3D VQI: 3D Visual Query Interface

Subhash Uppalapati, John C. Femiani, Anshuman Razdan, and Kevin Gary.

Division of Computing Studies
Arizona State University at the Polytechnic Campus
Suppala1@asu.edu

Abstract

The 3D Visual Query Interface (3D VQI) is a web-based database application that allows for shape-based searches in addition to more traditional text-based queries of the 3D vessel dataset. This interface allows researchers remote access to vessel collection that can be analyzed from anywhere in the world. The interface is divided into two sections with right half supporting the ability to draw profile curve both symmetric and asymmetric. This query can be further refined using the left half which provides contextual based information. Using a curve-matching algorithm the software then searches the database and returns a series of potential matches sorted by the percentage of similarity between the curves. Each match can then be selected for more detailed viewing of the 3D model and data or for further analysis.

1 Problem Statement

The 3D Visual Query Interface (VQI) is a web application which allows shape-based matches in addition to text-based search into a 3D vessel database. The application focuses on providing the ability for the user to draw the profile curve as one of the main input criteria for shape-based matching in addition to text-based search. Based on the search conditions given, the application executes a shape matching algorithm and contextual-based search and displays all the matching results. The application also needs to display the 3D view of the vessel in a more robust way reducing the burden on the users to install software. It also needs to have cross-browser and cross-platform capabilities and provides help, demo and feedback features for the users to effectively use the web interface.

2 Introduction

3D knowledge often plays an important role in archaeology. Archaeologists characterize the development of Native American cultures by studying the 3D form of pottery. Access to a large repository of 3D vessels via the web can provide a means to study pottery when access is limited by geography or other barriers. Quantitative methods of reasoning about the shape of a vessel are becoming far more powerful than was possible when vessel shape was given a mathematical treatment by G. Birkhoff [1]. Therefore physical measurements and other interactions using an Online, interactive digitized 3D version of a specimen will facilitate new techniques for research and education by archaeologists.

This report describes the methodology used in the development of the VQI web application on content-based retrieval and 3D interactive display of archaeological vessels. This report uses vessels from the Classic Period (A. D. 1250 – 1450) of the prehistoric Hohokam culture area of the Southwest (Salt/Gila River Valleys) near present-day Phoenix, Arizona. The main components of this work are a visual query applet, which allows archaeologists to form a shape-based query, and a 3D Display component, which provides a 3D interactive platform to inspect and make measurements on 3D vessels. Obtaining the geometric features involves obtaining shape information from the scanned three-dimensional data of archaeological vessels, using 2D and 3D geometric models to represent scanned vessels, extracting features from geometric models and storing the feature information in database for Web-based retrieval.

In a typical workflow Figure 1, digital samples are acquired from physical specimens using 3D data acquisition devices and the resulting point set is wrapped by a 3D surface, cleaned of spurious digitization artifacts. It is segmented and annotated by a domain expert to produce 3D content with added semantic information. The compressed, encrypted, or watermarked geometric and semantic information is part of the database, so that it can be used in searching documents that include descriptions and other structured domain specific data.

The contributions of the project are as follows:

• Provide access to 3D content in the form of a database of digitized pottery samples.

• Demonstrate a visual query interface based on profile curves, silhouettes, or cross-sections of 3D objects as a simple cost-effective and intuitive way to formulate 3D queries into a shape database when laser scanners or other 3D digitization tools are not available.

• Provide a cross-platform, portable, cross-browser viewer that allows interaction and measurement of the pottery samples.

3 Prior Art

Several systems exist that provide different ways to form a query, retrieve matching objects, and/or display the 3D content on the web. One of sketch-based query model algorithms is the pseudo-3D sketching [16]. In pseudo-3D sketching, as the name implies the user inputs the cross-section of desired shape and also mentions the size in the still undefined dimension. In the interactive 3D model sketching based on the user's hand movements from consecutive frames will be extracted. One other example is Princeton search engine [6] which expects outline sketches up to three views and matches to pre computed sketches of a 3D model using Fourier descriptors of rendered images. The Purdue search engine [21] matches 2D views using 2.5D spherical harmonic descriptors.

Some of the shape matching algorithms and their problems in shape retrieval, shape recognition and classification, shape alignment and registration and shape approximation and simplification were discussed by [3]. To find the available 3D models on the web Thomas Funkhouser [17] developed a focused crawler for 3D models. Their system provides a wide variety of query interfaces based on text, 3D shape, and 2D shape, which were designed to be easy to use, and hide parameters of the underlying matching methods.

This report is an extension of the earlier work done in the domain-specific query interface. Rowe et al. [18] describe a system where the user can draw a 2D outline of a ceramic vessel as a shape query into a database of 3D models of such vessels. For example, the web site of the State Hermitage Museum in St. Petersburg uses the “Query by Image Content” method to allow users to search paintings by drawing simple colored sketches [19].

A detailed study of different content-based 3D shape retrieval methods were discussed by Johan W.H. Tangelder [21]. For the shape-based queries, Thomas Funkhouser [17] has developed a new matching algorithm that uses spherical harmonics to compute discriminating similarity measures without requiring repair of model degeneracy or alignment of orientations. It provides 46–245% better performance than related shape matching methods during precision-recall experiments. The most significant feature of [20] 3D shape similarity comparison method is that it accepts polygon soup and other ill-defined 3D models. Their approach is to use the rendered appearance only of the model as the basis for shape similarity comparison. Their method removes scale and positional degrees-of-freedom by using normalization and the three rotational degrees of freedom by using a combination of discrete sampling of solid angles and a rotation- invariant 2D image similarity comparison algorithm.

Most of the online websites that are available are still in experimental condition, query methods are difficult to use and not easy enough to understand the working mechanism. This gives scope for efficiently finding the 3d models on the web, the different query interfaces to use and the efficiency and effectiveness of the matching and 3D display of the object.

For the researchers in the field of Archaeology, to find the pottery based on textual data or of a particular shape, we developed a web based application. The system provides query interface to search based on both contextual data and 2D shape. The interface is developed keeping in mind the usability and simplicity. We used the latest efficient and effective shape matching algorithm for finding the similar objects (as demonstrated in [2]). The contribution of this report is an investigation of the design and implementation trade-offs in building such a 3D model search engine based on these methods.

4 System Design and Architecture

Once acquired, the aim is to make 3D digitized pottery samples accessible to the researchers. Archaeologists must be able to retrieve an annotated 3D model from the collection based on contextual or semantic information as well as by the shape or geometric similarities. Figure 1 describes the flow chart of the Web-based VQI. The query process in VQI combines a sketch-based interface which allows archaeologist's to draw the profile curve of a pot, in addition to traditional text and metric data. A free-form profile sketch can be created in the interface window with options to draw asymmetric curve and to smooth the curve. The user can also upload image of the pot that he has, and draw outline in the applet.

One of the most difficult tasks in forming a 3D shape-based query is identifying an example shape. Techniques include using an existing 3D model such as a laser scan, providing a query interface with basic 3D modeling tools to build an example, or conducting a query based on sketches from multiple 2D points of view [4]. Pottery is often radial symmetric, so query is formulated based on a profile curve. A perfect match in the data base would be a surface of revolution generated from the profile curve itself.

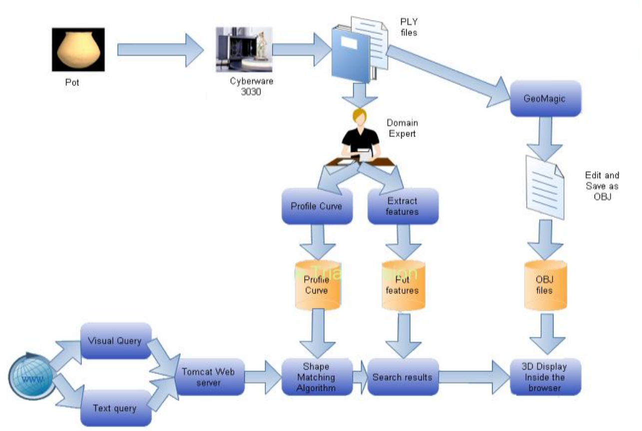


Figure 1: Flow Diagram

4.1 Acquiring Digital 3D Content

After digitizing archaeological vessel via a 3D laser scanner (Cyberware 3030), a triangle mesh is obtained that is constituted of triangular faces, edges and vertices. The triangle mesh is used as a raw data for further analysis. This raw mesh captures the shape of the vessel but, but not in a way that is easy to interpret or work with in order to formulate a query. At this stage a domain expert identifies meaningful features [1] [1, 8]. The first step is to identify the salient cross-sections in a vessel, so that later a 2D query can be formulated by sketching a profile curve. In order to get a 2D profile curve from a vessel, archaeologists interactively select a cutting plane that intersects the triangle mesh of the vessel. If a triangle intersects the plane, then the intersection is a line segment connecting two points. The intersecting segments form several connected poly lines represented by 2D chain codes. The longest one is extracted as the shapes profile curve. NURBS curves are fit to the points of each chain code to form least squares approximation. Curvature, or turning angle in discrete point sets, provides useful information such as convexity, smoothness, and the locations of inflection points of the curve needed by vessel analysis. NURBS curves reduce noise that may be present in a dense profile curve and allow thereby reduce the impact of digitization errors or irrelevant details in the profile curve. Details are in [4]. Curvatures or angles in space don't really have a sign, but when shapes are planer one can convert curvatures or angles into signed values based on the right hand rule [4,5], see Figure 1. By the threshold of the magnitude of curvature plots from above corner points and inflection points are identified by zero crossings of the curvature plot. In addition wherever the cosine of the tangent angle crosses zero identifies points of vertical tangency. The number and position of these landmark points, along with the profiles endpoints, are used to categorize the overall shape of each pot.

4.2 User Interface Design

The user interface is developed keeping in view of the users of the application. It is developed so that the navigation from one page to another is smooth. It is also developed with the aim to keep unique look and feel throughout the website, and user friendliness. The only requirement for the website is to have Java Runtime Environment (JRE) in the machine used to access the website.

4.2.1 Visual Query Interface (VQI)

The query interface supports either text-based search or Visual Query based search or search based on both text and visual based search. The query input page is shown in Figure 2.

The text-based search works on the contextual data. The user can enter the time period a particular pot belongs to or the type of the pot or the archaeological site it belongs to. The user needs to provide one of these major criteria to effectively search for related pots in the database. Optional criteria that can be entered by the user include rim diameter, vessel volume, maximum diameter, and vessel height.

Visual query applet enables the users to draw profile curve of the pot as the search input. The applet supports asymmetric drawing of the curve by checking the asymmetric checkbox. Which means the user can draw profile curve differently to the left and right of the middle line. Once a basic outline of the curve has been drawn, user can smooth the curve by checking the smooth checkbox. To aid the user in better drawing the profile curves, there is provision for uploading the image of the pot to the applet. By uploading the image to the applet, user can draw on the outline of the image and he can remove the image from the applet by using remove button to see how the profile curve looks. To completely erase the drawing window and remove the image, user has the option to click on Clear button.

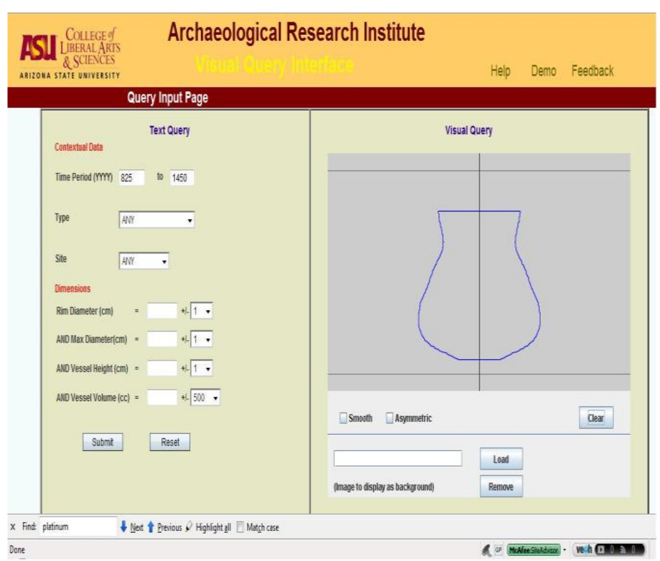


Figure 2: Query Input Page

4.2.2 Search Results

Once the user submits the query, search will be performed based on the criteria provided by the user. If the user provides the visual query then Shape matching algorithm will be used to match the pots from the database to the input profile. The matching of each pot will be represented in percentages from zero to hundred with hundred means exact match. Shape matching algorithm ranks the query results by descriptive and spatial similarity to the query image. Query response information is presented sequentially over several screens, each providing an additional level of information about the selected objects. The query results are displayed in a table with its detail and can be ordered based on the curve matching percentage, specimen number or vessel type. Figure 3 shows the first result screen, which displays thumbnail images and brief descriptions of the top search results. Also presented is a large image of the pot of the selected row, selectable using radio button, along with more detailed descriptive and calculated information.

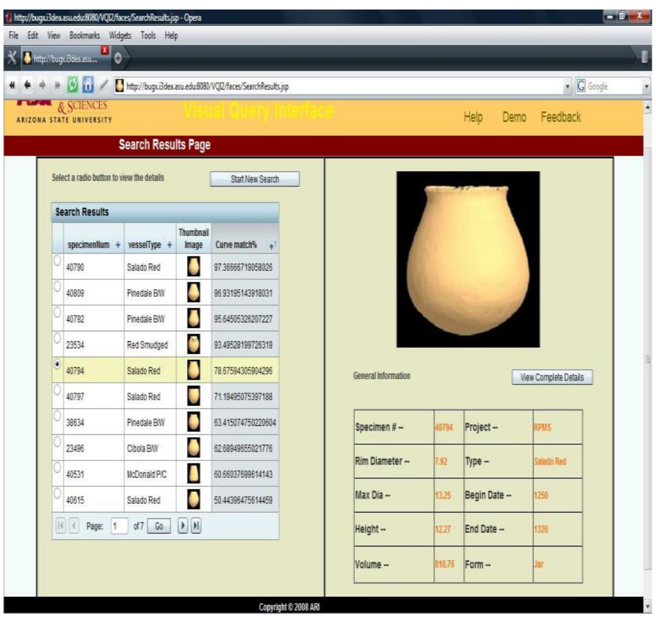


Figure 3: Search Results Page

4.2.3 Interactive 3D Display

The final presentation of 3D objects, the 3D display interface, makes the artifacts more accessible because it allows interactive exploration of the surface geometry as if the object was in the same room. In order to be accessible a viewer must be portable, it must be simple or automatic to install, users must be able to use it without admin rights on the machine, and it should be capable of launching and loading surface geometry quickly.

There are several options for deploying 3D content on the web, including the option of using an existing viewers such as JavaView [5]. Initially we chose to build a custom viewer. Java's WebStart Technology [7], which is bundled with recent versions of the Java Runtime Environment, allows digitally signed extensions such as JOGL or Java3D [13] to be deployed with a standalone application. After further research we found the JNLPApplet launcher [Add] which enables the applets embed within a web page without requiring the applet to be signed or performing any manual installation on the client's computer. This means that embedded 3D viewers can leverage the full power of Java for transmitting data and controlling behavior of 3D content available through a web page.

The viewer allows data to be encrypted, and it allows the rendered images watermarked. Intellectual property is protected by restricting the amount or type of data that can be saved for each sample.

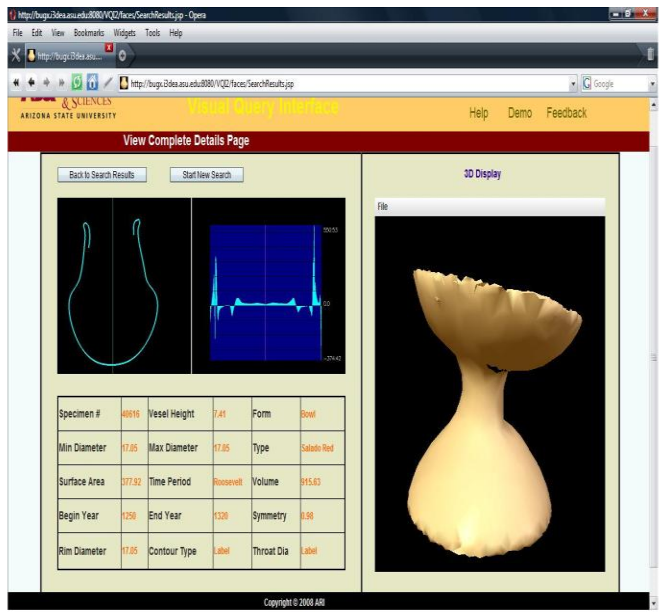


Figure 4: View Complete Details Page

4.2.4 Feedback Page

The feedback page allows the users to input their comments and problem experienced in using the website. This helps the developers to understand the user's needs and problems and helps in better building of website to make it more useful for the users.

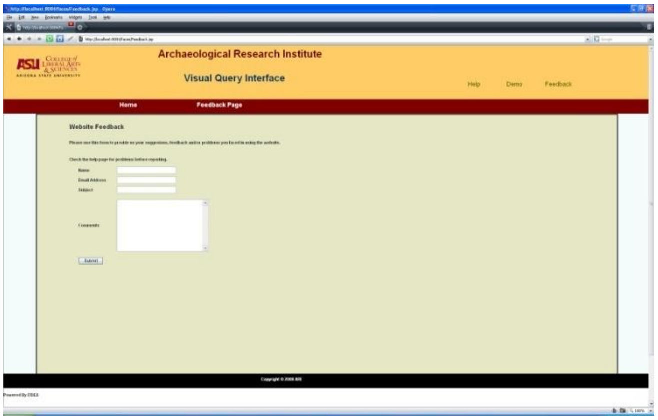


Figure 5: Feedback Page

4.3 System Architecture

The whole system containing the visual query and text-based query, search results, the shape matching algorithm and the 3D display is implemented in Java. The complete application comes as a Web Application Archive (WAR) and can be deployed on any webserver which supports J2EE 1.4. The application is platform independent and has cross browser support.

The original pots from the Archaeological Research Institute (ARI) were laser scanned using Cyberware 3030 resulting in PLY files. The PLY files were then edited to remove the markers used for scanning and save the 3D scan in the OBJ format. The OBJ files are saved in the MySQL database. The PLY files are studied by an archaeology domain expert and he selects the cutting plane for obtaining the profile curve. And features of the pot were extracted from the pot from the study of the expert. Then the contextual data, the thumbnail image, profile curve and all the computed data is saved in the database.

The application interface provides the ability to do either a text-based, visual query based or both text and visual query based search for the matching of pots from the pots database. Each request comes to Tomcat 5.5 web server which runs on Windows 2003 server. At the server side the shape matching algorithm pre computes the geometries of the profile curves from their profile curve descriptions chosen by the expert. Based on the contextual criteria provided and the shape drawn in 2D, the matching results will be computed and will be displayed in an order based on the matching percentage.

The application also supports the 3D display of the object inside the browser. For this, Java 3D is used to create a JApplet and by using JNLP Applet launcher it is possible to display the 3D applet inside the browser. This requires minimum install from the user and makes the 3D display hassle free unlike other viewers of 3D objects. This also gives the flexibility of the 3D display to work on cross browser and cross platform.

4.3.1 Database Server

All the information related to the pots is stored in the database. The DBMS used is MySQL 5.x which has new features like triggers, stored procedures and views included.

4.3.2 Application Server

The application server used to serve the web application is Tomcat 5. It handles all the user requests coming from the browser, maintains sessions, communicates with the database to retrieve records, responds to the user.

4.4 Shape matching Algorithm

Shapes are compared using normalized correlation of the coordinates of the two curves. The cross correlation is maximized over all possible translations, rotations and a finite rage of scales. The curves are aligned at the centers and a range of intervals extending from the middle 75% to the entire length of the longer curve are compared with the shorter curves. The rigid body motion that maximizes the normalized correlation between the two curves is identified using the closed form method presented by Horn [22]. Normalized correlations generally fall in the range from minus one to one, but in our case results will be positive. A correlation of one is a 100% match; a correlation of zero is the worst match possible. Score of minus one means that the curves are identical but the points were listed in opposite order. That case is not possible since Horn’s method would have flipped the curve to make the correlation positive.

4.5 Caching of vessel geometry calculations

Every time user submits the visual query, the input curve is matched with the profile curves stored in the database using the shape matching algorithm. One way of doing it is for every request, accessing the database for the profile curves, computing the geometries based on profile curve and then matching it with the input curve.

Theoretically the performance can be improved by caching the computed geometries of the profile curves in the application. When the web application is accessed for the first time, a connection will be made to the database and all the profile curves will be fetched from the database. Then using the application logic, all the required calculations will be made on the profile curves and the objects are stored in a HashMap.

Now, for every new request the application accesses HashMap to see if the calculated object is available. If it is available the shape matching algorithm is executed to find the matching percentage. If not the profile curve is queried from the database and after calculations, the object is added to the HashMap for future access.

4.6 Technologies and Tools

During the development of the web application several technologies and tools are used. The major components are described below.

4.6.1 Java Server Faces (JSF)

The framework followed for the development of the web application is Java Server Faces (JSF) [11] [12]. It is built on top of Model-View-Controller (MVC) architecture. User interfaces are developed using JSF technology.

4.6.2 Java 3D

Java 3D is a scene graph based technology useful for programming for the Java. It is built on top of OpenGL for displaying graphics. In this web application, 3D viewer for the pot display is developed using Java 3D inspired by one of the example PrintCanvas3D available at [13].

4.6.3 SQL

Structured Query Language (SQL) has been used extensively to retrieve and manage data available in the vessel database.

Based on the contextual data provided by the user, search query will be formed programmatically. Over the results obtained after contextual search, shape-based matching will be performed to obtain the matching percentage.

4.6.4 NetBeans IDE

NetBeans [14] Integrated Development Environment has been used for the development of the web application. The palette provided by the IDE is used for developing the user interfaces. Debugger tool is used to debug the code to fix the issues. Profiler is used to understand the hot spots in the application, the memory performance and CPU performance. Unit test cases are written to carry out method level testing.

4.6.5 TOAD

Tool for Application Developers (TOAD) makes database and application development faster and easier and simplifies day-to-day administration tasks [15].

TOAD data modeler has been used to model the database and ER diagrams. From the data modeler the SQL file to create the database can be directly exported. It also gives the option to specify the foreign key constraints. Once the database is modeled, it is created using the TOAD for MySQL software which can be used to administer the MySQL database. By using this freeware, the database administrators are connected to the DB server from remote machine to administer the data. Efficient queries are created using the query builder and the results can be immediately checked by executing the queries.

4.7 Project Management

The web application started with the requirements phase, continued with the design, development and testing. For managing the website Trac [10] is used which is Project management and bug/issue tracking system and SVN is used for code versioning.

4.7.1 Trac

Trac is used for project management by following systematic development. The project has several milestones created with each milestone having requirements as tickets. All activities will be tracked as ticket system. It has integrated wiki which is used to describe project terminology, all necessary information. It also supports the integration of SVN versioning system so the code can be browsed through the website. All the executables required to run this web application are available to download from the Downloads section of the website which can be accessed by registered users. Meeting minutes of weekly meetings is also maintained on this website.

4.7.2 Version Control

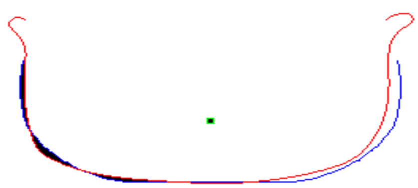
In this project Subversion, popularly known as SVN is used for version control. Versioning helps in saving the work with the ability to revert back to previous versions. It also helps in team oriented development where multiple developers are working on the same code base at the same time.

5 Experimental Results

Several experiments have been performed on the shape matching algorithm on how best the similar pots are getting retrieved from the database.

5.1 Experiment 1

In this experiment, the input is a bowl shaped pot and the results are observed based on the percentage of matches for different types of pots. Blue color line is for the input drawn using the visual query applet provided by the web application. Read color lines indicate the profile curve of the vessels present in the database. When the shape matching algorithm is applied on these two curves the matching percentages in the range from zero to hundred are shown in the table. The results are shown in Figure 6.

Vessel ID	Matching %	Curves
1950.001.00162	89.15	

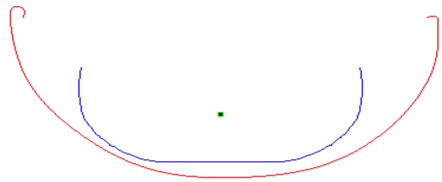
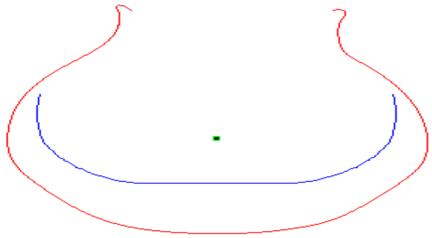
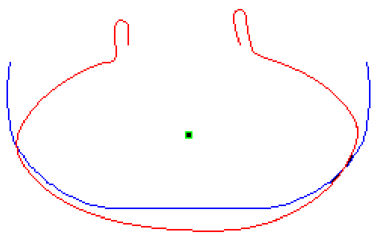
1963.0 02.000 04	88.21	
1950.0 01.001 47	87.75	
1950.0 01.001 58	85.00	

Figure 6: Experiment 1 results

5.2 Experiment 2:

In this experiment the input curve is a complex pot with one inflection point. Blue curve indicates the curve drawn using the visual query applet and the red curve is the profile curve of the actual pot available in the pots database. The results are shown in Figure 7.

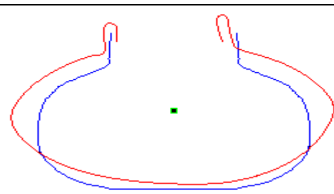
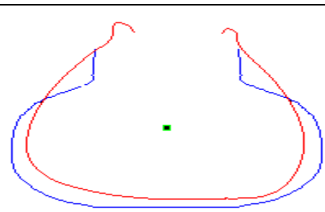
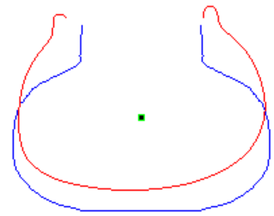
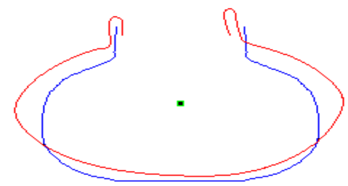
Vessel ID	Matching %	Curves
1950.001. 00158	90.70	
1950.001. 00154	87.21	
1959.053. 00003	86.46	
1950.001. 00147	75.40	

Figure 7: Experiment 2 Results

8 Future Work

3D VQI is a robust web application to aid researchers at remote location to access the pots database. It provides the ability to view the 3D display of the pot to study its features and geometries. The researcher can perform search on the pots database by providing a 2D profile curve-based search. Some of the features that can be added in the future are given below.

The Visual query applet can be enhanced to provide sample profile curves which the user can select and modify. The smooth functionality can be improved and by getting feedback from the users on the usability of the curve drawing, changes can be implemented.

The shape matching algorithm which forms the core of the shape matching can be enhanced with the results it is giving and with the feedback from the domain experts.

The 3D display applet which will display the 3D pot inside the browser depends on the requirement that the user has JRE installed in his machine. A message can be shown in cases where the OBJ file is not present or the JRE is unavailable.

9 Acknowledgements

The foundation for this project is from its parent projet Visual Query Interface (VQI). This report inherits the geometric calculations and acquiring the digital content from VQI. VQI has been a truly interdisciplinary project and the successes resulted from a dynamic interaction between researchers in a broad range of disciplines at Arizona State University. I thank Dr. Arleyn Simon and Dr. Stephen Savage from ARI, who have provided the requirements and domain knowledge related to the application. I sincerely thank Dr. Anshuman Razdan, who accepted to be my advisor and has given me the opportunity to work under him. I thank John C. Femiani who has written the shape matching algorithm and helped me throughout the development of the project. I also thank Dr. Kevin Gary to be a part of my committee.

10 References

[1] Birkhoff, G. Aesthetic Measure. Harvard University Press, 1933
[2] Berthold K.P. Horn, Closed-form solution of absolute orientation using unit quaternions, Journal of the Optical Society of America, 1987
[3] Remco C. Veltkamp, Shape matching: Similarity measures and algorithms, 2001.
[4] Bae, M. Curvature and Analysis of Archaeological Shapes. MS Thesis, Arizona State University, 1999
[5] Java View – Interactive 3D geometry and visualization, Nov 2008
Available:
http://www.javaview.de/
[6] Princeton 3D Model Search Engine, Nov 2008
Available:
http://shape.cs.princeton.edu/search.html
[7] Java Web Start – Frequently Asked Questions, Nov 2008
Available:
http://java.sun.com/j2se/1.5.0/docs/guide/javaws/developersguide/faq.html
[8] A. Razdan, D. Liu, M. Bae, M. Zhu, G. Farin, A. Simon, and M. Henderson. Using geometric modelling for archiving and searching 3d archaeological vessels. Proc.Conf.on Imaging Science, Systems and Technology, Las Vegas, June, 2001.
[9] Razdan, A. and Henderson, M. Feature Based Neighborhood Isolation Techniques for Automated Finite Element Meshing, Geometric Modeling for Product Engineering, Editors Josh Turner, Ken Press, North Holland, 1989
[10] TRAC integrated SCM & Project management, Nov 2008
Available:
http://trac.edgewall.org/
[11] Chris Schalk (Author), Ed Burns (Author) and James Holmes (Author), JavaServer Faces: The Complete Reference, McGraw-Hill Osborne Media; 1 edition (August 25, 2006), ISBN-10: 0072262400
[12] Developing Web Applications with Java Server Faces, Nov 2008
Available:
http://java.sun.com/developer/technicalArticles/GUI/JavaServerFaces/
[13] Java 3D Web start example programs, Nov 2008
Available:
http://java.sun.com/developer/technicalArticles/GUI/JavaServerFaces/
[14] NetBeans Integrated Development Environment, Nov 2008
Available:
http://www.netbeans.org/
[15] Tool for Application Developers (TOAD), Nov 2008
Available:
http://www.toadsoft.com/
[16] Konstantinos Moustakas., Georgios Nikolakis and Dimitrios Tzovaras, 3D content-based search using sketches., Springer-Verlag London Limited 2007
[17] Thomas Funkhouser and Patrick Min. A Search Engine for 3D Models, ACM 0730-0301, 2002
[18] Jeremy Rowe, Anshuman Razdan and Arleyn Simon, Acquisition, Representation, Query and Analysis of Spatial Data: A Demonstration 3D Digital Library. IEEE, 2003
[19] C. Faloutsos and R. Barber. Efficient and Effective Querying by Image Content, Journal of Intelligent Information Systems, 231-262 (1994)
[20] Ryutarou Ohbuchi, Masatoshi Nakazawa and Tsuyoshi Takei. Retrieving 3D Shapes Based On Their Appearance, ACM SIGMM, 2003.
[21] Johan W.H. Tangelder and Remco C. Veltkamp. A Survey of Content Based 3D Shape Retrieval Methods, Proceedings of the Shape Modeling International 2004

APPENDIX: 1 Procedure to deploy the web application on the server

The procedure to deploy on a target server involves the following steps.

Location of downloads:
http://bugx.i3dea.asu.edu/projects/VQI2/downloads

1.Install MySQL 5.x database for DBMS.

1.1 Download the MySQL database from the downloads section of the Trac website.

1.2 Install using the downloaded software by following on screen instructions.

2. Create the DB with the script provided

2.1 The SQL script to create the database is available in the downloads section. Download and execute that using MySQL command line tool.

2.2 Check all the tables created in the database.

3. Install Tomcat web server 5.x.

3.1 Copy the file from http://apache.inetbridge.net/tomcat/tomcat-5/v5.5.26/bin/apache-tomcat-5.5.26.exe onto the server.

3.2 Double click on the file and press 'Run'.

3.3 Click on 'Next', read the license, and press 'I Agree'.

3.4 Accept the defaults and press Next.

3.5 You may keep the destination folder the same or point to new location if you would like to.

3.6 Press the Next button.

3.7 Change the administrator user name and password and click on Next.

3.8 It will automatically find the best JRE that suits and click on Install. (If JRE is missing on the source please follow this link to download the latest JRE http://java.sun.com/javase/downloads/index.jsp)

3.9 Press 'Finish' with 'Run Apache Tomcat' selected.

3.10 The tomcat service should now successfully be installed.

4. Deploying the WAR file

4.1 Download the WAR file available in the Downloads section.

4.2 Copy the WAR file to Webapps folder in the Tomcat installation directory.

4.3 Edit the conf\Catalina\localhost\VQI2 XML document to reflect the username and password of the MySQL database installed earlier.

4.4 Restart the Tomcat service.