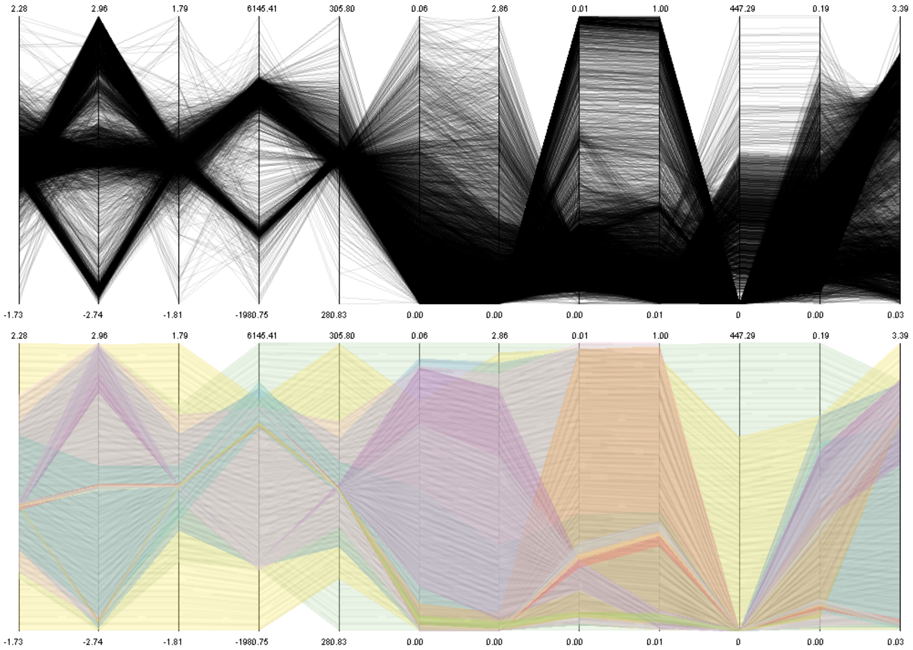
Visual Abstraction for Information Visualization of Large Data

Diploma Thesis by Matej Novotný
Faculty Of Mathematics, Physics and Informatics, Comenius University Bratislava



The real journey of discovery is not made by seeking new lands but by opening new eyes. - Marcel Proust

Acknowledgments

However this thesis is presented by me, it would never be finished and would never reached its current quality without a help from certain people. I would therefore like to dedicate the first page in my thesis to those that helped me the most.

First of all I would like to thank Robert Kosara and Helwig Hauser from VRVis Research Center in Vienna, who are the two most important people behind this project. Sharing their knowledge, advices and experience with me and also both their support and critique are the vital and invaluable contributions to the work presented in this thesis. I also thank the rest of the VRVis staff and namely Martin Gasser, Helmut Doleisch, Markus Hadwiger. Special personal acknowledgments go to Christoph Berger.

I would also like to thank Andrej Ferko for giving me a perfect starting point in my cooperation with VRVis and Marek Zimanyi who was the official project leader for the Slovak part of the cooperation.

I sincerely thank my teachers for the education they gave me and the love for knowledge they taught me, my friends for answering my stupid questions and giving me their opinions on my work, my girlfriend Janka and my family for supporting me throughout the months of my work and tolerating my neglecting them.

I hereby declare, that the research and results presented in this thesis were conducted by myself using the mentioned literature and advices of my supervisors. This work has been done in the basic research on visualization at the VRVis Research Center, which is funded by the Austrian Kplus project.

Matej Novotný

Contents

1 Introduction **4**

2 Information Visualization **6**
2.1 Visual data exploration 9
2.2 Data of interest 9
2.3 Information visualization system and interaction 10
2.3.1 Interaction 10
2.3.2 Multiple viewports and linking 11
2.4 Visualization techniques for displaying information 11
2.4.1 Dimensional subsetting – scatterplots 12
2.4.2 Dimension reduction – MDS and SOM 12
2.4.3 Dimensional embedding 13
2.4.4 Pixel oriented techniques 13
2.4.5 Axis reconfiguration 14
2.5 Large data information visualization 14
2.5.1 Loss of speed 15
2.5.2 Occlusion 15
2.5.3 Aggregation 15
2.5.4 State of the art for large data information visualization 16
2.5.5 Hierarchical parallel coordinates 17

3 Clustering **19**
3.1 Groups, patterns, noise and outliers 19
3.2 Types of clustering 20
3.2.1 Top-down clustering 20
3.2.2 Bottom-up clustering 21
3.3 Clustering large and multivariate data 22
3.3.1 k-means and its modifications 22

4 Visual Abstraction **24**
4.1 Level of detail 25
4.2 Visual abstraction for information visualization 26
4.3 Visual exploration vs. automatic data mining 27

5 Large Data Visualization Using Parallel Coordinates 28
5.1 Parallel coordinates 29
5.2 2D binning 30
5.3 Multivariate abstraction in parallel coordinates 31
5.3.1 Transparency 31
5.3.2 Colors and textures 34
5.4 Data management 35
5.5 Incorporating the project into the Simvis software 37

6 Results 39
6.1 Speed improvement 39
6.2 Occlusion and aggregation problems solved 40

7 Conclusions and future work 44

Bibliography 45

Chapter 1

Introduction

For all we know, visual representation of knowledge and information has accompanied human civilization from its very beginning. It has been a useful tool for storing, communicating and exploring information thanks to the human visual system which is a broad communication channel capable of perceiving a rich volume of information in a single moment. With the development of computers and automatic data processing the need for an effective visualization grew even stronger, since the understanding of information inside the human mind happens in a different way than it does in the computer. Therefore the success of most computer-aided efforts strongly depends on the communication between the two counterparts – human and computer. The results of any simulation, scan or survey are easily hidden or lost when an improper visualization is used. Visual exploration becomes a valuable part of most scientific research conducted with the aid of computers.

Both human and computer are powerful processing units, but the information exchange is limited by the capabilities of a computer display. Plus the information is often multivariate, structured or hierarchical, therefore it is vital to use the communication resources wisely. Many sophisticated techniques to visualize even very complicated information were developed. Unfortunately their displays get easily cluttered by large data and in those cases the visual exploration becomes a slow process. Moreover the high number of visual elements in the display can cause an improper interpretation of the data because of occlusion and aggregation and can lead to wrong results.

The graphical and human processing resources could be saved if a simplified representation of the same data was presented to the viewer. Displaying less stimuli while preserving the intrinsic nature of the visualized data avoids the overall clutter and incomprehension regarding the display. This thesis introduces a new way to gain such a modification – visual abstraction. With the aid of unsupervised data mining – a domain of computer science focused on automatical exploration of interesting features inside the data –, an abstract representation of the original data is built in several levels of abstraction. The viewer is provided with a certain level of abstraction and is allowed to drill down to the detailed data or abstract to even higher levels.

This concept is illustrated on a popular infovis tool – the parallel coordinates and the abstract information is obtained using k-means clustering. The effects are observed on several large data sets and a comparison between a traditional visualization technique and visual abstraction is presented.

Chapter 2

Information Visualization

Sight is one of the most important and most sophisticated senses of a human. The ability to perceive a large amount of information in a fraction of a second makes the human visual system probably the broadest communication channel between our mind and our surroundings. The visual representation of information usually does not depend on any language and compared to text or sound, which are sequential media, in many cases it requires less time to perceive. Visualization has accompanied the human civilization for a long time and still remains one of the most powerful ways to store and communicate information. As the old saying goes, a picture is worth a thousand words.

Information comes in many forms. Science describes the real world in exact and measurable variables like distance, quantity, position or ratio. Usually a certain object or event is described by several variables of various types. The data of this origin is usually multivariate and can be effectively stored in a table or a database, but investigation of such a representation can be a hard task (Figure 2.4, left). The best way to communicate information from a computer to a human is probably through visualization. It is a process of transforming usually scientific information into a graphical representation that the human perceives more easily. Visualization as a part of computer graphics can be divided into three main groups, each of them trying to display its specific information to its specific target observer (Figure 2.1).

Volume visualization treats data inside a three-dimensional grid. The samples are organized as voxels (elemental cubic units of the discretized 3d space) and usually represent values measured or simulated at the corresponding position in a real world scene. Volume visualization is strongly utilized in medicine (computed tomography, magnetic resonance imaging) or in visualizing physical phenomena like fire, clouds or fog.

Flow visualization puts stress on displaying movement of small parts of a larger mass in a certain environment. Similarly to volume visualization, the scene is usually divided into elemental units. The values represent physical properties of the mass such as velocity, direction, density, pressure and others.

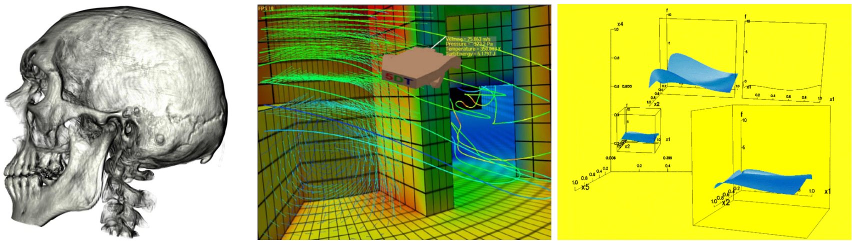


Figure 2.1: Examples of the three main domains of the computer visualization: volume visualization of a human skull, flow visualization of an indoor air conditioning, information visualization using the worlds within worlds method.

Information visualization is oriented on displaying usually arbitrary data without any assumed knowledge about it. It has virtually no limits as for the structure, dimensionality or type of the data, even flow or volumetric data can be observed using information visualization. Therefore it is popular in statistics, economy or engineering.

Information visualization (infovis) tries to display general data in a way that is easier perceived by human than its usual representation. Records of the historical attempts to achieve such a display prove that infovis has been a useful scientific tool for a long time now. One of the best known and also an early example of the benefit of infovis is Dr.Snow's map of the Soho quarter [51] in London. It illustrates both the confirmatory and the discovery goals of infovis.

In September 1854, an epidemic of cholera broke out in Soho. During first three days of the month 127 people died and the number of death cases reached 500 by 10 September [51]. John Snow already had a suspicion that the virus is transmitted through sewage-tainted water, but the authorities together with the sewer maintenance company refused to trust his theories. His research among the locals accompanied by a graphical representation of death cases on a map of Soho (Figure 2.2) lead him to discovering the root of the outbreak, which was a water pump on Broad Street. When they had the pump removed, the spread of cholera dramatically stopped, which eventually confirmed his theory.

Even the technological contribution to visual communication during the 20th century is not sufficient without an adequate attention to the type of the visualization. The space shuttle Challenger launched in January 1986 exploded two minutes after the lift off because of a leakage near the main fuel tank [37]. The engineers recommended to delay the flight because of the chilly weather, but their presentation was not persuasive enough and the authorities decided otherwise. Seven astronauts died because the link between ambient temperature and resilience of large rubber rings was shown in a very incommunicative way (Figure 2.3a). The overhead projector and the slides of the presentation were of no use compared to the impression that could have been achieved by simply drawing the data into a scatterplot [52] (Figure 2.3b).

With the arrival of modern computers with graphical interface, infovis became a

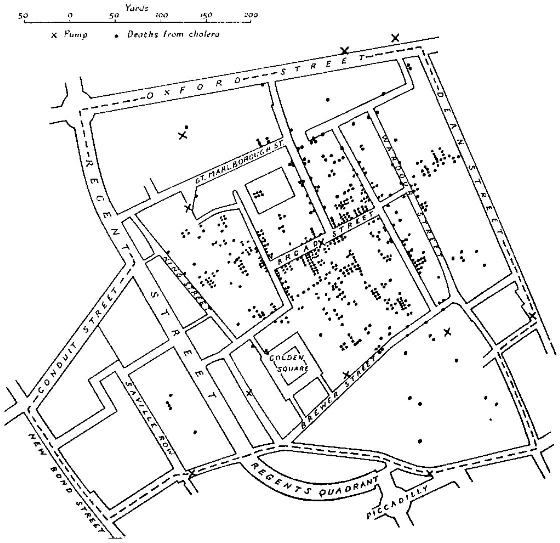
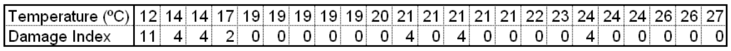
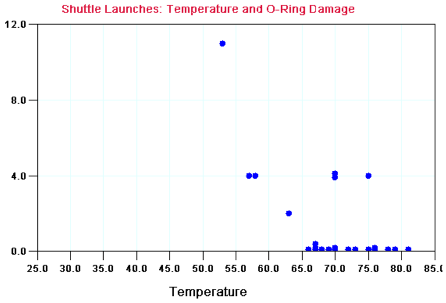


Figure 2.2: John Snow's map of the cholera spread in the Soho quarter. The black dots mark the deaths of cholera. In the visual center of the dots lies a black cross marking the position of the Broad Street pump, which was indeed the local center of the epidemic.



Temperature (ºC)	12	14	14	17	19	19	19	19	19	20	21	21	21	21	21	22	23	24	24	24	26	26	27
Damage Index	11	4	4	2	0	0	0	0	0	0	4	0	4	0	0	0	0	4	0	0	0	0	0

a)



b)

Figure 2.3: The relation between O-rings damage and temperature in a tabular and in a graphical representation. It is hard to imagine that anyone seeing this graph would have decided to launch at 30 degrees. (Temperatures in Fahrenheit)

part of the computer graphics domain. Computers with their processing power are a valuable tool for handling real world data or to conduct complex computations in order to gain more knowledge about the world. On the other side of the computer display lies another powerful processing entity – the human brain. Yet the communication between them flows mostly through the limited resources of the display [51]. This fact turns the visualization into one of the crucial parts of the whole data exploration process. No matter how meaningful the data is, the meaning of it and even its existence can easily end up hidden or lost if an improper visualization is chosen. The purpose of infovis is to avoid such cases and to communicate as much information as possible through the computer display. This quality is called visual effectiveness.

2.1 Visual data exploration

A good reason for an effective visualization is the aim to better involve the human in the data mining process. Computers have huge storage capacity combined with fast computational ability. Unlike them, the power of human processing lies in the flexible and creative mind with the ability to learn and to think intuitively. Human intervention in the exploratory data analysis improves the situations where automatic algorithms often fail. The qualities of the human visual system lie in its sensitivity to specific stimuli. They can be used to improve the exploration process. It takes a moment for the eye to find the nearest neighbor of a spot in a diagram. Detecting underlying structures becomes much easier with the help of human intuition. These and many other tasks are quite complicated for a machine, but humans easily manage them with the only limitation being the clarity of the display. The combined effort of visual and automatic data exploration yields good results even in case of noisy or inhomogeneous data [28].

The process of visual data exploration can usually be divided into three steps, defining the Information Seeking Mantra: *Overview first, zoom and filter, and then details on demand.* [46]. For the purposes of any further investigation, the user has to be provided with an overview of the whole data first, upon his request the view can zoom in on a desired area and filter out the rest. Then the area of interested can be shown in more detail.

2.2 Data of interest

The target areas of infovis involve many different domains. For example economical sciences usually provide statistical data about currencies or stocks. Biology works with color, length, age, feeding time etc. Many other types of data are common in engineering, networks or traffic control. Regardless of the differences between them the data they produce share many similar properties. They usually are multivariate, covering many aspects of the observed samples. They are stored in a computer since most of them are automatically processed before visualization. And they keep growing in volume every day.

The various data do not only share similar form. The real world origin of the data often implies presence of patterns, similar groups, noise or outliers in the content of the data. Understanding the meaning of such features through the visual exploration can help better to know the data and to gain further knowledge about the subject they describe.

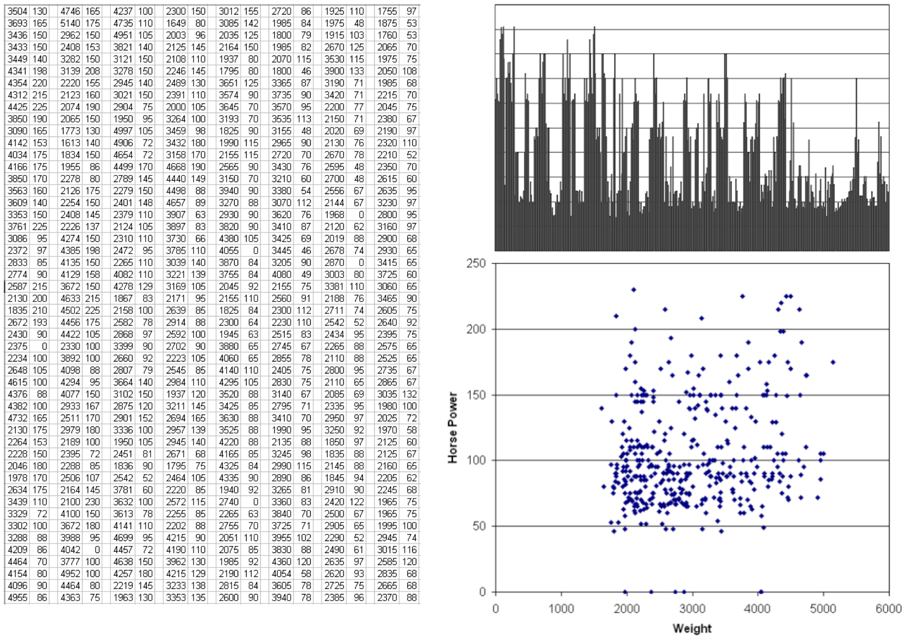
2.3 Information visualization system and interaction

Visualization creates a powerful communication channel between the human and the computer. Organization of this channel plays a crucial part in overall comprehension of the information provided by the graphical display. Information visualization is a bidirectional process with the computer on one side and the human on the other. The computer filters out unnecessary data and projects the rest on the screen using some visualization method. The user interacts with the data by sending his requests to the computer. The requests are processed inside the computer and a new visualization is provided. Repeating this cycle leads the user to investigating the data, understanding its meaning and observing the areas of his interest.

2.3.1 Interaction

User interaction plays an important role in the process of visual data exploration. Rarely it occurs that the first graphical form of the data perfectly satisfies the needs of the target user. Transformations of the display, changing the parameters of the projection and data management are necessary for obtaining the best visualization. The user intervention to the visualization process happens on different occasions. If the user is not satisfied with the portion of the data that is displayed, he transforms the display by rotating, panning, zooming or specifying new values to determine the data to be visualized. If the desired part of the data is displayed, but in an inappropriate way, either the data or the display have to be filtered to provide a better view.

The parameters of the chosen interaction can be entered by specifying the exact values, or they can be adjusted directly on screen. Direct manipulation takes advantage of fast feedback and fast user interaction. It is often a more convenient way to interact with the infovis display than a numerical input. Using direct manipulation, the user is not obligated to know the precise values since he is provided with an instant overview of the effects of his interaction. The realtime response of the system to the user stimuli gives the user the impression he is in direct contact with the data and it helps him coordinate his exploration. This approach is utilized in simple geometric transformations like scaling or rotation and also when performing selection. The main criterion for the direct manipulation is the speed of the feedback. A behavior that refreshes its state at least approximately 10 times per second is considered realtime in terms of information visualization. If the refresh rate falls under this level, the transformation of the display becomes a trial-and-error process [14].



3504	130	4746	165	4237	100	2300	150	3012	155	2720	86	1925	110	1755	97
3693	165	5140	175	4735	110	1649	80	3085	142	1985	84	1975	48	1875	53
3436	150	2962	150	4951	105	2003	96	2035	125	1800	79	1915	103	1760	53
3433	150	2408	153	3821	140	2125	145	2164	150	1985	82	2670	125	2065	70
3449	140	3282	150	3121	150	2108	110	1937	80	2070	115	3530	115	1975	75
4341	198	3139	208	3278	150	2246	145	1795	80	1800	46	3900	133	2050	108
4354	220	2220	155	2945	140	2489	130	3651	125	3365	87	3190	71	1985	68
4312	215	2123	160	3021	150	2391	110	3574	90	3735	90	3420	71	2215	70
4425	225	2074	190	2904	75	2000	105	3645	70	3570	95	2200	77	2045	75
3850	190	2065	150	1950	95	3264	100	3193	70	3535	113	2150	71	2380	67
3090	165	1773	130	4997	105	3459	98	1825	90	3155	48	2020	69	2190	97
4142	153	1613	140	4906	72	3432	180	1990	115	2965	90	2130	76	2320	110
4034	175	1834	150	4654	72	3158	170	2155	115	2720	70	2670	78	2210	52
4166	175	1955	86	4499	170	4668	190	2565	90	3430	76	2595	48	2350	70
3850	170	2278	80	2789	145	4440	149	3150	70	3210	60	2700	48	2615	60
3563	160	2126	175	2279	150	4498	88	3940	90	3380	54	2556	67	2635	95
3609	140	2254	150	2401	148	4657	89	3270	88	3070	112	2144	67	3230	97
3353	150	2408	145	2379	110	3907	63	2930	90	3620	76	1968	0	2800	95
3761	225	2226	137	2124	105	3897	83	3820	90	3410	87	2120	62	3160	97
3086	95	4274	150	2310	110	3730	66	4380	105	3425	69	2019	88	2900	68
2372	97	4385	198	2472	95	3785	110	4055	0	3445	46	2678	74	2930	65
2833	85	4135	150	2265	110	3039	140	3870	84	3205	90	2870	0	3415	65
2774	90	4129	158	4082	110	3221	139	3755	84	4080	49	3003	80	3725	60
2587	215	3672	150	4278	129	3169	105	2045	92	2155	75	3381	110	3060	65
2130	200	4633	215	1867	83	2171	95	2155	110	2560	91	2188	76	3465	90
1835	210	4502	225	2158	100	2639	85	1825	84	2300	112	2711	74	2605	75
2672	193	4456	175	2582	78	2914	88	2300	64	2230	110	2542	52	2640	92
2430	90	4422	105	2868	97	2592	100	1945	63	2515	83	2434	95	2395	75
2375	0	2330	100	3399	90	2702	90	3880	65	2745	67	2265	88	2575	65
2234	100	3892	100	2660	92	2223	105	4060	65	2855	78	2110	88	2525	65
2648	105	4098	88	2807	79	2545	85	4140	110	2405	75	2800	95	2735	67
4615	100	4294	95	3664	140	2984	110	4295	105	2830	75	2110	65	2865	67
4376	88	4077	150	3102	150	1937	120	3520	88	3140	67	2085	69	3035	132
4382	100	2933	167	2875	120	3211	145	3425	85	2795	71	2335	95	1980	100
4732	165	2511	170	2901	152	2694	165	3630	88	3410	70	2950	97	2025	72
2130	175	2979	180	3336	100	2957	139	3525	88	1990	95	3250	92	1970	58
2264	153	2189	100	1950	105	2945	140	4220	88	2135	88	1850	97	2125	60
2228	150	2395	72	2451	81	2671	68	4165	85	3245	98	1835	88	2125	67
2046	180	2288	85	1836	90	1795	75	4325	84	2990	115	2145	88	2160	65
1978	170	2506	107	2542	52	2464	105	4335	90	2890	86	1845	94	2205	62
2634	175	2164	145	3781	60	2220	85	1940	92	3265	81	2910	90	2245	68
3439	110	2100	230	3632	100	2572	115	2740	0	3360	83	2420	122	1965	75
3329	72	4100	150	3613	78	2255	85	2265	63	3840	70	2500	67	1965	75
3302	100	3672	180	4141	110	2202	88	2755	70	3725	71	2905	65	1995	100
3288	88	3988	95	4699	95	4215	90	2051	110	3955	102	2290	52	2945	74
4209	86	4042	0	4457	72	4190	110	2075	85	3830	88	2490	61	3015	116
4464	70	3777	100	4638	150	3962	130	1985	92	4360	120	2635	97	2585	120
4154	80	4952	100	4257	180	4215	129	2190	112	4054	58	2620	93	2835	68
4096	90	4464	80	2219	145	3233	138	2815	84	3605	78	2725	75	2665	68
4955	86	4363	75	1963	130	3353	135	2600	90	3940	78	2385	96	2370	88

Figure 2.4: An excerpt from the cars data set as a tabular data, a bar chart and a scatterplot.

2.3.2 Multiple viewports and linking

The wide variety of infovis techniques, further described in Section 2.4, offers many ways to display the data. Each of them has its pros and cons and a careful combination of multiple views can help eliminate the disadvantages of the included techniques. The best condition for an effective usage of multiple views is a diverse nature of the data [53], like surveys including nominal, geographic and numerical variables, or the necessity to observe a portion of the data in a different way than the rest, like a scatterplot view combined with parallel coordinates for example (Figure 5.4). The multiview environment is only effective when the user is fully aware of the link between the views and the relation of the data displayed in one view to the data in other views. Samples selected in one view should also be made selected in other views as well. This process is called linking views and it is a popular concept among visualization techniques [53].

2.4 Visualization techniques for displaying information

Throughout the years of development of information visualization many shapes and methods were used for the graphical representation of data. The wide variety of the techniques for depicting mostly quantitative information is hard to be classified into a working taxonomy. In spite of that, several classes can be created depending on how a particular technique deals with the visualization of multidimensional data [20]:

- Dimensional subsetting
- Dimension reduction
- Dimensional embedding
- Pixel oriented techniques
- Axis reconfiguration

In general, infovis deals with data sets of high dimensionality and even if there are cases where the data has a ‘displayable’ low dimensionality, multidimensional data in this work is considered to have 4 and more dimensions.

2.4.1 Dimensional subsetting – scatterplots

A simple way to gain a useful graphical representation of multidimensional data is to restrict the dimensionality of the visualized data to a lower number, usually not higher than three, by producing a low-dimensional projection of the whole data set. Visualization of the projection is a much easier task and can be done using a bar chart (Figure 2.4, top right), a scatterplot (Figure 2.4, bottom right) or any other basic visualization method.

The scatterplot is a simple yet visually powerful graphical display. Two axes are placed perpendicular, each representing one dimension. A point is drawn for every sample at its appropriate position according to the axes. This is very similar to plotting a graph of a function or a cartesian product of two sets. The power of scatterplot lies in its ability to depict correlations inside the data, therefore it is a favorite visual tool for statistics.

Scatterplots can be combined to create a so-called scatterplot matrix [10]. Scatterplots are produced for all combinations of two dimensions from the whole set of dimensions in the data and are placed as tiles into a matrix. The matrix provides an overview of all the two dimensional slices.

A natural extension of the traditional scatterplot is a three dimensional scatterplot [34, 4]. The problems with hardly interpretable display and with occlusion were a unwanted drawback of having an extra dimension shown in a scatterplot. But the modern graphical hardware allowing sophisticated rendering methods and realtime interaction with such a display improves the situation greatly and makes the 3D scatterplots a valuable tool for visual exploration.

2.4.2 Dimension reduction – MDS and SOM

Traditional scatterplots produce a planar representation of the data according to two specific dimensions. This approach is effective but neglects the information hidden in the other dimensions. The dimension reduction approach also strives to decrease the number of dimensions to be visualized, but instead of picking a dimensional subset it displays the data in a two dimensional view while maintaining most of the intrinsic relations inside it. For example multidimensional scaling [36] (MDS) is a visualization technique that displays multivariate data in a way that maintains short distances between similar samples and long distances between dissimilar ones. Using a similarity matrix for all the samples from the data set it places the samples into the view and reconfigures their positions to get the best approximation of the mutual similarities. The result of the multidimensional scaling can be a view that actually differs from the geometrical representation of the original data, but visually keeps the information given by the similarity matrix. The drawback of this benefit is the higher computational complexity than simple dimensional subsetting. Self Organizing Maps [33] (SOM) is a visualization technique that has a similar philosophy as the multidimensional scaling, but takes advantage of neural networks as a tool for creating planar representation of multidimensional data.

2.4.3 Dimensional embedding

Similarly to subsetting, the dimensional embedding utilizes creating a low-dimensional projection of the original data. The projection is produced as a slice at a certain position inside a high-dimensional space. Several slices can be produced at different positions and placed accordingly into a wrapping space. This requires dividing the dimensions into those that are a part of the slice (the embedded dimensions) and those that create the wrapping space (the embedding dimensions). An example of this approach is worlds-within-worlds [17], an information visualization metaphor that allows to navigate across the wrapping space and explore the slices at the given position (Figure 2.1, right). Due to the non-zero space taken by the visual representation of the slice, it is naturally not possible to display all the slices at once. Navigating in worlds-within-worlds makes it possible to display every slice though not all at once.

A different tradeoff was chosen in dimensional stacking [40]. Here the space is discretized which limits the total number of different slices into a finite number and it is therefore possible to display them all at once. The embedding dimensions are usually two or three and subtracting them from the original dimensions can still leave a number of dimensions that is hard to display. In this case the remaining dimensions are further recursively embedded until the dimensionality of the slice is sufficiently decreased.

2.4.4 Pixel oriented techniques

Unlike other methods that work mostly in the geometrical space of the data, it is also possible to focus on the viewport with an intention to put as much information onto a single pixel as possible (Figure 2.5). Data can be represented on a per pixel basis by areas of different shapes and colors. Methods using this approach are capable of displaying a large number of samples [30, 31] thanks to the minimal size of visual elements. The drawback of the pixel-based layout is a decreased capability of displaying high number of dimensions and to observe complex relations. Pixel oriented techniques are therefore most suitable for large databases of a low dimensionality like for example source codes or internet site accesses.

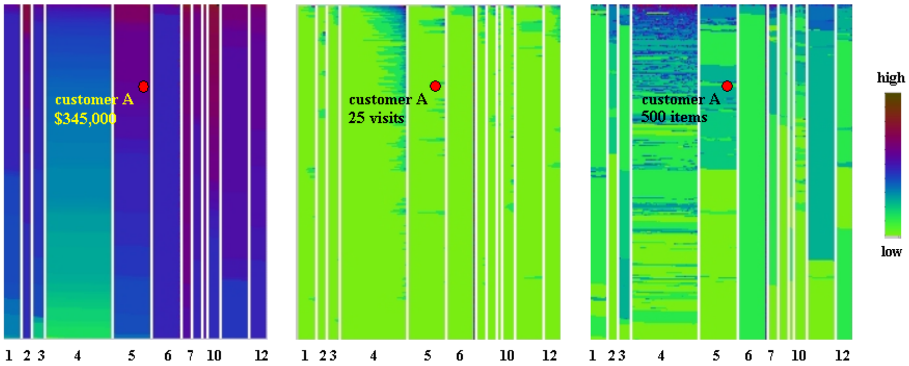


Figure 2.5: A pixel bar chart [30].

2.4.5 Axis reconfiguration

Another way to handle multivariate information is to project it into a special coordinate system which could improve the visual exploration of the data. Its axes can be far from the usual concept of a coordinate system. For example starplot [8] places the axes into a radial layout and draws the sample as a line connecting the positions of values on the axis representing the respective dimension. This creates a ‘star’ for every data sample. But the axes don’t have to be represented as lines at all. Chernoff faces [9] are a visualization method that maps different data properties to facial characteristics (size of the nose, angle of the eyebrows, smile, frown etc.) utilizing the natural human sensitivity to facial expressions. Both these techniques visualize a single data sample by a relatively large representation which makes them not suitable for large data visualization. Another visualization method, the parallel coordinates [25] utilizes the axis reconfiguration approach as well, but is not much less limited in both the number of dimensions and the volume of visualized data. The coordinate axes are placed parallel into a two dimensional plane and the data samples are depicted as polylines connecting particular values on the axes. Parallel coordinates as the working ground to illustrate visually effective information visualization are further described in Section 5.1.

2.5 Large data information visualization

The fast development of technology for automatic data processing causes a great increase in the volume of processed data. It is estimated that one exabyte of unique data is produced every year [29]. Much of this data is also a source object for infovis. Faster computer hardware produces more precise simulation data, larger storage devices provide larger databases, increasing volume of online commerce involves information about millions of customers. Information visualization has therefore often to deal with this large data, unfortunately many contemporary visualization methods suffer from various problems connected to large data.

2.5.1 Loss of speed

As mentioned in Section 2.3 the response time of the application affects human interaction and reducing the speed below a certain value can have negative influence on the process of visual data exploration. Two major aspects delimit the speed of the interaction with an infovis display and can cause a significant degradation to the interaction between the user and computer.

First the speed of the feedback is determined by the simple refresh rate of the graphical interface. With a growing number of visual elements on the display the graphical hardware spends more time rendering a single frame of the whole view. If the reduced frame rate falls below 10 frames per second it is hard to maintain an impression of a real time manipulation. This criterion is important especially during simple transformations of the display – rotating, panning, scaling and zooming.

The second aspect that affects the overall response time of an infovis display is the speed of the data operations while performing selection, brushing or filtering. Geometrical transformations of graphical representation of the data are usually not sufficient to complete these tasks. Usually at least some interaction with the data needs to be conducted.

Displaying large data sets often obstructs an effective interaction with the display. Not only the low frame rate caused by the growing number of visual elements make the simple interaction worse. Also the high dimensionality together with the large number of samples significantly prolongs the time required to process more complex tasks. The response times rise from milliseconds to seconds and the feedback is often too slow for an effective visual exploration.

2.5.2 Occlusion

The problem with occlusion is not strictly bound to large data visualization. Any visualization method that does not prevent overlapping of samples has to deal with occlusion. It happens when two or more graphical elements have to be placed at the same location. This results in drawing one over another. One of them might obstruct seeing the others and hide important information (Figure 2.6). Misinterpretation of the data is likely to happen in such a case. The probability of two elements being drawn at the same place grows with the number of data samples in the view. Therefore large data visualization faces the occlusion problem very often. Sometimes this can be fixed by changing the projection or rotating the view, but many methods do not involve such a transformation and a different solution has to be proposed.

2.5.3 Aggregation

Another problem with many samples being displayed on the same portion of screen is determining the number of them. Several samples are occluded by the topmost elements and their existence together with the number of them remains hidden. Despite that, a technique is usually capable of displaying only a limited number of distinguishable samples in the same area and from this number on the visual representation remains the same regardlessly of the actual number of elements drawn. (Figure 2.6, left).

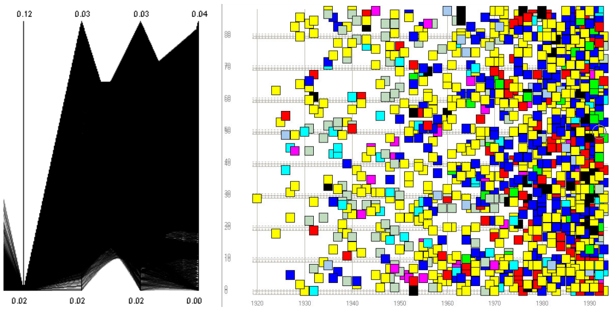


Figure 2.6: Large data causes a significant clutter in information visualization displays. Parallel coordinates view (left) and a scatterplot (right).

The population of a certain area is an important information and discarding it can lead to unwanted consequences on understanding the data. Figure 6.2 shows a case where the information about the number of samples in different regions couldn't be acquired visually and the display gave a false impression of a homogeneous area.

The aggregation problem can be partially solved using semitransparent graphics. This way the opacity of different areas depicts their density because it grows with more samples being drawn over the same place. Unfortunately using standard graphical routines for drawing semitransparent elements, the relationship between the density and the opacity is linear. But the linear mapping is not suitable for the purposes for large data visualization, because the whole range between zero and maximum opacity is divided into intervals representing mutually equal intervals in terms of the density. If the maximum opacity is to correspond with the maximum density, the differences between the smallest densities are not visible, since usually they are assigned the same opacity value. And vice versa, if the differences on the low density level have to be visible, the maximum opacity would correspond to a relatively low density. All the densities above this value will be depicted as the same. Therefore it is necessary to implement a logarithmic mapping, which is capable of displaying important differences on both ends of the density spectrum.

Another drawback of drawing semitransparent elements is that it pushes outliers out of attention since they will not be drawn sufficiently opaque to be noticed. Fixing this problem requires much additional processing towards outlier detection.

2.5.4 State of the art for large data information visualization

Compared to the history of infovis the problem with large data is a relatively young issue. Unfortunately the nature of infovis methods does not allow to rely solely on faster computers and sophisticated graphical hardware, though the speed of interaction is strongly influenced by this factor and can be at least partially remedied by it. The other problems caused by the large data are still an issue and it is necessary to improve the situation either by avoiding them or fixing them. This often requires choosing a different point of view or a new geometrical representation.

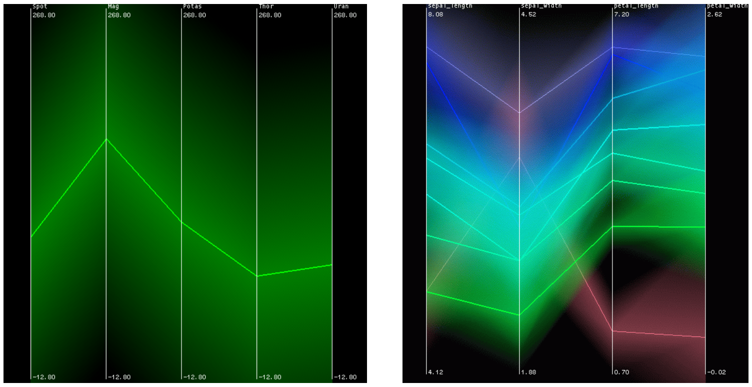


Figure 2.7: Hierarchical parallel coordinates [19].

In general the various approaches to the problem can be described by the domain they are focused on and the visualization techniques they affect. The data-based methods use mathematical and algorithmical solutions for data reduction thus relieving the visualization of the pressure caused by the large data. This approach can be seen in several ad hoc optimizations like [45] or [44]. Much research towards large data visualization is also being done in the field of Self Organizing Maps [27, 35], because the nature of the algorithm provides preconditions for an effective visualization.

The display-oriented methods are trying to put as much information to the screen as possible while reducing the overall clutter. Pixel-based methods such as [31], [30] are capable of displaying even millions of items without overlapping or aggregation. But the number of dimensions that such methods can display is quite low and the limits of human perception to pixel-size elements have to be taken into account when displaying large data on a big screen. Other works extend traditional methods to display larger volumes of data by using semitransparent elements [16] (this topic is further addressed in Section 2.5.3).

2.5.5 Hierarchical parallel coordinates

One of the most interesting works on visual abstraction is presented in [19]. Similarly to the work presented in this thesis, the data are organized into clusters before the visualization and then the clusters are displayed in parallel coordinates (Figure 2.7). The visual representation of the cluster consists of a fully opaque polyline as the centroid of the cluster and then the area of the cluster fades out on both sides of the centroid to fully transparent borders of the cluster. This visualization effectively encodes basic statistical information about the cluster such as the mean and the extents. But other important information like density or population is missing, which might cause possible disadvantages of the display.

- Outliers can be easily lost in such a display since their presence is often implied by a low density or a extremely wide extent of a cluster. In hierarchical parallel coordinates, this information is not obvious and therefore the applied clustering had to treat outliers very well, which results in high computational demands.

- The clusters with similar extents but different number of members are displayed in the same way, which may cause bad interpretation of their relative importance.
- Opacity as the value that communicates distance from the centroid gives an impression that the data inside the cluster is organized into a normal distribution. Which in many cases is not true.

Unlike the hierarchical parallel coordinates, the work presented in this thesis either addresses these problems or avoids them. Plus it presents an effective structure to store the hierarchical information And although the parallel coordinates are chosen as the main visualization method for this thesis, the concept presented here is easily transferred to other visualization methods (scatterplots, MDS) as is not strictly bound to the parallel coordinates.

Chapter 3

Clustering

Almost every set of data that in some way reflects the real world contains groups of similar samples. Such a group can represent a trend among the observations, like for example a group of customers with similar preferences or specimens of the same origin. A single member sample of such a group usually does not carry much information important for the first glance of the data. Therefore replacing the samples from within the group by statistical information about the group significantly reduces the number of data and creates an abstract and less cluttered information.

Clustering is a process that divides the observed space into areas that are thought to represent separate phenomena of the real world data. Its result is an estimate of position and properties of the groups that are contained in the data. The clusters are a good starting point for creating abstract representation of the data.

The abstraction usually discards some parts of the original information by aggregating the samples into the clusters. It is an inevitable drawback of a simpler representation. Therefore it is important to maximize the profit gained by the clustering. The visualization of such an abstract information should use all the graphical resources provided to effectively display as much information about the abstract data as possible.

3.1 Groups, patterns, noise and outliers

The samples of similar origin or having similar properties tend to create groups. This behavior can easily be discovered in many visualizations and new groups can be found by the naked eye (Figure 3.1). Other patterns can be discovered and explored visually too. Direct/indirect proportionality, trends, equilibriums – many of them would require intensive effort to be discovered on a computational basis. Using the visual exploration makes this discoveries much easier and a new hypothesis might be formulated upon a visual inspection of the data.

A common feature of the real world data is also the presence of samples that do not follow any pattern or do not explicitly lie in a group. This samples might reflect special cases worth further investigation but it is hard to discriminate between an error and an outlier. The question of the correct tradeoff between suppressing noise and

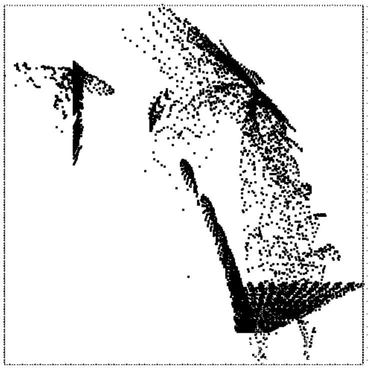


Figure 3.1: A scatterplot of a flow data. Three major features are visible in the data.

emphasizing outliers is a frequent problem in information visualization.

3.2 Types of clustering

The procedure of creating clusters can happen in several ways [48]. The samples can be repeatedly grouped together if they are considered similar until all samples are members of some group or are declared outliers or noise. This is called an agglomerative or a bottom-up approach. Its opposite is the divisive or top-down approach that divides the whole data space into smaller and smaller parts until the supposed groups are isolated in separate parts. The variety of clustering methods is caused by their specific properties. There is no perfect clustering method for arbitrary data therefore different problems require different solutions and the proper clustering algorithm is usually chosen according to the type of data, its origin and the target application. Comparing two algorithms or their results is usually done via comparing the Euclidean sum of squares (ESS) of the discovered clusters. The correlation between reality and estimate by clustering can be expressed using ESS, with lower values meaning a better clustering. It is necessary to notice that the ESS naturally favors spherical clusters. Clusters of rectangular or even chain-like shapes are hard to discover using ESS and therefore other criteria have to be used or different clustering methods (e.g. single linkage) applied.

3.2.1 Top-down clustering

A typical example for the divisive class of algorithms is similar to building a quad tree. The space is recursively divided into smaller and smaller subsets by a hyperplane at specific coordinates and clusters are formed from samples in the same subset. If a cluster matches desired properties it is not divided further. The fact that this algorithm does not strongly depend on the similarity concept compared to the bottom-to-top algorithms relieves it from thorough computation of mutual distances among samples.

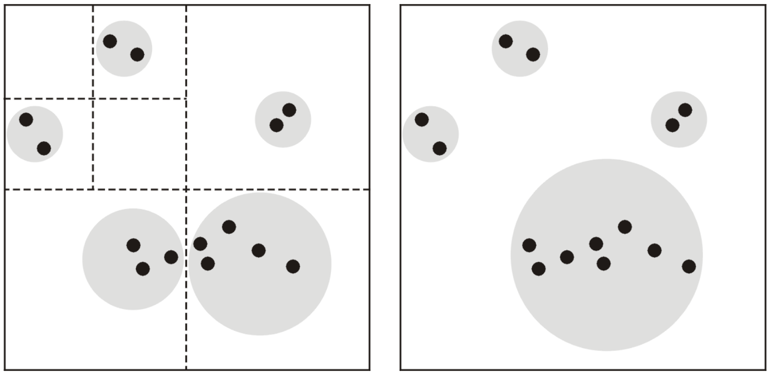


Figure 3.2: Clustering approaches: The top-to-bottom approach (left) accidentally divides the largest group into two clusters in contrast to the bottom-to-top approach (right).

This gives the divisive algorithms a speed advantage and they tend to work faster than agglomerative algorithms.

But the ignorance of mutual distances not only speeds up the overall process, it also may cause errors in the clustering. A careless deciding about the position of the dividing hyperplane can lead to separation of samples that should be placed in the same cluster, as can be seen in Figure 3.2 (left). Another disadvantage of the top-down approach becomes apparent when clustering multivariate data. The divisive nature of the algorithms implies a memory complexity of $O(n^r)$, where n is the number of dimensions and r is the depth of recursion. The combination of a deep recursion and high number of dimensions can result in very high memory demands. Unfortunately, high number of dimensions is a common phenomenon among infovis data. On the other hand, reducing the depth of recursion yields worse estimates. Both of these facts strongly limit the usage of divisive algorithms for purposes of infovis.

3.2.2 Bottom-up clustering

The agglomerative algorithms, also called bottom-up algorithms, start to operate on the lowest level of the data set. They either compare samples with each other in order to create a cluster, or look for the nearest cluster of the sample. Accordingly, the results are usually better than those of divisive algorithms, because it is almost impossible to separate two similar samples. But the mutual comparison among all the samples produces a serious bottleneck of every bottom-up algorithm – a quadratic time complexity multiplied by the time necessary to compute a single distance in the provided data space. Obviously large data will cause these algorithms to slow down, unless some modifications are done towards decreasing the number of necessary comparisons. Another optimization comes with using a table to store distances or similarity values for pairs of samples. It is a very helpful tool, but memory issues arise again when dealing with large data.

An important feature provided by bottom-up clustering is the capability of building hierarchies of clusters. A specific level of the hierarchy corresponds to a clustering at a certain precision level. A popular algorithm called k-means, which is further described in Section 3.3.1 or the minimum spanning trees (MST) algorithm are good examples of building such hierarchies. The MST algorithm comes from the domain of graph theory (it is an approximate solution of the traveling salesman problem), where it is used to create a skeleton of a graph with respect to minimizing the cost function for the edges of the skeleton [15]. If the cost function represents distance between two nodes and the nodes represent samples, the result of the MST algorithm can be used to produce a clustering of the data [3]. All it takes is to ’cut’ the tree at a specified level. Building a cluster hierarchy with such a tree provided is then trivial.

3.3 Clustering large and multivariate data

The proper tradeoff between memory and time demands is a common problem in the computer science domain. And clustering is a typical example of such a situation. The divisive algorithms work faster, but multivariate data can radically enlarge the memory consumption. And there is also the risk of incorrect assignment of samples to clusters. Agglomerative algorithms provide good results without any special memory demands, but the time complexity might reduce the usefulness of the resulting application. Possible solutions can be found in combining these two approaches in a way that mutually minimizes their disadvantages or in modifications of algorithms in order to save space or time. More optimizations towards speed increase are also possible using various heuristic methods [13, 49], since their computational and memory demands are much smaller and the results are satisfyingly similar to exact methods.

For the purposes of visualization, the bottom-up approach seems to be a better choice, since the most negative aspect of such algorithms lies in the time it takes for them to finish. As the clustering is supposed to be performed in the beginning of the visualization this handicap will not affect the visualization in a significant way.

3.3.1 k-means and its modifications

The k-means, an agglomerative algorithm, has several clones and many modifications, but what all of them have in common is the core concept of the algorithm, to produce a given number (k) of clusters with locally minimal ESS. The basic algorithm begins with marking k samples as starting points for the future clusters. Then all of the remaining samples are one by one assigned to the cluster with the nearest centroid and the centroid of the cluster is then recomputed. This process repeats until no samples need to be reassigned [21]. It is obvious that the operation of the algorithm depends on the selection of the number of the k starting points and their placement. It is recommended to choose the k-tuple as the one with maximal mutual distances [18]. Unfortunately there are not recommendation about the value of k, since it strongly depends on the data itself. Therefore it is common to let the user specify the k value or to switch between various possible values.

Determining the best initial points for the k-means algorithm on large data is not a trivial process. It is almost impossible to calculate all mutual distances in a reasonable time and even with the distances already given or computed, finding a k-tuple with maximal mutual distances is not a trivial problem. For the purposes of large data certain modifications need to be done to the original k-means algorithm. A fast choice is to select the initial points randomly, but this approach often produces incorrect results especially for low values of k. Yet it is worth considering for high k since the random choice usually produces a relatively homogenous spread of the initial points. It is not really correct for a clustering, but can be found helpful in data reduction.

The modification to k-means used in this project to demonstrate the process of creating an abstract hierarchy utilizes randomness as well, but with much better results. In the beginning every sample is randomly assigned to a cluster. Centroid of every cluster is computed and all the samples are tested for the nearest cluster and reassigned to it. The centroid recalculations and sample reassignment is performed repeatedly for all samples until a stable solution is found [41]. Several problems might occur. The initial random assignment might use less then k different values across the whole data set, which can be easily fixed. Another problem is a possible infinite loop between several stages that can not converge. The behavior inside this loop is similar to some configurations of 'The Game Of Life' [22]. It is almost impossible to completely avoid this problem, but the chances of its occurrence can be eliminated down to a safe level with randomly changing the order of samples during their test and reassignment stages.

Chapter 4

Visual Abstraction

The concept of describing reality in an abstract way is widely used in every communication media. A simple reason for using less detailed information are the numerous aspects and incomputable details a single event or subject involves. For a comprehension of the subject often a much simpler description is sufficient. A book or a movie represent events focusing on the important parts. The pictures on public places need only few resources to successfully give directions on restaurants, restrooms or exits. Visual abstraction is a means to display ’less’ with the meaning of ’more’. The most important quality of an effective abstraction is to preserve the meaning of the displayed data on a certain level of truthfulness and not to lose connection between different abstraction levels. The process of drilling through abstract layers to the precise information is similar to approaching a distant object. As we shorten the distance from the object we might see a tall shape transform to a person, the person to a man, the man to a friend and finally on the lowest level of abstraction we might see the expression on the face of our friend. It is vital for the various levels of abstraction not to display a distracting or incoherent information. The main purpose for using visual abstraction in computer graphics is to clean up the display and to economically use hardware resources. Both of them are a precious material for computer aided visualization. The need for a faster or easier to understand display creates a good starting point for visual abstraction in many subareas of computer graphics.

Non-photorealistic rendering (NPR) was originally used to simulate various artistic techniques on synthetic images (Figure 4.1, left). But the concept of abstracting the original information into a different shape exceeded the field of aesthetic computer graphics and is now also used to emphasize interesting areas and suppress the context. NPR is also used to enhance volume visualization of medical data [23]. As shown in Figure 4.1 (right), the skin is rendered as a contour enveloping the other tissues and organs. This mode keeps the precise notion of skin in the picture yet the visual representation of it does not obstruct other elements in the scene.

Another example of effective using visual abstraction is rendering visually effective route maps [2]. The usual maps either display the whole route and neglect important details or show the details but require a large area to draw the rest of the route using the same scale. A useful improvement is to use different zoom factors for areas of different

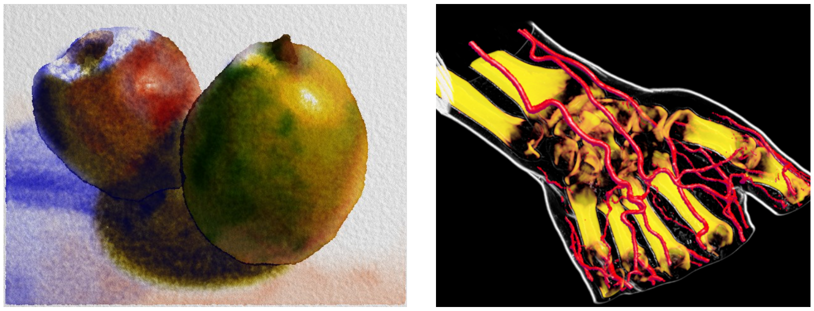


Figure 4.1: A synthetic watercolor painting as presented in [11] (left) and non-photorealism in scientific visualization: the skin is rendered as a contour to enhance the view (right) [23].

importance. The idea of putting more stress on certain areas of the data at the expense of truncating the less important parts is also known as the hyperbolic projection [39]. Compared to the NPR, the effective route maps not only utilize a different rendering mode, they also positively distort the original information in order to emphasize important parts.

The individual legs of a journey do not deserve the same amount of attention when it comes to orientation and directions. A highway usually takes up the longest part of the whole route, but requires only little attention. By contrast a small exit might be of a high importance and missing it might mean a long detour. Usual route maps stem from cartographic projections and use the same scale for all parts of the route (Figure 4.2, left). The maps also often contain distracting and unimportant information like negligible geographical features or insignificant curves on the road. All these features make a regular route map hard to read and high amount of attention is necessary to comprehend the information displayed by it. The system for rendering effective route maps evaluates the importance of every single part of the route and chooses the proper scale for it. Sometimes it is also necessary to reorganize the original layout since the emphasized parts could overlap with the rest. The resulting structure (Figure 4.2, right) can look quite different from the original cartographic representation, but as the user study [2] shows, it is considered much more useful and clearer.

4.1 Level of detail

An abstract overview is generally a good improvement to many visualization techniques. But the neglected details might be found useful later on during a more thorough investigation. For that purpose, implementing various levels of abstraction is a valuable idea. The abstract information can be gained from the original data with various precision and the different abstract representations can be stored in a structure called level-of-detail (LOD). Three dimensional rendering uses LOD for storing different levels of decimated meshes of models because distant objects do not require a

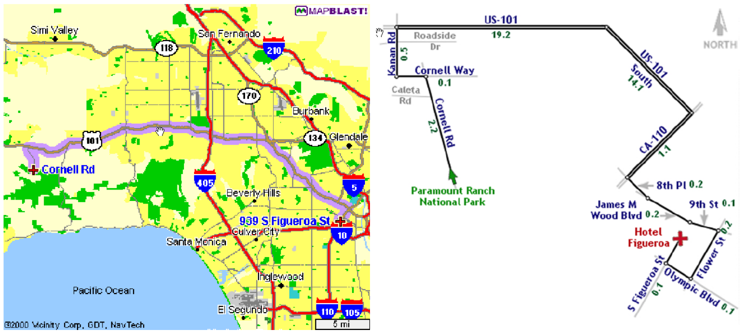


Figure 4.2: Route map visualization in traditional software (left) and using visual abstraction (middle) [2].

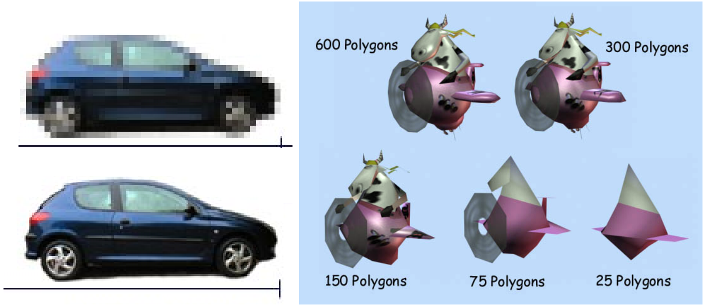


Figure 4.3: Two stages of a progressive rendering process (left) and several levels of detail of a 3D model (right).

precise geometry and thus replacing them with a simpler mesh saves many hardware resources [42]. Progressive rendering takes advantage of LOD to quickly draw the crudest level first. This provides the user with a preview and the further levels are rendered afterwards to enhance the image until the final, most detailed, stage is achieved.

4.2 Visual abstraction for information visualization

The subject of computer aided information visualization is not displaying objects or events directly, but rather displaying observed features of usually a larger number of objects. Visualization methods alone produce an abstract graphical display that is dissimilar from the shape of observed objects. Therefore the meaning of visual abstraction transforms itself into a slightly different form. The whole data set becomes the basis for building an abstraction upon. Not the features of objects but objects (samples) themselves are the details which the abstraction removes in order to increase the clarity of the display. The abstract information should be based on the original data, maintain its meaning and intrinsic nature, but require less attention and hardware resources. To achieve this, a certain knowledge about the underlying structures inside the data are necessary. Infovis tries to treat its data as generally as possible, because of the wide variety of target areas of application. Unlike flow visualization [43] or medical visualization [23], it is impossible to assume any a priori knowledge about the infovis data. The process of abstracting to a higher level from the original data has to be as general as possible. Therefore unsupervised data mining or clustering seem to be a good choice. With the abstract information provided, the cluttered graphical representation of a large data display can be replaced by new visual elements corresponding to higher levels of abstraction.

4.3 Visual exploration vs. automatic data mining

Using machine-based approaches to data mining might seem a little counterproductive to visual data exploration. Both of them are trying to achieve the same – understand the meaning and the nature of the data – but each of them uses different methods and works in a different way. The combination is not a common phenomenon. The computer is usually used either as the data explorer (in case of unsupervised data mining) or serves as a mere tool for an easier access to the data. But as presented in Section 6, the mutual effort of human and computer leads to promising results. Computer intervention in some stages of the exploratory process together with an appropriate graphical representation of the new abstract information prevents the user from being deluged by too many visual stimuli. Yet it preserves the structure of the data to a large degree. To avoid the mutual counteractions of human and computer, it seems a good choice to use the data mining only to build an abstract structure above the original data. These actions are less computationally intensive as a full and thoroughly precise data mining but are satisfactory for the purposes of abstraction and data reduction. Further computer operations on the data should be left upon users decision. The freedom of choice between visual exploration and computer data mining will allow the user to explore the data his own way and adopt the behavior of the system to conditions of the target application.

Chapter 5

Large Data Visualization Using Parallel Coordinates

The cooperation of automatic data mining and visual abstraction creates an effective tradeoff between the computational complexity on the side of the machine and visual effectiveness on the side of the human. By sacrificing some time for preprocessing and preparation of the abstract structure the former visualization pipeline [7] is extended by a branch that involves data mining and building the abstract structure using the results of the mining (Figure 5.1). Thus the resulting visualization needs more time to initialize the abstract information, but the visual differences and improvements are obvious and definitely worth the time spent preprocessing. To illustrate the concept of visual abstraction as a means for visually effective information visualization of large data new graphical elements and a new data structures were developed. Nonetheless, the approach used here can be extended to almost every visualization method and use arbitrary abstract information to build the level-of-abstraction structures above the original data. The structures and approaches used here do not depend on the type of clustering

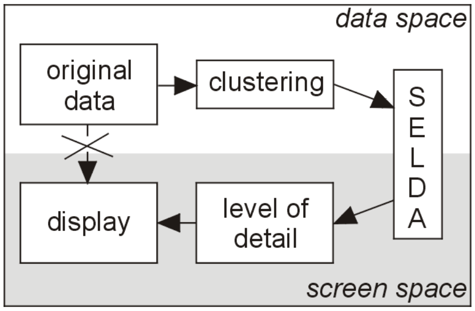


Figure 5.1: The shortest route is not always the fastest. A modified pipeline provides abstract information visualization.

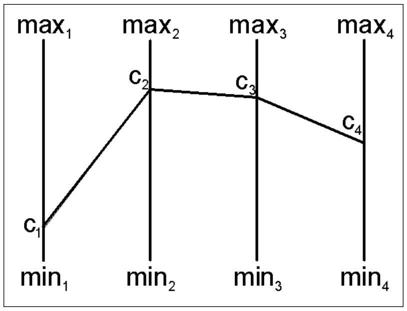


Figure 5.2: Parallel Coordinates. Point $C(c_1, c_2, c_3, c_4)$ is represented by a polygonal line.

or the data mining method.

5.1 Parallel coordinates

A popular visualization technique, already used by statisticians in 1970's and rediscovered by [25] utilizes the axis reconfiguration approach in order to display a relatively high number of dimensions on a two dimensional display. Every n-dimensional point is represented by a polygonal line. Similarly to starplots the shape of the polygonal line depends on the coordinates of the point. The N axes of the parallel coordinate system are placed vertically and equidistant to each other. They define the projection of R^N onto the screen. A point C with coordinates $(c_1, c_2, \ldots, c_N)$ is represented by a polygonal line connecting the positions of c_i on their respective axes (Figure 5.2). This projection provides a 2-dimensional display of the whole data set and is capable of displaying up to tens of different dimensions. Therefore it is widely popular among scientists that work with multivariate data like statisticians, physicists or economists.

Many complex and multidimensional patterns can be observed in parallel coordinates with the naked eye (Figure 5.3). Positive or negative correlations are easily seen between two adjacent axes and the exploration can then be easily extended to other axes as well [24]. The geometry inside parallel coordinates is affected by the projection and creates a dual relation between a point in Euclidean space and a polyline in parallel coordinates [26]. The spatial envelopes of points are easily constructed in parallel coordinates as areas bounded by the marginal values for all the dimensions (Figure 5.6, bottom).

An unpleasant drawback of the point-line duality is the amount of space occupied by a single data sample. This predetermines the display to be heavily cluttered when filled with many samples. Interaction and understanding of such a display is usually very complicated (Figure 5.6, top). But the modifications of parallel coordinates [47, 50] and also the efforts towards efficient displaying of large data in parallel coordinates [19] make this method a promising working ground for visual abstraction and large data visualization.

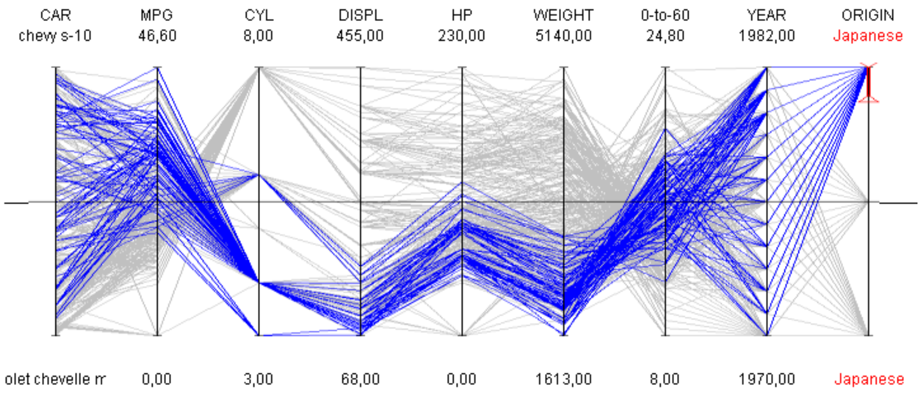


Figure 5.3: Visual exploration in parallel coordinates: highlighting all the Japanese cars (ORIGIN axis) reveals that they are very light (WEIGHT axis), have medium acceleration (0-60 axis) and below medium horsepower (HP axis). Image taken from [1]

5.2 2D binning

Observation of two dimensional relations in parallel coordinates is easy and intuitive. In spite of the fact that other techniques display more of these relations (a scatterplot matrix or dimensional stacking), the number of axes seen at once still makes parallel coordinates a worthy tool for exploring patterns between two adjacent axes.

This thesis presents a solution, where the two dimensional subspace defined by two of the axes is divided into several parts and the samples are classified by the part they reside in. The samples in the same part are assigned to the same bin and the bins together with their samples are colored according to the characteristic of the bin. The three basic characteristics, as seen in Figure 5.5, have their colors assigned and the resulting color of the bin is a combination of the two nearest ones. The color is then extended to the remaining segments of the polylines to emphasize any common behavior of samples that are declared similar in the 2d binning. The same can be done in the linked scatterplot view (Figure 5.4).

The size of the bins is chosen as a 2^k-fraction of the size of the original space. This facilitates on-the-fly transitions into a higher or lower level of abstraction if necessary. Also the adjacent bins have their boundary values modified if there is a relevant assumption that they are a part of the same bigger structure (Mathematically this means that their boundary values are close enough).

2d binning is similar to divisive methods for clustering. As mentioned in 3.2.2, the divisive methods are not much suitable for cases involving many dimensions. But for a two dimensional subset the 2D binning is a very effective tool. The importance of 2D binning is even increased by linking the parallel coordinates with a scatteplot, which provides an additional view of the data (Figure 5.4) and helps to observe the relations between two dimensions even in situations where the large data would produce an image that is hard to interpret. By changing the order of axis the behavior inside an

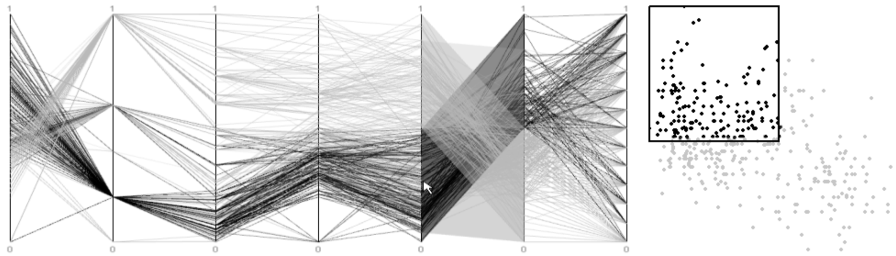


Figure 5.4: 2D binning in parallel coordinates and scatterplots: Data are highlighted according to the occupied portion of the two dimensional subspace between the fifth and the sixth channel.

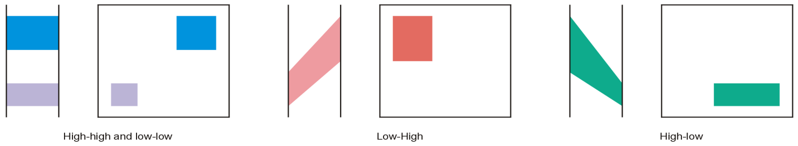


Figure 5.5: Three main characteristics of data in a scatterplot and their corresponding visualization in parallel coordinates.

arbitrary subset can be observed in both scatterplot and parallel coordinates.

5.3 Multivariate abstraction in parallel coordinates

For displaying multidimensional abstract information in parallel coordinates a slightly different approach has to be chosen than for displaying individual samples. Since the abstract structures in this project are mostly clusters, their visual representation can be approximated by their bounding box. The polylines that belong to samples from within such a cluster are replaced by a polygonal shape covering the area occupied by the cluster. Depending on the chosen level of abstraction the number of clusters varies and usually many of them overlap each other. In order to avoid unwanted occlusion the polygons are rendered semitransparent so that all parts occupying the same area are at least partially visible in the resulting image and the information is not lost.

5.3.1 Transparency

The transparency of the graphical elements is controlled via changing the alpha (opacity) value. The opacity value is also used to communicate additional information about the cluster. Several types of opacity mapping are implemented. Most of them utilize a non-linear mapping, as described in 2.5.3. The function $f(x)$ mentioned in following paragraphs is a function that projects the $\langle 0, 1\rangle$ interval onto itself in a logarithmic way.

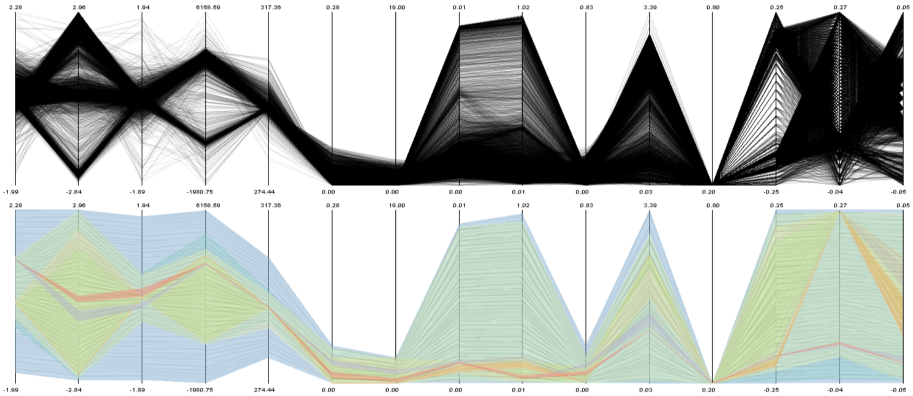


Figure 5.6: Large data visualization in traditional parallel coordinates. Virtually no patters or groups can be observed in the cluttered areas (top). It is also hard to determine the number of samples in different regions. Visually effective information visualization using parallel coordinates (bottom). 10 thousand samples, 15 dimensions.

- Uniform mapping assigns all clusters the same opacity value, usually $1/k$ with k being the number of them. This mode is useful to get a glance of the data and to easily distinguish occupied areas from unoccupied ones. Apart from that this mode is of little use.

- Population mapping puts stress on the number of samples contained in a cluster. The opacity value equals $f(\frac{p_i}{p_{all}})$ where p_{all} is the number of all samples and p_i is the number of samples in the cluster (population of the cluster). Such a mapping emphasizes highly populated clusters and is useful when looking for the strongest trends inside the data. However, it handles outliers and small populated clusters poorly.

- Density mapping seems to be the best choice from among the selected modes. It takes into account the population and also the size of the clusters. The opacity value equals $f(\frac{p_i.s_{all}}{p_{all}.s_i})$ where s_{all} is the volume of the bounding box of all data and s_i is the volume occupied by the i-th cluster. This approach emphasizes both small but dense clusters and big clusters with high population. Another advantage of the density mapping is that it draws less attention to clusters that are relatively highly populated, but are sparse because of their large size.

The different modes can be interactively changed and can serve different purposes. Nonetheless, the density mapping seems to be the most useful mode, since density as the ratio of population to size gives clues about the dispersion inside the cluster and thereby the ‘truthfulness’ of it.

In order to preserve outliers or small clusters in the visualization an ordering of the clusters is performed before their rendering. For every axis in the view, the clusters score points according to the size of the interval they occupy on the axis. The sum of these points gives the resulting rank of the clusters. Then they are rendered in

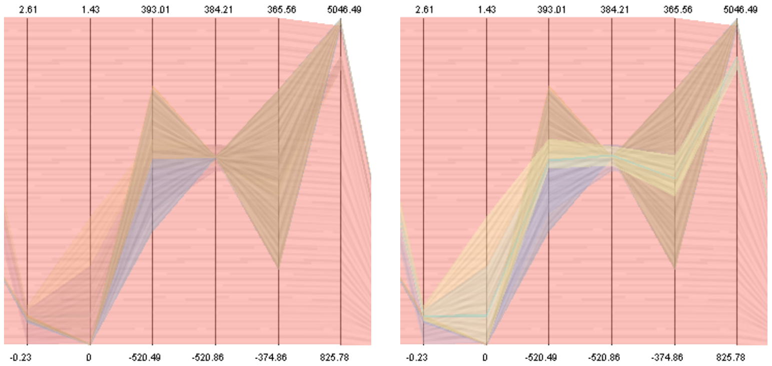


Figure 5.7: The transparent areas before (left) and after (right) ordering the clusters according to their size.

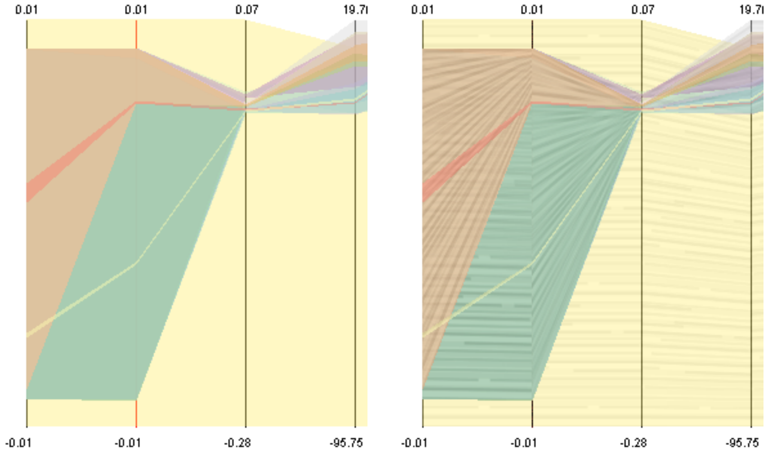


Figure 5.8: The effect of texturing. The real shape of the cyan area is questionable in the view without the texture (left). After the texture is applied it becomes clear, that it partially lies under the light brown area.

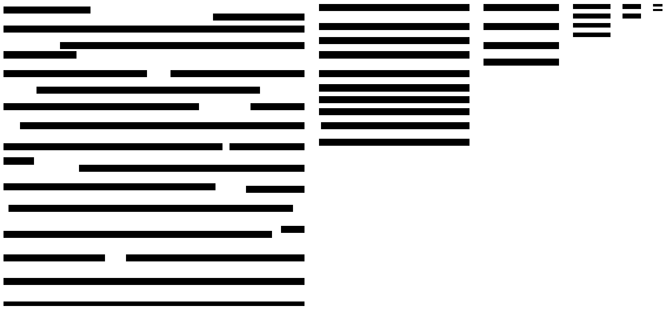


Figure 5.9: The individual mipmap levels designed for an effective hatching using homogenous texture coordinates.

a descending order, with the biggest clusters first. The ordering prevents the bigger clusters from being rendered over the smaller ones and significantly improves the view (Figure 5.7).

5.3.2 Colors and textures

The visual difference between the abstract structures inside the data is easily achieved by using different colors for different objects. But several aspects have to be carefully pondered. The colors have to be as mutually different as possible and share the same intensity. Some colors appear brighter to the human eye than others therefore it is necessary to compensate the effect to avoid some objects being incorrectly emphasized. According to [6] the palette of these properties should not contain neither sequential schemes (useful for ordered data) nor diverging schemes. A qualitative scheme acquired from [5] is applied to color the individual elements. The palette distinguishes the objects by hue while keeping the color intensity relatively the same for all colors. The resulting effect is smooth and well arranged.

Although the colors in the chosen palette are various, often it happens that areas of similar colors are placed near each other. And sometimes a specific configuration of overlapping polygons might create an incorrect impression (Figure 5.8). This can be remedied by drawing a certain texture to simulate the directions of the underlying polylines. The texture is originally monochrome, but it is tinted according to the color of the appropriate region. It is also important for the texture to be seamless and not to produce alias. After applying the texture, the clutter in the image is only slightly increased, but many questionable areas in the display are enhanced and a more correct information can be observer visually (Figure 5.8). The human visual system is quite sensitive to shapes and patterns, therefore the information about overlapping clusters is very easily discovered by perceiving the change of the hatching of the respective area.

As mentioned above, the hatching of the area is achieved by applying a stripe texture. In order to emphasize the shape of the area and to give a similar impression as the original polylines, this texture has to be appropriately stretched. Since the view

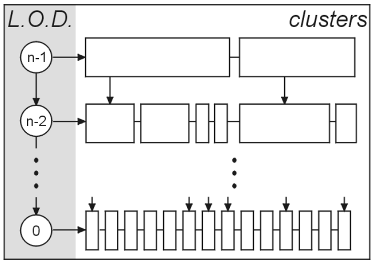


Figure 5.10: Diagram of the SELDA structure. Notice that only one pointer is used to address the children of a node.

is implemented into an OpenGL environment, the `glTexCoord4f()` function was used. Using the homogenous texture coordinates that this function provides brought up the problem of mipmapping in areas where the texture has to be stretched unevenly. Regular mipmaps produce a very distorted image in such areas because of using the smallest mipmap for narrow parts of the textured quad. This was fixed using a special mipmap [32], that maintains similar appearance of the texture on different levels of the mipmap (Figure 5.9).

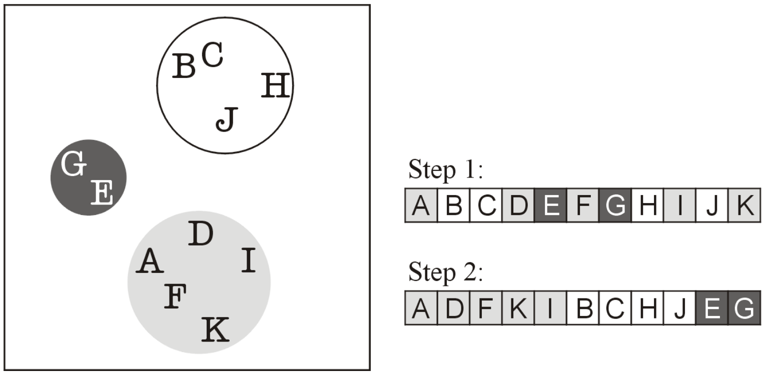
5.4 Data management

For an effective visual abstraction certain structures have to operate on the original data to supply the necessary abstract information. The need for coherent multiple levels of detail implies a hierarchical nature of the structure. Similar samples are members of the same cluster and similar clusters are children of the same cluster on a higher level of abstraction. A tree-like structure is a natural choice for such needs but the large data issue has to be taken into account. Usual trees used to represent hierarchical information might grow to gigantic volumes when a high number of leaves has to be included. Since the data itself often occupies hundreds of megabytes it is undesirable to build a structure of a similar size.

Another important aspect is the security of the data. Often an infovis display is linked with other displays in order to cover different aspects of the data. An inappropriate intervention to the source data might cause the visualization in other views to fail, therefore the data should be treated remotely and independently of the other views.

The Structure for Efficient Large Data Abstraction (SELDA) is optimized for large abstract hierarchies and fast level of detail transitions. It contains a tree of clusters connected to the relevant abstraction level, a table for resolving members of a cluster (Data Index Table – DIT) and a table that stores cluster membership for every data sample (Cluster Index Table – CIT). Every node of the tree represents one cluster in the abstract hierarchy and stores statistical information about it: population, boundary values, mean and variance.

The target depth of such an abstract tree is probably not higher than four or five and the number of clusters is relatively low compared to the number of samples. Therefore



Step 1:

A	B	C	D	E	F	G	H	I	J	K

Step 2:

A	D	F	K	I	B	C	H	J	E	G

Figure 5.11: Data Index Table reordering. In the first step, clusters are assigned to data. In the second step, the indices are ordered according to their cluster membership.

the biggest concern about the size of the structure relates to the lowest level of abstraction, where a large number of samples has to be assigned to their parent cluster. Usually a cluster would have a pointer stored for each of his children, which results in high memory demands. To avoid this the following optimization was implemented. The references to samples in DIT are ordered according to their cluster membership. The resulting table contains references grouped in their respective clusters (Figure 5.11). This allows for using only one pointer to refer to all the samples inside a parent cluster. Together with the number of child samples stored in the cluster two numbers total are necessary to delimit the references to children samples. Thanks to this modification fixed and predictable memory demands are assured for the price of simple preprocessing. The concept of sorting children nodes can be applied to all levels of abstraction, as shown in Figure 5.10, the structure uses only one pointer for all children nodes of one parent node. This optimization causes the structure to take up much less memory than an ordinary tree.

Building of SELDA is a three-step process. In the first step, clustering or any other data mining method is performed to obtain information about groups of samples and their members. This information is stored into CIT. In the second step, the DIT is sorted as described earlier and basic clusters on the lowest level of abstraction are formed. The basic clusters are the clsuters on the lowest level of abstraction. They are produced as the result of clustering the original data and therefore are built using different rules than the upper levels of SELDA. Finally higher levels of the hierarchy are abstracted from the basic clusters.

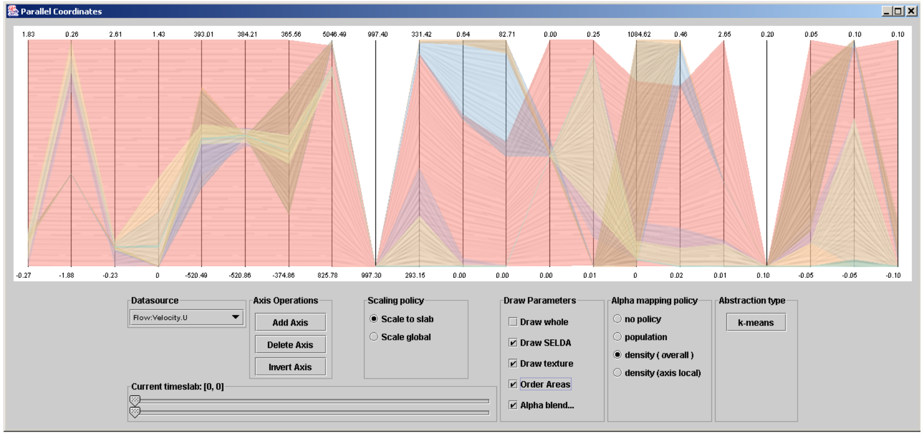
Dividing the building of the abstract structure into several steps offers a possibility to apply different similarity measures for joining clusters on higher levels of abstraction than those used for building basic clusters from original data samples. For example a Euclidean distance can be used as a criterion for building basic clusters. Afterwards the higher levels of abstraction can be formed using rather statistical then geometrical properties, e.g. similar size and variance or population. The criterion used during this implementation to declare two clusters similar was the shortest distance between them – an analogy of the nearest neighbor algorithm. It is virtually impossible to conduct the nearest neighbor calculations for the large number of objects in the original data set. But as soon as the number of treated objects is reduced to the number of freshly created basic clusters, this easy and effective method can be successfully applied.

5.5 Incorporating the project into the Simvis software

The implementation of the research presented in this thesis was developed as a part of the Simvis software, which is a visualization workbench used and developed by VRVis Research Center for Virtual Reality and Visualization, Vienna. The core idea of the Simvis software is to perform professional visualization on a consumer hardware [12]. The system offers several visualization methods: 2D and 3D scatterplots and histograms, volumetric visualization and parallel coordinates. The system provides linking between these views and an interactive way to specify the areas of interest in complex multivariate data [38]. Moreover the system is capable of combining the selections and managing them using its Feature Definition Language (FDL).

The system is focused on visualization of flow and industrial data usually of a simulation origin. It is stores it in its own data format – the High Performance Unstructured Mesh (HUM). The structures presented in this thesis along with the algorithms for their management were designed to be easily transferred to different infovis environments and to access the data remotely in order to maintain the consistency with other views. Therefore a new layer was inserted that intermediates between the HUM layer and SELDA. This layer provides an on-the-fly normalization of the data for the purposes of computing distances and building clusters.

The implementation of parallel coordinates works in an OpenGL-accelerated window inside a Java graphical user interface (Figure 5.12). This interface is used to perform basic interaction with the parallel coordinates as well as with the different rendering modes and abstraction parameters. The computations themselves are provided by a C++ code compiled in the Win32 native environment.



Datasource: Flow:Velocity.U

Axis Operations

Scaling policy
● Scale to slab
○ Scale global

Current timeslab: [0, 0]

Draw Parameters
□ Draw whole
☑ Draw SELDA
☑ Draw texture
☑ Order Areas
☑ Alpha blend...

Alpha mapping policy
○ no policy
○ population
● density (overall)
○ density (axis local)

Abstraction type

Figure 5.12: The GUI of the parallel coordinates inside the Simvis environment

Chapter 6

Results

Conventional methods for information visualization experience difficulties when trying to display large data. The improvement gained by visual abstraction can be observed in using the implementation of the method described in Section 5. Five data sets were tested all of which contain flow data and include more than ten thousand samples each with a dimensionality above fourteen.

filename	samples	dimensions	processing time	response time 1	response time 2
03_test.hum	17,100	15	40sec	4 sec	<0.1 sec
mixed_test.hum	10,260	16	25sec	3 sec	<0.1 sec
blood.hum	162,300	15	6min 20sec	12 sec	<0.1 sec
classic_box.hum	20,000	17	45sec	6 sec	<0.1 sec
bomb_test.hum	5,500	24	50sec	1 sec	<0.1 sec

Data sets of such a volume are not rare today and it is likely that they already belong to the smaller ones compared to huge data produced by current physical simulation or a census. Even with the data set being far below the volume of the largest data sets, the conventional parallel coordinates fail to render a fast reacting display and the clutter in many areas makes the visual exploration much harder. Significant improvement in large data visualization was achieved using visual abstraction in parallel coordinates. For the price of relatively cheap preprocessing (thousands of samples organized into clusters in less then a minute in four of five cases) the display gets much clearer and it is easier to perceive.

6.1 Speed improvement

The interaction with such a display is many times faster than with traditional infovis displays and the graphical hardware demands do not increase with increasing data, since they only depend on the number of clusters created not on the number of individual samples. The difference can be seen in the table. Response time 1 refers to

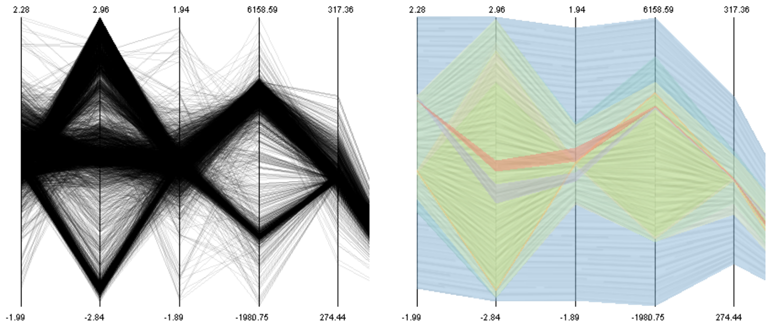


Figure 6.1: Occlusion: What appeared as a homogenous region in the middle part of the traditional parallel coordinates (left) reveals two different groups in abstract view (right).

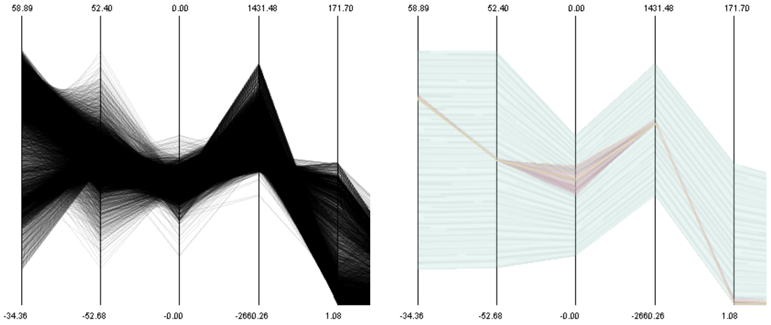


Figure 6.2: Aggregation: It is now clear to see (right) where the most of the data lies and that the rest is less populated.

the the display without visual abstraction, Response time 2 to the display with visual abstraction. The second display reacts interactively within a fraction of a second.

6.2 Occlusion and aggregation problems solved

The other situations that are common for large data visualization are improved as well. Unlike the original dense display full of polylines, the abstract display offers a much more comprehensible view. Using transparency and texturing together with the abstract representation of the samples reduces the clutter and minimizes the chances for occlusion or aggregation problems to occur.

As can be seen in Figure 6.1 (left) the usual parallel coordinates display the area as a dense bunch of various lines. After replacing them by an abstract representation, the underlying structures are uncovered, revealing a group of data that could not be seen

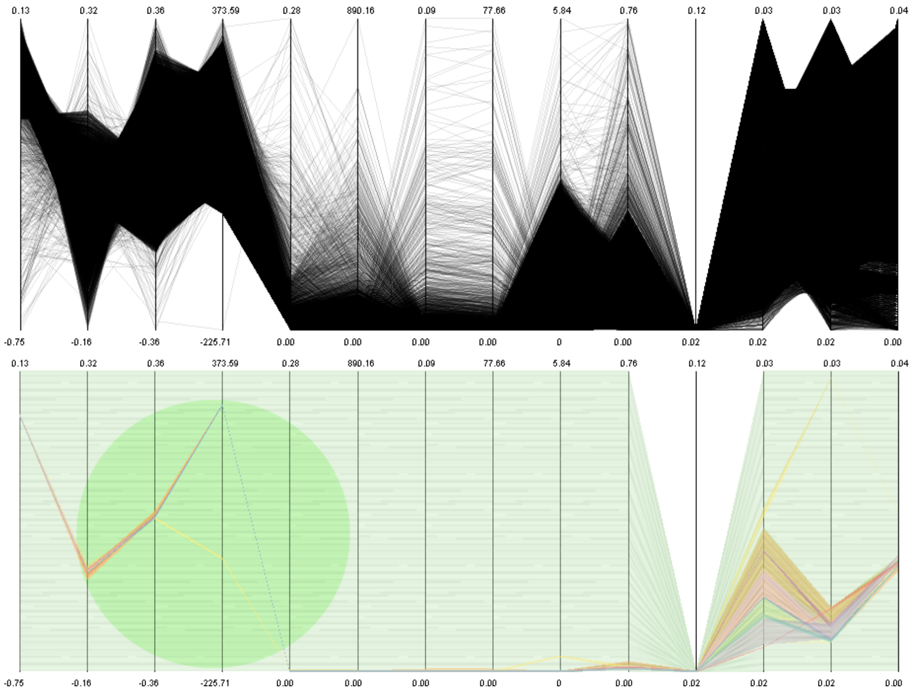


Figure 6.3: Most of the data in the blood data set was found to be focused in a tight cluster, which surprisingly splits into two interesting branches. (The picture was enhanced for printing purposes). This data set is an excellent example how large data might cause a bad interpretation of visualization. Even very dense structures that would be expected to appear in the abstract view were assigned to the 'background' when compared with the highly populated and very dense 'main' cluster.

before (Figure 6.1 (right)). Actually two separate groups (the red and the blue stripe) are discovered in the middle area in a place that was considered to be homogenous in regular parallel coordinates display. A similar improvement is obvious on the display in Figure 6.2. The original view gives the impression of a homogenous area with a relatively equal distribution of the samples. But after a clustering of the data was performed, it shows up that the majority of the data resides in a small area around a specific value with rest being probably just a dense noise (Figure 6.2 (right) or Figure 6.3).

An obvious difference in the visible shapes can be seen in Figure 6.1 and in other images as well. The structures that were visible in former visualization sometimes stand back and new structures emerge, some samples that appeared as separate from a more dense structures are now inside the same cluster and vice versa – some structures that appeared to be compact are split into two or more clusters. To explain these changes, it is necessary to realize the high number of dimensions that contribute to the final result of the clustering. A compact structure in one subset of dimensions can be a sparse and wide-spreading cloud of data in another subset. Or samples that appear to be separate can actually be closer than expected when all the dimension are taken into account.

Differences can be observed also among the abstract displays of the same data. The factor that causes different results of the clustering is the clustering itself with its parameters. The modified k-means algorithm used in this thesis starts with a random assignment and another randomness affects its progress. This, together with the different values of the k parameter, might cause different results. But as the test cases show, the differences between the various results on the same data set are only minor and the core nature of the data is the same in all of them (Figure 6.4).

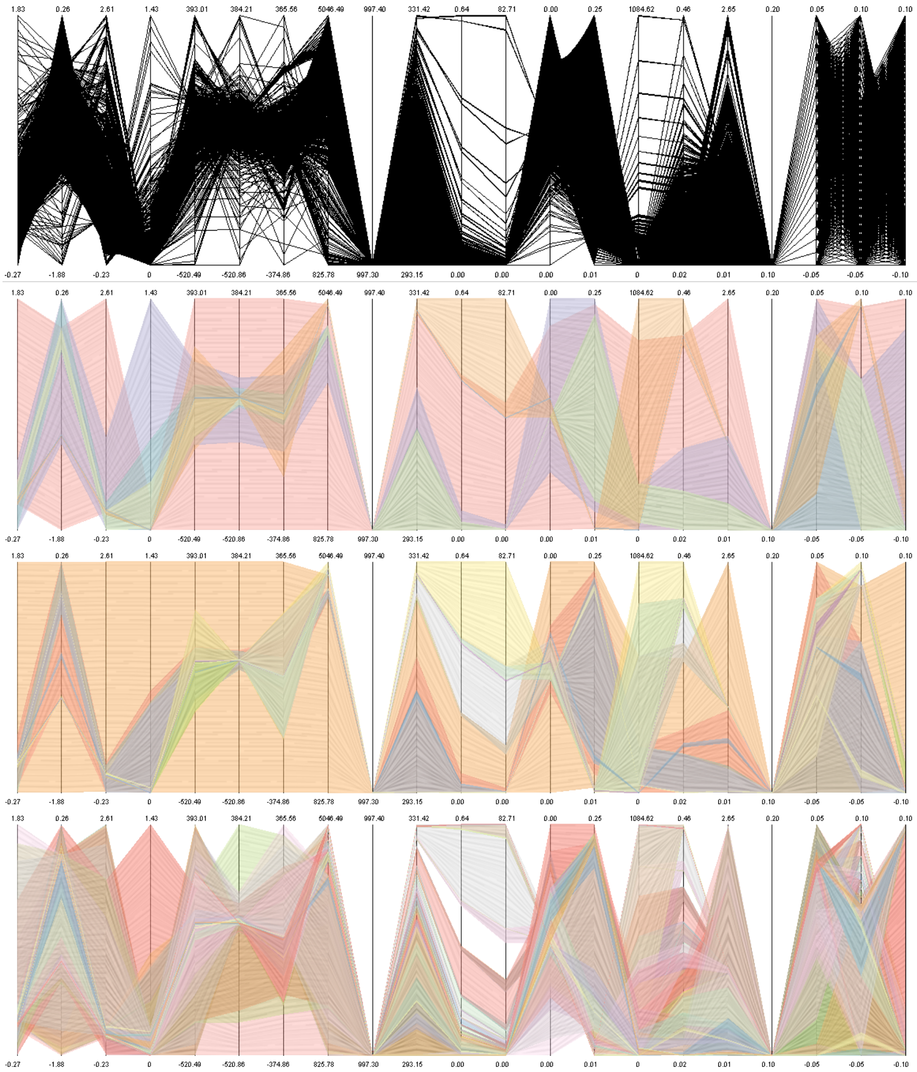


Figure 6.4: Various k-means results on the same data set, from top to bottom – original data set, k=6, k=12, k=64

Chapter 7

Conclusions and future work

As the results might imply, the cooperation between human and machine promises a successful improvement towards visually effective information visualization for large data. Though support provided by automatic data mining is a valuable contribution most of the algorithms producing sufficient results are not optimized for handling large data. Heuristic methods and other modifications are worth considering. Possible ways to optimize the current state can lead through modifying the used k-means algorithm or incorporating the approach presented in [49]. Additional research is necessary to be able to define an ideal abstraction method for large data.

Another interesting idea is to focus more on an efficient data reduction than a computationally complex data mining. The data mining is a stronger mechanism and produces more abstract results, but it is also very complicated and computationally complex compared to a simple data reduction. The shift towards data reduction can be achieved by tweaking the contemporary clustering algorithms (e.g. setting the k parameter of the k-means algorithm to an especially high value) or by implementing other data reduction methods such as the vector quantization. The clusters obtained that way can be relatively small and might be found useful for further abstraction or visualization, since they discard only little information but they reduce the number of the data to be visualized by a significant factor. The concept presented here can easily be used to take advantage of almost every data mining method and it can also be extended to other visualization methods such as scatterplots, which in addition turn out to be a valuable partners to cooperate with parallel coordinates during the visual data exploration.

The results described in the previous chapters show how effective the information visualization techniques can be for large data. Replacing the plain data with an abstract information gained from aggregating similar samples naturally produces a less precise representation, but opens new views on the data revealing structures that were not seen before.

Bibliography

[1] Parallel coordinate explorer: http://www.cs.uta.fi/˜hs/pce.

[2] Maneesh Agrawala and Chris Stolte. Rendering effective route maps: Improving usability through generalization. In *(SIGGRAPH) 2001, Computer Graphics Proceedings*, pages 241–250. ACM Press / ACM SIGGRAPH, 2001.

[3] T. Asano, B. Bhattacharya, M. Keil, and F. Yao. Clustering algorithms based on minimum and maximum spanning trees. In *Proceedings of the fourth annual symposium on Computational geometry*, pages 252–257. ACM Press, 1988.

[4] Barry G. Becker. Volume rendering for relational data. In *IEEE Symposium on Information Visualization (InfoVis '97)*, pages 87–91, Washington - Brussels - Tokyo, October 1997. IEEE.

[5] Cynthia A. Brewer. The color brewer: http://www.personal.psu.edu/faculty/c/a/cab38/colorbrewerbeta2.html.

[6] Cynthia A. Brewer. Color use guidelines for mapping and visualization. *Visualization in Modern Cartography*, pages 123–148, 1994.

[7] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman, editors. *Readings in information visualization: Using vision to think*. Morgan Kaufmann Publishers, San Francisco, 1999.

[8] John Chambers, William Cleveland, Beat Kleiner, and Paul Tukey. *Graphical Methods for Data Analysis*. Wadsworth, 1983.

[9] H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68:361–68, 1973.

[10] William S. Cleveland. *The Elements of Graphing Data*. Wadsworth Inc, 1985.

[11] Cassidy J. Curtis, Sean E. Anderson, Joshua E. Seims, Kurt W. Fleischer, and David H. Salesin. Computer-generated watercolor. *Computer Graphics*, 31(Annual Conference Series):421–430, 1997.

[12] Helmut Doleisch, Martin Gasser, and Helwig Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *Proceedings of the symposium on Data visualisation 2003*, pages 239–248. Eurographics Association, 2003.

[13] W. Eddy, A. Mockus, and S. Oue. Approximate single linkage cluster analysis of large datasets in high dimensional spaces. *High Dimensional Spaces. Comp. Stat. and Data Analysis*, 28:29–43, 1996.

[14] S. Eick and G. Wills. High interaction graphics, 1995.

[15] Jason Eisner. State-of-the-art algorithms for minimum spanning trees. Technical report, University of Pennsylvania, 1997.

[16] E.Wegman and Q. Luo. High dimensional clustering using parallel coordinates and the grand tour. *Computing Science and Statistics*, 28:361–8, 1997.

[17] S. Feiner and C. Beshers. Worlds within worlds: metaphors for exploring n-dimensional virtual worlds. In ACM, editor, *Third Annual Symposium on User Interface Software and Technology UIST*, pages 76–83, New York, NY 10036, USA, October 1990. ACM Press.

[18] Edward Forgy. Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications,. *Biometrics*, 21, 1965.

[19] Ying-Huey Fua, Matthew O. Ward, and Elke A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In David Ebert, Markus Gross, and Bernd Hamann, editors, *IEEE Visualization '99*, pages 43–50, San Francisco, 1999. IEEE.

[20] Ying-Huey Fua, Matthew O. Ward, and Elke A. Rundensteiner. Navigating hierarchies with structure-based brushes. In *INFOVIS*, pages 58–64, 1999.

[21] Keinosuke Fukunaga. *Statistical Pattern Recognition*. Academic Press, San Diego, 2nd edition, 1990.

[22] Gardner. Mathematical games: John horton conway's book covers an infinity of games. *SCIAM: Scientific American*, 1976.

[23] Markus Hadwiger, Christoph Berger, and Helwig Hauser. High-quality two-level volume rendering of segmented data sets on consumer graphics hardware. In *Proceedings of IEEE Visualization*, 2003.

[24] A. Inselberg. Multidimensional detective. In *IEEE Symposium on Information Visualization (InfoVis '97)*, pages 100–107, Washington - Brussels - Tokyo, October 1997. IEEE.

[25] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multidimensional geometry. In *IEEE Visualization '90 Proceedings*, pages 361–378. IEEE Computer Society, October 1990.

[26] Alfred Inselberg, Tuval Chomut, and Mordechai Reif. Convexity algorithms in parallel coordinates. *Journal of the ACM*, 34(4):765–801, October 1987.

[27] Samuel Kaski. Data exploration using self-organizing maps. *Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No. 82*, March 1997.

[28] Daneil A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8, 2002.

[29] Daniel A. Keim. Visual exploration of large data sets. *Communications of the ACM (CACM)*, 44(8):38–44, 2001.

[30] Daniel A. Keim, Ming C Hao, Umesh Dayal, and Meichun Hsu. Pixel bar charts: a visualization technique for very large multi-attribute data sets. *Information Visualization*, 1(1):20–34, 2002.

[31] Daniel A. Keim and Annemarie Herrmann. The gridfit approach: An efficient and effective approach to visualizing large amounts of spatial data,. In *IEEE Visualization '98*, pages 181–188. IEEE, 1998.

[32] Allison W. Klein, Wilmot W. Li, Michael M. Kazhdan, Wagner T. Correa, Adam Finkelstein, and Thomas A. Funkhouser. Non-photorealistic virtual environments. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 527–534. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[33] Teuvo Kohonen. *Self organizing maps*. Springer, New York, 2000.

[34] Robert Kosara, Gerald N. Sahling, and Helwig Hauser. Linking scientific and information visualization with interactive 3d scatterplots. In *Short Communication Papers Proceedings of the 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, pages 133–140, 2004.

[35] M. Kreuseler, N. López, and H. Schumann. A scalable framework for information visualization. In *2000 IEEE Symposium on Information Visualization (InfoVis '00)*, pages 27–38. IEEE, 2000.

[36] J. Kruskal and M. Wish. *Multidimensional Scaling*. Sage Publications, 1978.

[37] Cynthia L. Lach. Effect of temperature and gap opening rate on the resiliency of candidate solid rocket booster O-ring materials. Technical Report NASA TP-3226, NASA - Langley Research Centre.

[38] Cynthia L. Lach. Interactive feature specification for focus+context visualization of complex simulation data. Technical Report VRVis-2002-047, VRVis Research Center.

[39] John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proc. ACM Conf. Human Factors in Computing Systems, CHI*, pages 401–408. ACM, 1995.

[40] J. LeBlanc, Matthew O. Ward, and N. Wittels. Exploring n-dimensional databases. In *Proceedings of the Conference on Visualization 2000*, pages 230–239. IEEE Computer Society Press, 1990.

[41] Clustan Ltd. K-means cluster analysis: http://www.clustan.com.

[42] David Luebke, Benjamin Watson, Jonathan D. Cohen, Martin Reddy, and Amitabh Varshney. *Level of Detail for 3D Graphics*. Elsevier Science Inc., 2002.

[43] Frits H. Post, Benjamin Vrolijk, Helwig Hauser, Robert S. Laramee, and Helmut Doleisch. The state of the art in flow visualization: Feature extraction and tracking. *Computer Graphics Forum*, 22(4):775–792, 2003.

[44] Aaron J. Quigley. Large scale 3d clustering and abstraction. In *Selected papers from the Pan-Sydney workshop on Visualisation*, pages 117–118. Australian Computer Society, Inc., 2001.

[45] Shashi Shekhar, Chang-Tien Lu, Sanjay Chawla, and Pusheng Zhang. Data mining and visualization of twin-cities traffic data. In *Technical Report TR 01-015, Dept. of CSE, Univ. of Minnesota*.

[46] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. IEEE Visual Languages 1996*, 1996.

[47] Harri Siirtola. Direct manipulation of parallel coordinates. In *Proc. of IEEE Conf. on Info. Vis.'00*, 2000.

[48] Serengul Smith. Machine learning techniques: http://www.cs.mdx.ac.uk/staffpages/serengul/clustering.htm.

[49] Jeremy Tantrum, Alejandro Murua, and Werner Stuetzle. Hierarchical model-based clustering of large datasets through fractionation and refractionation. In David Hand, Daniel Keim, and Raymond Ng, editors, *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-02)*, pages 183–190, New York, July 23–26 2002. ACM Press.

[50] Holger Theisel. Higher order parallel coordinates. In *Proc. Vision, Modeling and Visualization 2000*, pages 119–125, 2000.

[51] Edward R. Tufte. *Envisioning Information*. Graphics Press, 1990.

[52] Edward R. Tufte and Dmitry Krasny. *Visual Explanations : Images and Quantities, Evidence and Narrative*. Graphics Press, 1997.

[53] Michelle Q. Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the working conference on Advanced visual interfaces*, pages 110–119. ACM Press, 2000.