In: Knowledge Processing and Data Analysis, Wolff, K.E.; Palchunov, D.E.; Zagoruiko, N.G.; Andelfinger, U. (Eds.), Lecture Notes in Computer Science, 2011, Volume 6581/2011, pp. 297-313, DOI: 10.1007/978-3-642-22140-8 20

Visual Data Mining and Discovery in Multivariate Data using Monotone n-D Structure

Boris Kovalerchuk[1], Alexander Balinsky[2]

[1]Department of Computer Science, Central Washington University, 400 E. University Way, Ellensburg, WA 98926-7520, USA

[2] Cardiff School of Mathematics, Cardiff University, Senghennydd Road, Cardiff, CF24 4AG, UK

Abstract. Visual data mining (VDM) is an emerging research area of Data Mining and Visual Analytics to gain a deep visual understanding of data. A border between patterns can be recognizable visually, but its analytical form can be quite complex and difficult to discover. VDM methods have shown benefits in many areas, but these methods often fail in visualizing highly overlapped multidimensional data and data with little variability. We address this problem by combining visual techniques with the theory of monotone Boolean functions and data monotonization. The major novelty is in visual presentation of structural relations between n-dimensional objects instead of traditional attempts to visualize each attribute value of n-dimensional objects. The method relies on n-D monotone structural relations between vectors. Experiments with real data show advantages of this approach to uncover a visual border between malignant and benign classes.

Keywords: Data Mining, Visualization, Structural relations, Monotonicity, Multidimensional data.

1 Introduction

Visual data mining is an emerging research area of Data Mining and Visual Analytics [Thomas et al., 2005] to gain a deep visual understanding of data. The purpose of this paper is to develop a technique for visualizing and discovering patterns and relations from *multidimensional binary data* using the technique of monotone Boolean functions. *Visualizing the border* between patterns is one of especially important aspects of visual data mining. In many situations, a user can easily catch a border visually, but its analytical form can be quite complex and difficult to discover. The simple borders of patterns that are visually far away from each other match our intuitive concept of the pattern and increase confidence to the robustness of data mining result as discovered patterns.

Deep visual understanding is a goal of visual data mining [Beilken, Spenke, 1999]. Known visualization techniques such as parallel coordinates have been successfully used in visual data mining, but for highly overlapped multidimensional data, the

resulting visualization often is unsatisfactory with a high-level occlusion. This problem is especially challenging in medical applications tracked with binary symptoms and complex dynamic optimization problems.

VDM is an especially challenging task when data richness should be preserved without excessive aggregation [Keim e al., 2002]. Another challenge is a lack of natural 3-D space and time dimensions in many tasks [Groth, 1998] that requires the visualization of abstract features. Quite often visual representations suffer from subjectivity, poor scalability, inabilities to perceive more than 6-8 dimensions, and the slow interactive examination of complex data [Last, Kandel, 1999; Keim et al., 2002].

Glyph or *iconic visualization* is an attempt to encode multidimensional data using parameters such as the shape, color, transparency, and orientation of 2-D or 3-D objects (2-D icons, cubes, or more complex *"Lego-type"* 3-D objects) [Ebert et al, 1996; Post, van Walsum et al., 1995; Ribarsky et al., 1994]. Glyphs can visualize *nine attributes* (three positions *x*, *y*, and *z*; three size dimensions; color; opacity; and shape). Texture can add more dimensions. Shapes of the glyphs are studied in [Shaw, et al., 1999], where it was concluded that with large super-ellipses, about 22 separate shapes can be distinguished on the average. An overview of multivariate glyphs is presented in [Ward, 2002]. Glyph methods commonly use data dimensions as positional attributes (*x,y,z*) to place glyphs, which often result in occlusion.

Other glyph methods place glyphs using *implicit or explicit structure* within the data set. In this paper, we show that the placement based on the use of the *data structure* is a promising approach to visualize a border between patterns for multidimensional data. We call this the **GPDS** approach (*Glyph Placement on a Data Structure*). In this approach, some attributes are *implicitly* encoded in the data structure while others are *explicitly* encoded in the glyph/icon. Thus, if the structure carries ten attributes and a glyph/icon carries nine attributes, nineteen attributes are encoded. We use simple 2-D icons as *bars* of different colors. Adding texture, motion and other icon characteristics can increase dimensions of data visualized. Collapsing of the bar to a single pixel is another way to make GPDS approach more scalable**.**

Alternative techniques such as g*eneralized spiral and pixel bar* chart are developed in [Keim et al., 2002]. These techniques work with large data sets without overlapping, but only with a few attributes (≤6). Other visualization methods, known as Scatter, Splat, Map, Tree, and Evidence Visualizer, that are implemented in MineSet (Silicon Graphics), permit up to eight dimensions to be shown on the same plot by using color, size, and animation of different objects [Last, Kandel, 1999].

The *parallel coordinate technique* [Inselberg, Dimsdale, 1990] can work with ten or more attributes, but suffers from record overlap and thus is limited to tasks with well-distinguished cluster records. In parallel coordinates, each vertical axis corresponds to a data attribute (x_i) and a line connecting points on each parallel coordinate corresponds to a record. Parallel coordinates visualize explicitly *every* attribute x_i of an n-dimensional vector ($x_1,x_2,\ldots,x_n$) in 2-D and place the vector using *all attributes* x_i, but each attribute is placed on its own parallel coordinate *independently* of placing other attributes of this vector and other vectors. This is one of the major reasons of occlusion and overlap of visualized data. The GPDS approach constructs a data structure and can place objects using *attribute relations*.

A typical example of current research in high-dimensional Visual Analytics is the work of Wilkinson et al. [2006], which uses the 2-D *distributions of pairwise*

orthogonal projections of the multidimensional Euclidean space. 2-D projections may not discover real *n*-D patterns and glyphs often lead to occlusion. We attempt to show only relevant structural relations between n-D vectors in 2-D or 3-D, not actual n-D vectors. A more extensive representation of related work is presented in [Peng et al., 2004; Mackinlay, 1997; de Oliveira, Levkowitz, 2003; Yang et al., 2007].

The proposed method to represent n-D data in 2-D or 3-D is based on the following principles of visual representations, learning, and discovery:

Principle 1: *Represent complexity, not abstract multiplicity*. The visual system was developed evolutionary to deal efficiently with dynamics of complex concrete objects in 3-D world. This ability does not imply the same efficiency to deal with multiple abstract multidimensional objects.

Principle 2: *Represent concrete complexity in low dimensions*. The human visual system has unique abilities to understand concrete complex information received via visual channels in low dimensions (2-D and 3-D). Therefore, it is desirable to transform the n-D data to concrete 2-D or 3-D entities.

Principle 3: *Represent individual attributes of n-D data in 2-D/3-D only if necessary*. This will help to minimize occlusion and loss of information. The parallel coordinates method that represents all attributes of n-D data suffers from occlusion.

Principle 4: *Represent structural relations between n-D data in 2-D or 3-D that have a clear meaning for the analyst*. Such an attempt can avoid occlusion and loss of information.

Principle 5: A human can *capture structural relations such as order (hierarchy, generalization) and monotonicity in n-D*. Therefore, these relations are first candidates to be a base for n-D data representation in 2-D and 3-D.

Principle 6: *If n-D data have no recorded natural hierarchy and n-D monotonicity, modify data representation, learn, discover, and build these structures* in a new representation with clear meaning for the analyst.

This paper is organized as follows. Section 2 contains definitions and mathematical statements. Section 3 describes the proposed visual representation of the n-D Boolean space in 2-D. Section 4 describes a learning process. The paper concludes with computational experiments and an outline of the further research and summary.

2. Chains and similarity distances between chains

We denote the set of all *n*-D binary vectors $E^n := \{0,1\}^n$ (*n*-D binary cube), which is a partially ordered set that form a lattice with the greatest element (1,1,…,1) and the smallest element (0,0,…,0) for the relation $\leq$ on vectors $\mathbf{a}=(a_{1,\ldots,}a_n) \in E^n$ and $\mathbf{b}=(b_{1,\ldots,}b_n) \in E^n$, I={1,2,…,n},

$$\mathbf{a} \leq \mathbf{b} \Leftrightarrow \forall\, i \in I\ a_i \leq b_i$$

Only some vectors in E^n are ordered in this way.

Consider a pair (M, ≤), where M is a set of elements and "≤" is a partial order relation on M.

Definition. A subset H ⊆ M is called a *chain* if any two elements **g**, **h** of H are comparable, i.e., satisfy linearity property, **g** ≤ **h** or **h** ≤ **g** [Birkhoff,1995; Grätzer, 1971].

In our case M=E^n with the partial order defined above.

Definition. Let **a**, **b**, **c**, and **d** be *n*-D Boolean vectors then a triple <**a**,{**b**,**c**},**d**> is called a *square* (see Fig. 1a) if **a**<**b**<**d**, **a**<**c**<**d, b**≠**c,** and |**a**|+1=|**b**|=|**c**|=|**d**|-1, where | . | is the *Hamming norm*, $|\mathbf{x}| = \sum_{i=1}^{n} x_i$, that is the number of 1s in the vector.

Vectors **b** and **c** are incomparable relative to ≥ which is denoted as **b**||**c** [Grätzer, 1971]. The concept of square for E^n was introduced in [Hansel, 1966] in slightly different terms. Below we introduce the concepts of adjacent and S-adjacent squares.

Definition. Squares S_1 = <$\mathbf{a}_1$,{$\mathbf{b}_1$,$\mathbf{c}_1$},$\mathbf{d}_1$> and S_2 = <$\mathbf{a}_2$,{$\mathbf{b}_2$,$\mathbf{c}_2$},$\mathbf{d}_2$> are called *adjacent* iff |{b_1,c_1} ∩ {b_2,c_2}| = 1. See Fig. 1b where $\mathbf{c}_1$ = $\mathbf{b}_2$.

Definition. Chains H_1 and H_2 are *S-adjacent where S:*= <a,{b,c},d> is a square, if one of them contains three nodes **a**, **b** and **d** of the square S and the other one contains the forth node **c** of S.

H_1 and H_2 are called *adjacent* if there is a square S such that they are S-adjacent.

In table 1 chains 0H1 and 0H2 are adjacent, because these chains are S_1-adjacent, that is they contain square S_1 = <$\mathbf{a}_1$,{$\mathbf{b}_1$,$\mathbf{c}_1$},$\mathbf{d}_1$> = < (000), {(001), (010)},(011)>. Similarly, chains 0H1 and 1H1 are also adjacent, because they contain another square S_2 = < $\mathbf{a}_2$,{$\mathbf{b}_2$,$\mathbf{c}_2$},$\mathbf{d}_2$> = <(001), {(011), (1010}, (111) >.

In the same way in Table 2, chains 0H1 and 1H1are adjacent with S = <(0011),{(0111), (1011)},(1111)> and in table 3 chains 0H1 and 1H1are adjacent with S = <(00111),{(01111), (10111)},(11111)>.

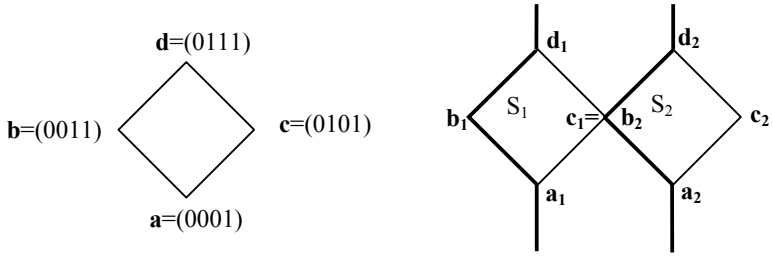


Fig. 1. Examples of a Boolean square (a) and adjacent squares (b)

In Fig 1 illustrates a square and a pair of adjacent squares. Below we define Hansel chains [Hansel, 1966, Kovalerchuk et al., 1996]

Definition. A set of chains {Q} in E^n is called a *partitioning set of chains (or P-set of chains)* if {Q} *partitions* E^n, i.e., chains in {Q} do not overlap and cover E^n completely.

The partitioning property is important for visual data mining of multidimensional data because it helps to avoid occlusion of n-D Boolean data, visualized in 2-D or 3-D as we show below in our design of the visual representation in section 3.

Below we define a set of Hansel chains {C} [Hansel, 1966] in E^n, which is a special type of P-set of chains with multiple useful properties. These properties of Hansel chains have been used to identify the minimal number of tests/questions needed to restore any monotone Boolean function [Hansel, 1966].

Hansel chains for E^n are defined recursively from chains in E^{n-1}. Therefore, we start from defining Hansel chains in E^1.

Definition: The single Hansel chain in $E^1 = \{0,1\}$ is defined as H := {0,1}, hence the set of Hansel chains in E^1 is the partition with a single class {H}.

These vectors are ordered by their Hamming norms, |1| > |0|.

Definition. The set of Hansel chains in E^2 consist of the two chains 1H:= {10} and 0H:={00, 01, 11}. These chains partition E^2. We will extend this notation later.

Note that the set of chains U = {{01}, {00, 10, 11}} is also a partitioning set of chains, but U is not the set of Hansel chains in E^2.

Definition. The set of Hansel chains {C} in E^3 consists of chains presented in table 1. These chains also do not overlap and cover E^3 completely.

Table 1. The Hansel chains in E^3

Chain	*Norm 0*	*Norm 1*	*Norm 2*	*Norm 3*
00H	000	001	011	111
10H		100	101	
01H		010	110	

The *general recursive process* of defining *a set of Hansel chains* in E^{n+1} from a set of Hansel chains in E^n is as follows [Hansel, 1966; Kovalerchuk et al., 1996].

The main steps of the recursive process are:

(1) *producing* two intermediate chains in E^{n+1} from each Hansel chain in E^n, and
(2) *modifying these intermediate chains* to produce Hansel chains in E^{n+1}.

To get the first intermediate chain, each element of a chain from E^n is expanded by adding a leading zero, and to get the second intermediate chain each element of the same chain from E^n is expanded by adding a leading one.

In particular, having a single chain H = {0, 1} in E^1 we get intermediate chains in E^2, 0H'={00, 01} and 1H'={10, 11} with two Boolean vectors each. Then the growth-cut process is applied to produce final Hansel chains in E^2: 0H= {00, 01, 11} and 1H = {10} with three elements and one element, respectively. The growth-cut process cuts the largest element of 1H' that is (11) and grows 0H' by adding (11) as a new top element to 0H' to get 0H= {00, 01, 11}.

For building Hansel chains in E^3 from 0H = {00,01,11} in E^2 we get first 00H'= {000,001,011}. Next, we produce a chain 10H' with leading 1 from H, 10H' = {100,101,111}. The *growth-cut step* cuts the greatest element from 10H' and grows 00H' by adding the greatest element to 00H. This produces chains 00H={000,001,011,111} and 10H={100,101} with 4 and 2 elements, respectively.

We have shown how chain 0H={00,01,11} produces chains in E^3. Similarly, chain 1H={10} that consists of a single node produces first two preliminary chains in E^3: 01H'={010} and 11H' = {110} that also consist of singe nodes. Then the growth-cut process produces the final chains 01H={010, 110} and 11H={} which is empty and deleted.

The application of these steps in E^3 produces sets of Hansel chains in E^4 (see Table 2), and from these we get a set of Hansel chains in E^5 (see Table 3). All Hansel chains of the recursively constructed sets of Hansel chains for higher n are produced from this set {H} of a single Hansel chain H in E^1.

Table 2. The set of Hansel chains in E^4

Chain	*Norm 0*	*Norm 1*	*Norm 2*	*Norm 3*	*Norm 4*
000H	0000	0001	0011	0111	1111
001H		1000	1001	1011	
100H		0100	0101	1101	
101H			1100		
010H		0010	0110	1110	
011H			1010		

Table 3. The set of Hansel chains in E^5

Chain	*Norm 0*	*Norm 1*	*Norm 2*	*Norm 3*	*Norm 4*	*Norm 5*
0000H	00000	00001	00011	00111	01111	11111
0001H		10000	10001	10011	10111	
0010H		01000	01001	01011	11011	
0011H			11000	11001		
1000H		00100	00101	01101	11101	
1001H			10100	10101		
1010H			01100	11100		
0100H		00010	00110	01110	11110	
0101H			10010	10110		
0110H			01010	11010		

Empty chains 1011H and 0111H are omitted.

Statement 1.

In any Hansel chain C in E^n constructed above recursively with nodes $\mathbf{h}_i$, C = { $\mathbf{h}_i$ | i=1:k}, the labelling of the nodes can be chosen in such a way that the *one-step property* $|\mathbf{h}_{i+1}| = |\mathbf{h}_i| + 1$ is satisfied.

Proof. We prove by induction that for each integer $n \geq 1$ the one-step-property of each Hansel chain in the set of Hansel chains in E^n holds. It is obvious that this holds for n = 1. Let $n \geq 1$. We now assume the induction hypothesis that the one-step-property of each Hansel chain in the set of Hansel chains in E^n holds and prove this statement for n+1. Let C := $\{\mathbf{h}_1,\ldots, \mathbf{h}_k\}$, $k \geq 1$, be a Hansel chain in E^{n+1}. We use from the recursive construction that C is either of the form (0) C=$\{0\mathbf{f}_1,\ldots, 0\mathbf{f}_{k-1}, 1\mathbf{f}_{k-1}\}$ where $\{\mathbf{f}_1,\ldots, \mathbf{f}_{k-1}\}$ is a Hansel chain in E^n or (1) C = $\{1\mathbf{g}_1,\ldots, 1\mathbf{g}_{k-1}, 1\mathbf{g}_k\}$ where $\{\mathbf{g}_1,\ldots, \mathbf{g}_{k+1}\}$ is a Hansel chain in E^n. In either case we get from the one-step-property of the Hansel chains in E^n that C has the one-step-property.

Statement 2. If chains H_1 and H_2 are S_1-adjacent and S_2-adjacent in such a way that square $S_1 = <\mathbf{a}_1, \{\mathbf{b}_1, \mathbf{c}_1\}, \mathbf{d}_1>$ contains elements $\mathbf{a}_1$, $\mathbf{b}_1$, $\mathbf{d}_1$ in chain H_1 and element c_1 in chain H_2 and square $S_2 = <\mathbf{a}_2, \{\mathbf{b}_2, \mathbf{c}_2\}, \mathbf{d}_2>$ contains elements $\mathbf{a}_2$, $\mathbf{b}_2$, $\mathbf{d}_2$ in H_2 and $\mathbf{c}_1 = \mathbf{b}_2$ (see Fig. 1b) then $|\mathbf{a}_1\text{-}\mathbf{a}_2| = |\mathbf{b}_1\text{-}\mathbf{c}_1| = |\mathbf{b}_1\text{-}\mathbf{b}_2| = |\mathbf{d}_1\text{-}\mathbf{d}_2| = 2$.

Proof. According to the definition of square S_i, $|\mathbf{a}_i| = |\mathbf{c}_i| - 1 = |\mathbf{b}_i| - 1$, i=1,2, therefore $|\mathbf{a}_i\text{-}\mathbf{c}_i| = |\mathbf{a}_i\text{-}\mathbf{b}_i| = 1$. Also $\mathbf{c}_1 = \mathbf{b}_2$, thus $|\mathbf{c}_1| = |\mathbf{b}_2|$ and $|\mathbf{a}_1| = |\mathbf{a}_2|$. Thus, $|\mathbf{a}_1| = |\mathbf{a}_2| = |\mathbf{c}_1| - 1$. Also $\mathbf{a}_1 \neq \mathbf{a}_2$ because $\mathbf{a}_1$ and $\mathbf{a}_2$ belong to different chains. Vector $\mathbf{a}_1$ alters $\mathbf{c}_1$ in one attribute and vector $\mathbf{a}_2$ alters $\mathbf{c}_1$ it in another attribute. Say, $c_{1i}=1$ and $c_{1j}=1$, then $a_{1i}=0$, and $a_{2j}=0$. Thus, the distance between $\mathbf{a}_1$ and $\mathbf{a}_2$ is 2, $|\mathbf{a}_1\text{-}\mathbf{a}_2| = 2$. Similarly, $\mathbf{d}_1$ and $\mathbf{d}_2$ alter $\mathbf{c}_1$ in two positions, thus, $|\mathbf{d}_1\text{-}\mathbf{d}_2| = 2$.

Statement 3 For all $\mathbf{x}$, $\mathbf{y}$ such that $\mathbf{x} \neq \mathbf{y}$, and $|\mathbf{x}| = |\mathbf{y}|$ the distance $|\mathbf{x}\text{-}\mathbf{y}| \geq 2$ and $\mathbf{x}$ and $\mathbf{y}$ exist such that $|\mathbf{x}\text{-}\mathbf{y}| = 2$

This means that if two Boolean vectors differ but have equal norms then the minimal Hamming distance between these vectors is equal to 2.

Proof. If this distance would be equal to 1, $|\mathbf{x}\text{-}\mathbf{y}| = 1$, then it will be only one *i* such that $x_i \neq y_i$, say $x_i=0$ and $y_i=1$. This will imply that $|\mathbf{x}| < |\mathbf{y}|$ which contradicts the condition that $|\mathbf{x}| = |\mathbf{y}|$.

Statements 2 and 3 help us to clarify and define the distance between Hansel chains using the distance between their elements. Say, we can measure the distance between two Hansel chains as a Hamming distance between their closest elements. Statement 2 tells us that there are elements of adjacent chains with the distance equal to 2 and statement 3 tells us that the distance between elements of different chains cannot be less than 2. Thus, the adjacent chains H_1 and H_2 are closest Hansel chains in this measure of the distance between Hansel chains.

Theorem (Hamming distance for adjacent Hansel chains).

If H_1 and H_2 are *adjacent Hansel chains* (of a set of Hansel chains in E^n (n>1), constructed recursively as described above), then for every element $\mathbf{h}_1 \in H_1$ and element $\mathbf{h}_2 \in H_2$ with equal norms, $|\mathbf{h}_1| = |\mathbf{h}_2|$ the Hamming distance is equal to 2, $|\mathbf{h}_1\text{-}\mathbf{h}_2| = 2$, that is

$$\forall\, \mathbf{h}_1 \in H_1\ \forall\, \mathbf{h}_2 \in H_2\ \ (|\mathbf{h}_1| = |\mathbf{h}_2| \Rightarrow |\mathbf{h}_1\text{-}\mathbf{h}_2| = 2). \qquad (1)$$

In other words, the distance between vectors of the same norm in adjacent Hansel chains is the constant 2 and this is the smallest distance for unequal Hansel chains H_1 and H_2.

Proof.

The plan of the proof is first to show that for each Hansel chain C in E^n the two generated preliminary chains 0C' and 1C' in E^n (which have the same cardinality, namely $|0C'| = |1C'| = |C| := k$) satisfy a specific properties described below. Then these properties are used for computing the Hamming distance between Hansel chains.

Consider preliminary chains $0C' = \{0\mathbf{h}_i\}$ and $1C' = \{1\mathbf{h}_i\}$ in E^{n+1} produced from chain $C = \{\mathbf{h}_i : i=1:k\}$ in E^n by adding leading 0 or 1 to each element of C to get 0C' and 1C', respectively. Chains 0C' and 1C' have the same number of elements k as chain C.

Elements in C are numbered in such a way that i=1 for the node with the smallest norm, $|\mathbf{h}_1|=\min\{|\mathbf{h}_i| \mid i=1:k\}$ and $|\mathbf{h}_{i+1}| = |\mathbf{h}_i| +1$ (See Statement 1).

The properties of these chains that we use below are: $|1\mathbf{h}_i|=|0\mathbf{h}_i|+1$ and $|1\mathbf{h}_i|=|0\mathbf{h}_{i+1}|$. These properties are true (see Statement 1). For i=1, $|1\mathbf{h}_1|=|0\mathbf{h}_1|+1$, thus, norms of all nodes in 1C' are greater than the norm of the first node of 0C', $0\mathbf{h}_1$. Similarly, the largest node of 1C', $1\mathbf{h}_k$ has no node with equal norm in 0C'. This consideration tells us that $|1\mathbf{h}_i|=|0\mathbf{h}_{i+1}|$, i=2,…,k-2.

The next step is computing the Hamming distance between $1\mathbf{h}_i$ and $0\mathbf{h}_{i+1}$ using decomposition where we use notation $\mathbf{h}_i=(h_{i1},\ldots,h_{in})$:

$$\left|1\mathbf{h}_i - 0\mathbf{h}_{i+1}\right| = \sum_{j=1}^{n+1}\left|1h_{ij} - 0h_{i+1\,j}\right| = \left|1h_{i,1} - 0h_{i+1,1}\right| + \sum_{j=2}^{n+1}\left|1h_{ij} - 0h_{i+1\,j}\right| =$$

$$\left|1-0\right| + \sum_{j=2}^{n+1}\left|h_{ij} - h_{i+1\,j}\right| = 1 + \left|\mathbf{h}_i - \mathbf{h}_{i+1}\right| = 1+1 = 2$$

Here we separated the leading digit (j=1) from the sum. The remaining parts are $\mathbf{h}_i$ and $\mathbf{h}_{i+1}$. We used the property of Hansel chains that $|\mathbf{h}_{i+1}|= |\mathbf{h}_i|+1$ (see Statement 1) and $|\mathbf{h}_i - \mathbf{h}_{i+1}|=1$.

This proof is for preliminary chains 0C' and 1C'. The modification of them to Hansel 0C and 1C chains does not change the norm of any element of the chains; it only relocates the largest node $1\mathbf{h}_k$ to 0C. The theorem is not applicable for this node, because the condition of the theorem in (1) was $|\mathbf{h}_1| = |\mathbf{h}_2|$ which is not true for $1\mathbf{h}_k$, as its norm is greater than the norm of any other node in 0C and 1C. This theorem is illustrated with actual distances computed for E^3-E^5 in Tables 4-6.

Statement 4. The distance between *n*-D Boolean vectors $\mathbf{a} =\{a_j\}$ and $\mathbf{b}=\{b_j\}$ with equal norms, $|\mathbf{a}|=|\mathbf{b}|$, is an even number.

Proof. If r is the number of bits j where $a_j = 1$ and, $b_j = 0$ then there should be r other bits where $a_j = 0$ and $b_j = 1$ to have $|\mathbf{a}|=|\mathbf{b}|$. In this case the total number of bits where **a** and **b** differ is 2r, which is their Hamming distance, $|\mathbf{a}\text{-}\mathbf{b}|=2r$.

Statement 5. If H_1 and H_2 are Hansel chains and H_2 and H_3 are also adjacent Hansel chains, then the Hamming distance $|\mathbf{a}\text{-}\mathbf{b}|$ between any **a** from H_1 and any **b** from H_3 such that $|\mathbf{a}|=|\mathbf{b}|$ is equal to 2, or 4.

Proof. It follows directly from the Theorem 1, Statement 4 and the fact the Hamming distance as every distance satisfies a triangle inequality: $|\mathbf{a}_1\text{-}\mathbf{a}_3| \le |\mathbf{a}_1\text{-}\mathbf{a}_2|+ |\mathbf{a}_2\text{-}\mathbf{a}_3|=2+2=4$. This distance cannot be the odd number 3.

Hypothesis (*constant distance hypothesis*). For all Hansel chains H_1, H_2 in E^n there exists a constant c, such that $|\mathbf{h}_1 - \mathbf{h}_2| = c$ for all $\mathbf{h}_1$, $\mathbf{h}_2$ from H_1 and H_2 with equal norms, $|\mathbf{h}_1|=|\mathbf{h}_2|$.

We explore for which *n* this Hypothesis is true. It is true for *n*=1, 2, 3 and 4. See Tables 4 and 5 for n=3 and n=4. Table 6 shows that is not true for *n*=5, that provides counterexamples for n=5. In Table 4 notation 22 means two distances. For instance for chains 0H1 and 1H1, the first 2 is the distance between lower nodes (001) and (100), and the second 2 is the distance between upper nodes: (011) and (101) from chains 0H1 and 1H1 respectively. In Table 6, notation 2242 indicates 4 distances for four nodes of chains 10H1 and 00H2. In these chains the distances between vectors (10011) and (01101) is equal to 4 and the distances between vectors (10001) and (00101) is equal to 2. This provides a counterexample.

Table 4. Distances between Hansel chain nodes with equal norms in E^3

	Chain 0H1	Chain 1H1	Chain 0H2
Chain 0H1	0	22*	22
Chain 1H1		0	22
Chain 0H2			0

Definition. Let $D_{HC}(H_1, H_2) = (\sum_{|\mathbf{h}_1|=|\mathbf{h}_2|} |\mathbf{h}_1 - \mathbf{h}_2|) / m$,

where $\mathbf{h}_{1i}$ and $\mathbf{h}_{2i}$ are nodes of chains H_1 and H_2, respectively, with the same Hamming norm $|\mathbf{h}_{1i}|=|\mathbf{h}_{2i}|$, m is the number of such pairs of nodes and $|\mathbf{h}_{1i}-\mathbf{h}_{2i}|$ is Hamming distance between Boolean vectors. D_{HC} is called an *averaged Hamming chain similarity measure*. Table 7 presents this measure for E^5.

Table 5. Distances between all Hansel chain nodes with the same norm in E^4

Chain	0H1	1H1	0H2	1H2	0H3	1H3
0H1	0	222	222	4	222	2
1H1		0	2	2	4	2
0H2			0	2	2	2
1H2				0	2	2
0H3					0	2
1H3						0

Table 6. Distances between all Hansel chain nodes with the same norm in E^5

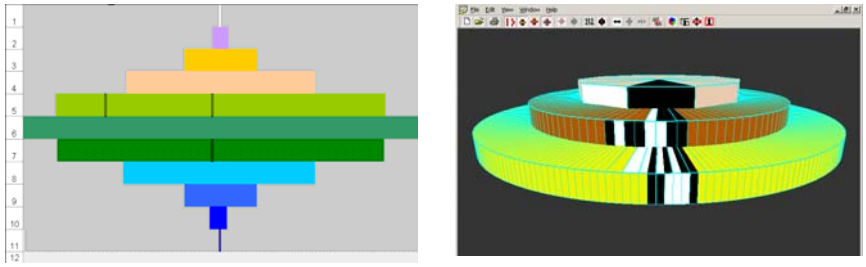
Chain	00H1	10H1	01H1	11H1	00H2	10H2	01H2	00H3	10H3	01H3
00H1	0	2222	2222	44	2222	42	44	2222	22	24
10H1		0	2222	22	2242	42	44	2422	22	24
01H1			0	22	2222	42	22	2422	24	22
11H1				0	44	22	22	44	24	22
00H2					0	22	22	22	44	44
10H2						0	22	24	22	22
01H2							0	22	22	22
00H3								0	22	22
10H3									0	22
01H3										0

Table 7. Distances D between Hansel chains in E^5

	00H1	10H1	01H1	11H1	00H2	10H2	01H2	00H3	10H3	01H3
00H1	0	2	2	4	2	3	4	2	2	3
10H1		0	2	2	2.5	3	4	2.5	2	3
01H1			0	2	2	3	2	2.5	3	2
11H1				0	4	3	2	4	3	2
00H2					0	2	2	2	4	4
10H2						0	2	3	2	2
01H2							0	2	2	2
00H3								0	2	2
10H3									0	2
01H3										0

3. Representing and drawing of n-D Boolean space in 2-D

Below we describe a structure to allocate Boolean vectors in 2-D. Vectors are ordered vertically by their Boolean norm (sum of "1"s) with the largest vectors are rendered on the top starting from (11111). This is a way to *use relations between attributes* for placing Boolean vectors. Each vector will be first placed in accordance with its norm (level) and then drawn as a colored bar: white for the 0 class, black for the 1 class. Fig. 3 depicts the hierarchy of levels of Boolean vectors in E^{10}, which is 11 levels from 0 to 10, where level 0 contains a single vector (0000000000) and level 10 contains a single vector (1111111111). Several vectors are shown in Fig. 3 as **small bars**.



(a) (b)

Fig. 3. (a) 2-D Multiple Disk Form (MDF) representation of 10-dimensional Boolean space without occlusion. (b) A 3-D version of MDF with grouping Hansel chains.

The top row bar shows vector (1111111111). The vectors on the third row cannot be identified from this figure without additional assumptions, but we can tell for sure that it has eight "1"s, because of its location on the third line that contains all vectors with norm 8. To specify it more we link each vertical position (column) with a specific vector. We use *decimal values* of Boolean vectors for this in one of the visual representations. Assume that Boolean vectors with the same norm are ordered as decimal numbers, where the leftmost column presents the largest decimal vector for the given norm, (1111111100), and the rightmost position is used for vector (0011111111) in the third row. Each level in Fig. 3a is called a disk and the entire visualization is called the **multiple disk form** (**MDF**). In MDF, every vector has a fixed horizontal and vertical position.

There is *no occlusion* of vectors in MDF. This is the fundamental advantage of the MDF representation in comparison with methods reviewed in section 1. In Fig. 3, all 1024 vectors are completely visible and their attributes are represented implicitly but unambiguously. The advantage of this procedure is to allow the user to compare more than one binary dataset or Boolean function at a time. This representation uses a distance between vectors as decimal numbers, $D_N(n(\mathbf{a}),n(\mathbf{b})) = |n(\mathbf{a})-n(\mathbf{b})|$, where $n(\mathbf{a})$ and $n(\mathbf{b})$ are decimal numbers representing vectors a and b.

The disadvantage of such MDF implementation called *process* P_0 is that distance D_N may not be relevant to the application domain. Thus, the task is to locate vectors in MDF in a way that will capture relations/structure that are *relevant* to the domain.

The *Hamming distance* D_H captures relevant relations in many domains. A partial order relation $\leq$ between vectors $\mathbf{a} \leq \mathbf{b} \Leftrightarrow \forall a_i \leq b_i$ is one that represents some ordinal structure of the attribute space. The following statement shows the link between D_H and this partial order.

Statement 6: If $D_H(\mathbf{a},\mathbf{b})=1$, then $\mathbf{a} < \mathbf{b}$ or $\mathbf{b} < \mathbf{a}$. If $\mathbf{a} < \mathbf{b}$ and $D_H(\mathbf{a},\mathbf{b})=k$, then a chain exist such that there are k-1 nodes between **a** and **b** on this chain.

Algorithm.

In the previous consideration [Kovalerchuk, Delizi, 2005] any relocation of chains in MDF was allowed. This means that Boolean vectors from different chains that are not similar can be located next to each other. That method does not control and prevents such occurrences. The vicinity of some Boolean vector **a** may contain both vectors that are similar to **a** (being on the same chain) and be dissimilar being on the other chains. In other words, this visual representation captures relations (similarity and monotonicity) between nodes along the chains, but does not ensure similarity for Boolean vectors from different chains.

Now we would like to impose limitations to ensure that chains that are located next to each other are *similar***.** We call this requirement the *Local Similarity Principle* (LSP)**.**

A **new algorithm** uses the statements and the theorem from section 2 and has three major steps:

(1) putting the largest chain H to the center of the MDF,

(2) ordering other chains relative to their closeness to H in the *averaged Hamming chain similarity measure,* $D_{HC}(U, H)$ defined in section 2,

(3) ordering chains with equal *similarity measure* D_{HC} to H relative to the location of the borders between patterns on these chains.

(4) locating chains in MDF in accordance their order obtained in (2) and (3), that is the closest chains are located next to largest chain first and all other chains are located next according their order.

Tables 8-11 show chains for E^1-E^5 located using this algorithm.

Table 8. Horizontal drawing of Hansel chains in E^1

Chain	*Norm 0*	*Norm 1*
H	0	1

Table 9. Horizontal drawing of Hansel chains in E^2

Chain	*Norm 0*	*Norm 1*	*Norm 2*
0H	00	01	11
1H		10	

Table 10. Horizontal drawing of Hansel chains in E^3

Chain	*Norm 0*	*Norm 1*	*Norm 2*	*Norm 3*
0H2		010	110	
0H1	000	001	011	111
1H1		100	101	

Table 11. Horizontal drawing of Hansel chains in E^4 in accordance with chain distances

Chain	*Norm 0*	*Norm 1*	*Norm 2*	*Norm 3*	*Norm 4*
1H3			1010		
0H2		0100	0101	1101	
0H1	0000	0001	0011	0111	1111
1H1		1000	1001	1011	
0H3		0010	0110	1110	
1H2			1100		

Table 12. Horizontal drawing of Hansel chains in E^5 in accordance with chain distances

Chain U	Distance D_{HC}(U, 00H1)	*Norm 0*	*Norm 1*	*Norm 2*	*Norm 3*	*Norm 4*	*Norm 5*
01H2	4			01100	11100		
01H3	3			01010	11010		
00H3	2			10010	10110		
00H3	2		00010	00110	01110	11110	
01H1	2		01000	01001	01011	11011	
00H1	0	00000	00001	00011	00111	01111	11111
10H1	2		10000	10001	10011	10111	
00H2	2		00100	00101	01101	11101	
10H2	3			10100	10101		
11H1	4			11000	11001		

Tables 9-12 can be converted to the visual form as shown in Fig 4 for E^{10}. It is the MDF from Fig. 3a rotated with chains of 10-D vectors from a cancer application. In Fig. 3, the white bars show benign tumor cases, the black bars cancer tumor cases

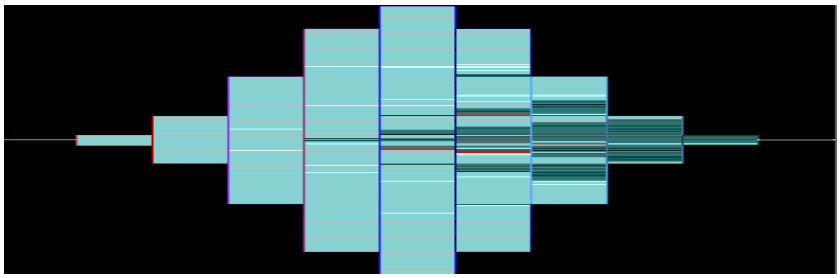


Fig. **4**. MDF in E^{10} with Hansel chains applied to cancer data: white bars represent benign tumor cases, black bars represent cancer tumor cases

4. Learning process by monotone extension and monotonization

Learning algorithms in data mining and machine learning include: (1) discovering a regularity R using training and testing data for specific classes, (2) generalizing R to new cases, and (3) applying R to classify these new cases. A complete generalization will classify all cases that is all vectors in n-D space. Often learning algorithms do not separate (1)-(3), but only produce a border between classes as a hyperplane or another discriminate surface in a n-D space. The lack of explicitly formulated regularity R is

a flaw of such algorithms. In this section, we describe a process of learning monotone regularities R (in a *n*-D Boolean space) that are explicitly defined.

Definition. Boolean function f: $E^n \to E = \{0,1\}$ is called a *monotone Boolean function* if

$$\forall\, \mathbf{x} \geq \mathbf{y} \Rightarrow f(\mathbf{x}) \geq f(\mathbf{y}) \qquad (2)$$

Definition. A Boolean function is called a *binary regularity* R if R: $E^n \to E$, where $E = \{0,1\}$ is a set of two labels (label of class 0, and label of class 1).

Definition. A binary regularity is called *a monotone regularity* if R is a monotone Boolean function.

Discovering monotone regularity. At first, we *discover monotonicity* on training data T_r by testing property (2) on all pairs of training data (vectors **x**, **y** with known R values.) The computational process can be shortened by using Hansel chains. If monotonicity of R is confirmed on T_r, then it is tested with vectors of a set Tt disjoint from Tr. If this test is also successful, then we *generalize* R that is we assume that R is a monotone regularity and test this generalization. Then we apply the generalized R as a monotone Boolean function to new cases by using **monotone extension** that holds for monotone functions

$$(\mathbf{x} \geq \mathbf{a}\ \&\ R(\mathbf{a})=1) \Rightarrow R(\mathbf{x})=1; \quad (\mathbf{a} \geq \mathbf{y}\ \&\ R(\mathbf{a})=0) \Rightarrow R(\mathbf{y})=0;$$

here **a** is a vector from training data with known class R(**a**), e.g., benign or malignant, and **x** is a new vector with unknown R(**x**). We do this expansion using Hansel chains. If **a** belongs to a chain H and R(a) = 1, then for all **x**>**a** on this chain we assign R(**x**)=1. Similarly, if **a** > **x** and R(**a**)=0, then we assign R(**x**)=0 for all **x** below **a** on that chain.

Classes and borders between classes

Definition. *Class 1* is a set of $\mathbf{x} \in E^n$ such that R(**x**) =1 and *Class 0* is a set of $\mathbf{x} \in E^n$ such that R(**x**)=0.

Let X be a subset of a chain H such that $\forall \mathbf{x} \in \mathbf{X}$ R(**x**)=1.

Definition. Let H be a chain in E^n, R be a monotone regularity, X be a subset of H. If $\forall \mathbf{x} \in \mathbf{X}$ R(**x**)=1 , then the minimal element in X is denoted by x_{min1} and is called a *lower one* in X for R.

If $\forall \mathbf{x} \in \mathbf{X}$ R(**x**)=0 , then the maximal element in X is denoted by x_{max0} and is called an *upper zero* in X for R.

If X=H then $\mathbf{x}_{max0}$ is an upper zero of the chain and $\mathbf{x}_{min1}$ is a lower one of the chain H for R. In the case of X=H, $\mathbf{x}_{max0} < \mathbf{x}_{min1}$ and $|\mathbf{x}_{max0} - \mathbf{x}_{min1}|=1$, that is $\mathbf{x}_{min1}$ is the next node on the chain after $\mathbf{x}_{max0}$. These two nodes form a *border between classes*, R(**x**)=0 and R(**x**)=1 on chain H.

Definition. A Hansel chain H is called a *0-1-Hansel chain*, if node **x** with R(**x**) = 0 and a node **y** with R(**y**) = 1 exist on H. Such a chain is denoted as $H_{0\text{-}1}$.

Nodes **x** and **y** can come from different sources. The first one is training data Tr. Let X be a subset of Tr on the chain H, $X \subseteq Tr \cap H$, such that it contains **x** and **y**

nodes, R(**x**)=1 and R(**y**)=0. Other sources of **x** and **y** could be experts and models of the domain.

It follows from the monotonicity of R, that there is exactly one maximal node $x_{max0,H}$ on H with $R(x_{max0,H})=0$, and exactly one minimal node $x_{min1,H}$ on H with $R(x_{min1,H})=1$; and $x_{max0,H} < x_{min1,H}$.

Definition. The *R-border on a 0-1-Hansel chain* H is the set of pairs $\{(x_{max0,H}, x_{min1,H}) \mid H \text{ is a 0-1-Hansel chain}\}$.

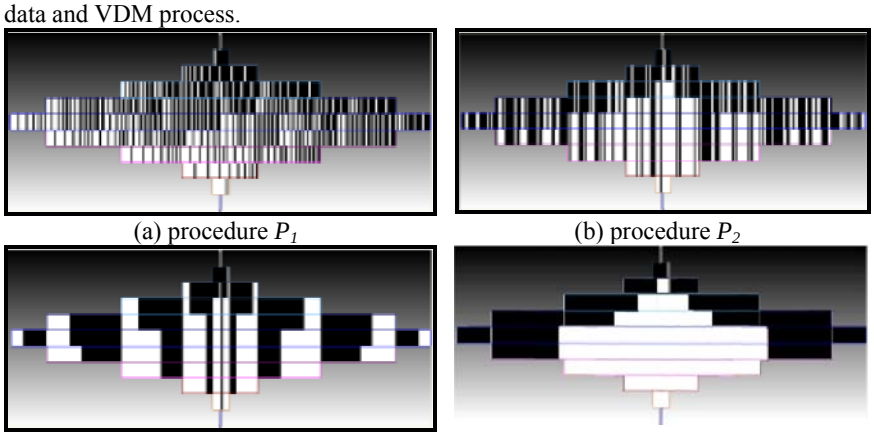
Definition. The R-border on a set of *0-1-Hansel chains* $\{H\}_{0\text{-}1}$ in E^n is a set of all the R-border on all 0-1 Hansel chains {H}.

Definition. The *width of the border* between classes in E^n on chain H for R is the Hamming distance between $\mathbf{x}_{max0}$ and $\mathbf{x}_{min1}$ on H, $|\mathbf{x}_{max0} - \mathbf{x}_{min1}|$.

The smallest width of this border is $|\mathbf{x}_{max0} - \mathbf{x}_{min1}|=1$. A wider border between classes indicates a better separation of classes for a given training dataset.

5. Computational Experiment

We conducted several computational experiment that show wide and narrow borders between patterns with different level of border clarity. This depends on both data and VDM process.



(a) procedure P_1 (b) procedure P_2

(c) procedure P_3 (d) procedure P_4

Fig. 5. Different visual border representations of vectors of two classes in MDF for the same simple monotone Boolean function.

Different way of locating Boolean vectors on MDF form produce very different results relative to the shape of the border between classes (see Fig. 5). They range from no border visible (Fig. 5a) to a very clear border (Fig 5d). Experiments with real breast cancer data show advantages of this approach to uncover a visual border between benign and malignant cases in breast cancer.

Fig. 4 shows analysis of multivariate monotonicity of breast cancer cases that is almost perfect with four exclusions that are shown in red. The MDF design with

Hansel chains located horizontally expands scalability of this visualization. To be able to see a large number of attributes a user can use zooming or scrolling MDF.

6. Conclusion and Future Work

Visual data mining had shown benefits in many areas. However, classical VDM methods do not address the specific needs for processing data that are highly overlapped in the visual space. This paper proposed a structural VDM approach and demonstrated its effectiveness in a medical application. To use this approach for Boolean vectors of a high dimensionality *n* we visualize only a part of the data structure that is actually needed for given data.

We exploit monotonicity properties to build the *complete visual border* between two classes. This is an advantage relative to many traditional machine learning and data mining methods.

The proposed method allows discovering the border between classes for a monotonized set of diagnostic features. In mathematical terms, this task is equivalent to the task known in discrete mathematics as *restoration of a monotone Boolean function*. The main idea of this process is based on decomposition of the binary cube E^n into Hansel chains that partition E^n lattice without overlap of chains.

It is a proven a theorem that all nodes with equal norms of adjacent Hansel chains have the same Hamming distance. It is also shown that if the Hansel chains are not adjacent this property is not true. It is shown that the constant distance hypothesis for Hansel chains holds in E^2-E^4, but it fails in E^5. As a future research it will be interesting to explore if another partitioning set of chains exists in E^n that differs from the set of Hansel chains that satisfy the constant distance hypothesis for E^5 and for greater n.

Using the results of the mentioned theorem and associated statements 1-6 we proposed an algorithm to locate Hansel Chains on the Multiple Disk Form for visual data mining. The advantage of this algorithm is that it preserves similarity and monotonicity of n-D data visualized in 2-D or 3-D.

Multiple non-monotone data can be monotonized. Therefore, future studies in monotonization include discovering limits of local monotonicity, and the decomposition of non-monotone Boolean functions and attributes into a combination of monotone Boolean functions.

By further developing these procedures for non-monotone Boolean data and k-valued data structures, this approach can be used in a variety of applications including tasks, where data are dynamically changed/updated and the visual border between patterns also dynamically changes.

Acknowledgement

The authors are grateful to Karl Erich Wolff and the anonymous reviewer for valuable comments and suggestions.

References

1. Beilken, C., Spenke, M., Visual interactive data mining with InfoZoom -the Medical Data Set. The 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD '99,
2. Birkhoff G., *Lattice theory*, AMS,1995
3. Card, S. K., Mackinlay, J, The structure of the information visualization design space. In Proc. IEEE Symposium on Information Visualization, pages 92–99, 1997
4. Ebert, D., Shaw, C., Zwa, A., Miller, E., Roberts, D. Two-handed interactive stereoscopic visualization. IEEE Visualization '96 Conference.1996, 205-210.
5. Hansel, G. , Sur le nombre des fonctions Booléennes monotones de n variables. C.R. Acad. Sci. Paris, 1966, v. 262, n. 20, 1088-1090.
6. Grätzer, G. ,*Lattice theory: first concepts and distributive lattices.* W. H. Freeman and Co., 1971.
7. Groth, D., Robertson, E., Architectural support for database visualization, Workshop on New Paradigms in Information Visualization and Manipulation, 53-55,1998, 53-55.
8. Inselberg, A., Dimsdale, B., Parallel coordinates: A tool for visualizing multidimensional Geometry. Proceedings of IEEE Visualization '90, Los Alamitos, CA, IEEE Computer Society Press, 1990, 360–375.
9. Keim, D., Ming C. Hao, Dayal, U., Meichun Hsu. Pixel bar charts: a visualization technique for very large multiattributes data sets. Information Visualization, March 2002, Vol. 1, N. 1, pp. 20–34.
10. Kovalerchuk, B., Triantaphyllou, E., Despande, A.,Vityaev, E., Interactive Learning of Monotone Boolean Functions. Information Sciences, Vol. 94, issue 1-4, pp. 87–118, 1996.
11. Kovalerchuk, B., Vityaev, E., Ruiz, J., Consistent and complete data and "expert" mining in medicine. Medical Data Mining and Knowledge Discovery, Springer, 2001:238–280.
12. Kovalerchuk., B., Delizy F.,Visual Data Mining using Monotone Boolean Functions, In: Visual and Spatial Analysis (Eds. Kovalerchuk B., Schwing J.), Springer 2005, 387-406.
13. Last, M., Kandel, A., Automated perceptions in data mining, invited paper. IEEE International Fuzzy Systems Conference Proc. Part I, Seoul, Korea, 1999, pp. 190–197.
14. de Oliveira, M., Levkowitz, H., From Visual Data Exploration to Visual Data Mining: A Survey. IEEE Trans. On Visualization and Computer Graphics, 2003, Vol. 9, No. 3, pp. 378-394
15. Peng, W., Ward, M., Rundensteiner, E., Clutter Reduction in Multi-Dimensional Data Visualization Using Dimension Reordering, *Proc. IEEE Information Visualization,* pp. 89-96, 2004.
16. Post, F., T. van Walsum, Post, F., Silver, D., Iconic techniques for feature visualization. In Proceedings Visualization '95, pp. 288–295, 1995.
17. Ribarsky, W., Ayers, E., Eble, J., Mukherja, S., Glyphmaker: creating customized visualizations of complex data. IEEE Computer, 27(7), 57–64, 1994.
18. Shaw, C., Hall, J., Blahut, C., Ebert, D., Roberts. A., Using shape to visualize multivariate data. CIKM'99 Workshop on New Paradigms in Information Visualization and Manipulation, ACM, 1999, 17-20.
19. Thomas,J., Cook, K., Eds. Illuminating the Path: The Research and Development, Agenda for Visual Analytics. National Visualization and Analytics Center, 2005. http://nvac.pnl.gov/agenda.stm
20. Ward, M., A taxonomy of glyph placement strategies for multidimensional data visualization. Information Visualization 1, 2002, 194–210.
21. Wilkinson, L., Anand A., Grossman R., High-Dimensional Visual Analytics: Interactive Exploration Guided by Pairwise Views of Point Distributions, IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 6, pp. 1363-1372, Nov/Dec, 2006
22. Di Yang, Rundensteiner E., Ward, M., Analysis Guided Visual Exploration of Multivariate Data, Proceedings of the IEEE Symposium on Visual Analytics, 2007.