EUROGRAPHICS 2001

******

TUTORIAL

******

Data Mining and Visualization of High Dimensional Datasets

Alfred Inselberg[*], Multidimensional Graphs Ltd [†]
&
School of Mathematical Sciences
Tel Aviv University, Israel
aiisreal@math.tau.ac.il

Manchester, UK – September 4, 2001

[*]Senior Fellow San Diego SuperComputing Center, California, USA
[†]36A Yehuda Halevy Street, Raanana 43556, Israel

Abstract

The desire to augment our 3-dimensional perception and the need to understand multivariate problems spawned several multidimensional visualization methodologies. Starting from early successes of visualization, like Dr. J. Snow's dot map in 1854 showing the connection of cholera to some water pumps in London, Scatter plots, Chernoff faces, Andrews plots, Projection Pursuit, Perceptualization of data, Data density, Trees and Castles, Kinematic displays, Bertin Permutation Matrices and other multivariate visualization techniques have been developed (see [12], [13] for a beautiful non-technical review). However, they are all limited to viewing small datasets having a few variables.

Here we focus on Parallel Coordinates which is a new methodology enabling the unambiguous visualization of multidimensional geometry and, in turn, multivariate relations. Parallel Coordinates is introduced and rigorously developed. Relations among N real variables are mapped uniquely into subsets of 2-space having geometrical properties enabling the visualization of the corresponding N-dimensional hypersurfaces.

After the basic representation results, associated algorithms for constructions, intersections, transformations, containment queries, proximity and others will be given. The development is interlaced with applications of the relevant results starting with demonstrations of Data Mining on real datasets (i.e. Feature extraction from LandSat data, Financial, Process Control, Pilot Selection, Raising the Yield and Quality of VLSI chips, and others). They are followed by Collision Avoidance Algorithms for Air Traffic Control which are based on the representation of lines in multidimensional space. The detection of coplanar points and the representation of planes and hyperplanes lead to some applications in Computer Vision, Geometric Modeling and elsewhere. More examples of Visual Data Mining are given. An efficient geometric classifier algorithm is motivated and it will be demonstrated on some challenging datasets. Finally, the representatioin of curves and hypersurfaces is taken up together with interactive applications to Process Control, Instrumentation and Heuristic Optimization. Nonlinear VISUAL models in terms hypersurfaces are constructed from data and used interactively for Decision Support, Sensitivity Analysis, studying feasibility and effect of contraints as well as trade-off analysis.

Getting Started

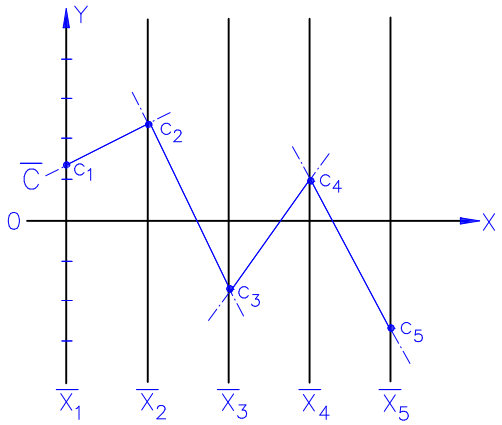

Figure 1: The polygonal line $\bar{C}$ represents the point $C = (c_1, c_2, c_3, c_4, c_5)$.

In principle, a large number of (even infinitely many) axes can be placed and be *seen* parallel to each other. The representation of points is deceptively simple and much development and additional ideas are needed to enable the visualization of *multivariate relations* or equivalently multidimensional objects. Specifically, the rep-

resentation of a p-dimensional object $2 \leq p$ will be obtained from it's $(p-1)$-dimensional components. For example, the representation (i.e. image) of a line is obtained from the *points* on the line, and is in fact the envelope of the polygonal lines representing the points. Next, the representation of a plane in R^3 is obtained from the representation of the *lines*, rather than the points, it contains. This leads to a recursion which turns out to work spendidly; but we are getting ahead of ourselves.

So far the most popular application is in (Visual) Data Mining, because parallel coordinates (abbr.||-coords) transform multivariate relations into distinct 2-D patterns [1]. Several software tools starting with EDA (Chomut [2]), followed by Finsterwalder[3], VisuLab(Hinterberger[10]), ExplorN (Bolorforoush (88), Carr (92)), WinViZ (Eickemeyer), VisDB(Keim [7]), Xmdv(Ward[8],[15]), XGobi (Swayne, Cook, Buja, [11]), Strata(||-coords by Gleason), Diamond([9]), PVE (Inselberg, Adams, Hurwitz, Chatterjee), Influence Explorer (Tweedie, Spence [14]) and others include ||-coords. Notably, a major project by Eurostat, the European Union's Statistical Office, is underway to develop customized data exploration software based on ||-coords. This type of application [4] hinges on :

- an informative display of the data,
- good choice of queries, and
- skillful *interaction* by the user of the display in search of patterns corresponding to relationships among the variables in the data.

Recently, it has become possible to automate the discovery process making the methodology available to a greater number of users [6].

The Plane with Parallel Coordinates

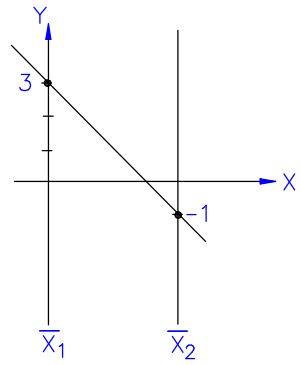

Figure 2: Points on the plane are represented by lines.

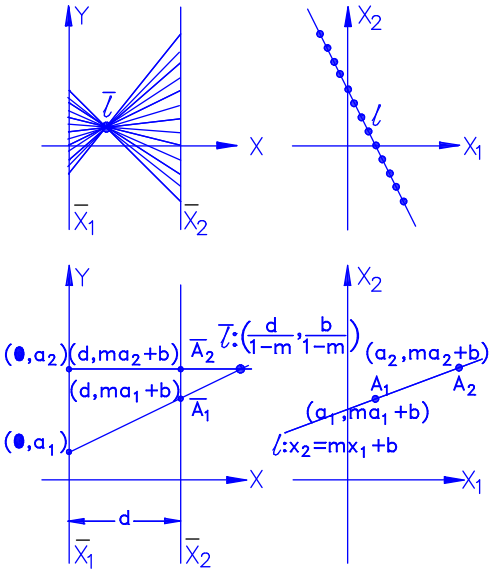

Figure 3: In the plane parallel coordinates induce a point $\longleftrightarrow$ line duality.

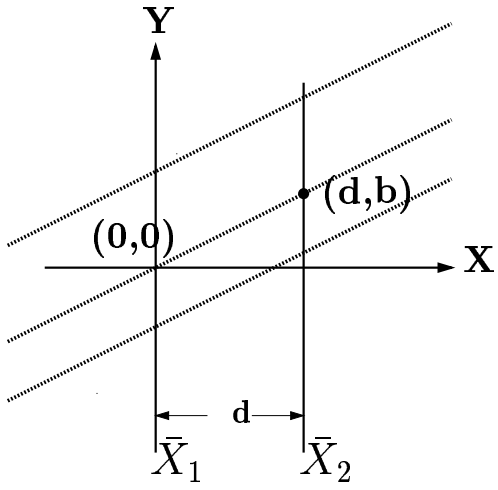

Figure 4: Lines representing points on the line $x_2 = x_1 + b$.

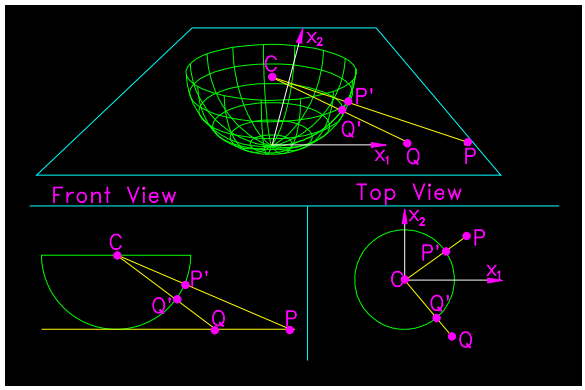

Figure 5: Model of the Projective Plane

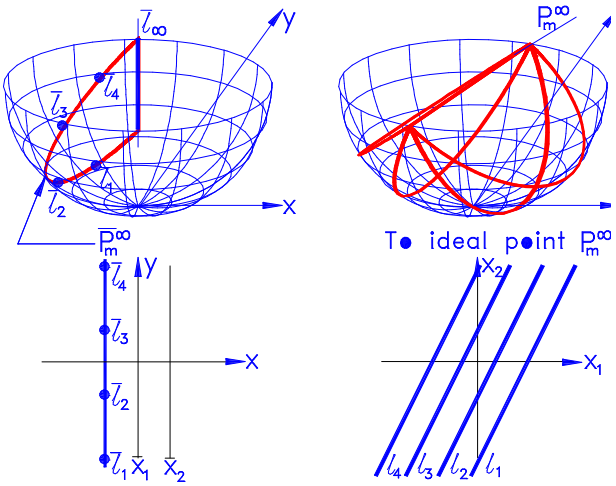

Figure 6: Under the duality parallel lines map into points on the same vertical line. On the projective plane model, the great semi-circles representing the lines share the same diameter – i.e. they have the same ideal point

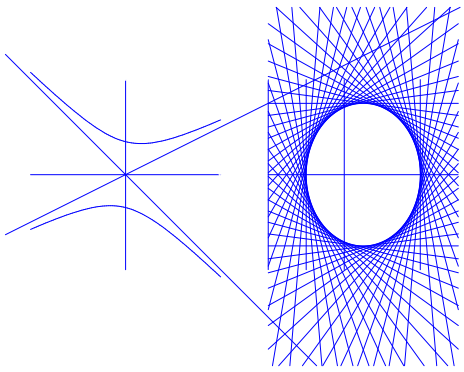

Figure 7: Hyperbola(point-curve) → Ellipse(line-curve) – Image Depends on orientation of Hyperbola.

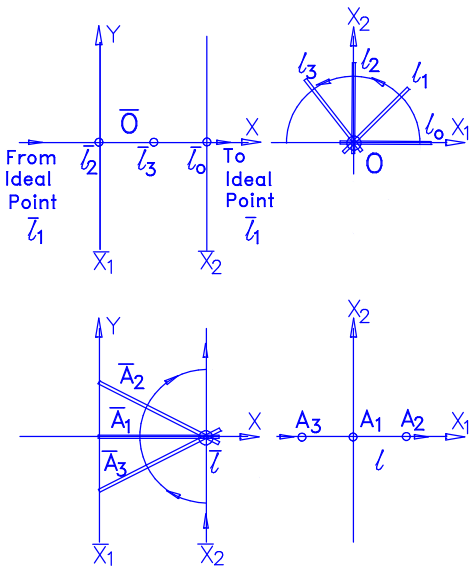

Figure 8: Duality between rotations and translations

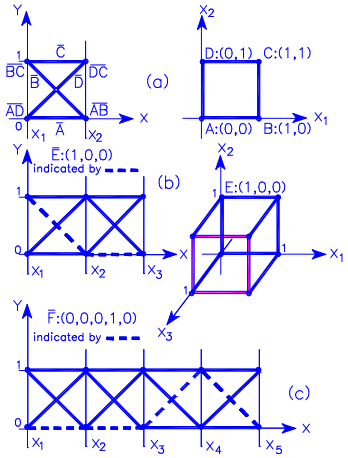

Figure 9: (a)Square,(b) 3-D cube (c) 5-D hypercube all with unit side

Multidimentional Lines

First of all let us agree on what we mean by a line ℓ in N-space and a good way is to specify ℓ as the set of points (given by N-tuples) which satisfy a set of N-1 linearly independent linear equations. After some manipulation, and with the exception of a few special cases (for rigorous treatment covering all the cases see [5]), such a set of equations can be put in the form :

$$\begin{aligned} \ell_{1,2} &: x_2 = m_2 x_1 + b_2 \\ \ell_{2,3} &: x_3 = m_3 x_2 + b_3 \\ &\dots \\ \ell_{i-1,i} &: x_i = m_i x_{i-1} + b_i \\ &\dots \\ \ell_{N-1,N} &: x_N = m_N x_{N-1} + b_N \end{aligned} \qquad (1)$$

that is each equation contains a pair of *adjacently* labeled variables. In the $x_{i-1}x_i$-plane the relation labeled $\ell_{i-1,i}$ is a line, and by our Line $\leftrightarrow$ Point duality which we have already found (eq. (3) in Chapter 1) it can be represented by a point $\bar{\ell}_{i-1,i}$. For convenience let us take the distance between each pair of adjacent axes one unit as shown in Fig. 10.
Since the y-axis is not coincident with the $\bar{X}_{i-1}$-axis, we need to translate the x-coord of the point representing the line (see eq. (2) and Fig. 2 in Chapter 1) by $(i-2)$. That is

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

or in homogeneous coordinates by :

$$\bar{\ell}_{i-1,i} = ((i-2)(1-m_i)+1, b_i, 1-m_i). \qquad (2)$$

Hence, there are $N-1$ such points for $i = 2, \dots, N$ which represent the N-D line ℓ. The fact that the indexing being an essential part of the representation is, often, not appreciated and causes misconceptions.This is in fact the first instance where we see the need for the indexing included in the representation mapping $\mathcal{J}$ (eq. (1) in the Introduction). Without the indexing, the points could represent anyone of $(N-1)!$ lines corresponding to all the possible ways the $(N-1)$ *independent* pairs of indices can be attached to the points. Since the display space is at a premium the indexing is usually not included in the picture, but it must always be accessible from some database. As we will see, there a number of construction algorithms based on this representation, *and crucially depending on the indexing.*

A polygonal line for which the linear portion containing the $(i-1), i$-segment passes through the point $\bar{\ell}_{i-1,i}\ \forall\ i = 2, \dots, N$ (another case of needing to be careful with the indices) represents a point on the line ℓ since the adjacent pair of y-coordinates of its vertices on the $\bar{X}_{i-1}$ and $\bar{X}_i$-

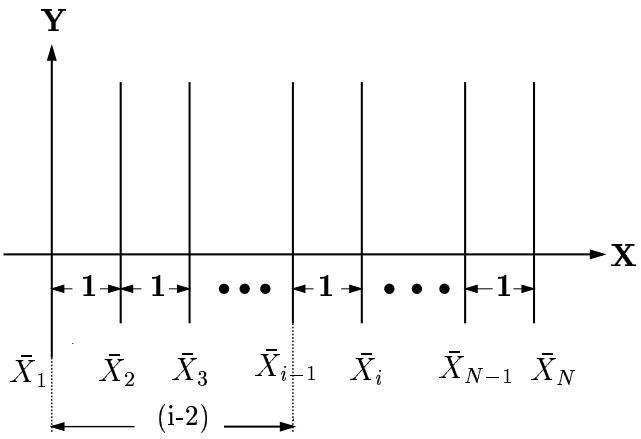

Figure 10: Spacing between adjacent axes is 1 unit.

axes simultaneously satisfy Equation (1). Such is the case in Fig. 11 where several polygonal lines representing points on an interval of a line in 10-D are shown. For example, the point of intersection of the polygonal lines between the $\bar{X}_2$ and $\bar{X}_3$-axes, is the point $\bar{\ell}_{2,3}$. All the nine points, representing that 10-dimensional lines can be seen (or constructed) with their horizontal positions depending on the first coordinate of eq. (2). Do not be mislead by the fact that all of the $\bar{\ell}$'s except $\bar{\ell}_{67}$ are shown between their corresponding axes. This is due to the choice of $m_i \leq 0$ made for display convenience only.

The indexed points representing an N-dimensional line have a striking and very useful property. For $i \neq j \neq k \in [1, 2, ..., N]$ the three points $\bar{\ell}_{i,j}, \bar{\ell}_{j,k}, \bar{\ell}_{i,k}$ are always collinear. This can be seen by considering two points $P_r = (p_{1,r}, \ldots, p_{N,r})$, $r = 1, 2$ on ℓ and their projections on the x_i, x_j, x_k three-space as shown in Fig. 12.

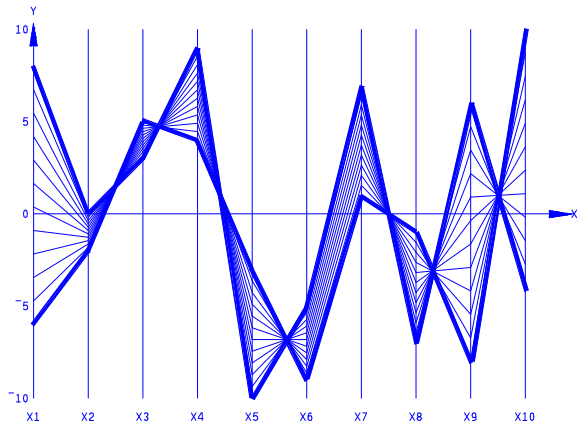

Figure 11: Line interval in 10-D. Heavier polygonal lines represent end-points. The nine points where the polygonal lines intersect represent the complete line.

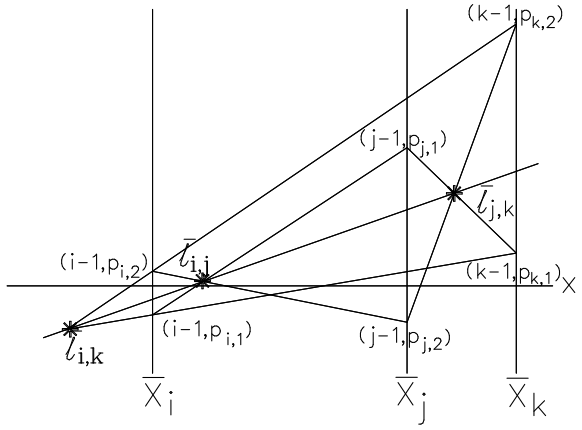

Figure 12: The Collinearity for the 3 points $\bar{l}_{i,j}, \bar{l}_{j,k}, \bar{l}_{i,k}$. The two triangles are in perspective with respect to the ideal point in vertical direction. The y-axis is offscale.

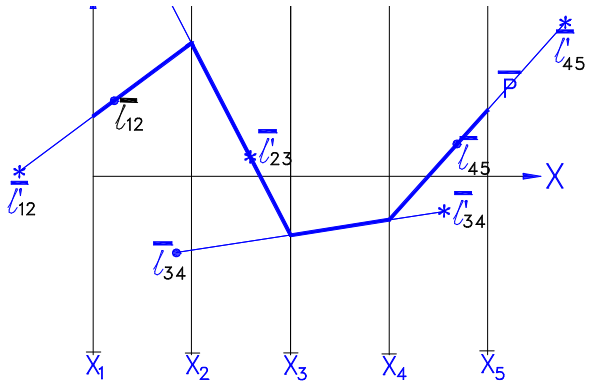

Figure 13: Rotation of a line in R^5 about one of its points

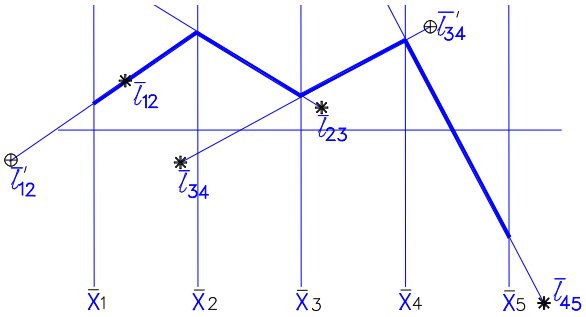

Figure 14: Two intersecting lines in R^5

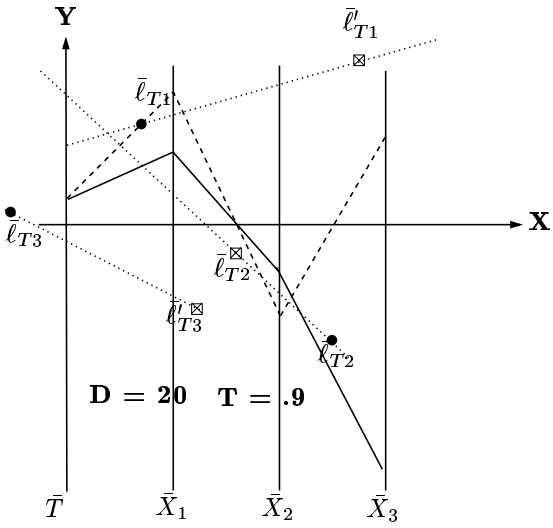

Figure 16: Non-intersection between two lines in 4-D. Here the minimum distance is 20 and occurs at time = .9. Note the maximum gap on the $\bar{T}$-axis formed by the lines joining the $\bar{\ell}$'s with the same subscript. The polygonal lines representing the points where the minimum distance occurs are shown.

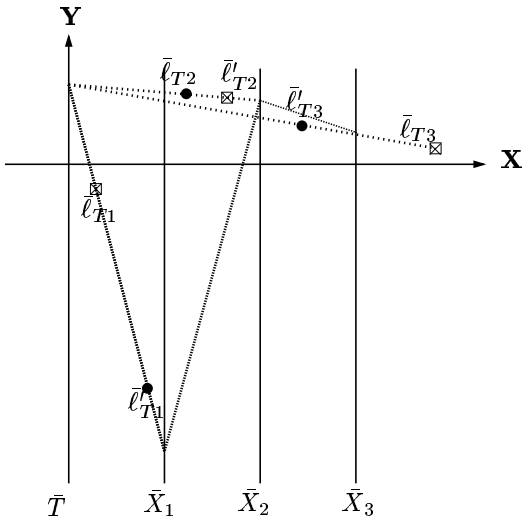

Figure 15: Intersection, for the base-variable line description, of two lines in 4-D. This provides the space **and time** coordinates of the place where two particles moving with constant velocity collide.

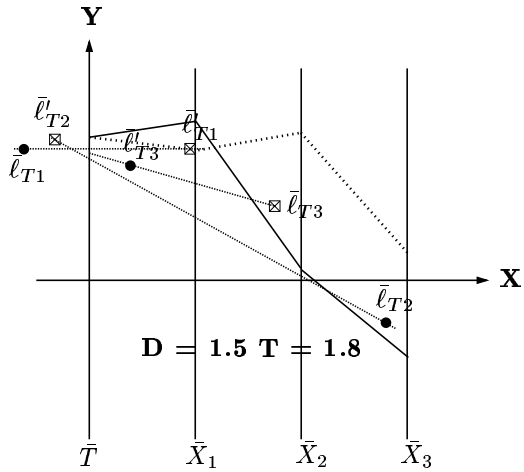

Figure 17: Near intersection between two lines in 4-D. Here the minimum distance is 1.5 and occurs at time = 1.8. Note the the diminished maximum gap on the $\bar{T}$-axis formed by the lines joining the $\bar{\ell}$'s with the same subscript. The polygonal lines representing the points where the minimum distance occurs are shown.

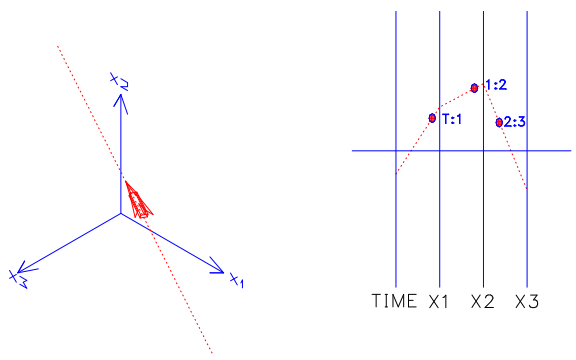

Figure 18: Path (left) and Trajectory (right) of an aircraft. In ||-coords the position at any given **time** may be displayed.

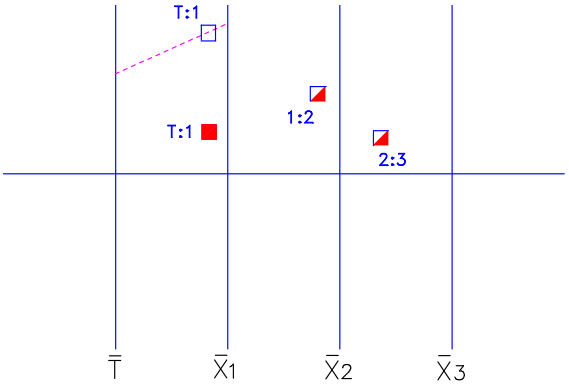

Figure 19: Two aircraft flying on the same path since their **1:2** and **2:3** points coincide. They have a constant separation their velocity being the same since **T:1** points have the same x-coordinate.

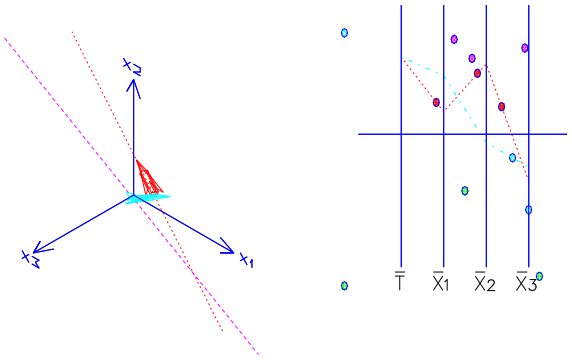

Figure 20: Closest approach of two aircraft. The time and closest positions are clearly seen in ||-coords. Appearances can be misleading in a 3-D (near perspective) display where the aircraft appear to be nearly colliding. It is even more uninformative in 2-D where only a projection is displayed.

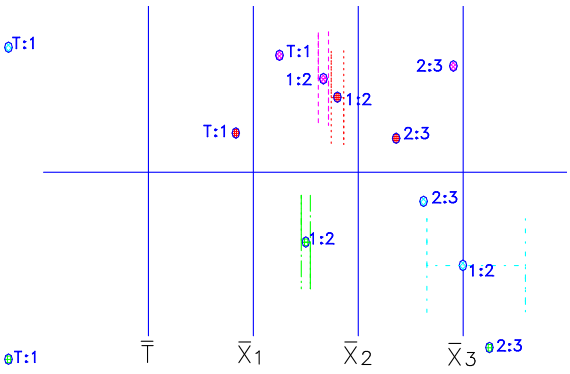

Figure 21: Transforming deviations in heading (angle) to lateral deviations

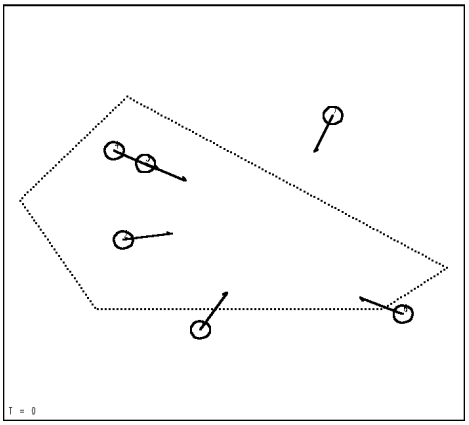

Figure 22: Six aircraft flying at the same altitude. These positions are at a certain time (taken as 0 seconds and shown on the left-hand-corner). Circles centered at each aircraft are the protected airspaces with the diameter being the minimum allowable separation. The arrows represent the velocities.

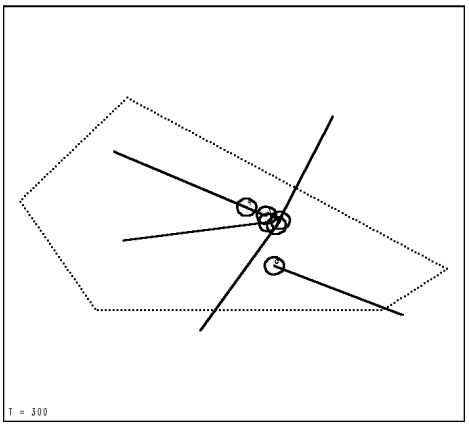

Figure 23: Conflicts, indicated by overlaping circles, within the next 5 minutes.

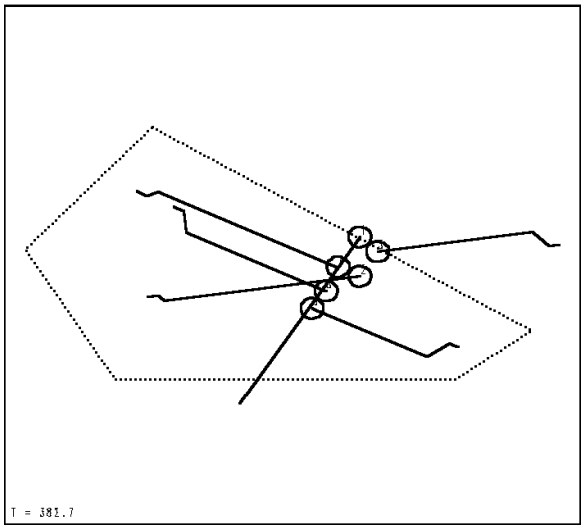

Figure 24: Conflict resolution with parallel-offset maneuvers. Three pairs of tangent circles.

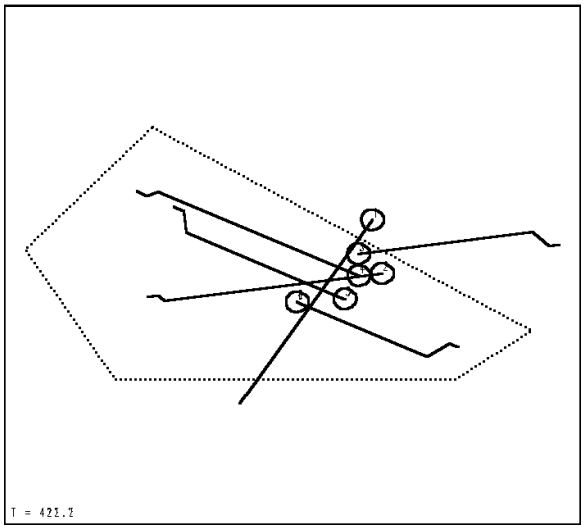

Figure 25: A triple tangency

Hyperplanes in R^N

$\mathcal{A}$ hyperplane in R^N can be translated to one which contains the origin, that is an $N-1$-dimensional linear subspace of R^N. Since R^{N-1} can be represented in ||-coords by $N-1$ vertical lines and a polygonal line representing the origin of the ||-coordinate system. Therefore, it is reasonable to expect a similar representation for hyperplanes in R^N.

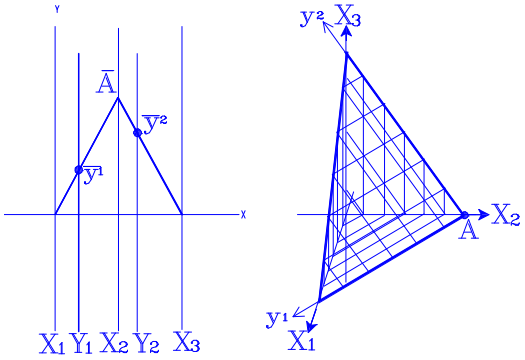

Figure 26: A plane π in 3-D can be represented by two vertical lines and a polygonal line representing one of the points of π.

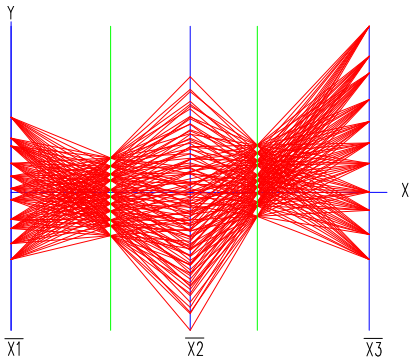

Figure 27: A set of coplanar points in R^3, note the two vertical lines pattern.

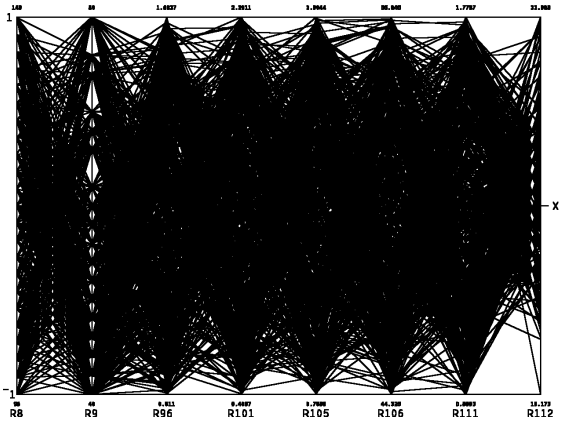

Figure 28: Industrial data. Note pattern between the R111 and R112 axes.

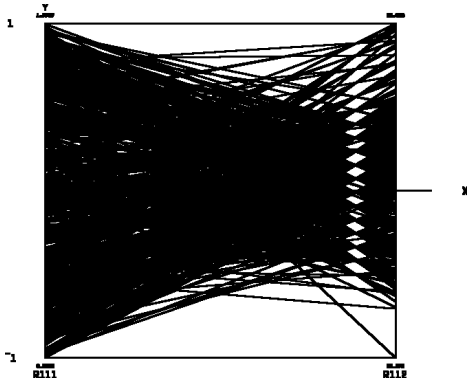

Figure 29: Industrial data with magnified portion between the R111 and R112 axes.

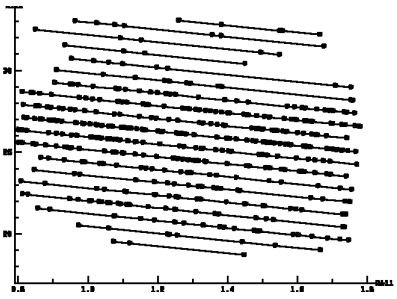

Figure 30: Industrial data showing the linear relation between R111, R112 and an unknown parameter.

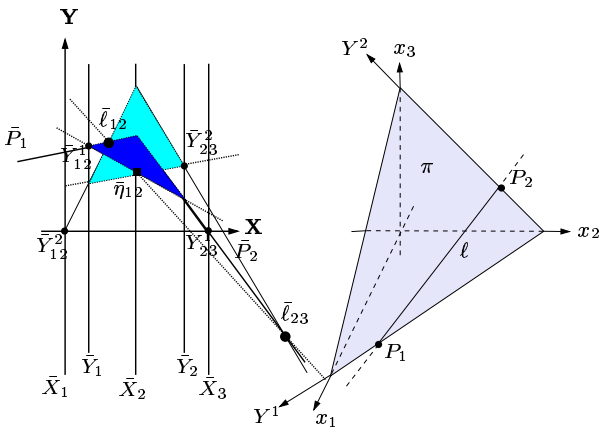

Figure 31: A line ℓ on a plane π is represented by one point $\bar{\eta}_{12}$ in terms of the planar coordinates $\bar{Y}_1$ and $\bar{Y}_2$ which is collinear with it's two point $\bar{\ell}_{12}$ and $\bar{\ell}_{23}$.

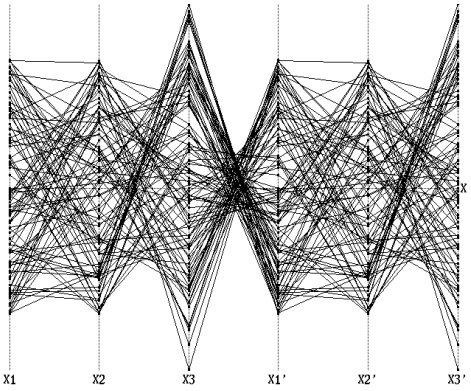

Figure 33: On the first 3 axes a set of polygonal lines representing a randomly sampled set of coplanar points in R^3 is shown.

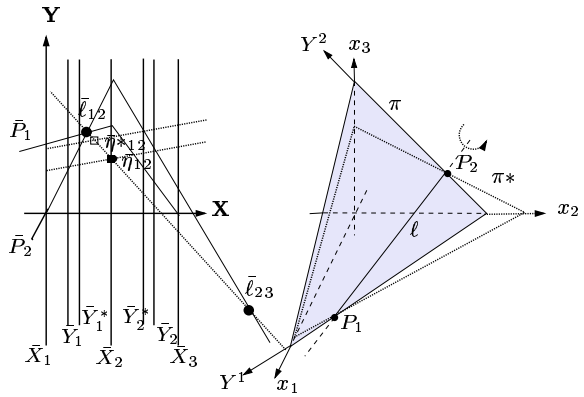

Figure 32: Rotation of a plane about a line $\leftrightarrow$ Translation of a point along a line.

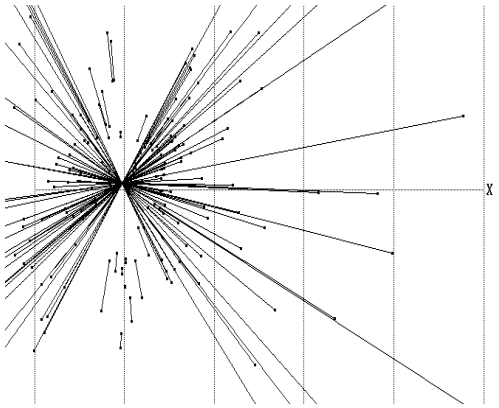

Figure 34: Coplanarity! The set of pairs of points representing lines on a plane is a pencil of lines on a point.

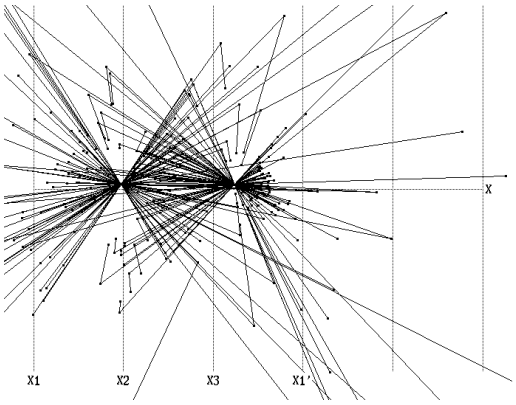

Figure 35: The plane π represented by two points

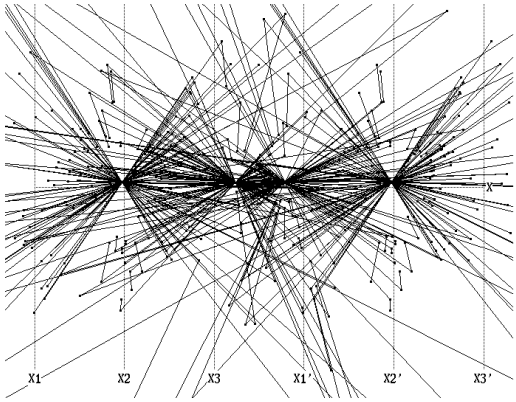

Figure 37: Four point representation of a plane π – 2 of them are redundant.

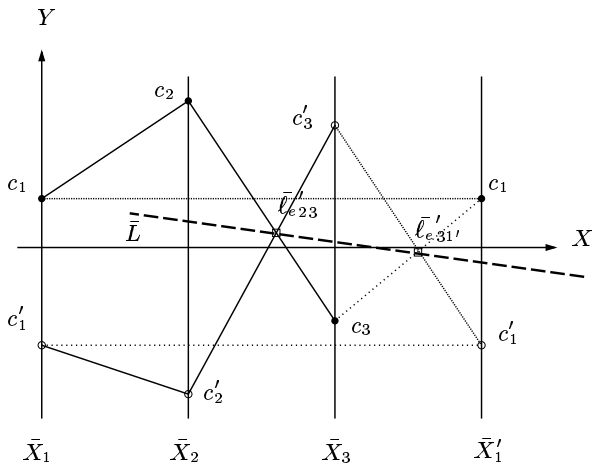

Figure 36: Transfering the values from the $\bar{X}_1$ to the $\bar{X}_{1'}$-axis.

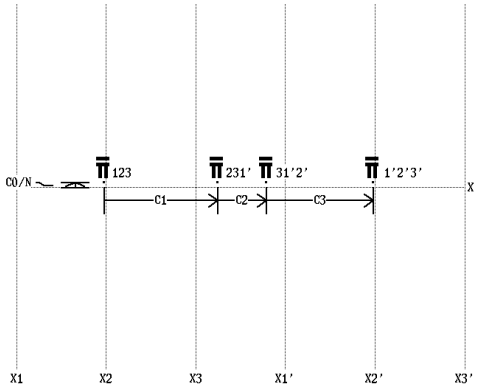

Figure 38: The distances between adjacent points are equal to the coefficients of π : $c_1x_1 + c_2x_2 + c_3x_3 = c_o$. That is the equation of the plane can be read from the picture!

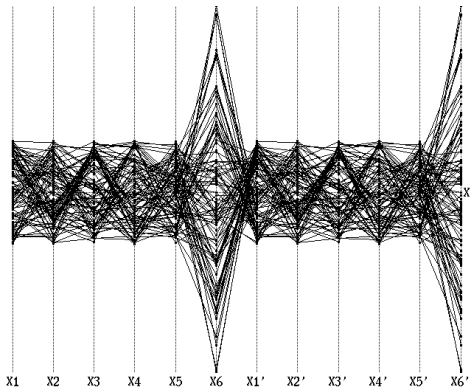

Figure 39: Polygonal lines on the $\bar{X}_1$ through $\bar{X}_6$ axes representing randomly selected points on a 5-flat $\pi^5 \subset R^6$

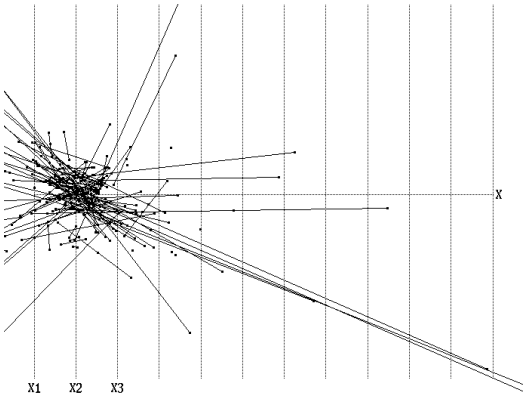

Figure 40: Portions, $\bar{\pi}^{1_i}_{12}$, , $\bar{\pi}^{1_i}_{23}$, of the 1-flats of π^5 constructed from the polygonal lines shown in Fig. 39. No pattern is evident.

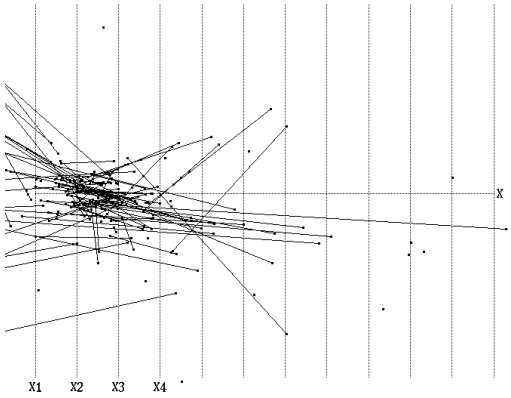

Figure 41: Portions, $\bar{\pi}^{2_i}_{123}$, , $\bar{\pi}^{2_i}_{234}$, of the 2-flats of π^5 constructed from the polygonal lines joining $\bar{\pi}^{1_i}_{12}$, $\bar{\pi}^{1_i}_{23}$, $\bar{\pi}^{1_i}_{34}$.

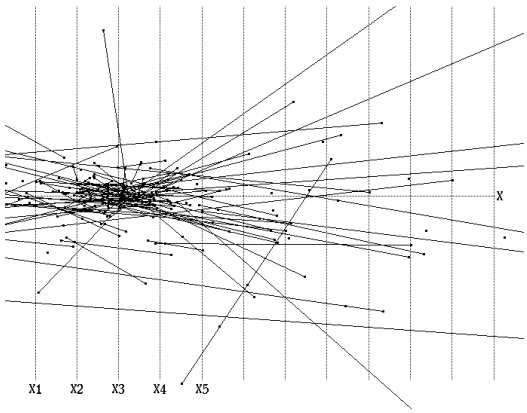

Figure 42: Portions, $\bar{\pi}^{3_i}_{1234}$, , $\bar{\pi}^{3_i}_{2345}$, of the 3-flats of π^5 constructed from the polygonal lines obtained by joining $\bar{\pi}^{2_i}_{123}$, $\bar{\pi}^{2_i}_{234}$, $\bar{\pi}^{2_i}_{345}$. Nothing yet ... but wait!

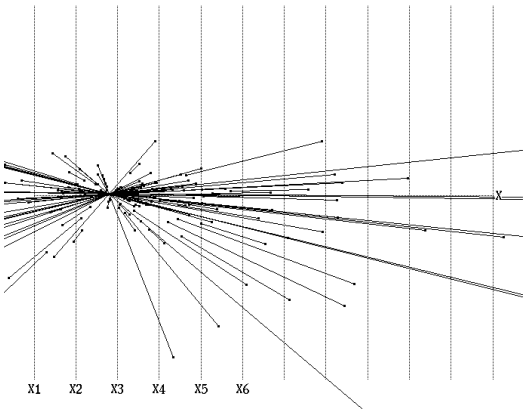

Figure 43: This is it! Portions, $\bar{\pi}^{4_i}_{12345}$, , $\bar{\pi}^{4_i}_{23456}$, of the 4-flats of π^5 constructed from the polygonal lines joining $\bar{\pi}^{3_i}_{1234}$, $\bar{\pi}^{3_i}_{2345}$, $\bar{\pi}^{3_i}_{3456}$, showing that the original points whose representation is shown in Fig. 39 are on a 5-flat in R^6. The remaining points of the representation are similarly obtained.

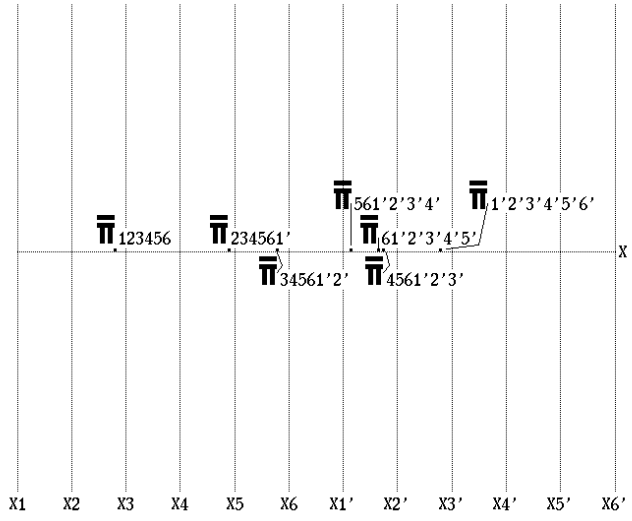

Figure 44: The full representation of π^5. The coefficients of it's equation are still the distances between consecutive by indices points.

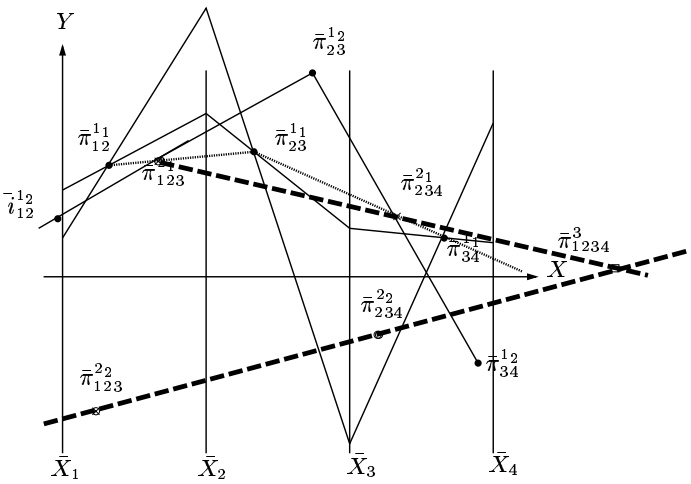

Figure 45: The recursive construction.

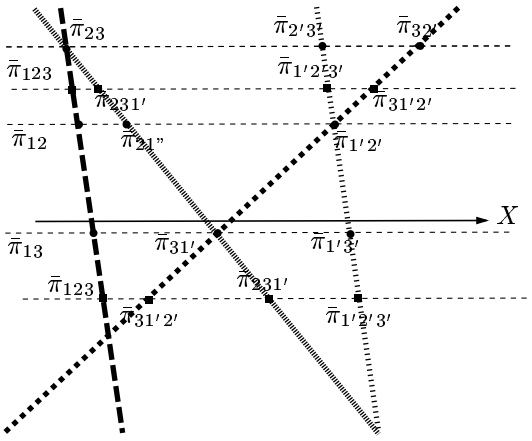

Figure 46: Rotation of a 2-flat (plane) about a 1-flat(line) in R^3 using the point representation. Notice the cases where some coefficients of the equation of the plane vanish.

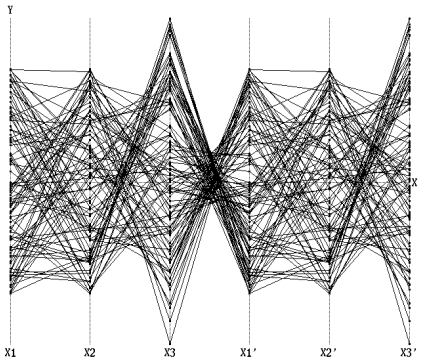

Figure 47: Polygonal lines representing a randomly selected set nearly coplanar points (i.e. on a "slab")

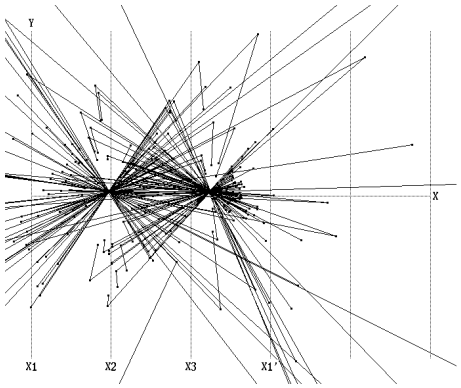

Figure 48: Representation of lines formed from the points shown in Fig. 47. The pattern for "near-coplanarity" is very similar to that obtained from coplanarity

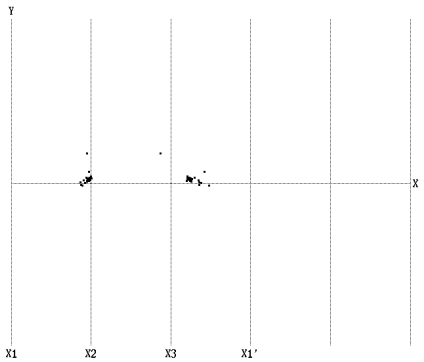

Figure 49: Close clusters from the intersection of the lines shown in Fig. 48

DATAMINING

Visual Data Mining

Selected Examples – an effort will be made to match the audience's interests

A Geometric Classifier

Classification is a basic task in data analysis and pattern recognition and an algorithm accomplishing it is called a **Classifier**. The input is a dataset P and a designated subset S. The output is a characterization, that is a set of conditions or rules, to distinguish elements of S from all other members of P. 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. This is the *geometrical* basis for the classifier presented here. The algorithm accomplishing this entails:

- ♢ use of an efficient "wrapping" algorithm to enclose the points of S in a hypersurface S_1 containing S and typically also some points of $P - S$; so $S \subset S_1$, of course such an S_1 is not unique.

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

- ♢ 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 $S \subset (S_1 - S_2) \cup S_3$,

- ♢ 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.

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 the output is a description of the hypersurface containing S the rule is given in terms of the minimum number of variables needed to describe S *without loss of information.* Unlike other methods, like the Principal Component Analysis (PCA), the classifier discards only the redundant variables. It is important to clarify this point. A subset S of a multidimensional set P is not necessarily of the same dimensionality as P. So the classifier finds the dimensionality of S in terms of the original variables and retains only those describing S. That is, it finds the *basis* in the mathematical sense of the smallest subspace containing S, or more precisely the current approximation for it. This basis is the minimal set M_r of variables needed to describe S. We call this dimensionality **selection** to distinguish it from dimensionality *reduction* which is usually done *with* loss of information. Retaining the original variables is important in the applications where the domain experts have developed intuition about the variables they measure. The classifier presents M_r *ordered according to a*

criterion which optimizes the clarity of separation. This may be appreciated with the example provided in the attached figure, in addition.

The implementation allows the user to 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 would be useful to have a rule involving such variables that are "accessible" in a meaningful way. There are also two options available :

- either minimize the number of variables used in the rule, or
- minimize the number of steps, in terms of the unions and (relative) complements, in the rule.

The classifier provides:

- 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 is data and "holes" with no data; that can provide significant insights to the domain experts,

The dataset chosen to illustrate has two classes to be distinguished consisting of pulses measured on two types of neurons in a monkey's brain (poor thing!). There are 600 samples with 32 variables. Remarkably, convergence was obtained and required only 9 of the 32 parameters. The resulting ordering shows a striking separation. In the attached figure the first pair of variables x_1, x_2 originally given is plotted showing no separation. In the adjoining plot the best pair x_{11}, x_{14}, as chosen by the classifier's ordering, shows remarkable separation. The result shows that the data consists of two "banana-like"[1] clusters in 9-D one (the complement in this case) enclosing the other (class for which the rule was found). Note that the classifier can actually describe highly complex regions. It can build and "carve" the cavity shown. It is no wonder that separation attempts in terms of hyperplanes or nearest-neighbor techniques can fail badly on such datasets. The rule gave an error of 3.92 % using train-and-test with 66 % of the data for training).

[1] Perhaps the monkey was dreaming about bananas during this fateful experiment ...

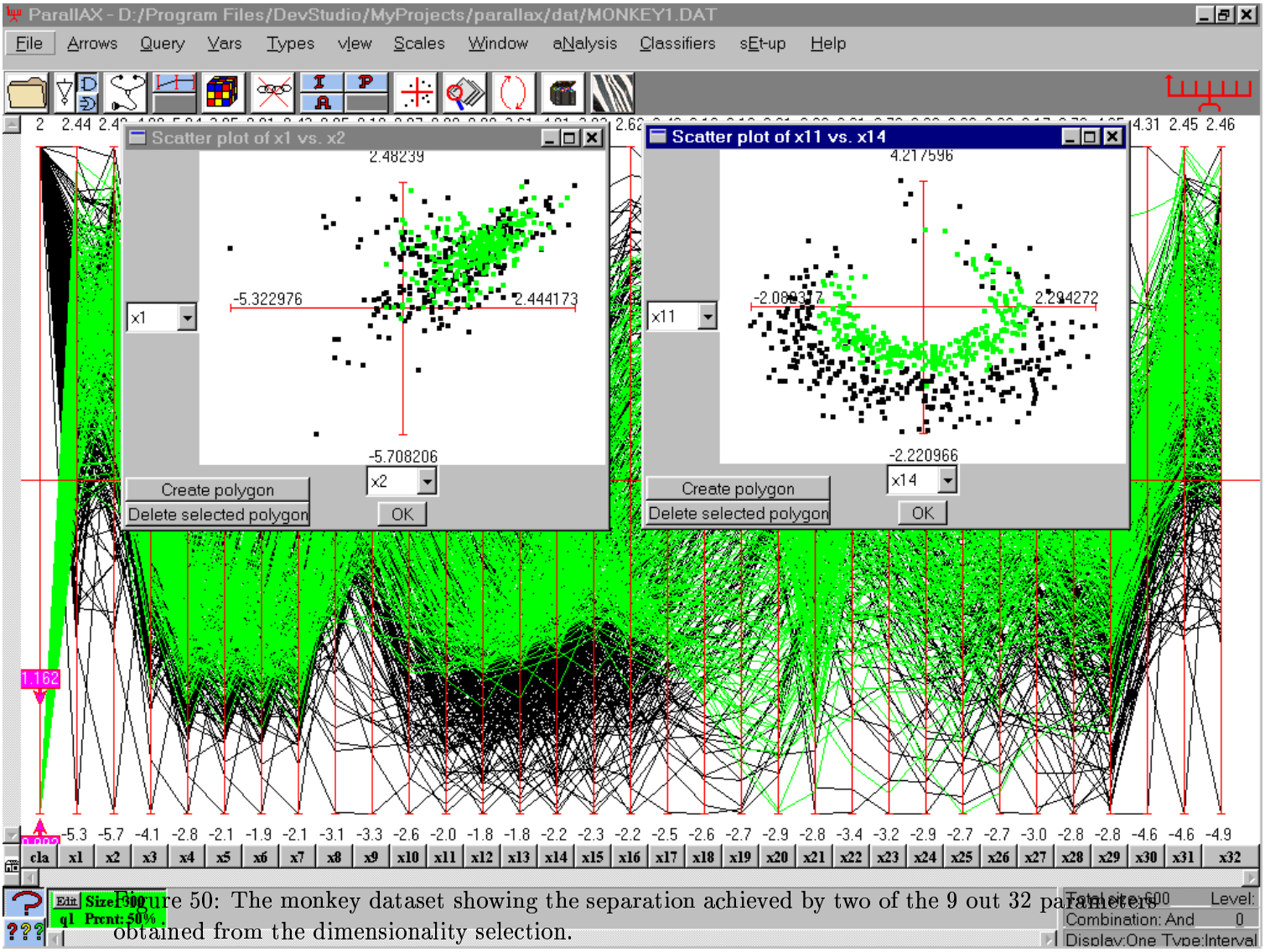

Figure 50: The monkey dataset showing the separation achieved by two of the 9 out 32 parameters obtained from the dimensionality selection.

Representing Curves

Generalized Conics – CONVEX SETS (The six cases) and other dualities

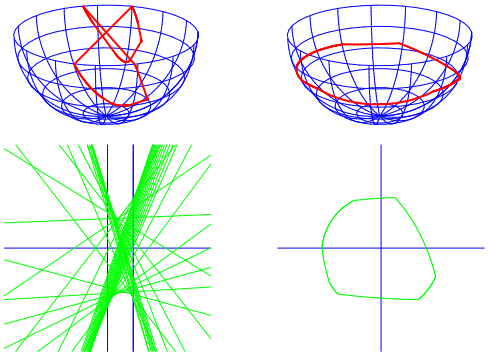

Figure 51: Case 1 – Estar (ellipse-like) to hstar (hyperbola-like), Dual of case 4. On Model of Projective Plane (abbreviated as MMP) the estar has no ideal points (i.e. does not contain any diameters of the cap) while the hstar has two corresponding to its asymptotes.

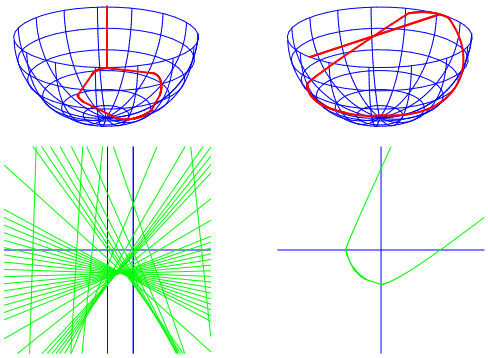

Figure 52: Case 2 – Pstar (parabola-like) to pstar, self-dual
On MMP a pstar is an estar with one diameter. This corresponds to the direction (ideal point) the pstar opens towards ℓ_∞.

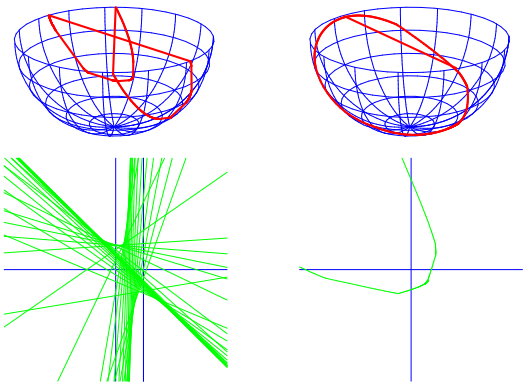

Figure 53: Case 3 – Pstar to hstar, Dual of Case 5 (not shown). A pstar having a supporting line with slope 1 is mapped into an hstar with one vertical asymptote (and where every vertical intersects only one chain in 1 point) and conversely.

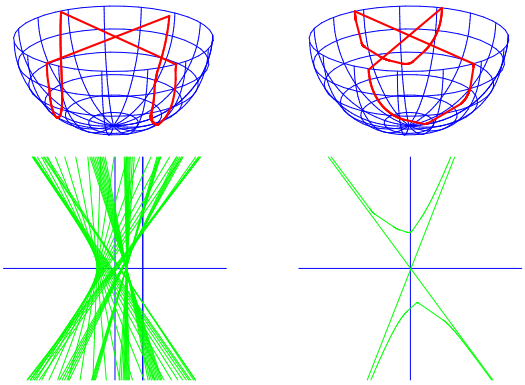

Figure 55: Case 6 – Hstar to hstar, Self-dual

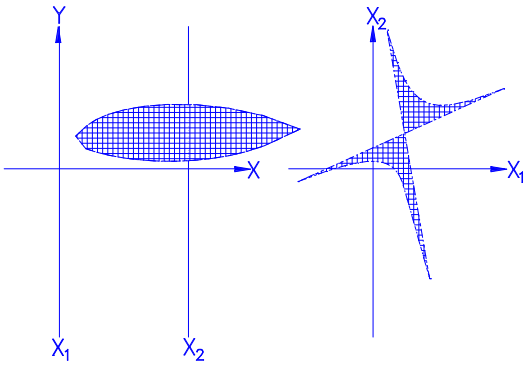

Figure 54: Case 4 – Hstar to estar, Dual of 1

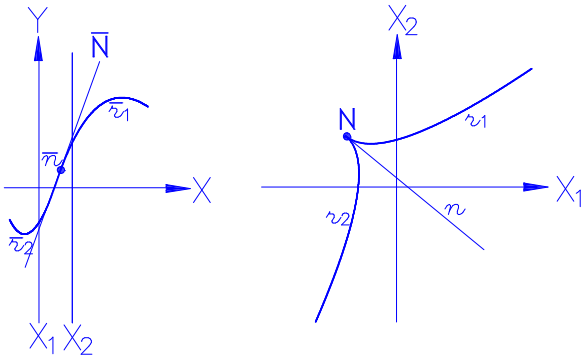

Figure 56: Cusps are transformed into inflection points and conversely

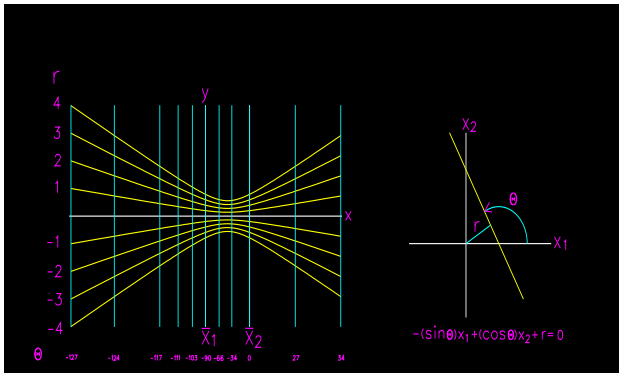

Figure 57: A family of line transformations
Fixing r and varying Θ defines a family of lines tangent to the circle whose parallel coordinate representation is a hyperbola while fixing Θ and varying r produces vertical lines.

Line Neighborhoods

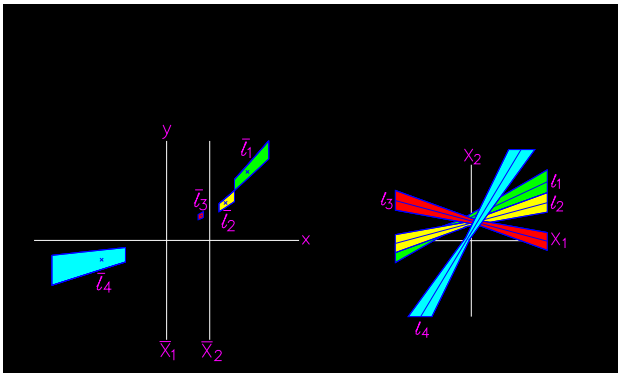

Figure 59: Several line neighborhoods
Here the transformed neighborhoods are distinct.

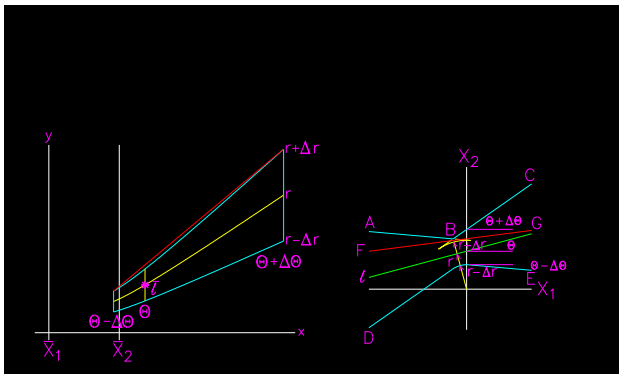

Figure 58: Line neighborhood in orthogonal and parallel coordinates
An unbounded region (on the right) is replaced by a bounded one.

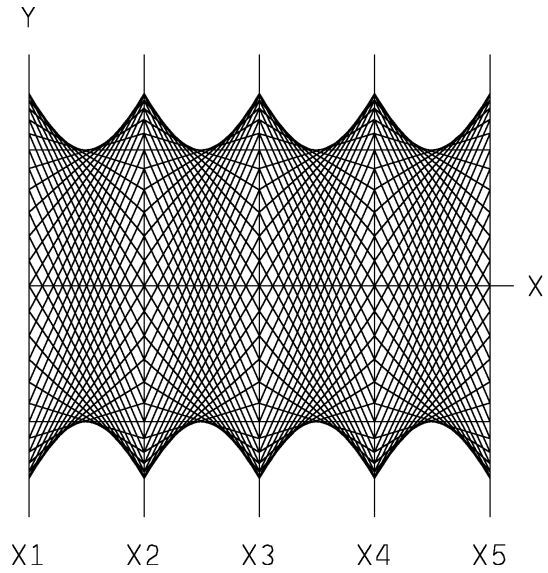

Figure 60: A sphere in R^5

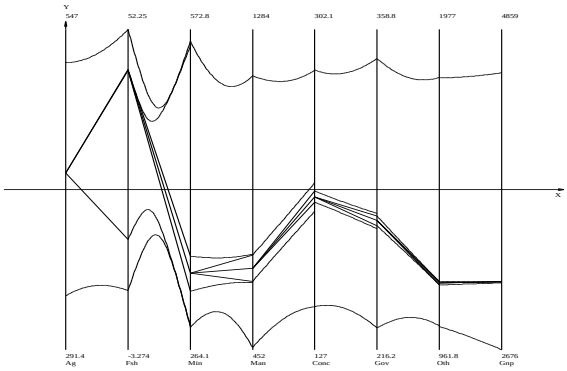

Figure 62: Model of a country's economy

Hypersurfaces

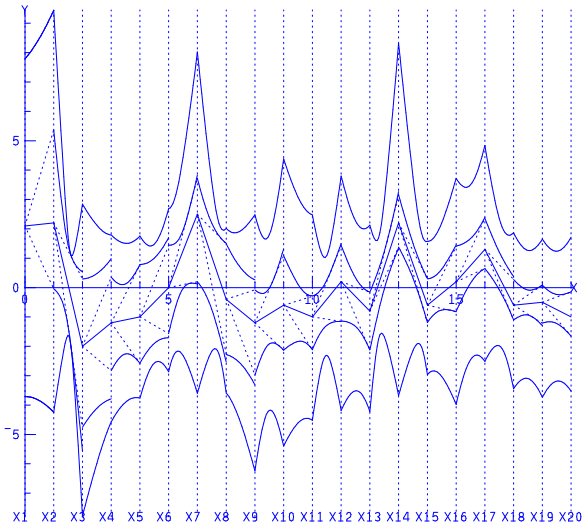

Figure 61: Finding a Feasible Point for a Process Represented by a Hypersurface

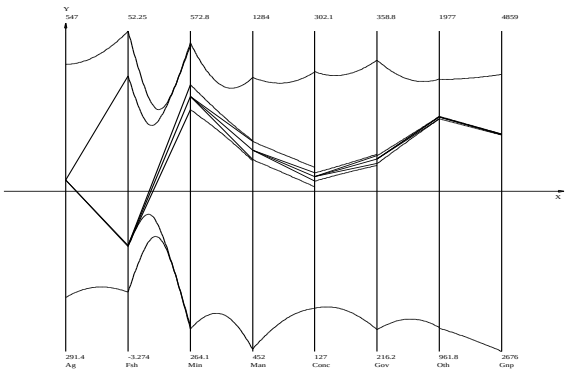

Figure 63: Competition for labor between the Fishing & Mining sectors – compare with previous figure

References

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

[2] T. Chomut. *Exploratory Data Analysis in Parallel Coordinates.* M.Sc. Thesis, UCLA Comp. Sc. Dept., 1987.

[3] R. Finsterwalder. *A Parallel Coordinate Editor as a Visual Decision Aid in Multi-Objective Concurrent Control Engineering Environment 119-122.* IFAC CAD Contr. Sys., Swansea, UK, 1991.

[4] A. Inselberg. *Multidimensional Detective, in Proc. of IEEE Information Visualization '97, 100-107.* IEEE Comp. Soc., Los Alamitos, CA, 1997.

[5] A. Inselberg and B. Dimsdale. Multidimensional lines i: Representation. *SIAM J. of Applied Math.*, 54-2:559–577, 1994.

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

[7] D. A. Keim and H. P. Kriegel. Visualization techniques for mining large databases: A comparison. *Trans. Knowl. and Data Engr.*, 8-6:923–938, 1996.

[8] A. R. Martin and M. O. Ward. *High dimensional brushing for interactive exploration of multivariate data, Proc. IEEE Conf. on Visualization, Atlanta, GA, 271-278.* IEEE Comp. Soc., Los Alamitos, CA, 1995.

[9] M. Schall. Diamond and ice : Visual exploratory data analysis tools. *Perspective, J. of OAC at UCLA*, 18(2):15–24, 1994.

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

[11] D. F. Swayne, D. Cook, and A. Buja. *XGobi : Interactive Dynamic Graphics in the X Window System.* JCGS, **7-1**, 113-130, 1998.

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

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

[14] L Tweedie and R. Spence. The prosection matrix : A tool to support the interactive exploration of statistical analysis. *Comput. Statit.*, 13-1:65–76, 1998.

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