words. The advanced search also allows to search in title and description separately. The user can also limit the search to one of the semantic categories, to a certain 3D object format and he can decide how many results to display.

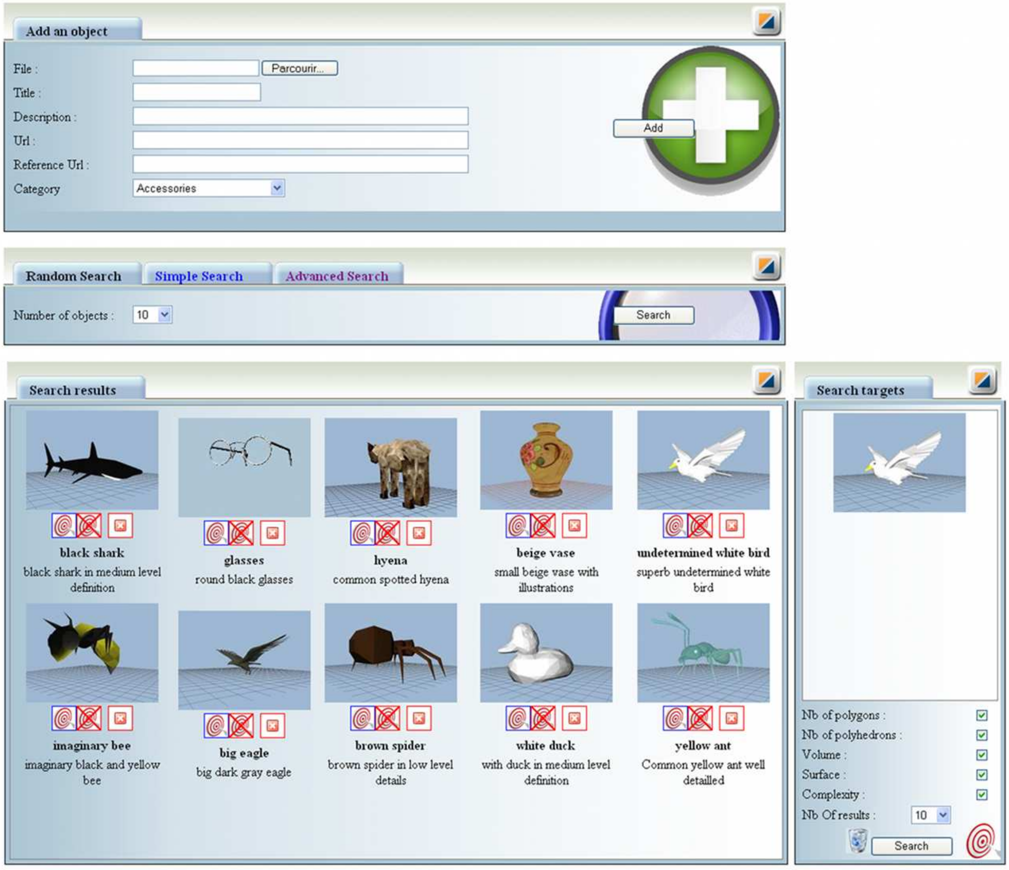

Figure 4: The user interface showing the results of a visual search.

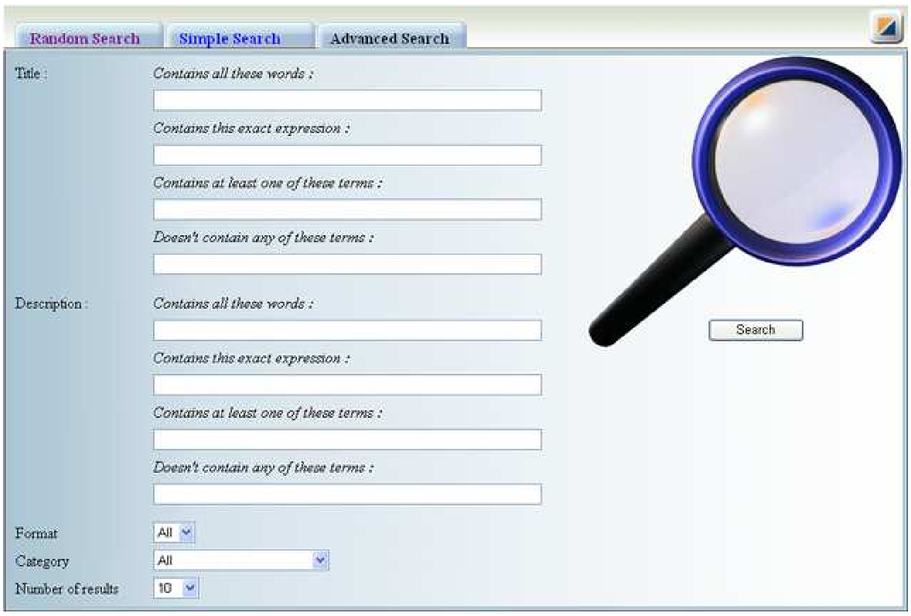

Figure 5: Advanced Textual Search allowing search with logical operators and in the descrption of the object and the title separately.

Figure 6: Simple geometic (visual) search results.

3.2.2 Geometric content–based search

The geometric search allows the user to choose one or several 3D objects as targets to search the n (defined by user) nearest neighbors. The user can select any combination of 5 predefined geometric criteria:

- the number of polygons making up the model,
- the number of polyhedrons making up the model,
- the volume,
- the surface,
- the index of complexity (surface divided by volume).

The aforementioned measurements are normalized between 0.0 and 1.0 taking into account the maximum and minimum values computed on all objects in the database. After normalization, the system computes distances between the target objects and all objects in the database. A simple Euclidian distance is used to calculate the visual similarity value (see Equation 1).

$$\text{distance} = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2} \tag{1}$$

The Euclidian distance computed during the first step is then divided by the square root of the number of selected criteria to normalize the value (see Equation 2). The ranking for each object in the database is finally computed based on its distance from the target(s).

$$\text{similarity} = 1 - \frac{\text{distance}}{\sqrt{\text{number of criteria}}} \tag{2}$$

The most similar objects according to this ranking are then retrieved and displayed to the user as shown on Figure 6. In the case of several input images, the average distance between each 3D object and all search targets is considered to compute the overall ranking. This again was a choice to keep the solution as simple as possible.

4 Conclusions and Future Work

The work described in this article details the creation of a simple, web–based 3D object retrieval system. It can be seen that components allowing the manipulation of 3D objects do exist also in a web-based environment with Java 3D. This framework allows for an easy manipulation of 3D content within a web browser. Search tools for textual retrieval have existed for a long time and are easily integrated. Visual search has so far only been integrated on a very basic level in our prototype and it is foreseen to develop this aspect of our system. A quantitative performance comparison is also necessary to evaluate the quality for visual as well as textual retrieval. Literature on such visual 3D retrieval exists and many techniques from content–based image retrieval can also be adapted to 3D content Other steps foreseen include the extension of the system to more than the two 3D formats currently included. Potentially the brokers of 3D objects on the web could be interested for indexing their collections and allowing new access methods for their clients.

5 Acknowledgements

This work was partially funded by the BeMeVIS project of the University of Applied Sciences Western Switzerland (HES–SO).

References

[1] P. G. B. Enser. Pictorial information retrieval. *Journal of Documentation*, 51(2):126–170, 1995.

[2] E. Hatcher, O. Gospodnetic, and M. McCandless. *Lucene in Action*. Manning Publishing Company, 2009.

[3] Denis Lalanne, Dalila Mekhaldi, and Rolf Ingold. Talking about documents: revealing a missing link to multimedia meeting archives. In *Document Recognition and Retrieval XI, IS&T/SPIE International Symposium on Electronic Imaging*, volume 5296, pages 82–91, 2004.

[4] Stéphane Marchand-Maillet. Content–based video retrieval: An overview. Technical Report 00.06, CUI - University of Geneva, Geneva, Switzerland, 2000.

[5] P. Min, M. Kazhdan, and T. Funkhouser. A comparison of text and shape matching for retrieval of online 3D models. In *European Congress on Digital Libraries (ECDL 2004)*, pages 209–220, Bath, UK, 2004.

[6] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. In *Shape Modeling Applications, 2004*, pages 167–178, Genova, Italy, 2004.

[7] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Armarnath Gupta, and Ramesh Jain. Content–based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[8] Stephen W. Smoliar and HongJiang Zhang. Content-based video indexing and retrieval. *IEEE MultiMedia*, 1(2):62–72, 1994.

[9] H. Sowizral, K. Rushforth, and M. Deering. *The Java 3D API Specification with CDrom*. Addison–Wesley Longman Publishing, 2000.

[10] Johan W. H. Tangelder and Remco C. Veltkamp. A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications*, 39(3):441–471, 2008.

[11] Frank B. ter Haar, Mohamed Daoudi, and Remco C. Veltkamp. Shape retrieval contest 2008: 3d face scans. In *IEEE International Conference on Shape Modeling and Applications, 2008*, pages 225–226, Stony Brook, NY, USA, 2008.

LIST OF PREVIOUS TECHNICAL REPORTS – INSTITUTE OF BUSINESS INFORMATION SYSTEMS

pdf versions available under http://publications.hevs.ch

10-01 A 3D object retrieval system using text and simple visual information
G. Rudaz, J. Hennebert, H. Müller ; July 2010