Visualization of Multi-Dimensional Data with Vector-fusion

RR Johnson
University of Utah
nDV-LLC

ABSTRACT

Multi-dimensional entities are modeled, displayed, and understood with a new algorithm vectorizing data of any dimensionality. This algorithm is called **SBP**; it is a vectorized generalization of parallel coordinates. Classic geometries of any dimensionality can be demonstrated to facilitate perception and understanding of the shapes generated by this algorithm. **SBP** images of a 4D line, a circle, and 3D and 4D spherical helices are shown. A strategy for synthesizing multi-dimensional models matching multi-dimensional data is presented. Current applications include data mining; modeling data-defined structures of scientific interest such as protein structure and Calabi-Yau figures as multi-dimensional geometric entities; generating vector-fused data signature "finger prints" of classic frequency spectra that identify substances; and treating complex targets as multi-dimensional entities for automatic target recognition. SBP Vector Data Signatures apply to all pattern recognition problems.

CR Categories: H.2.8 Database Applications, Data Mining; I.2.6 Learning, Knowledge acquisition; I.3.3 Picture/Image Generation, Viewing Algorithms; H.1.2 User/Machine Systems, Human Information Processing; H.5.1 Multimedia Information Systems, Animations, Artificial, augmented, virtual realities; I.2.10 Vision & Scene Understanding, Representations, data structures, & transforms; I.5 Pattern Recognition; I.5.4 Applications, Computer Vision.
Additional Keywords: Multidimensional Visualization, Vector Data Fusion, Multidimensional Geometry.

1. OVERVIEW

Presented here is a new way to see structures of any dimension. The new way begins conventionally by showing a 3rd dimension, D3, along a line at an angle α in 2 dimensions, **Figure 1**. Each of the next dimensions would be shown at its own different angle in an expanded Figure. The structures to be seen are described by data from a spreadsheet. The columns are called dimensions and the rows are called vectors. The cells and their values are treated as component vectors, each is drawn at its own different angle as assigned in that expanded Figure.

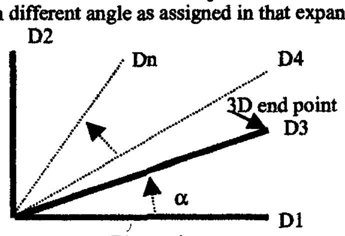


Figure 1

Robert R. Johnson is a Life Fellow of IEEE, Professor Emeritus in Computer Science at University of Utah, and Managing Partner of nDV-LLC.
Mailing address:
3857 S. Eagle Point Dr, Salt Lake City, UT. 84109
email: john97john@aol.com

SBP is an algorithm that adds each component vector to the prior one, summing the whole row to a single-end-point resultant. This vector summing process for n dimensions is the same as that done for 3 dimensions summed to a 3D point in **Figure 1**. A 3D and 5D vector sum is shown in **Figure 2.**

Images created by **SBP** are not intuitively obvious. Geometric examples of a 4D straight line, a 2D spherical helix (a circle), a 3D spherical helix, and a 4D spherical helix are given to initiate familiarization, see **Figure 4.** A helix with a small pitch looks like a surface. A web site is cited [6] that gives other examples including 12D data (a spreadsheet with 12 columns). That site is animated. Animation improves understanding of **SBP** structures.

Several possible future applications of **SBP** are cited. An example application which would use the technique described for a Vector Data Signature of frequency spectra could be dynamic generation of a single number computer ID (in 3 dimensions) for one's own written signature spectrum.

An application to scientific data could be the generation of multi-dimensional structural models for entities such as proteins, the structure of which is traditionally displayed as a complex ribbon in 3 dimensions.

2. BACKGROUND

Visualization entails building models of entities for display, hopefully revealing their structure and the relationships among their elements. Entities of interest for multi-dimensional visualization are systems described by data; the elements of that data are typically presented in a spreadsheet. The *columns* are called dimensions, and the *rows* are instances where *values* in the cells of the row are examples satisfying the conditions of that system.

A general structure for visualizing multi-dimensional datasets is parallel coordinates [1],[3]. Parallel Coordinates allow one to see any number of dimensions concurrently by arranging the coordinates parallel to each other. Shortcomings of version [3] of parallel coordinates include:

1. One point from nD orthogonal space maps into n points in parallel coordinates,
2. Straight lines in orthogonal space do not map into straight lines in parallel coordinates,
3. The sequence in which the parallel co-ordinates are considered determines the apparent relationships among the elements of the data defined system. This is the "pesky problem" cited by Inselberg of determining the proper sequence in which to consider the coordinates. **nDV**'s version of parallel coordinates [4],[6] eliminates shortcoming (2), but (1) and (3) remain common to Inselberg's and **nDV**'s versions of parallel coordinates.

Key problems and thorny issues in visualization of multi-dimensional entities are cited in [2]. **SBP** appears to satisfy the 3 issues cited: a geometry is illustrated here, that geometry is used to initiate intuitive perception, and evaluations via applications are being undertaken. Much work remains; this is just a glimpse, seeing some multi-dimensional entities, evaluating the process, and planning what remains to be done.

Explanations of **nDV**'s approach to parallel coordinates are in [4] and [6]. Parallel coordinate perspectives are especially important in the analysis of unknown data structures. This will be summarized in **4.3** below.

The background for overcoming the shortcomings of parallel coordinates is the basic technique for showing a 3rd dimensional coordinate in 2-space by drawing it at an angle between the first two orthogonal coordinates, **Figure 1**.

3. INTRODUCTION

Seeing the 4th and higher dimensions is easy when *these higher dimensions are not at right angles to each other. These higher dimensions are added as additional lines in Figure 1, each at their own unique angle.*

SBP (Single-point Broken-line Parallel-coordinates) [6] is an extension of parallel coordinates where the parallel coordinates are broken into pieces associated with the vectorized value of each dimension. These component vectors can be summed to give a single-end-point resultant for each row of the spreadsheet, as in **Figure 2**. This new algorithm overcomes all 3 shortcomings cited above for visualization via parallel coordinates.

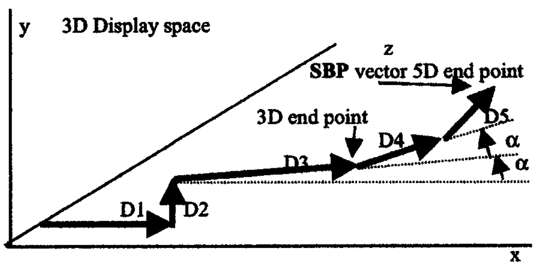


Figure 2 - **SBP** vector principle
(x,y, z define Display space)

In 2-space, *plotting n coordinates each at their own incremental angle α in Figure 2 which is less than 90°* makes it practical to draw multi-coordinate data and to generate single-end-points for each row in that data. Doing so provides an algorithm for data of arbitrary dimensionality that is extendable, generalizeable, and visualizable. **SBP** extends the classic technique of showing a third dimension at an angle in 2-space. **SBP** uses a different angle in 2-space for each higher dimension where α from **Figure 2** is a standard increment added for each dimension.

Understanding the relationships and structure of **SBP** models is best done using animated **SBP** models of simple geometric shapes as demonstrated on nDV's web site: www.globalsvcs.com/ndv^ [6].

To recap these introductory concepts: using data as presented in a spreadsheet, **SBP** treats each column's data as a component vector. The amplitude of this component vector is the value in the cell, and the column's identification designates the unit (angle) vector for this component vector. **SBP** generates the vector sum of all component vectors in a spreadsheet row as in **Figure 2**. The single-end-point vector resultant for each row is *Vector-fused data.* Each row of data in a spreadsheet is summarized as a single 3D point in Cartesian display space. **SBP** generates these single-end-points of n dimensional data from the n component vectors of each row. **SBP** facilitates numerical analyses of data based on vector fusion; facilitates visualization of data that has been simplified via vector fusion; and reduces huge databases in an interesting fashion.

4. SBP MODEL

4.1. Description

"Points" are a critical concept when visualizing the values of a spreadsheet. The concept of the tag associated with each point is also important.

A point consists of:

A. a value (a number), and
B. its tag that consists of:
   1. A designator indicating columnar association (unit angle vector in 2D or [x,y] display space),
   2. A row number association (the unit vector z associated with depth in 3D display space), and
   3. The name of the spreadsheet.

To minimize visual clutter, tags are not printed in Display space.

The explicit construction used by **SBP** is summarized in **Figure 3. SBP** is also explained and demonstrated in [4], [6].

Pairing of all component vectors was chosen to create images of 4D and 5D cubes similar to those drawn manually. The two component vectors in each pair are plotted at 90° to each other, as are the first two dimensions in **Figure 1** and **Figure 2.** Pairing is not necessary in general for **SBP;** the component vectors could each be at their own angle as shown for subsequent dimensions in **Figure 2.** But pairing is helpful in visual comparisons with classical structures, and pairing into orthogonal sets is very helpful in understanding the complex shapes generated for multi-dimensional entities. The effect of pairing is demonstrated statically in **Figure 4** for View 2 of the circle and View 2 of the 4D Spherical Helix; the right-angle line sets in red are the orthogonally paired component vectors.

Helices of the various curved geometric shapes are created in nDV's Geometry since data point summations are lines or curves, not surfaces.

In **Figure 3,** two small spreadsheets are given; the first is for 4-dimensional **SBP** vectors, the second is for 6-dimensional vectors. Three rows of zeros space the vectors apart. The first model illustrates 4 **SBP** component vectors, each of 4 dimensions (Dimensions B,C,D,E from the upper spreadsheet). Note that the first two vectors are comprised of 4 unit-length components, one for each dimension. The second two vectors have differing length components as defined in their spreadsheet. The four vectors correspond to the four rows: Row 3, Row 4, Row 8, and Row 9 of the spreadsheet. Each vector is made up of components from each dimension (column). The images are prints of actual **SBP** component vectors from their defining spreadsheets with these Excel row and column labels.

An important detail of **SBP** component vectors is that they are plotted as vector resultants – i.e., each vector component is added as a vector to the previous component vectors. *Orthogonal pairs of successive component vectors are plotted at* *successive angles with respect to each other.* In the case of 4 dimensions, the second pair of component vectors is plotted at 45° with respect to the first pair. In the case of 6 dimensions, the second pair is plotted at 30° with respect to the first pair, and the 3rd pair at 30° to the 2nd pair or at 60° to horizontal.

The general rule is to plot paired orthogonal sets of component vectors at ($\alpha = 180° / n$) degree increments where n is the number of dimensions involved.

The lower spreadsheet in **Figure 3** defines four vectors each of 6-Dimensions (B, C, D, E, F, G) that are separated from each other by 3 rows of zeros (to provide visual separation). The four vectors are defined for the cell values in rows 22, 23, 27, 28.

The Edge View is a rotation of the **SBP** models in the first view. Each pair of orthogonal component vectors is plotted at a slight "Broken-line" piece of "Parallel-coordinate" distance, and it is plotted into the "z" or "depth" direction of the 3D display space used for **SBP** visualizations. The Edge View shows a small "Broken-line" offset, and it shows the larger parametric steps of the defining parametric equations in the depth or z direction of display space.

SBP models are best viewed dynamically in conventional 3D display space [6]. Dynamic or animated viewing of **SBP** models, and of **nDV**'s Parallel Coordinate models, conveys deeper understandings of the structures involved than is possible with the static images presented in the original Parallel Coordinate models [3].

4.2. Characteristics of SBP Data Models

The principal attributes of **SBP** models are:

A. Great reduction in the visual complexity from that of the Parallel Coordinate models shown in [3] and [4] when only **SBP** end-points are displayed.
B. The n points (per n-D vector) in Parallel Coordinates are reduced to one resultant point.
C. The coordinate sequence dependency of Parallel Coordinate models is eliminated [4],[6].
D. Shapes seen are not intuitively obvious initially. Multi-dimensional structures are complicated.

***SBP** vector end points are commutative with respect to column sequence. Commutativity arises directly as one property of vector addition.*

Understanding **SBP** mapping is helped by observing and studying **SBP** models of classic geometries, **Figure 4**. Since these geometries are generated by parametric expressions, "solid" shapes are helices of cylindrical, spherical, or toroidal form.

Straight lines from classical orthogonal space appear as straight lines, and circles as circles in these multi-dimensional images of SBP. ***These characteristics are also properties resulting from vector addition.***

3D **SBP** spherical helix structures have the characteristic "spherical" line signature of **Figure 4.** The "sphericity" in **SBP** is not that of 3D spherical helices in orthogonal space because SBP models are basically 2D representations of n dimensions. Helices with very small pitch and many points can appear as surfaces. Note the shape of the particular 4D spherical helix in **Figure 4**. One needs to see these shapes in animation to understand and recognize the signature forms of basic geometric shapes in **SBP**.

4.3. Purpose of Geometric Images

Visualization studies can comprise at least four generic kinds of analyses, all of which draw upon geometric images for understanding:

A. Analysis of each coordinate's data to determine its internal structure [4].
B. Determining if the multi-dimensional data implies the presence of any higher geometric symmetries. In other words, can the data be fitted to any classical multi-dimensional geometries?
C. Utilizing Vector Data Signature capability to identify and simplify complex entities.
D. Research into the power and meaning of the characteristic signature shapes generated by **SBP**.

Geometric images demonstrate **SBP** signatures for a range of n-D models using classic simple geometric shapes. **SBP** straight lines, circles, and spherical helices of 3,4, 6 and 8 dimensions are given in [4], [6].

Parametric expressions for geometric shapes are a key tool for studying data structure in Parallel Coordinates and in **SBP**. Parametric expressions reveal elements that create specific geometric shapes. Use of parametric representations enables mathematical generalization processes for creating successively higher dimensional versions of those shapes. As shown in the equations of **Figure 4**, higher dimensional definitions in parametric equation form are then used to generate images of those higher dimensional structures. Familiarity with **SBP** images of high dimension geometric shapes, and knowledge of the Fourier components that constitute such shapes is very helpful in syntheses matching data to the appropriate high dimension geometry. **nDV**'s parallel coordinate images are used to understand each dimension's structure by itself. **SBP** images display the cumulative structural effect of all dimensions. **SBP** images are easy to generate and examine. They also show the effect of all n dimensions on each other.

For data of unknown structure, Fourier analyses of each dimension's data down each column gives the sine and cosine makeup of each dimension. With this Fourier knowledge, the complete spectrum of circular or spherical components describing each coordinate's structure is identified. **4.3.B.** above uses this process to identify higher geometric symmetry.

Understanding of structure is enhanced by viewing the animated, fly-by, structures in display space (both Maple for 3D and SBP for nD) compared to static prints.

5. VECTOR-FUSED DATA SIGNATURES

5.1. Derivation

SBP generates a single point 3D vector resultant from the data in all n columns for each row of the spreadsheet. This discrete point is the vector resultant of that row, providing a 3D end-point identifying the n dimensional data of that row. This **SBP** vector can be visualized in 3D display space, and it can be manipulated mathematically; this end-point resultant symbolizes that data, and it is a repetitive, reproducible, quantitative measure of that data.

The single-end-point is the *vector-fused signature* of that data.

The set of component vectors are the unique vector definition of that data because the single-end-points could possibly be generated by other component vector combinations. When concerns about uniqueness arise, the set of component vectors provide unique identification.

5.2. Applications

A key application of Vector-fused Data Signatures is identification of substances by their infrared, Raman, x-ray, sonic, or other spectra. Spectral analyses are powerful tools in science used to characterize substances and their structure. A spectrum usually arises as an analog image generated by detectors of energy and drawn typically as a function of frequency. Such spectra are intricate patterns that are difficult to recognize visually.

Vector-fused Data Signatures provide a new way to identify spectra: *immediate digital recognition of a substance by its vectorized spectral signature.*

The analog image of a spectrum can be converted to digital data for representation in a spreadsheet by sampling the frequency and amplitude values of that image. An example of this is given in the next section. Quantization of images needs to be at a sufficiently high resolution to capture all important details. Each sampling frequency is considered to be a dimension so that columns correspond to discrete frequencies. A spreadsheet row of amplitude values at those sampling frequencies is the digitized spectrum.

The **SBP** vector sum for the row of amplitude values at each sampling frequency is a single 3-dimensional point in **SBP** display space. This single-end-point is the Vector (**SBP**) Data Signature for the values in that row. Vector Data Signatures can be identified, compared, added, multiplied (weighted), and visualized as a single point in normal 3D display space.

Vector Data Signatures are single point identifiers (in 3 dimensions) of multi-dimensional entities having any number of dimensions.

Identification of spectra with their Vector Data Signature can become a powerful way to see, manipulate, identify, and understand substances and mixtures of substances.

5.3. Example: a Vector Data Signature for Spectra.

The technique for identifying spectra using **SBP** begins by representing a typical spectrum as a row in a spreadsheet. The following examples of spectra are from reference [5]. The single-point 3D **SBP** Vector Data Signatures (DS) of the spectra of some chemicals are [DS_{31} = (value of **x**, value of y, value of **z**) in **SBP** space]:

tert-butylacetylene: DS_{31} = (149,1534, -1.5)
diethyl-ether(mix) DS_{31} = (96, 942, -1.5)
(adulterated mixture with tert-butylacetylene)
tert-butyle group DS_{31} = (78,1553,-1.5)
isopropyl group DS_{31} = (151,1315,-1.5)

The subscript indicates the number of sample points per spectrum; here 31 samples are taken per spectrum. The larger the number of samples, the more accurate the signature. *For identification purposes, the same number of sample points needs to be used for each spectrum.* Sampling of chemical spectra with 1800 sample points appears to be proving adequate and accurate.

To understand how to use **SBP** generated signatures, begin by assuming that a simple line spectrum looks like:

Further assume that there may be, for example, 1800 possible spaces for a spectral line, some of which are filled as sketched, most of which have nothing. The set can then be considered a row of a spreadsheet 1800 columns wide. Some cells have a value, in this case unit value, the others zero. Column designation corresponds to sampling frequency (or wavelength). **SBP** treats the row index as a small uniform offset in the **z** direction (as in the Edge View of **Figure 3**) and cumulatively generates the values for **z** shown in the DS_{31} signatures above.

Making an **SBP** model from one row of a spreadsheet generates one 3D point as the vector resultant of all 1800 component vectors (coordinates). Any time that this specific spectrum is encountered, there will be one point at that location in **SBP** space, **Figure 5**. Many different spectral measurements of the substance that generates this specific spectral pattern will generate a cluster of points. The better the measurements, the tighter the cluster. Similarly for a second substance B; its spectrum will create a point at a different location in **SBP** display space.

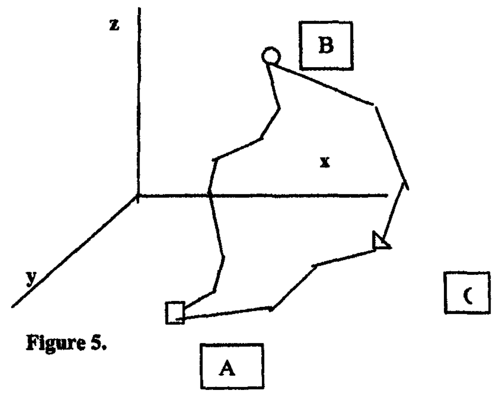


Figure 5.

Assume that there are 3 clusters representing 3 substances at A□, B ○, and C◺ . Anytime a spectrum is taken of one of those substances, there will be another point added to one of the clusters. Assume that one wishes to determine the % mix of substances A and B. To do this, one needs to map the profile of points that represent increasing percentages of B as one moves from A to B. The line shown in **Figure 5** indicates one possible profile. Once calibrated, the spectrum of a mixture of A and B generates a known point on this profile. Similarly, one can calibrate percentage mixes of C going from A to C, and then from C to B. Calibrating these spectra for mixes of A,B, and C will generate a surface contained inside these boundaries.

The spectrum of a sample containing an unknown mix of these substances will generate a point that fits on this surface. Once such a surface has been calibrated, one sample spectrum of the unknown will immediately identify which substances are in the mix as well as their percentages in that mix.

The sample spectrum above had only "ones" in it – i.e., a frequency line was either present or not. Actual spectra, such as those used to generate the DS $_{31}$ signatures above, have values associated with each frequency (the amplitude at that frequency). **SBP** generates vectorized single-end-points for any set of values in a row across a spreadsheet. Any spectrum (or any other kind of data such as target data) can be represented as a row in a spreadsheet. Its vector signature can then be generated by **SBP**.

5.4. Meaning of Data Signatures

Vector data signatures are a computer manipulatable "finger print" for complex, multi-dimensional entities. **SBP** end-points are a 3-dimensional signature for an n-dimensional entity. However, it is possible that other n-dimensional entities can generate this same 3D end-point.

*Uniqueness of an **SBP** signature is confirmed only by comparison of the authentic set of component vectors with the set of component vectors comprising the unknown vector to be identified.*

6. CLASSES OF APPLICATIONS

6.1. ATR.

Automatic target recognition is an application where targets (or objectives) are characterized by many experimentally determined criteria. A particular example of "target" recognition is with Vector Data Signatures. As in the case of identifying specific spectra, "target" characteristic signatures can be assigned to any image or relationship that can be articulated. Experimental descriptions, image descriptions, numerical characterizations are appropriate. Vector Data Signatures identify anything that can be quantified.

6.2. Multi-dimensional models of scientific data

Fitting multi-dimensional data to multi-dimensional geometries is a new research capability enabled by being able see multi-dimensional structures and by being able to synthesize multi-dimensional structures from the data of each dimension. This is **4.3.B** above.

Proteins, for example, are typically modeled as long, twisting, folded ribbons in 3D space. These models are typically assembled from many samples of 1D data structured to fit into a three dimensional image. Approximately 14 different canonical fold structures are used in the 3D synthesis. The resultant is the long ribbon structure used to illustrate a protein in 3D drawings.

One issue is, what are the "native" dimensions of a protein, and a second issue is, how many dimensions are there? An nD visualization hypothesis is that protein data could be fitted to an 8-dimension toroidal helix. The nD torus is a more general structure than are nD spheres. nD spherical helices are synthesized by Fourier components only.

Fitting structures from nature to multi-dimensional geometries is greatly facilitated by being able to see and treat the data as multi-dimensional. Seeing data in its appropriate nD geometry may reveal more symmetry or relevant structure of that data than do multi-dimensional models viewed with only three dimensions at a time in the manner of Maple or Mathematica program packages.

7. SUMMARY

Data-defined multi-dimensional entities can be modeled and visualized using either **nDV's** Parallel Coordinates or **SBP** Vector Coordinates. These new capabilities use two different techniques to facilitate understanding: animation, and mappings that maintain straight lines as straight lines to assist intuitive perception. **SBP** generates single-end-point geometries.

The techniques are of a general utility. They can be used to model, show, see, analyze, and to come to understand the relationships and structures of any multi-dimensional system.

These techniques are extendable:

A. They are being used to visualize systems that are "small" in the sense of a person's ability to see and perceive entities of relatively simple complexities, and
B. They can be applied to any size data. **SBP** Vector Data Signatures are applicable to all pattern recognitions and reduce the amount of data to be analyzed or visualized.

Further work remains to fully characterize **SBP** mappings and to verify the results mathematically. The purpose of models and visualization techniques is to facilitate the understanding of structures within multi-dimensional datasets. These models are intended to help answer the question: *What is this data telling us?* However, other techniques classically used to analyze data are still relevant. Visualization of multi-dimensional entities is expected to be of significant assistance in directing and using other data analysis methodologies. *It helps to see it.*

The new capabilities for presenting multi-dimensional visual models lead to new approaches for data analysis. Synthesis of models for multidimensional data using, for example, Fourier techniques, is a new capability resulting from the work presented here.

These new methods support construction, use, and interpretation of entities that are believed to be usefully represented as multi-dimensional systems. Many systems of scientific interest may be modeled as multi-dimensional entities because such systems are often generated or characterized by many separate parameters. Examples considered for the **SBP** approach are protein structures, Calabi-Yau figures in super-string theory, and the ultimate structure of the universe.

References

[1]. Card,S.K., Mackinlay,J.D., Shneiderman,B. *Readings in INFORMATION VISUALIZATION, Using Vision to Think*, Morgan Kaufman, 1999

[2]. Grinstein, G., Inselberg, A., Laskowski, S., Key Problems and Thorny Issues in Multidimensional Visualization, *Proceedings Visualization '98*, IEEE. pgs. 505-6. ISDN 0-8186-9176-x

[3]. Inselberg, A., Dimsdale, B., Parallel Coordinates, A Tool for visualizing multivariate Relations, *Human-Machine Interactive Systems*, Plenum Publishing, 1991

[4]. Johnson, RR., Millar,R., Palmer,M., Palmer,E., U.S. Patent # 5,917,500 Issued 6/29/99

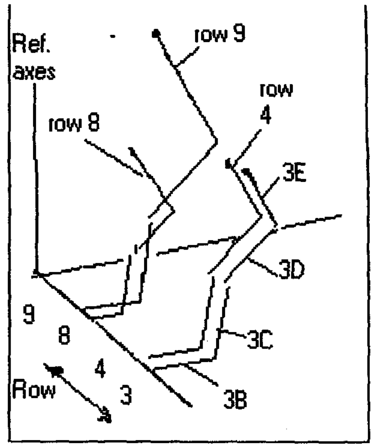
[5]. Lambert,J.B., Shurvell,H.F., Lightner,D., Cooks,F.G., *Introduction to Organic Chemistry,* Macmillan, 1987, pgs. 183 and 201.

[6]. nDV's web site: www.globalsvcs.com/ndv^

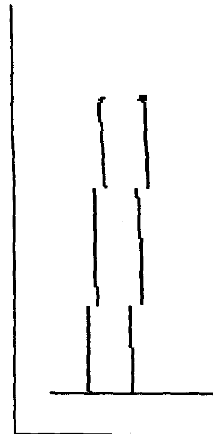
	Dimension B	Dimension C	Dimension D	Dimension E
Row 3	1	1	1	1
Row4	1	1	1	1
	0	0	0	0
	0	0	0	0
	0	0	0	0
Row 8	0.5	0.7	1	1.2
Row 9	1	1.2	1.5	2

Defining Spreadsheet

4 dimensions (B,C,D,E) with cell values corresponding to component vector lengths Bi,Ci, Di, Ei for the 4 component vectors of rows 3,4,8,9



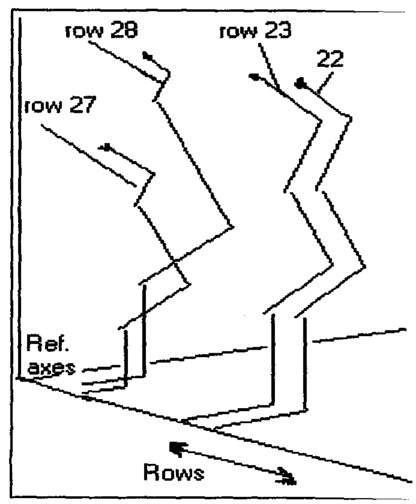
SBP Single-end-point representation in 3d display space of these 4 component vectors: rows 3,4,8,9



Edge view of first tw
6d SBP vectors, r
22, 23

	Dimension B	C	D	E	F	G
Row 22	1	1	1	1	1	1
Row 23	1	1	1	1	1	1
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
Row 27	0.5	0.7	1	1.2	0.5	0.7
Row 28	1	1.2	1.5	2	0.5	0.5

Defining spreadsheet for four six dimensic on vectors: (Rows 22,23,27,28)



Four 6d SBP vectors as modeled in 3d SBP Display Space

Figure 3: SBP Vector Structure