University of Huddersfield

University of Huddersfield Repository

Pein, Raoul Pascal, Amador, Milton, Lu, Joan and Renz, Wolfgang

Using CBIR and semantics in 3D-model retrieval

Original Citation

Pein, Raoul Pascal, Amador, Milton, Lu, Joan and Renz, Wolfgang (2008) Using CBIR and semantics in 3D-model retrieval. In: 2008 8th IEEE International Conference on Computer and Information Technology. IEEE, pp. 173-178.

This version is available at http://eprints.hud.ac.uk/3617/

Using CBIR and Semantics in 3D-Model Retrieval

Raoul Pascal Pein[1,2], Milton Amador[3], Joan Lu [1], and Wolfgang Renz [2]

[1] Department of Informatics, School of Computing and Engineering, University of Huddersfield, Queensgate, Huddersfield HD1 3DH, United Kingdom , <R.P.Pein@hud.ac.uk> <j.lu@hud.ac.uk>
[2] Multimedia Systems Laboratory (MMLab), Faculty of Engineering and Computer Science, Hamburg University of Applied Sciences, Berliner Tor 7, 20099 Hamburg, Germany , <pascal.pein@haw-hamburg.de> <wr@informatik.haw-hamburg.de>
[3] European Aeronautic Defence and Space Company EADS Innovation Works, Dept. IW-SE, 81663 Munich, Germany, <Milton.Amador@eads.net>

Abstract

This paper proposes a generic design for 3D model retrieval. It has been developed in cooperation with EADS, which deals with Computer Aided Design / Computer Aided Engineering (below CAD/CAE) data through the product life cycle. Sharing these models with users across the enterprise is a challenging task. CAD/CAE models may be hundreds of megabytes large and stored in proprietary formats. Browsing and previewing this data efficiently requires new tools.

One way to leverage the collaboration in and outside of CAD domains is to offer through a repository an access to a neutral 3D data format. Also functionalities for semantic enrichment of 3D models, for the retrieval of context sensitive information and for 3D model retrieval based on 3D similarity search and CBIR techniques should be provided.

1. Introduction

Within the development process of huge enterprises there is an increasing need for improved capabilities in the field of retrieving CAD/CAE data and an easy way to preview CAD in heterogeneous environments, without using a CAD application or other engineering tools. Despite all the interfaces and applications available for exchanging data between CAD applications and other common desktop applications, 3D data created with CAD tools is difficult to share with other users across the enterprise. Existent Product Data Management (*PDM*) systems and CAD Management tools still do not offer a global and open approach for visualisation and retrieval of CAD data in a collaborative way. There are also no appropriate tools for a 3D similarity search in distributed CAD data sources. The search of 3D models is becoming an increasingly important task in heterogeneous CAD environments. Especially for CAD applications the identification of similar 3D models can help to reduce the costs of developing new 3D models by maximizing the reuse of existing models.

One way to leverage the collaboration in and outside of CAD domains is to offer through a repository an access to neutral 3D data converted from CAD data. The conversion process can take place on server side, where 3D models are maintained, or on CAD source side, where the CAD data is available. The advantage of this approach lies in the fast transfer of 3D model data through the network. The size of a strong compressed neutral data format can reach nearly a hundredth size of a native CAD file. In addition to this approach, the retrieval of 3D model data out of the repository is also essential.

In 2007 EADS Innovation Works generated in collaboration with the University of Applied Sciences in Hamburg a concept for collaborative working within a CAD centric engineering environment and illustrated this concept with a demonstrator end of 2007 [15]. Special focus was set on CAD data previews, semantic enrichment of 3D models, the retrieval of context sensitive information and 3D model retrieval based on 3D similarity search.

In the last decade much research effort has been put into content-based image retrieval (below *CBIR*) [7, 22, 13].

Many new technologies have been developed to handle the increasing amount of digital images. The arising need for 3D model retrieval (below *3DMR*) solutions induced the development of another retrieval branch [10, 8, 5].

Like images the nature of 3D models is very difficult to capture in means of common retrieval solutions. Inverted indexes which are widely used in text search engines are not applicable in this area. As 3D models are usually constructed, they potentially contain more meta information than pixel images (see section 3). Nevertheless there are still many model features which are not tangible directly. Here the most convenient approach is to adopt the feature vector paradigm.

Several different feature vector models have been developed and also compared [28]. The next step was the combination of several features in parallel. This is necessary to overcome the potential lack of differentiability in independent feature spaces. The analysis of multiple features suggested an increasing result quality when using multiple features in the same query [23].

It is remarkable that the feature vector paradigm is a very universal approach. Basically there is absolutely no more difference between naive implementations of CBIR and 3DMR retrieval engines. Both engines work on a repository of extracted features to calculate similarities between the query object and the available content. Yet, a closer view again reveals deviations between these worlds. Query composition, browsing and the result representation require techniques aligned to 3D specifics.

This paper discusses a possible design for a 3DMR program based on a previously published CBIR framework [20], the EADS demonstrator [15] and a newly developed query language [**?**].

2. Related Work

Several different 3DMR engines have already been presented. Similar to CBIR systems many engines have seemingly been developed to test a predefined single feature vector. The survey by Tangelder et al. [26] covers several different feature extraction methods and describes their basic advantages and disadvantages.

A group at the University of Konstanz did a lot of research in the area of 3D retrieval over the last years. Bustos et al. [2, 3] systematically surveyed retrieval effectiveness and efficiency of many available feature vectors. Vranić developed a web enabled demo allowing to compare the quality of these different features [27, 28] working with the same interface. Further a recent dissertation by Schreck [23] based on the previous findings concludes that it is often beneficial to combine multiple features in a single query. To achieve a visible improvement, the features need to be independent from each other. These low level features can be evaluated by using the Princeton Shape Benchmark [24].

A major problem of CBIR and 3DMR is to close the semantic gap. A recent survey by Liu et al. [13] indicates that current approaches are still far away from being successful, but they are able to gradually narrow down the gap.

The search engine by Liu and Razdan [12] is specialized on finding digitizes Indian pottery. It analyses 3D model information from laser scanners and geometric modelling. The extracted shape feature information is then stored in an XML schema. It is intended to be used by archaeologists.

A different approach has been chosen by Parikh et al. [19]. Here the challenge is to assemble a complex object from its broken pieces. The common feature vector approach does not work in this situation because the global similarity of two objects is unimportant. It is necessary to analyse the breaking edges of each part. This is more a key/lock or jigsaw problem instead of a simple retrieval based on global similarity. Interestingly technologies from 2D applications have been adopted successfully in this research.

3. From 2D to 3D

In order to examine the relationship between CBIR and 3DMR it is essential to know the different ways of data representation. The four major classes are compared in table 1. Basically both 2D and 3D objects can either be sampled or constructed.

	2D	3D
sampled	pixels	voxels
synthetic/constructed	vectors	models/meshes

Table 1. Classification of 2D/3D Data

The sampled formats are usually created by capturing an arbitrary real life object (e.g. photographs, magnetic resonance tomography). These images usually contain noise and details are inaccurate because of the restricted resolution. A computer cannot understand the actual meaning of the content. It is simply a data matrix with arbitrary values. Here the basic technologies from CBIR are the best way to handle large amounts of information.

A completely different approach is the use of synthetic/constructed objects. Vector graphics and CAD models are computer generated, don't contain environmental noise and are by design exact and structured. They may even contain additional meta information like part names, hierarchical scene graphs, textures and physical parameters. This significantly boosts machine readability. Computers are able to decompose complex structures correctly. Dependent on the effort designers put into annotation, techniques similar to full text search engines are feasible. Exceptions

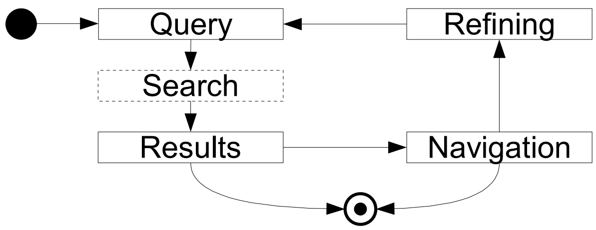

Figure 1. Retrieval-Workflow

are simple polygonal surface meshes which could contain holes or intersecting polygons.

Further the classes from table 1 can be transformed into each other. The way down or to the right is very difficult and requires additional assumptions about the missing information. Much easier and less error prone is the way back to more basic formats. Here all the important information is already available and it's the task of an algorithm to drop irrelevant information. From right to left one dimension is eliminated and from bottom to top the structured information is lost.

It can be useful to discard parts of the original information. It is essentially the same as CBIR systems do when extracting certain features. In addition all retrieval techniques available to low levels are also available for high levels. The 3D descriptor from Chen et al. [5] for example tackles the 3D problem by reducing the model to a set of 2D silhouettes from different angles whose shapes are compared in the retrieval. The voxel based feature vector by Vranić [27] also performs a transformation on the original model.

4. Design Guidelines

In the following a basic design of a 3DMR system is described. Many parts can be constructed very similar to the underlying CBIR framework [20]. For this reason only the differences are emphasized. To solve the additional issues with 3D models, multiple solutions are proposed.

4.1. Retrieval Workflow

The retrieval of any document follows a basic work flow (fig. 1). First the user composes a query which contains a description of the information he wants. This query is then sent to the search engine and the user is waiting for the results. The search engine then interprets the request to perform the retrieval as wished. After a set of results is generated the control is passed to the user again.

If the query was precise enough the desired information should be displayed in a prominent position. The user can examine the results and pick the correct documents easily. In this case the retrieval was successful.

If the user does not spot the desired information directly, he needs to browse the elements of the result set more precisely. In many cases the correct document did not gain the highest rank, because other documents were estimated to be closer to the query from the retrieval engine. Result sets which are too large to be browsed manually or obviously do not contain the correct documents are perfectly possible. Then the user needs to refine the previous query or compose a completely new one. The incorrect result set may give hints how to improve the next iteration of the retrieval.

4.2. Query Composing

In order to start a search the basic concepts of the documents to be retrieved are in the mind of the user and he needs appropriate tools to tell the engine what to do. Full text search engines allow the user to enter a query string containing certain keywords. It is also possible to search on the document semantics. But this solution requires sufficiently annotated documents to be efficient.

CBIR systems often don't support high-quality text based retrieval. Every annotation requires additional effort. Even the techniques to extract information from text with embedded images like VIPS [4] have their restrictions. The text has to be written by somebody and the extraction algorithm has to assign the correct keywords to the images.

Instead they usually offer to specify a *query-by-example* or *query-by-sketch* from which the features are extracted. The examples can be retrieved by providing a set of random images from the repository or by uploading an external image. A query language able to capture proprietary CBIR feature vectors has been developed on the basis of a full text query language [**?**].

The same techniques are applicable for the use of 3D examples. The new problem arises when creating a *query-by-sketch*. Instead of a simple 2-dimensional pixel/vector image the sketch needs to be 3-dimensional. In some cases it is still possible to provide a 2D sketch to the engine (e.g. [8, 5, 16]). Unfortunately, this approach is only applicable with features based on silhouettes. Volumetric calculations cannot be captured this way.

Thus an easy-to-use editor for a 3D model is required. Early adoptions like "Teddy" [11] were already quite powerful. Teddy allows to sketch 3D models quite fast by inflating 2D polygons into the third dimension and by providing several tools for cutting, extruding and other 3D related operations based on gestures. More recent technologies are "SKETCH" [29] and FiberMesh [17] which provide similar operations. All of them are designed to be intuitive and to hide the mathematical background.

4.3. Navigation

Dependent on the size of the result set, browsing is more or less straightforward. The simplest approach is to generate a one dimensional list of all relevant documents where the hits may be sorted by relevance. The most important documents should be located in the first few positions.

At this point, a problem of 3DMR emerges. It is difficult to create an appropriate preview of each model allowing the user to find the correct hits. The models need a textual description, a 2D representation from a certain angle or even an interactive lightweight 3D viewer. Each kind of preview has certain drawbacks. A text is only available, if it had been entered by someone. 2D images are easily computed and integrated, but how does the engine find the perfect camera position? Capturing thumbnails of multiple view points (e.g. x/y/z-axis and an arbitrary slant view) and zooming to fit could solve this issue in many cases. The approach to have a complete 3D viewer for each model might still overextend the user's computer.

If only a single dimension of sorting is available, it is crucial to use the correct sorting while only the query itself can give the engine the required hints. For this reason several approaches to result clustering have been developed, e.g. Carrot[2] [25]. This kind of post-processing arranges the results in a short and concise tree, sorted by topics where each leaf contains a sub-result. Similar techniques are also possible in 3DMR environments. Here the clusters can be based on low-level model features, a composition tree or higher level semantics.

Because of the multi dimensional nature of a feature based retrieval it is also possible to generate a multi dimensional result space [21]. Similar items are clustered in higher dimensions depending on the feature vector applied. Displaying more than 3 dimensions simultaneously is rather impossible and workarounds may be very confusing. Thus the user needs some control about the most important dimensions actually used.

4.4. Repository Management

Every reading access to a data repository can be interpreted as a search query [15]. In the simplest case all available documents are listed but in most cases the result is already filtered. It does not matter whether the query is actually represented by a folder name (like in file systems), a category or a complex query including wild cards, boolean operators and such. This allows users to generate arbitrary lists of documents fitting the current task.

As a consequence the index structures need to be highly flexible. The text based content can be easily handled by common full text search engines and semantics are covered by Semantic Web techniques. Indexing the 3DMR content

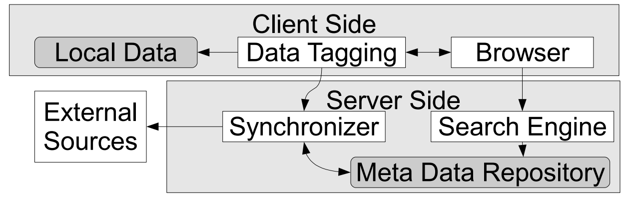

Figure 2. OntoCollab-Design

requires most attention. Like in CBIR, high dimensional feature vectors depend on well adapted index structures in order to achieve a high performance. For average features, generic solutions like R-Trees [9] may be sufficient.

Further it is advisable to have an automated background process to keep the index consistent with the original data.

4.5. Annotation

Annotating the data added by users is a crucial part of the overall search engine. Sound annotated documents are easier to retrieve. Having no additional annotation the engine only analyses the content automatically and the index may be missing valuable meta information. In this case the low level features are the best bet to retrieve a desired document. Additionally it should be tracked which user created or changed a document, adding another way to browse the repository (i.e. by user, department, etc.).

For this reason it should be simple to do this annotation. A straightforward method is to use tags for each document [1]. A set of meaningful prepared tags (e.g. projects, location, task) allows many ways to filter and browse the content. Further it should be able to annotate the documents at any time and by any competent person similar to a wiki. Maybe a user finding a 3D model or an image has additional knowledge to be contributed.

5. OntoCollab Case Study

The design guidelines proposed in section 4 have been applied to the OntoCollab study at EADS [15]. This study is based on the real needs of a big company. Below the basics of the demonstrator are introduced. The prototype design (fig. 2) follows the previously published framework [20] which is responsible to store and manage the repository.

5.1. Motivation

In early stages of aircraft development there are several design issues within the cabin configuration and a huge number of systems, tools and respective data involved in distributed environments. The OntoCollab study is focussed on the support of the cabin configuration process for fast context sensitive information retrieval and the association and consolidation of distributed proprietary data sources on a meta data level. It highlights the benefits of using an ontology based common repository in a CAD field by using open standards, specially XML. The OntoCollab approach shall offer all participants (engineers, project managers, designers etc.) a collaborative way of sharing data and information across separate departments.

The promising advantages for future applications are a platform and application independent handling of distributed product data for pre-investigations and design in multidisciplinary projects within the aircraft product development. The OntoCollab approach is divided in four main researched areas:

- Metadata Repository
- GUI with Metadata and Topic Map Browser
- 2D/3D Viewer (for neutral models)
- Semantic Search Engine (including 3D similarity search)

5.2. Query Composing

The OntoCollab *Search Engine* understands some basic search queries which are composed in the *Browser*. The communication protocol applied is based on OpenSearch [18] and allows to choose from multiple query languages. A full text retrieval is based on the Lucene engine [14] and the integrated 3DMR engine uses an extremely plain language which only contained the name of a single model. The Topic Map Browser accesses the repository via XQuery. A meta language based on XML provides the ability to use multiple language simultaneously where the sub results are merged afterward. The successor of this meta language is the query language recently developed[**?**].

5.3. Annotation

To manage and collect the required meta data from distributed sources several approaches have been developed within the OntoCollab study.

The manual linking and annotation of categorised neutral 3D models is one of the most important topics in this study. This approach provides methods related to the association of data with 3D models via manual linking in the *Data Tagging* module.

The user shall have the possibility to link e.g. documents manually to 3D models related to a specific project or category by drag&drop. Link information and meta data of document is held server-sided and can be retrieved by other users. The user shall also have the possibility to add notes to categorised 3D objects, and also to modify and delete them, if required.

5.4. Repository Management

Meta data extraction, data indexing and the transfer process from distributed data sources is also an important issue. This approach provides functionalities for data indexing by extracting specific, required meta data from distributed data sources including CAD data, documents, drawings etc..

This approach also considers the connecting of distributed data sources (systems, databases etc.) to the repository for meta data transfer via Service Oriented Architecture (*SOA*). SOA-services shall trigger the extraction processes and transfer the required meta data from distributed systems, e.g. from CAD systems, clients, databases etc., to the common repository. Needed conversions and periodic updating of meta data are being processed automatically by SOA-adaptors, the *Synchronizer*.

5.5. Navigation

Via a GUI the user shall be able to retrieve respective meta data held in the repository. He shall also reference data by a drag&drop functionality into a personalized or given shared folder structure. This shared folder structure shall be defined by the user or a user group individually.

The delivery of topic maps [6] for accessing associated information of distributed data is also considered in the OntoCollab approach. Topic maps shall offer the user the possibility to navigate semantically through meta data elements with respective relations.

To display the results of a 3DMR query, prepared thumbnail images have been created of each model. The use of four sub images, each one showing a different angle was the most accepted solution. A 3D browser showing a large scale model of each model shall be available directly from the results.

6. Conclusion

Achievements In this paper we presented a design to build powerful 3DMR systems. It is not attempted to bridge the semantic gap rather than to introduce a way to merge the results of current research on both sides. The search engine should be able to understand low level features as well as high level ones. The control which features are used in a retrieval should be an optional choice for the user. Sometimes a search engine is not powerful enough and users need a comfortable way to browse the content.

Problems Remaining Of course this approach does not solve all the problems connected with feature vector based search engines. The features are highly compressed and thus inaccurate. The feature similarity does often deviate from the human perception. Also the problem of indexing structures is still not solved for arbitrary feature vectors and similarity measures. Further result sets are ”fuzzy” and not clearly separated into hit/miss.

Many of these problems would be unimportant, if users would be able and willingly to annotate each document or model sufficiently. Thus it is necessary to give users the appropriate tools to do annotation quick and easily.

Future Work In following projects it is planned to enhance the CBIR framework to fully support an kind of data type by using the feature vector paradigm. Further it is necessary to find generic and fast solutions to solve the indexing problem in order to improve scalability.

References

[1] J. Barrett. Emblems and Tagging in GNOME, 2006.

[2] B. Bustos, D. Keim, D. Saupe, T. Schreck, and D. Vranic. An experimental comparison of feature-based 3D retrieval methods. *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, pages 215–222, Sept 2004. 1335197.

[3] B. Bustos, D. A. Keim, D. Saupe, T. Schreck, and D. V. Vranić. Feature-based similarity search in 3D object databases. *ACM Comput. Surv.*, 37(4):345–387, 2005. 958278.

[4] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma. VIPS: a Vision-based Page Segmentation Algorithm. Technical Report MSR-TR-2003-79, Microsoft Research, Nov 2003.

[5] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung. On Visual Similarity Based 3D Model Retrieval. In *Eurographics*, pages 223–232, Granada, Spain, 2003.

[6] A. Christensen. Semantische Anreicherung von Suchanfragen auf Basis von Topic Maps., June 2005. Diplomarbeit.

[7] J. Eakins and M. Graham. Content-based Image Retrieval. A Report to the JISC Technology Applications Programme. Technical report, University of Northumbria at Newcastle, Jan. 1999.

[8] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A Search Engine for 3D Models. *ACM Transactions on Graphics*, V(N), 2002.

[9] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *ACM SIGMOD Conf. on the Management of Data*, pages 47–57, 1984.

[10] M. Heczko, D. A. Keim, D. Saupe, and D. V. Vranić. Verfahren zur Ähnlichkeitssuche auf 3D-Objekten. In *Datenbanksysteme in Büro, Technik und Wissenschaft (BTW), 9. GI-Fachtagung,*, pages 384–401, London, UK, 2001. Springer-Verlag.

[11] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3D freeform design. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[12] D. Liu and A. Razdan. *Computational and Information Science*, volume 3314/2004 of *Lecture Notes in Computer Science*, chapter Knowledge-Based Search Engine for Specific 3D Models, pages 530–537. Springer Berlin / Heidelberg, 2004.

[13] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40:262 282, 2007.

[14] Apache Lucene, 2006.

[15] S. Mantei, R. P. Pein, and M. Amador. OntoCollab Demonstrator - semantic concepts for distributed collaborative engineering. EADS Innovation Works, 2007.

[16] P. Min, J. Chen, and T. Funkhouser. A 2D sketch interface for a 3D model search engine. In *SIGGRAPH '02: ACM SIGGRAPH 2002 conference abstracts and applications*, pages 138–138, New York, NY, USA, 2002. ACM.

[17] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. FiberMesh: designing freeform surfaces with 3D curves. *ACM Trans. Graph.*, 26(3):41, 2007.

[18] OpenSearch, 2006.

[19] D. Parikh, R. Sukthankar, T. Chen, and M. Chen. Feature-based Part Retrieval for Interactive 3D Reassembly. In *WACV '07: Proceedings of the Eighth IEEE Workshop on Applications of Computer Vision*, page 14, Washington, DC, USA, 2007. IEEE Computer Society. 1264666.

[20] R. P. Pein and Z. Lu. A Flexible Image Retrieval Framework. In Y. Shi, G. D. van Albada, J. Dongarra, and P. M. A. Sloot, editors, *International Conference on Computational Science (3)*, volume 4489 of *Lecture Notes in Computer Science*, pages 754–761. Springer, may 2007.

[21] J. Pu, Y. Kalyanaraman, S. Jayanti, K. Ramani, and Z. Pizlo. Navigation and Discovery of 3D models in a CAD Repository. Revised Draft, Accepted for Publication in IEEE Computer Graphics and Applications.

[22] M. Renz and W. Renz. Neue Verfahren im Bildretrieval. Perspektiven für die Anwendung. In R. Schmidt, editor, *Proceedings der 22. Online-Tagung der DGI*, pages 102–128, May 2000.

[23] T. Schreck. *Effective Retrieval and Visual Analysis in Multimedia Databases*. PhD thesis, Universität Konstanz, 2007.

[24] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton Shape Benchmark, 2004.

[25] J. Stefanowski and D. Weiss. *Advances in Web Intelligence*, volume 2663 of *Lecture Notes in Computer Science*, chapter Carrot 2 and Language Properties in Web Search Results Clustering, pages 240–249. Springer-Verlag Berlin Heidelberg, 2003.

[26] J. Tangelder and R. Veltkamp. A survey of content based 3D shape retrieval methods. *Shape Modeling Applications, 2004. Proceedings*, pages 145–156, June 2004. 1314502.

[27] D. V. Vranić. *3D Model Retrieval*. PhD thesis, Universität Leipzig, Institut für Informatik, 2004.

[28] D. V. Vranić. Content-based Classification of 3D-models by Capturing spatial Characteristics, 2006.

[29] R. C. Zeleznik, K. P. Herndon, and J. F. Hughes. SKETCH: an interface for sketching 3D scenes. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses*, page 19, New York, NY, USA, 2007. ACM.