A Shape-based Searching System for Industrial Components

Chun Pan Cheung
National Institute of Standards and Technology
100 Bureau Drive, Stop 8940
Gaithersburg, MD 20899-8940
1-301-975-3682
bcheung@nist.gov

Afzal Godil
National Institute of Standards and Technology
100 Bureau Drive, Stop 8940
Gaithersburg, MD 20899-8940
1-301-975-4262
godil@nist.gov

Abstract

A web-based application has been developed for shape-based searching for industrial components. The application is based on Flex [2010], which is excellent at providing highly interactive user interfaces, and O3D, which can display 3D models in a web browser efficiently. By combining the two technologies, the shaped-based industrial components searching system provides an effective way for users to search for industrial components and obtains other related information. This highly interactive 3D shape retrieval system is based on Service Oriented Architecture (SOA) that relies on the REST based web-service model to create efficient web application.

Disclaimers

NIST does not endorse or recommend any of the mentioned products, companies, or sites in this paper and such mentions does not imply that the cited products, companies, or sites are better or worse than similar products, companies, or sites.

Keywords

Shape-based searching, O3D, Flex, CAD/CAM

1 Introduction

A web-based application has been developed for shape-based searching for industrial components, as shown in Figure 1. The application is based on Flex which is excellent at providing highly interactive user interfaces and O3D, which can display 3D models in a web browser efficiently. By combining the two technologies, the shaped-based industrial components searching system provides an effective way for users to search for industrial components and obtain other related information. This highly interactive 3D shape retrieval system is based on Service Oriented Architecture (SOA) that relies on the Representational State Transfer (REST) based web-service model to create efficient web application.

Large number of 3D CAD models are created by engineering and manufacturing companies and stored in local repositories. The 3D CAD models provide the high fidelity description as well as the design and manufacturing details of the engineering part. Effective searching of these 3D repositories for 3D industrial components which are similar to a given 3D query model has become an important area of research. Traditional text based search engines are widely used in many 3D repositories to retrieve 3D objects. These text based search strategies only allow users to search 3D models by inputting keywords, file types, metadata or by just browsing the thumbnail of 3D models, which may not meet all the demands. Besides, describing a component often requires the use of jargons, which users may not be familiar with. Moreover, a text based method requires manually annotating the 3D shapes and it may not be able to properly represent the shape of these 3D components. Shape based search methods do not require any manual annotation. The only thing they require is the shape descriptor which can automatically extract shape features and properly represent the shape information

Searching for components across the manufacturer's supply chain efficiently has become a very important issue. Using 3D shape searching early in the design cycle can detect duplicate components and can locate similar components manufactured across supply chain. Hence, there will be cost saving associated with reuse of components by identifying and reusing of the existing designs and manufacturing processes.

One of the main areas of research in 3D Shape retrieval fields is the shape descriptor (feature descriptor). The shape descriptors can be classified into four main categories: view-based, graph-based, statistics-based and transformed-based, or as local and global shape descriptors. Recent work shows that view-based approaches perform better than other approaches for 3D shape retrieval. The view-based approaches can also be used for query-based searching interface, based on depth images, binary images and by 2D/3D sketching.

We have developed different types of shape descriptors for 3D shape retrieval from a CAD database [Dutagaci and Godil 2010; Iyer et al. 2005; Li and Godil 2009; Li and Godil 2008; Lian et al. 2010; Wagan et al. 2008]. For more information about different 3D shape retrieval methods we refer the reader to the survey paper of Tangelder et al. [2004] and for review of different applications the paper of Bustos et. al. [2005].

For industrial components, displaying details on a component is very important. If 2D images of the components are used, each object would require a lot of images, and even then some details might be left out because there is not an image of the component at a certain angle. Thus displaying the components in 3D would be a more suitable choice. The problem is, displaying 3D objects requires a lot of processing power. There are different 3D engines that use GPU for processing the 3D objects, but most of them cannot be used in a web browser. For those web browser-based ones, most of them rely on CPU for processing. The problem with using CPU for processing 3D objects is that it has lower performance than using GPU, and it takes away processing power required by other processes running in parallel. This problem is even more glaring in industrial components searching systems because the 3D objects of the components have to be very detailed, and thus require even more processing power. Thus a web-based 3D engine that uses GPU to process 3D objects is required. O3D is an open-source web API developed by Google [2010]. It uses GPU for 3D object processing, and supports a wide range of graphic cards. Moreover, it is available on Windows, Mac and Linux. Thus it is a suitable choice for displaying industrial components in 3D in web browsers. Flex provides a uniform interactive user interface across all platforms. This paper proposes a web application that combines the use of Flex and O3D (figure 1). In Figures 2a, 2b and 2c, we show the retrieval results visually from the Purdue Engineering Benchmark [2006] for a pipe, fender gearbox, and a bearing block like part.

2 Architecture

In the proposed system, the architecture can be divided into back-end and front-end, as illustrated in figure 3. The back-end is composed of a database for storing information of the industrial components, pre-calculated similarity results between the components for better searching performance, and a web server for providing search results in XML format. The front-end provides a graphical user interface (GUI) for searching and displaying search results. Asynchronous JavaScript and XML (AJAX) is used for communication between the back-end and front-end.

3 Back-end

The back-end of the system is responsible for storing the 3D models and information of the industrial components. It accepts search queries and provides data to the front-end for display. MySQL is used for storing information of components, as well as the information of the providers that carry the components, e.g. name and price of the component, contact method of the providers, etc. Instead of XML, a database is used because pricing and information of providers might change frequently. A database is more efficient at data that requires frequent updates. For demonstration, the CAD database from Purdue University [2010] is used. The 3D Models of the industrial components are saved as O3DTGZ format, which is the format that O3D uses (O3D provides a converter tool to convert DAE format into O3DTGZ format). To increase the performance of searching, similarities between the components are pre-calculated and are saved.

When the web server receives a search query sent by AJAX, it combines the information of components from the database and the pre-calculated similarity files to produce result in XML format. The result is then sent back to the GUI and is rendered accordingly.

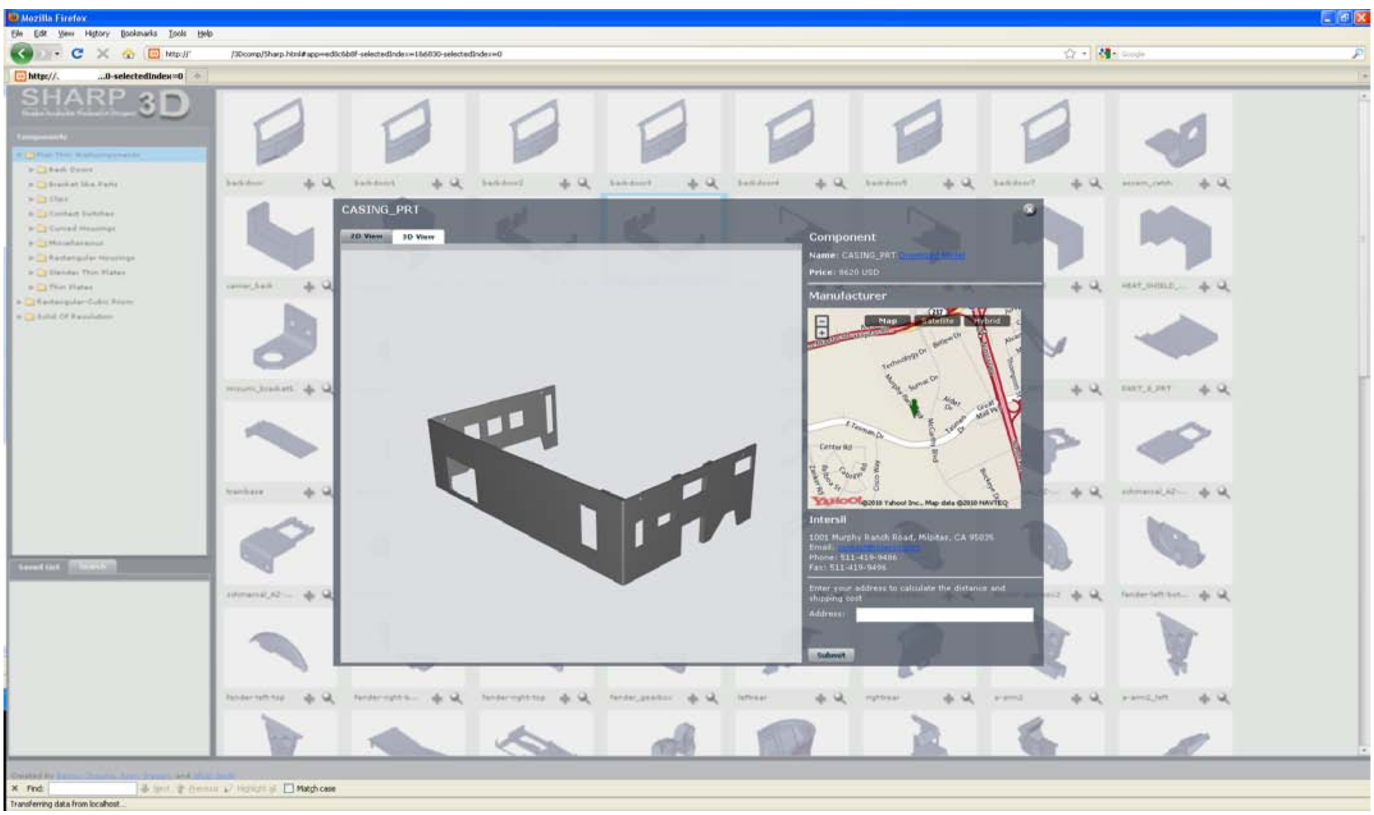

Figure 1: *A web-based industrial component searching system using shape-based searching with 3D display.*

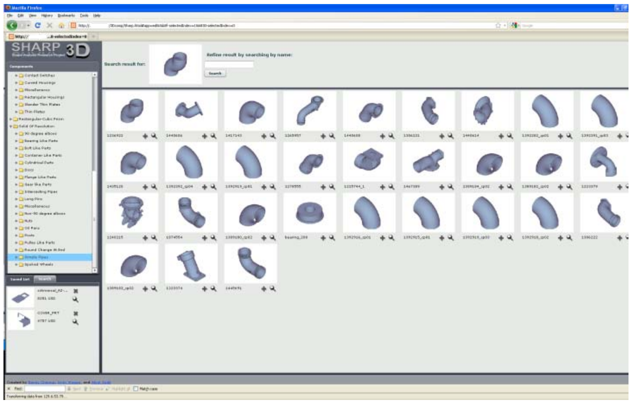

Figure 2a: *Pipe searching results shows components similar in shape to the component displayed on the top.*

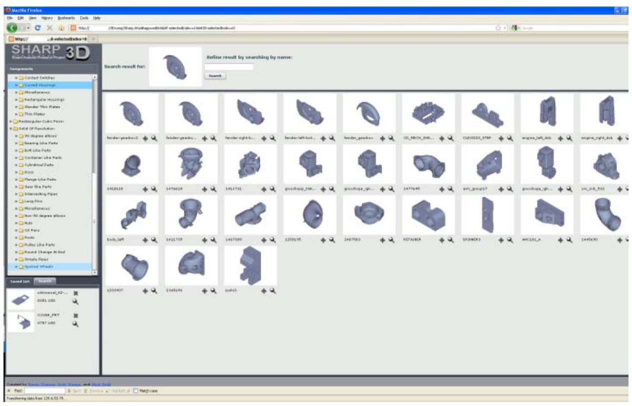

Figure 2b: *Fender gearbox searching results shows components similar in shape to the component displayed on the top.*

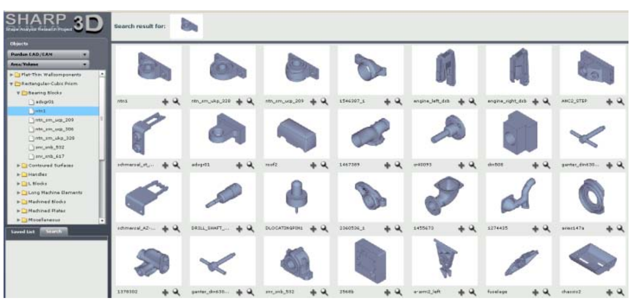

Figure 2c: *Bearing Block searching results shows similar in shape to the component displayed on the top.*

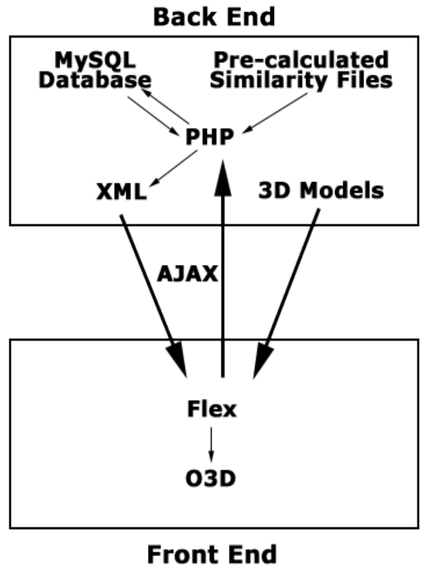

Figure 3: *The architecture of the web-based industrial component shaped-based searching system.*

Figure 4: *Sketch pad for user to search industrial component by sketching.*

4 Front-end

The front-end provides the GUI for users to submit search queries and displaying the result. Instead of relying on HTML, Flex is used in the system for providing the GUI. It is based on Adobe Flash, which is available on all platforms and widely used. It eliminates the problem of different HTML rendering behaviors between different browsers, and thus can deliver a uniform Rich Internet Application (RIA) across all platforms. Besides, it has high accessibility because it can either be installed on a computer or accessed from a web browser. More importantly, Flex facilitates high level of user interactivity. For example, as illustrated in figure 4, the system provides a sketch pad for user to sketch the object they want to search for. Upon the user pressing the search button, the picture is saved and sent back to the server for searching. This would be much harder to implement without the use of Flex.

In the interface, there is an expandable list of components on the left. The list is sorted by categories of the components. Each category can have sub-categories. The list allows easy access to components if the user knows which category the component he is looking for belongs to. If a user clicked on a category, all the components that belong to that category will be displayed on the right side. If a user clicks on a component, a list of components, sorted by similarity in shape to that component, will be displayed as shown in figures 2a, 2b and 2c.

Under the component list, there is a box that has two tabs. As illustrated in figure 5, the saved list tab displays the list of components that a user has saved. It acts as bookmarks to components so that the user can quickly retrieve components that interest him/her. By clicking on the magnifying glass icon, a user can search for components that are similar in shape to that component.

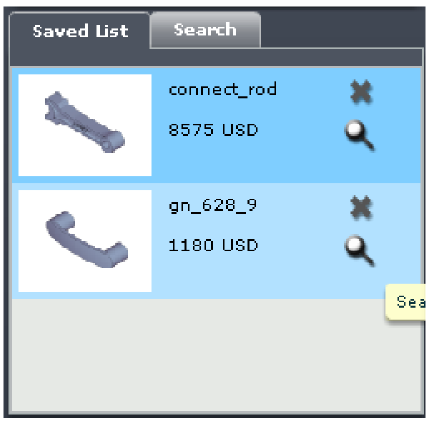

Figure 5: *The saved list that lists components that the user has saved.*

As illustrated in figure 6, the Search tab provides different searching methods. User can search for a component by name, or by sketching as explained above. User can also search by uploading an image. Upon receiving the image, the web server will search for components using a view-based searching algorithm.

If user clicks on a category in the list of components, the right portion of the interface displays a list of components belonging to that category. If a search query is submitted, the right portion will display the result in descending order of similarity. If a user clicks on a component, a pop-up will be displayed as shown in figure 7. On the right side of the pop up, information of the component, e.g. name, price, provider and its address, is listed. A user can also download the 3D model of the component. On the left side, a 2D image of that component will be displayed by default. A user can also switch to 3D view that allows rotations and zooming in/out.

In the previous version of the system, the Flash-based 3D engine was used. Since Flash uses CPU for rendering graphics, the 3D engine does not take advantage of the graphical processing power that GPU nowadays provides. Its 3D display performance is acceptable for less detailed components, but for more detailed components, like those in the Purdue database, its performance is below average. For the more complex objects, it would take a

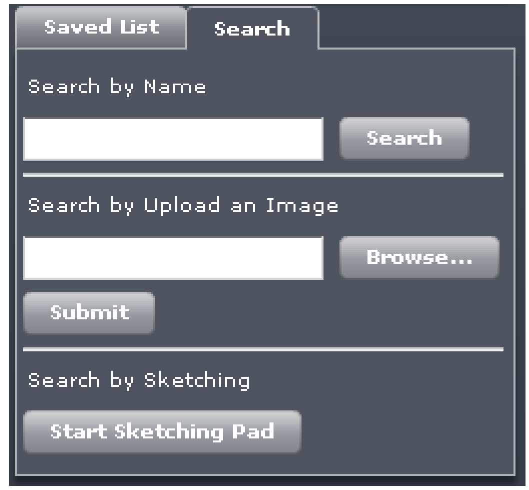

Figure 6: *Different searching methods provided by the system.*

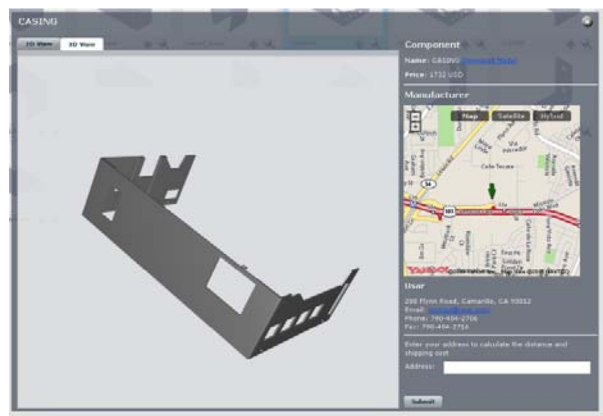

Figure 7: *Pop-up that shows a component's information as well as its 2D/3D view.*

very long time to load, during which the computer would be unresponsive. After the object is rendered, the frame rate became very low when rotating the object. That is the main reason O3D is being adapted.

In the current version of the system, O3D 3D engine is used for displaying 3D objects. Flex has the ability to display a HTML page as a component, and it can also communicate with JavaScript. These two properties are very important for integrating O3D into Flex because O3D is JavaScript based. Although tweaks have to be made to make sure proper unloading of a 3D object, integrating O3D into Flex was not difficult.

O3D uses GPU for 3D graphics processing, and thus its performance is much better than Flash-based 3D engine. For 3D objects that would take over a minute to load under the Flash-based 3D engine, O3D will take only seconds. Object rotation using O3D does not stress out the CPU and is rendered smoothly.

Besides providing good performance, the advantage of using the O3D 3D engine is its accessibility. It is a web browser plug-in that is available to many browsers on different platforms. Besides, it supports a wide range of graphic cards and does not require the latest, most powerful ones to provide good rendering performance.

More importantly, O3D provides high interactivity with the component displayed. For example, it can detect and return the coordinates of where on the component has been clicked. One of the applications of the functionality is to allow users to mark points of interest on a component so that searching by local features can be performed.

5 Conclusion

By combining Flex and O3D, the proposed system provides a highly interactive industrial components searching system. It allows a user to search for components by 3D shape, which is a more effective way of searching than textual search. It can display the components in 3D and allows users to rotate the components and zoom in/out, so that users will be able to look at the details of a component from any direction.

Acknowledgements

This work was supported by the SIMA and IDUS program.

References

ADOBE FLEX. Retrieved April 11, 2010, from Adobe: http://www.adobe.com/products/flex/.

BUSTOS, B., KEIM, A.D., SAUPE, D., SCHRECK, T., AND VRANIC, V.D. 2005. Feature-based similarity search in 3d object databases, *ACM Computing Surveys* 37(4), 345–387.

DUTAGACI, H., AND GODIL, A. 2010. View Subspaces for Indexing and Retrieval of 3D models. in *SPIE conference, 3D Image Processing (3DIP) and Applications,* California, USA.

GOOGLE O3D. Retrieved April 11, 2010, from Google: http://code.google.com/apis/o3d/.

IYER, N., JAYANTI, S., LOU, K., KALYANARAMAN, Y. AND RAMANI, K. 2005. Three dimensional shape searching: State-of-the-art review and future trends, *Computer-Aided Design* 37 (2005) (5), 509–530.

JAYANTI, KALYANARAMAN, Y., IYER, N., AND RAMANI, K. 2006. Developing an Engineering Shape Benchmark for CAD Models, *Computer-Aided Design*, Vol. 38, No. 9, 2006, 939-953.

LI, X., AND GODIL, A. 2009. Exploring The Bag-of-Words Method For 3D Shape Retrieval. in *IEEE International Conference on Image Processing - ICIP 2009,* Cario, Egypt.

LI, X., GODIL, A., AND WAGAN, A. 2008. Visual Based 3D CAD Retrieval Using Fourier Mellin Transform. in *IEEE International Conference on Shape Modeling and Applications,* (New York, USA).

LIAN, Z., GODIL, A., AND SUN, X. 2010. Visual Similarity based 3D Shape Retrieval Using Bag-of-Features. in *IEEE 2010 Shape Modeling International Conference,* Aix-en-Provence, France.

CAD DATABASE. Retrieved April 11, 2010, from Purdue University http://shapelab.ecn.purdue.edu/.

TANGELDER, J.W.H., AND VELTKAMP, R.C. 2004. A survey of content based 3d shape retrieval methods. i*n: Proceedings of the Shape Modeling International, pp. 145–156*.

WAGAN, A., GODIL, A., AND LI, X. 2008. 3D CAD Retrieval using partitioning the objects. in *IEEE International Conference on Shape Modeling and Applications,* New York, USA.