

ComVis: a Coordinated Multiple Views System for Prototyping New Visualization Technology

Krešimir Matković

VRVis Research Center

Matkovic@VRVis.at

Wolfgang Freiler

VRVis Research Center

Freiler@VRVis.at

Denis Gračanin

Virginia Tech

gracanin@vt.edu

Helwig Hauser

University of Bergen

Helwig.Hauser@uib.no

Abstract

There is a large number of interactive visualization tools, however no universal tool exists that covers all relevant aspects for all possible application domains. We have developed a tool, ComVis, which was intended to be used as a research prototype for new visualization techniques. We have identified some interesting aspects from developers and users point of view during tool development. In this paper we describe lessons learned during the process, and share our findings with visualization research community. Examples at the end prove the usefulness of the developed tool. One particular example, the concept of families of function graphs and application to analysis of fuel injection concludes the paper.

Keywords— Information visualization, visual analysis, visualization tool, system design.

1 Introduction

Coordinated multiple views are a well established and proven technology for interactive visual analysis [10] and, at the same time, an active area of research. A number of available tools and systems successfully capitalize on this approach, ranging from research prototypes to commercially available tools. Some of these systems are designed with a specific application domain in mind while the others are more general.

The system design is very much dependent on the intended use. The commercial tools undergo strict design and specification phases, after which there are no significant changes. On the other hand, the research tools follow more evolutionary, or even rapid-prototyping development process. They will almost never reach stability of the commercial tools, but can be used as a prototype to quickly evaluate new ideas.

We had decided to design such a research tool called ComVis that would allow rapid prototyping and testing of new interactive visual analysis techniques and paradigms. Therefore, the tool and the development team (students) may change often while new ideas and designs can be fre-

quently tested. Later, if they prove to be useful, they will be implemented in an optimized way within the scope of a more mature tool, and for a specific application domain.

It is interesting to take a look back after developing such a tool and identify what issues and solutions are relevant for other tools. In this paper we describe lessons learned and discuss some of the most important design aspects and share them with the visualization research community.

In the remainder of the paper we first discuss the related work and concepts, followed by an overview of the system design and related issues. A description of several successful case studies and related experiences conclude the paper.

2 Related Work

A complete survey of available interactive visualization tools (commercial and academic) exceeds the scopes of this paper so we only discuss some selected tools. The related work section of Yang's dissertation [15], however, provides a very good survey.

Many of these tools cope with multidimensional data and can visualize large data sets. However, no universal tool exists that covers all relevant aspects for all possible applications, since the requirements and goals are very different in different application domain. Still, there are some very good and useful systems we would like to mention.

The department of Computer Oriented Statistics and Data Analysis at the University of Augsburg offers very powerful visualization tools. The tool used for exploratory analysis of large multidimensional data is called Mondrian [13]. It is a general purpose tool which supports multiple linked views and interactive brushing.

Another well known tool is GGobi [11] (and its predecessor XGobi). GGobi supports multiple linked views and can be connected to the R statistical analysis system.

The development of Xmdv tool [14] started in 1993. Xmdv supports multiple linked views and advanced interactions like brushing and linking or zooming and panning.

The IVEE system [1] was a research platform which later became Spotfire. Spotfire is probably the best known and most widespread commercial visualization tool today.

There are various products in the Spotfire suite, tailored for different application domains and they certainly set a high standard for commercial interactive visualization tools.

Inxight [3] is a spin-off from Xerox PARC and they offer StarTree, Table lens and Time Wall tools. An interesting comparison of commercial tools (Spotfire, Table lens and InfoZoom) is provided in [5].

The Tableau Software [12] is another commercially successful and very innovative tool.

The visualization group at the University of Konstanz offers solutions for huge data sets. The solutions range from VisBD [4] developed in the mid 1990s to recent work including large displays and pixels based techniques.

The NVAC developed several tools used to analyze huge data sets such as In-Spire and Starlight [9]. They are intended for visual analytics customers with high demands.

Finally there are also toolkits which make it possible to design new information visualization systems based on generally useful functionality. The InfoVis toolkit [2] is one of the best known relevant examples.

3 Overall Design Goals

Even though a number of tools already exists, new tools are created over and over due to some specific requirements and design goals. Our initial design goal was rapid prototyping of new visualization techniques within the scientific context. New ideas should be easy to test, and it should be possible for students to start coding easily. In order to support these basic requirements, the system has to be:

- flexible in a way that adding new views and supporting new data types is easy,
- intuitive to use and at the same time also support advanced interaction (multiple, iterative brushing),
- easy to use for domain experts from different domains (medical, engineering, etc.), and
- able to read generally used data formats to provide access to existing data

Figure 1 illustrates the high level system design of developed ComVis tool. There are two main components, the data manager and the list of views. The data manager takes care of data storage and retrieval. It also stores brushes information.

Each view has two main parts, the view data and the window itself. The view data depends on the view settings and contains actual data to display. Such an organization makes it possible to efficiently implement new views. All implementations are based on a common abstract view implementation that takes care of the common view tasks.

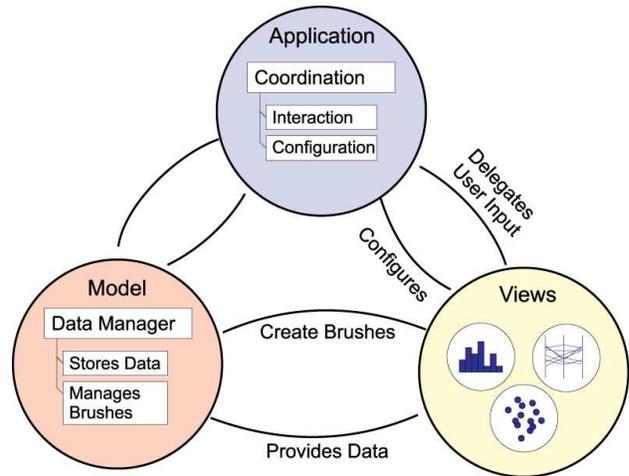


Figure 1: High level system design of ComVis tool.

4 Selected Design Solutions

Since it is not possible to describe in this paper all the aspects of the system design, we will illustrate the most interesting aspects and describe the lessons learned that can be useful to other researchers.

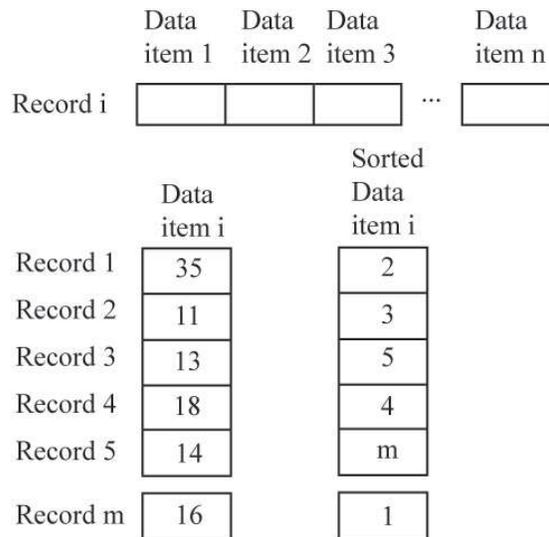


Figure 2: Column sorting and table lookup.

In a multivariate tabular dataset, all data items in a column are of the same data type while different columns can contain different data types. Therefore, it makes more sense to store data organized in columns than in rows. That especially applies to general purpose visualization tools where the attributes of the data records are not known during the design process. If we sort the columns first, data query can be very efficient. In order to keep the original

row information we use an additional indexed column for each column. The idea is illustrated in Figure 2.

There are many characteristics common to all views. One example is the use of axis and scales. We designed general axis and scale components to take care of mapping from absolute values to relative, pixel based, view values. The components are also used to map the values during zooming and to support logarithmic mapping. General axis and scales components make the view design much easier as data mapping is already solved for virtually all cases.

We identified many aspects and features that were appreciated by the users. Without those features the users would not be likely to use the system. The ease of use for visualization non-experts is crucial if the tool is to be used.

The ComVis supports multiple views, but all of them are fixed inside the main window. The user can choose from about dozen predefined settings and maximize each view, but can not arbitrarily arrange all the views. Although this might seem as a drawback, the domain experts from various fields were very satisfied with this solution.

Besides various visualization views, we have implemented data view as well. It is a table displaying original numbers. It has been often stressed out that the users need details at the end which is supported by our findings. The users start with numerical data and, although they use visualization for analysis and understanding, they need the exact numbers during the whole visual analysis process.

The ability to save and load the current status of the analysis (complete data, settings, view configuration, brushes, etc.) is also necessary. We have often exchanged project files among the team members and one could easily continue where someone else stopped. This would not be possible using the screen-shots only. Of course, image export for reporting is welcomed, but nearly not as much as project saving.

We allow the user to configure all the views and set various parameters, e.g. point size or color. This feature was often used by the experienced users. Usable defaults and automatic selection based on data amount help here a lot. An average user wants to start as soon as possible. If the default settings are inadequate, there is a great danger of losing a user, or at least getting an unsatisfied one.

Brushing is one of the essential features of every coordinated and multiple view system. ComVis supports two types of brushes, the single brush and composite brush.

The composite brush allows the user to combine several brushes using Boolean operations in an iterative way. The user selects a Boolean operation to be used with the next brush. The operation is then applied to the current selection and the new brush. In this way the user can easily broaden (OR) or narrow (AND, SUB) the selection. The ease of use and intuitiveness of such an iterative compos-

ite brush is appreciated by all users. The default brushing mode is the single brush mode. Once the user gets used to the single brush mode, it is easier to switch to the composite brush mode that is sometimes too complicated for the novice users. There is also a possibility to exactly specify brush values but that seems to be a feature used only by the most experienced users, although not often. Figure 3 shows various views and one user selection.

An important aspect of every tool that can handle data is interoperability with other programs. While visual data analysis tools like ComVis are dedicated to visualizing multidimensional datasets, there is often the need to edit datasets, perform various calculations and conversions. Such operations are often carried out in spreadsheet applications and therefore data have to be imported from and exported to common formats.

We use the comma separated values (CSV), because it can be opened and saved by many applications. Besides data exchange through the CSV files, ComVis supports copy and paste with common spreadsheets. This is an essential feature for the experienced spreadsheet users used to simply select data, copy it and paste it somewhere else. This is also true in the opposite direction, copying data from a spreadsheet into ComVis. If the users identify an interesting data selection using ComVis, they can just copy the selection and they paste it to a spreadsheet for further processing. Many of the users would certainly not use ComVis without this feature.

5 Examples

We have implemented ComVis, based on the design principles described in Section 4. There are currently about a dozen available view types. Besides conventional views such as histogram, 2D or 3D scatter plot, and parallel coordinates, ComVis supports more advanced views. We have implemented the curve view [7], the segmented curve view [6], and the color lines view [8] that support families of function graphs.

We have also included support for set-typed data through extensions of conventional views and the new view called set-o-gram. Special attention paid to various data types (families of curves, set-typed data, families of surfaces) proved to be very useful. We have analyzed a wide variety of problems together with domain experts from various fields ranging from ECG curves of pregnant sheep, or transplanted heart, up to design of common rail injection system in modern diesel engines. Due to the limited space we cannot describe all these cases in detail. However we would like to describe one additional feature, the concept of families of function graphs. This idea was extensively used in almost all advanced analysis we made. It is not a straightforward one, but after some time all domain experts were impressed with possibilities.

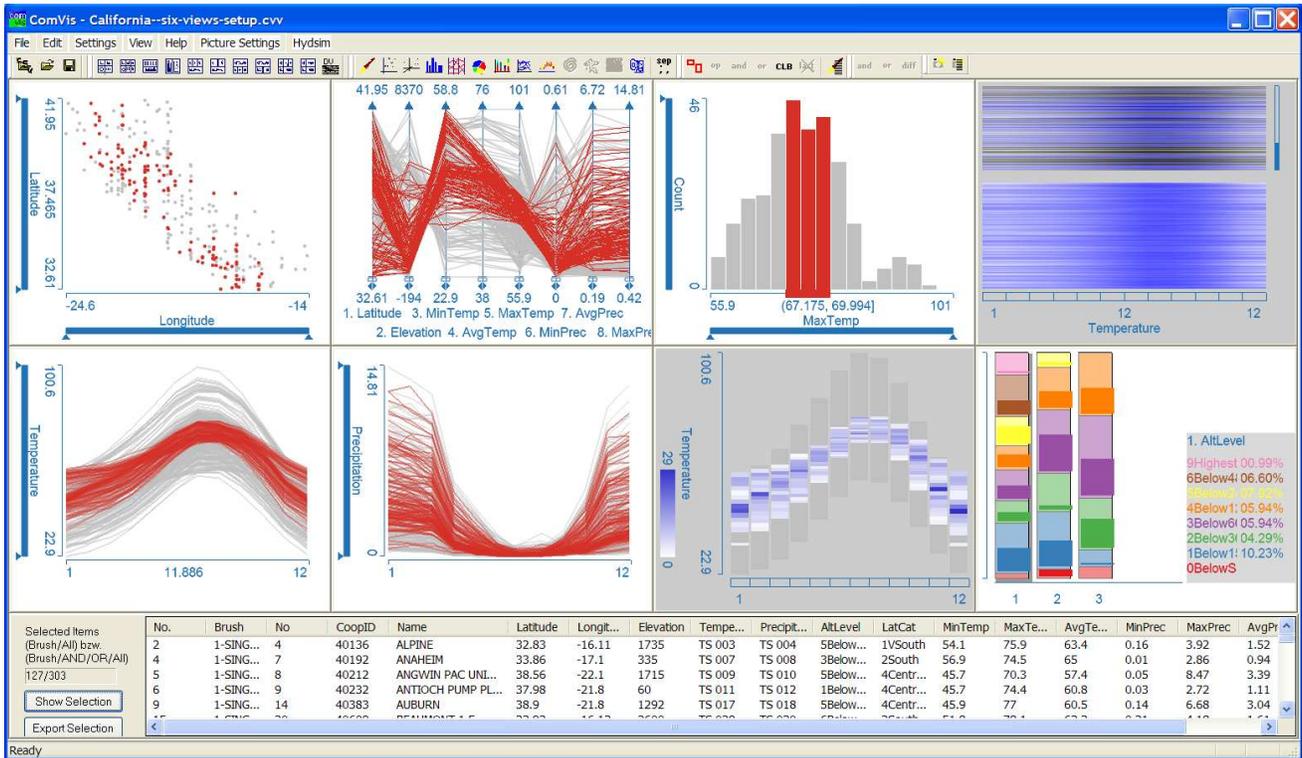


Figure 3: An example ComVis configuration. Eight views, scatter plot, parallel coordinates, histogram, color lines, curves, segmented curve view and bar charts, are used to analyze meteorology data. The user selected three histogram bins using the mouse, and all views highlighted corresponding items. The context, not selected items, are depicted in gray.

The majority of visualization tools use multidimensional data where each dimension can be a scalar value (numeric, categorical, ordinal, etc.). The time series data is usually considered as a multidimensional space where time is one scalar dimension and some time dependant attribute another. However, there are many practical situations where each function graph (not necessary a time series) has some additional attributes.

Imagine temperature curves for meteorological stations in a country. Each station has a temperature curve assigned and numerous scalar attributes, such as longitude, latitude, elevation, maximum temperature, station ID, to name just a few. All stations put together form a data set, and one attribute, the temperature curve, can be observed as a family of function graphs. Still, each curve belongs to a record with many other parameters. When the user selects any parameter or the curve itself, the user wants to see the corresponding records with all scalar attributes and function graphs selected. ComVis supports such a data model, and the curve view, segmented curve view and color lines view can be used to depict families of function graphs.

The common rail injection system follows that data model. System designers, engineers, create a model, set

up parameters and run the simulation. Control parameters are usually (in our case) scalar values and represent various control parameters of injection system. Output parameters are scalars and families of function graphs.

There are many (often conflicting) goals of Diesel engine design including high power, good fuel efficiency, meeting emission regulations, low noise levels and drivability. The fuel injection system is the key Diesel engine component to achieve those goals. The common rail injection system can be controlled in a very flexible way.

The common rail injection system have several attractive characteristics: injection pressure and quantity can be controlled with a high degree of flexibility, multiple fuel injections are possible within one injection cycle and the time and duration of the injections can be controlled precisely by the engine control unit based on the engine speed and load. These characteristics are key factors in meeting current and future very stringent emission regulations. In our case study we use the simulation results for the common rail Diesel fuel injection system.

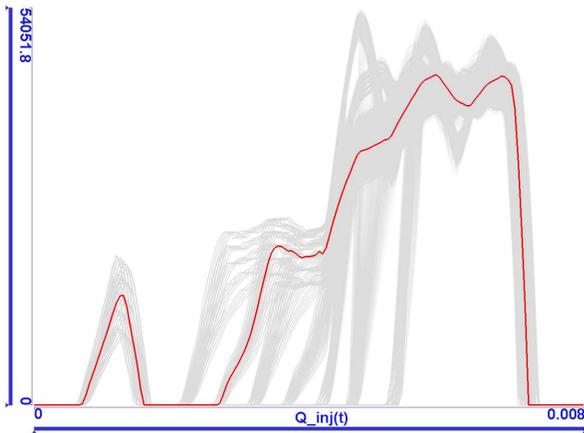


Figure 4: A typical shape of the fuel injection rate graph is highlighted in red. There is a short pilot injection first, followed by the main injection. The graphs resulting from other combinations of control parameters are gray.

The injection shape depends mainly on three factors: the nozzle geometry, injection pressure and timings for valve opening and closing procedures. The injection pressure is controlled by the injection pressure modulation device. The characteristics of the pressure on the injector's inlet are described by three parameters. The injector valve actuator that controls the injection timing is described by its opening/closing times and velocities. Consequently, we have five different control parameters. We varied each parameter in several steps and run simulation for every possible combinations of parameter values. The total number of variations of the independent variables (different sets of simulation boundary conditions) was 4375.

For each combination of the independent variables the simulator computes three sets of time-dependent results: $Q_{inj}(t)$: injection rate, $P_{inj}(t)$: injection pressure, and $A_n(t)$: needle lift. In other words, there are three families of function graphs in this data set. Furthermore, some time-dependent results are computed.

The typical shape of the injection curve is shown in Figure 4. There are usually one or two small peaks called *pilot injections* during the first quarter of the injection procedure in order to reduce combustion noise and NO_x emission in combination with the main injection. As Figure 4 shows, there was one pilot injection in our case.

Arbitrary shaped injection rate graphs cannot be produced in simulation because of the physical requirements of the combustion in the engine. The engine will not run properly if the injection rate function graph does not follow shapes similar to the ones in Figure 4. The goal is to find combinations of simulation parameters that control the volume of the pilot injection and produce the desired shape of

the main injection. In this example we have gained a valuable insight into the fuel injection simulation data set and thereby into the fuel injection process, too.

We also showed the usefulness of the tool in many other cases as well. Timing Chain Drive analysis was illustrated in [6], interactive visual analysis of traffic sensing data was described in [7], meteorology data and a new view were introduced in [8], to name just a few.

Conclusions

We have identified many interesting aspects during development of our tool. We hope they can help developers of similar tools. The developed tool, ComVis, facilitates fast development, prototyping and testing of new ideas for visualization techniques. We have tested it in various contexts, with domain experts from many fields. The ease of use, useful default settings, easy data exchange with spread sheets were the key aspects which helped novice users to quickly start using the tool and then detect all advanced features. Extensive collaboration with domain experts from various application domains yielded important feed-back and experience that was used to further refine ComVis. Future works will focus on multi-user and collaborative features, as well as on the support for large scale displays. Since ComVis lends itself well to experiments, an added bonus is that it can be used as an educational tool where students can use different visualization techniques and add their own techniques. ComVis was used in our lectures and for various student projects in Vienna (Austria), Zagreb (Croatia), Bergen (Norway), and Blacksburg (USA).

Acknowledgements

There are many people who supported development of ComVis in some way. In addition to all the paper coauthors we would like to specially thank Georg Stonawski from VRVis. He made the whole development possible and supported the idea from the very beginning.

References

- [1] C. Ahlberg and E. Wistrand. Ivey: an information visualization and exploration environment. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 66–73. IEEE Computer Society, 1995.
- [2] Jean-Daniel Fekete. The infovis toolkit. In *INFOVIS '04: Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04)*, pages 167–174, Washington, DC, USA, 2004. IEEE Computer Society.
- [3] Inxight software inc., 2008, <http://www.inxight.com/>.

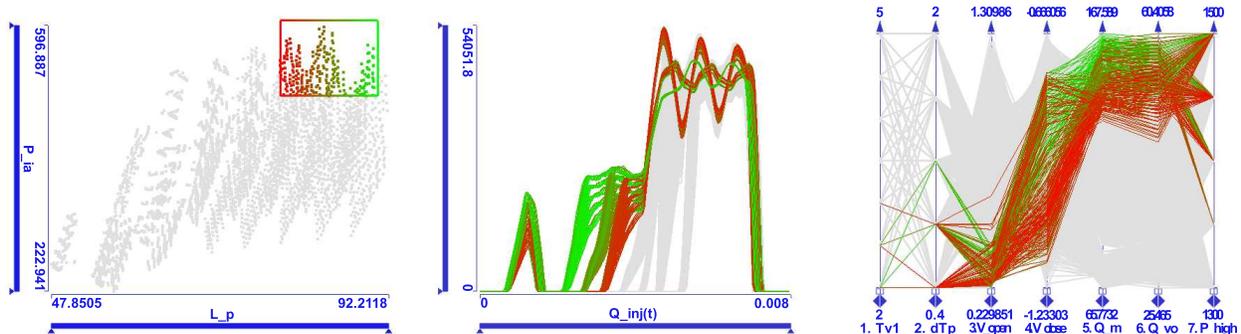


Figure 5: We investigate the conditions when fuel is injected deep in the combustion chamber and with high power. The corresponding items are brushed in the scatterplot diagram. The linked injection rate function graphs show that this requires boot-shaped main injections. The desired needle opening and closing velocities are highlighted in the parallel coordinate view.

- [4] Daniel A. Keim and Hans-Peter Kriegel. Visdb: A system for visualizing large databases. In *SIGMOD Conference*, page 482, 1995.
- [5] Alfred Kobsa. An empirical comparison of three commercial information visualization systems. In *INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, page 123, Washington, DC, USA, 2001. IEEE Computer Society.
- [6] Zoltan Konyha, Krešimir Matković, Denis Gračanin, Mario Djuras, and Josip Jurićand Helwig Hauser. Interactive visual analysis of a timing chain drive using segmented curve view and other coordinated views. In Gennady Andrienko, Jonathan C. Roberts, and Chris Weaver, editors, *Proceedings of the 5th International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV2007)*. IEEE Computer Society Press, July 2007.
- [7] Zoltán Konyha, Kresimir Matković, Denis Gračanin, Marko Jelović, and Helwig Hauser. Interactive visual analysis of families of function graphs. *IEEE Transaction on Visualization and Computer Graphics*, 12(6):1373–1385, November-December 2006.
- [8] Krešimir Matković, Denis Gračanin, Zoltan Konyha, and Helwig Hauser. Color lines view: An approach to visualization of families of function graphs. In *Proceedings of the 11th International Conference Information Visualisation (IV2007)*. IEEE Computer Society Press, July 2007.
- [9] National visualization and analytics center. <http://nvac.pnl.gov/>.
- [10] Jonathan C. Roberts. State of the Art: Coordinated & Multiple Views in Exploratory Visualization. In Gennady Andrienko, Jonathan C. Roberts, and Chris Weaver, editors, *Proceedings of the 5th International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV2007)*. IEEE Computer Society Press, July 2007.
- [11] Deborah F. Swayne, Duncan Temple Lang, Andreas Buja, and Dianne Cook. GGobi: evolving from XGobi into an extensible framework for interactive data visualization. *Computational Statistics & Data Analysis*, 43:423–444, 2003.
- [12] Tableau tool. <http://www.tableausoftware.com/>.
- [13] Martin Theus. Interactive data visualization using mondrian. *Journal of Statistical Software*, 7(11), 11 2007.
- [14] Matthew O. Ward. Xmdvtool: integrating multiple methods for visualizing multivariate data. In *VIS '94: Proceedings of the conference on Visualization '94*, pages 326–333, Los Alamitos, CA, USA, 1994. IEEE Computer Society Press.
- [15] Jing Yang. *A General Framework for Multi-Resolution Visualization*. PhD thesis, Worcester Polytechnic Institute, 2005.