Report On Multiscale Visualization Using Data Cubes

Rakesh Rajpurohit MT2013122 IIIT-Bangalore

***Abstract*—Multiscale pan-and-zoom systems are effective because it support the approach of overview of the data before refining their view for more focused and detailed. But, generating abstract overview of large data sets is difficult, most of the visual system use only visual abstraction and these systems provide them only single zooming path on their data and thus a single set of abstract views. This paper mainly describes:**

1) **a multiscale visualization of data cubes with data + visual abstraction.**
2) **a method for traversing nodes at different levels of zooming garph for independently zooming along one or more dimensions.**

This paper describes the four design patterns for creating multiscale visualization using that formalism.

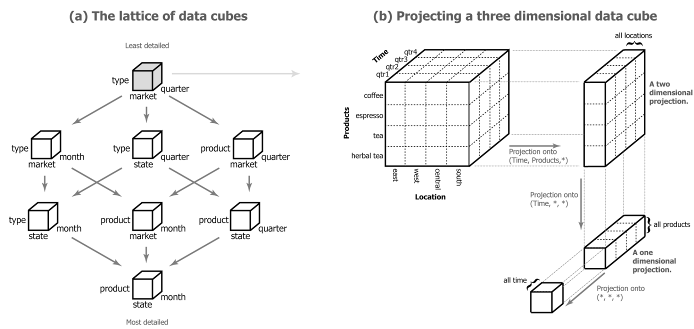

Fig. 1. Figure 1: (a) The lattice of data cubes for a data base with three dimensions: Products (with levels Type and Product), Time (with levels Quarter and Month), and Location (with levels Market and State). (b) Several projections of the least detailed data cube in the lattice.

I. Introduction

The Multiscale Visualization works on Overview first, zoom and filter, then details-on-demand. At a high level the data size is very large, So the data is highly abstracted as user zooms the data density decreases and more detailed data can be presented. Basicaly two types of abstraction performed Visual and data abstraction. Data abstractions change the data before mapping to visual representation. It includes the aggregation, filtering, sampling and statistical summerization. Visual abstractions only change the visual representation of the data points not the underlying data. Data cubes are used for abstracting and summarizing relational database and Polaris, a tool for visual exploring relational databases.The data cubes are used not only because they provide a powerful mechanism for data abstraction, but also because many large and important data sets are already stored in relational databases and data cubes. This paper presented the zoom graph and four design patterns for describing and developing multiscale visualization of hierarchically structured data.

II. Related Works

Existing multiscale visualization Techinques are:-

- Cartography: small scale maps of large scale source
- Multiscale information visualization
  - Pad++: alternate desktops [2]
  - DataSplash: layer manager of graph.
  - XmdvTool: multiscale views using hierarchical clusters
  - ADVIZOR: multiple visual metaphors

Main limitations of these are One zoom path and Primarily visual abstraction

III. Data Cubes

Data cubes categorize information into two classes: dimensions and measures, corresponding to the independent and dependent variables, respectively. For example, U.S. states are a dimension, while the population of each state is a measure. Within a cube, the data is abstractly structured as an n-dimensional data cube. Each axis corresponds to a dimension in the data cube and consists of every possible value for that dimension. Every cell in the data cube corresponds to a unique combination of values for the dimensions.Each cell contains one value per measure of the data cube.Thus far, we have considered dimensions to be flat structures. However, most dimensions have a hierarchical structure. For example, rather than having a single dimension state, we may have a hierarchical dimension location that has levels for country, state, and county. If each dimension has a hierarchical structure, then the data must be structured as a lattice of data cubes, where each cube is defined by the combination of a level of detail for each dimension.Abstraction in this model means choosing a meaningful summary of the data. Choosing a data abstraction corresponds to choosing a particular projection in this lattice of data cubes:

- which dimensions we currently consider relevant.The relevant dimensions identifies which projection (from n-dimensions down to the number of relevant dimensions) of that cube is needed.
- the appropriate level of detail for each relevant dimensional hierarchy. Specifying the level of detail identifies the cube in the lattice,

Figure 1 shows a simple lattice and projection.

IV. Polaris

A Polaris specification uses a formal table algebra to specify the table configuration of the visualization. Each expression in the table algebra defines an axis of the table: how the table is divided into rows or columns. The main components of an expression are the operands and the operators. Each operand is the name of a field and can be one of two types: a dimension is an ordinal operand (O) while a measure is a quantitative operand (Q). The type of the operand determines how the field is encoded into the structure of the table: ordinal fields partition the table into rows and columns while quantitative fields are spatially encoded as axes within the table panes. The four types of operators in this algebra, in order of precedence, are dot (.), cross (x), nest (/), and concatenate (+). Parentheses can be used to alter the precedence of the operators. Each operand can be interpreted as an ordered set and the precise semantics of each operator are defined in terms of their effects on these operand sets. The dot operator specifies the desired level of detail within a dimensional hierarchy. The cross and nest operator behave like a cross-product of two vectors (nest only produces pairs for which data exist), and the concatenate operator represents the union of two sets.

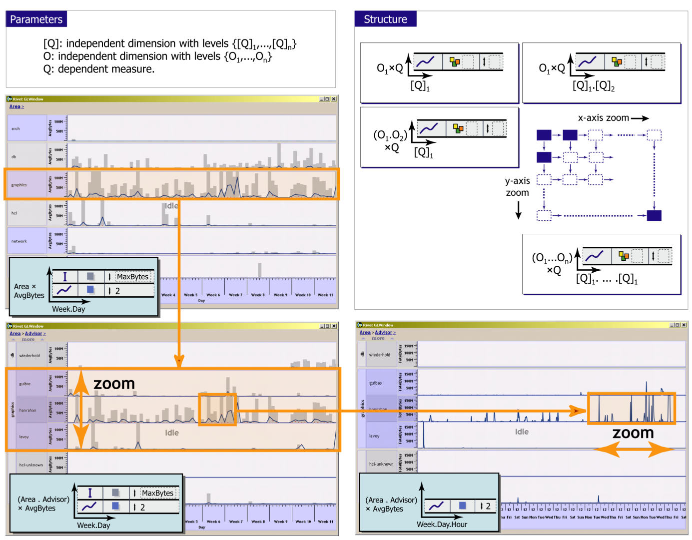

Fig. 2. Figure 2: The Zoom Graph for the Chart Stacks Pattern as well as screenshots of a visualization of a 12-week trace of an in-building mobilenetwork developed using that pattern. The top visualization shows a line chart of average bytes/hour for each day for each research area. The line charts are layered above a high-low bar encoding the maximum and minimum bytes/hour. In the next visualization, the user has zoomed in on the y-axis, breaking apart the charts to create a chart for each advisor within the research groups. In the final visualization, the user has zoomed on the x-axis, increasing the granularity of the line chart to hourly values from daily values

V. Zoom Graph

A multiscale visualization is described using a grpah to allow multiple zoom paths for any given point. An individual zoom can either change the data abstraction, the visual abstraction or both. The zooming can tied to an axis. The multiscale visualization correspond to a a zoom graph is implemented using the rivet [4].

VI. MULTISCALE DESIGN PATTERNS

In this section, the four standard zooms presented and expressed them using formal notation for zoom graphs. These zooms have traditionally been used in domain-specific applications, and also specific examples for each, the notation expresses each pattern as a general class of multiscale visualizations.

A. Chart Stacks

This pattern applies when analysts are trying to understand how a dependent measure (such as profit or number of network packets) varies with two independent hierarchical ordinal dimensions, one derived from continuous data (such as time). The hierarchy derived from continuous data is encoded in the x-axis of each chart while the other hierarchy determines the y-axis structure of the table (e.g., the order and number of rows). The y-axis for each individual chart encodes the dependent measure.

In this pattern can independently zoom along either the x- or y-axis, leading to a graph describing the multiscale visualization and can choose any path through this graph. Each zoom corresponds to changing the data abstraction: the dot operator is applied to the table algebra expression corresponding to the relevant axis. Zooming along the x-axis changes the granularity of each individual chart while zooming along the y-axis changes the number of charts. The zoom graph for this pattern is shown in Figure 2.

The example data set:-Mobile Netowrk Trace

- Time(Q): Independent hierchical dimensions (derived from continous data) describing when packets were sent. Levels: Week, day, Hour, minutes.
- User(O): Independent hierchichal dimensions describing the user of the machine that sent the packets. Levels: Area(”Research area”, e.g. graphics), Advisor, Project, Username.
- PacketCount(Q): Dependent measures indicating the packets sent.

B. Thematic Map

This pattern is applicable when visualizing geographically-varying dependent measures that can be summarized at multiple geographic levels of detail (such as county or state). Thus, the data contains an ordinal dimension hierarchy that characterizes the ge- ographic levels of detail, two independent spatial dimensions (e.g., latitude and longitude), and some number of dependent measures. Examples of this type of data are census or election data. Typically, this type of data is visualized as a map with measures encoded in the color of the area features or as glyphs layered on the map. In this pattern as the viewer zooms, more detail is displayed. There are two types of zooms in this pattern: the data abstraction can change by changing the specification and the visual abstraction changes as layers are added to display more details: both the county name and population values are displayed as text. The data set for this pattern is : Population Data

- Latitude, Longitude (Q): independent quantitive dimensions.

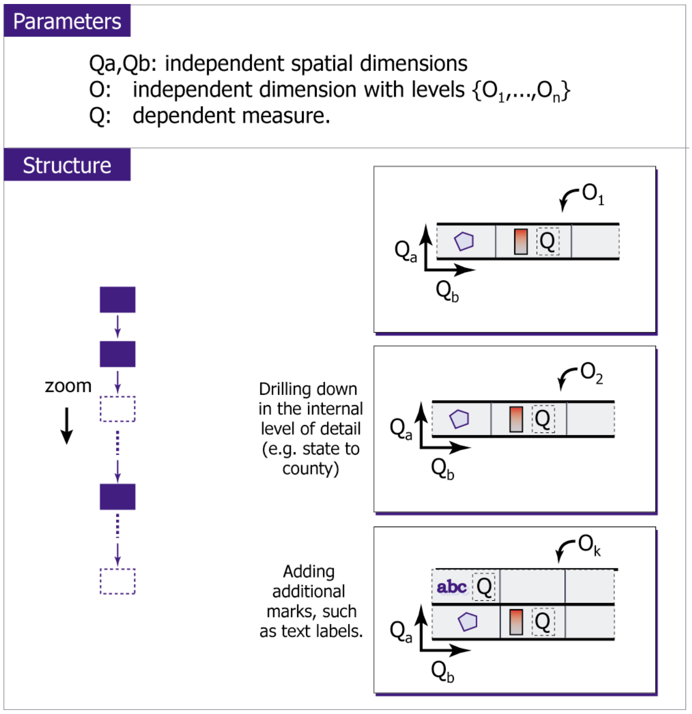

Fig. 3. Figure 3: Zoom graph for Pattern 2: Thematic Maps.

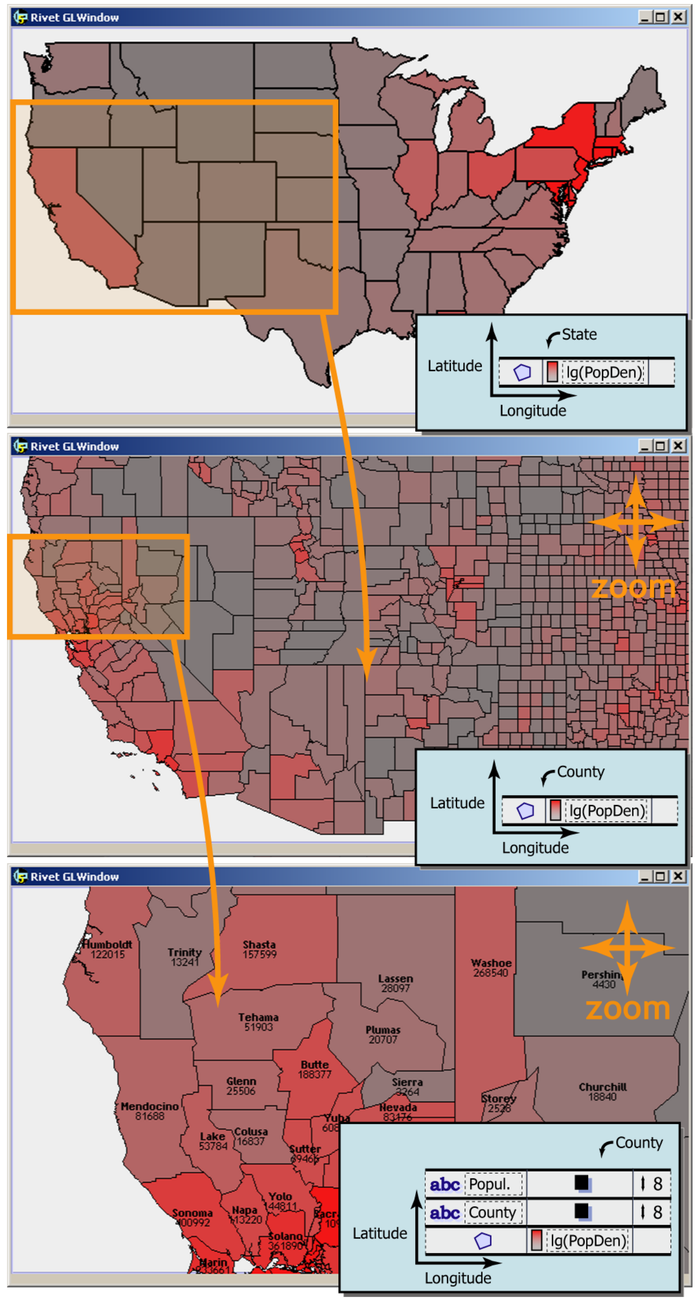

Fig. 4. Figure 4: A series of screenshots of a multiscale visualization of the population of the USA, developed using the Thematic Maps pattern. The initial view is at the state level of detail, with each state colored by population density. As the user zooms in, with the x and y dimensions lock-stepped together, the visualization changes data abstraction, drilling down to the county level of detail. As the user zooms in further, the visual abstraction changes as layers are added to display more details: both the county name and population values are displayed as text.

- PopDensity, Population (Q): Dependent measures.
- SpatialLOD(O): Independent hierarchical ordinal dimendion describing the available geographic levels of detail. Levels: State, County.

C. Dependent Quantitative-Dependent Quantita- tive Scatterplots

This pattern don't have any inherent mapping to physical world like thematic map. So data used can be any set that can be categorized according to some hierarchies. Most of the corporate data warehouse fall into this category.The data abstraction can change by either adding or removing fields in the internal level of detail protion of the specification. Visual abstraction can change by either adding retinal encoding to the current layer or by adding information in the additional layer. The data set is:- Coffe Shop Data

- Products(O): Independent hierarchical ordinal dimension describing the product sold. Levels: Product Type, Product
- Time(Q): Independent hierarchical ordinal dimension derived from continous data describing when a business transaction took place. Levels: Year, Quarter, Month.
- Location(O): Independent ordinal dimension describing the location of stores. Levels: Market(e.g."east"), state.
- Profit, Sales(Q): Dependent measures describing business activity.

multiscale visualization of this data set is shown in Figure 6.

D. Matrices

This pattern used when the analyst is exploring how a dependent measure varies with the values of two independent dimension fierachies. This type of data can be effectively visualized as a table, where the rows encode one hierarchy and the columns encode a different hierarchy and a glyph in each cell represents the measure.

Zooming is either by aggregation rows(or columns) or breaking single row(or column) down into multiple row(or columns).

The data set fits this particullarly of DNA microarray

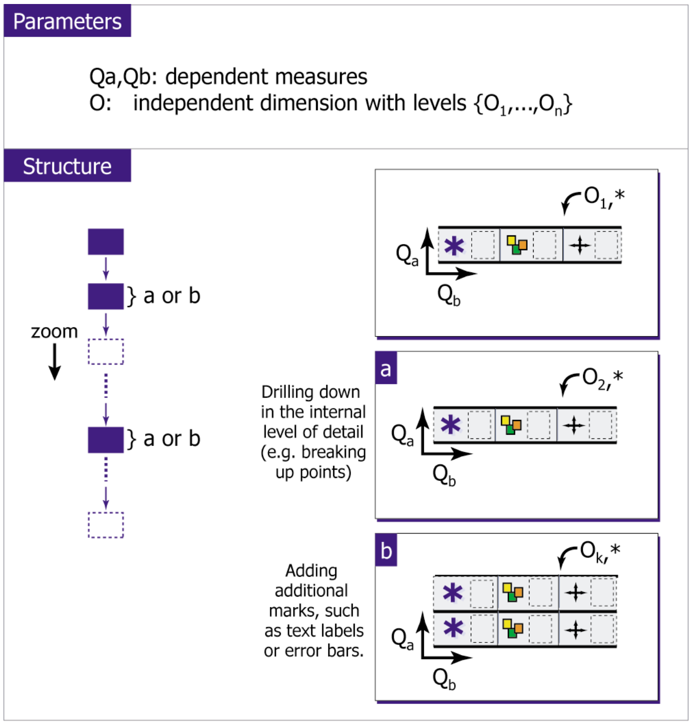

Fig. 5. Figure 5: Zoom graph for Pattern 3: Dependent Quantitative-Dependent Quantitative Scatterplots

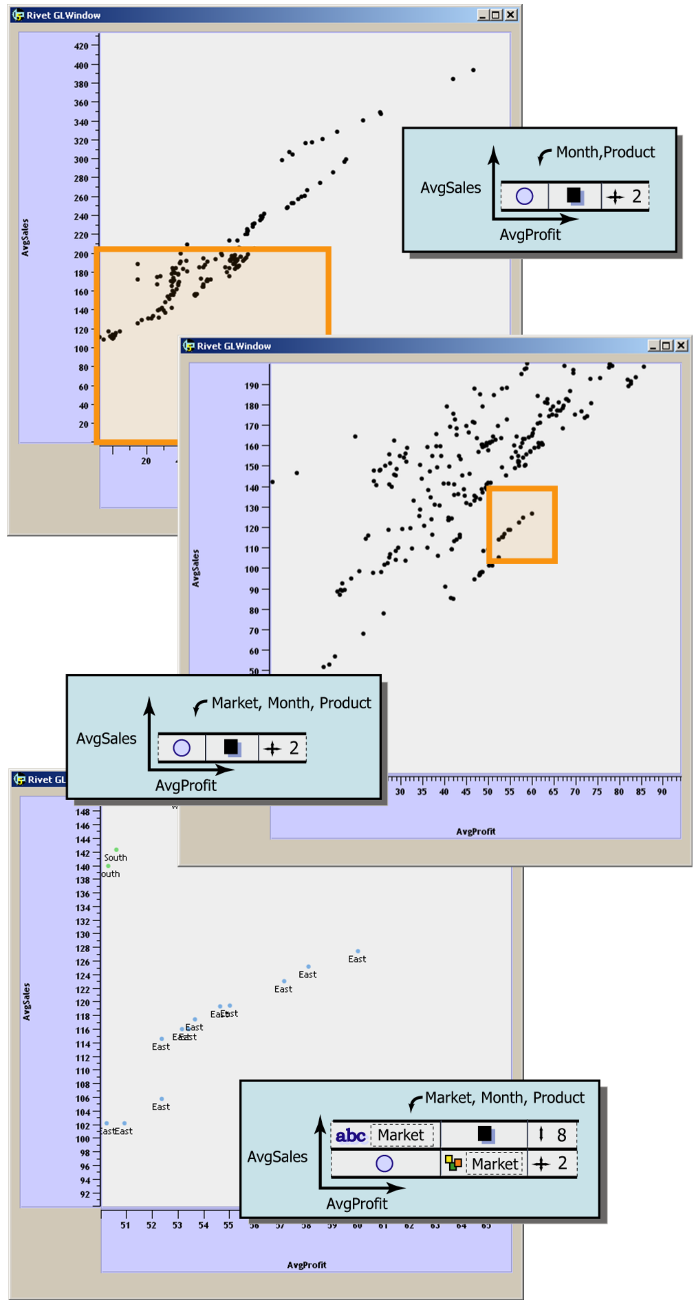

Fig. 6. Figure 6: A series of screenshots of a multiscale visualization of average sales versus average profit over a two-year period for a hypo- thetical coffeeshop chain. In the first visualization, each point rep- resents profit and sales for a particular month and product, summed over all locations. In the next visualization, the user zooms, chang- ing the data abstraction: points that were originally aggregated over all locations are now broken down by market, resulting in four points for every original point. As the user zooms in further, the visual abstraction changes as layers are added to display more de- tails: each point is colored according to market and a text label is added to redundantly encode the market name.

data. The gens can be clustered to form one hierarchy and experiments annother hierarchy.

The data set is: Yeast Microarray Data

- Gene Clusters(O): Independent hierarchical ordinal dimension describing a hierarchical clustering of the gene data created using Eisen's Clustre software. Levels: GeneRoot, Level 1-12, Genes
- Array (Experiment) Clusters(o): Independent ordinal dimension describing a hierarchical clustering of the array data created using Eisen's Cluster software. Levels: Array Root, Level 1-8 Arrays.
- Expression (Q) Dependent measure describing the amount of gene expression on a microarray.

VII. Conclusion

This paper gives the techiniques to combine the Data Abstaction and visual Abstraction techinque so the relational database stored in data cubes can be effectively visualize with multiple paths of zooming.

- Multiscale visualization with both visual and data abstraction using generalized mechanisms: Data Abstraction Data Cubes Visual Abstraction Polaris
- Zoom Graphs for specifying and implementing multiscale visualizations
- Design Patterns for implementation of multiscale visualization on some example data sets.

VIII. Future Work

- Designing new patterns.
- Transitions between levels-of-detail.
- Communicate parent-child relationships.
- Non-uniform branching.
- Animation/dissolve/fade?
- Data management Prefetching and caching of large data sets.

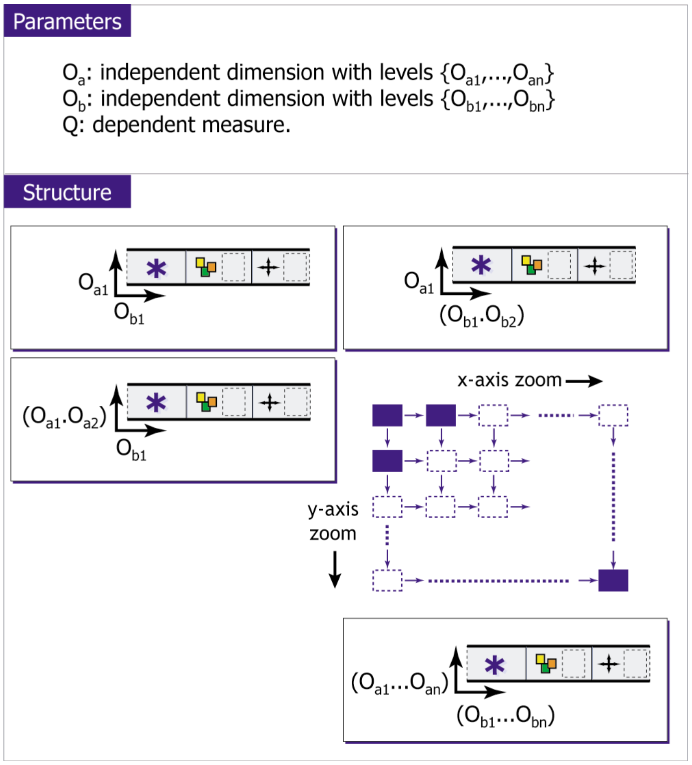

Fig. 7. Figure 7: Zoom graph for Pattern 4: Matrices

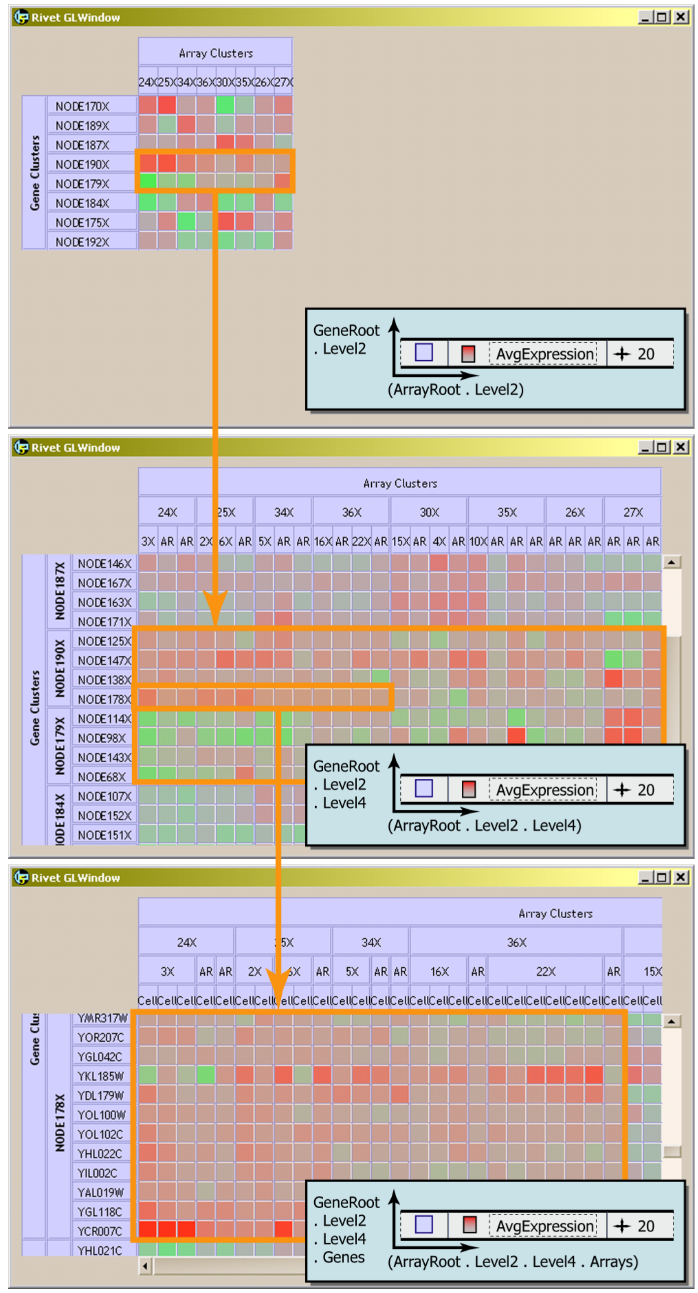

Fig. 8. Figure 8: A series of screenshots of a multiscale visualization of yeast microarray data developed using the Matrix pattern. The first visualization shows the highest level gene clusters on the y-axis, the microarray experiment clusters on the x-axis, and the average gene expression in each cell. In the next visualization, the user zooms on both axes to show more detailed information for both gene and array clusters. In the final visualization, the user has zoomed to show the original measurements for each gene in each microarray experiment.

REFERENCES

[1] J. Abello and J. Korn. MGV: A System for Visualizing Massive Multidigraphs. *In IEEE Trans. on Visualization and Computer Graphics, 8(1), January 2002*, pp. 21-38.

[2] B. Bederson, J. Hollan, K. Perlin, J. Meyer, D. Bacon, and G. Furnas. Pad++: *A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics. In J. of Visual Languages and Computing, 7, 1996*, pp. 3-31.

[3] J. Bertin. Semiology of Graphics: Diagrams, Networks, Maps. *Univ. of Wisconsin Press, 1983.*

[4] R. Bosch, C. Stolte, D. Tang, J. Gerth, M. Rosenblum, and P. Hanrahan. *Rivet: A Flexible Environment for Computer Systems Visualization. In Computer Graphics, 34(1), February 2000.*

[5] M. Eisen. *Cluster and Treeview. http://rana.lbl.gov.*