

# A Framework For 3D Object Retrieval Algorithm Analysis

Tiago Lourenço      Alfredo Ferreira      Manuel J. Fonseca

Department of Computer Science and Engineering  
INESC-ID/IST/Technical University of Lisbon  
{tiago.lourenco,alfredo.ferreira}@ist.utl.pt, mjf@inesc-id.pt

---

## Abstract

*The increasing number of three-dimensional objects available on digital format triggered a great interest in research in this domain. Finding efficient methods of analysis, comparison and retrieval of 3D models has become an important task.*

*However, despite the existence of some benchmarks with collections of 3D models, annual contests with specific tracks to compare techniques, and even a framework online (MMW.com) which allows to compare the performance of descriptors, there is no integrated system that provides, in a centralized manner, the necessary tools to study and compare the various techniques associated with 3D object retrieval.*

*In this article, we present a modular and scalable web-based system that allows the addition of new components, like shape descriptors or segmentation algorithms, with minor effort by researchers who developed them.*

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications—H.3.m [Information Storage and Retrieval]: Miscellaneous—

---

## 1. Introduction

The increasing number of three-dimensional objects stored in collections triggered the research on three-dimensional model analysis, comparison and retrieval methods. During the last few years, researchers have been developing several algorithms related to shape description, segmentation of 3D objects, among others.

However, despite the augmented interest in this area there is no simple and dedicated system which offers to researchers a centralized repository for algorithms and techniques developed within this context. The reduced number and availability of tools to support researchers' work on this domain is a major problem that we aim to minimize.

The work we carried out focuses on the development of a framework that will provide a simple and centralized workbench for 3D shape analysis, classification and retrieval, the Shape Analysis Workbench (ShaAna WB). Despite its simplicity, this workbench covers a wide range of topics in the field, such as object segmentation, best view selection, shape description or mesh reconstruction. This innovative system should allow researchers to study and compare available techniques with minor effort. It also allows researchers to provide their algorithms to be added to the system. The so-

lution is based on a client-server architecture working over the internet, thus easily available.

In next section we present some related work. In Section 3 we introduce our solution, the ShaAna Workbench, briefly describing the system architecture. Next, we exhibit the methodologies adopted for the evaluation of our work, particularly with regard to the API designed and the user interface. Finally, we produce some conclusions and point out future work.

## 2. Related Work

With the increasing amount of multimedia information available, is necessary to have effective retrieval algorithms. To support the research on this topic, standard evaluation techniques and methodologies were required.

The AIM@SHAPE Shape Repository [AIM06] is a shared repository with a collection of more than five hundred models. Its most striking feature is a full documentation of the most interesting geometric properties provided by detailed metadata of the common shape ontology.

However, the AIM@SHAPE Shape Repository only provide offline solutions. The models are obtained and each re-

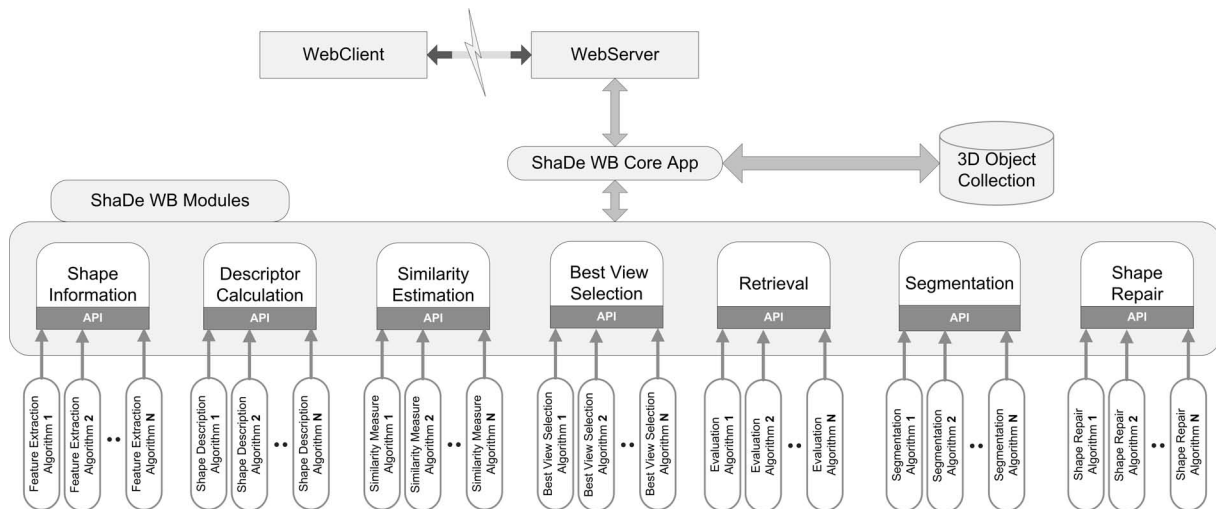


Figure 1: Architecture of the proposed solution.

search group makes its own analysis of algorithms locally, having little contact with the analysis made by other groups.

In turn, Bonhomme *et al.* deployed MyMultimediaWorld.com (MMW.com) [BMC\*08], an online platform for sharing various types of media, including video, image, audio and 3D objects. This platform follows the MPEG standards, using MPEG-4 for the representation of media and MPEG-7 for its description. Additionally, it also has an open Application Programming Interface that allows the addition of new descriptors. Unfortunately, this solution for 3D retrieval is part of a larger and complex platform for multimedia analysis, and is focused more on the retrieval and not so much in the analysis of three-dimensional models.

### 3. Framework for Algorithm Analysis

To the extent of our knowledge, there are no tools that allow comparison across most of the topics studied in the area of 3D object retrieval. In that context, we propose a framework which is modular, supports various 3D shape descriptors and evaluation measures, and provides a simple way of analysing three-dimensional models on diverse topics such as, segmentation, selection of the best views or reconstruction of meshes.

#### 3.1. System Overview

Our solution is based on a simple client-server architecture, depicted in Figure 1. The system relies primarily on two main components: a front-end web page and a back-end site. The front-end is basically a simple web page that serves as an interface, offering the ability to test and compare the various algorithms. The back-end has the core of the application and responds to requests sent by the front-end, providing the

necessary services for the effective functioning of the system.

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

As illustrated in the architecture diagram, we subdivide each module in sub-modules for each particular feature. This modular and scalable architecture make easier the addition of features, such as a new shape descriptor. For that, we simply add a new sub-module that implements the shape descriptor to the descriptor calculation module. This type of behaviour is similar in other modules, since each module has an API that provides the interface of the services and facilitates the addition of the new sub-modules. The API is adapted through a XML configuration file that each algorithm has to follow. These algorithms that are added are no more than simple executable files (which for now only run on Windows). So, when someone wants to run a particular algorithm, the system will run the corresponding executable file and show the results produced. Currently, some algorithms have been integrated into the system, Kazhdan *et al.* Spherical Harmonics descriptor [KFR03], Attene's Hierarchical Fitting Primitives segmentation algorithm [AFS06] and Mesh Fix shape repair algorithm [Att10], Mortara *et al.* Tailor segmentation algorithm [MPS\*03] and Best View Selection algorithm [MS09].

Additionally, we connected a database of 3D objects to the application core. In an initial phase, we are using the SHREC

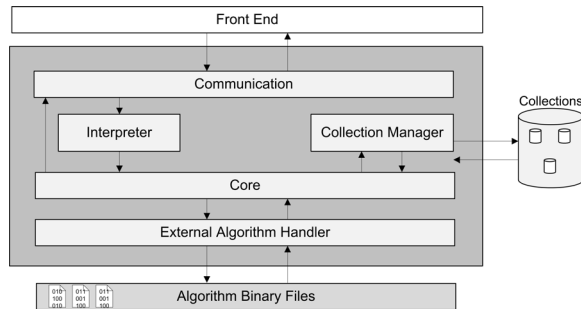


Figure 2: Architecture of the system's Back-End.

2007 Watertight collection. However, like other components in the proposed architecture, it will be relatively easy to add other collections taking into account that this operation is not done regularly, since the classification and integration is done manually.

Any information generated by the user is stored locally on the server while the user is using the system, and then all data is deleted. Also, for now, we are only keeping the shape signatures for feature vectors.

### 3.2. Back-End

The back-end is composed by a server that responds to the requests of the front-end by executing some algorithm and returns the results back to the front-end. This communication is accomplished through the exchange of XML messages between the two components. Through Figure 2 it is possible to see in more detail the modules that comprise the system back-end.

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

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

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

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

### 3.3. Front-End

As referred in section 3.1, our system comprises a front-end web, which is the interface where users can analyse models, shape descriptors, etc. It presents different views of the models, graphical representations of the shape signatures or charts illustrating the results.

The user has the possibility to load an external model or use a model of any of the existing collections (for now only SHREC 2007 Watertight collection is available), which were previously added to our system.

Then, it is possible to use the algorithms integrated into the framework. For example, compute the SHA (Spherical Harmonics) and see its visual representation (see Figure 3), repair the mesh of an object using the algorithm Mesh Fix, segment a model using HFP (Hierarchical Fitting Primitives) or calculate the best view of a model. the user can

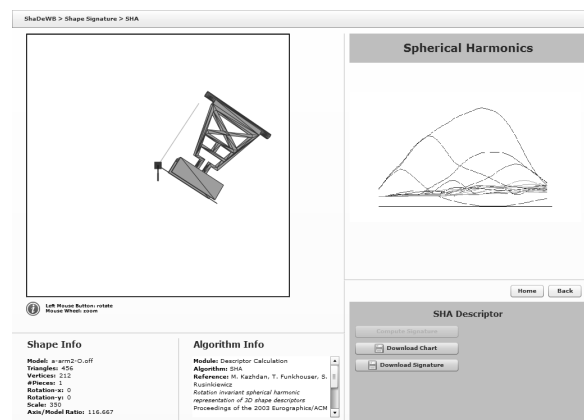


Figure 3: Spherical Harmonics calculation screen.

see the three-dimensional and exercise control over it. It is also possible to choose a module that contains algorithms (currently, only the modules of Descriptor Calculation, Best View Selection, Segmentation and Shape Repair have algorithms integrated).

#### 4. Evaluation

A preliminary evaluation of the ShaAna WB involved the use of the system by researchers and gather their views, through questionnaires and informal conversations on two main points, the API and the user interface.

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

In order to evaluate the user interface of the system, we had the participation of seven researchers who used the system freely. In the end, they answered a questionnaire that aimed to gauge the state of satisfaction of the current prototype and to improve the system user interface.

In general, the opinion about the ShaAna WB was good, and all users have given an overall positive view of the system's interface. The users identified several misconceptions or situations that should be targeted for improvement, pointing suggestions for changes. This evaluation allowed a glimpse of the work that needs to be done in the future so that the user interface will respond better to what the researchers want.

#### 5. Conclusions and Future Work

In this document, we presented a framework for algorithm analysis, the ShaAna Workbench. This framework is based in a client-server architecture that works over the internet. The system is composed by a front-end web that behaves as the user interface and gives the users the possibility to test and compare different 3D shape techniques. It also consist of a back-end site which has the core of the application and that is developed in a modular and scalable way to allow the addition of new shape descriptors, segmentation algorithms, etc.

Despite some positive feedback, as a result of an user evaluation, there is still much work to be done hereafter. We intend to adopt further contacts with other researchers in this area, trying to awaken their interest in our solution, thus providing their algorithms to be integrated in the ShaAna WB. Also related to this task should come up a further development of the API of the system as more algorithms are added.

The complete solution will allow the study and comparison of existing and new 3D shape retrieval related techniques with minor effort. We believe that the outcome of our work will be an important contribution for this field of research.

This solution is an initial version of our work, being limited to the use of models defined by polygon meshes, following the most common approach. However, our approach is flexible and can be easily adapted to different circumstances, being generally needed only simple changes to the executable files provided by the authors.

#### Acknowledgments

The work described in this paper was partially supported by the Portuguese Foundation for Science and Technology (INESC-ID multiannual funding) through the PIDDAC Program funds and also through the project 3DORuS, reference PTDC/EIA-EIA/102930/2008.

#### References

- [AFS06] ATTENE M., FALCIDIENO B., SPAGNUOLO M.: Hierarchical mesh segmentation based on fitting primitives. *Vis. Comput.* 22, 3 (2006), 181–193. 2
- [AIM06] AIM@SHAPE: The aim@shape shape repository. <http://shapes.aimatshape.net>, 2006. 1
- [Att10] ATTENE M.: A lightweight approach to repairing digitized polygon meshes. *The Visual Computer* (2010). 2
- [BMC\*08] BONHOMME B. L., MUSTAFA B., CELAKOVSKY S., PREDÁ M., PRÉTEUX F. J., DAVCEV D.: On-line and open platform for 3d object retrieval. In *3DOR* (2008), pp. 73–79. 2
- [KFR03] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Rotation invariant spherical harmonic representation of 3d shape descriptors. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 156–164. 2
- [MPS\*03] MORTARA M., PATANÈ G., SPAGNUOLO M., FALCIDIENO B., ROSSIGNAC J.: Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. *Algorithmica* 38, 1 (2003), 227–248. 2
- [MS09] MORTARA M., SPAGNUOLO M.: Semantics-driven best view of 3d shapes. *Comput. Graph.* 33, 3 (2009), 280–290. IEEE International Conference on Shape Modelling and Applications 2009. 2