An Annotation System for 3D Fluid Flow Visualization

Maria M. Loughlin
Cambridge Research Lab
Digital Equipment Corporation

John F. Hughes
Department of Computer Science
Brown University

Abstract

Annotation is a key activity of data analysis. However, current systems for data analysis focus almost exclusively on visualization. We propose a system which integrates annotations into a visualization system. Annotations are embedded in 3D data space, using the Post-it® metaphor. This embedding allows contextual-based information storage and retrieval, and facilitates information sharing in collaborative environments. We provide a traditional database filter and a Magic Lens® filter to create specialized views of the data. The system has been customized for fluid flow applications, with features which allow users to store parameters of visualization tools and sketch 3D volumes.

1 Introduction

In a study to characterize the data analysis process, Springmeyer et al. [15] observed scientists analyzing different types of scientific data. The study found that recording results and histories of analysis sessions is a key activity of the data analysis process. In each session, the scientists recorded notes, and inspected previous notes. Two distinct types of annotating were observed:

• recording, or preserving contextual information throughout an investigation

and

• describing, or capturing conclusions of the analysis sessions.

Despite the importance of annotation, current systems for data analysis emphasize visualization, focusing on the generation of visual displays. Little or no annotation support is available: for example, Springmeyer et al. noted that the recording media used by scientists in their study included notebooks, scratch paper, and Post-it notes.

In this paper, we describe a system that supports annotation as an integrated part of a fluid flow visualization system. Unlike typical annotations on static 2D images, our system embeds annotations in 3D data space. This immersion makes it easy to associate user comments with the features they describe. To avoid clutter and data hiding, annotations are represented by graphical annotation markers that have associated information. Therefore graphical attributes of the markers, such as size and color, can be used to differentiate annotations with different functions, authors, creation dates, etc.

Annotations can easily be added, edited and deleted. Also, multiple sets of annotations can simultaneously be loaded into a visualization. This allows scientists, collaborating on a data set, to use annotations as a form of communication, as well as a history of data analysis sessions. Annotation markers also aid scientists in navigating through the data space by providing landmarks at interesting positions. Figure 1(a)-(c) shows the visualization environment, annotation markers, and the annotation content panel. Figure 1(d) shows a Magic Lens filter which hides the annotation markers and widget handles. The implementation has been applied to three-dimensional Computational Fluid Dynamics (CFD) applications. However, the techniques can be used in visualization systems of many disciplines. The design can also be extended to 3D stereo and virtual-reality environments.

The rest of this paper is organized in five sections. In section 2 we review previous approaches to annotation. Section 3 describes design guidelines for annotation systems. Section 4 details our implementation of annotation within a 3D modeling and animation system. In the last two sections, we discuss possible future work and present our conclusions.

2 Background

Scientific visualization systems provide little, if any, support for annotation. For example, Application Visualization Sys-
Figure 1: The visualization and annotation system
(a) hedgehog and streamlines showing 3D fluid flow, (b) annotation markers (small geometric objects) placed at points of high velocity, (c) annotation content panel, (d) Magic Lens filter hiding annotation markers and widget handles.
and the files are labeled to mark their association. This separation of data and annotations means that some effort is required to find the data features described by annotations. The 3D data space of many scientific applications provides the context in which annotations should be placed. Recording annotations in this space capitalizes on human’s spatial senses by facilitating the retrieval of information based on its spatial location in the visualization.

However insertion of annotations in the data space creates an immediate conflict between the annotation and visualization functions: both compete for screen territory. We do not wish to impose any restrictions on the amount of information that can be recorded. At the same time, since information is contained in the data itself, we do not wish data to be obscured by annotations. Our approach is to decompose an annotation into:

- an annotation marker or small geometric object that identifies the position of the annotation in the data space
- an annotation content in which a user stores information.

The geometry and graphical attributes of markers are chosen so that they are easily distinguished from existing visualization tools. By clicking on a marker, a user can expand the associated annotation to read or edit its content. Separation of the annotation’s content from the annotation marker in this way allows direct insertion of arbitrarily large annotations.

**Guideline 2:** Annotations must be powerful enough to capture information considered important by the user.

There are different types of information. Tanimoto [16] distinguishes between data (raw figures and measurements), information (refined data which may answer the users’ questions) and knowledge (information in context). Bertin [3] classifies the levels of information in a similar way. He considers information as a relationship which can exist between elements, subsets or sets. The broader the relationship, the higher the level of information. We assume that an annotation system should be able to store information at each of these levels - scientists need to record both the data values at probe points in the data set, and a higher-level analysis of these figures.

Although some data, such as date of creation and author, are likely to be relevant to all applications, it is possible that knowledge can be captured only when an annotation system is customized for a specific application. The customization would ensure that annotations can represent information relevant in the context of the application. For example, if the data of a particular application is time-varying, the annotation system should provide time-varying annotations that can track the features being described.

In our annotation system, we provide support for different types of information in two ways. First, within each
annotation, scientists can record both numerical and textual
details, and high-level information specific to fluid flow. This
is discussed in section 4.4. Second, the system supports
hierarchically-organized annotations. The hierarchical struc-
ture allows scientists to record facts in separate annotations,
and group related annotations in sets that describe broader
observations.

It is also important to consider the modalities that are
available for capturing information in an annotation system.
Two dimensional text, graphics and images are the standard
annotation modalities; aural annotation is also a candidate.
Chalfonte, in an experiment on the use of annotation for
collaborative document authoring, found aural annotations
a richer and more effective medium for high-level commu-
nication [5]. Freestyle shows that coordinating hand/cursor
movements with textual and aural annotations is also effec-
tive.

In our current implementation, we use 2D text and 3D
volumes to store information. In the future, we would like to
use different interaction techniques for information capture.

Guideline 3: We need to consider the established rules
of user interface (UI) design, because the UI of an annotation
system will play a key role in determining its acceptance (or
lack thereof) by scientists.

We considered many UI rules [7] and designed our an-
notation system accordingly. One rule states that a UI should
allow users to work with minimal conscious attention to its
tools. We achieve this goal by using a direct manipulation
interface, that is, an interface in which the objects that can
be manipulated are represented physically. For example, the
volume of data affected by the Magic Lens filter can be con-
trolled directly by moving and resizing the physical represen-
tation of the lens. Another design rule states that an interface
should provide feedback, e.g., on the current settings of do-
main variables. In our system, annotation markers give visual
feedback on the location of annotations and marker geometry
gives feedback on annotation content.

Because the geometric data space of fluid flow appli-
cations has three dimensions, we considered design issues
specific to 3D graphical user interfaces [6]. One issue is the
complexity introduced by 3D viewing projections, visibility
determination, etc. A second issue is that the degrees of free-
dom in the 3D world are not easily specified with common
hardware input devices. A third issue is that a 3D inter-
face can easily obscure itself. We use guidelines outlined by
Snibbe et al. [14] to deal with these problems. For example,
we provide shadows, constrained to move in a plane, to sim-
plify positioning of annotation markers (see section 4.3.2).
We provide feedback on the orientation of the data by option-
ally drawing the principal axes and planes. We also ensure
that annotations do not obscure data, by making it easy for
a user to change the viewpoint and resize or hide annotation
markers.

4 Implementation

This section describes the annotation system we have imple-
mented. We begin by setting a context for the system with a
description of fluid flow visualization and the software devel-
opment environment. Then we discuss the main components
of the annotation system: the annotation markers, support for
information capture, and interaction techniques.

4.1 Fluid Flow Visualizations

Computational fluid dynamics (CFD) uses high speed com-
puters to simulate the characteristics of flow physics. Com-
puted flow data is typically stored as a 3D grid of vector and
scalar values (e.g., velocity, temperature, and vorticity val-
ues), which are static in a steady flow, and change over time
in an unsteady flow. CFD visualization tools allow a scientist
to examine the characteristics of the data with 3D computer
displays.

Interaction with the visual representation is essential in
the exploration and analysis of the data, and has three goals:
feature identification, scanning, and probing [9]. Feature
identification techniques help find flow features over the en-
tire domain, and give the scientist a feel for the position of
interesting parts of the flow volume. An example of this type
development system is a vector hedgehog, a three-dimensional
array of velocity vectors. Scanning techniques are used to inter-
actively search the domain, by varying one or more parame-
ters, through space or through scalar and vector field values.
Scanning techniques include cutting planes (planar surfaces
which slice the 3D grid and show scalar field value at each
grid point of the plane) and iso-surfaces (three-dimensional
surfaces of a constant scalar value). Probing techniques are
localized visualization tools, typically used to gather quan-
titative information in the final step of investigating a flow
feature. Examples of probing tools include streamlines and
particle paths, which show the path in which a particle would
flow if positioned in a steady or unsteady fluid flow.

The Computer Graphics Group at Brown University has
developed a flow visualization system, to study new modes
of interaction with flow tools. The annotation system was
developed as part of this flow visualization system. This
provided a test-bed for techniques to integrate visualization
and annotation functionality.

4.2 The Development Environment

The annotation system was developed using FLESH, an ob-
ject oriented animation and modeling scripting language [11],
and C++. In the FLESH language, scenes are described as
collections of objects. The FLESH objects defined for the an-
notation system include geometric objects such as annotation
markers, 3D volumes and lenses, and non-geometric objects,
such as holders for collections of annotations and an annota-
tion filter. Some of these FLESH classes have corresponding
C++ classes, in which data is stored and compute-intensive
operations performed. This allows us to benefit from the
4.3 Annotation Markers

Annotations are represented in the 3D data space by small geometric markers. Each marker has an associated content which the user can edit at any time.

4.3.1 Marker Geometry and Graphical Attributes

The geometry of a marker gives visual feedback on the content of the annotation. In the fluid flow visualization system, the user can define annotation keywords (e.g., plume, vortex), and select a geometry to associate with each keyword. Then, when the user assigns a keyword to an annotation in the system, the annotation’s marker takes the associated shape. It is likely that other mappings between graphical attributes of markers and annotation content would also be useful. For example, the color saturation of a marker could depend on the age or priority of the annotation.

The graphical attributes of annotations are also user-customizable. The size and color of all markers in one level of hierarchy can be changed. We predict that this feature would be useful if many scientists work collaboratively on a data set, and each scientist defines a unique color and size for her markers.

4.3.2 Marker Behavior

Since the function of a marker is simply to identify points of interest in the visualization, its behavior is quite simple. A marker is created when the user presses the annotation push-button. It appears at the point on which the user is focussed, making it easy for the user to position it near the feature of interest.

The scientist can translate and rotate markers with simple mouse movements. He can also project interactive shadows of the marker on the planes defined by the principal axes [10]. Each shadow is constrained to move in the plane in which it lies. If a user moves a shadow, the marker moves in a parallel plane. This constrained translation helps in precisely positioning a marker.

Markers can be highlighted in response to a filter request. In the current system, the color of a marker changes to a bright yellow when highlighted. This simple approach seems adequate. However, the user may change this highlight behavior, by, for example, having highlighted markers flash between alternating colors.

Since the features of unsteady fluid flows change over time, a user would like the annotation describing a particular feature to follow the feature’s movement in the visualization. The current annotation system provides partial support for this by allowing the user to specify the position of an annotation at any number of points in time. The annotation markers then linearly interpolate between the specified positions in time.

4.4 Knowledge Stored

Our annotations can store generic information, as well as information specific to fluid flow applications. The generic information includes keyword, textual summary and description, author, and date. Some of this information (author and date) are captured implicitly when the annotation is created. The rest must be explicitly added after the scientist has opened the annotation by clicking on it. This data entry is performed via a 2D Motif panel of buttons and text widgets. We consulted with fluid flow experts to understand how the information content of annotations could be customized for fluid flow applications.

4.4.1 Parameters of Visualization Tools

One of the key additions to the annotation system suggested by the fluid flow experts results from the interactive nature of fluid flow analysis. As described earlier, a scientist must insert flow visualization tools (such as streamlines and iso-surfaces) in the data space to see the underlying data. Much time is spent determining which tools most effectively highlight a feature, and positioning and orienting both the tools and viewpoint to best show off the feature being described. Springmeyer et al. observed this activity of the data analysis process, and described it as orientating the data, or altering a representation to gain perspective.

To support this activity, our concept of an annotation was expanded to include parameters of flow visualization tools. When a user wishes to store the parameters of a set of tools, he or she presses a button to indicate that a set of tools is being saved, and then clicks on the tools of interest. The time-varying location, orientation, size, and other parameters of the tools are saved with the annotation. This can be repeated any number of times for different groupings of tools with different parameters. When an annotation is restored, the user is presented with a list of all saved sets of tools, and can recover each set of tools to see how they illustrate the annotated feature.

4.4.2 3D Volume Descriptions

It also became obvious that annotation markers, which are appropriate for locating point features in a visualization, are not sufficient for CFD applications. Fluid flows contain volume features, such as vortices (masses of flow with a whirling or circular motion) and plumes (mobile columns of flow). Users may want to associate an annotation with a region of the data space, rather than a single point in the space. We therefore need a way to sketch a volume in the data space. The volume-sketching method must be intuitive, so that flow scientists (who may not be interested in becoming artistic volume sculptors!), can easily describe the volume. Also, the resolution of the volumes sketched need only be as precise as the grid on which the flow field is defined.

We provide a simple method to sketch volumes. The user positions “pegs” that define the extreme vertices of the volume to be drawn. The pegs are created and moved within
the visualization in a way similar to the creation and translation of annotation markers. When the user is done positioning pegs, the system computes their convex hull using the quickhull algorithm [2]. The boundary of the volume, defined by the convex hull, is rendered in either wireframe or transparent mode. Vertices can be added, deleted and moved, and the volume redrawn, until the volume is accurate. Figure 2 shows a volume which has been defined in this way.

This implementation provides a simple means to draw volumes. However, since it uses the convex hull of the pegs, certain shapes, such as a 3D “L” shape, cannot be sketched.

4.5 Retrieving the Annotations

Effective information retrieval and communication requires that a user can easily identify annotations relating to a specific topic, by a specific author, etc. The annotation system facilitates such data filtering in two ways.

First, a traditional database filter is provided. The user can specify data selection criteria (such as the annotation creation date, author, or keyword), via a 2D Motif panel. The markers of annotations that satisfy the search criteria are highlighted.

A second filter uses the Magic Lens metaphor introduced by Bier et al. [4]. A Magic Lens filter is a rectangular frame, placed in front of the visualization, that appears as if it moves on a transparent sheet of glass between the display and the cursor. The lens performs some function (which may use information from application-specific data-structures) on the application objects behind it.

Four functions are defined for the lens in the annotation system. The first sets the color of all objects, except annotation markers, to gray. This helps users find markers in a cluttered scene. The second lens function displays only the annotations that satisfy the criteria specified in the Motif database filter. The third lens function hides all annotation markers behind the lens. Finally, the default function hides all annotation markers and all interaction handles on the visualization tools behind the lens. Many other interesting lens functions could be defined. One such function could remove all fluid flow tools except those in the user sketched volume behind the lens.

We believe that the magic lens filter alleviates the problem of visualization and annotation functions sharing the same screen space. Using the lens, a scientist can tightly integrate the two functions when appropriate. When she wishes to focus exclusively on either visualization or annotation, the clutter introduced by the other component can be hidden.

5 Future Work

The work described in this paper could be expanded in a number of ways, in both the fluid flow application and in new environments.

There are a number of opportunities for the fluid flow application. The facility for recording parameters of visualization tools could be extended to record view parameters. Then, flow tools could automatically be viewed from the same viewpoint and with the same magnification as when their parameters were saved. Annotations could also become more active in the data investigation process. For example, annotation markers could be used as seed points for automatic flow feature-characterization code. The output of the feature-characterization code (i.e., specifications of the feature found) could then be added to the annotation content. Feature-characterization code could also be used to improve support for time-varying annotations. If the location of an annotation marker were constrained to the feature's position (as found by feature-characterization code), the marker would follow the movement of the feature over time.

We would also like to implement annotations in other applications and environments. For example, virtual reality environments pose many new research problems. User studies would have to be performed to determine which annotation modalities would be appropriate in this space. If textual annotations were appropriate, we would have to determine where to place the text: floating in space near the marker, or on 2D panels which exist in the virtual space, or perhaps in some other place. New interaction mechanisms for annotation markers and filters should also be developed.

Finally, we would like to expand the scope of annotations. Springmeyer et al. noted that scientists tend to record their interactions with visualization systems. Perhaps the annotation system could help in recording and examining these edit trails. Also, scientists spend time comparing different data sets. The current annotation system could be redesigned to fit in the context of more than one data set.

We hope that further experience with the current system and its extension to other applications and environments will allow us to evaluate our design guidelines, and develop principles for customization of a general purpose annotation system.
6 Conclusion

The importance of annotation in data analysis and the lack of annotation support in data analysis tools led us to develop a system that integrates annotation and visualization. In our system, annotations are embedded in the 3D space of CFD data. Co-location of annotations and data allows users to navigate through the information by spatial association. Each annotation is composed of a small geometric marker and a content that can include textual, graphical and other domain-specific information. This allows unobtrusive annotations with an unlimited amount of information. Filters are provided to help sort annotations and create customized views of the information.

Initial feedback from scientists leads us to believe that the close integration of annotation and visualization facilitates the ongoing recording activity observed by Springmeyer et al. At the same time, the ability to group annotations in disjoint sets and filter annotations supports the organization of analysis conclusions, i.e., the describing activity. Furthermore, annotations can be used as a means of communication between collaborating scientists, and as a way to present information in an educational tool.

7 Acknowledgments

The authors of this paper would like to thank the members of the Graphics Group at Brown for their helpful comments and support. Thanks also to the members of the Visualization group at Digital Equipment Corporation’s Cambridge Research Lab for their review of this paper. The paper is based on the Master’s thesis of the first author, whose attendance at Brown University was made possible by Digital Equipment Corporation’s Graduate Engineering Education Program. This work was supported in part by grants from Digital Equipment Corporation, NSF, DARPA, IBM, NCR, Sun Microsystems, and HP.

References


