Research and Practice on 3D Model Retrieval and Visualization Technology in Institutional Repositories: Postprint

Authors: Wu Zhiqiang, Zhu Zhongming, Liu Wei, Zhang Wangqiang, Yao Xiaona

Date: 2017-11-08T00:00:00+00:00

Abstract

[Purpose] Based on the extended functional requirements of institutional repositories, this study investigates 3D model retrieval and display technologies, and designs and implements content-based 3D model retrieval as well as Web3D display and interaction capabilities. [Method] Building upon the open-source 3D model retrieval algorithm from National Taiwan University, we adopt an off-screen rendering approach to obtain orthogonal projections of models, extract 3D model features, utilize Java3D to generate 3D model thumbnails, and employ Three.js to implement web-based online display and interaction of 3D models. [Result] Users can retrieve 3D models by submitting a 3D model URL or through upload, and on the 3D model browsing page, they can rotate and zoom the model using the mouse for detailed inspection. [Limitation] The retrieval results and efficiency of the model can satisfy current basic requirements; however, the recall and precision rates of the models need further improvement. Continuous attention to 3D model retrieval-related technologies is required to optimize and supplement the 3D model retrieval functionality. [Conclusion] After applying this model to the CSpace system, it can effectively expand the support capabilities for 3D models and provide users with more diverse methods for 3D model retrieval and application.

Full Text

Abstract

This study investigates 3D model retrieval and display technologies to expand the functionality of institutional repositories. We designed and implemented a content-based 3D model retrieval system with Web3D display and interactive capabilities. Building upon the open-source 3D model retrieval algorithm from

National Taiwan University, we modified the approach to obtain orthogonal projections through off-screen rendering for feature extraction. Java3D was employed to generate 3D model thumbnails, while Three.js enabled web-based presentation and interaction. Users can retrieve 3D models by submitting URLs or uploading files, and can rotate and zoom models using mouse controls during browsing. While current retrieval results and efficiency meet basic requirements, recall and precision rates need further improvement. Continued attention to advances in 3D model retrieval technology will be necessary to optimize and enhance these functions. Integration with the CSpace system effectively expands 3D model support capabilities, providing users with more diverse retrieval and application methods.

Keywords: Institutional Repository, CSpace, 3D Model Retrieval, Web3D, Three.js

1. Introduction

Institutional Repositories (IRs) serve as critical infrastructure for research and educational institutions to store, organize, and manage their intellectual output. With the development of open access initiatives and web technologies, an increasing number of institutions have established their own IR systems to organize and manage research outputs internally while promoting open sharing according to relevant protocols, thereby facilitating dissemination and utilization of research results and expanding academic influence. However, most current IRs primarily support traditional text-based content such as journals, conference papers, and reports, with insufficient support for non-text resources including images, audio, video, and 3D models. As the fourth generation of multimedia formats, 3D models offer more natural and intuitive representation compared to sound, images, and video, while also providing interactive capabilities that other media lack. With advances in computer hardware/software, computer graphics, and computer vision, 3D models have found widespread application in industrial manufacturing, film production, virtual reality, education, and scientific research, becoming a common form of knowledge asset. This necessitates that IRs adapt to diverse knowledge output formats and support 3D model storage, organization, and management.

In terms of management capabilities, IR support for 3D models can be categorized into two approaches: The first employs methods similar to text content management, supporting description, upload, and preservation of 3D model objects while providing metadata-based retrieval and discovery. The second approach resembles the management of images and audio/video, offering online browsing capabilities and enabling content-based retrieval and visual navigation through content analysis and indexing technologies. The second approach proves more user-friendly and intuitive for information retrieval and discovery. Previously constrained by technological limitations, 3D model organization could only adopt the first approach. However, recent advances in computer graphics and pattern recognition have made the second approach feasible.

This paper surveys the state-of-the-art in 3D model retrieval and display technologies, investigates these technologies in detail, and integrates them with the functional requirements of the Chinese Academy of Sciences Institutional Repository (CSpace). Using open-source visual similarity-based 3D model retrieval algorithms and the Web3D display tool Three.js, we have constructed a 3D model retrieval and display system.

Compared to images, audio, and video, 3D models exhibit richer content representation and unique characteristics: numerous file formats exist due to diverse authoring tools (the 3D Object Converter alone supports at least 720 formats, far exceeding the complexity of image, audio, and video formats); complete model files typically separate shape and texture data; most 3D model files can be opened with text editors; higher computer graphics processing capabilities are required; and models possess translation, scaling, and rotation invariance. These characteristics make 3D model retrieval and display methods diverse and technically complex.

2. Related Technologies

2.1 3D Model Retrieval

3D model retrieval technology emerged in the early 1980s. 经过近 40 年的发展，该领域已经积累了大量的理论研究与实践成果，也发布了一些实证检索系统。Notable international systems include the Princeton 3D Model Search Engine from Princeton University's Shape Retrieval and Analysis Lab, the 3D model retrieval system from the University of Konstanz, and the content-based 3D model retrieval system from Carnegie Mellon University. Domestic research, though starting later, has produced representative achievements such as the visual similarity-based online 3D model retrieval system from National Taiwan University' s Communication and Multimedia Laboratory, and the Colored 3D Model Database (CDB) from Jilin University, which provides an empirical platform for color and texture-based retrieval.

Retrieval methods fall into two categories: text-based and content-based. Text-based retrieval requires prior annotation of 3D models, often failing to meet precise search needs. With enhanced computer graphics capabilities, content-based retrieval has become a research focus. Content-based retrieval includes file-based, 2D sketch-based, and 3D sketch-based methods. Most current 3D model retrieval systems employ file-based retrieval, where users select a target 3D model to find similar models in the database.

2.2 3D Model Display

3D model display comprises model rendering and user interaction, with interaction being the key aspect, including functions such as zooming and rotation that enable users to examine model details from multiple perspectives and levels. Display methods are divided into web-based browsing (Web3D) and client-based browsing. Client-based browsing is supported by numerous software packages requiring only installation. Web3D implementation technologies include cloud rendering, Java3D, Away3D, Unity Web Player 3D, Cult3D, and Three.js.

Cloud rendering performs 3D model rendering on the server side, requiring no client configuration and enabling display across platforms including mobile devices, tablets, and PCs. During display, the client loads only rendered images rather than model files, eliminating loading delays. Theoretically, any model size can be displayed depending on server configuration, with enhanced security.

Java3D primarily implements Web3D through Java Applets, requiring JRE installation and browser security configuration, which complicates user operation. Away3D, Unity Web Player 3D, and Cult3D all require plugin installation. Three.js is an open-source WebGL framework that functions as a standard JavaScript file callable from web pages. WebGL support is required, which is available in most modern browsers (Chrome, Firefox, 360 Security Browser 6.0, IE11), with WebGL support being the future direction of browser development.

Through investigation and experimental research, we found that while numerous 3D model retrieval systems and platforms exist, most do not provide open APIs or detailed algorithm descriptions, preventing effective integration into IR systems. The visual similarity-based algorithm from National Taiwan University offers publicly available source code with demonstrated high recall and precision rates. Its source code can be modified and customized according to IR system requirements, facilitating integration. Therefore, we selected this algorithm for our IR 3D model retrieval research.

From a user experience perspective, plugin-free Web3D display requiring no user configuration proves most user-friendly. Three.js and cloud rendering meet these criteria. Cloud rendering demands substantial hardware and software investment, while Three.js encapsulates many low-level Web3D functions. By referencing its API documentation and including relevant JS files, developers can implement algorithms supporting multi-terminal display (PC, tablet, mobile), aligning with CSpace' s adaptive interface requirements. Therefore, we adopted Three.js for IR 3D model display research and implementation.

3. System Functions

The 3D model retrieval and display system for institutional repositories provides the following functions:

(1) **Model Format Conversion**: With diverse 3D model formats, unified format conversion is essential to avoid compatibility issues in subsequent feature extraction, thumbnail generation, and model display. Upon user submission, the system first checks the format and converts non-standard formats.

(2) **Feature Extraction**: 3D models offer rich expressive features including shape, contour, and color. Selecting appropriate parameters for feature

extraction prepares data for similarity calculation and retrieval.

(3) **Indexing**: Selecting appropriate storage and designing indexing mechanisms for 3D model indices ensures consistency with the model database during addition or deletion operations.

(4) **Retrieval**: Extracting features from user-submitted models, calculating similarity against all models in the repository based on the index, applying similarity thresholds, and returning sorted lists of similar models.

(5) **Thumbnail Generation**: To facilitate user selection, 2D thumbnails are displayed on retrieval result pages sorted by similarity. Users can click thumbnails to enter the model display page.

(6) **Model Display and Interaction**: In the web-based 3D display area, users can rotate and zoom models using mouse controls for detailed examination.

The system functional framework is shown in [Figure 1: see original paper], comprising two parts: retrieval (format conversion, feature extraction, indexing, retrieval) and Web3D display (thumbnails, model rendering, interaction).

4. Design and Implementation of Key Functions

4.1 3D Model Processing

The implementation requires various processing tools with specific environmental dependencies. Embedding these directly into existing IR systems would affect OS compatibility and complicate installation and upgrades. Therefore, we employed Java RMI (Remote Method Invocation) to configure 3D model processing on a Windows Server 2003 machine, enabling remote calls from the IR system via IP address and port, as illustrated in [Figure 2: see original paper].

The client first connects to the processing server, uploads the 3D model, and upon successful transfer, the server performs format conversion, feature extraction, and thumbnail generation based on client parameters, returning feature files, thumbnails, converted models, and processing results.

Core code for IR system calls to 3D model processing:

```
// Connect to 3D model processing server
if (!connect()) return "Can not connect the 3D server!";
// Upload 3D model
if (!objFeatureGenerate.upload(fileName, fileToByte(srcFile)))
    return "Upload 3D file failed!";
// Model conversion
if (isconvert2obj) {
    if (!objFeatureGenerate.convert2obj(fileName))
        return "convert2obj 3D file failed!";
```

```
if (objFeatureGenerate.isFindObj(fileNameWithoutExt+".obj")) {
    // Generate 3D model thumbnail
    if (isGetImage) {
        if (!objFeatureGenerate.generateImage(fileNameWithoutExt))
            return "Generate 3D image failed!";
        // Generate 3D model feature parameter document
        if (isGenerateFeature) {
            if (!objFeatureGenerate.generateFeature(fileNameWithoutExt))
                return "Generate 3D feature failed!";
            // Download 3D model feature document
            try {
                if (isGetImage && objFeatureGenerate.isFindObj(fileNameWithoutExt+".jpg'
                    save(fileNameWithoutExt + ".jpg");
                if (isconvert2obj) {
                    if(isSaveObj && objFeatureGenerate.isFindObj(fileNameWithoutExt+".ol
```

The client call class and server-side method interfaces are shown in [Figure 3: see original paper] and [Figure 4: see original paper] respectively. The package names must remain consistent between client and server to avoid method lookup errors.

The 3D model processing component implements three main functions:

(1) **Format Conversion**: Given the diversity of formats, we adopted the widely-used obj format for uniformity. Conversion utilizes 3D Object Converter, supporting common formats (3ds, dxf, cad, geo, stl, c4d, rwx) to obj. The software runs on the processing server and is invoked via the convert2obj method.

(2) **Thumbnail Generation**: Since 3D model files are text-based, generating thumbnails requires reading the file, rendering to an off-screen buffer, and extracting image data. Java3D reads files and performs off-screen rendering, saving thumbnail data as JPG files. Due to complex texture loading, thumbnails are generated without texture information and stored as child entries of 3D models in the IR system.

(3) **Feature Extraction**: Features are algorithm-dependent. After experimentation, we selected the visual similarity-based algorithm from National Taiwan University, which assumes that similar 3D models appear similar from all viewing angles. Similarity is measured by summing similarities across multiple views under light fields, using LightField Descriptors. The original algorithm stored rendered images in Glut Windows before screen capture, which could produce incomplete images if windows were occluded. We modified this to off-screen rendering and adjusted parameter calling methods for Java integration.

4.2 3D Model Indexing and Retrieval

We used Solr-4.10.2 for index maintenance. The indexing scheme corresponds to the retrieval algorithm, which stores model lists in text files accessed via file paths. Each line represents a 3D model feature storage path (e.g., Example/m0). Solr indices primarily store feature path information. The schema.xml configuration defines fields and types as follows:

```
<fieldtype name="string" class="solr.StrField" sortMissingLast="true" omitNorms="true"/>
<field name="id" type="string" indexed="true" stored="true" multiValued="false" required="tr
<field name="_{version}" type="long" indexed="true" stored="true"/>
```

3D model files are stored as Bitstream objects in the IR system, with feature files named using Bitstream IDs. Upon submission, the system identifies 3D models by file extension and triggers format conversion, feature extraction, thumbnail generation, and Solr indexing. Detailed Bitstream creation, modification, and deletion mechanisms for Solr indexing are described in reference [17].

The retrieval process is shown in [Figure 5: see original paper]. When retrieving, the system first checks the submitted model format, converting non-obj formats before feature extraction. It then compares features against all models, calculates similarity scores, sorts results, and applies a similarity threshold of 1000 (values >1000 are discarded, with 0 indicating identical models). Results are presented as thumbnails sorted by similarity, with pagination via Ajax for performance when many similar models exist.

4.3 Model Display and Interaction

Three.js is an open-source WebGL framework encapsulating many low-level Web3D functions. While its example obj loader provides basic display, it lacks adaptive sizing and centering. We addressed this using Three.js' s Box3 function to center models and calculate dimensions, then set camera z-axis parameters based on model size to ensure proper display scaling.

Rotation and zoom interactions are implemented via OrbitControls.js. The display function is embedded in display-item.jsp, rendering models without textures on a white canvas with cyan lighting. The function displayModel(container, objname) enables loading multiple models, where container specifies the display area and objname provides the model file path. Code for adaptive sizing:

```
var box3 = new THREE.Box3();
box3.setFromObject(object);
box3.center(object.position);
object.position.multiplyScalar(-1);
var a = new THREE.Vector3();
a = box3.size();
var objectSize = Math.max(a.x, a.y);
var fov = camera.fov * (Math.PI / 180);
```

```
var distance = Math.abs(objectSize / Math.sin(fov / 2));
camera.position.z = distance;
```

4.4 Implementation Results

To validate our approach, we downloaded 20+ obj format models from National Taiwan University' s system and converted subsets to 3ds, dxf, cad, geo, stl, c4d, and rwx formats using 3D Object Converter. Upon submission, the system automatically identified 3D models by extension, triggered format conversion for non-obj files, and completed feature extraction, thumbnail generation, and indexing.

Users can click the camera icon in the search interface to upload models. The system identifies 3D formats and initiates retrieval. Results demonstrate that various format uploads successfully complete automatic conversion, thumbnail generation, feature extraction, and indexing, with results sorted by similarity and displayed as thumbnails. Thresholds differentiate exact matches from similar models. The retrieval results page is shown in [Figure 6: see original paper].

Clicking thumbnails enters the model viewer, where mouse controls enable rotation and zooming for detailed examination, as shown in [Figure 7: see original paper].

5. Conclusion

Institutional repositories are vital infrastructure for managing institutional knowledge assets. As information technology evolves, knowledge output formats diversify. Addressing CSpace requirements for 3D model retrieval and display, this study implemented a 3D model retrieval system using open-source algorithms and Web3D display with Three.js. Testing confirms that retrieval results and efficiency meet current needs, though recall and precision require improvement. Future work will monitor advances in 3D model retrieval technology, optimize the existing system, and research semantic retrieval methods to provide better 3D model support services in institutional repositories.

References

[1] Zhang Xiaolin. Trends and Challenges for Institutional Repositories[J]. New Technology of Library and Information Service, 2014(2): 1-7.

[2] Lu Tong. Retrieval of 3D CAD Model: Survey[J]. Computer Science, 2012, 39(4): 14-22, 27.

[3] 3D Object Converter[EB/OL]. [2016-06-10]. http://3doc.i3dconverter.com/features.html.

[4] Princeton 3D Model Search Engine[EB/OL]. [2016-06-10]. http://shape.cs.princeton.edu/search.html.

[5] Zarpalas D, Kordelas G, Daras P. Recognizing 3D Objects in Cluttered Scenes Using Projection Images[C]//Proceedings of the 18th IEEE International

Conference on Image Processing. 2011: 673-676.

[6] AMP 3D Model Retrieval[EB/OL]. [2016-06-10]. http://chenlab.ece.cornell.edu/projects/3DModelRetrieval

[7] 3D Model Retrieval System Based on LightField Descriptors[EB/OL]. [2016-06-10]. http://3d.csie.ntu.edu.tw/.

[8] Cheng Yanzhi, Lü Tianyang, Wang Sen, et al. 3D Model Retrieval Based on Surface Color Properties[C]//Proceedings of NDBC2009. 2009: 366-372.

[9] Java3D[EB/OL]. [2016-03-20]. https://java3d.java.net/binary-builds.html.

[10] Away3D[EB/OL]. [2016-04-10]. http://away3d.com/.

[11] Unity3D[EB/OL]. [2016-04-10]. http://unity3d.com/cn/webplayer/.

[12] Cult3D[EB/OL]. [2016-04-10]. http://www.web3d.com.cn/new/teach/cult3d/.

[13] Three.js[EB/OL]. [2016-04-10]. http://threejs.org/.

[14] RMI[EB/OL]. [2016-04-10]. https://docs.oracle.com/javase/tutorial/rmi/.

[15] Chen D, Tian X, Shen Y, et al. On Visual Similarity Based 3D Model Retrieval[C]//Proceedings of Computer Graphics Forum. Blackwell Publishing, Inc, 2003: 223-232.

[16] Solr[EB/OL]. [2016-04-10]. http://lucene.apache.org/solr/.

[17] Wu Zhiqiang, Zhu Zhongming, Liu Wei, et al. The Image Retrieval of LireSolr-Based for Institutional Repository[J]. Research on Library Science, 2016(14): 58-63, 39.

Author Contributions

Wu Zhiqiang: Drafted the manuscript, designed and implemented system functions; Zhu Zhongming: Proposed research ideas and designed the study; Liu Wei: Deployed and tested system functions; Zhang Wangqiang: Implemented 3D model indexing interfaces and revised the manuscript; Yao Xiaona: Conducted 3D model retrieval technology research.

Conflict of Interest Statement

All authors declare no conflict of interest.

Support Data

Support data is self-archived by the authors, E-mail: wuzq@llas.ac.cn. [1] Wu Zhiqiang. GenerateFeature.zip. Remote 3D Model Processing Program.

Received Date: 2016-08-25 **Revised Date:** 2016-09-26

News Item

HathiTrust Research Center Releases Extracted Features Dataset Version 1.0

HathiTrust recently released an open dataset: the Extracted Features (EF) Dataset Version 1.0 from the HathiTrust Research Center (HTRC). This dataset provides open access to data extracted from the full text of the HathiTrust Digital Library (HTDL).

The EF dataset opens the entire HathiTrust collection for academic exploration, enabling scholars to investigate historical and cultural trends, the rise and fall of topics in the corpus, and the evolution of words and writing structures from the 16th to the late 20th century. It provides quantitative information about words, lines, sentence components, and other details for every page of every volume in the HathiTrust Digital Library, allowing researchers to conduct in-depth content analysis of specific volumes.

The dataset was extracted from 2 trillion words across 500 million pages in 13.7 million volumes. The initial version draws from HathiTrust' s public collections and has already enabled novel research in economics, history, linguistics, literature, and sociology.

"The Extracted Features dataset creates unprecedented opportunities for research and teaching," said J. Stephen Downie, co-director of HTRC and professor at the University of Illinois Urbana-Champaign. "We look forward to seeing how the academic community utilizes the EF dataset in scientific research, laboratories, and classrooms."

Michael Furlough, executive director of HathiTrust, stated: "We established HTRC to help researchers fully mine the HathiTrust collections. The EF dataset release provides researchers with a novel and effective way to explore HathiTrust by extracting relevant feature data from the entire corpus."

(Compiled from: https://librarytechnology.org/news/pr.pl?id=22069) (Journal News)

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.