Quarterly Status Report
September – December 2009

Computational Archival Analysis and Visualization of Large-scale Electronic Records Collections using Cyberinfrastructure

Texas Advanced Computing Center, University of Texas at Austin

To submit to:

Barry I. Schneider
Program Director
Office of Cyberinfrastructure, Room 1145 S
The National Science Foundation
4201 Wilson Boulevard
Arlington, Virginia 22230

Irene D. Lombardo
Staff Associate
Office of Cyberinfrastructure, Room 1145 S
The National Science Foundation
4201 Wilson Boulevard
Arlington, Virginia 22230

Robert Chadduck
Computer Engineer, Principal Technologist for Advanced Research
Center for Advanced Systems and Technologies
The National Archives and Records Administration
College Park, Maryland 20740

Table of Contents

1. INTRODUCTION

Our research involves conducting computational analysis and visualization of large heterogeneous electronic records collections without external metadata for archival processing and access purposes. Our methodology involves a combination of RDBMS, text analysis, and visualization techniques.

Our research questions are:

- What can we learn inductively about large scale, heterogeneous, complex digital collections with no external metadata?
- What information can be extracted from the structure and content of this type of digital collections?
  - How can we visually represent this information meaningfully for analysis and access purposes?
  - Does this visual representation allow discovering patterns and outliers in content and structural arrangement and description?
- Are data-driven, inductive representations useful for archival purposes (analysis, reference, processing, access)?

The challenges involved in this research are:

- Researching/developing and testing content analysis techniques for archival processing and access of large data collections.
- Identifying meaningful levels of abstraction and visualization techniques to make sense of the data for archival analysis and access purposes.
- Understanding the cyberinfrastrucure requirements to handle the different tasks at the required levels of scale.

We describe the progress achieved in two areas of this research: a) visualization of the structure and technical make-up of large records collections, and b) collection's content analysis. By the end of this research we expect to mash both components into an integrated framework. We also discuss metadata extraction, and the cyberifrastructure used in this project.

2. ACCESSING THE TRANSCONTINENTAL PERSISTENT ARCHIVES PROTOTYPE (TPAP)

The TPAP is stored and updated in a remote IRODS implementation. We access the testbed collection via two methods: a) using icommands, and b) using a java interface named Jargon. We used the former method to transfer the test-bed collection to our data storage resource Corral[1] and the latter to extract metadata remotely from the test-bed collection.

[1] The explanation about the resource Corral is included in the cyberinfrastructure section. We currently have a 10 terabytes allocation in Corral for this project.

2.1 Accessing the IRODS server using icommands to transfer the testbed collection

We developed a number of Python scripts involving IRODS commands to transfer the collection in an automated manner. We use our local copy of the collection to conduct content analysis and to extract file format identification data.

2.2 Accessing the iRODS server using Jargon to extract metadata

IRODS provides an API known as Jargon to facilitate the interaction of a Java based client program with the IRODS web server. This API contains methods that allow a user to connect and query the IRODS database. We wrote a Java client that establishes connection with the IRODS server using the server's URL. Once the connection is successfully established we traverse through the hierarchical archival data in a Breadth First Search manner and aggregate metadata in a comma-delimited file. The resultant file contains the name, path and size of the file in each row, information that we consider as basic structural metadata.

The step-by-step operation of the client is as follows:

- Java Client, which establishes connection with the IRODS server using the URL of IRODS server along with login, credentials and creates an IRODS File Object.
- Starting with the root node, client carries out a Breadth First Traversal through the hierarchical archival data.
- Conversion of IRODS File Object to Native Java File object followed by extraction of the metadata from the native java file object such as name of the file, path, size etc.
- Generation of a comma-delimited file by the client, which contains the collection's structural metadata.

Currently in our database we have information about 114,071 directories, 3,067,229 files, which have a reported total size of 4.5 Terabytes, from an extraction conducted in September 2009.

3. TECHNICAL METADATA EXTRACTION AND MANAGEMENT

In this project we work with structural metadata extracted from the file system, and with file format identification metadata.

3.1 File identification metadata

We use file format identification metadata for purposes of understanding a RG's structure/arrangement in relation to its technical make-up, and for preservation documentation and planning purposes. To extract this metadata from the collection, we use DROID (Digital Record Object Identification), an automatic file format identification tool developed by the UK National Archives. Currently we run DROID in our local copy of the collection stored in Corral. In the future we would like to extract metadata directly from the collection stored in IRODS via an intermediate application using Jargon and a DROID API.

DROID uses a signature file to match inspected files with their respective formats. We applied an XSLT to convert DROID's signature file data into an HTML table. This allowed us to categorize the file formats that DROID presently identifies into broader categories such as: images, audio, GIS, database, scripts, text, video, drawing, graphics, 3D/graphics, datasets, word processor, etc. Appendix 1 Technical Metadata Table, shows some rows of the table. This categorization is useful for purposes of testing how large amounts of file format information can be further abstracted to meaningful and manageable levels. The output file generated by DROID in XML format is imported to the database.

3.2 Metadata Database

Although storing data in XML format is a convenient "light weight" way for data transformation and sharing, it doesn't provide access efficiently enough to scale up for the entire test-bed collection used in this project. We store all the extracted information using a relational database management system (RDBMS). The RDBMS serves as a centralized storage manager on disk and provides efficient data access that is scalable to the requirements of this project. The RDBMS has mature mechanisms to support concurrent access and provides low level data aggregation and transformation.

The RDBMS used in this project is Microsoft SQL server 2008. The database is hosted in Corral. We have developed code and procedures to integrate extracted metadata as well as to export it out from database in xml format for data sharing. The schema diagram of the relational database is shown in figure 1 below:

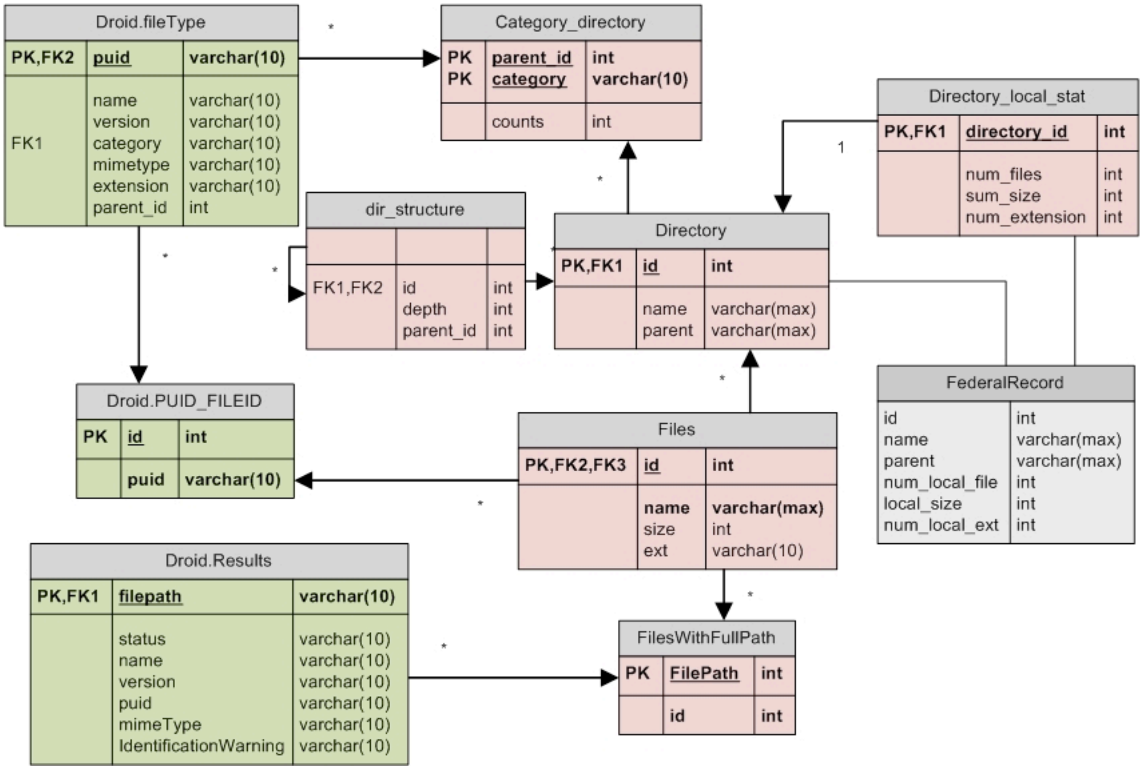


Figure 1 Metadata database schema

In the schema there are three groups of tables and views indicated with three different colors:

- *Metadata Extraction Tables*: The three tables with green background in the figure are used to store the results of file type identification by DROID. The *Droid.Results* table stores imported raw output from DROID. Rows in the *Droid.Results* table are uniquely identifiable by *filepath*. The *Droid.Filetype* table stores metadata imported from the DROID database about each file type. Additionally, the *category* field is used to label each file type as one of the manually defined file format groups. The *Droid.PUID_FILEID* maintains association between each file in the collection with one file type.
- *Directory Structure Tables:* The six tables with red background stores the collection's structural metadata. The *Files* and *Directories* are used to store raw metadata extracted from the IRODS server. As we retrieve more metadata, we can expand upon columns of these two tables. The *dir_structure* stores the hierarchical structure of the collection. We opted to have a separate *FilesWithFullPath* table to store the file path separately in consideration to issues of records duplication in which one file might correspond to multiple locations. This table is used to associate files with results from other processes such as ffile type identification. Additionally, statistical information is pre-aggregated per directory to summarize the number of files, number of different file extension types, and total size of all files included in a given directory. *Directory_local_stat* table, as well as to summarize number of files by categories in *category_directory* table.
- *Federal Records Table:* shown with gray background is a materialized view specifically created for the visualization. The view is a join between *Directories* and *Directory_local_stat_*table. As we progress, we will create additional views to support more complicated visualizations.

4. VISUALIZATION FOR ANALYSIS OF LARGE DIGITAL COLLECTIONS

In the first quarter of the project we worked on building a data-driven visual representation of the collection based on structural and technical metadata. Our initial development goal was to represent the extracted properties of this very large and varied digital collection at different levels of abstraction. As the visualization progressed, we focused on investigating whether such representation would allow and or facilitate archival analysis.

4.1 Treemap visualization

The treemap is an effective way to visualize hierarchical structures as well as classification distributions of different types of data. We chose it to represent the NARA testbed collection because it facilitates finding patterns and meaningful information. Due to the particularities of this research, we introduced various adaptations to the basic treemap application. Our treemap application required implementing database connectivity modules and usage of the Java Prefuse library.

The visualization of the testbed collection was built with structural metadata. In figure 2 below, it is possible to observe the scope of the entire testbed collection and of the individual RGs. Users can make selections in the visualization interface based on the metadata elements stored in the database to obtain different views of the collection. These selections are aggregated at the directory level, mapped to different color values, and rendered on demand. Presently, by interacting with the different visualization features, a user can:

- View the scope of the entire collection considering file sizes and file numbers density per directory.
- Compare and contrast the different RGs.
- Browse the collection across RG.
- Study the structure/arrangement of the different RGs including the different configurations of the sub-groups within and at different structural levels.
- Identify duplicate directories.
- Identify and study file-naming conventions for directories at the different hierarchical levels considering that file naming constitutes a way of grouping and description.
- Identify similar subjects/themes across the collection according to directory naming conventions.
- Identify patterns that correspond to arrangements based on file types, e.g GIS data
- Compare and contrast the arrangement of the different RGs. Associate the arrangement with types of files and themes across RGs.
- Determine categories of file formats present at each directory level. To synthesize the information and because one directory may contain various different file formats, file format identification information is aggregated by categories and rendered in the visualization as a color range according to the number of files in each category (See Section 3.1 and Appendix 1, Technical Metadata Table).
- Determine when directory labels describe types of file formats included within. For example, we noticed that across collections many directories are named GIS, images, documentation, and data. In many cases these directories contain GIS file types, image files, pdf files, and database dumps. Studying labeling patterns facilitates navigating and understanding how people manage information.

We show examples of the different functionalities in the following pages.

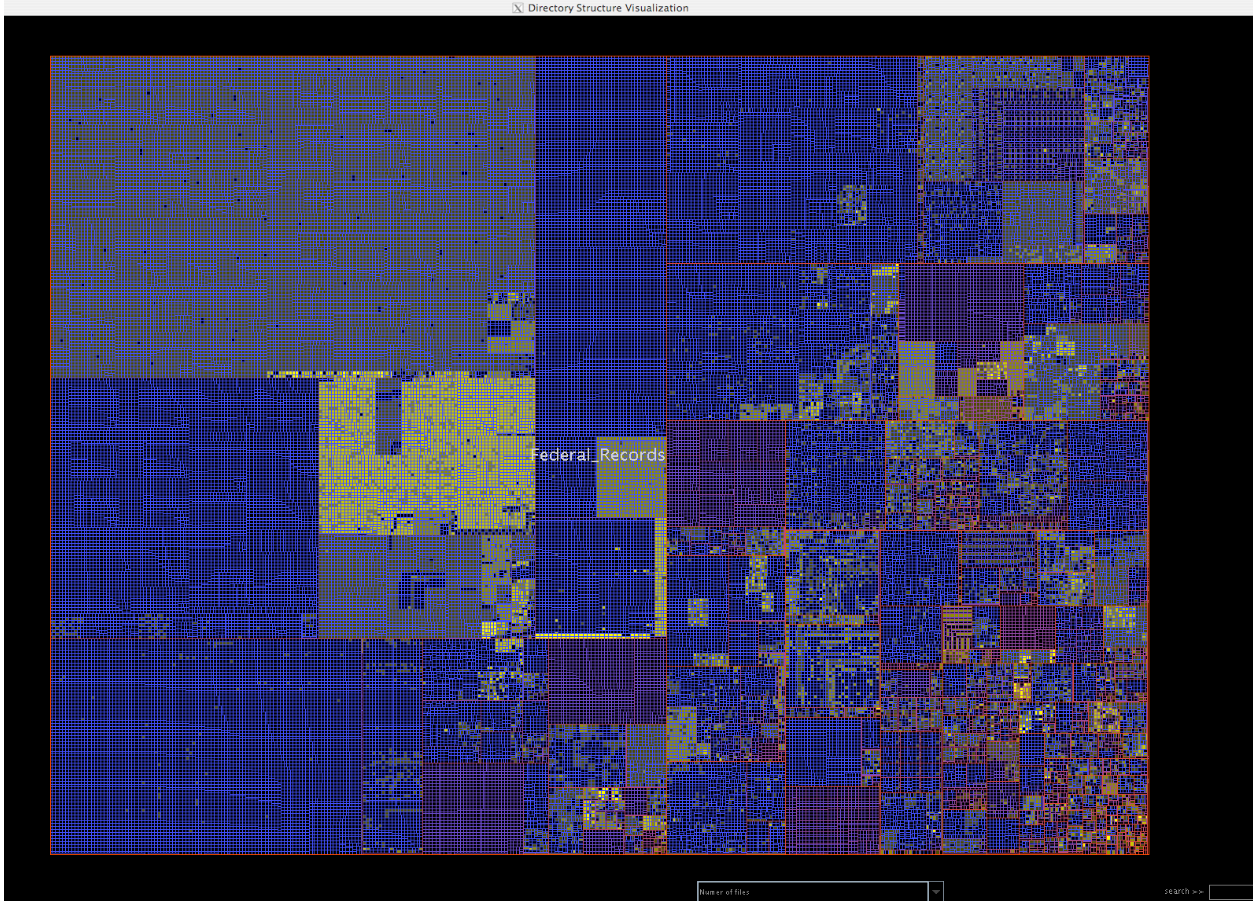
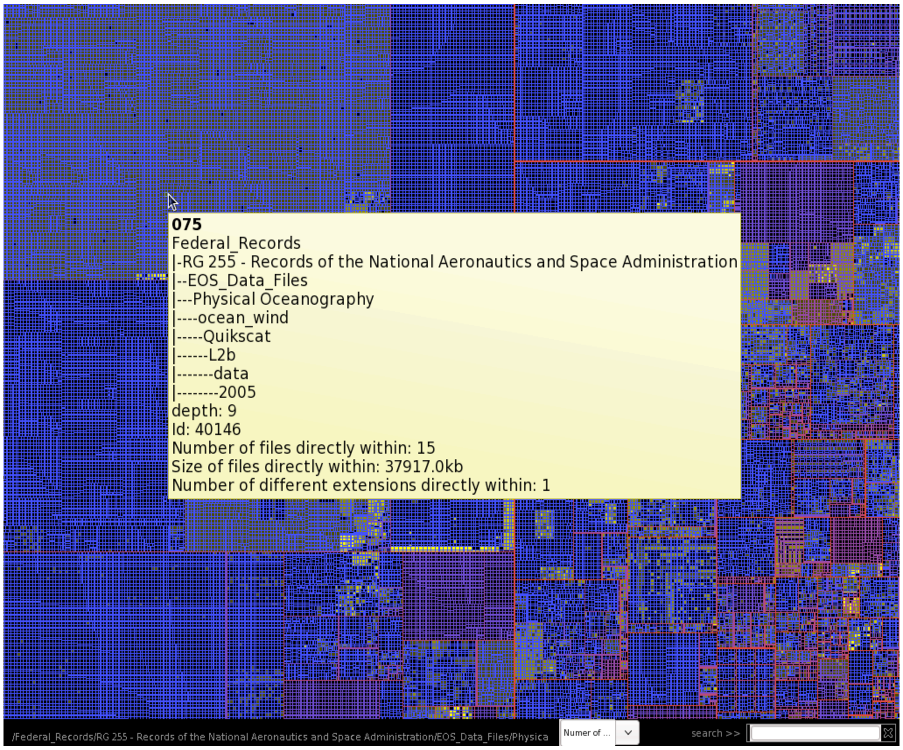


Figure 2 Treemap visualization of part of the NARA testbed collection

4.2 Features and user interaction with the visualizations



```
075
Federal_Records
|-RG 255 - Records of the National Aeronautics and Space Administration
|--EOS_Data_Files
|---Physical Oceanography
|----ocean_wind
|-----Quikscat
|------L2b
|-------data
|--------2005
depth: 9
Id: 40146
Number of files directly within: 15
Size of files directly within: 37917.0kb
Number of different extensions directly within: 1
```

/Federal_Records/RG 255 - Records of the National Aeronautics and Space Administration/EOS_Data_Files/Physica

Figure 3. Snapshot of tooltip.

Tooltips: Each rectangle in the treemap represents a directory. When the user moves the mouse pointer over a particular rectangle, a tooltip is generated which provides all the information (eg. complete directory structure from root to itself, Sizes of the file directly within it etc) about that directory. A snapshot of this feature is shown in figure 3.

Keyword search: The user can supply text keywords in the visualization interface and the search takes place in real time. If the query word matches any part of the path of the directory, the results are highlighted on the visualization. Figure 4 shows a group of RG. The directories colored pink are the result of the keyword search “images.”

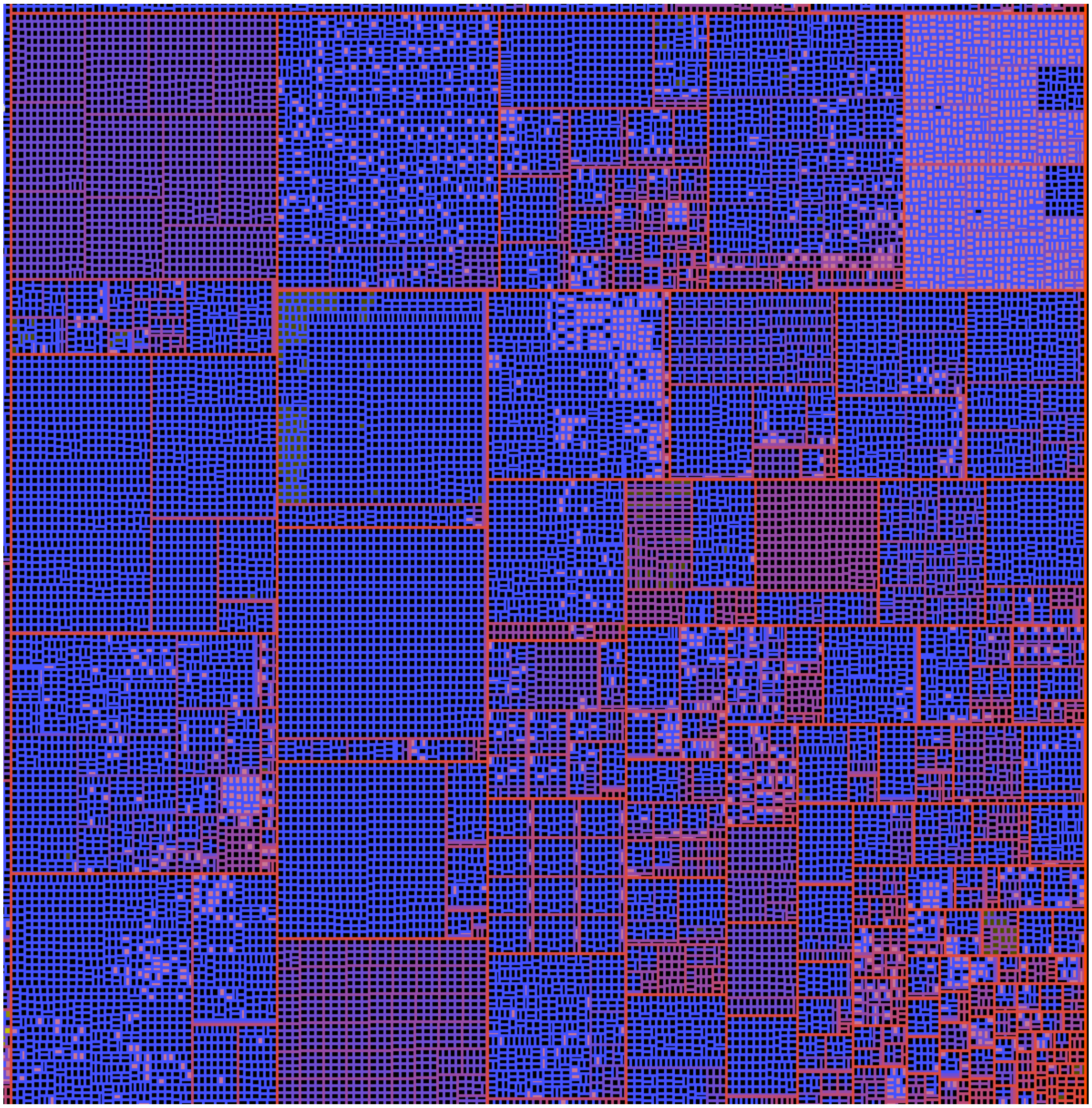


Figure 4. Examples of search results for "image" highlighted in pink across various RG.

Included in this snapshot and towards the right bottom are the smallest RGs in the collection. Note the regular patterns of some RGs, all of which denote some kind of structured configuration.

Highlight properties of the directory: The visualization supports highlighting of file properties, The user can choose to see patterns such as: similar file sizes, files of a particular type (images, videos etc.) and number of files represented by a suitable color-coding from less (dark) to more number of files (yellow). Figure 5 shows files of similar sizes highlighted with same shades of green-yellow. These views allow understanding the reasons behind the visible pattern. In the example shown in Figure 5, notice the regular pattern on the upper right side, which corresponds to the way in which the different files that comprise GIS data are arranged in nested directories.

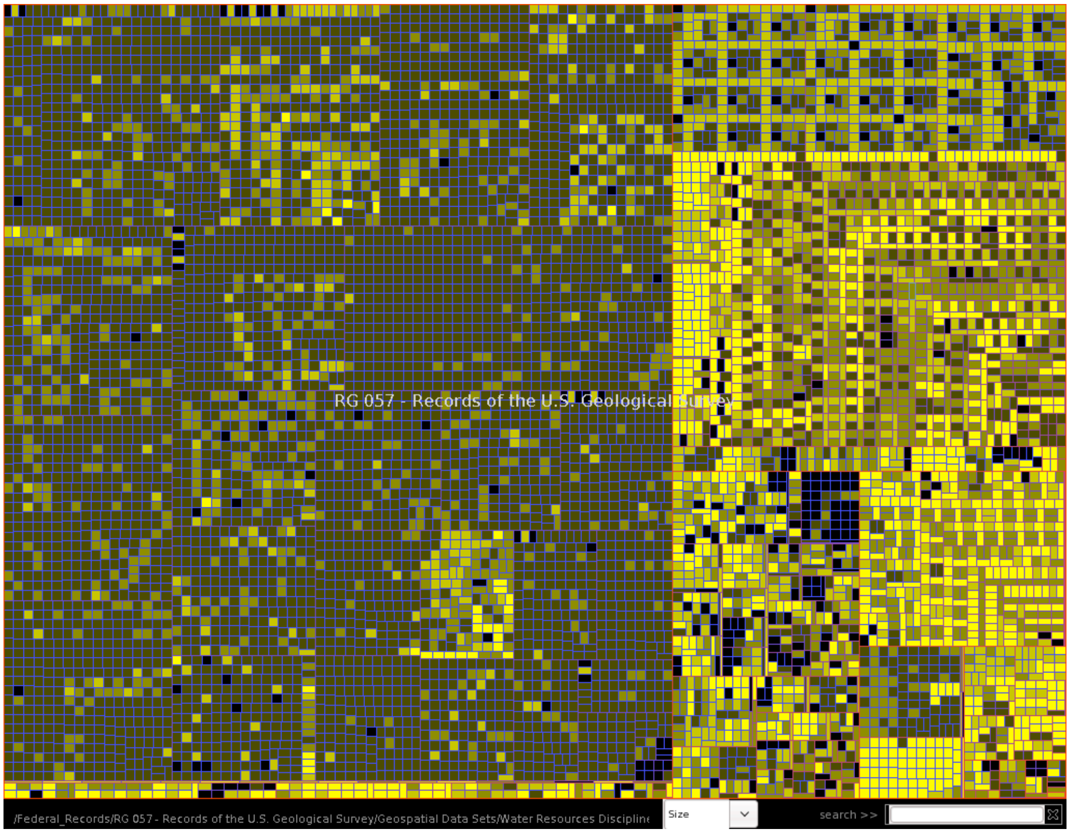

RG 057 - Records of the U.S. Geological Survey
/Federal_Records/RG 057 - Records of the U.S. Geological Survey/Geospatial Data Sets/Water Resources Disciplin
Size
search >>

Figure 5. Examples of highlighted directory properties (file sizes) in RG 057, Records of the US Geological Survey.

File format category: Figure 6 below shows directories containing, highlighted in yellow, image files as identified by DROID. It also shows a detailed view (zooming function) of the structure of the directory. Highlighted in pink are directories labeled "images." The analysis in this case allowed determining that the directories' labels match their contents. The latter is not always the case, and the exploration functionality allows us to establish whether there is a match between a label and the file formats included within or not.

Zooming: A basic zoom in or zoom out function allows detailed views of the arrangement of these records and the location of the directories named "images."

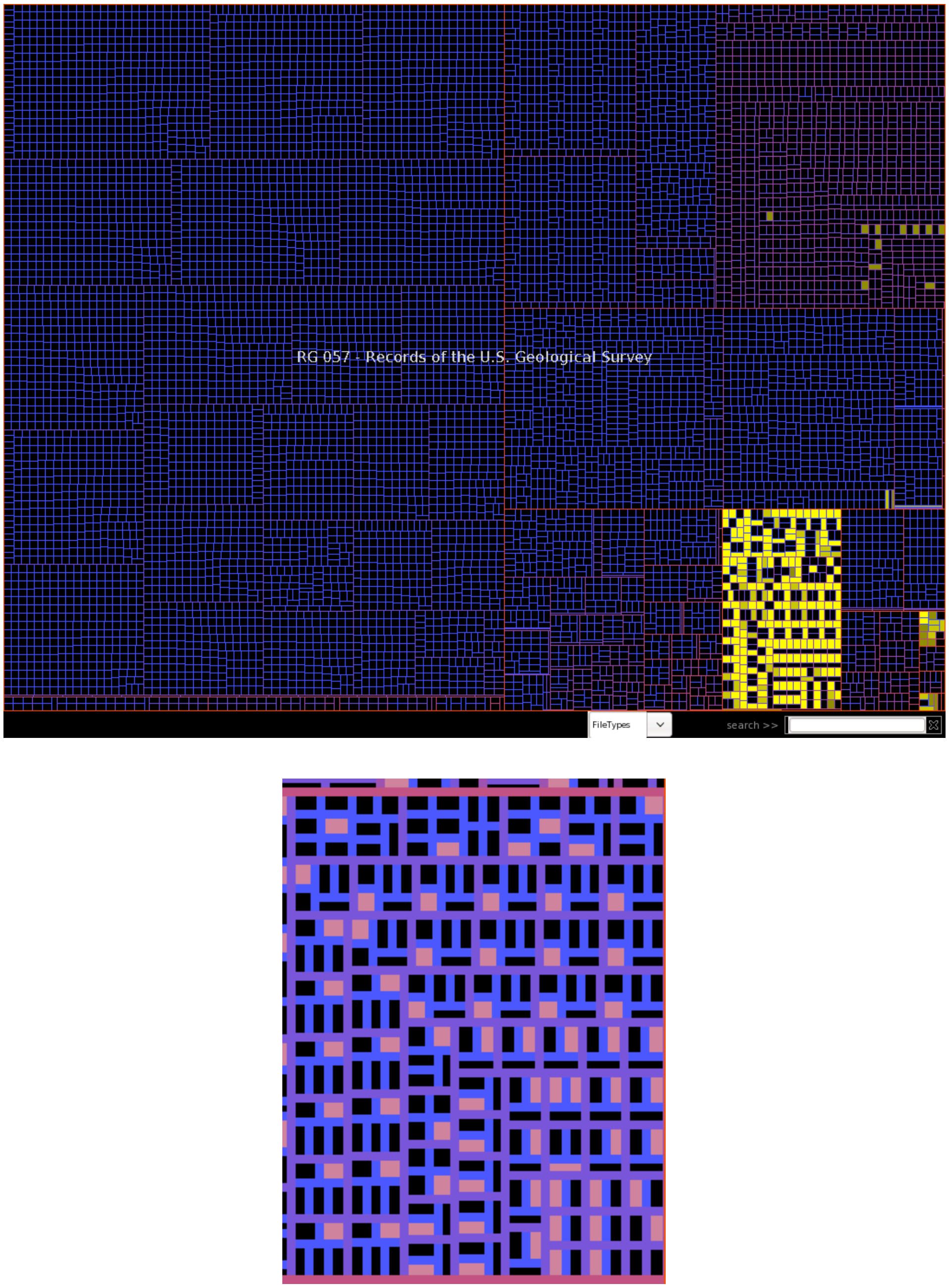
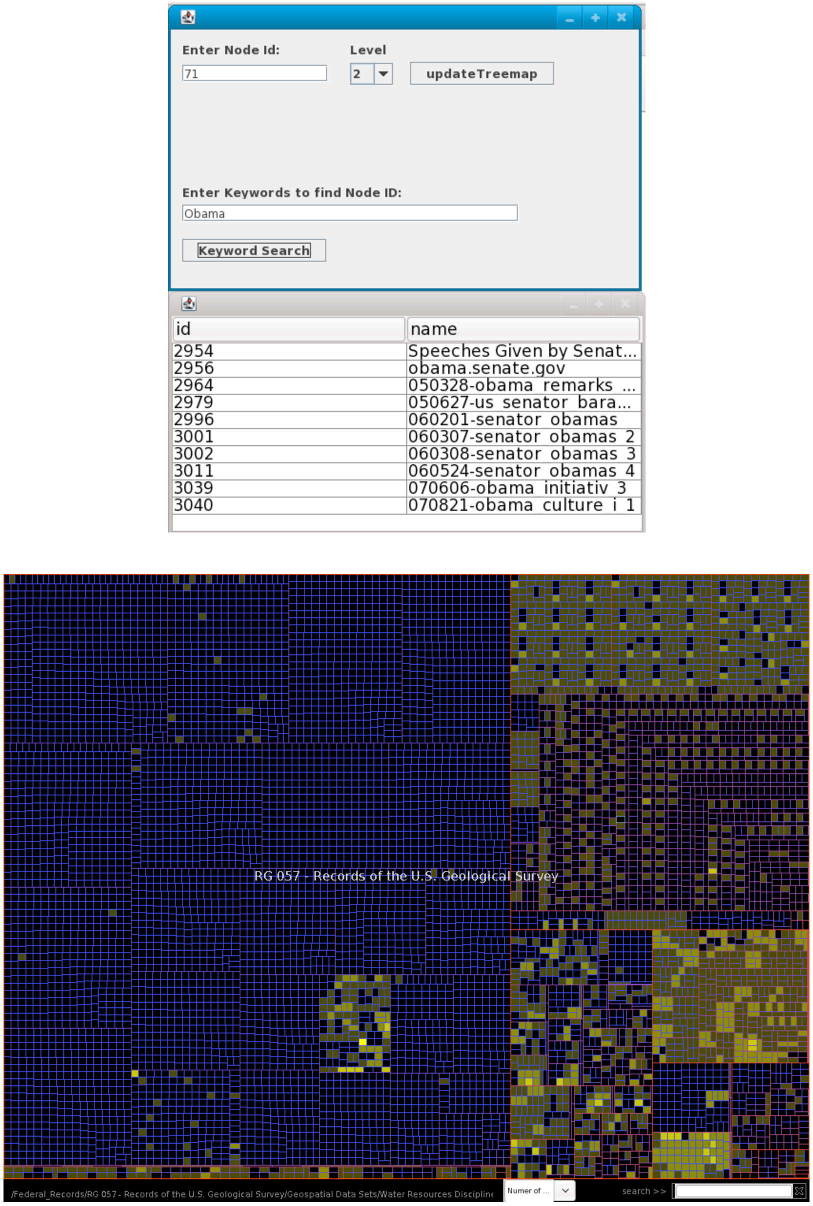


Figure 6. Relationship between the technical make up of the directory content and its labeling.

Dynamic root and levels of visualization: This feature allows the user to make any directory/node id as the root of the treemap and visualize only the sub-tree with that directory as a root. This helps in understanding deeply nested hierarchical structures, their technical make-up and labeling structure. In addition to input the node directly, the UI also features a search interface (Figure 7).



Enter Node Id: 71

Level: 2

Enter Keywords to find Node ID: Obama

id	name
2954	Speeches Given by Senat...
2956	obama.senate.gov
2964	050328-obama remarks ...
2979	050627-us senator bara...
2996	060201-senator obamas
3001	060307-senator obamas 2
3002	060308-senator obamas 3
3011	060524-senator obamas 4
3039	070606-obama initiativ 3
3040	070821-obama culture i 1

RG 057 - Records of the U.S. Geological Survey
/Federal_Records/RG 057 - Records of the U.S. Geological Survey/Geospatial Data Sets/Water Resources Disciplin
Numer of ...
search >>

Figure 7. User interface to select root directory (top), and example of user selected root view (bottom).

To better understand the arrangement/labeling/contents of very nested directories we can generate the visualization only up to a certain number of levels rather then from root to leaf. The user can specify this by selecting the number of levels to be visualized from the UI. It can be very helpful in making observations while going from broader to finer views or vice versa. Figure 8 shows one such visualization in which only 3 levels are shown.

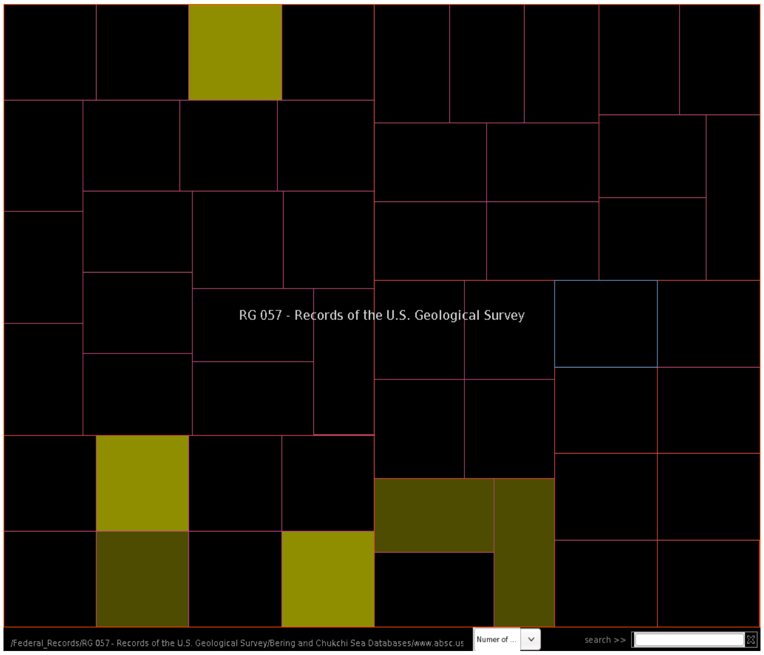

RG 057 - Records of the U.S. Geological Survey
/Federal_Records/RG 057 - Records of the U.S. Geological Survey/Bering and Chukchi Sea Databases/www.absc.us
Numer of ...
search >>

Figure 8. Example of dynamic root levels.

When we decided to use treemaps and work with the files properties we did not know that we could infer so much information about the collection. It was by reviewing the entire collection and conducting observations by comparison and contrast that we started to note arrangement configurations and patterns. These are useful to identify the collection's composition, establish preservation and arrangement priorities, and to identify the functions of the files and in some cases of the records composed by files. Moreover, we are exploring whether the visualization can be used as a map to navigate the collection or better yet, as an initial visual finding aid that can be complemented with other information resources for purposes of appraisal. Identifying and understanding these structures proved valuable for purposes of computational analysis. As an example, in our own tensor analysis research, we use the visualization to identify text datasets that are organized by year and have an orthogonal arrangement.

4.3 Personal records collections

Considering that most of the files belong to harvested websites, the testbed collection has significant structure built into its different RG. We wanted to compare if the visualization was also useful to represent non-structured collections such as those that are found in shared directories in networked organizations, or constitute personal collections for which there is no documentation about how the collection was created or maintained, nor index or catalogue, and precise information about them is difficult to obtain. We considered this a useful proof of concept. These types of collections are ubiquitous and considered problematic to be explored by archivists and records managers. We suggest that the analysis methods that we are employing can serve for preliminary exploration, to navigate, and to make inferences about these types of collections.

For this, we used a personal records collection consisting of 81,000 files kept over time on the shared directory of a private institution. This experience allowed introducing some different features to the visualization and to work with dates included in the file properties to determine if we could track the evolution of collections over time.

4.4 Using the metadata from the RDBMS to build a written finding aid

We wanted to test if we could easily map the metadata stored in the RDBMS to a standard descriptive metadata schema that archivists can use as finding aid to the collection. Mapping to EAD did not work well given that that schema is meant to describe boxes and files. We decided to map it to a simple set of descriptive elements in which the nested branches are series and correspondent sub-series. The table below shows one mapping exercise.

Table 1: Mapping structural metadata to a custom-made descriptive schema.

Descriptive elements	content	structural/descriptive
RG 056	branch 0	
title	Include name of directory	General Records of the Department of the Treasury
author	Include Department name	Department of the Treasury
size	aggregate size of all leaves	
series 1	branch 1	
title	Include name of branch 1	Treasury Orders
number of files	aggregate total number of leaves 4	
types of files	aggregate number of files by file format for leaves 4	
sub-series 1.1	branch 2.1	
title	Include name of branch 2.1	By Number
subjects/toc	Include all the names of encompassing branches 3 as a TOC	100 Secretary of the Treasury, 101 Heads of Bureaus, 102..
number of files	aggregate number of leaves 4	
sub-series 1.2	branch 2.2	
title	Include name of branch 2.2	By Subject
toc	Include all the names of encompassing branches 3 as a TOC	Advances, Alcohol and Tobacco, Alternative Dispute
number of files	aggregate number of leaves 4	
sub-series 1.3	branch 2-3	

title	Cross Index	Cross Index
toc	it does not exist in this case	
number of files	aggregate number of leaves 4	

We included basic elements and treated sub-directories as series and sub-series, and from the directory labels we can identify how the collection was structured, if by subject, by date, by number, etc. This mapping was useful to understand the parsers and algorithms that will need to be built to use this data to build written finding aids and will also illustrate and complement the data mining part of this research. However, the mapping system by itself, may not work to describe very nested or very disorganized RGs, as hierarchies do not necessarily map directly to what may be considered a series or sub-series.

5. CONTENT ANALYSIS

This aspect of the research complements the structural analysis of the collection (visualization). The analysis goal is to identify themes and the relationships within and across large and complex records collections (text) in connection to provenance and time. We are researching on two techniques, tensor analysis and paragraph alignment.

5.1 Implementation of tensors for archival analysis and access

Tensor mathematics is an extension of the ideas behind existing text mining techniques such as LSI (Latent Semantic Indexing). LSI improves on keyword searching by in effect searching on concepts. This method prevents finding documents that are accidental keyword matches, but unsuitable concept-wise, and enables finding documents that are conceptually relevant but may not contain certain requested keywords. Tensor mathematics improves on this by recognizing that concepts are not part of a linear spectrum, but may span multiple dimensions. For example, tensor analysis will make it possible to find relevant documents in the collection and introduce a time dimension to discern the evolution in the topic understood as its relevance over time. In this research we extract the relevant topics from the content of documents and the dimensions from the collection’s structural metadata described in the labels of the directories in which the documents are included. Given the diversity of labels that exist in the NARA testbed, these dimensions can be time, geographical places, document types (reports, budgets, manuals, etc), media formats, names, subjects, institutions, functions, roles, etc. At the moment we are experimenting with time but would like to add other dimensions in the future.

In the first months of this project we have reviewed the literature, determined software and hardware infrastructure, selected the collections to experiment with, and pre-processing the texts .

We set up the following packages:

- the Matlab Tensor Toolbox
- Wordnet
- the Natural Language Tool Kit (NLTK)

- and wrote tools to tie these packages together.

We targeted the following data collections, all of which had a temporal arrangement:

- RG 064 - Records of the National Archives and Records Administration/United States Government Manual
- RG 220 - Records of Temporary Committees, Commissions, and Boards/National Commission on Terrorist Attacks upon the US/govinfo.library.unt.edu
- RG 453 - Records of the United States Commission on Civil Rights [USCCR]/Press Releases/www.usccr.gov
- RG 046 – Records of the U.S. Senate/Speeches Given by Senator Barack Obama

The first pre-processing step was to build a vector space model. One of the advantages of the Matlab Tensor Toolbox is that it allows one to process texts that are available in diverse text formats (pdf, html, doc,). However, this tensor toolbox (the only available at present) is for prototyping and it will not scale up. In the future we will go with NLTK and Python. Our tokenizer is roughly based on the Porter stemmer in NLTK and we added a preprocessing step to extract proper names and add them to the token list explicitly, since they should not be stemmed. We show preliminary results for two datasets.

Directory Analyzed: /corral/tacc/dmc/nara/Federal_Records/RG046 – Records of the U.S. Senate/Speeches Given by Senator Barack Obama/obama.senate.gov/speech/

The directory contains HTML files with speeches delivered by Senator Obama over a period of 2 years from February 2005 to August 2007. The speeches address various issues like Medicine, War, Energy and Economy.

The height of the bars in the plot shows the occurrences of terms related to the respective subject over time.

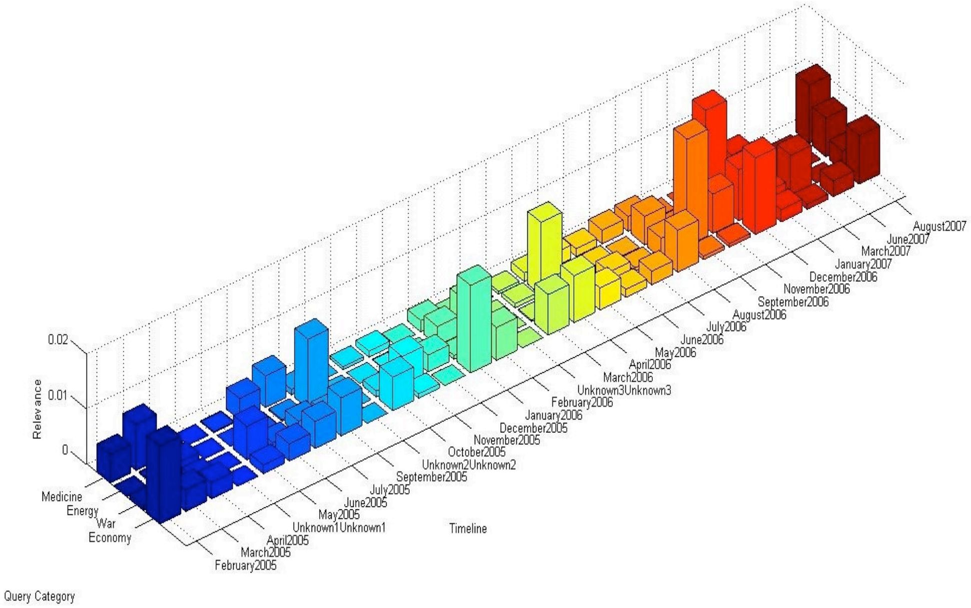
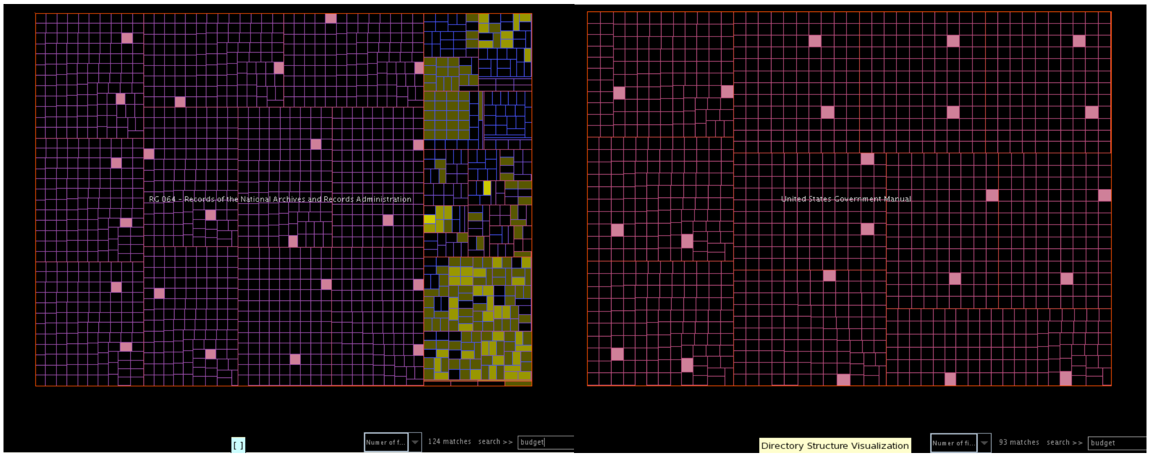


Figure 9: Tensor analysis of the speeches of Senator Barack Obama (2005-2007).

The following analysis corresponds to the RG 064 which contains a sub-set of documents, "US Government Manual," orthogonally arranged by agency and year. We selected this collection based on the analysis conducted through the visualization as we could detect its regularity and labeling. Figure 11 below shows snapshots of this record series. The image to the left shows all the series within the RG, the one to the right shows only the US Government Manual group. Highlighted in pink are the directories containing "budget" in their label.

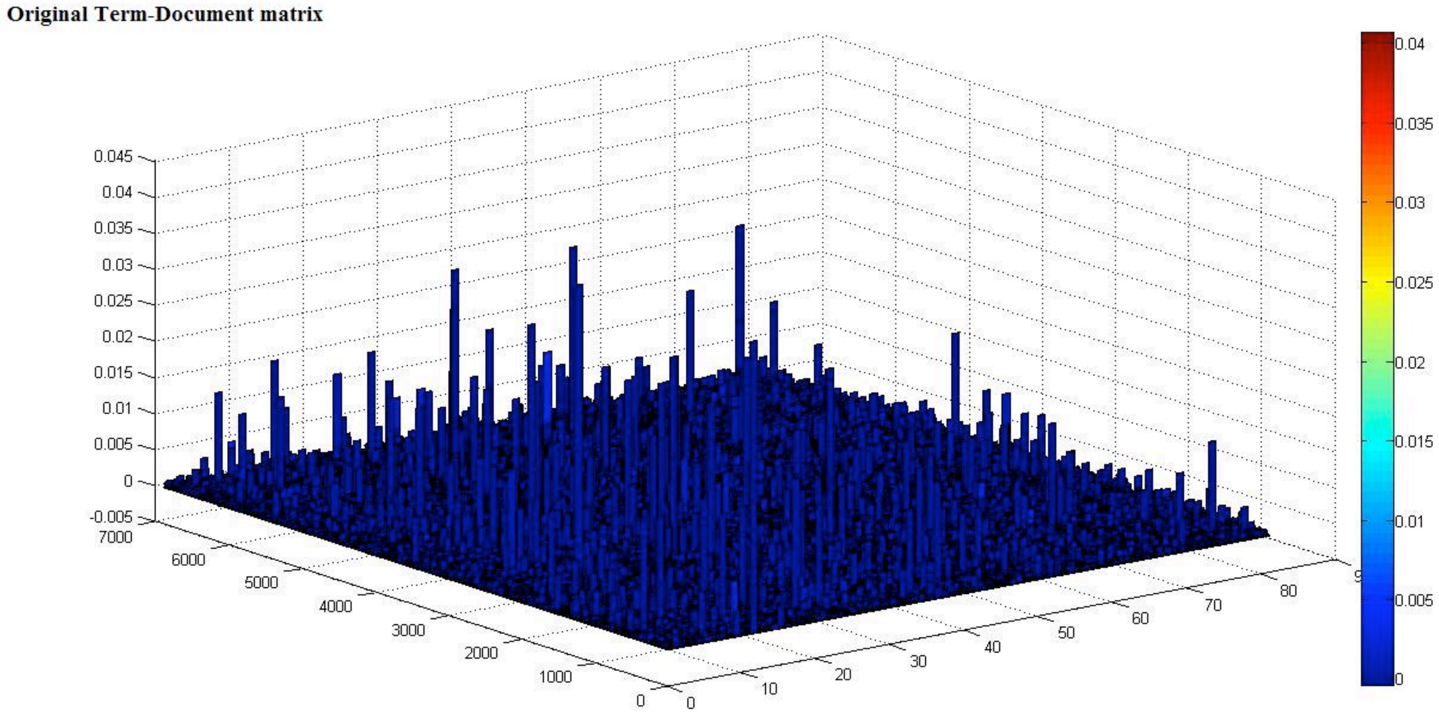

RG 064 - Records of the National Archives and Records Administration
[ ]
Numer of f...
United States Government Manual
Directory Structure Visualization
Numer of fi...

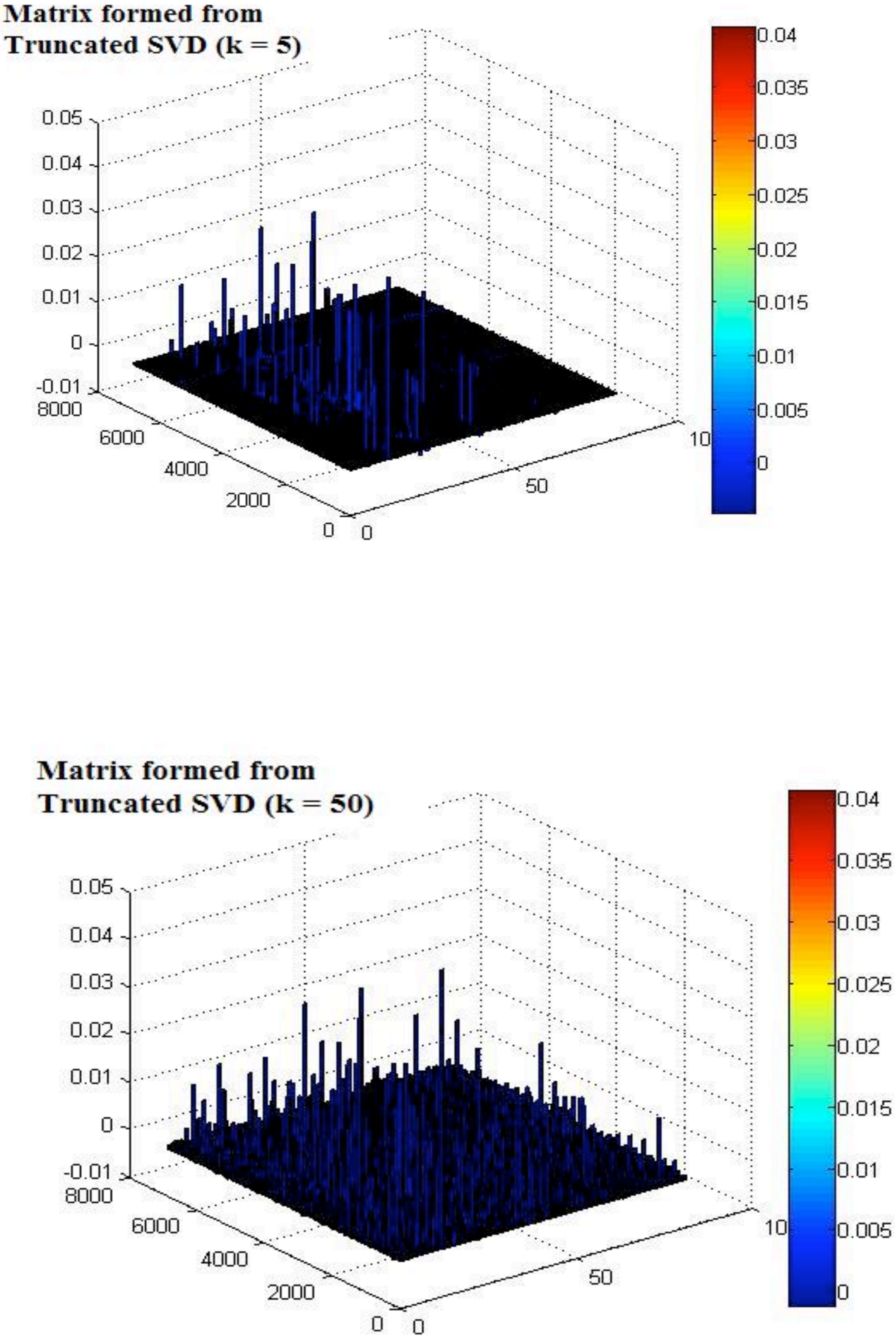
124 matches search >> budget

93 matches search >> budget

Figure 11. Orthogonal arrangement of the Record Series US Government Manual.

We have started experimenting with multi-dimensional decompositions of the term-document matrix. Using a 3D CP (Canonical decomposition/Parallel factors) decomposition, we are working towards discerning tendencies over time: instead of simple query matching to documents (as is done with singular value decomposition (SVD) /LSI, we can now find the best matching document in a given time period, or determine in what time period a given concept was of particular concern, regardless the collection /agency.





Matrix formed from Truncated SVD (k = 5)

Matrix formed from Truncated SVD (k = 50)

Figure 10. Experimenting with truncated SVD.

Directory analyzed: /corral/tacc/dmc/nara/Federal_Records/RG 064 - Records of the National Archives and Records Administration/United States Government Manual

The LSI method was being used and an important requirement was the correct use of the value *k* in a truncated SVD of the original term-document matrix. The first image in the figure (top) shows the cityplot for the original matrix (rank = 81). The successive plots show the plots for the truncated SVD with *k* = 5 and *k* = 50. Clearly the higher *k* value reproduces the original data more accurately. We are experimenting with an automatic strategy for choosing *k*.

We will validate our techniques in two different ways. First we will construct artificial collections that we design to have a known behavior and our techniques will be validated if we are able to reconstruct this behavior. Second, we will inspect the NARA collection and manually check whether our results make sense and are optimal.

At the moment we are exploring some small datasets, which can still be processed on a single CPU, but in the future we will look into multithreading or parallelizing the calculations on a large-core-count shared memory machine or a distributed cluster.

6. ONGOING WORK

In this section we describe projects that we started almost at the end of this reporting period.

6.1 User testing of the visualization interface

We are still improving aspects of layout and navigation in the visualization, as well as investigating how to incorporate other abstractions to show for example file format risk assessment results.

Up to this point of the development, only the archivist in our research team has tested the visualization. Starting in mid February, we will have a group of digital librarians and archivists from our campus to test aspects of usability, visual literacy, and to learn how they see the application as a tool to explore, preserve, present, collaborate, conduct research, appraise and process electronic records. Based on the feedback we will improve the existing visualization.

Over the next phases of this research we will also use the visualization as a foundation to show content analysis results.

6.2 Enable remote visualizations and rendering

For everyday analysis of the treemap visualization we use our desktop screens. We execute the visualization application in any of our analysis servers (Colorado and or Spur) and query the database located in our storage resource. We use either VNC (virtual network computing), or local XDMX (distributed multi-head x project) to render the visualization in our desktop. In these cases, the latency involved in interacting with the visualization and rendering the results on the fly are not significant. However, conducting analysis of such large data on the regular desktop screen has visual limitations. To overcome them we use various work-such as zooming in and out, and repositioning the vis many times. We are interested in testing the visualization in large displays to conduct observations at greater detail, and to use screen space to open different RGs for comparison and contrast. We also consider the need for virtual collaboration and distributed analysis of collections by researchers and archivists in different locations.

Remote rendering on high-resolution displays offers the challenge of pushing high-resolution visualizations from the remote visualization machine to the display, and from the host node of the display to the rendering nodes. TACC's visualization resources, which are detailed in Section 7 Cyberinfrastructure, offer the possibility to test the treemap visualization in two types of large displays: a) Stallion, tiled (distributed rendering nodes) and b) Bronco, big screen display (one rendering node). To interact remotely we open a VNC server session in the remote vis resource Spur and render the visualization through a VNC viewer in the display systems. The VNC layer is what presents limitations in rendering the visualization's functionalities.

In both displays, and especially in Stallion, the resolution of the image is impressive, allowing for detailed observations of the static visualization. The problem arises when users interact with the visualization, as the latency significantly diminishes the application's functionality. The latency is less in Bronco than in Stallion due to the fact that the latter is a distributed rendering system and the former only has one rendering node. VNC is a technology designed as a desktop application and not suitable to handle interactivity of high-resolution visualizations in large displays. We will continue testing other alternatives. In the near future, we will test the visualization using Longhorn, our newest visualization and data analysis cluster. We will also test latency using XDMX in a smaller size tile display (3x3). Even though XDMX does not support remote interaction, we will test latency using it on a smaller size tile display (3x3). Other aspects such as the position and the movement of the users in relation to the large display while conducting analysis need to be further studied.

6.3 Data mining

Being able to visualize the structure/contents and file density of certain directories allowed us to identify the distribution data as well as the different types of arrangements of each RG Our next step is to use data mining techniques to find patterns and relationships in the data across the entire collection. In this case, we will be able to incorporate file format information and file naming conventions and investigate the collection at a more granular level. Based on our experience and the selection of data mining methodologies we assume that we may be able to:

- Detect the association between the names of the directories, the names of the files and the types of files using association rule mining.
- Predict naming conventions usage based on the location of, and words used in the directory/file names using existing natural language processing tools.
- Identify common arrangement patterns used at different hierarchical levels within and across the collections through classification.
- Identify trends in file format usage based on file format identification information.

In synthesis, we want to derive knowledge from the data stored so we can understand emerging data organizational trends. We want to test if these methods can help at all to understand large collections and if so, use them to help users access collections in different ways.

6.4 Paragraph alignment for email analysis

Email collections have been studied to find communication patterns between correspondents based on the emails' header information, which includes to, from, date, and subject. In this case we want to identify communication between various authors based on the email contents to trace messages about similar and connected activities across time using paragraph alignment.

Paragraph alignment is a method developed by Xu and Esteva to find related activities in groups of documents that have different lengths and or cover more than one activity. In this method we draw from local alignment of biological sequence in which sequences are broken into n-gram for similarity computations and then assembled to derive an overall similarity. Here we adapt a similar approach and compute the similarity between paragraphs to determine content relations between documents. We will test this method with email, which in many cases contains copy/pasted paragraphs as well as threads of other emails.

For this research we are using the ENRON email dataset, which was available as txt files exported from their original email application. The dataset maintains the order in which the staff members kept and organized their emails. Throughout the research process we preserve the information about the original location of the messages in their naming convention, which allows us to trace not only authorship but also the structural location of the email. We have installed all the text analysis open source programs and the pieces of code needed to process the data. We are currently testing the scalability of the different software pieces for a corpus of almost 500.000 emails.

7. CYBERINFRASTRUCTURE

For this project we use the following resources:

7.1 Colorado

Colorado is a dedicated analysis and web server for the NARA project. The reasoning for using a server of our own instead of using the TACC clusters is as follows. While in the future we envision using large-scale clusters like Ranger, in the development phase of our project having a dedicated server is far more efficient. Ranger is entirely batch-oriented, which means that the turnaround time on a single run can be hours, up to days in busy periods. Clearly this is not acceptable in a development phase where large numbers of small tests need to be run. Furthermore, having our own server makes installing custom software possible, which on Ranger this is an involved process. Since we are targeting large-scale computations, we have chosen for Colorado a 24-core setup, so that fairly realistic parallel testing is possible. The combined memory on this server makes it possible to test problems that go far beyond the capabilities of a typical desktop machine.

Below are the specs:

4 Six core processors (1 Intel(R) Xeon(R) CPU E7450 @ 2.40GHz x 24)
OS: RHEL 5.4
Kernel version: 2.6.18-164.9.1.el5
MemTotal: 49452416 KB

There are 6 hard drives of 146 GB each divided in 3 logical disks (sda, sdb and sdc – reserved for future use--) mirrored using RAID-1

The server was configured and is maintained by Freddy Rojas, from TACC’s Advanced Computing Team. He is also in charge of installing all the software packages which significantly facilitates our research.

7.2 Corral

The NARA testbed and the database are stored in Corral a data storage system deployed in April 2009 by TACC to support data-centric science. Corral consists of 1.2 Petabytes of online disk and a number of servers, including a MS SQL server, providing high-performance storage for all types of digital data. Corral supports MySQL and Postgres databases, high-performance parallel file system, web-based access, and other network protocols for storage and retrieval of data to and from sophisticated instruments, HPC simulations, and visualization laboratories. Corral is also mounted to our dedicated NARA analysis server Colorado allowing a seamless transition between storage and analysis.

7.3 Spur

To render the treemap visualization we use Spur, TACC's Terascale Sun Visualization Cluster, containing 128 compute cores, 1 TB aggregate memory and 32 GPUs to interact remotely with the visualization through a VNC server. Spur shares the InfiniBand interconnect and Lustre Parallel file system of Corral, which allows us to perform visualization tasks without migrating the data to another file system and to integrate rendering tasks on a single network fabric. To render the visualization we currently use one node and interact remotely with the visualization through a VNC session.

7.4 Visualization Displays

7.4.1 Stallion

The Stallion cluster provides users with the ability to perform visualizations on a large 15x5 tiled display of Dell 30-inch flat panel monitors, for 307 megapixel resolution. This configuration allows for an exploration of visualizations at an extremely high level of detail and quality. The cluster allows users to access to over 36GB of graphics memory, 108GB of system memory, 19TB aggregate local disk storage, and 100 processing cores. A large, shared file system is available to enable the storage of terascale size datasets.

7.4.2 Bronco

The flat screen area gives users a 20 ft. x 11 ft., 4096 x 2160 resolution display, which is driven by a Sony SRX-S105 overhead projector and a high-end Dell workstation. This configuration provides users with the added flexibility to run a wide variety of applications, as only one workstation is required to drive the display. The projector gives exceptional brightness and a high resolution, 9M pixel viewing area.

8. CONFERENCES AND OUTREACH

We presented preliminary aspects of this research in the following venues:

- SAA Research Forum 2009
- Nomadic 09, Histories of Art and Sciences, University of Porto, Portugal
- ThatCamp, a Digital Humanities event on the UT campus.
- Visualization Lab Open Houses
- Dr. Weijia Xu used examples of the treemap visualization during his Fall 2009 Data Analysis and Visualization course in Scientific and Statistical Computing in UT Austin.

We have submitted papers to the following conferences:

- Digital Humanities 2010
- Archiving 2010
- JCDL 2010

9. FINANCIAL REPORT

Computational Analysis and Visualization of Large-Scale		
Electronic Records Collections for Archival Analysis		
National Archives and Records Administration		
Personnel	%	9/1/09 - 12/31/09
Victor Eijkhout, Research Scientist	0.30	$11,788
Maria Esteva, Research Associate	0.40	$10,933
Weijia Xu, Research Associate	0.30	$5,924
Graduatee Research Assistant	0.50	$7,917
Graduate Research Assistant	0.50	$7,917
Salary		**$44,479**
Fringe		$12,620
Travel		
Total Direct Costs		**$57,099**
UT Overhead (50% of MTDC)		$28,549.50
Compute Server (Capital over $5K)		$14,359.80
Tuition		$7,872.00
Total Expenditures		107,880