Global Journal of Computer Science and Technology
Volume 12 Issue 8 Version 1.0 April 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

3D Searching

By Pranav Agarwal
Pune University MAEER's MIT COE, India

Abstract - As the number of 3D models available on the Web grows, there is an increasing need for a search engine to help people. Unfortunately, traditional text-based search techniques are not always effective for 3D data. The key challenges are to develop query methods simple enough for novice users and matching algorithms robust enough to work for arbitrary polygonal models. We present a web-based search engine system that supports queries based on 3D sketches, 2D sketches, 3D models, and/or text keywords. We also present a web-based search engine system that supports multimodel queries which include both text query and sketch query. This results in faster retrieval of the result and the percentage efficiency also increases. The net result is a growing interactive index of 3D models available on the Web (i.e., a Google for 3D models).

Keywords : *Search engine, sketch query, text query, multimodel query, teddy, sketch, repository.*

GJCST Classification: *H.3*

Strictly as per the compliance and regulations of:

3D Searching

Pranav Agarwal

Abstract - As the number of 3D models available on the Web grows, there is an increasing need for a search engine to help people. Unfortunately, traditional text-based search techniques are not always effective for 3D data. The key challenges are to develop query methods simple enough for novice users and matching algorithms robust enough to work for arbitrary polygonal models. We present a web-based search engine system that supports queries based on 3D sketches, 2D sketches, 3D models, and/or text keywords. We also present a web-based search engine system that supports multimodel queries which include both text query and sketch query. This results in faster retrieval of the result and the percentage efficiency also increases. The net result is a growing interactive index of 3D models available on the Web (i.e., a Google for 3D models).

Keywords : Search engine, sketch query, text query, multimodel query, teddy, sketch, repository.

I. Introduction

Over the last few decades, computer science has made incredible progress in computer aided retrieval and analysis of multimedia data. For example, suppose you want to obtain an image of a horse for a Power point presentation. A decade ago, you could: 1) draw a picture, 2) go to a library and copy a picture, or 3) go to a farm and photograph a horse. Today, you can simply pick a suitable image from the millions available on the web. Although web search is commonplace for text, images, and audio, the information revolution for 3D data is still in its infancy. However, three recent trends are combining to accelerate the proliferation of 3D models, leading to a time in the future when 3D models will be as ubiquitous as other multimedia data are today: (1) new scanners and interactive tools are making construction of detailed 3D models practical and cost effective, (2) inexpensive graphics hardware is becoming faster, causing an increasing demand for 3D models from a wide range of people, and (3) the web is facilitating distribution of 3D models.[1]

a) Need for 3d Search Engine

Now a days, developments are changing the way we think about 3D data. For years, a primary challenge in computer graphics has been how to construct interesting 3D models. In the near future, the key question will shift from "how do we construct them?" to "how do we find them?". For example, consider a person who wants to build a 3D virtual world representing a city scene. He will need 3D models of cars, street lamps, stop signs, etc. Will he buy a 3D modeling tool and build them himself? Or, will he acquire them from a large repository of 3D models on the Web? We believe that research in retrieval, matching, recognition, and classification of 3D models will follow the same trends that can already be observed for text, images, audio, and other media. An important question then is how people will search for 3D models. Of course, the simplest approach is to search for keywords in filenames, captions, or context. However, this approach can fail: (1) when objects are not annotated (e.g., "B19745.wrl"), (2) when objects are annotated with inspecific or derivative keywords (e.g., "yellow.wrl" or "sarah.wrl"), (3) when all related keywords are so common that the query result contains a flood of irrelevant matches (e.g., searching for "faces" – i.e., human not polygonal), (4) when relevant keywords are unknown to the user (e.g., objects with misspelled or foreign labels), or (5) when keywords of interest were not known at the time the object was annotated. In these cases and others, a 3D search engine is needed.[1]

b) How to Search For 3d Models

We hypothesize that shape-based queries will be helpful for finding 3D objects. For instance, shape can combine with function to define classes of objects (e.g., round coffee tables). Shape can also be used to discriminate between similar objects (e.g., desk chairs versus lounge chairs). There are even instances where a class is defined entirely by its shape (e.g., things that roll). In these instances, "a picture is worth a thousand words." Our work investigates methods for automatic shape-based retrieval of 3D models.

The challenges are two-fold. First, we must develop computational representations of 3D shape (shape descriptors) for which indices can be built and similarity queries can be answered efficiently. In this paper, we investigate combinations of 3D sketching, 2D sketching, text, and interactive refinement based on shape similarity. We have integrated these methods into a search engine that provides a publicly available index of 3D models on the Web (Figure 1.1).

Author : Computer Department, Pune University MAEER's MIT COE, India. E-mail : Pranav.pranaw@gmail.com

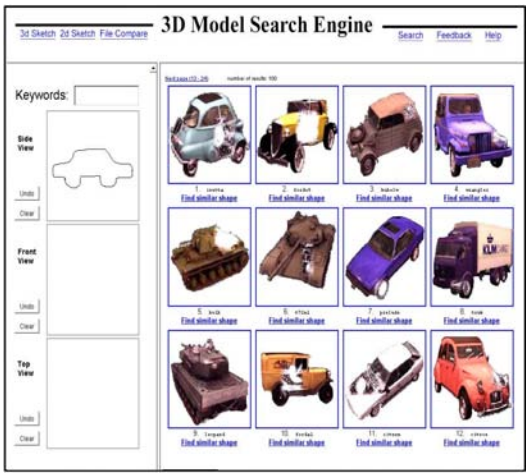


Fig. 1.1: Screenshot of our search engine

It allows a user to specify a query using any combination of keywords and sketches (left). Then, for each query, it returns a ranked set of thumbnail images representing the 16 best matching 3D models (right). The user may retrieve any of the 3D models by clicking on its thumbnail, and/or he may refine the search by editing the original input or by clicking on the "Find Similar Shape" link below any thumbnail.[1]

II. System Overview for 3D Models

The organization of our system is shown in Figure 2.1. Execution proceeds in four steps: crawling, indexing, querying, and matching. The first two steps are performed off-line, while the last two are done for each user query. The following text provides an overview of each step and highlights its main features:

1) Crawling: We build a database of 3D models by crawling the Web. 3D data still represents a very small percentage of the Web, and high quality models represent an equally small percentage of all 3D data. So, we have developed a focused crawler that incorporates a measure of 3D model "quality" into its page rank. Using this crawler, we have downloaded 17,834 VRML models from the Web. We augment this database with 2,873 commercial models provided by 3D vendors.
2) Indexing: We compute indices to retrieve 3D models efficiently based on text and shape queries. In particular, we have developed a new 3D shape descriptor based on spherical harmonics that is descriptive, concise, efficient to compute, robust to model degeneracies, and invariant to rotations.
3) Querying: We allow a user to search interactively for 3D models. Our system supports query methods based on text keywords, 2D sketching, 3D sketching, model matching, and iterative refinement. We find that methods based on both text and shape combine to produce better results than either one alone.
4) Matching: For each user query, our web server uses its index to return the sixteen 3D models that best match the query. Our method answers 3D shape queries in less than a quarter of a second for our repository; and, in practice, it scales sub-linearly with the number of indexed models. The main research issue at the heart of this system is how to provide shape-based query interfaces and matching methods that enable easy and efficient retrieval of 3D models from a large repository. In the following two sections, we discuss these issues in detail for different query interfaces.[1]

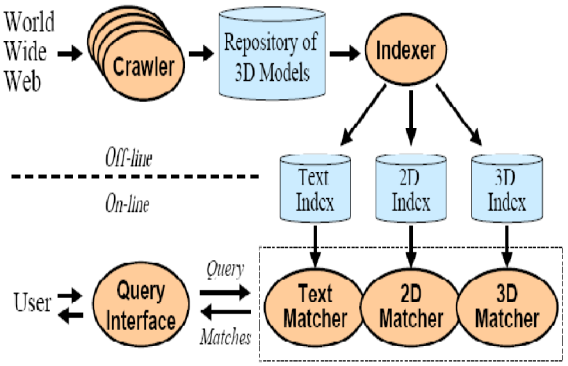


Figure 2.1: System Organization

a) Sketch Query

Of course, shape similarity queries are only possible when the user already has a representative 3D model. In some cases, he will be able to find one by using a text search. However, in other cases, he will have to create it from scratch (at least to seed the search). An interesting open question then is "What type of modeling tool should be used to create shapes for 3D retrieval queries?". This question is quite different than the one asked in traditional geometric modeling research. Rather than providing a tool with which a trained user can create models with exquisite detail and/or smoothness properties, our goal is to allow novice users to specify coarse 3D shapes quickly. In particular, the interface should be easy to learn for first time visitors to a website. Of course, this requirement rules out almost every 3D modeling tool available today – i.e., it would not be practical to require everybody who wants to use a 3D search engine to take a three week training course to learn the complicated menu structure of a commercial CAD tool. Instead, we have investigated two alternatives.

The first approach is to specify shape queries with a simple 3D sketching tool, such as Teddy [2] or Sketch [3]. To investigate this approach, we have developed a query interface in which the user creates a simple 3D model with Teddy, and then the system retrieves similar models (see Figure 3.1).

Unfortunately, our early experiences suggest that even its simple gesture interface is still too hard for

novice and casual users to learn quickly. During informal studies, we observed that most people do not readily understand "extrusions" and "cuts," and they have a difficult time getting used to rotating a 3D model to get the proper viewpoint for modeling operations. Moreover, only certain types of shapes can be created with Teddy. We believe that making 3D tools even simpler would require further constraints on the types of shapes that could be produced. Thus, we were motivated to look for alternate sketching paradigms.[2][3]

Fig 3.1: 3D sketch query interface

Our second approach is to draw 2D shapes with a pixel paint program and then have the system match the resulting image(s) to 2D projections of 3D objects (Figure 3.2). The main advantage of this approach is that the interface is easy to learn. All but the most novice computer users have used a 2D paint program before, and there are no complicated viewing or manipulation commands. Of course, the main disadvantage is that 2D images generally have less shape information than 3D models. We compensate for this factor somewhat by allowing the user to draw multiple 2D projections of an object in order to better define its shape.

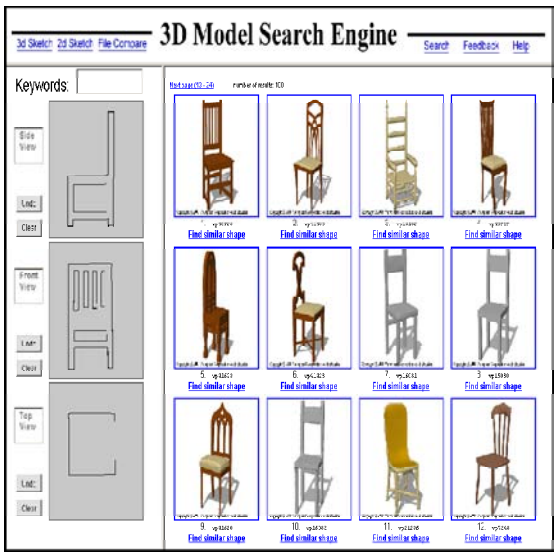


Fig. 3.2: 2D sketch query interface

b) Text Query

Our system also supports searching for 3D models by matching keywords in their textual descriptions. To support this feature, we construct a representative document for each 3D model. The text in that document includes the model filename, the anchor and nearby text parsed from its referring Web page, and ASCII labels parsed from inside the model file. Each document is preprocessed by removing common words (stop words) that don't carry much discriminating information, such as "and", "or", "my", etc. We use the SMART system's stop list of 524 common words as well as words specific to our domain (e.g. "jpg", "www", "transform", etc.). Next, the text is stemmed (normalized by removing inflectional changes) using the Porter stemmer. Finally, synonyms of the filename (without the extension) are added using Word-Net.

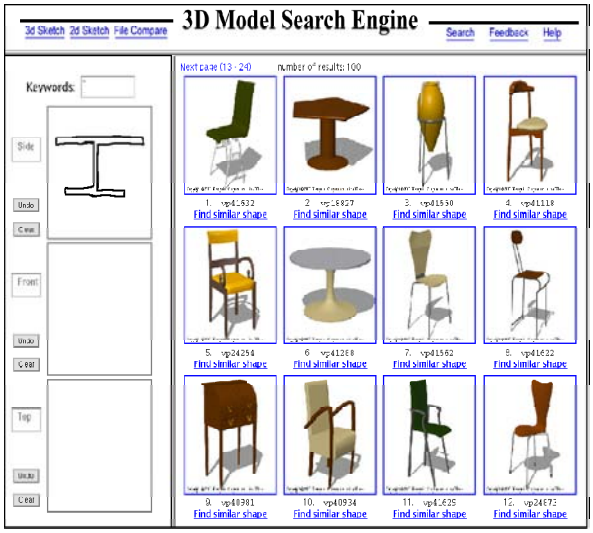
In order to match documents to user-specified keywords or to other documents, we use the TF-IDF/Rocchio method [5], a popular weighting and classification scheme for text documents. This method assigns a similarity score based on a term's frequency in the document and its inverse frequency over all documents.

Fig 3.3: Text interface

c) Multi-Model Query

Since text and shape queries can provide orthogonal notions of similarity corresponding to function and form, our search engine allows them to be combined. We support this feature in two ways. First, text keywords and 2D/3D sketches may be entered in a single multimodal query. Second, text and shape information entered in successive queries can be combined so that a user can refine search terms adaptively. For instance, if a user entered text keywords in a first query, and then clicked a "Find Similar Shape" link, the text and 3D shape would combine to form a second query. These types of multimodal queries are often helpful to focus a search on a specific subclass of objects (Figure 3.4). For example, a query with only keywords can retrieve a class of objects (e.g., tables), but it is often hard to home in on a specific subclass with text alone (e.g., round tables with a single pedestal). Similarly, a query with only a sketch can retrieve objects with a particular shape, but it may include objects with different functions (e.g., both tables and chairs). Multimodal input can combine ways of describing objects to form more specific queries (Figure 3.4(c)).[1]

(a) Text Query



(b) 2D sketch query

(c) Multimodal query

Fig. 3.4: Multimodal queries are often effective at finding specific types of objects.

III. Limitation

Better 2D image matching methods: our 2D sketching interface would be more effective with better image matching algorithms. Sometimes users create query sketches with interior texture and/or details (e.g., eyes and mouth of a human face), and our search engine matches them with projected images containing only boundary outlines (e.g., just the outline of the face). For matching purposes, the interior details in sketches are "interpreted" as boundaries of holes in projected images, and unexpected results are sometimes returned to the user. Of course, this problem could be rectified somewhat by providing users with instructions or examples about how to draw their sketches.

New modeling tools: future 3D modeling systems should consider integrating shape based matching and retrieval methods into interactive sketching tools. For instance, consider a 3D model synthesis paradigm in which a user draws a rough sketch of a desired 3D model and the system "fills in the details" semi-automatically by suggesting matching detailed parts retrieved from a large database. In such a paradigm, the user could retain much of the creative control over model synthesis, while the system performs most of the tedious tasks required for providing model detail.

IV. Conclusion

In summary, it investigates issues in building a search engine for 3D models. The main research contributions are: (1) New query interfaces that integrate text, 2D sketches, 3D sketches, and 3D models. (2) We provide a large repository of 3D models and a way to find the interesting ones.

References Références Referencias

1. Thomas Funkhouser, Patrick min, Michael Kazhdan, Joyce Chen, Alex Halderman, and David Dobkin Princeton University, "A Search Engine for 3D Models" ACM Transactions on Graphics, Vol. V, No. N, 10 202002,.
2. Zeleznik, R. C., Herndon, K. P., and Hughes, J. F. 1996. Sketch, "An Interface for Sketching 3D Scenes", In Proceedings of SIGGRAPH 96. Computer Graphics Proceedings, Annual Conference Series. 163–170.
3. Igarashi, T.,Matsuoka, S., and Tanaka, H. 1999. Teddy: "A Sketching Interface for 3D Freeform Design". In Proceedings of SIGGRAPH 1999. Computer Graphics Proceedings, Annual Conference Series., Los Angeles, CA, 409–416.

Global Journal of Computer Science and Technology Volume XII Issue VIII Version I April 2012
14

This page is intentionally left blank