International Symposium on Visual Analytics Science and Technology (2010), pp. 1–7
J. Kohlhammer and D. Keim (Editors)

Interactive Visual Analysis of Families of Surfaces: An Application to Car Race and Car Setup

K. Matković[1], D. Gračanin, [2] R. Splechtna, [1] and H. Hauser [3]

[1] {Matkovic|Splechtna}@VRVis.at, VRVis Resarch Center in Vienna, Austria
[2]gracanin@vt.edu, Virginia Tech, Blacksburg, VA, USA
[3] Helwig.Hauser@UiB.no, Univerisity of Bergen, Norway

Abstract

Modern simulations often produce time series, or even functions of two variables as outputs for single attributes. Such complex data require carefully chosen and designed analysis procedures and the corresponding data model. The use of previously developed curve and surface views provides strong support for visual exploration and analysis of complex data. In this paper we describe how interactive visual analysis can support users in getting insight into complex data. The case study, based on TORCS 3D racing cars simulator, illustrates our approach and its successful application to a real world problem. The analysis of the car parameters and driving performances during races provides an insight and explanation for race results. That insight is then used to fine-tune car parameters to achieve better driving performance.

1. Introduction and Related Work

Simulation is a standard part of almost all engineering and scientific disciplines. It helps domain experts and scientists to better understand various phenomena, especially when they run simulations for various combinations of input parameters. The data generated by modern simulation is large and can be much more complex than a simple table of columns and rows. Interactive visual analysis can be used then to explore and analyze the data, help gain an insight, and better understand the data. It can also help in identifying outliers, finding unusual behavior which can be an error in the model or a result of examined phenomena.

A single simulation run determined with a set of control parameters can generate various outputs. Some of them are scalars, while others are series of data, i.e. functions of one or more variables represented as curves (one variable) or surfaces (two variables). We call all curves (surfaces) originating from different simulation runs of the same attribute a family of curves (surfaces).

In our previous work [KMG*06] we described the curve view and used it in a tool for interactive visual analysis to study data series characteristics of engineering data. A family of curves (one-parameter data series) is presented as an aggregated 2D plot view (the curve view) of all the curves. That approach can be applied to various problems using a more general data model and related visual analytics techniques. The analysis of data that contains families of surfaces was introduced in [MGKH09].

In this paper we illustrate an analysis of complex data originating from multiple (over 9000) simulation runs of driving a car on a race track. The TORCS [TOR] open car simulator was used. We describe how coordinated multiple views and advanced interaction techniques can support the user's discovery process in understanding car behavior. Although it is easy to find a setup which resulted in the fastest overall time, we show how an interactive process can support the understanding of the car performance. The user understands a complex system much better, and can make decisions easier when the optimum case is not possible. We have applied technology introduced in [MGKH09] in order to understand the car behavior and to tune the car for a race.

The body of literature about visualization of large, high-dimensional and time-dependent data sets is very large and the field is still an area of active research [Tuf01]. Exploration of large data sets [Kei01] is based on presenting the data in a visual form that would allow analysts to interact with it. Data visualization techniques, when suited for the given data set, reduce the cognitive load while performing analysis tasks. A visualization technique should have limited visual overlap, fast learning, and good recall. Furthermore, good integration with traditional techniques (including simulation) improves the data exploration process.

Time-dependent data is a very important category of data sets. Brushing the time axis to display details of the selected time frame is one very common and useful interaction technique used with static representations. Müller and Schumann provide an overview (taxonomy) of the visualization methods for time-dependent data [MS03] and discuss general aspects of time-dependent data. Aigner et al. [AMM*08] provide an overview of visual methods for analyzing time-oriented data and discuss general aspects of time-dependent data. The time factor requires a special treatment during visual exploration. They distinguish between two cases based on the time dependence of the visual representations, time-dependent (dynamic) and time-independent (static) representation.

Time–dependent data visualization methods usually deal with each dimension as a scalar value (e.g., numeric, categorical, nominal) and handle time-dependent data as an isolated case or aggregate the data into scalar values. Problems that exhibit a complex internal data structure can be tackled by interactive visual analysis of family of curves [KMG*06]. That approach provides analysis procedures and practical aspects of the interactive visual analysis specific to this type of data. Multiple linked views support iterative visual analysis by providing means to create complex, composite brushes that span multiple views and that are constructed using different combination schemes.

Interaction techniques allow the user to better understand the data set because of the ability to interact with that set. One of the most common interaction techniques is Focus+Context (F+C) visualization [KHG03]. When the amount of data is too large to display, the user should be able to focus on some specific data while keeping track of the entire data set (context).

The visual analytics framework supports, among others, the search for an insight into available data by using interactive visual analysis, visualization techniques and visualization metaphors. In other words, "*visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces*" [TC05]. We can build on top of the existing visualization techniques and create a synergy by combining and linking different views of the same data and thus supporting analytical reasoning and the discovery process.

2. Data Model

Data analysis approaches usually use a relatively simple multi-dimensional data model [Sam06] (simple with respect to the separate data dimensions). Each data dimension has values that are either literals or numbers. However, very often we deal with data sets that do not fit in this simple model.

For example, let us consider a data set that consists of the area averaged temperature time series for several countries. In the simplest case we have one measurement for each country so we can view this data set as a set of three data points and three dimensions (country, time, and temperature). A data point is a three-tuple containing values for each of the three dimensions. As the measurements accumulate, we have more and more data points (three-tuples). We can aggregate the data points based on the same value for country to get temperature data series for each country.

We can refine our data model so that for each value for country, there is exactly one data point. The data point is now a two-tuple and has two dimensions, country and temperature. The values of the temperature data series dimension are now time/temperature pairs or two-tuples. The data model now has a two-level structure. We can "split" the time dimension into two new dimensions, year and month. The data model still has a two-level structure but the values of the temperature data series dimension are now sets of three-tuples, $(year, month, temperature)$. While these model refinements are rather trivial in this simple example, they illustrate the rationale for a two-level data model that allow us to aggregate data points based on the values in a selected dimension(s) and restructure the data set to have a relatively small number of data points while preserving the information content.

More formally, in our approach we are considering a two-level data set that consists of data points (tuple values) of n dimensions (Figure 1). The data set is $D = \{\mathbf{x}^1, \ldots, \mathbf{x}^i, \ldots, \mathbf{x}^d\}$, d is the size of the data set (the number of data points) and each data point $\mathbf{x}^i = (x^i_1, \ldots, x^i_j, \ldots, x^i_n)$ is a collection of attributes, one for each dimension. A tuple attribute x^i_j can be categorical, numerical, or a data series.

For each tuple $\mathbf{x}^i$ and each data series attribute x^i_j in a data tuple, we have a separate set of "sub-tuples" with its own cardinality and dimension. The set of sub-tuples is defined as $D_{ij} = \{\mathbf{x}^1_{ij}, \ldots, \mathbf{x}^k_{ij}, \ldots, \mathbf{x}^{d_{ij}}_{ij}\}$, d_{ij} is the number of sub-tuples. A sub-tuple in D_{ij} has a form $(x_{ij,1}, \ldots, x_{ij,n_{ij}})$, n_{ij} is a sub-tuple size and each sub-tuple attribute is either categorical or numerical. The sub-tuple $\mathbf{x}^k_{ij}$ is then $(x^k_{ij,1}, \ldots, x^k_{ij,n_{ij}})$. One can consider a data set with more than two levels of tuples, however we focus on two-level data sets.

Our discussion is limited to three-tuples and less ($n_{ij} \leq 3$), i.e. a data series can be a sequence of numbers, a sequence of pairs of numbers or a sequence of three-tuples. In case of the sequence of pairs of numbers (x_1, x_2), one dimension is used as an independent variable and other as the dependent variable. We can use a function graph (curve) to represent data series. In case of the sequence of three-tuples (x_1, x_2, x_3), one dimension is used for a dependent variable and the remaining two dimensions for independent variables, i.e. a function of two variables that can be visualized as a surface.

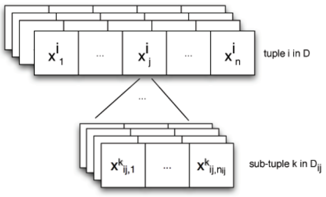

Figure 1: *Generic data tuple. Each item can be scalar, but can also be a mapping. A data set with an attribute containing several sub-tuples contains a family or families of curves or surfaces.*

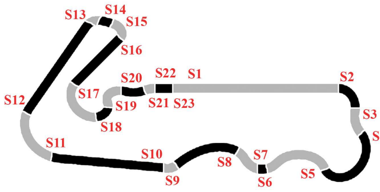

Figure 2: *Track layout and segments location as used in the simulation.*

3. Case Studies

We use TORCS, The Open Racing Car Simulator [TOR], to generate a complex data set used in the case study. TORCS provides over 50 different cars, 20 tracks 50 opponents to race against. The simulation features a simple damage model, collisions, tire and wheel properties (springs, dampers, stiffness, ...), aerodynamics (ground effect, spoilers, ...) and much more.

The idea is to vary a limited set of parameters and to run the simulation with the same track and driver for 20 laps for each car setting. Afterwards we analyze the data in order to get insight into the car behavior and to be able to tune the car for the race. For the selected driver (*bt1*) and the race track (*wheel–1*, Figure 2) a 20 laps practice run was simulated for various combinations of attributes. We did not change the driver or track in order to focus on the car setup only.

Measurements were gathered (simulated) at 23 control points corresponding to 23 segments that constitute the race track (Figure 2). The following data attributes were used:

- **Wing Angle**: 8–18 degrees (six steps).
- **Brake Ratio**: 0.41–0.50 (five steps).
- **Max. Brake Pressure**: 6,500–19,000 (six steps).
- **Gear Sets**: Six different gear box sets (six steps).
- **Front Spring**: 1,000–2,000 (three steps).
- **Rear Spring**: 1,000–2,000 (three steps).

We have run all possible combinations of parameters which resulted in ($6 \times 5 \times 6 \times 6 \times 3 \times 3 = 9720$) simulation runs. The wing angle corresponds to the rear wing (the only one our car had. The larger the angle is, the larger the down-force is. More down-force is useful for better car handling (sudden moves are allowed) but reduces the top speed. We have also used six different gear boxes. Each gearbox had different ratios for six gears. Front spring and rear spring are self explanatory. Break pressure is the maximum possible pressure applied when the break is fully pressed, and break ratio describes the ratio of pressures applied to front and rear wheels. We have recorded car data (during the runs) for 23 spots on the track. The spots were chosen always at the beginning of a curve or of a straight segment. In addition there was a measuring point at the end. Minimum and maximum values are:

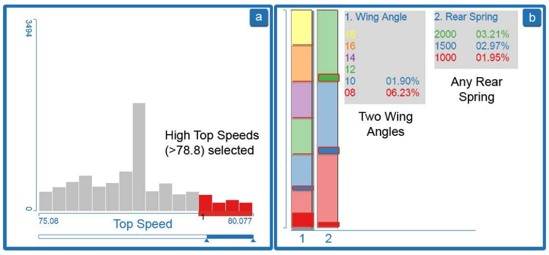

Figure 3: *Initial hypothesis confirmation: high top speeds are selected in the histogram view and corresponding parameters are depicted in a stacked bars view. Only two wing angles allowed the maximum top speeds, but any rear spring setting is possible. Larger wing angle increase down-force (and car handling will be easier), but top speed is reduced.*

- **End Time**: 1,505–1,978 (seconds).
- **Best lap Time**: 74.516–94.184 (seconds).
- **Top Speed**: 57.037–80.077 (miles per hour).
- **Damage**: 0–7,352 (number).

For each of 23 control points we measured speed, RPM, current time, gear, and top speed. We also measured best lap time, damage and end time. Damage is a value computed by TORCS showing the state of the car. If damage exceeds a certain threshold, the simulation run stops.

There are 9720 simulation runs, each determined with a set of independent variables (car settings) and each providing various measuring data of different kind. There are scalar numeric values, such as total time, but there are also more complex outputs such as speed dependent on lap and segment. One possibility would be to consider lap and segment additional independent variables and to have then $23 \times 20 = 460$ speed values per run. We have organized the simulation data as collection of families of surfaces dependent on the segment and the lap it is gathered from. This data set follows the data model described in Section 2. Figure 3h shows one surface from a family as a 3D surface that represents speed as a function of lap and segment. Note the exceptionally low speed at segment 5, lap 1 which needs further investigation.

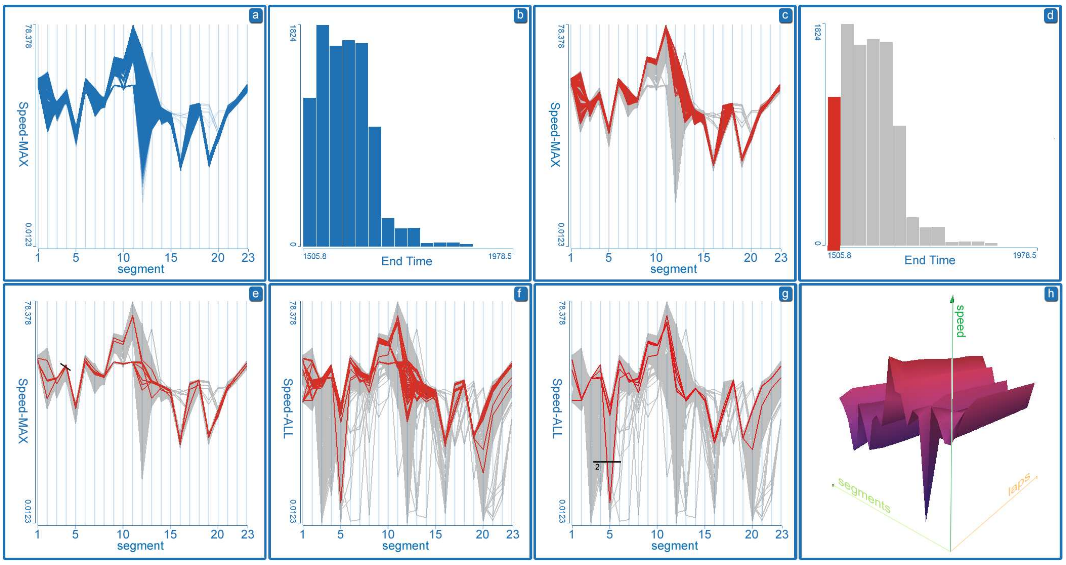

Figure 4: *Interactive visual analysis of a family of surfaces using the curve view and other multiple linked views.*

To summarize, our data space has seven control parameters, four regular (numerical) output parameters and five output parameters that are of form $f(lap, segment)$. There are 9720 simulation runs or data points $\mathbf{x}^i$ and five families of surfaces, where each family has 9720 surfaces.

3.1. Discovery Process

The multiple linked views setup supporting analysis of families of surfaces [MGKH09] was used in the discovery process. Conventional views were used to detect some simpler dependencies, and more complex analysis of families of surfaces is supported by curve view, 2D-surface view, and 3D-surface view. We have described the expected behavior between the wing angle and top speed parameters. We can confirm the hypotheses with two simple views.

We will depict top speed using a histogram (Figure 3a), and depict wing angle and rear spring strength using a stacked bar display. If we now select all the cars with the top speed greater than 78.8 using a simple brush in the histogram, we can see that only cars with wing angle 8 and 10 fulfill this criterion (stacked bar on the left). Note also that there are cars having wing angle 8 and 10 which do not fulfill the criterion (parts of the bars which are not highlighted). Actually only 6.23% of the total car setting variations fulfill it for wing angle 8, and only 1.90% for wing 10 (Figure 3b). We can also see that there are more cars with stronger rear spring (stacked bar on the right) in the selection.

3.2. Families of Surfaces

If we want to tune the car and better understand the car's behavior, we need some additional analysis techniques. We will use families of surfaces to analyze the data. Recall that we have 9720 simulation runs (surfaces) in each family. We have five families in our data set, one for each output parameter.

As we have a family of surfaces it is clear that it is impossible to depict all 9720 surfaces simultaneously in a 3D view [War04]. We use a 2D curve view which depicts a family of curves in order to analysis surfaces. The main idea is to depict various projections instead of surfaces. Projections orthogonal to an axis are supported and there are projections along x and along y axis. A surface from a family can be depicted using only one curve or several curves. If one curve is used it can be maximum, minimum, median, etc., curve. The surfaces we are dealing with are discrete, there is a value for each segment and lap. If we select the projection along laps, e.g, then the maximum curve contains the maximum values of the attribute for each lap. If we want a projection along segments then we get a curve across segments which shows the maximum value from all laps for a particular segment. It is possible to depict all curves as well. After the data set is drilled down to just a few surfaces from a family, a 3D surface view or a 2D surface view (a height-map) can be used.

3.3. Case Study: Segment Speed and Overall Time

We examine the current speed attribute first. We assume that cars which were the fastest overall are not the fastest in each segment. It may be better to slow down at certain turns in order to score a better lap time. We are using many views in analysis, and due to the limited space here we will show only the most interesting ones. We start with selecting the current speed attribute and depict it so that we keep the segments axis and display maximum across the laps for each segment. There is one curve for each surface, and it is created so that we take the maximum speed for each segment across all laps.

Figure 4a shows this family of surfaces. The speed at various segments varies a lot. Note that the speed span at various points varies a lot, too. At some segments all cars had similar maximum speed, and on some segments the range is significantly larger. Note also few (light blue) outliers in segments 13, 16, 19, and 20. At the same time we will use a histogram to depict the total time needed to finish 20 laps (Figure 4b). If we now select fast cars, (first bin in the histogram representing top 1/16 of time) we can see that the fast cars were not the fastest in all segments (Figures 4c and 4d).

We see that there are faster cars in segment 4 and there are cars which were significantly faster in segments 16, 19, and 20 (outliers). Those cars, although faster in some segments, did not score the best total times. This means that if we want to have a good overall result we have to drive carefully at some segments. Before we proceed, let us remind you that we are observing maximum speed at each segment. This means that it is possible that one car was very fast in segment 16, for example, only in one lap. This car could have been very slow in all other laps, but it would have a high maximum here and the maximum curve would be very high. There were 20 laps for each car.

Let us examine cars that have a high maximum speed in segment 4. We simply brush the curve view (draw a line and all curves intersecting the line will be selected). Figure 4e shows the result. We can clearly see the two clusters in the middle section of the graph.

Since we are using the maximum cut and we have an unusual case here (fast cars in segment 4 that are slow overall) we display all surface curves, not only the maximum. Figure 4f shows all curves. We can see that the surface with high maximum had a very low minimum in segment 5. Note that we still do not know how often that happened. It could be an exception, or a rule. Let us refine the selection by selecting only surface(s) which have a very low peak in segment 5 (Figure 4g). We see that this is only one surface (the tool displays the number of selected points) but we still do not see if this was an exception or a rule. We will use the 3D surface view to examine this case (Figure 4h).

It is now clear that the high maximum values in segment 4 were from the first lap. The driver entered segment 4 too fast resulting in a very low speed in segment 5 (hard breaking or getting off the track). The driver learned the lesson and adjusted segment 4 speed in the subsequent laps. This resulted in the lower overall time, but the driver finished the race.

We can detect something else from the previous images as well. We have seen two clusters in our selection for segments 9 to 13. Interestingly, when we have selected our previous case, only one cluster is left. This means that cars that did not have low speed in segment 5 did have a very low speed in segments 9 to 13. Interestingly, segments 9 to 13 are places where most of the cars achieved very high speed. We can see from the track layout (Figure 2) that this is is a section consisting of two long straight segments and one segment with not so sharp turn. An ideal section for high speed! So what prevented our cars from driving faster here? If we examine the input parameter space we can see that all of them had gear box 5. The gear box 5 settings do not allow high top speed. The cars were not that bad in segments with more turns, but lost the race in the straight segments.

3.4. Case Study: Good Lap Time, Low Damage, Learning Curve

Figure 5 provides another example for interactive analysis. Parallel coordinates are used to depict scalar output dimensions (Figure 5). We have brushed best lap time and low damage as cars of interest. We would like to have a car which can make a good lap time and while keeping the damage low. Interestingly, the high top speed setups are not included now.

Figure 5b shows a parameter distribution for such a case. We can see that we need to have wing angle 8, 10, or 12 (first column) and not a soft rear spring (column 6). Gearbox 5 (column 4) is also not allowed.

Top speed represents the current top speed in each segment and lap. It is cumulative and it is valid for the whole race. That means if a driver achieves a top speed in segment 6, lap 2 (for example) all further values of top speed will be the same. This is the top speed achieved up to now.

We have zoomed to the upper section of top speeds in Figure 5c. We can see two kinds of patterns. There are cars with horizontal maximum top speed over the laps. These cars achieved top speed at the beginning of the practice. On the other hand, cars with the increasing top speed were going faster and faster, and achieved top speed at the end. Note that their top speed exceeded the top speed of the first group.

Let us exclude the horizontal lines (drivers who do not learn, i.e. who achieved top speed very early (Figure 5d). We have also excluded wing 8 cases which is not visible due to space limitations. As stated above, wing 8 cars are very fast (low down-force) but can not stand sudden moves. Sudden moves are common in the race (and not common in the practice where the driver is alone on the track) so we will consider higher wing angles only. Note the gear setup distribution for the selection. Only gear boxes 0, 3, and 4 allow such a combination of output parameters (best lap time, low damage, increasing top speed over laps) if wing 8 is not allowed.

Note also two obvious clusters in Figure 5d. Let us select the bottom cluster now (Figure 5e). All three gear boxes are allowed. We will select gear box 0 (Figure 5f), gearbox 3 (Figure 5g), and gearbox 4 (Figure 5h) now (selection itself not depicted). We can see that gear box 0 and 3 have similar

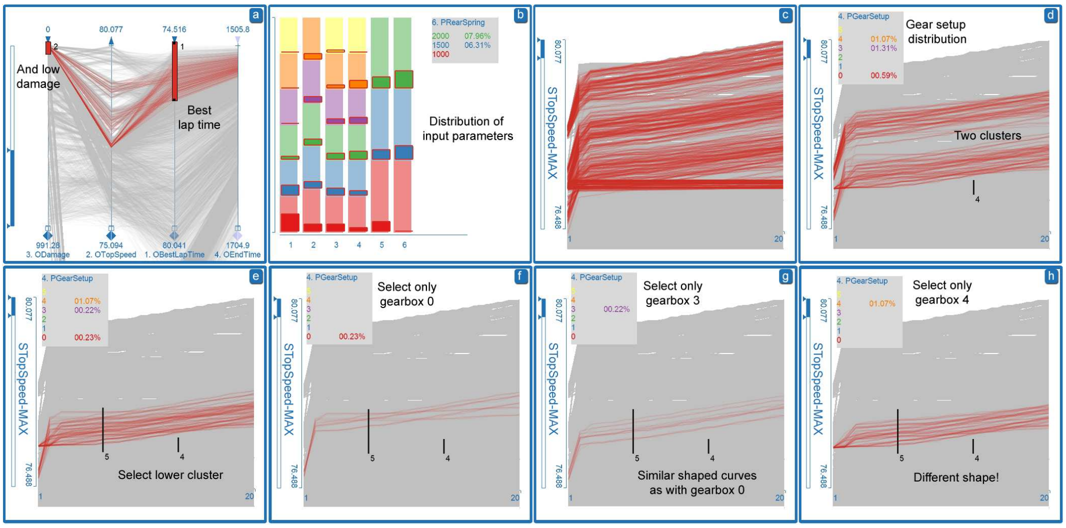

Figure 5: *Interactive visual analysis: finding a car with a good lap time and low damage.*

shapes, gear box 0 has higher values. Gear box 4 has slightly lower end values and a different shape.

The driver's learning curve was different. Note that we had the same driver and track in all cases. The percentages in top right and bottom left indicate that percentage for gear box 4 is the same (1.07%). That means that there will be no gear box 4 cases in the upper cluster! It is slower, has a different learning curve and can not have values from the upper group. If you carefully observe Figure 5d, you will notice that there are no curves with gear box 4 shape.

4. Conclusions

The analysis of relationships within a complex data set is a common task in many application domains. A novel combination of linked views, advanced brushing, curve views and derived surface views represents a valuable tool for interactive visual analysis and analysis of data sets that include multiple families of function graphs (curves/surfaces). The case study demonstrates how surface views provide an insight into the analyzed data sets that would be impossible otherwise. The cuts and related features proved to be very useful in gathering insight about the problem at hand.

Acknowledgments

The case study data was generated using TORCS, a 3D racing cars simulator [TOR]. Part of this work was done in the scope of the VSOE VCV program at the VRVis Research Center in Vienna and at the Center for HCI at Virginia Tech.

References

[AMM*08] AIGNER W., MIKSCH S., MÜLLER W., SCHUMANN H., TOMINSKI C.: Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics 14*, 1 (2008), 47–60. 2

[Kei01] KEIM D. A.: Visual exploration of large data sets. *Communications of the ACM 44*, 8 (Aug. 2001), 38–44. 1

[KHG03] KOSARA R., HAUSER H., GRESH D. L.: An interaction view on information visualization. In *State-of-the-Art Proceedings of EUROGRAPHICS 2003 (EG 2003)* (Sept.1–6 2003), pp. 123–137. 2

[KMG*06] KONYHA Z., MATKOVIĆ K., GRAČANIN D., JELOVIĆ M., HAUSER H.: Interactive visual analysis of families of function graphs. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2006), 1373–1385. 1, 2

[MGKH09] MATKOVIC K., GRACANIN D., KLARIN B., HAUSER H.: Interactive visual analysis of complex scientific data as families of data surfaces. *IEEE Transactions on Visualization and Computer Graphics 15* (2009), 1351–1358. 1, 4

[MS03] MÜLLER W., SCHUMANN H.: Visualization for modeling and simulation: Visualization methods for time-dependent data - an overview. In *WSC '03: Proceedings of the 35th Conference on Winter Simulation* (2003), vol. 1, Winter Simulation Conference, pp. 737–745. 2

[Sam06] SAMET H.: *Foundations of Multidimensional and Metric Data Structures*. Elsevier, Amsterdam, 2006. 2

[TC05] THOMAS J. J., COOK K. A. (Eds.): *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Center, 2005. 2

[TOR] TORCS: The open racing car simulator (TORCS). http://torcs.sourceforge.net/ [last accessed on February 27, 2010]. 1, 3, 6

[Tuf01] TUFTE E. R.: *The Visual Display of Quantitive Information*, 2nd ed. Graphics Press, Cheshire, Connecticut, 2001. 1

[War04] WARE C.: *Information Visualization: Perception for Design*, second ed. Morgan Kaufmann Publishers, 2004. 4