Interactive Poster: Creating and Managing "Lookmarks" in ParaView

Eric T. Stanton

W. Philip Kegelmeyer

etstant@sandia.gov wpk@ca.sandia.gov

Sandia National Laboratories

ABSTRACT

This paper describes the integration of *lookmarks* into the ParaView visualization tool. Lookmarks are pointers to views of specific parts of a dataset. They were so named because lookmarks are to a visualization tool and dataset as bookmarks are to a browser and the World Wide Web. A lookmark can be saved and organized among other lookmarks within the context of ParaView. Then at a later time, either in the same ParaView session or in a different one, it can be regenerated, displaying the exact view of the data that had previously been saved. This allows the user to pick up where they left off, to continue to adjust the view or otherwise manipulate the data.

Lookmarks facilitate collaboration between users who wish to share views of a dataset. They enable more effective data comparison because they can be applied to other datasets. They also serve as a way of organizing a user's data. Ultimately, a lookmark is a time-saving tool that automates the recreation of a complex view of the data.

1 BACKGROUND

ParaView is an open source, multi-platform visualization application with a focus on supporting large datasets. It is built on the Visualization Toolkit (VTK) and uses its visualization pipeline model consisting of readers and filters. A *reader* is an object that inputs a single dataset of a certain type and forms the beginning of the pipeline. The way VTK operates on a dataset is by setting the input of a process object called a *filter* to the output of the reader or of another filter. Thus, a tree of filters spawning from a single reader takes form, each performing a particular operation on the data. The reader and each filter of this tree can be set "visible" or "invisible"; only visible filters are used to render the current view.

2 DESIGN

The following is one way of conceptually dividing up the lookmark functionality. Each section describes the role that component plays in the design.

2.1 The Lookmark

A lookmark represents all the information necessary to reconstruct a view of a dataset in ParaView. Its ParaView state script captures the state of the filters that contributed to the view at the time the lookmark was created. Its thumbnail of the view is used to provide a visual cue to the user. Additionally, it contains a name, the path to the dataset, and comments the user might have made about the lookmark.

In designing lookmarks in ParaView, the analogy of bookmarks in a web browser was often used as a guide. Thus it seems logical that an interface equivalent to a browser's bookmark manager would be necessary for a user to view and organize lookmarks.

2.2 The Lookmark Manager

The Lookmark Manager is a user interface integrated into ParaView (see Figure 1). It is here that the user can import lookmarks, toggle from one to another, save and remove them, and create new lookmarks of the data. In addition, the Lookmark Manager is hierarchical so that the user can organize the lookmarks into nested categories.



Figure 1: A Lookmark Manager

2.3 The Lookmark Widget

The user interacts with lookmarks through the lookmark widgets displayed in the Lookmark Manager. Each widget contains a thumbnail preview of the lookmark, a collapsible comments area, the name of the default dataset on which to apply the lookmark, and the name of the lookmark itself (see Figure 2). A singleclick of the thumbnail generates that lookmark in the ParaView window. A checkbox placed in front of each widget provides multiple lookmark selections for updating, moving, and removing lookmarks (see Section 3 for more information).



Figure 2: A Lookmark Widget

2.4 The Lookmark File

The state of all lookmarks in the Lookmark Manager can be saved to a text-based, XML-formatted *lookmark file*. This file maintains the hierarchical structure of the original Lookmark Manager. It can be imported into the Lookmark Manager of any ParaView session because the lookmarks it contains are not tied to a specific dataset. Each XML tag represents a lookmark and its name-value pair attributes correspond to the attributes of the lookmark.



3 USE CASES

3.1 Adding a Lookmark of the Current View

To add a lookmark of the current view to the Lookmark Manager, first a lookmark object is initialized and saved in an internal data structure. This object's attributes include a unique name of the form *Lookmark*#, the current dataset's file path, the ParaView session state script, a data structure storing references to the filters that contribute to the current view, and a base64-encoded thumbnail of that view. Second, a lookmark widget is created using these attributes and is appended to the bottom of the Lookmark Manager.

3.2 Generating a Lookmark in the ParaView Window

When a lookmark is clicked, all of the existing ParaView filters in the filter window are first set invisible so that they do not affect the view. Then the state script stored with the lookmark is used to set various attributes of the reader, such as timestep, variables to be turned on, and coloring. The rest of the state script is then run. What results is the creation of a filter tree originating at the reader that generates the desired view of the data in the ParaView window (see Figure 3).



Figure 3: Clicking the lookmark in (a) generates the view in (b)

In Section 2.4 it was said that lookmarks are not tied to any particular dataset. Therefore, there must be a mechanism in place to control the dataset to which a lookmark gets applied. When a lookmark is first added, the dataset is stored as that lookmark's *default dataset*. When the "Apply to default dataset" toggle in the Lookmark Manager is checked, any lookmark that is clicked will be applied to its own default dataset, which is opened first if it is not already. Only if the dataset cannot be found at its last known path will the user be prompted for a new path. When the "Apply to default dataset" toggle is *not* checked and a lookmark is clicked, the user is first asked to select a dataset from either a list of datasets currently open in ParaView or from the file system, with the option of saving it as the new default dataset.

When a lookmark is clicked, the names of the filters it creates will have the lookmark name appended to them. This distinguishes the lookmark's filters from others in the filter window. Clicking this same lookmark again will cause these to be deleted if possible (i.e. if they have *not* been set as inputs to other filters), and the lookmark is then generated as described above. This enables the user to operate in "browsing" mode within the Lookmark Manager, visiting lookmarks at will while avoiding duplicate sets of filters.

3.3 Removing Lookmarks

Just as a lookmark can be added to the Lookmark Manager, it can also be removed. Doing so deletes both the lookmark object from memory and the lookmark widget from the Lookmark Manager. Furthermore, any number of lookmarks can be removed at the same time.

3.4 Modifying Lookmarks

The lookmark widget can be modified in four ways. Its name can be changed by double-clicking the current name, typing a new name into the text field, and then pressing the Enter or Return key. Comments can be written in the designated text area to annotate the lookmark. As described in the previous section, its default dataset can be changed. Finally, the actual view that a lookmark will recreate can be modified as well. Once a lookmark has been generated, it can be updated by the user, which means storing the state of the current view with that lookmark and updating its thumbnail accordingly.

3.5 Saving Lookmarks to a Lookmark File

By saving the contents of the Lookmark Manager, the user is capturing both the state of each lookmark and its location in the hierarchy, all in a single lookmark file. This file can be imported to a different ParaView session, bringing the Lookmark Manager to the same state it was when the save operation took place.

3.6 Importing Lookmarks into the Lookmark Manager

When the user first starts ParaView, the Lookmark Manager is empty. The only actions available to the user are to import the lookmarks contained in an existing lookmark file into the Lookmark Manager or to add a new lookmark of the current view of the dataset. The former is initiated by pressing the "Import" button in the Lookmark Manager and selecting a lookmark file from the file chooser window. Before doing this, however, the user must decide whether the lookmark file's contents should replace or append to any existing lookmarks in the Lookmark Manager and select the appropriate checkbox. Once the lookmark file is selected, the attributes of each lookmark entry in the file are used to initialize a new lookmark object and lookmark widget as described in Section 3.1.

4 SUMMARY

Lookmarks provide a way for analysts to save and manage views of what they consider to be the important regions of a dataset, in a fashion analogous to how bookmarks are used by a web browser. They make use of ParaView state scripts without duplicating data. They automate the mundane task of recreating a complex filter tree. Through the use of a lookmark file, lookmarks can be shared with peers for collaboration. They make it possible to toggle back and forth between completely disparate views. By making it easy to record and recreate views, lookmarks aid in breaking down very large datasets into manageable pieces and in better understanding the behavior of the data.

ACKNOWLEDGEMENTS

This research was supported by the Department of Energy through the ASC VIEWS Data Discovery Program, Contract number: DE-AC04-76DO00789.

REFERENCES

KITWARE, INC. 2003. The VTK User's Guide. Kitware, Inc.

KITWARE, INC. http://paraview.org.