♣ *Visualization & Data Mining for High Dimensional Datasets*

⋆ ⋆ ⋆ *Tutorial* ⋆ ⋆ ⋆

SIAM Conference on Data Mining

Phoenix, Arizona
April 28, 2011

Alfred Inselberg[1]
School of Mathematical Sciences
Tel Aviv University
Tel Aviv, Israel
aiisreal@math.tau.ac.il ∗ **www.math.tau.ac.il/ ˜ aiisreal**

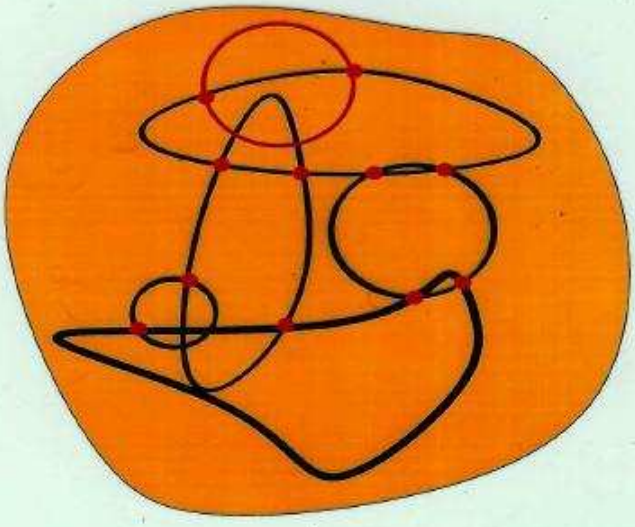
♣

[1]Senior Fellow San Diego SuperComputing Center & National University of Singapore

Preview

VISUALIZATION

Insight through *Images*



*C*ollection of application dependent *Mappings* :

$$Problem\ Domain \rightarrow\ VisualRange$$

Involves: *G*EOMETRY, *C*OGNITION, *A*RT, ?

Goal: *V*isual Model to help our *Intuition*

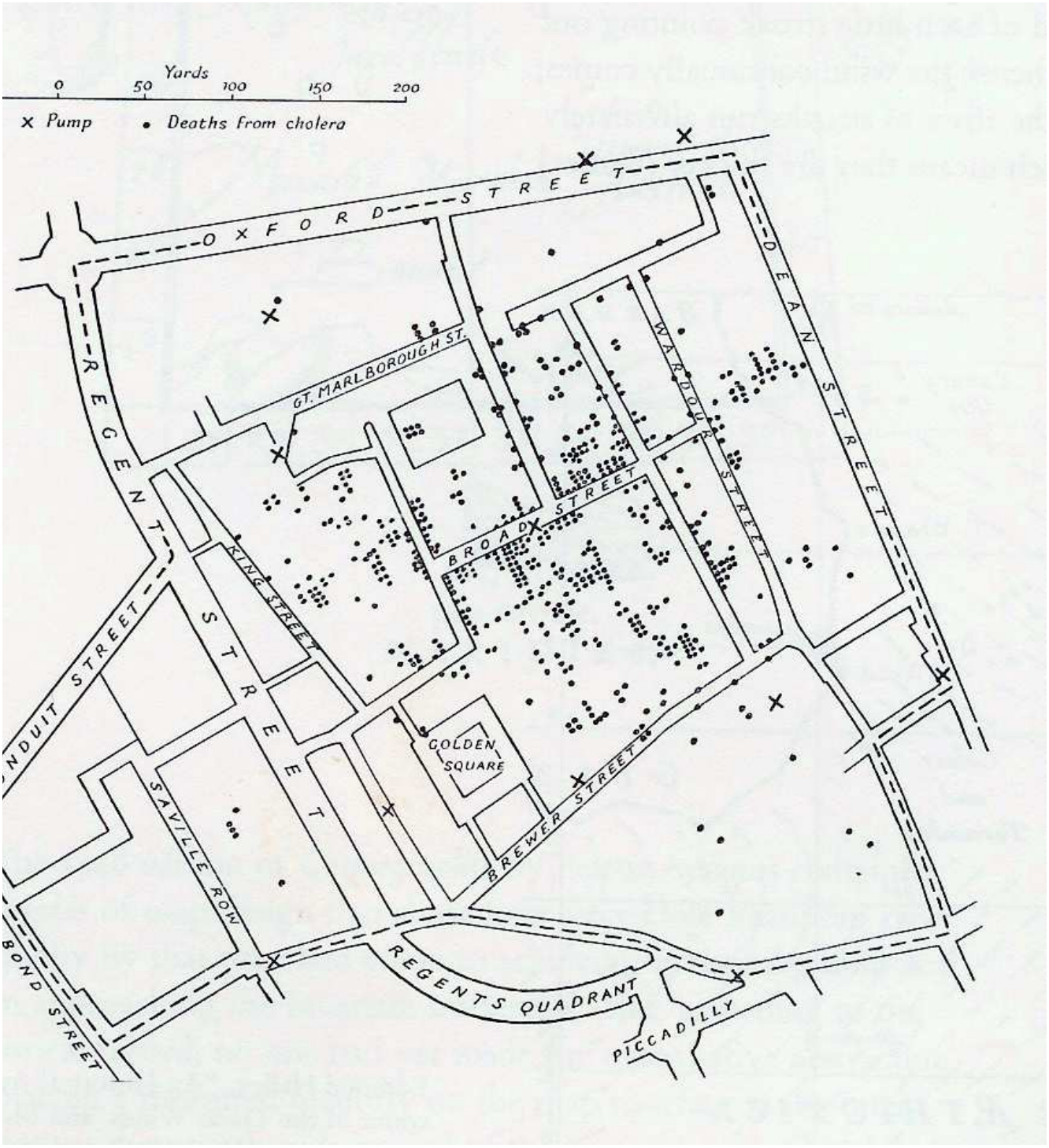


Figure 1: Cholera epidemic in London 1854. Dr. Snow placed dots at the addresses of the deceased and saw the concentration of deaths around the Broad street water pump. From E.W.Gilbert, Geog. J. 124] (1958) – By permission from E.R.Tufte " The Visual Display of Quantitative Information", Graphic Press 1983 p. 24

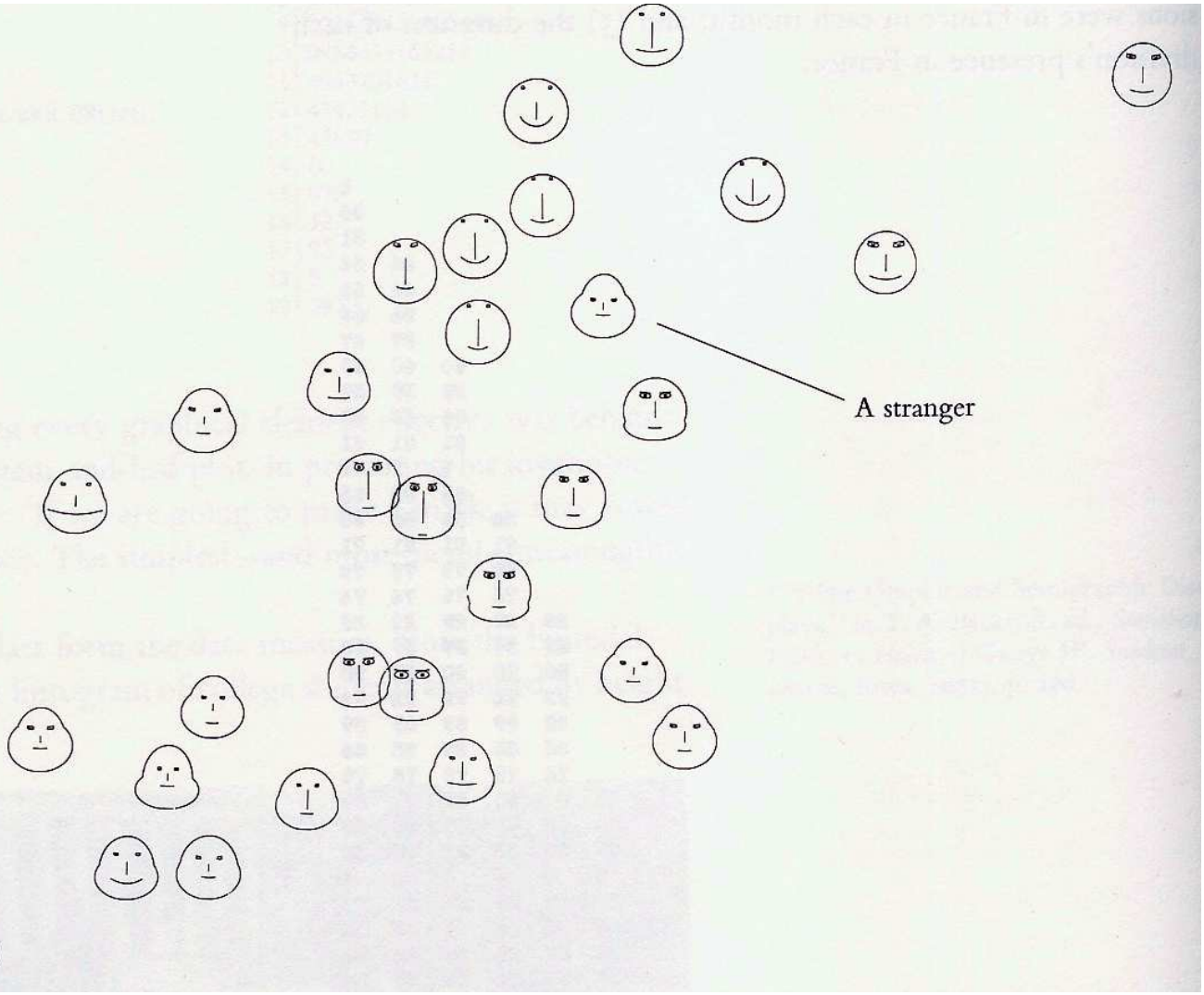


Figure 2: Data mapped into faces; each parameter corresponds to a facial feature. H. Chernoff, JASA 68 (1973)

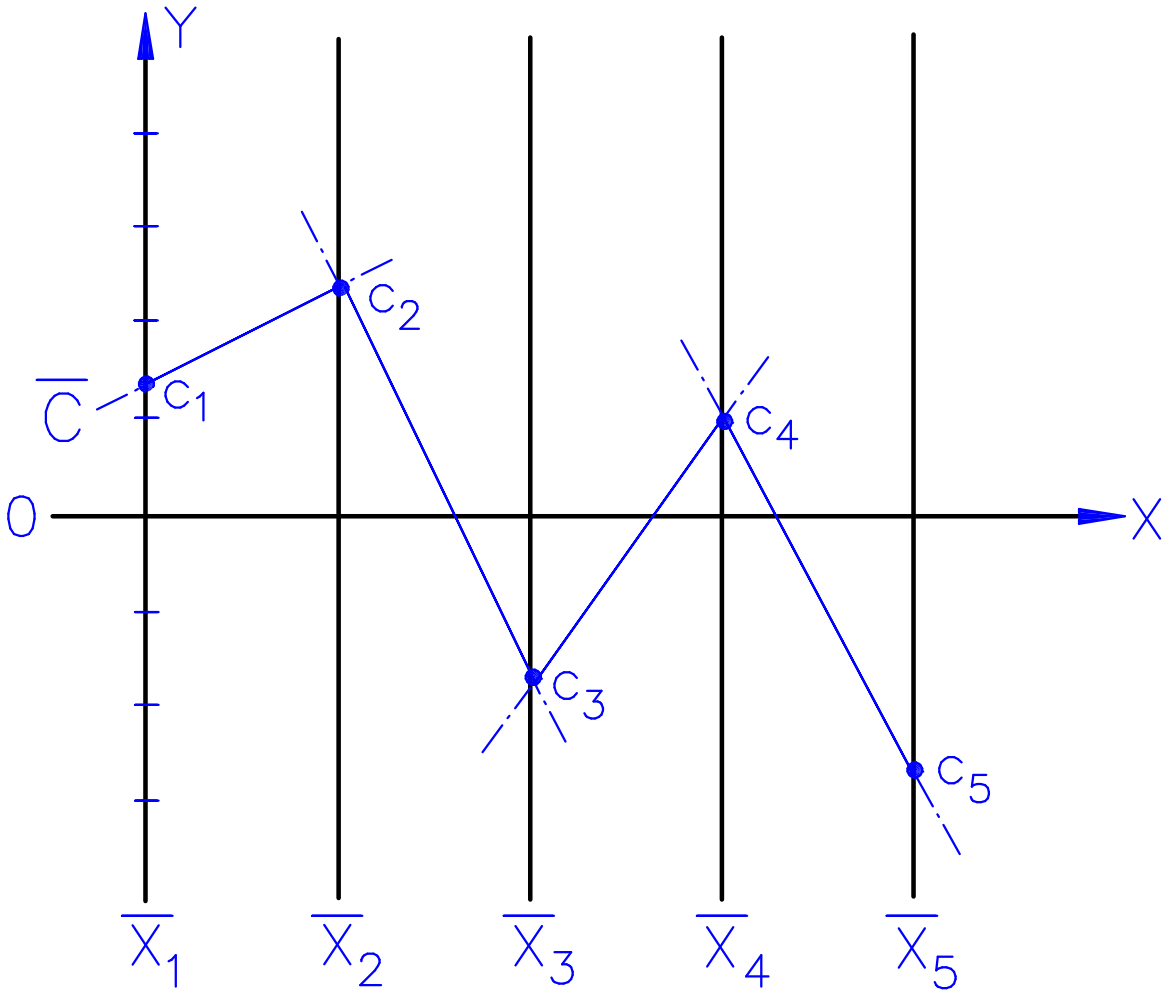


Figure 3: Parallel Coordinates – Point in 5-D , $C = (c_1, c_2, c_3, c_4, c_5)$

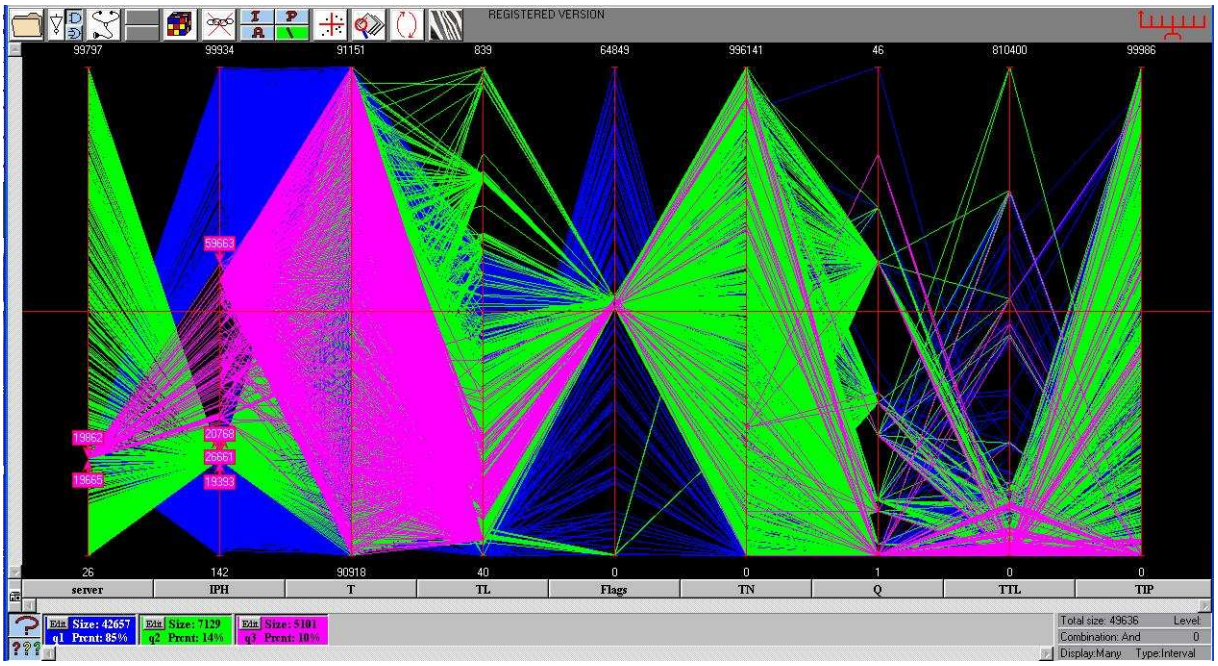


Figure 4: Detecting Network Intrusion from Internet Traffic Flow Data. Note the many-to-one relations, how many can you spot?

Information Visualization & Data Mining

A large collection of methodologies tracing the development of the field can be found in [14].

Introduction

The first, and still more popular application, of parallel coordinates is in exploratory data analysis (EDA); discovering data subsets (relations) satisfying given objectives. A dataset with M items has 2^M subsets anyone of which may be the one we really want. With a good data display our fantastic pattern-recognition ability can cut great swaths searching through this combinatorial explosion and also extract insights from the visual patterns. These are the core reasons for data visualization. With parallel coordinates (abbr.||-coords) the search for relations in multivariate datasets is transformed into a 2-D pattern recognition problem. Guidelines and strategies for knowledge discovery are illustrated on several real datasets one with hundreds of variables. A geometric classification algorithm is presented and applied to complex datasets. It has low computational complexity providing the classification rule explicitly and *visually*. The minimal set of variables required to state the rule is found and ordered by their predictive value. Multivariate relations can be modeled as hypersurfaces and used for decision support. A model of a country's economy reveals sensitivies, impact of constraints, trade-offs and sectors unknowingly competing for the same resources. Foundational background is provided where needed. Collision avoidance algorithms for air traffic control are discussed separately in the section on multidimensional lines.

Many researchers contributed to the development and applications of parallel coordinates (in alphabetical order). The Andrienkos [1], J. Dykes et al. [7], R. Edsall [8] and A. MacEachren et al. [6] introduced ||-coords to GIS and geovisualization; Kim Esbensen contributed to data analysis and dualities, H.Carter, C.Gennings, and K. Dawson on response surfaces in statistics [16], Amit Goel [17] on Aircraft Design, Chris Jones on optimization [29]; John Helly [20] on early application to data analysis; Hans Hinterberger [39] contributions in comparative visualization and data density analysis; Matt Ward et al. introduced hierarchical parallel coordinates [15] and added much to the field [45]; Helwig Hauser's innovative contributions included parallel sets for categorical data [19]; Antony Unwin et al. made wide-ranging numerous contributions [44], Li Yang [46] applied ||-coords to the visualization of association rules; H. Choi and Heejo Lee [4] important contribution and P. Hertzog [21] on detecting network intrusion, G. Conti [5] produced a ground-breaking book on security data visualization; and there are exciting recent works by H. Ye and Z.Lin's [47] with a novel contribution to optimization (simulated annealing), T. Kipouros et al. [30] with a sophisticated optimization for turbomachinery design [30], S. El Medjani et al. [10] proposed a novel application using ||-coords as straight-line detectors, R. Rosenbaum and H. Schumann [37] on progression in visualization, F. Rossi [38] on visual data mining and machine learning, Huamin Qu et al,

[22] on air pollution analyis and clustering, M. Tory et al.. on ||-coords interfaces [40], J. Johanson et al. [27], G. Ellis and A. Dix [11] on clustering and clutter reduction, H. Siirtola and K.J.Räihä on interaction with ||-coords, B. Pham, Y. Cai and R. Brown on traditional Chinese medicine [35] and there are proposals to enhance ||-coordsd by using curves [32], place them in 3-D [34], [28], or modify them as starplots [12] and C.B. Hurley and R.W. Olford contributed the definitive study on axes permutations [**?**]. The list is by no means exhaustive.

Origins

For the visualization of multivariate problems numerous mappings encoding multidimensional information visually into 2-D or 3-D (see [14] and [41], [42], [43]) have been invented to augment our perception, which is limited by our 3-dimensional habitation. Wonderful successes like Minard's "Napoleon's March to Moscow", Snow's "dot map" and others are *ad hoc* (i.e. one-of-a-kind) and exceptional. Succinct multivariate relations are rarely apparent from **static** displays; **interactivity** is essential. In turn, this raises the issues of effective *GUI* – Graphic User Interface, queries, exploration strategies and information preserving displays.

The case for Visualization

$\mathscr{S}$earching a dataset with M items for interesting, depending on the objectives, properties is inherently hard. There are 2^M possible subsets anyone of which may satisfy the objectives. The *visual cues*, our eyes can pick from a good data display, help navigate through this combinatorial explosion. How this is done is part of the story. Clearly, if the transformation : *data* $\rightarrow$ *picture* clobbers information a great deal is lost right at the start. We postulate that a display of datasets with N variables suitable for *exploration* satisfy the following requirements:

1. **should preserve information** – the dataset can be completely reconstructed from the picture,
2. **has low representational complexity** – the computational cost of constructing the display is low,
3. **works for any N** – not limited by the dimension,
4. **treats every variable uniformly**,
5. **reveals multivariate relations in the dataset** – the most important and controversial single criterion,
6. **is based on a rigorous mathematical and algorithmic methodology** – to eliminate ambiguity in the results. Also dataset an be recognized after rotations, translations, scalings and perspective transformations.

These and additional issues comprising the discovery process are better appreciated via the exploration of real datasets. The basic queries are introduced with an example of satellite Subsequently, they are combined with boolean operators to form complex queries applied to financial data. An example with several hundred variables is discussed briefly before moving to automatic classification. Visualization and ||-coords play key roles in the geometric algorithm's conception, internal function and visual presentation

of the classification rule. The minimal set of the variables needed to state the rule is found and ordered according to their predictive value.

Mathematical background is interspersed to provide a deeper understanding and wiser use of ||-coords and its applications. Specifically,

1. learning the *patterns* correspoding to the basic relations and seek them out for EDA,
2. understanding the design and use of the *querries*,
3. motivating further sophisticated applications to Statistics like *Response Surfaces* [16], and
4. understanding that the relational information resides in the *crossings*,
5. **concentrating the relational information in the data in clear patterns eliminating the polygonal lines altogether** as with the "proximate planes" is feasible. encouraging research on efficient (parallel) algorithms for accomplishing this on general datasets. Hence eliminating the "clutter" by reducing the display into patterns corresponding, at least approximately, to the multivariate existing in the data. .

Before entering the nitty gritties we pose a visualization challenge which we ask the reader to ponder. For a plane

$$\pi : c_1x_1 + c_2x_2 + c_3x_3 = c_0 \ , \tag{1}$$

allow the coefficients to vary each within a small interval. This generates a family of "close" let's call them *proximate*) planes :

$$\Pi = \{\pi : c_1x_1 + c_2x_2 + c_3x_3 = c_0, \quad c_i \in [c_i^-, c_i^+], \quad i = 0,1,2,3\} \ . \tag{2}$$

These are the planes generated by small rotations and translations of π with respect to the 3 coordinates axes. Altogether they form a "twisted slab" which even in 3-D with *orthogonal axes* is difficult to visualize. Conversely given lots of points in 3-D how can it be discovered, using **any** general visual method you like, that they lie on a twisted slab and how such a creature can be visualized and described precisely; for $N = 3$ and then for *any N*?

Exploratory Data Analysis with ||-coords

Multidimensional Detective

$\mathscr{P}$arallel coordinates transform multivariate relations into 2-D patterns suitable for exploration and analysis. For this reason they are included in lots of software tools. The queries "'parallel coordinates + Software" on Google returned about 31,000 "hits" and "Scatterplot matrix + Software" about 15,000. Irrespective of the apparent *2:1* relative ratio, the comparable numbers for the two astounded me having heard the appellations: "esoteric", "unnatural", "difficult", "squiggly" and more for ||-coords after their introduction.

The exploration[1] paradigm is that of a *detective*, starting from the data, searching for clues leading to conjectures, testing, backtracking until *voila* ... the "culprit" is discovered. The task is especially intricate

[1]The venerable name "Exploratory Data Analysis" *EDA* is used interchangeably with the currently more fashionable "Visual Data Mining".

when many variables (i.e. dimensions) are involved calling for the employment of a *multidimensional* detective (abbr. *MD*). As if there were any doubts, our display of choice is ||-coords where the data appears in the by now familiar squiggly blotches and which, by means of *queries*, the *MD* skilfully disects to find precious hidden secrets.

During the ensuing interaction think, dear reader, how similar queries can be done using other exploration methodologies including the ubiquitous spread-sheets. More important, what visual clues are available that would **prompt** use of such queries. This is a good place to point out a few basics. In ||-coords due to the *point* $\leftrightarrow$ *line* and other dualities, some but *not* all actions are best performed in the dual. The queries, which are the "cutting tools", operate on the display i.e.*dual*. Their design should exploit the methodology's strengths and avoid its weaknesses; rather than mimic the action of queries operating on standard "non-dual" displays. As a surgeon's many specialized cutting tools, one of our early software versions had lots of specialized queries. Not only was it hard to classify and remember them but they still could not handle all situations encountered. After experimentation, I opted for a few(3) intuitive queries called **atomic** which can be combined via *boolean* operations to form complex intricate cuts. Even for relatively small datasets the ||-coords display can look uninformative and intimidating. Lack of understanding the basics of the underlying geometry and poor choice of queries limits the use of ||-coords to unrealistically small datasets. Summarizing, the requirements for successful exploratory data analysis are:

- an informative display *without loss of information* of the data,
- good choice of queries, and
- skillful interaction with the display.

An Easy Case Study – Satellite Data

$\mathscr{T}$he first admonition is

- **do not let the picture intimidate you,**

as can easily happen by taking an uninformed look at Fig. 6 showing the dataset to be explored. It consists of over 9,000 measurements with 9 variables, the first two (X,Y) specify the location on the map in Fig. 5(left), a portion of Slovenia, where 7 types of ground emissions are measured by satellite. The ground location, (X,Y), of one data item is shown in Fig. 5 (right), which corresponds to the map's region and remains open during the exploration. The query, shown in Fig.6, used to select the data item is called *Pinch*. It is activated by the button **P** on the tool bar. By means of this query, a bunch of polygonal lines (i.e. data items) can be chosen by being "pinched" *in-between* the axes. The cursor's movement changes the position of the *selected* arrow-head which is the larger of the two shown. In due course various parts of the *GUI* are illustrated(*Parallax*[2]). Aside from the starting the exploration without biases it is essential

- **to understand the objectives.**

Here the task is the detection and location of various ground features (i.e. built-up areas, vegetation, water etc) on the map. There is a prominent lake, on the lower-left corner with an unusual shape the upward pointing "finger". This brings up the next admonition, that no matter how messy it looks

- **carefully scrutinize the data display for clues and patterns.**

Follow up on anything that catches the eyes, gaps, regularities, holes, twists, peaks & valleys, density contrasts like the one at the lower values of $B3$ through $B7$. Using the *Interval* query, activated by the **I** button, starting at the minimum we grab the low range of $B4$ (between the arrowheads) stopping at the dense part as shown in Fig. 7. The result, on the left of Fig. 8, is amazing. Voila we found the water, the lake is clearly visible together with two other regions which in the map turn up to be small streams. Our scrutiny having been rewarded we recall the adage

- **a good thing may be worth repeating.**

Examining for density variations now *within the selected lower interval of* $B4$ we notice another. The lowest part is much denser. Experimenting a bit, appreciating the importance of interactivity, we select the sparse portion, Fig. 9, which defines the water's edge (right) 8 and in fact more. By dropping the lower arrow we see the lake filling up starting from the edge i.e. shallow water first. So the lower values of $B4$ reveal the water and the lowest "measure" the water's depth; not bad for few minutes of playing around. But all this pertains to a single variable when we are supposed to be demonstrating *multivariate* exploration. This is a valid point but we did *pick* $B4$ among several variables. Further, this is a nice "warm-up" for the subsequent more involved examples enabling us to show two of the queries. The astute observer must have already noticed the regularity, the vertical bands, between the $B1, B2$ and $B3$ axes. This is where the *angle* query, activated by the **A** button, comes into play. As the name implies it selects groups of lines within a user-specified angle range. A data subset is selected between the $B2, B3$ axes as shown, with enlarged inter-axes distance better showing the vertical bands, in Fig. 10 (left) to select a data subset which corresponding on the map to regions with high vegetation. Clicking the **A** button and placing the cursor on the middle of one axis opens an angle, with vertex on the mid-range of the previous(left) axis,

[2]MDG's Ltd proprietary software–All Rights Reserved, is used by permission

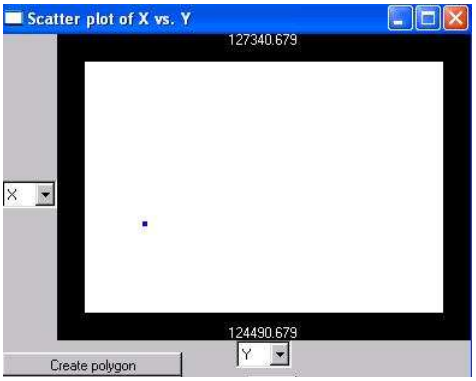


Figure 5: Seven types of ground emissions were measured on this region of Slovenia.

Measurements recorded by the LandSat Thematic Mapper are shown in subsequent figures. Thanks to Dr. Ana Tretjak and Dr. Niko Schlamberger, Statistics Office of Slovenia for providing the data. (Right) The display is the map's rectangular region, the dot marks the position where the 7-tuple shown in the next figure was measured.

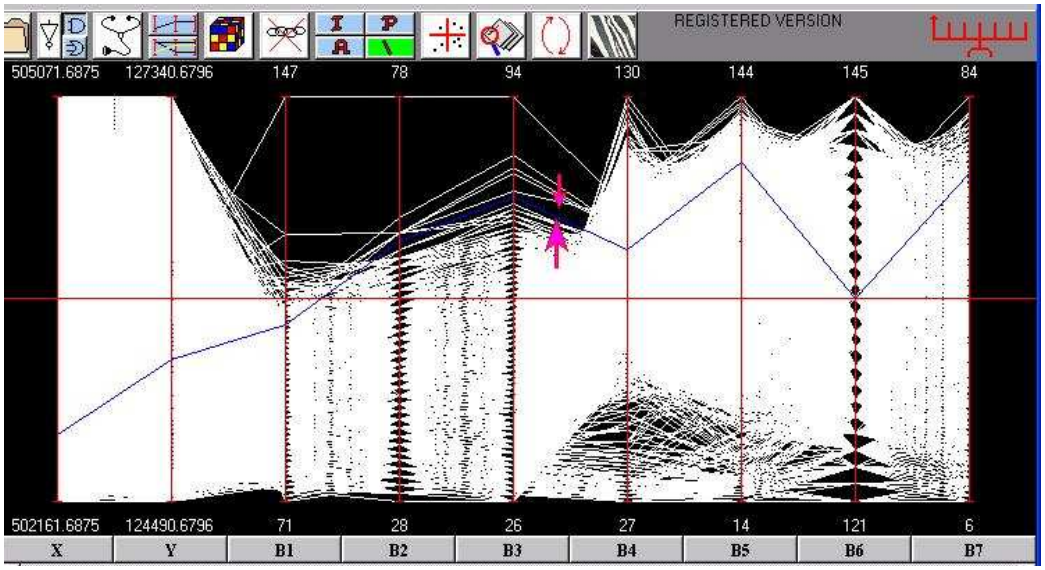


Figure 6: Query on Parallax showing a single data item.

The X, Y (position, also shown on the right of Fig. 5), and values of the 7-tuple $(B1, B2, B3, B4, B5, B6, B7)$ at that point.

whose range is controlled by the arrow movements on the right axis. Actually this "rule" (i.e. relation among some parameters) for finding vegetation can be refined by twicking a couple of more parameters. This raises the topic of rule finding in general, *Classification*, which is taken up in Section .

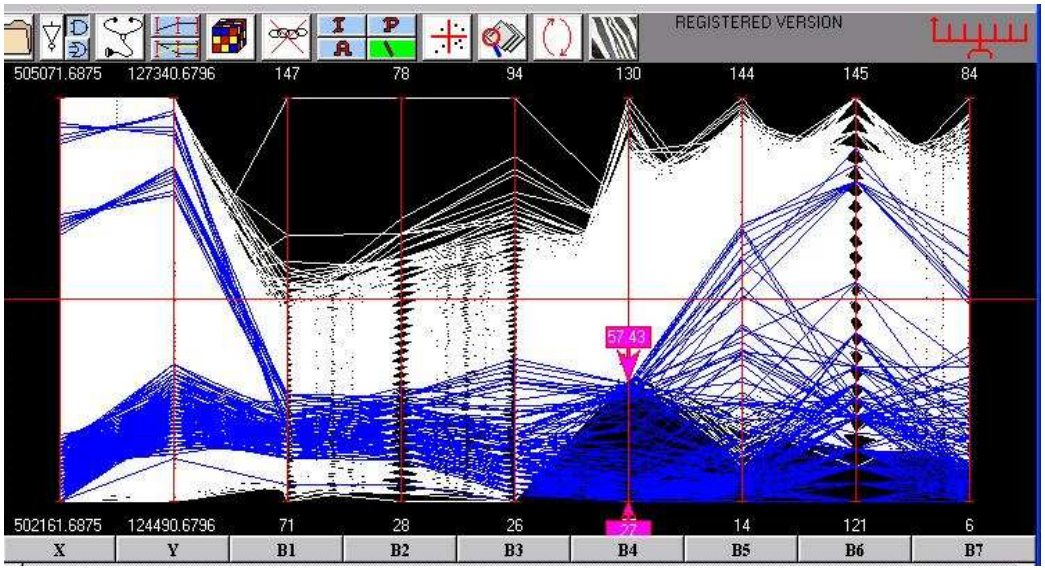


Figure 7: Finding water regions.

The contrast due to density differences around the lower values of $B4$ is the *visual cue* prompts this query.

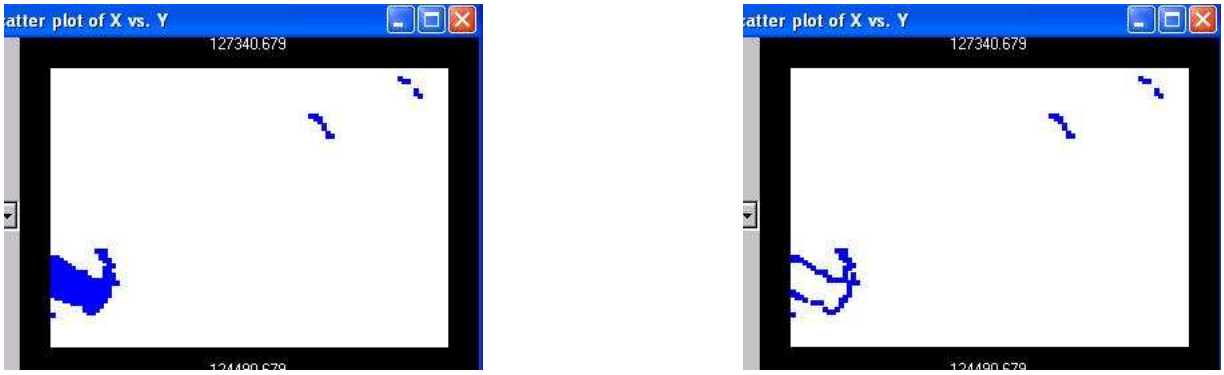


Figure 8: (Left)The lake – result of query shown in Fig. 7.
On the right is just the lake's edge. It is the result of query shown in Fig. 9.

The *angle* and *pinch* queries are motivated by the ℓ *line* $\rightarrow$ *point* $\bar{\ell}$ duality

$$\ell : x_2 = mx_1 + b \leftrightarrow \bar{\ell} = (\frac{d}{1-m}, \frac{b}{1-m}) \tag{3}$$

in ||-coords illustrated in Fig. 11 where the inter-axes distance is d. As seen from its x-coordinate, the point $\bar{\ell}$ lies between the parallel axes when the line's slope $m < 0$, to the right of the $\bar{X_2}$ axis for $0 < m < 1$ and left of $\bar{X_1}$ for $m > 1$. Lines with $m = 1$ are mapped to the *direction* with slope b/d in the on the xy-plane; with d the inter-axes distance and b the constant (intercept) in the equation of ℓ. This points out that dualities properly reside in the *Projective*, the *directions* being the *ideal points*, rather than the Euclidean plane. For sets of points having a "general" direction with negative slope, i.e. are "negatively correlated",

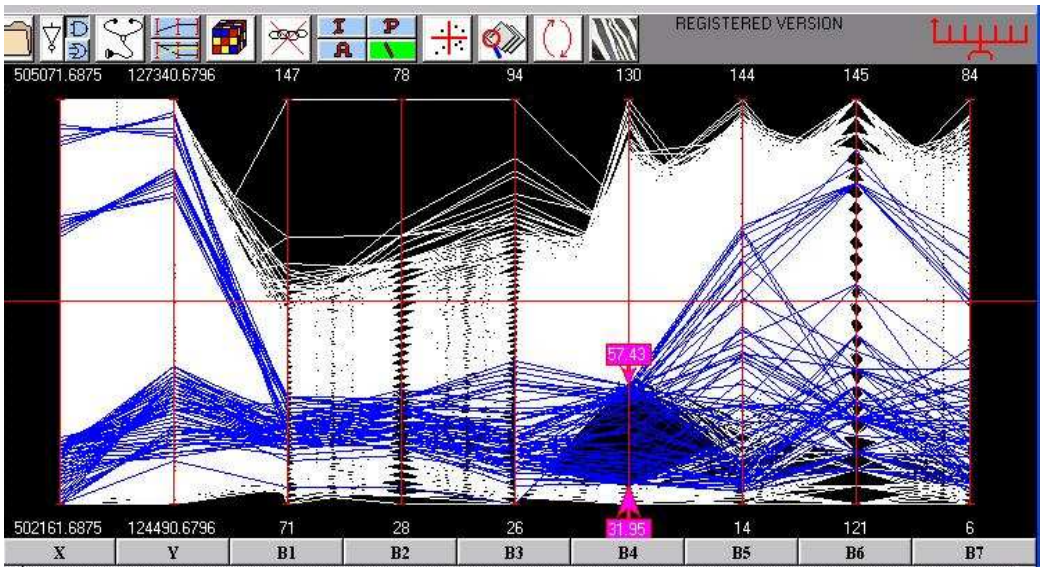


Figure 9: Query finding the water's edge.

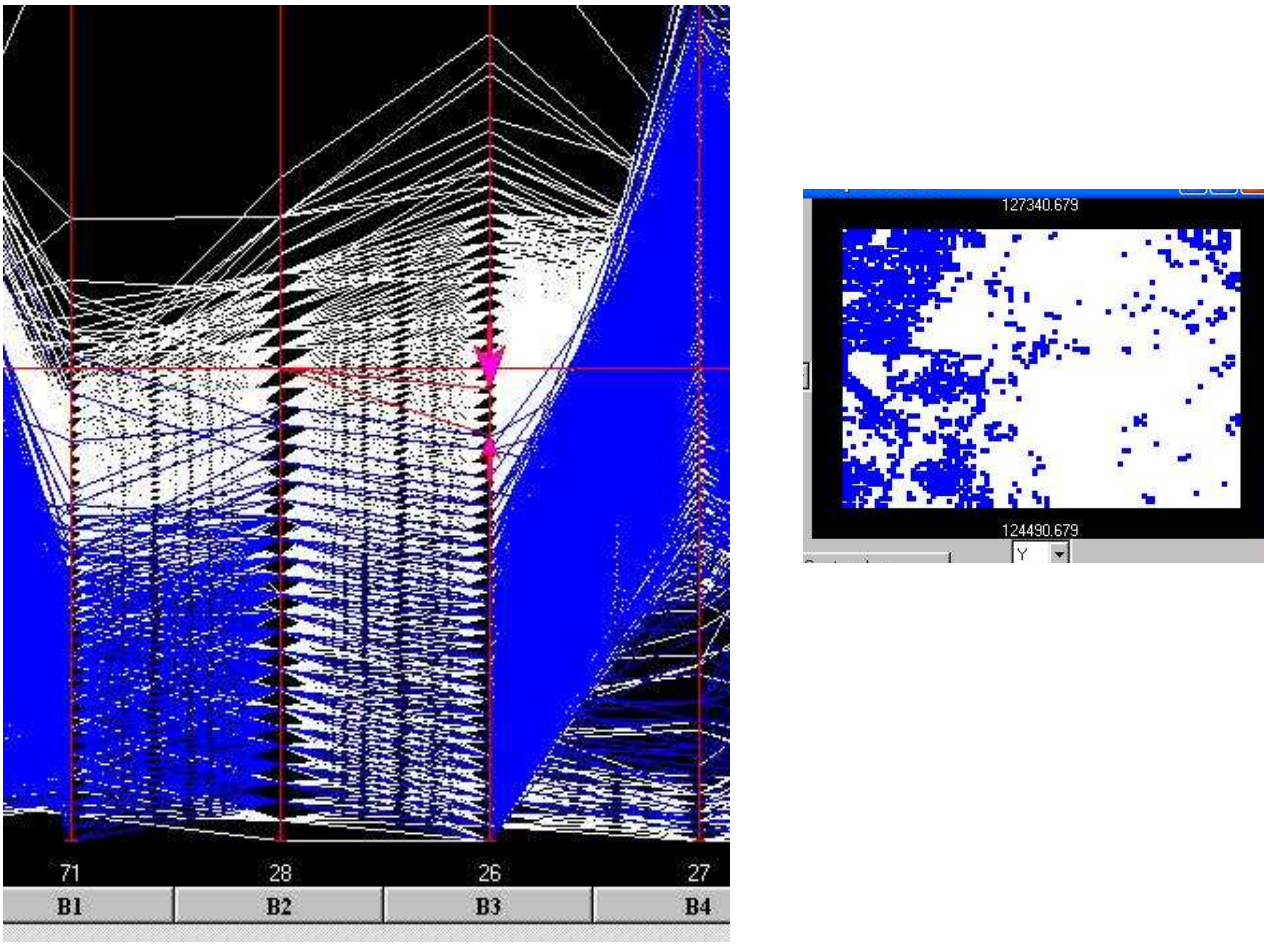


Figure 10: Finding regions with vegetation.

Here the *angle query* is used (left) between $B2, B3$ axes. Note the arrow-heads on the $B3$ axis which specify the angle-range for the selected lines.

the lines representing them in $\|$ cross each other roughly in between the axes and they can be *selected with the pinch query*. For positively correlated sets of points their corresponding lines cross outside the axes and can be *selected with the angle query*. All this exemplifies the need to understand some of the basic geometry so as to work effectively with the queries and of course designing them properly. The three atomic queries having been introduced there remains to learn how they can be combined to construct complex queries.

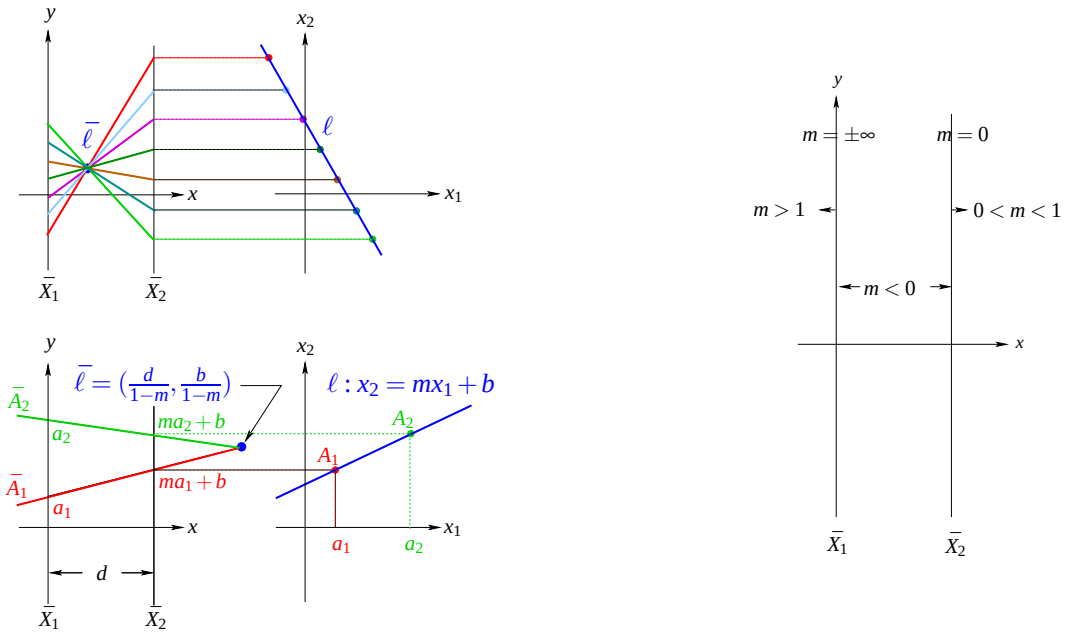


Figure 11: Parallel coordinates induce a *point* $\bar{\ell} \leftrightarrow \ell$ *line* duality (left).
(Right) The horizontal position of the point $\bar{\ell}$ representing the line ℓ is determined only by the line's slope m. The vertical line $\ell : x_1 = a_1$ is represented by the point $\bar{\ell}$ at the value a_1 on the $\bar{X}_1$ axis.

Prior to that, Fig. 10 (left) begs the question: "what if the $B2$ and $B3$ axes were *not* adjacent"? Then the pattern and hence their pairwise relation would be missed. Hence the axes-permutation used for the exploration is important. In particular what is the minimum number of permutations among N-axes containing the *adjacencies* for all pairs of axes? It turns out [**?**]: M permutations are needed for even $N = 2M$ and $M+1$ for odd $N = 2M+1$. It is fun to see why. Label the N vertices of a graph with the index of the variables X_i, $i = 1, \ldots, N$ as shown in Fig. 12 for $N = 6$. An edge joining vertex **i** with **j** signifies that the axes indexed by **i , j** are adjacent. The graph on the left is a *Hamilton path* for it contains all the

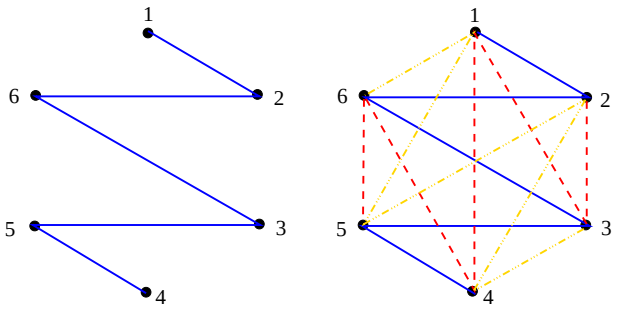


Figure 12: (Left) First Hamiltonian path on vertices **1, . . . , 6**.
It corresponds to the (axes) index permutation **126354**. (Right) The complete graph as the union of the 3 distinct Hamiltonian paths starting successively at the vertices **1, 2, 3**.

vertices. Such paths have been studied starting with Euler in the 18th century with modern applications to the "travelling salesman" problem and elsewhere ([18] pp.66 , [3] pp. 12). The graph corresponds to the axes index permutation **126354**. On the right, the union with the additional two Hamiltonian paths, starting at vertices **2** and **3**, forms the complete graph which contains all possible edges. Hence the 3 permutations **126354 , 231465, 342516** contain all possible adjacent pairs; just try it. The remaining permutations are obtained from the first by successively adding *1 mod 6* , and this works for general N [31]. As of this writing the authoritative reference on axes permutations is by C. Hurley and W. Olford [**?**]. Before leaving this interesting subject we pose the *triad permutation* problem. Namely, what is the minimum number of permuations needed to obtain all possible adjacent *triples* of axes?

Returning to EDA, the icon with the *Rubik's Cube* on *Parallax*'s toolbar activates a *permutation editor* which automatically generates the Hamiltonian permutations (abbr. *HP*). After scrutinizing the dataset display the recommended next step is to run through the $O(N/2)$ *HP*. This is how all nice adjacencies such as the one in Fig. 10 are discovered. Then using the editor, patch your own custom-made permutation containing all the parts you like in the *HP*. With this preprocessing cost, referred to earlier in list item **??** of the introduction, the user sets her own best permutation to work with. Of course, there is nothing to prevent one from including axes several times in different positions and experimenting with different permuations in the course of the exploration.

Compound Querries – Financial Data

$\mathcal{T}$o be explored next is the financial dataset shown in Fig. 13, the goal being to discover relations useful for investments and trading. The datat for the years 1986 (second tick on the 3rd axes) and 1992 are selected and compared. In 1986 the **Yen** had the greater volatility among the 3 currencies, interests varied in the mid-range, gold had a price gap while **SP**500 was uniformly low. By comparison in 1992, the **Y**en was stable while the **S**terling was very volatile (possibly due to Soros' speculation), interests and

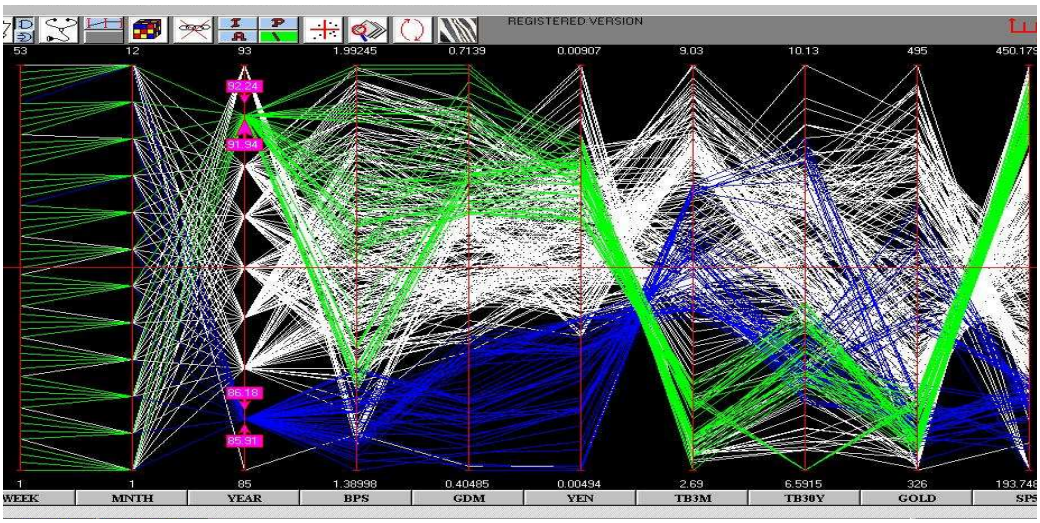


Figure 13: Financial data.

Quotes by **W**eek-on Monday, **M**onth, **Y**ear first 3 axes fix the date; **S**terling , **D**mark, **Yen** rates per $ 4th, 5th, 6th axes; **3M**TB, **30Y**TB interest rates in %, 7th, 8th axes; **G**old in $/ounce, 9th, **SP**500 index values on 10th axes.

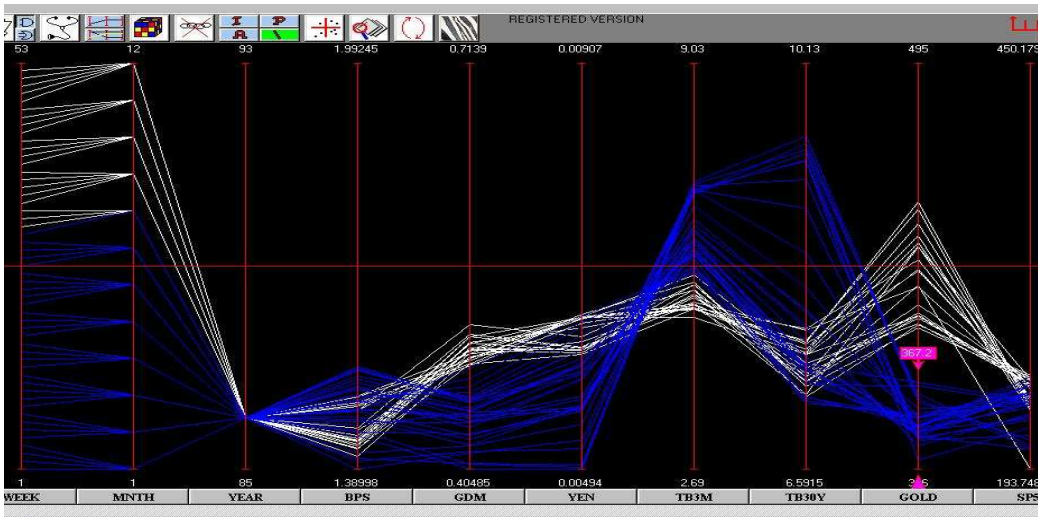


Figure 14: . Gold prices In 1986.

Gold prices jumped in the 2nd week of August. Note the correlation between the low **Yen**, high **3M**TB rates and low **G**old price range.

gold price were low and the **SP**500 was uniformly high. Two **I**nterval queries are combined with the *OR* boolean operator (i.e. Union) to obtain this picture. We continue

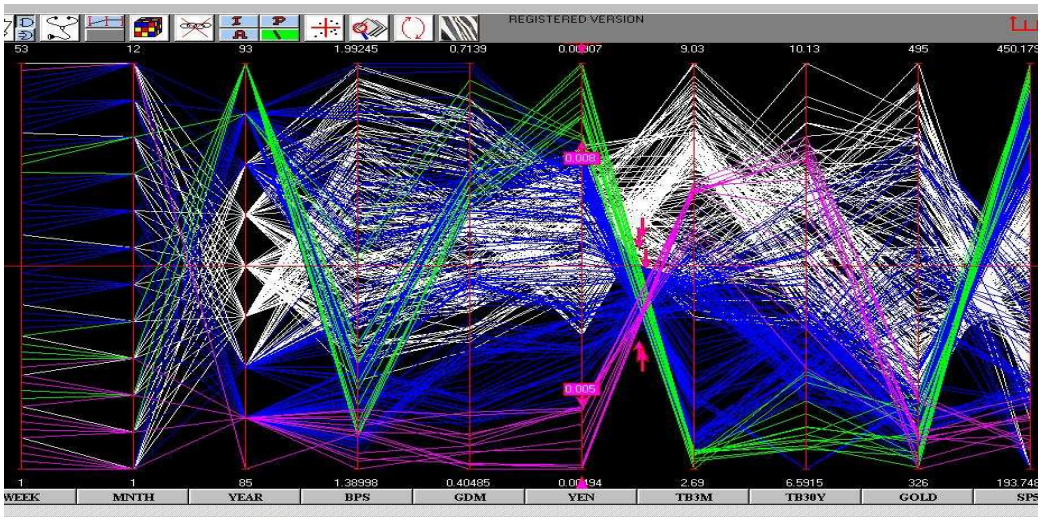


Figure 15: Negative correlation.

The crossing lines between the 6th and 7th axes in Fig. 13 show strong negative correlation between **Yen** and **3M**TB rates. One cluster is selected with the **P**inch query and combined with the high and low ranges on the **Yen** axis. Data for the years 1986 and 1992 are selected.

- **“looking for the gold” by checking out patterns that caught our attention.**

The data for 1986 is isolated in Fig. 14 and the lower range in the gold price gap is selected. Gold prices were low until the 2nd week in August when they jumped and stayed higher. The exploration was carried out in the presence of four financial experts who carefully recorded the relation between low **Yen**, high **3M**TB rates and low **G**old prices By the way, *low* **Yen** rate of exchange means the Yen has high value relative to the US $.

There are two bunches of crossing lines between 6th and 7th axes in Fig. 13 which together comprise more than 80 % of the dataset. This and recalling the previous discussion on the *line* ← *point* mapping in Fig. 11 points out the strong negative correlation between **Yen** and **3M**TB rates. The smaller cluster in Fig. 15 is selected. Moving from the top range of any of the two axes, with the **I** query, and lowering the range causes the other variable’s range to rise and is a nice way to show negative correlation interactively. For the contrarians among us, we check also for positive correlation Fig. 16. We find that it exists when **G**old prices are low to mid-range as happened for a period in the 90’s. This is a free investment tip for bucking the main trend shown in Fig. 15. It is also a nice opportunity for showing the *inversion* feature activated by the icon with 2 cyclical arrows. A variable is selected and the min/max values on that axes are inverted. Diverging lines (as for + correlation) now intersect Fig. 17 making it easier visually to spot the crossing and hence the correlation. Actually, the recommendation is to work with the **A** query experimenting with various angle ranges using the inversion to check out or confirm special clusters. When stuck don’t just stand there but

- **vary one of the variables watching for interesting variations in the other variables.**

Doing this on the **Yen** axis, Fig. 18, we strike another gold connection. The (rough) intersection of a bunch of lines joining **Yen** to the **D**mark corresponds, by the duality, to their rate of exchange. When the rate of exchange changes so does the intersection **and the price of Gold**! That is movements in currency exchange rates and the price range of **G**old go together. Are there any indications that are associated

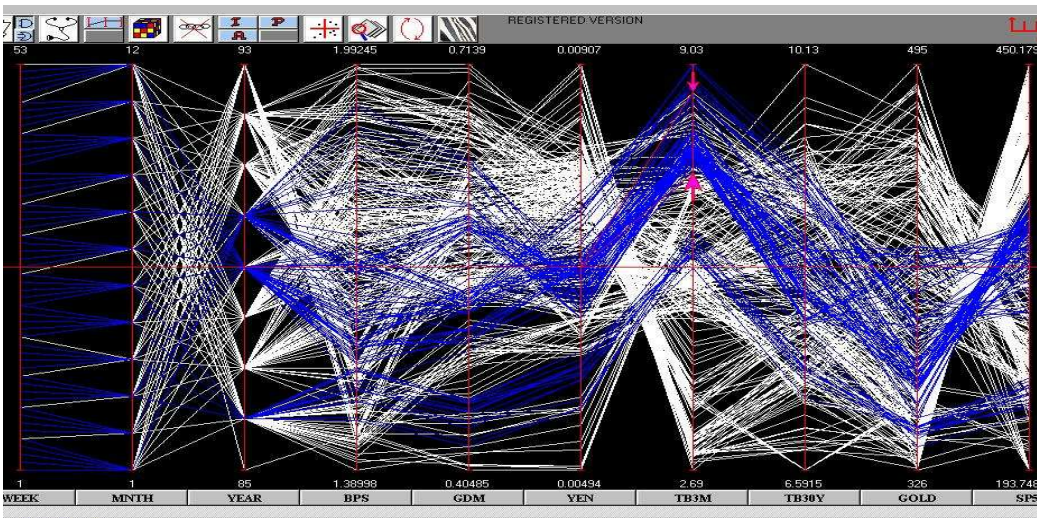


Figure 16: Positive correlation.

A positively correlated where the **Yen** and **3M**TB rates move *together when* **G***old prices are low to mid-range*.

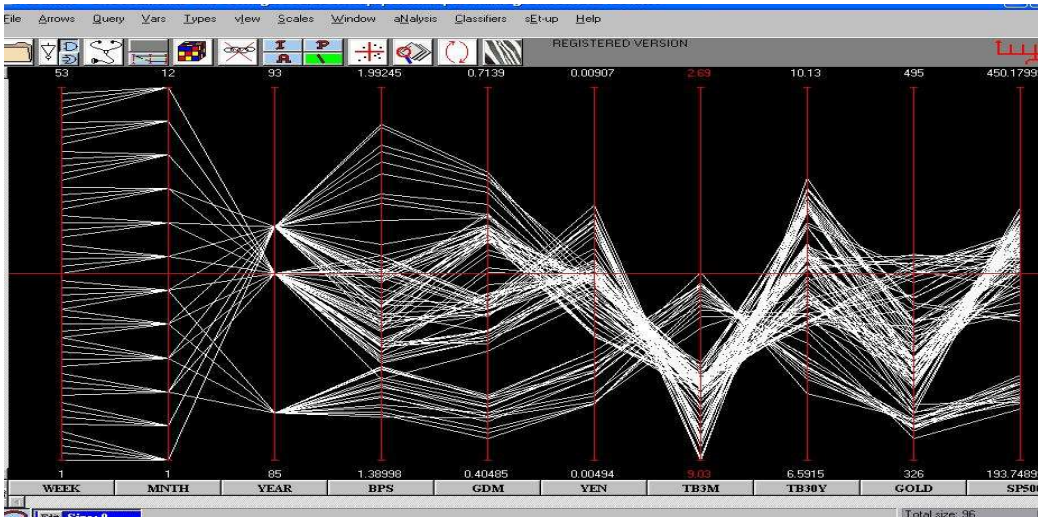


Figure 17: Inverting the **3M**TB axis.

Now the lines between the **Y**en-**3M**TB and **3M**TB-**30M**TB axes in Fig. 16 cross.

with the high range of **G**old? The top price range is selected, Fig. 19, and prompted by the result of the previous query we check out the exchange rate between **S**terling and **D**mark (or **Yen**) and the resul is stunning: a perfect straight line. The slope *is* the rate of exchange which is constant when **G**old tops out. The relation between **S**terling and **D**mark is checked for different price ranges of **G**old, Fig. 20, and the only regularity found is the one straight-line above. Aside from the trading guideline it establishes, it suggests "behind-the-scenes manipulation of the **G**old market" ... we could have said that but we won't. We perish this thought and proceed with the boolean complement, Fig. 21 of an **I** (or any other) query. Not finding anything we select a narrow but dense range on the **Yen**, Fig. 22 and notice an interesting relation between **D**mark, interest rates and **G**old.

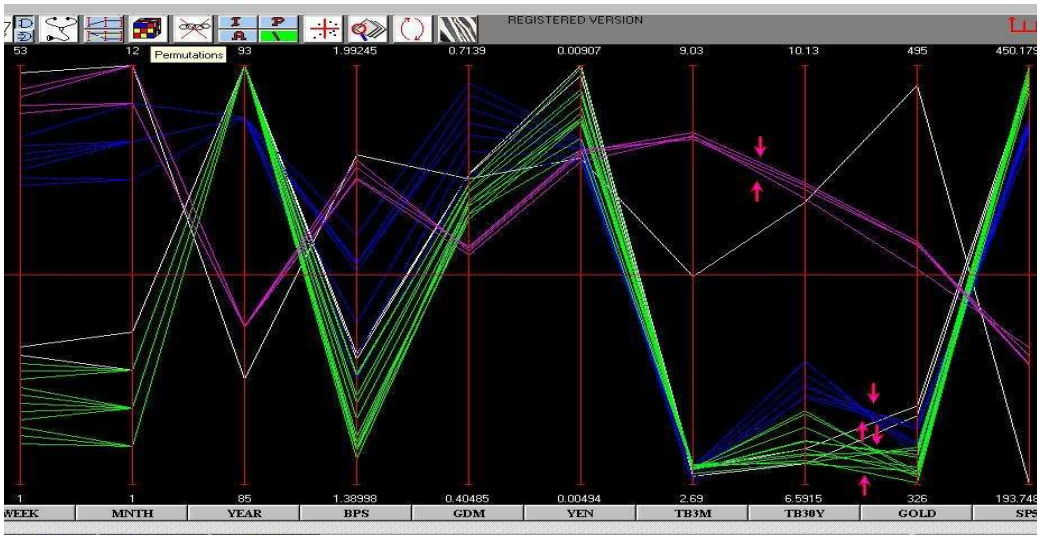


Figure 18: Variations in currency exchange rates.

Variations in the rate of exchange of the currencies correlate with movements in the price of **G**old.

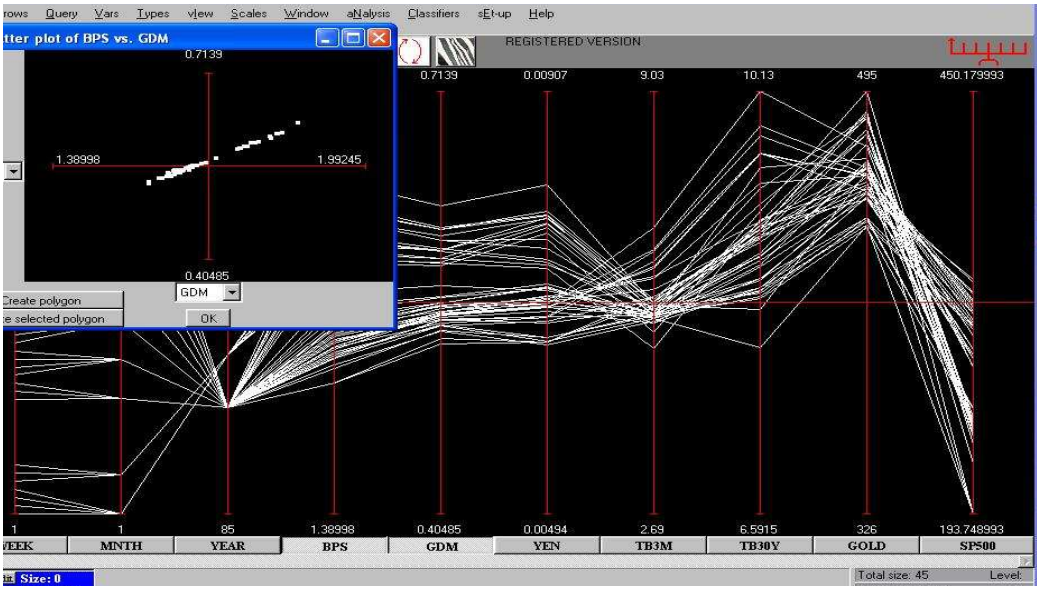


Figure 19: High **G**old.

Note the perfect straight line in the **S**terling vs. **D**mark plot. The slope is the *rate of exchange* between them and which remains constant when **G**old prices peak.

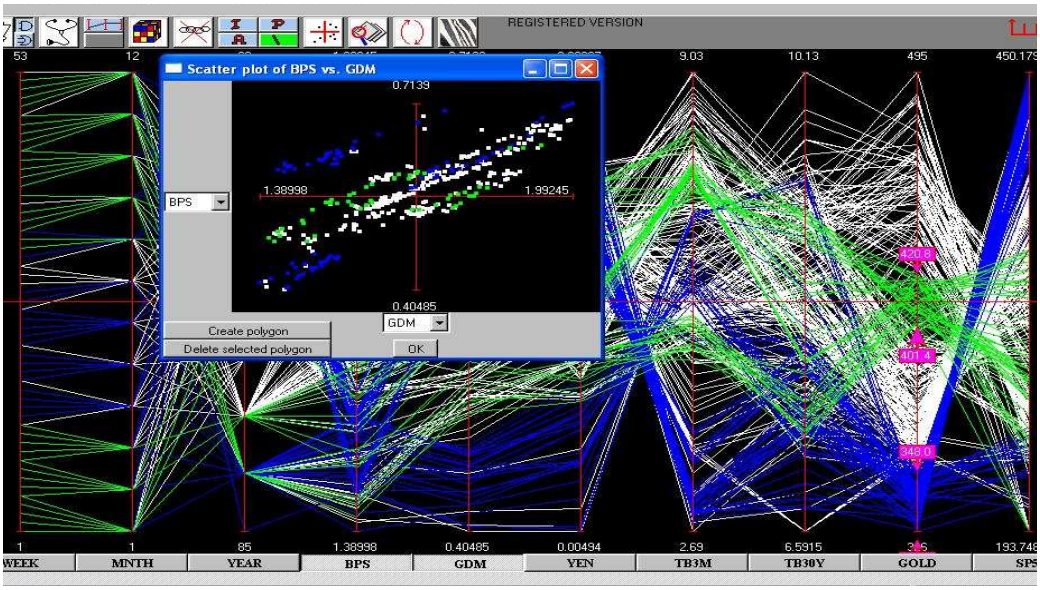


Figure 20: Two price ranges of **G**old.

The associated **S**terling vs. **D**mark plots show no regularity.

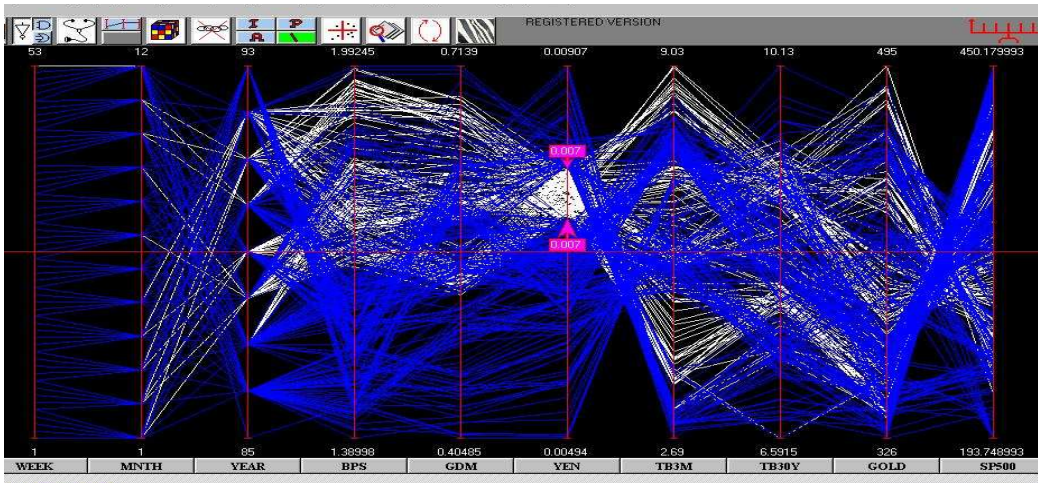


Figure 21: The complement of an **I** query.

There is an exploratory step akin to "multidimensional contouring" which we fondly call **Zebra** activated by the last icon button on the right with the appropriate skin-color. A variable axis is selected, the **SP**500 axis in Fig. 23, and divided into a number (user specified) intervals (here it is 4) and colored differently. This shows the connections (influence) of the intervals with the remaining variables which here is richly structured especially for the highest range. So what does it take for the **SP**500 to rise? This is a good question and helps introduce Parallax's classifier. The result, shown in Fig. 24 confirms the investment community's experience that low **3M**TB and **G**old predict high **SP**500. A comparison with the results

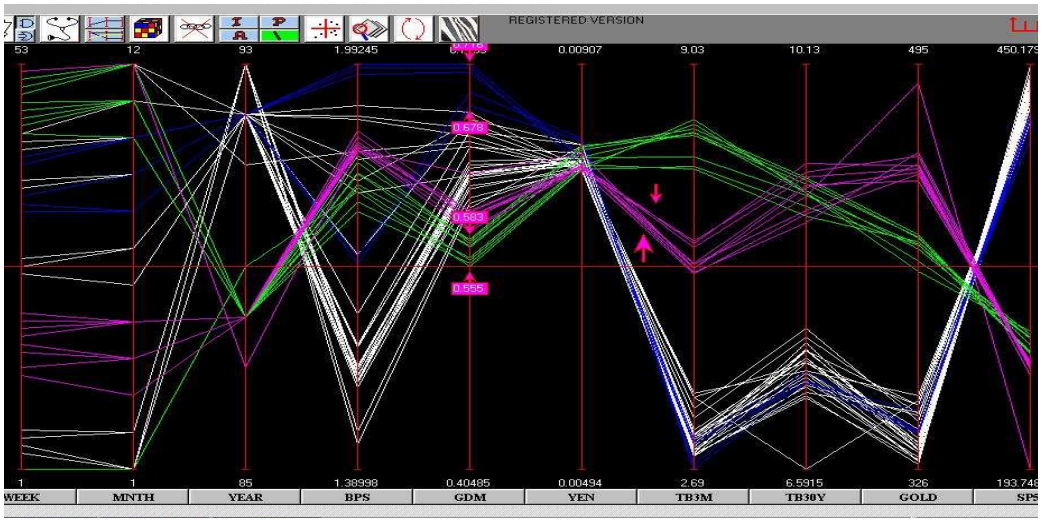


Figure 22: **Yen** stable.

For the **Yen** trading in a narrow range, high **D**mark goes with low **3M**TB rates, low **D**mark goes with high **3M**TB rates, while mid **3M**TB rates go with high **G**old.

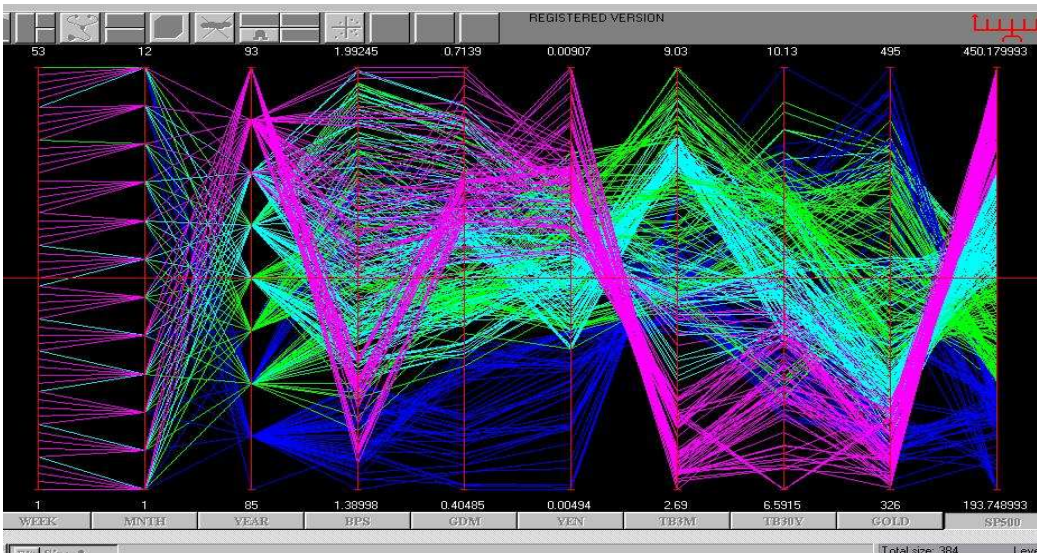


Figure 23: The *zebra* query.

It partitions and colors the segments of values differently. A variable, here the **SP**500 axis, is divided it into equal (here 4) intervals. This quickly reveals interelationships. Note especially those for the highest **SP**500 range and see next figure.

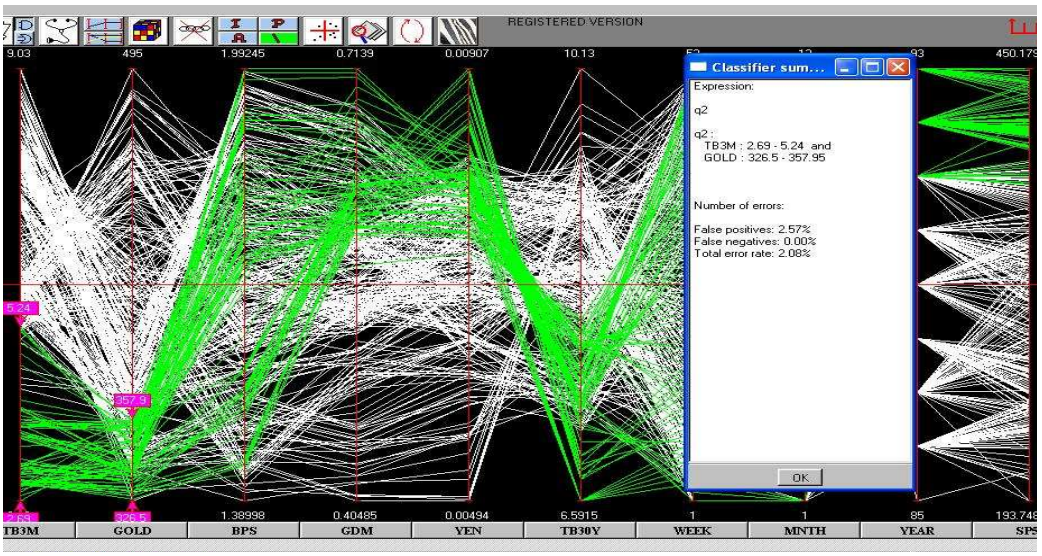


Figure 24: The rule for high **SP**500.

Both **3M**TB (the “short-bond” as it is called) and **G**old are low and in this order of importance.

obtained on this dataset with other visualization tools would be instructive though unfortunately not available. Still let us consider such an analysis done by the scatterplot matrix. There are 10 variables (axes)

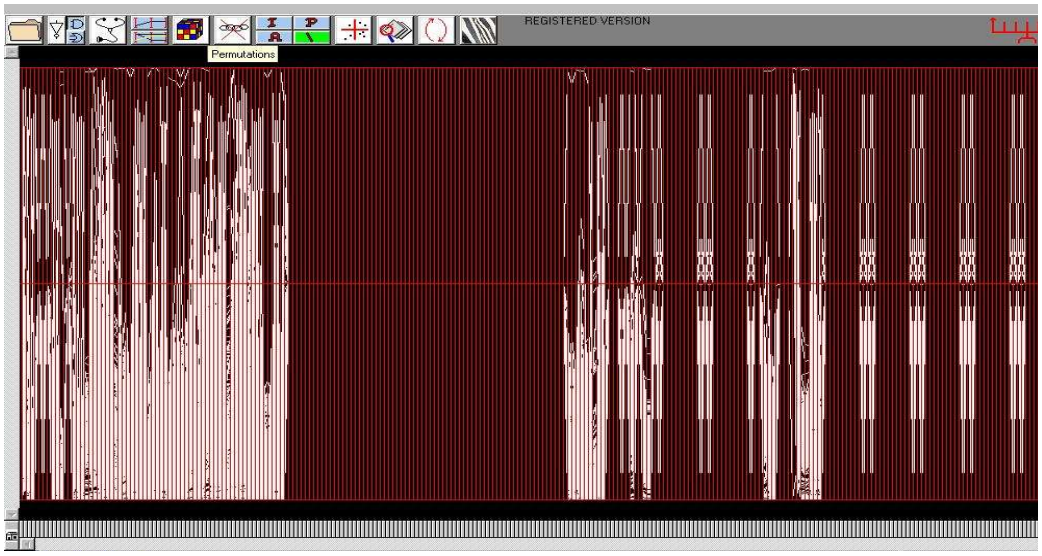


Figure 25: Manufacturing process measurements – 400 variables.

which requires 45 pairwise scatterplots each, even with a large monitor screen being no larger than about $2.5 \times 2.5\,cm^2$ square. Varying 1, 2 or more variables in tandem and observing the effects ***simultaneously*** over **all** the variables in the 45 squares may be possible but quite challenging. By contrast, the effects of varying **D**mark, *conditionally* for stable **Y**en, are easily seen on the two interest rates, **G**old as well as the remaining variables in *one* Fig. 22. This example illustrates the difficulties due to high *Representational Complexity* (see Section item # 2) which is $O(N^2)$ for the scatterplot matrix but $O(N)$ for ||-coords and made even clearer with the next dataset.

Hundreds of variables

An important question frequently asked is "how many variables can be handled with ||-coords?" The largest dataset that I have effectively worked with had about 800 variables and 10,000 data entries. With various techniques developed over the years and the automatic classifier discussed in the next section much larger datasets can be handled. Still the relevant admonition is

- **be sceptical about the quality of datasets with large number of variables.**

When hundreds or more variables are involved, it is unlikely that there are many people around who have a good feel for what is happening as confirmed by my experience. A case in point is the dataset shown in Fig. 25 consisting of instrumentation measurements for a complex process. The immediate observation is that lots of instruments recorded 0 for the duration something which was unnnoticed. Another curiosity was the repetitive patterns on the right. It turns that several variables were measured in more than one location using different names. When the dataset was cleaned-up of the superfluous information it was reduced to about 90 variables as shown in Fig. 26 and eventually to about 30 which contained the information

of real interest. By my tracking the phenomenon of repetitive measurements is widespread with at least 10 % of the variables, occurring in large datasets, being duplicates or near duplicates, possibly due to instrument non-uniformities, as suggested in the 2 variable scatterplot2 in Fig. 26. Here the repetitive observations were easily detected due to the fortuitous variable permutation in the display. Since repetitive measurements occur frequently it may be worth adding to the software an automated feature to detect and exhibit the suspect variables. This brief exposure is just an indication that large (in dimension – i.e. in number of variables) datasets can still be gainfully explored in ||-coords.

There follows a different example of EDA on a process control dataset s [23] where compound queries turned out to be very useful and where we learn to add, to the list of exploration guidelines, arguably the most important one:

- **test the assumptions and especially the "I am really sure of"s.**

Production of VLSI (chips)

The dataset, displayed in Fig. 27, consists of production data of several batches of a specific VLSI (computer chip) with measurements of 16 parameters involved in the process. The parameters are denoted by $X1, X2, \ldots, X16$. The *yield*, as the % (percent) of useful chips produced in the batch, is denoted by $X1$, and $X2$ is a measure of the *quality* (given in terms of speed performance) of the batch. Ten different categories of *defects* are monitored and the variables' scales of $X3$ through $X12$ are inverted so that 0 (zero) amount appears at the top and increasing amounts appear proportionately lower. The remaining $X13$ through $X16$ denote some physical parameters.

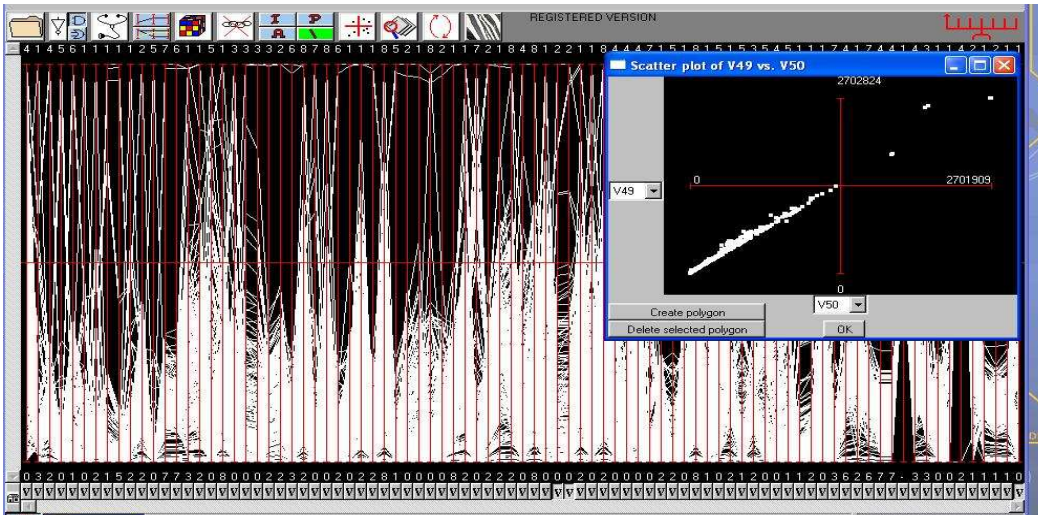


Figure 26: The above dataset after "clean-up" with about 90 variables left.

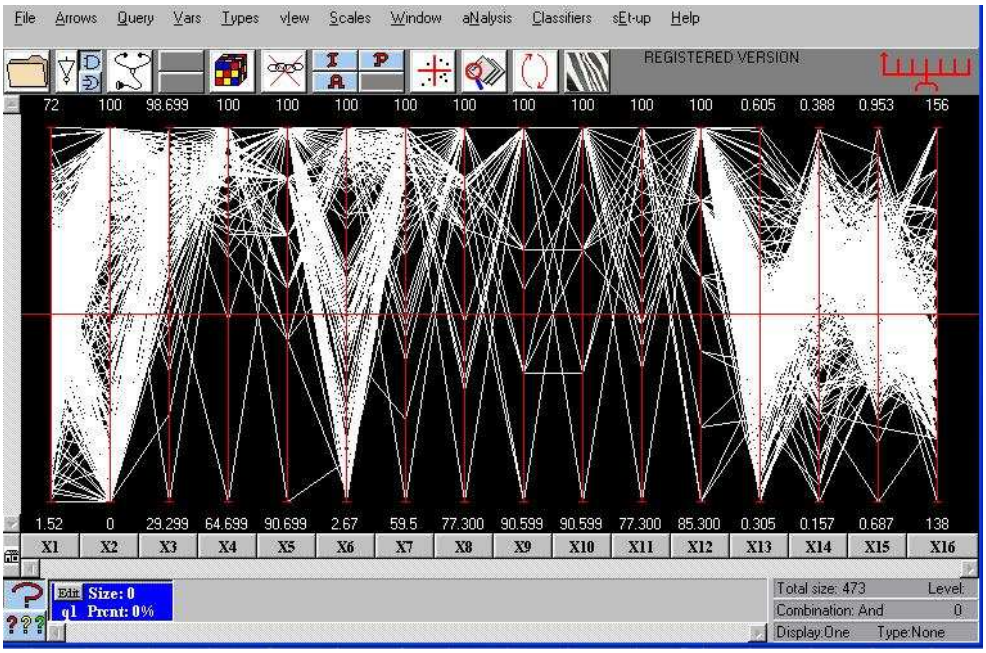


Figure 27: Dataset – VLSI production with 16 parameters

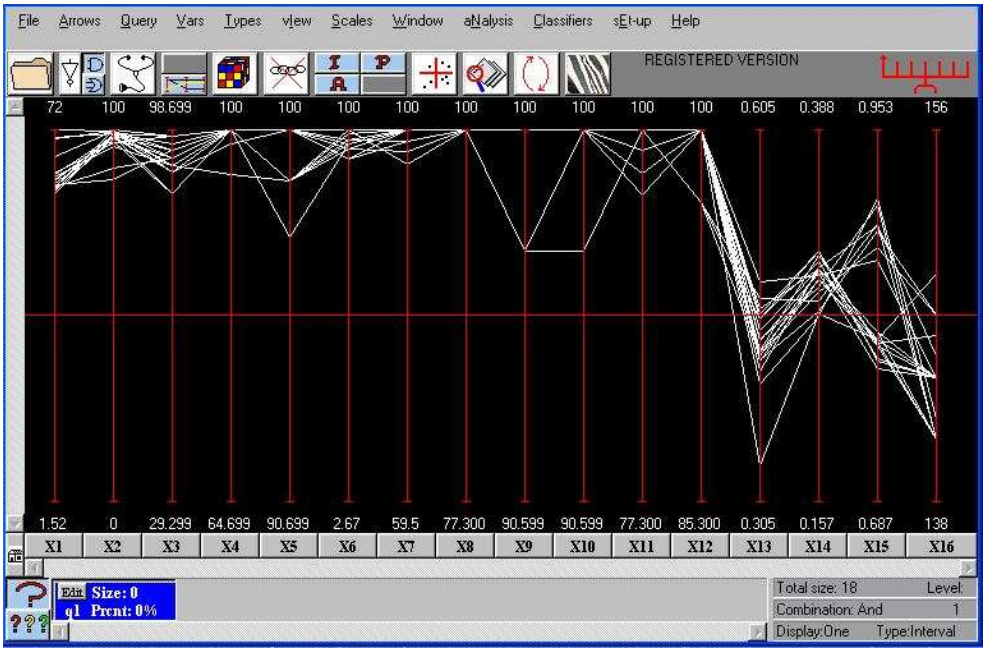


Figure 28: The batches high in Yield, $X1$, and Quality, $X2$.

Figure 29: The batches with zero in 9 out of ten defect types.

Since the goal here is to raise the yield, $X1$, while maintaining high quality, $X2$, we have a case of multi-objective optimization due to the presence of more than one objective. The production specialists believed that it was the presence of defects which prevented high yields and qualities. So their purpose in life was to keep on pushing – at considerable cost and effort – for zero defects.

With this in mind the result of our first query is shown in Fig. 28 where the batches having the highest $X1$ and $X2$ have been isolated. This in an attempt to obtain clues; and two real good ones came forth. Notice the resulting range of $X15$ where there is a significant separation into two clusters. As it turns out, this gap yielded important insight into the physics of the problem. The other clue is almost hidden. A careful comparison – and here interactivity of the software is essential – between Fig. 27 and Fig. 28 shows that some batches which were high in $X3$ (i.e. due to the inverted scale low in that defect) were not included in the selected subset. That casts some doubt into the belief that zero defects are the panacea and motivates the next query where we search for batches having zero defects in at least 9 (excluding $X3$ where we saw that there are problems) out of the 10 categories. The result is shown in Fig. 29 and is a shocker. There are 9 such batches and all of them have poor yields and for the most part also low quality! That this was not questioned and discovered earlier is surprising. We scrutinize the original picture Fig. 27 for visual cues relevant to our objectives and our findings so far. And ... there is one staring us in the face! Among the 10 defects $X3$ through $X12$ whatever $X6$ is, it's graph is very different than the others. It shows that the process is much more sensitive to variations in $X6$ than the others. For this reason, we chose to treat $X6$ differently and remove its zero defect constraint. This query (not shown) showed that the very best batch (i.e. highest yield with very high quality) does not have zeros (or the lowest values) for $X3$ and $X6$; a most heretical finding. It was confirmed by the next query which isolated the cluster of batches with the top yields (note the gap in $X1$ between them and the remaining batches). These are shown in Fig. 30

and they confirm that small amounts (the ranges can be clearly delimited) of $X3$ and $X6$ type defects are essential for high yields and quality.

Returning to the subset of data which best satisfied the objectives, Fig. 28 in order to explore the gap in the range of $X15$, we found that the cluster with the high range of $X15$ gives the lowest (of the high) yields $X1$, and worse it does not give consistently high quality $X2$, whereas the cluster corresponding to the lower range has the higher qualities and the full range of the high yield. It is evident that the small ranges of $X3$, $X6$ close to (but not equal to) zero, together with the short (lower) range of $X15$ provide necessary conditions for obtaining high yields and quality. This is also indicated in Fig. 30. By a stroke of good luck these 3 can also be checked early in the process avoiding the need of "throwing good money after bad"(i.e. by continuing the production of a batch whose values of $X3$, $X6$ and $X15$ are not in the small "good" ranges we have found).

These findings were significant and differed from those found with other methods for statistical process control[2]. This approach has been successfully used in a wide variety of applications from the manufacture of printed circuit boards, PVC and manganese production, financial data, determining skill profiles" (i.e. as in drivers, pilots), etc.

Classification

Though it is fun to undertake this kind of exploration, the level of skill and patience required tends to discourage some users. It is not surprising then that the most persistent requests and admonitions have been for tools which, at least partially, automate the knowledge discovery process [25]. Classification is a

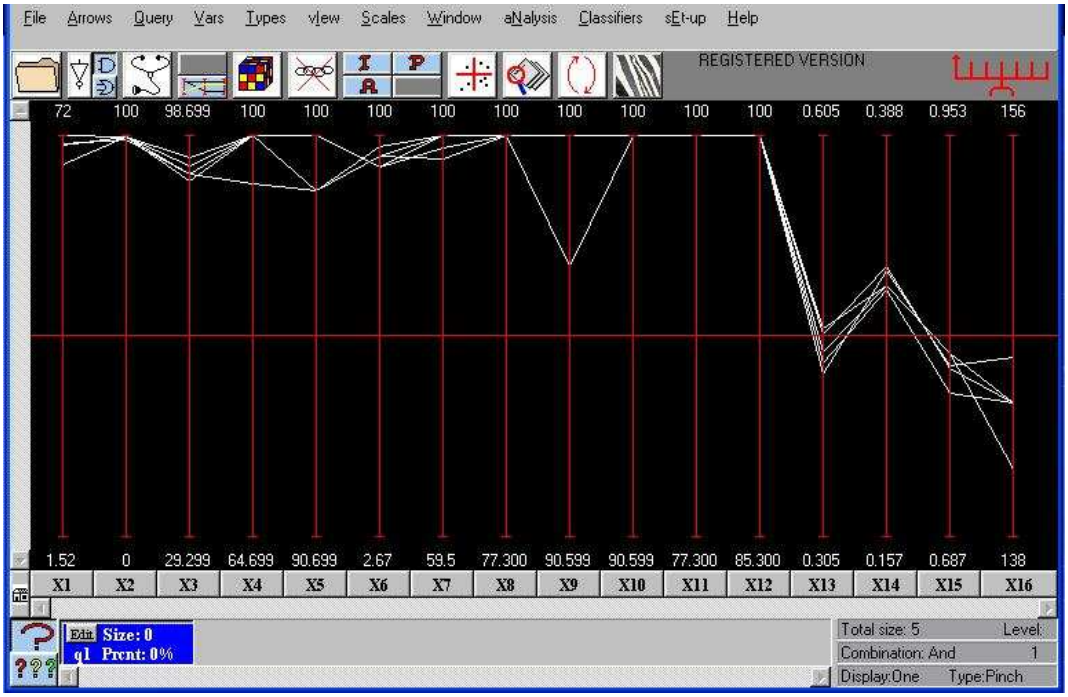


Figure 30: The batches with the highest Yields.

They do not have the lowest defects of type $X3$ and $X6$.

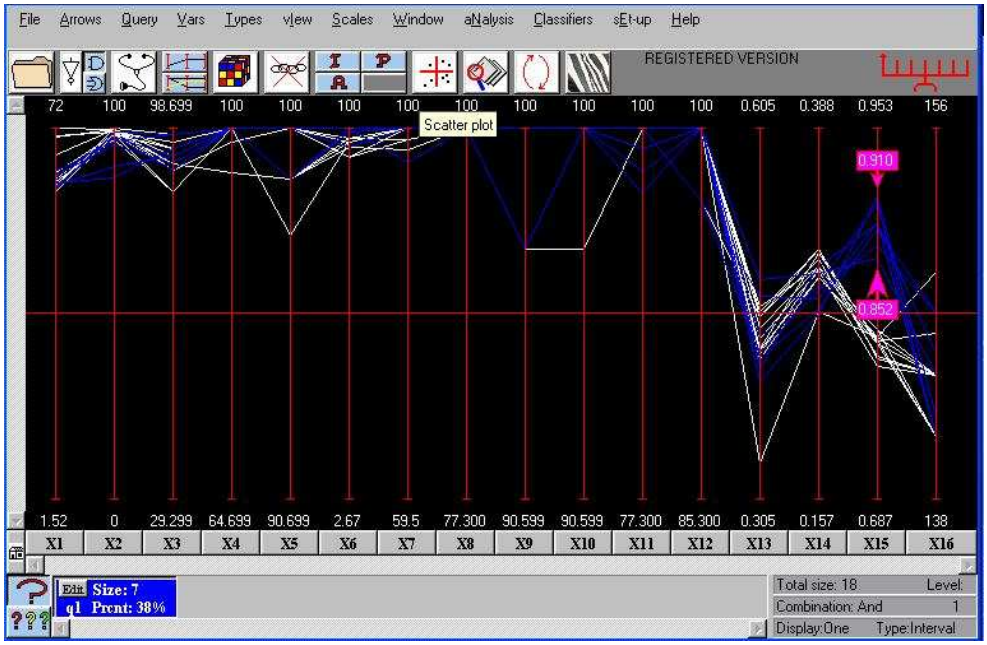


Figure 31: Only the lower range of $X15$ is associated with the highest Yields and Quality.

basic task in data analysis and pattern recognition and an algorithm accomplishing it is called a **Classifier** [36], [13], [33]. The input is a dataset P and a designated subset S. The output is a characterization, a set of conditions or rules, to distinguish elements of S from all other members of P the "global" dataset. The output may also be that there is insufficient information to provide the desired distinction.

With parallel coordinates a dataset P with N variables is transformed into a set of points in N-dimensional space. In this setting, the designated subset S can be described by means of a hypersurface which encloses just the points of S. In practical situations the strict enclosure requirement is dropped and some points of S may be omitted ("false negatives"), and some points of $P-S$ are allowed ("false positives") in the hypersurface. The description of such a hypersurface is equivalent to the rule for identifying, within some acceptable error, the elements of S. Casting the problem in a geometrical setting leads us to *visualize* how such may work. This entails:

1. use of an efficient "wrapping" (a convex-hull approximation) algorithm to enclose the points of S in a hypersurface S_1 containing S and in general also some points of $P-S$; so $S \subset S_1$.

2. the points in $(P-S) \cap S_1$ are isolated and the wrapping algorithm is applied to enclose them, usually also enclosing some points of S_1, producing a new hypersurface S_2 with $S \supset (S_1 - S_2)$,

3. the points in S not included in $S_1 - S_2$ are next marked for input to the wrapping algorithm, a new hypersurface S_3 is produced containing these points as well as some other points in $P-(S_1-S_2)$ resulting in $S \subset (S_1-S_2) \cup S_3$,

4. the process is repeated alternatively producing upper and lower containment bounds for S; termination occurs when an error criterion (which can be user specified) is satisfied or when convergence is

not achieved. After termination is obtained two error measures are available to estimate the rule's precision :

- *Train & Test*. A portion of the dataset (usually 2/3) selected at random is used to derive the classification rule, which is then tested on the remaining 1/3 of the data.
- Cross-Correlation.

It can and does happen that the process does not converge when P does not contain sufficient information to characterize S. It may also happen that S is so "porous" (i.e. sponge-like) that an inordinate number of iterations are required. On convergence, say at step $2n$, the description of S is provided as :

$$S \approx (S_1 - S_2) \cup (S_3 - S_4) \cup \ldots \cup (S_{2n-1} - S_{2n}) \tag{4}$$

this being the terminating expression resulting from the algorithm which we call **Nested Cavities** (abbr. **NC**).

The user can select a subset of the available variables and restrict the rule generation to these variables. In certain applications, as in process control, not all variables can be controlled and hence it is useful to have a rule involving only the accessible (i.e. controllable) variables. An important additional benefit, is that the minimal set of variables needed to state the rule is found and ordered according to their predictive value. These variables may be considered as the best *features* to identify the selected subset. The algorithm is display independent there is no inherent limitation as to the size and number of variables in the dataset. Summarizing for **NC**,

- an approximate convex-hull boundary for each cavity is obtained,
- utilizing properties of the representation of multidimensional objects in ||-coords, a very low polynomial worst case complexity of $O(N^2|P|^2)$ in the number of variables N and dataset size $|P|$ is obtained; it is worth contrasting this with the often unknown, or unstated, or very high (even exponential) complexity of other classifiers,
- an intriguing prospect, due to the low complexity, is that the rule can be derived in near real-time making the classifier adaptive to changing conditions,
- the minimal subset of variables needed for classification is found,
- the rule is given explicitly in terms of conditions on these variables, i.e. included and excluded intervals, and provides "a picture" showing the complex distributions with regions where there are data and "holes" with no data providing important insights to the domain experts.

A neural-pulse dataset has interesting and unusual features. There are two classes of neurons whose outputs to stimuli are to be distinguished. They consist of 32 different pulses measured in a monkey's brain (poor thing!). There are 600 samples with 32 variables (the pulses)[3]. Various classification methods were unable to obtain a rule. With **NC** convergence is obtained requiring only 9 of the 32 parameters for the classification rule for class # 1. The resulting ordering shows a striking separation. In Fig. 32 the first pair of variables x_1, x_2 in the original order is plotted on the left. On the right the best pair x_{11}, x_{14}, as chosen

[3]I am grateful to Prof. R. Coiffman and his group at the CS & Math. Depts at Yale University for giving me this dataset.

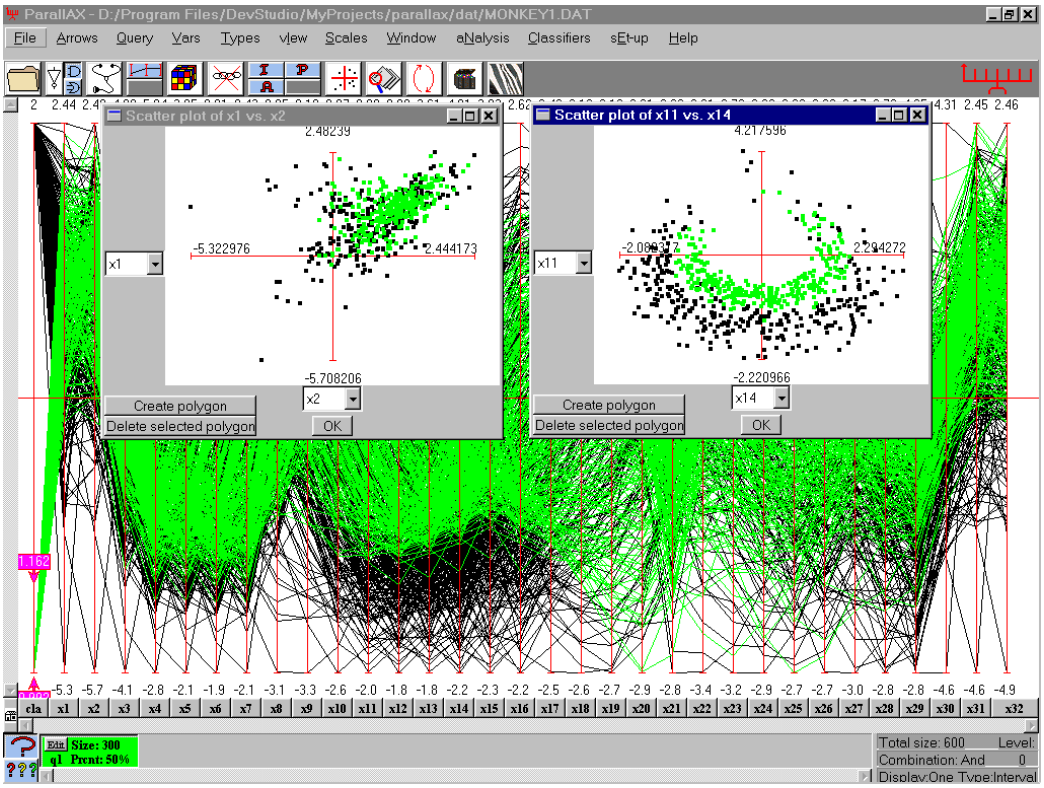


Figure 32: The neural-pulses dataset with 32 parameters and two categories.

Dataset is shown in the background. On the left plot are the first two parameters in the original order. The classifier found the 9 parameters needed to state the rule with 4 % error and ordered them according to their predictive value. The best two parameters are plotted on the right showing the separation achieved.

by the classifier's ordering speaks for itself. By the way, the discovery of this manually would require constructing a scatterplot matrix with 496 pairs, then carefully inspecting and comparing the individual plots. The implementation provides all the next best sections, some of which are shown in Fig. 33, to aid the visualization of the rule. The dataset consists of two "pretzel-like" clusters wrapping closely in 8-D one enclosing the other; showing that the classifier can actually "carve" highly complex regions with the cavity shown. One can understand why separation of clusters by hyperplanes or nearest-neighbor techniques can fail badly for such datasets. The rule has 4 % error some of which are shown in Fig. 33.

The rules are explicit, "visualizable", optimally ordering the minimal set of variables needed to state the rule without loss of information. There are variations which apply in some situations where the **NC** classifier fails, such as the presence of several large "holes" (see [25]). Further, keeping in mind that the classification rule is the result of several iterations suggests heuristics for dealing with the pesky problem of *over-fitting*. The iterations can be stopped just where the corrections in eq. (4) become very small, i.e. the S_i consist of a small number of points. The number of iterations is user defined and the resulting rule yields an error in the test stage more stable under variations in the number of points of the test set. In addition, the user can exclude variables from being used in the description of the rule; those ordered last are the ones providing the smaller corrections and hence more liable to over-correct.

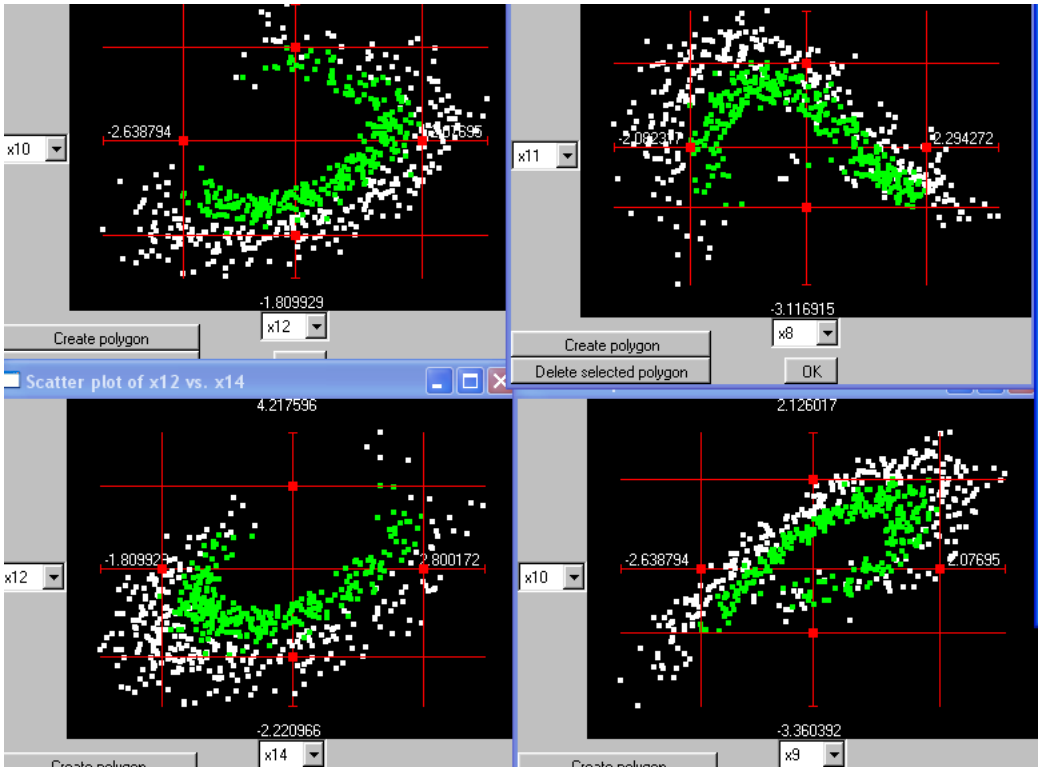


Figure 33: Neural dataset classification.

Further cross-sections of the hypersurface corresponding to the classification rule.

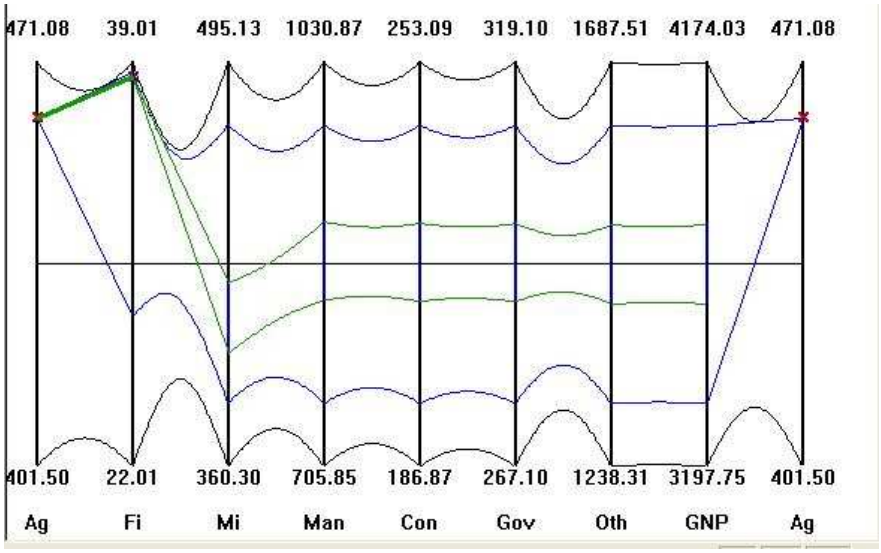


Figure 34: Model of a country's economy.

Choosing high **Ag**ricultural and high **Fish**ing output **forces** low **Min**ing output.

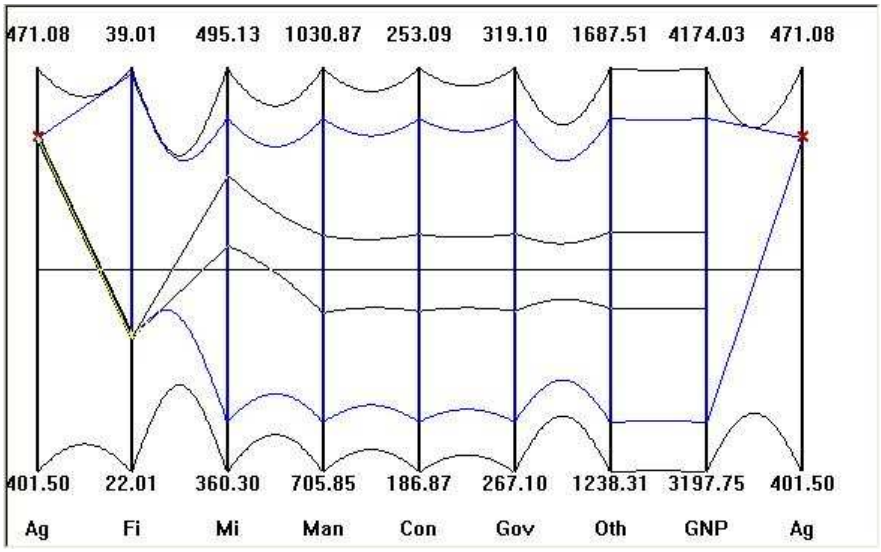


Figure 35: Competition for labor between the **Fish**ing & **Min**ing sectors

*F*inally we illustrate the methodology's ability to model multivariate relations in terms of hypersurfaces – just as we model a relation between two variables by a planar region. Then by using the interior point algorithm, as shown in Fig. 41 of the next section, with the model we can do trade-off analyses, discover sensitivities, understand the impact of constraints, and in some cases do optimization. For this purpose we shall use a dataset consisting of the outputs of various economic sectors and other expenditures of a particular (and real) country. It consists of the monetary values over several years for the **Ag**ricultural, **Fish**ing, and **Min**ing sector outputs, **Man**ufacturing and **Con**struction industries, together with **Gov**ernment, Miscellaneous spending and resulting GNP; eight variables altogether. We will not take up the full ramifications of constructing a model from data. Rather, we want to illustrate how ||-coords may be used as a modeling tool. Using the Least Squares technique we "fit" a function to this dataset and are not concerned at this stage whether the choice is "good" or not. The function obtained bounds a region in $\mathbb{R}^8$ and is represented by the upper and lower curves shown in Fig. 34.

The picture is in effect a simple visual model of the country's economy, incorporating its capabilities, limitations and interelationships among the sectors. A point interior to the region, satisfies all the constraints simultaneously, and therefore represents (i.e. the 8-tuple of values) a feasible economic policy for that country. Using the interior point algorithm we can construct such points. It can be done interactively by sequentially choosing values of the variables and we see the result of one such choice in Fig. 34. Once a value of the first variable is chosen (in this case the **Agr**icultural output) within its range, the dimensionality of the region is reduced by one. In fact, the upper and lower curves between the 2nd and 3rd axes correspond to the resulting 7-dimensional hypersurface and show the available range of the second variable **Fish**ing reduced by the constraint. This can be seen (but not shown here) for the rest of the variables. That is, due to the relationship between the 8 variables, a constraint on one of them impacts all the remaining ones and restricts their range. The display allows us to experiment and actually see the impact

of such decisions downstream. By interactively varying the chosen value for the first variable we found, that it not possible to have a policy that favors **Agr**iculture without also favoring **Fish**ing and vice versa.

Proceeding, a very high value from the available range of **Fish**ing is chosen and it corresponds to very low values of the **Min**ing sector. By contrast in Fig. 34 we see that a low value in **Fish**ing yields high values for the **Min**ing sector. This inverse correlation was examined and it was found that the country in question has a large number of migrating semi-skilled workers. When the fishing industry is doing well most of them are attracted to it leaving few available to work in the mines and vice versa. The comparison between the two figures shows the competition for the same resource between **Min**ing and **Fish**ing. It is especially instructive to discover this interactively. The construction of the interior point proceeds in the same way. In the next section in the discussion on surfaces this construction is shown for higher dimensional hypersurfaces.

Parallel Coordinates – The Bare Essentials

The following short review of ||-coords together Fig. 11 and the discussion on duality provide the essential background on ||-coords to make this chapter self-contained. The detailed development of Parallel Coordinates is contained in [24].

Lines

$\mathscr{A}$n N-dimensional line ℓ can be described by the $N-1$ linear equations:

$$\ell \quad : \quad \begin{cases} \ell_{1,2} & : \; x_2 & = & m_2 x_1 + b_2 \\ \ell_{2,3} & : \; x_3 & = & m_3 x_2 + b_3 \\ & \cdots & & \\ \ell_{i-1,i} & : \; x_i & = & m_i x_{i-1} + b_i \\ & \cdots & & \\ \ell_{N-1,N} & : \; x_N & = & m_N x_{N-1} + b_N \quad , \end{cases} \tag{5}$$

each with a pair of adjacently indexed variables. In the $x_{i-1}x_i$-plane the relation labeled $\ell_{i-1,i}, N = 2,\ldots,N$ is a line, and by the *line* $\leftrightarrow$ *point* duality, eq. (3), it can be represented by the point

$$\bar{\ell}_{i-1,i} = (\frac{1}{(1-m_i)} + (i-2)\ , \ \frac{b_i}{(1-m_i)}) \tag{6}$$

Here the inter-axes distance is 1 so that $i-2$ is distance between the y (or $\bar{X}_1$) and $\bar{X}_{i-1}$ axes. Actually any $N-1$ independent equations like

$$\ell_{i,j} : x_i = m_{i,j} x_j + b_{i,j}\ , \tag{7}$$

can equivalently specify the line ℓ, for eq. (7) is the projection of ℓ on the $x_i x_j$ 2-D plane and $N-1$ such independent projections completely describe ℓ. There is a beautiful and very important relationship illustrated in (left) Fig. 36. For a line ℓ in 3-D the three points $\bar{\ell}_{12}, \bar{\ell}_{13}, \bar{\ell}_{23}$ are collinear, this line is denoted by $\bar{L}$, and any two points represent ℓ. It is easy to see that a polygonal line on all the $N-1$ points, given by eq. (6) or their equivalent, represents a point on the line ℓ. Conversely, two points determine a line ℓ. Starting with the two polygonal lines representing the points, the $N-1$ intersections of their $\bar{X}_{i-1}, \bar{X}_i$ portions are the $\bar{\ell}_{i-1,i}$ points for the line ℓ. A line interval in 10-D and several of its points is seen on the (right) Fig. 36. By the way, the indexing of the points $\bar{\ell}$ is essential.

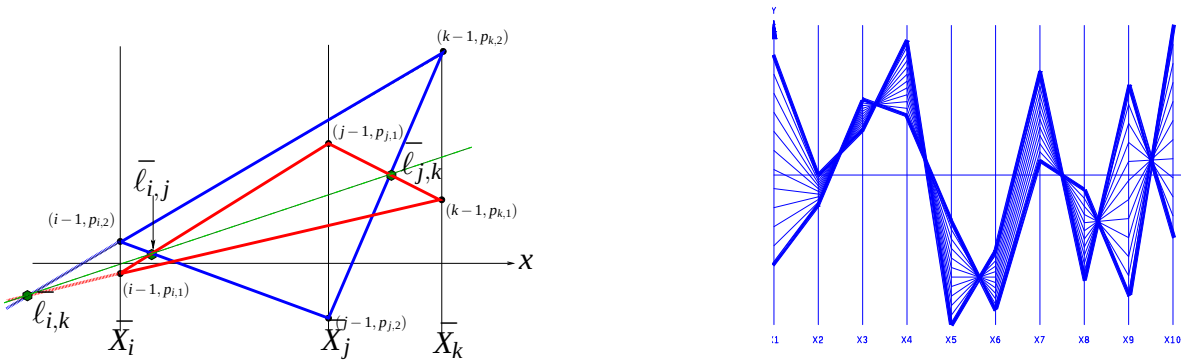


Figure 36: Properties of multidimensional lines.
(Left)The 3 points $\overline{\ell}_{i,j}, \overline{\ell}_{j,k}, \overline{\ell}_{i,k}$ are collinear for $i \neq j \neq k$. (Right) A line interval in 10-D.

Planes & Hyperplanes

While a line can be determined from its projections, a plane even in 3-D can not. A new approach is called for [9]. Rather than discerning a p-dimensional object from its points, it is described in terms of its (p-1)-dimensional subsets constructed from the points. Let's see how this works. In Fig. 37 (left) polygonal lines representing a set of coplanar points in 3-D are seen. From this picture even the most persistent pattern-seeker can **not** detect any clues hinting at a relation among the three variables much less a linear one. The plane has dimension $p = 1$ so we look at *lines* (having dimension $p - 1 = 1$) on the plane constructed so that each pair of polygonal lines the lines $\overline{L}$ of the 3 point collinearity shown in Fig. 36 (left) are obtained. The result, shown on the right, is stunning. All the $\overline{L}$ lines intersect at a point which turns out to be characteristic of coplanarity but not enough to specify the plane. Translating the first axis $\overline{X}_1$ to the position $\overline{X}_{1'}$, one unit to the right of the $\overline{X}_3$ axis and repeating the construction, based on the axes

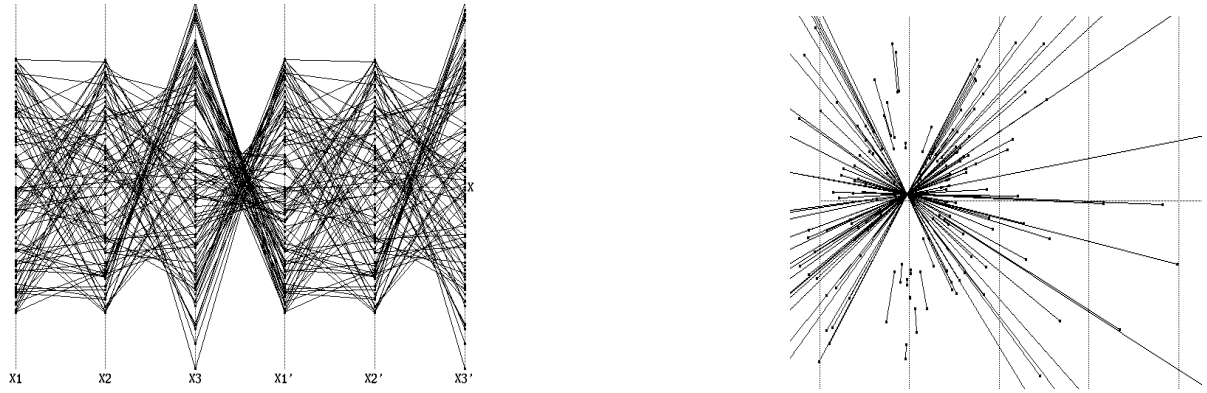


Figure 37: Coplanarity¿
(Left)The polygonal lines on the first 3 axes represent a set of coplanar points in 3-D.(Right) Coplanarity! Forming lines on the plane, with the 3 point collinearity, the resulting lines intersect at point.

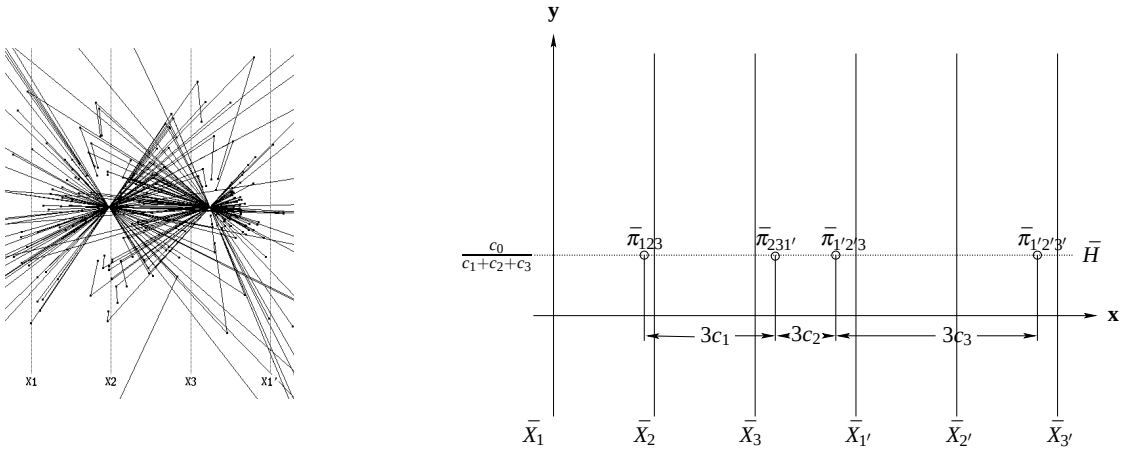


Figure 38: Plane representation.

(Left)The two points where the lines intersect uniquely determine a plane π in 3-D.(Right) From four points, similarly constructed by consecutive axes translation, the coefficients of $\pi : c_1x_1 + c_2x_2 + c_3x_3 = c_0$ can be read from the picture!

triple $\bar{X}_2, \bar{X}_3, \bar{X}_{1'}$, yields a second point shown in Fig. 38(left). For a plane described by:

$$\pi : c_1x_1 + c_2x_2 + c_3x_3 = c_0 \ , \tag{8}$$

the two points, in the order they are constructed, are respectively

$$\bar{\pi}_{123} = \left(\frac{c_2 + 2c_3}{S} \ , \quad \frac{c_0}{S} \right) \ , \ \bar{\pi}_{1'23} = \left(\frac{3c_1 + c_2 + 2c_3}{S}, \frac{c_0}{S} \right) \ , \tag{9}$$

for $S = c_1 + c_2 + c_3$. Three subscripts correspond to the 3 variables appearing in the plane's equation and the axes triple used for their construction, and distinguish them from the points with two subscripts representing lines. The 2nd and 3rd axes can also be consecutively translated, as indicated in Fig. 37(left), repeating the construction to generate two more points denoted by $\bar{\pi}_{1'2'3}, \bar{\pi}_{1'2'3'}$. These points can also be found otherwise in an easier way. The gist of all this is shown in Fig. 38(right). The distance between successive points is $3c_i$. The equation of the plane π can actually be read from the picture!

In general, a hyperlane in N-dimensions is represented uniquely by $N-1$ points each with N indices. There is an algorithm which constructs these points *recursively*, raising the dimensionality by one at each step, as is done here starting from points (0-dimensional) constructing lines (1-dimensional). By the way, all the nice higher dimensional projective dualities like *point* $\leftrightarrow$ *hyperplane*, *rotation* $\leftrightarrow$ *translation* etc hold. Further, a multidimensional object, represented in $\|$-coords, can still be recognized after it has been acted on by projective transformation (i.e. translation, rotation, scaling and perspective). The recursive construction and its properties are at the heart of the $\|$-coords visualization.

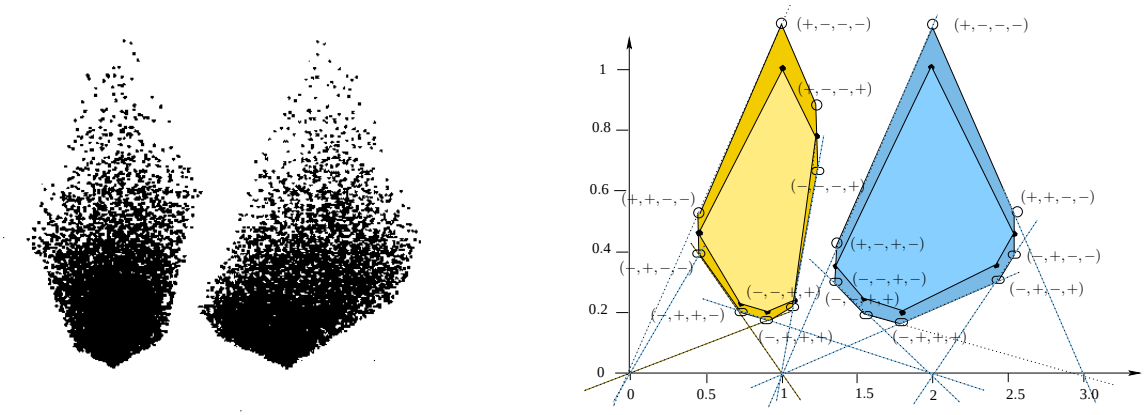


Figure 39: A family of close planes.

(Left) Pair of point clusters representing close planes.(Right) The hexagonal regions (interior) are the regions containing the points $\bar{\pi}_{123}$ (left) and $\bar{\pi}_{1'23}$ for the family of planes with $c_0 = 1$ and $c_1 \in [1/3, 1.5], c_2 \in [1/3, 2.5], c_3 \in [1/3, 1]$. For c_0 varying, here $c_0 \in [.85, 1.15]$, the regions (exterior) are octagonal with two vertical edges.

Challenge: Visualizing Families of Proximate Planes

$\mathscr{R}$eturning to 3-D, it turns out that for points as in Fig. 37 which are "nearly" coplanar (i.e. have small errors) the construction produces a pattern very similar to that in Fig. 38(left). A little experiment is in order. Let us return to the family of *proximate* (i.e. close) planes generated by

$$\Pi = \{\pi : c_1x_1 + c_2x_2 + c_3x_3 = c_0, \quad c_i \in [c_i^-, c_i^+], \quad i = 0,1,2,3\}\,, \tag{10}$$

randomly chosing values of the c_i within the allowed intervals to determine several planes $\pi \in \Pi$, keeping at first $c_0 = 1$, and plotting the two points $\bar{\pi}_{123}, \bar{\pi}_{1'23}$ as shown in Fig. 39 (left). Not only is closeness apparent but more significantly the distribution of the points is not chaotic. The outline of two hexagonal patterns can be discerned. The family of "close" planes is visualizable but also the variations in several directions. It is possible to see, estimate and compare errors or proximity [26].

It can be proved that in 3-D the set of pairs of points representing the family of proximate planes form two convex hexagons when $c_0 = 1$ with an example is shown in Fig. 39 (right), and are contained in octagons each with two vertical edges for varying c_0. In general, a family of proximate hypeplanes in N-D is represented by $N-1$ convex $2N$-agons when $c_0 = 1$ or $2(N+1)$-agons for c_0 varying. These polygonal regions can be constructed with $O(N)$ computational complexity. Choosing a point in one of the polygonal regions, an algorithm matches the possible remaining $N-2$ points, one each from the remaining convex polygons, which represent and identify hyperplanes in the family by $N-1$ points.

We pose the thesis that visualization is not about seeing lots of things but rather discovering **relations** among them. While the display of randomly sampled *points* from a family of proximate hyperplanes is utterly chaotic (the mess in Fig. 37 (right) from points in just *one* plane), their **proximate coplanarity relation** corresponds to a clear and compact pattern. With $\|$-coords we can focus and *concentrate* the

relational information rather than wallowing in the details, ergo the remark “without loss of information” when referring to ||-coords. This is the methodology’s real strength and where he future lies. Here then is a visualization challenge: how else can proximate coplanarity be detected and seen?

Nonlinear Multivariate Relations – Hypersurfaces

$\mathscr{A}$ relation among 2 real variables is represented geometrically by a unique region in 2-D. Analogously, a relation between N variables corresponds to a hypersurface in N-D, hence the need to say something about the representation of hypersurfaces in ||-coords. A smooth surface in 3-D (and also N-D) can be described as the envelope of all its tangent planes. This is the basis for the representation shown in Fig. 40 (left). Every point of the surface is mapped into the two points representing its *tangent plane at the point*. This generates 2 planar regions and for N-D there are $N-1$ such regions. These regions are *linked*, just as the polygons above, to provide the proper $N-1$ points representing each tangent hyperplane and from which the hypersurface can be reconstructed. Classes of surfaces can be immediately distinguished from their ||-coords display (see the chapter on surfaces for extensive treatment). For developable surfaces the regions consists of boundary curves only with no interior points, regions for ruled surfaces have grids consisting of straight lines, quadric surfaces have regions with conic boundaries these are some examples.

There is a simpler but inexact surface representation which is quite useful when used judiciously. The polygonal lines representing points on the boundary are plotted and their envelope “represents” the surface; the “ ” are a reminder that this is not a *unique* representation. In Fig. 41 (left) are the upper and lower envelopes for a sphere in 5-D consisting of 4 overlapping hyperbolae which must be distinguished from those in Fig. 40 (right), which is exact and, interestingly enough are also hyperbolae, the curves determined by points representing the sphere’s *tangent planes*. Retaining the exact surface description (i.e. its equation) internally, interior points can be constructed and displayed as shown for the 5-D sphere in Fig. 41 (left). On the right the same construction is shown but for a more complex 20-dimensional convex hypersurface (“model”). The intermediate curves (upper and lower) also provide valuable information and previews of coming attractions. They indicate a neighborhood of the point (represented by the

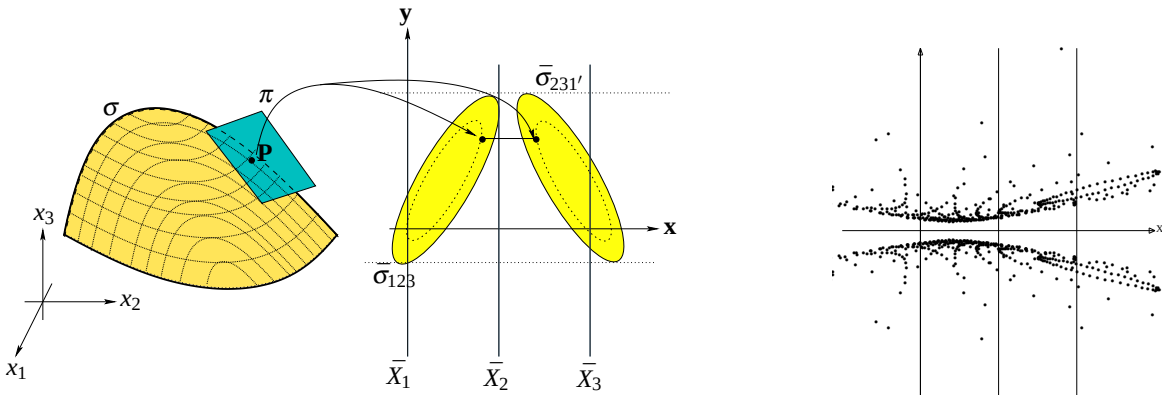


Figure 40: Surface representation.

(Left)A smooth surface σ is represented by two planar regions $\bar{\sigma}_{123}, \bar{\sigma}_{231'}$ consisting of pairs of points representing its tangent planes. (Right) One of the two hyperbolic regions representing a sphere in 3-D.

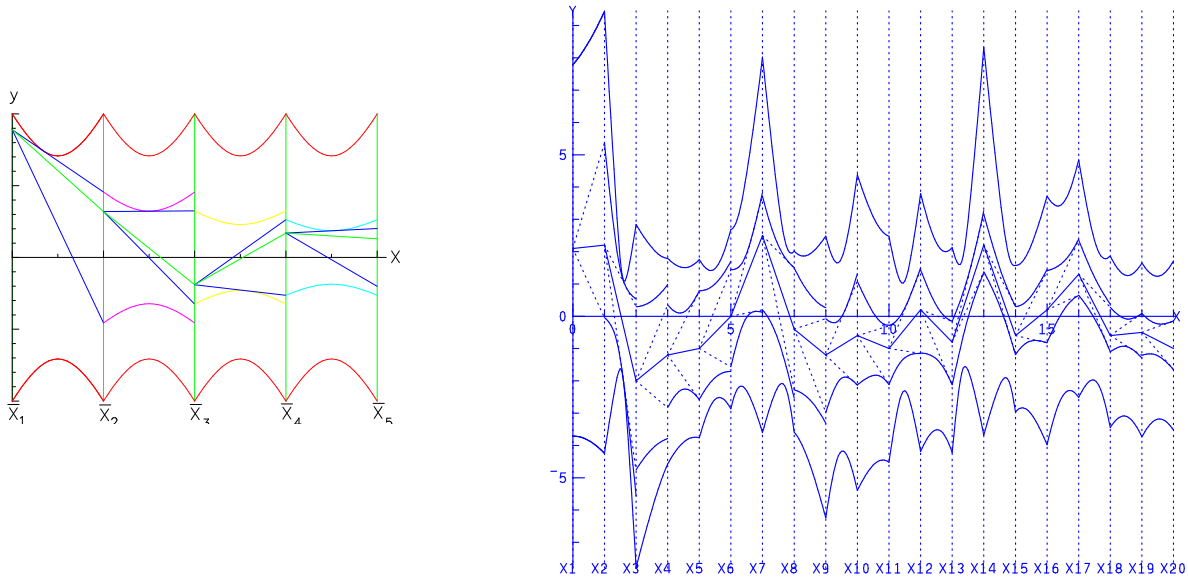


Figure 41: Interior point construction.

(Left)A sphere in 5-D showing the construction of an interior point (polygonal line). (Right) The general interior point (polygonal line) construction algorithm shown for a convex hypersurface in 20-D. polygonal line) and provide a feel for the local curvature. Note the narrow strips around $X13, X14, X15$ (as compared to the surrounding ones), indicating that at this state these are the critical variables where the point is bumping the boundary. A theorem guarantees that a polygonal line which is in-between all the intermediate curves/envelopes represents an interior point of the hypersurface and all interior points can be found in this way. If the polygonal line is tangent at anyone of the intermediate curves then it represents a boundary point, while if it crosses anyone of the intermediate curves it represents an exterior point. The later enables us to see, in an application, the first variable for which the construction failed and what is needed to make corrections. By varying the choice of value over the available range of the variable interactively, sensitive regions (where small changes produce large changes downstream) and other properties of the model can be easily discovered. Once the construction of a point is completed it is possible to vary the values of each variable and see how this effects the remaining variables. So one can do trade-off analysis in this way and provide a powerful tool for, Decision Support, Process Control and other applications. As new data becomes available the model can be updated with decisions being made based on the most recent information. This algorithm is used in the earlier example on a model for a country's economy shown in Figs. 34, 35.

Future

$\mathscr{S}$earching for *patterns* in a ||-coords display is what skillful exploration is about. If there are multivariate relations in the dataset the patterns *are there* though they may be covered by the overlapping polygonal lines and that is not all. Our vision is not multidimensional. We do not perceive a room which is 3-dimensional from its points which are 0-dimensional, but from the 2-dimensional planes which enclose

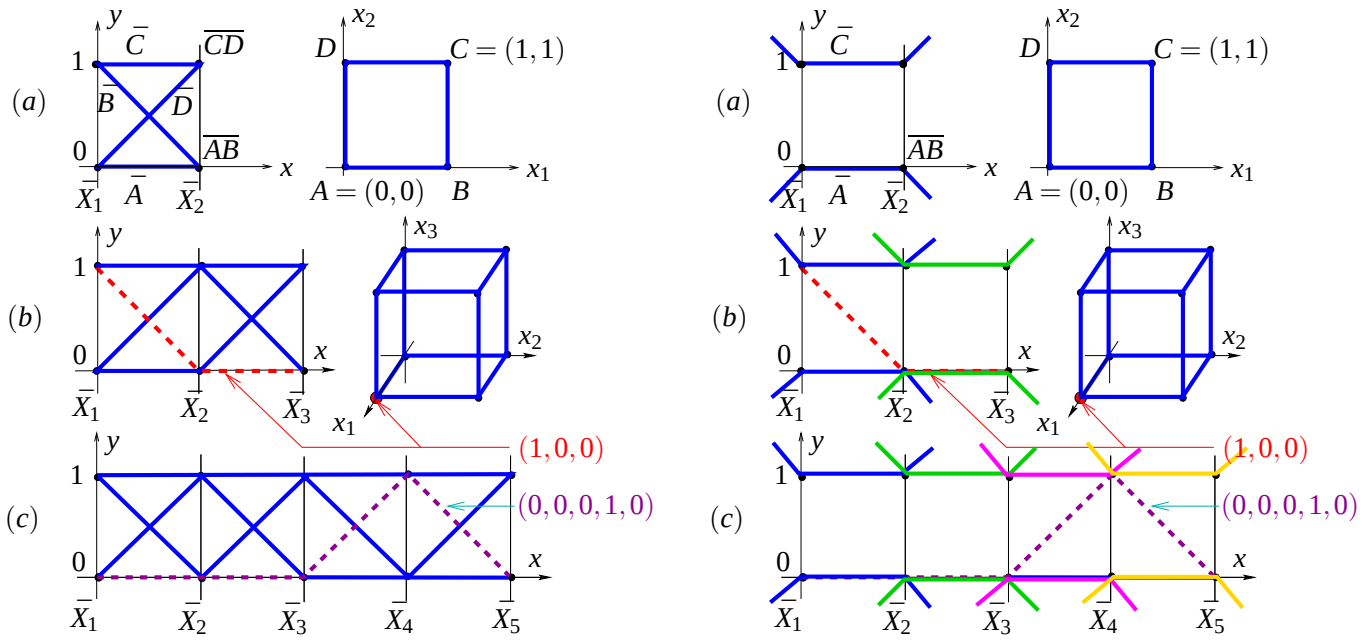


Figure 42: Square, cube and hypercube in 5-D on the left represented by their vertices and on the right by the tangent planes. Note the hyperbola-like (with 2 assymptotes) regions showing that the object is convex.

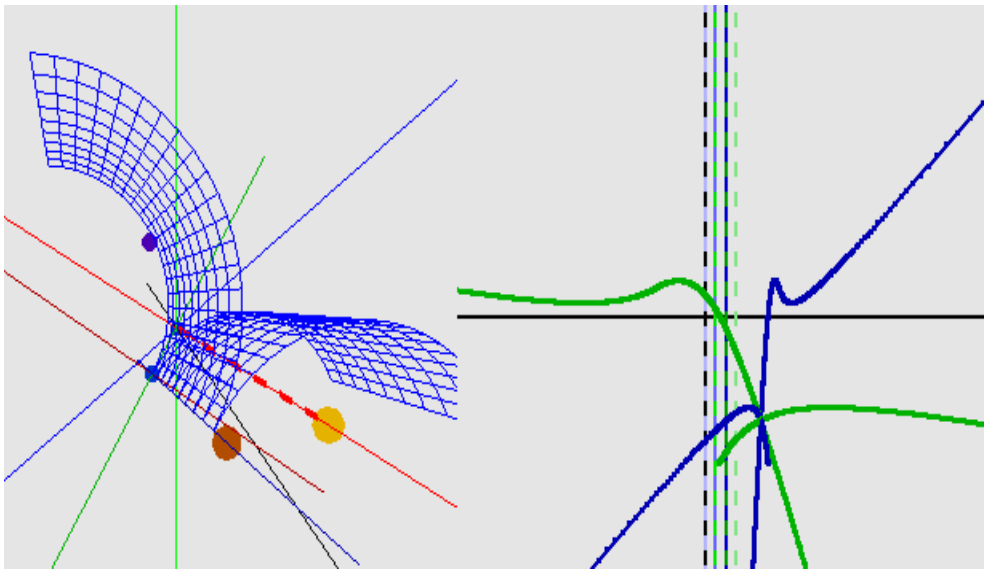


Figure 43: Developable surfaces are represented by curves. Note the two dualities *cusp* $\leftrightarrow$ *inflection point* and *bitangent plane* $\leftrightarrow$ *crossing point*. Three such curves represent the corresponding hypersurface in 4-D and so on.

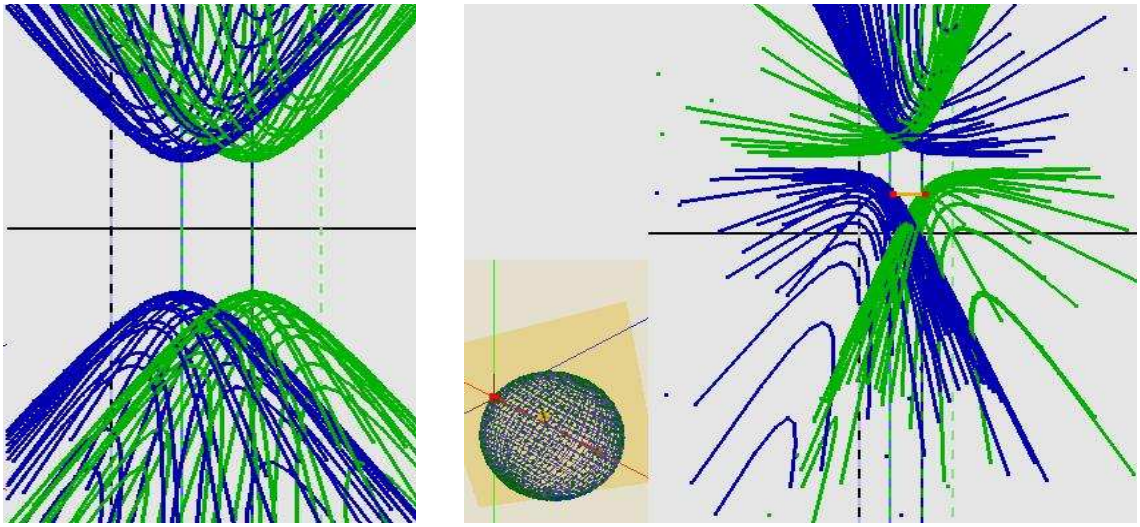


Figure 44: Representation of a sphere centered at the origin (left) and after a translation along the x_1 axis (right) causing the two hyperbolas to rotate in opposite directions. Note the *rotation* $\leftrightarrow$ *translation* duality. In N-D a sphere is represented by $(N-1)$ such hyperbolic regions — pattern repeats as for hypercube above.

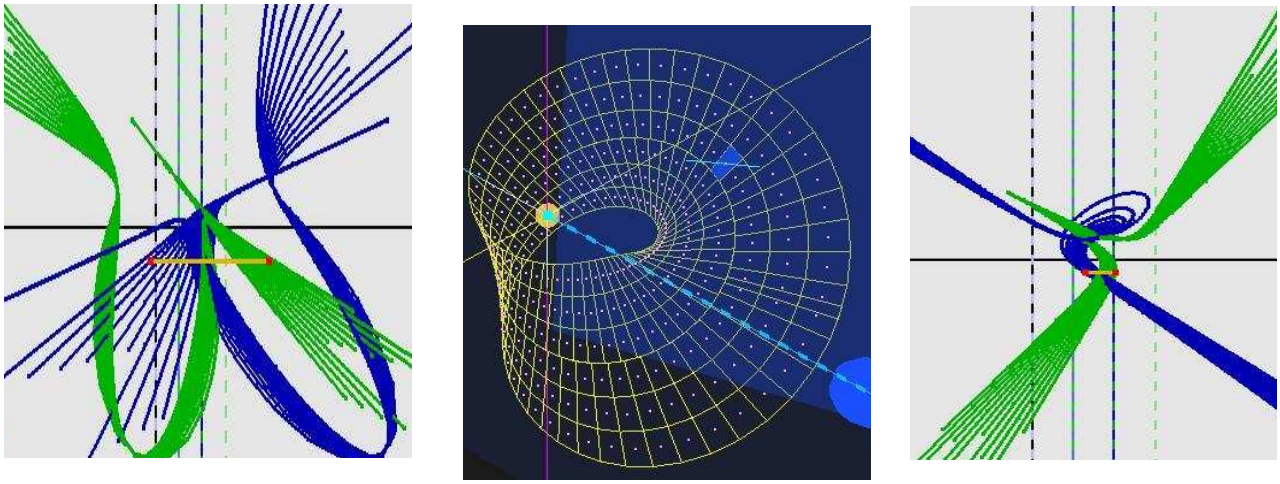


Figure 45: Möbius strip and its representation for two orientations. The two cusps on the left show that it corresponds to an "inflection-point in 3-D" – see the duality in Fig 43. The curves tending to infinity in the same direction upwards and downwards show that it is closed.

and define it. The recursive construction algorithm does exactly that for the visualization of p-dimensional objects from their $p-1$-dimensional subsets; one dimension less. We advocate including this algorithm within our armory of interactive analysis tools. Whatever p-dimensional relations exist are revealed by the pattern from the representation of the tangent hypeplanes of the corresponding hypersurface. The polygonal lines are completely discarded for the *relation is concentrated in the pattern*: Linear relations

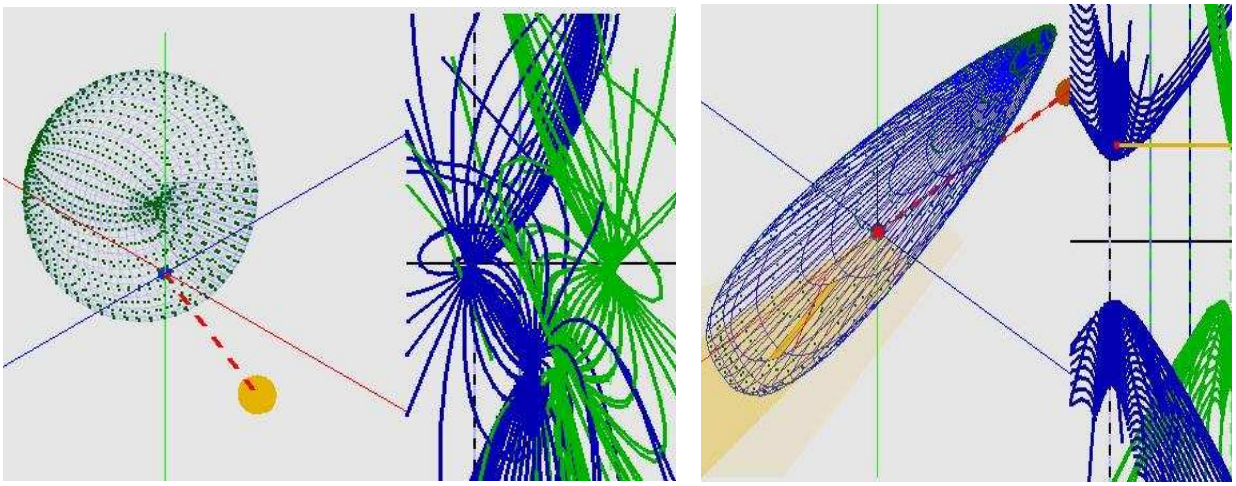


Figure 46: Representation of a surface with 2 “dimples” (depressions with cusp) which are mapped into “swirls” and are **all** visible. By contrast, in the perspective (left) one dimple is hidden. On the right is a convex surface represented by hyperbola-like (having two assymptotes) regions.

into points, proximate coplanarity into convex polygons, quadrics into conics and so on. Note further, again with reference to Figs. 36 and 37, that relational information resides at the *crossings*. What can be achieved for the representation of complex relations by patterns is exemplified by the pictures in Fig. 42 through Fig. 46. These are state of the art results showing what is achievable and how easily it generalizes to N-D. Can one imagine a higher dimensional convex surface or various kinds of non-convexities much less the *non-orientable* the Möbius strip. It is possible to do such a process on a dataset though at present it is computationally slow. The challenge is to speed up the algorithm for real-time response and thus break the gridlock of multidimensional visualization. There will still be work and fun for the multidimensional detectives visually separating and classifying the *no longer hidden* regions identifying complex multivariate relations which evaded us until now.

Bibliography

[1] N. Adrienko and G. Adrienko. *Constructing Parallel Coordinates Plots for Problem Solving*. in Proc. 1st Inter. Symp. on Smart Graphics, A.Butz, A. Krueger, P.Oliver, and M. Zhou (eds.), ACM Press 9-14, New York, 2001.

[2] E. W. Bassett. Ibm's ibm fix. *Industrial Computing*, 14(41):23–25, 1995.

[3] B. Bollobas. *Graph Theory*. Springer-Verlag, New York, 1979.

[4] H. Choi and H. Lee. *PCAV: Internet Attack Visualization in Parallel Coordinates, LNCS 3783, 454-466*. Springer-Verlag, New York, 2005.

[5] G. Conti. *Security Data Visualization*. No Starch Press, San Francisco, 2007.

[6] J. A Dykes, A.M. MacEachren, and Kraak M.J. (eds). *Exploring Geovisualization*. Elsevier, Amsterdam, 2005.

[7] J. A. Dykes and D. M. Mountain. Seeking structure in records of spatio-temporal behaviour: visualization issues, efforts and applications. *Comput. Statit.& Data Anal.*, 43 (Data Visualization II special ed.):581–603, 2003.

[8] R. Edsall. The parallel coordinate plot in action: Design and use for geographic visualization. *Comput. Statit.& Data Anal.*, 43-4:605–619, 2003.

[9] J. Eickemeyer. *Visualizing p-flats in N-space using Parallel Coordinates*. Ph.D. thesis, Dept. Comp. Sc., UCLA, 1992.

[10] S. El Mejdani, R. Egli, and F. Dubeau. Old & new straight-line detectors, descriptions & comparisons. *Pattern Recognition*, 41:1845–66, 2008.

[11] A. Ellis, G. Dix. Enabling automatic clutter reduction in parallel coordinates. *Trans. on Vis. & Comp. Grap.*, 12-5:717–724, 2006.

[12] N. Elmquist, J. Stasko, and P. Tsingas. Datameadow: A visual canvas for analysis of large-scale multivariate data. *Proc. IEEE Vast Conf.*, to appear, 2007.

[13] G. Fayad, U. M. Piatesky-Shapiro, P. Smyth, and R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, Cambridge Mass., 1996.

[14] M. Friendly and al. *Milestones in Thematic Cartography*. www.math.yorku.ca/scs/SCS/Gallery/milestones/, 2005.

[15] Y. H. Fua, M. O. Ward, and E. A. Rundensteiner. *Hierarchical Parallel Coordinates for Exploration of Large Datasets – Proc.of Conf. on Vis., 43-50*. IEEE Comp. Soc. Press, Los Alamitos, CA, 1999.

[16] C. Gennings, K. S. Dawson, W. H. Carter, and R. H. Myers. Interpreting plots of a multidimensional dose-response surface in parallel coordinates. *Biometrics*, 46:719–35, 1990.

[17] A. Goel. *Vizcraft: A Multidimensional Visualization Tool for Aircraft Configuration Design*. IEEE Vis Conf., 425-8, 1999.

[18] F. Harary. *Graph Theory*. Addison-Wesley, Reading, Mass., 1969.

[19] H. Hauser. *Parallel Sets: Visual Analysis of Categorical Data*. Proc. IEEE Infovis, 2005.

[20] J. Helly. *Applications of Parallel Coordinates to Complex System Design and Operation*. Proc. Nat. Comp. Grap. Assoc. vol.III, 541-546, 1987.

[21] P. Hertzog. *Visualizations to Improve Reactivity towards Security Incidents inside Corporate Networks, SIGSAC Conf. Proc., 95-102*. ACM New York, 2006.

[22] Q. Huanin, C. Wing-Yi, X. Anbang, C. Kai-Lun, L. Kai-Hon, and G. Ping. Visual analysis of the air pollution problem in hong kong. *Trans. on Vis. & Comp. Grap.*, 13-6:1408–15, 2007.

[23] A. Inselberg. Visual data mining with parallel coordinates. *Comput. Statit.*, 13-1:47–64, 1998.

[24] A. Inselberg. *Parallel Coordinates : **VISUAL** Multidimensional Geometry and its Applications*. Springer, New York, 2009.

[25] A. Inselberg and T. Avidan. *The* Automated *Multidimensional Detective, in Proc. of IEEE Information Visualization '99, 112-119*. IEEE Comp. Soc., Los Alamitos, CA, 1999.

[26] A. Inselberg and P. L. Lai. *Visualizing Families of Close Planes, 66*. Proc. 5th Asian Conf. on Stat., Hong Kong, 2005.

[27] J. Johansson, P. Ljung, M. Jern, and M. Cooper. *Revealing Structure within Clustered Parallel Coordinates Displays, Proc. IEEE Infovis, 125-132*. IEEE Comp. Soc., Los Alamitos, CA, 2005.

[28] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure in visualizations of dense 2d and 3d parallel coordinates. *Inf. Vis. Pal-Grave-Mcmillan Journ.*, 5:125–136, 2006.

[29] C. Jones. *Visualization and Optimization*. Kluwer Academic Publishers, Boston, 1996.

[30] M. Kipouros, T. Mleczko and M. Savill. *Use of Parallel Coordinates for Post-Analyses of Multiobjective Aerodynamic Design Optimization in Turbomachinery (to appear)*. Proc. 4th AIAA Meet., 2008.

[31] D.E. Lucas. *Recréations Mathematiques, vol II*. Gauthier Villars, Paris, 1892.

[32] J. Martin, G. Kennedy. *Using Curves to Enhance Parallel Coordinates Visualization*. Proc. IEEE Infovis, London, IEEE Comp. Soc., Los Alamitos, CA, 10-16, 2003.

[33] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[34] J. Nakano and K. Honda. *Three-dimensional parallel coordinates plot used for variable selection*. Proc. COMPSTAT 2006, (eds. A. Rizzi, M. Vichi), Springer, 187-195, 2006.

[35] B. Pham, Y. Cai, and R. Brown. *Visualization Techniques for Tongue Analysis in Traditional Chinese Medicine*. Proc. Med. Imag. SPIE 14-19, 2004.

[36] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, 1993.

[37] H. Rosenbaum, R. Schumann. *Chances and Limits of Progression in Visualization, in SimVis. Proc.* Sim. & Vis., Magdeburg, Germany, 2007.

[38] F. Rossi. *Visual Data Mining and Machine Learning, in 14th Europ. ESANN Proc., 251-264*. ESANN, Bruges Belgium, 2006.

[39] C. Schmid and H. Hinterberger. *Comparative Multivariate Vis. Across Conceptually Different Graphic Displays, in Proc. of 7th SSDBM*. IEEE Comp. Soc., Los Alamitos, CA, 1994.

[40] M. Tory, S. Potts, and T. Moller. A parallel coordinates style interface for exploratory volume visualization. *Trans. on Vis. & Comp. Grap.*, 11-1:71–80, 2005.

[41] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphic Press, Connecticut, 1983.

[42] E. R. Tufte. *Envisioning Information*. Graphic Press, Connecticut, 1990.

[43] E. R. Tufte. *Visual Explanation*. Graphic Press, Connecticut, 1996.

[44] A.R. Unwin, M. Theus, and H. Hofmann. *(Eds.) Graphics of Large Datasets*. Springer, New York, 2006.

[45] M.O. Ward. *XmdvTool: integrating multiple methods for visualizing multivariate data, Proc. IEEE Conf. on Vis., CA, 326-333*. IEEE Comp. Soc., Los Alamitos, CA, 1994.

[46] L. Yang. Pruning & visualizing generalized association rules in parallel coordinates. *IEEE Know. & Data Engr.*, 15(1):60–70, 2005.

[47] H. Ye and Z. Lin. Speed-up simulated annealing by parallel coordinates. *Euro J. of OR*, 173:59–71, 2006.