Multiscale Visualization Using Data Cubes

A 2002 Paper by

Chris Stolte, Diane Tang, and Pat Hanrahan

Presented by

Curran Kelleher

Abstract

Most analysts start with an overview of the data before gradually refining their view to be more focused and detailed. Multiscale pan-and-zoom systems are effective because they directly support this approach. However, generating abstract overviews of large data sets is difficult, and most systems take advantage of only one type of abstraction: visual abstraction. Furthermore, these existing systems limit the analyst to a single zooming path on their data and thus a single set of abstract views.

This paper presents: (1) a formalism for describing multiscale visualizations of data cubes with both data and visual abstraction, and (2) a method for independently zooming along one or more dimensions by traversing a zoom graph with nodes at different levels of detail. As an example of how to design multiscale visualizations using our system, we describe four design patterns using our formalism. These design patterns show the effectiveness of multiscale visualization of general relational databases.

Abstract

This paper presents: … *(2) a method for independently zooming along one or more dimensions by traversing a zoom graph with nodes at different levels of detail.*

Abstract

Most analysts start with an overview of the data before gradually refining their view to be more focused and detailed. Multiscale pan-and-zoom systems are effective because they directly support this approach. However, generating abstract overviews of large data sets is difficult, and most systems take advantage of only one type of abstraction: visual abstraction. Furthermore, these existing systems limit the analyst to a single zooming path on their data and thus a single set of abstract views.

This paper presents: (1) a formalism for describing multiscale visualizations of data cubes with both data and visual abstraction, and (2) a method for independently zooming along one or more dimensions by traversing a zoom graph with nodes at different levels of detail. As an example of how to design multiscale visualizations using our system, we describe four design patterns using our formalism. These design patterns show the effectiveness of multiscale visualization of general relational databases.

1 Introduction

two types of abstraction

data abstraction and *visual abstraction*.

Data Abstraction

1 Introduction

Data abstractions (e.g., aggregation or selection) change the underlying data before mapping them to visual representations.

2 Related Work

Data abstraction refers to transformations applied to the data before being visually mapped, including aggregation, filtering, sampling, or statistical summarization.

Multiscale Visualization in Cartography

Data cubes are a commonly accepted method for abstracting and summarizing relational databases.

Data Abstraction: Data Cubes

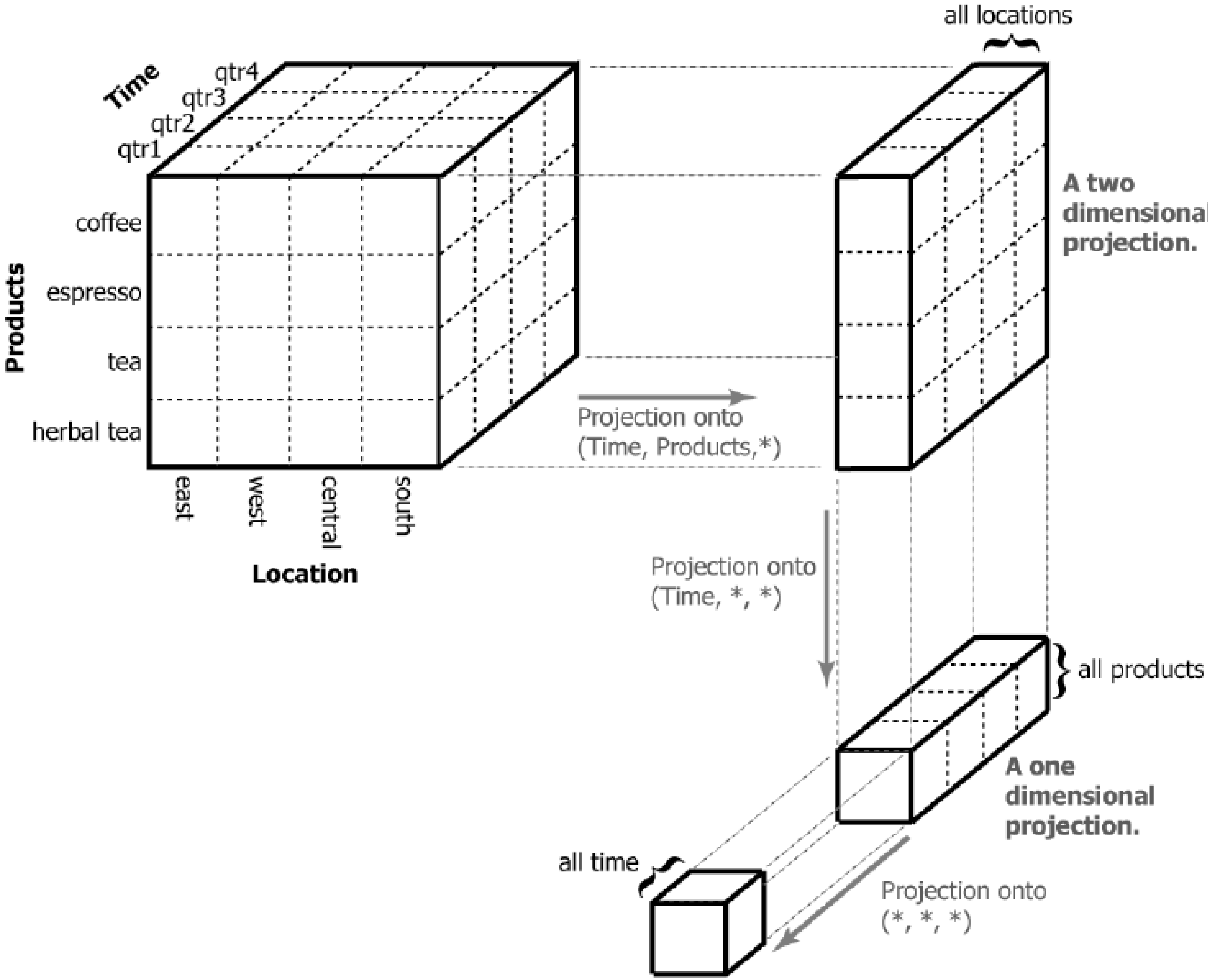
Data cubes categorize information into two classes: dimensions and measures, For example, U.S. states are a dimension, while the population of each state is a measure.

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. For example, an axis corresponding to states would have fifty values, one for each state.

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

Projecting a three dimensional data cube

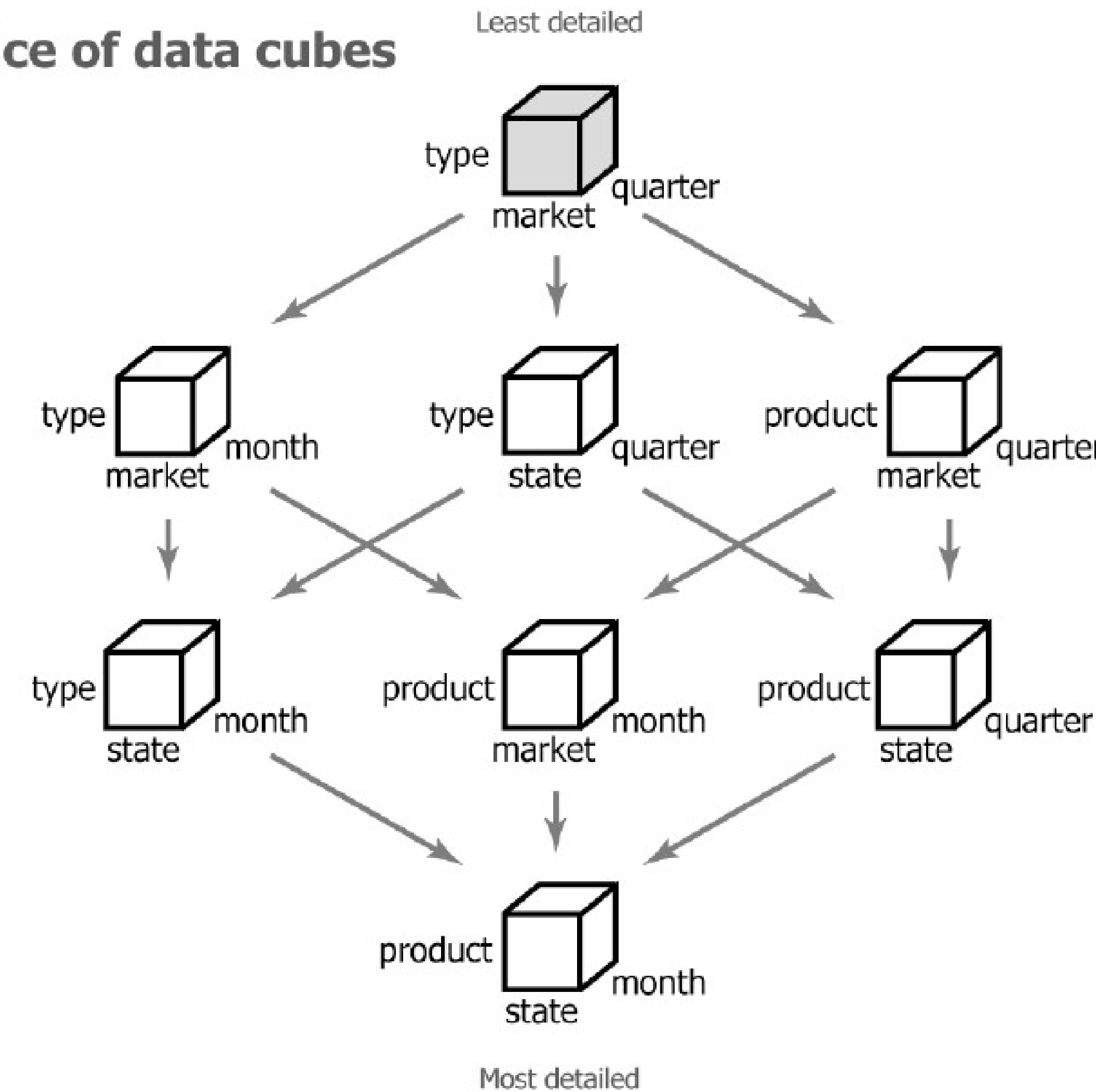


Thus far, we have considered dimensions to be flat structures. However, most dimensions have a hierarchical structure.

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

The lattice of data cubes

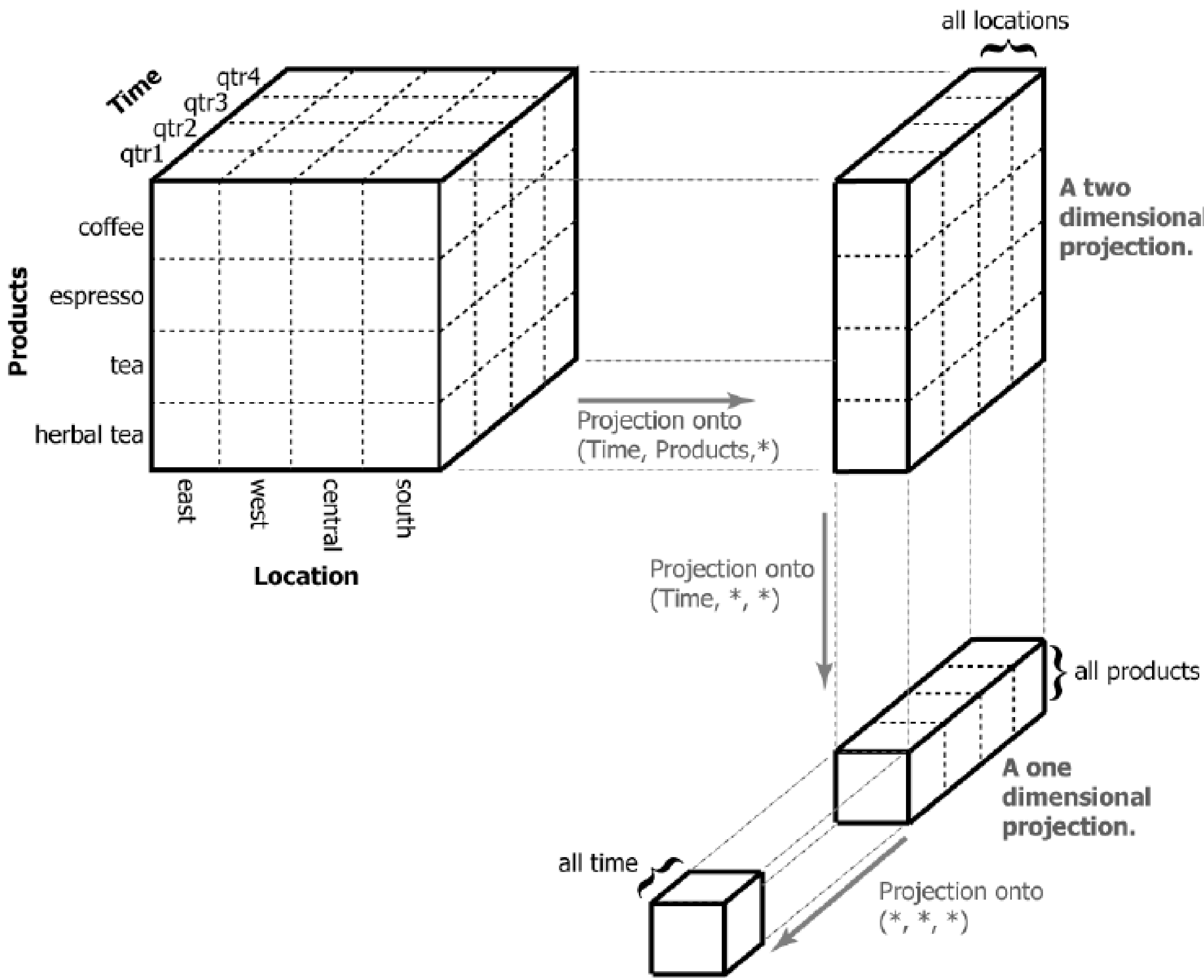


each cube is defined by the combination of a level of detail for each dimension.

Data 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:

Projecting a three dimensional data cube



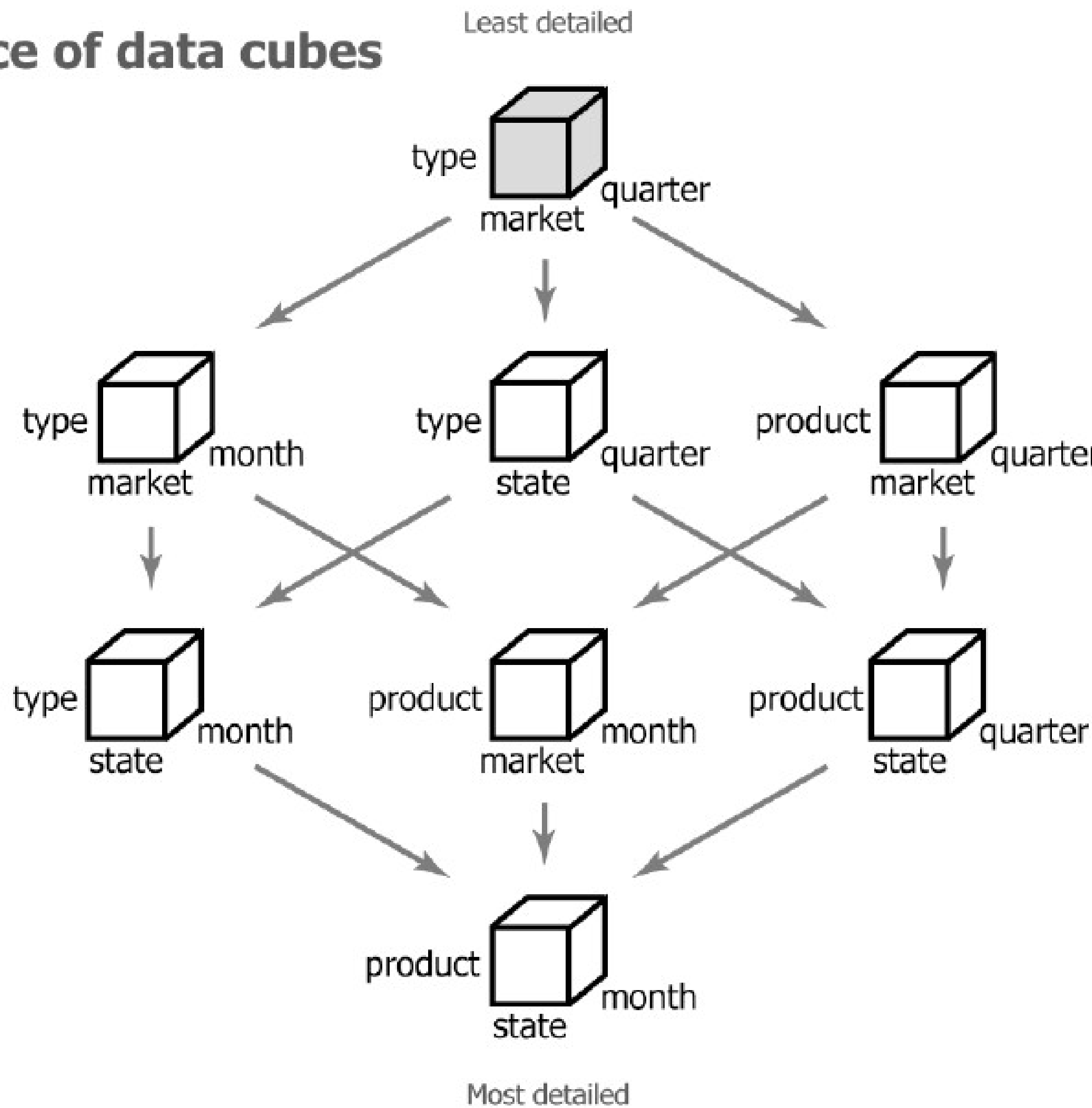
particular *projection* in this lattice of data cubes: (a) which dimensions we currently consider relevant and (b) the appropriate level of detail for each relevant dimensional hierarchy.

(b)

level of detail

identifies the cube in the lattice

The lattice of data cubes



(a)

the relevant dimensions identifies which projection

of that

cube is needed.

Visual Abstraction

1 Introduction

[illegible]

[illegible] Visual abstractions change the visual representation of data points (but not the underlying data itself) to provide more information as the user zooms [illegible]

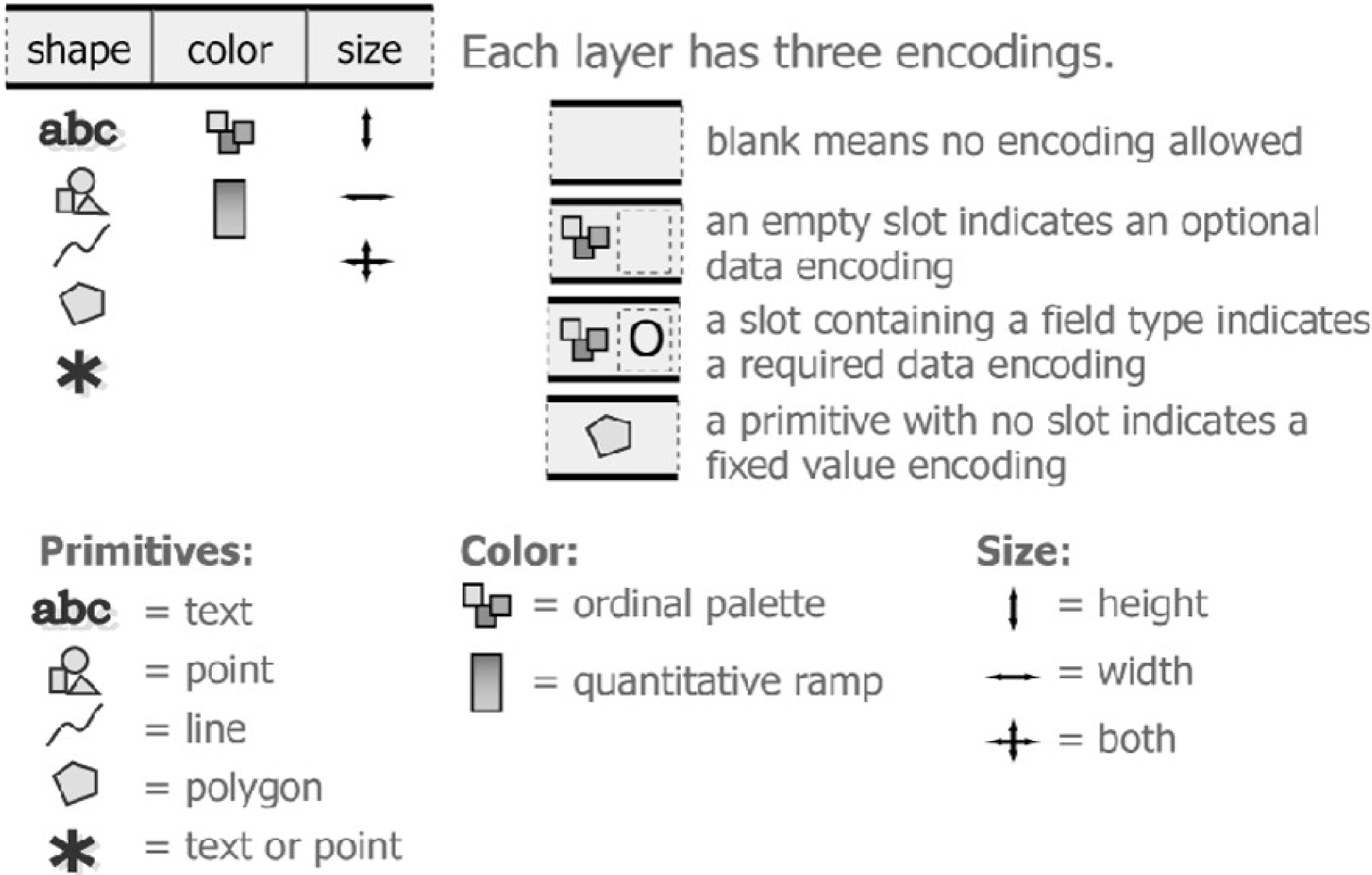
2 Related Work

Visual abstraction refers to abstractions that change the visual representation (e.g., a circle at an overview level versus a text string at a detailed level), change how data is encoded in the retinal attributes of the glyphs (e.g., encoding data in the size and color of a glyph only in detailed views), or apply transformations to the set of visual representations (e.g., combining glyphs that overlap).

3.2 Visual Abstraction: Polaris

Previously, we presented the Polaris database exploration tool

Visual Abstraction in Polaris



Polaris

database exploration tool consisting of three parts: (1) a formal specification language for describing table-based visualizations, (2) a user interface for automatically generating instances of these specifications, and (3) a method for automatically generating the necessary database queries to retrieve the data to be visualized by a specification.

Abstract

This paper presents: (1) a formalism for describing multiscale visualizations of data cubes with both data and visual abstraction,

Polaris

(1)

(2)

(3)

We later extended all three parts to support hierarchically structured data cubes

Abstract

Most analysts start with an overview of the data before gradually refining their view to be more focused and detailed. Multiscale pan- and zoom systems are effective because they directly support this approach. However, generating abstract overviews of large data sets is difficult, and most systems take advantage of only one type of abstraction: visual abstraction. Furthermore, these existing systems limit the analyst to a single zooming path on their data and thus a single set of abstract views.

This paper presents: (1) a formalism for describing multiscale visualizations of data cubes with both data and visual abstraction, and (2) a method for independently zooming along one or more dimensions by traversing a zoom graph with nodes at different levels of detail. As an example of how to design multiscale visualizations using our system, we describe four design patterns using our formalism. These design patterns show the effectiveness of multiscale visualization of general relational databases.

3 Multiscale Visualizations

In this section, we present our system for describing multiscale visualizations that support multiple zoom paths and both data and visual abstraction.

3 Multiscale Visualizations

we think of multiscale visualizations as a graph, where each node corresponds to a particular set of data and visual abstractions and each edge is a zoom.

3 Multiscale Visualizations

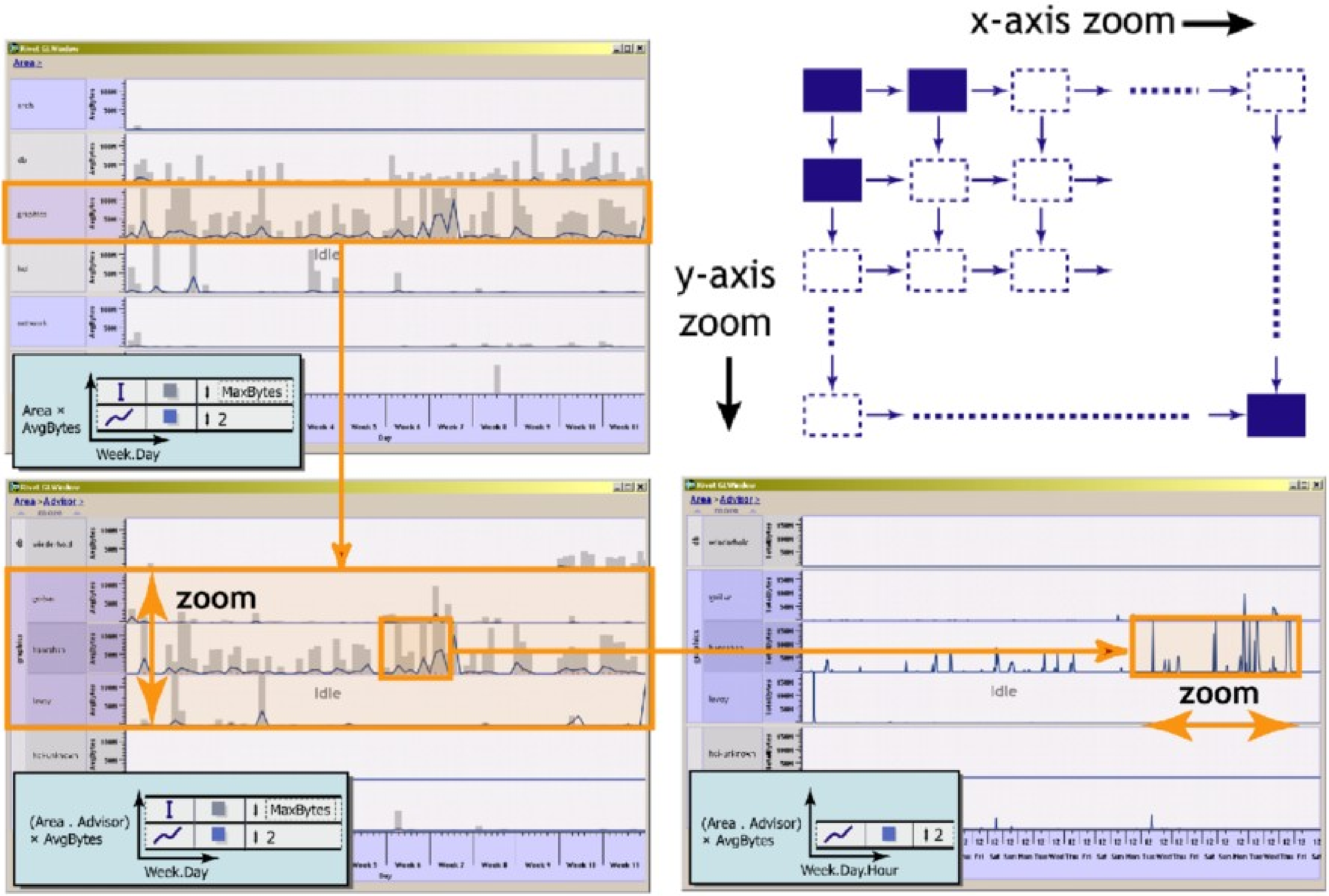
In this section, we present our system for describing multiscale visualizations that support multiple zoom paths and both data and visual abstraction. Rather than considering multiscale visualizations as simply a series of linear zooms, we think of multiscale visualizations as a graph, where each node corresponds to a particular set of data and visual abstractions and each edge is a zoom. Zooming in a multiscale visualization is equivalent to traversing this graph. Each node in this graph can be described using a Polaris specification that identifies the visual representation and abstraction and can be mapped to a unique projection of the data cube, which is a data abstraction of the underlying relational data.

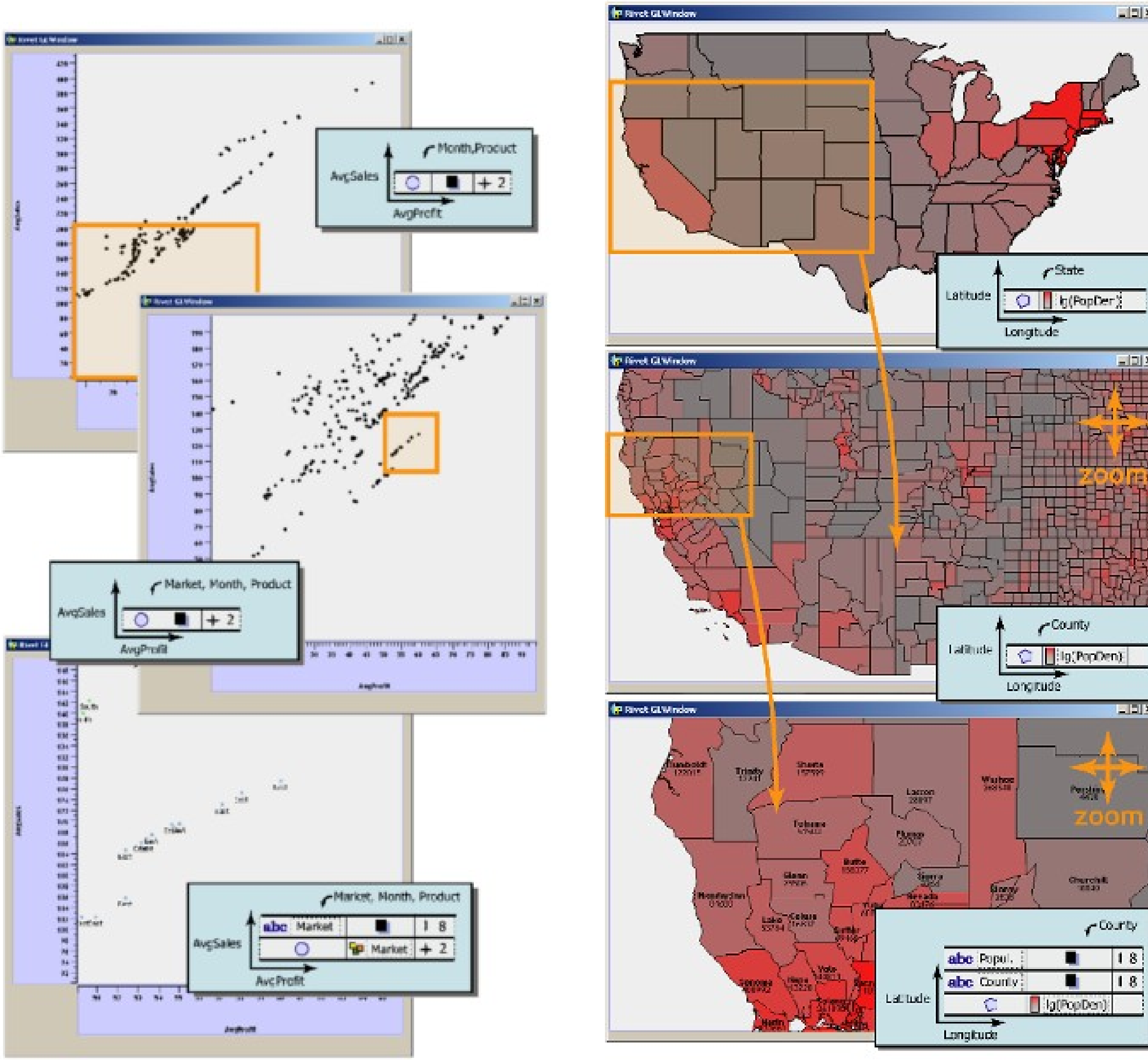
In the remainder of this section, we first review the two technologies we use to perform data abstraction (data cubes) and visual abstraction (Polaris). When reviewing Polaris, we also introduce a graphical notation for describing the key elements of a specification. Finally, we present how we can create a zoom graph of

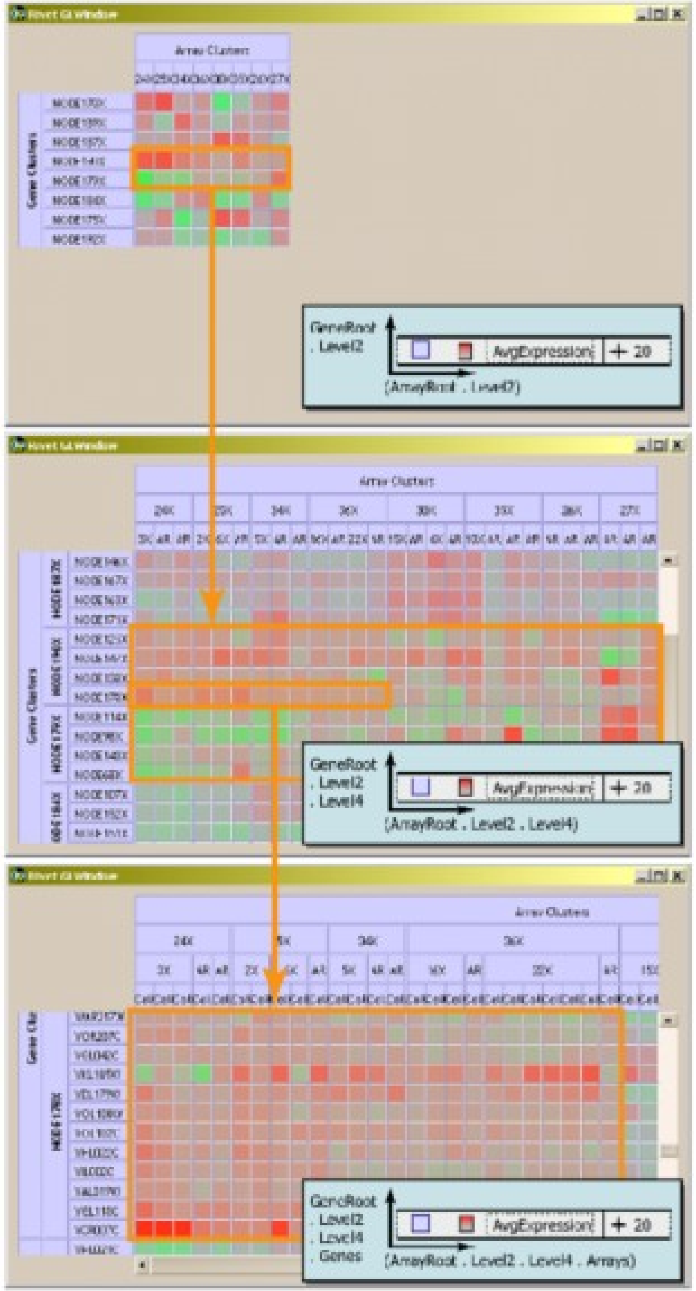
3 Multiscale Visualizations

Each node in this graph can be described using a Polaris specification that identifies the visual representation and abstraction and can be mapped to a unique projection of the data cube, which is a data abstraction of the underlying relational data.

Traversing the Zoom Graph




Month,Product
AvgSales
+ 2
AvgProfit
Market, Month, Product
AvgSales
+ 2
AvgProfit
Market, Month, Product
abc Market
I 8
AvgSales
Market
+ 2
AvgProfit
Rivet GLWindow
State
Latitude
lg(PopDen)
Longitude
Rivet GLWindow
zoom
County
Latitude
lg(PopDen)
Longitude
Rivet GLWindow
zoom
County
abc Popul.
I 8
abc County
I 8
Latitude
lg(PopDen)
Longitude



Array Clusters
Gene Clusters
GeneRoot
. Level2
AvgExpression
+ 20
(ArrayRoot . Level2)
GeneRoot
. Level2
. Level4
(ArrayRoot . Level2 . Level4)
GeneRoot
. Level2
. Level4
. Genes
(ArrayRoot . Level2 . Level4 . Arrays)

The End.