CENTRE FOR DISTANCE EDUCATION
ANNA UNIVERSITY

DMC8019

MASTER OF COMPUTER APPLICATIONS

DATA VISUALIZATION TECHNIQUES

CENTRE FOR DISTANCE AND ONLINE EDUCATION

ANNA UNIVERSITY
CHENNAI – 600 025

<u>SYLLABUS</u>

OBJECTIVES:

 Understand the categories of data quality principles.

 Describe data through visual representation.

 Provide basic knowledge about how large datasets are represented into visual graphics

 Easily understand about the complex relationships within the data.

 Design effective visualization techniques for any different problems.

UNIT I INTRODUCTION

Visualization – visualization process – role of cognition – Pseudocode conventions – Scatter plot - Data foundation: Types of data - Structure within and between records - Data preprocessing – Human perceptions and information processing.

UNIT II VISUALIZATION FOUNDATIONS

Semiology of graphical Symbols – Eight Visual Variables – Historical Perspective- Visualization Techniques for spatial data – One-dimensional data- two dimensional data – Three dimensional data- dynamic data – combining techniques- Visualization of Geospatial data – Visualization of Point, line, area data.

UNIT III DESIGNING EFFECTIVE VISUALIZATION

Steps in Designing Visualization – problems in Designing Effective Visualization – Comparing and evaluating visualization techniques – Visualization Systems.

UNIT IV INFORMATION DASHBOARD DESIGN

Characteristics of dashboards – Key goals in visual design process – Dashboard display media – Designing dashboards for usability – Meaningful organization – Maintaining consistency – Aesthetics of dashboards – Testing for usability – Case Studies: Sales dashboard, Marketing analysis dashboard.

UNIT V VISUALIZATION SYSTEMS

Systems based on Data type-systems based on Analysis type – Text analysis and visualization – Modern integrated visualization systems – toolkit - Research directions in visualization – issues of cognition, perception and reasoning –issues of evaluation - issues of Hardware.

OUTCOME:

On completion of the course the student should be able to:

1. Describe principles of visual perception
2. Apply visualization techniques for various data analysis tasks – numerical data
3. Apply visualization techniques for various data analysis tasks – Non numerical data
4. Design effective visualization techniques for different problems
5. Design information dashboard.

REFERENCES:

1. Matthew O. Ward, Georges Grinstein, Daniel Keim "Interactive Data Visualization: Foundations, Techniques, and Applications", CRC Press; 2nd edition, 2015

2. 2Stephen Few, "Now you see it: Simple Visualization Techniques for Quantitative Analysis", 1st Edition, Analytics Press, 2009.

3. Stephen Few, "Information Dashboard Design: The Effective Visual Communication of Data", 1st Edition, O'Reilly,2006.

4. Ben Fry, "Visualizing data: Exploring and explaining data with the processing environment", 1st Edition, O'Reilly, 2008.

CONTENTS

UNIT 1

1 INTRODUCTION TO DATA AND INFORMATION VISUALIZATION

1.1 Introduction

Data visualization is a crucial aspect of understanding and communicating information effectively. This unit introduces the fundamental concepts of visualization, its historical development, its relationship with other fields, and the cognitive and perceptual aspects involved in visualization.

1.2 Learning Objectives

By the end of this unit, learners should be able to:

- Define data visualization and explain its importance.
- Describe the history and evolution of visualization techniques.
- Differentiate between scientific visualization and information visualization.
- Understand the role of perception and cognition in visualization.

1.3 Overview

This unit covers the foundations of data visualization, including its significance, historical development, relationships with other fields, and perceptual and cognitive aspects. Topics include the visualization process, the human visual system, perception in visualization, and cognitive aspects that impact data representation.

1.4 What Is Visualization?

Visualization is the process of conveying information through graphical representations. Since ancient times, humans have used images to communicate ideas even before the development of written language. A single image can present a large amount of information in a fraction of the time required to read text. This is because the human brain processes images in parallel, whereas reading is a sequential process. Furthermore, visualizations transcend language barriers, allowing different people to interpret the same information regardless of their linguistic background.

1.4.1 Visualization in Everyday Life

Visual representations of data are ubiquitous in daily life. Some common examples include:

- A formatted number indicating a country's Gross National Product (GNP).
- Newspaper tables presenting statistical information.
- Train and subway maps displaying routes and schedules.
- Regional maps helping travelers navigate new locations.

- Weather charts tracking storm movements and forecasting weather conditions.
- Stock market graphs indicating trends and economic performance.
- Medical imaging, such as 3D reconstructions of injuries from CT scans.
- Instruction manuals using step-by-step diagrams for assembling products.
- Road signs conveying traffic rules and navigation instructions.

In professional fields, visualization is also widely employed:

- Financial analysis and stock market predictions.
- Engineering designs and structural analysis.
- Medical imaging, such as MRI scans for disease diagnosis.
- Scientific simulations of astronomical or biological phenomena.
- Population behavior studies, such as smoking trends.
- Marketing and advertising through graphical campaign materials.

Visualization serves as a supplement or an alternative to textual and verbal information, offering a richer and more intuitive understanding of data.

1.4.2 Why Is Visualization Important?

Visualization plays a crucial role in data interpretation and decision-making. Humans are highly visual creatures, relying on sight to comprehend and analyze information. There are two key aspects that highlight the importance of visualization: data distortion and human interpretation.

The way data is presented can significantly influence decision-making. Consider Figure 1.1, which illustrates the same dataset using different scaling methods:

(a) Uniform scaling of both x and y axes results in a clustered data visualization.
(b) Expanding the y-axis creates a horizontal pattern.
(c) Expanding the x-axis generates a vertical pattern.
(d) Adjusting the scale based on the data range provides a more accurate representation.

These variations demonstrate how scaling choices can alter perceptions of data structure. Misrepresentation can lead to incorrect conclusions, emphasizing the need for careful visualization design.

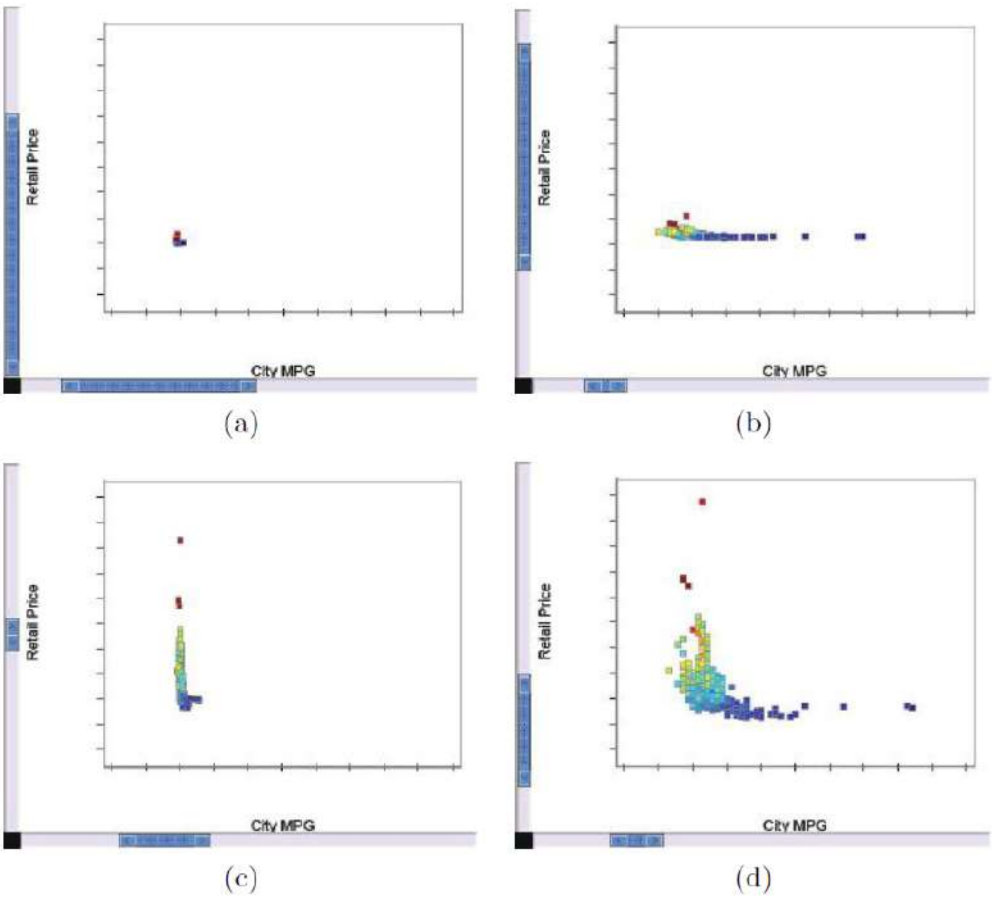


Figure 1.1 The appearance of data varies with scale: (a) uniform x and y, (b) enlarged y, (c) enlarged x, (d) scale based on data range.

A real-world study in 1999 by Linda Elting and colleagues explored how different visualization methods affected clinical decision-making. They presented 34 clinicians with preliminary results from a hypothetical clinical trial using four visualization formats as given in figure 1.2:

- A simple table.
- Pie charts.
- Stacked bar charts.
- An icon-based representation where each rectangle represented a patient.

The icon-based display led to correct decisions 82% of the time, compared to 68% for tables and only 56% for pie charts or bar graphs. This experiment highlighted how visualization design directly impacts decision accuracy. Furthermore, it revealed that personal biases and preferences affect how people interpret visual data.

Figure 1.2 Different visualizations of a clinical trial showed that the icon display (lower right) was most effective for deciding to stop the trial, while bar and pie charts were the least effective.

The Growing Importance of Visualization

With the increasing availability of data and technological advancements, visualization has become a vital tool in various domains. The ability to analyze and present information effectively is essential for businesses, researchers, and policymakers. Key areas where visualization plays a crucial role include:

- Exploring large datasets in finance, security, healthcare, and biology.
- Enhancing communication of research findings and marketing insights.
- Assisting in decision-making for environmental, economic, and social issues.

Given the expanding volume of data, there is a growing need for effective visualization techniques that make complex information accessible and comprehensible. In the following chapters, we will explore various visualization methods and their applications, helping to bridge the gap between raw data and meaningful insights.

History of Visualization

Understanding the history of visualization helps us appreciate its evolution and significance. This section provides a brief overview, offering key insights into early visualization techniques and their impact on modern practices.

1.5.1 Early Visualizations

Primitive Visual Records

One of the earliest methods of graphically recording information dates back to prehistoric times. The Chauvet-Pont-d'Arc Cave, located in southern France, contains over 250 paintings that were created around 30,000 years ago. These paintings likely served as a means to convey information to future generations.

Early Writing Systems and Logograms

Ancient writing systems initially used pictorial representations to encode symbols and words. These early scripts, known as logograms, played a crucial role in communication. The Kish limestone tablet is regarded as the oldest written document, originating from Mesopotamia. It exhibits a combination of pictographic symbols and early cuneiform script, marking the transition from pure imagery to written language. This tablet is preserved at the Ashmolean Museum in Oxford.

Another significant writing system was Egyptian hieroglyphics, categorized into three types: logograms (representing whole words), phonograms (depicting sounds), and determinatives (used to clarify meanings in sequences of glyphs).

The Role of Maps in Early Visualization

Maps have long been an essential visualization tool for navigation, trade, and strategic planning. One of the earliest known road maps is the Peutinger Map, which outlines 70,000 miles of Roman highways with distances and key landmarks such as rivers and forests. This map, employs a distorted scale where east-west distances are exaggerated relative to north-south distances. This distortion allowed for greater practicality in its original papyrus roll format.

During medieval times, maps provided insights into both geographical and conceptual knowledge. The Hereford Map from the 13th century, for instance, represents the world with Jerusalem at its center and includes both real and mythical elements, reflecting the era's blend of fact and belief.

Early Statistical and Scientific Visualizations

One of the most famous early visualizations of statistical data was John Snow's 1854 cholera map. This map used bars to represent cholera deaths at various locations. Snow's analysis revealed a clustering of deaths around the Broad Street water pump, leading him to hypothesize that contaminated water was the source of the outbreak. His intervention—removing the pump handle—helped contain the epidemic, demonstrating the power of data visualization in public health (see Figure 1.3).

Other early visualizations included time series charts. Abu Rayhan Muhammad al-Biruni, an Islamic scholar from the 11th century, created one of the earliest depictions of planetary motion and the phases of the moon. These time-series diagrams predated the Cartesian coordinate system.

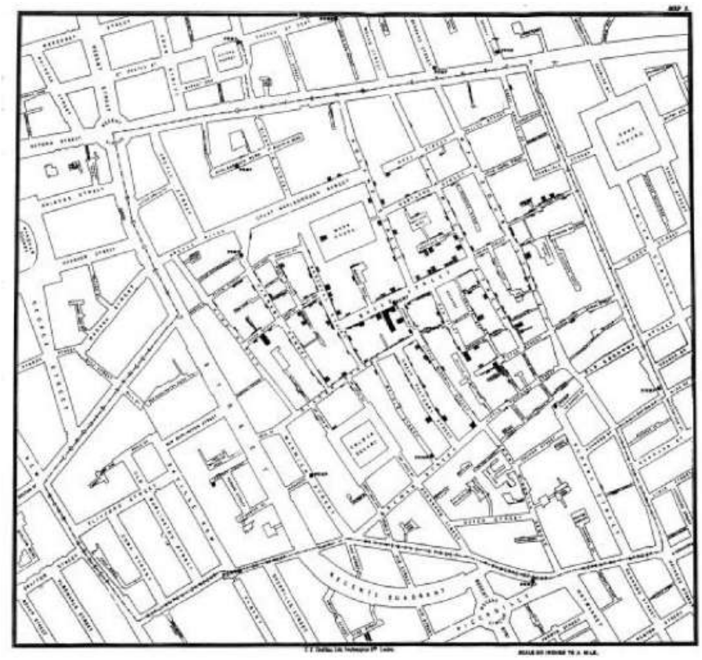


Figure 1.3 A map showing Cholera-related deaths in London (1854) highlights their concentration around the Broad Street Water Pump, with some notable outliers

Minard's Pioneering Work

Charles Minard's 1869 visualization of Napoleon's Russian campaign is an exemplary early infographic. It integrates multiple data dimensions: army size, location, movement, and temperature. The width of the flow represents troop numbers, illustrating their dramatic reduction from over 400,000 to just 10,000.

Pioneers of Information Visualization

The development of coordinate-based graphs significantly advanced data visualization. William Playfair, a Scottish engineer, pioneered statistical graphics, including time-series plots and trade balance graphs. Joseph Priestley's timeline of famous individuals' and Florence Nightingale's coxcomb chart on mortality rates further exemplify early applications of visualization.

Leonardo da Vinci contributed significantly to medical visualization with his anatomical drawings, which remain influential in medical education today.

1.5.2 Visualization Today

Modern visualizations combine qualitative and quantitative elements to facilitate information interpretation. For instance, the Tokyo Underground map (see Figure 1.4) uses logical distortions to highlight key relationships rather than geographical accuracy.

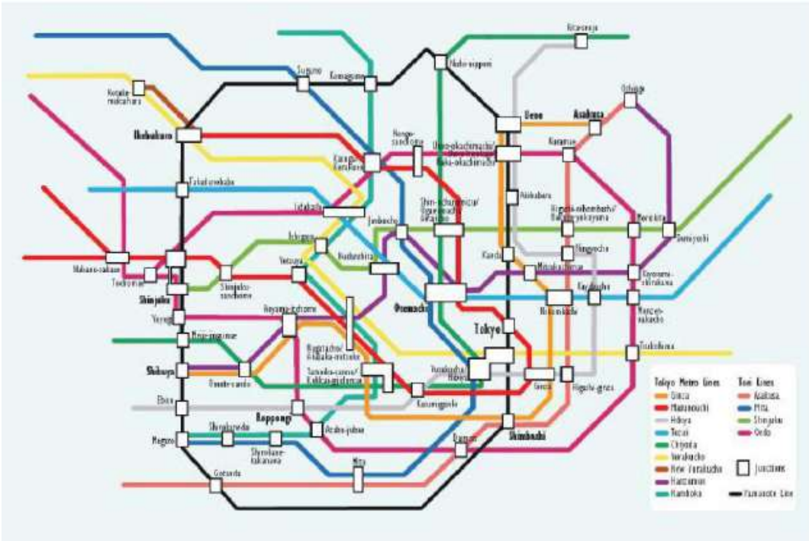


Figure 1.4 Tokyo underground metro map

Digital Mapping and Navigation

Today's digital maps, such as Google Maps offer real-time navigation, turn-by-turn directions, and interactive features. These maps balance clarity and accuracy to enhance usability.

Financial and Economic Visualization

Financial charts, such as stock market trends, provide insights beyond numerical statements. Similarly, real-time economic indicators like the U.S. National Debt Clock offer up-to-date numerical representations of fiscal conditions.

Medical and Scientific Visualization

Advances in medical visualization have revolutionized diagnostics. Electrocardiograms (ECGs) provide graphical depictions of heart activity aiding in the identification of heart conditions. Similarly, scientific scatterplots, such as yeast mechanism analysis facilitate pattern recognition and statistical modeling.

Modern visualization also supports biomedical research. Blood vessel mapping, enhances anatomical studies, while aerospace simulations optimize engineering designs. Bioinformatics tools, such as gene expression networks utilize color-coded pathways to represent complex biological interactions.

1.6 Relationship Between Visualization and Other Fields

1.6.1 Understanding the Difference Between Visualization and Computer Graphics

Visualization and computer graphics are closely related, but they serve different purposes. Initially, visualization was seen as a subfield of computer graphics because it relies on graphical techniques to represent information visually. However, while both fields use graphical primitives such as points, lines, areas, and volumes, their fundamental goals differ.

Key Differences

- Computer Graphics primarily focuses on creating and managing graphical objects and their visual properties.
- Visualization goes beyond graphics by incorporating data-driven representation, where spatial positions, population distributions, and physical measurements play a crucial role.

While visualization depends on computer graphics for rendering visual representations, it extends into other domains, including human-computer interaction, perceptual psychology, databases, statistics, and data mining. These disciplines contribute to how information is perceived, interacted with, and interpreted by users.

Role of Computer Graphics in Visualization

Computer graphics provide the foundational tools needed for visualization. This includes:

- Graphics Programming Languages: OpenGL, DirectX, Processing, Java3D
- Graphics Hardware: Nvidia and AMD/ATI graphics cards
- Rendering Techniques: Flat shading, Gouraud shading, Phong shading, ray tracing, radiosity
- Output Formats: JPEG, TIFF, AVI, MPEG

Thus, computer graphics serve as the backbone of visualization, ensuring effective and efficient rendering of data representations.

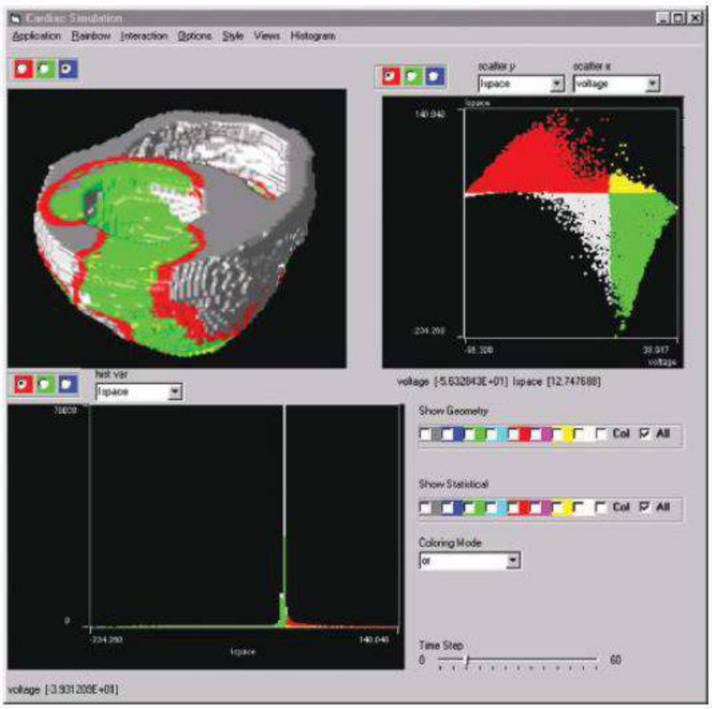


Figure 1.5 A visualization of a patient's heart, supplemented with additional visual representations of parameters that are not easily depicted on the 3D model.

1.6.2 Scientific Data Visualization vs. Information Visualization

The field of visualization is often categorized into two main types: scientific visualization and information visualization. While both involve data representation, they differ in their data sources and application areas.

Scientific Data Visualization

Scientific visualization deals with data that has inherent spatial properties, often derived from scientific experiments and simulations. Examples include:

- Medical imaging (e.g., visualizing a patient's heart in 3D)
- Molecular modeling (e.g., representing molecules with spheres and rods)
- Fluid dynamics (e.g., airflow patterns around an aircraft)

(*See Figure 1.5 for an example of a patient's heart visualization.*)

Information Visualization

Information visualization, on the other hand, is used for abstract data that may not have an inherent spatial structure. Examples include:

- Social networks (*Refer to Figure 1.6 for social network representations.*)
- Gene relationships in DNA (*See Figure 1.7 for a visualization of gene relationships.*)
- Statistical data (e.g., scatterplots of patient health records)

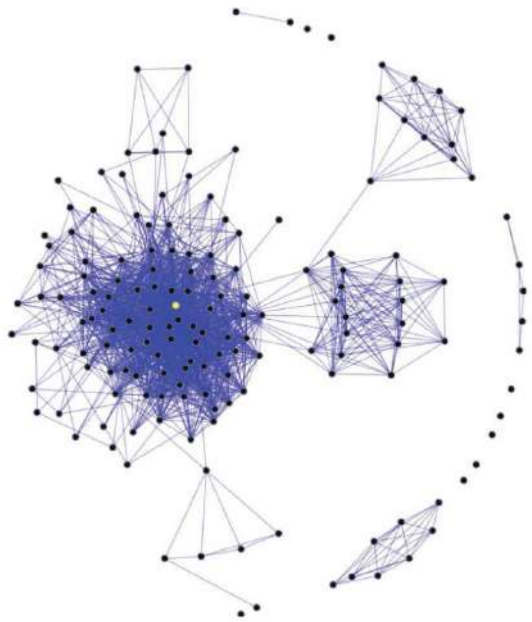


Figure 1.6 A simple social network

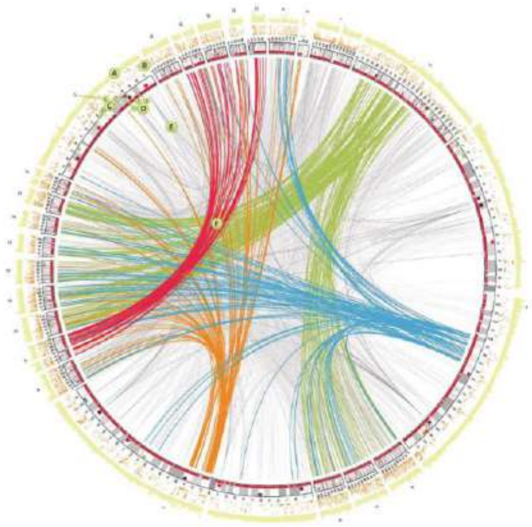


Figure 1.7 A network visualization of gene relationships in human DNA, created using the Circos package.

A Unified Approach

Historically, scientific and information visualization were treated as distinct fields. However, modern visualization approaches integrate both, recognizing that different types of data require different visualization techniques. For example, biomolecular chemistry has evolved from simple stick-and-ball models to more complex and realistic representations that incorporate both scientific and information visualization techniques.

In summary, visualization is an interdisciplinary field that extends beyond computer graphics, incorporating elements from multiple domains to effectively communicate information through visual means.

1.7 The Visualization Process

1.7.1 Understanding the Visualization Process

The visualization process involves multiple steps, beginning with analyzing the available data and determining the type of insights that need to be extracted or conveyed. The data can come from various sources and may range from simple to complex structures. The goal of visualization may include:

- Exploration – Identifying interesting patterns or anomalies.
- Hypothesis Confirmation – Validating existing theories or insights.
- Presentation – Communicating findings to an audience.

Key insights from visualizations include identifying:

- Anomalies – Data points that deviate from expected patterns.
- Clusters – Groups of similar data that indicate a pattern or phenomenon.
- Trends – Changes in data over time, which can support predictive modeling.

1.7.2 Mapping Data to Visualization

To visualize data effectively, a mapping process is required. This involves defining how raw data will be represented visually using graphical elements like:

- Points, lines, and shapes – Representing different data attributes.
- Size, position, and orientation – Depicting relationships and distributions.
- Color and texture – Highlighting variations and patterns.

For example, a list of numbers can be visualized as:

- A scatter plot, where numbers are mapped to the y-axis, and their index to the x-axis.
- A bar chart, where numbers control the height of bars.
- A heatmap, where numbers influence the color intensity of squares.

1.7.3 Interactive Visualization

Modern visualization is highly interactive, allowing users to:

- Refine views dynamically – Modify the visualization to extract meaningful insights.
- Manipulate variables – Adjust mappings, colors, and perspectives.
- Customize the display – Tailor visualizations for different users and contexts.

The effectiveness of a visualization depends on various factors, including:

- User preferences – Different users perceive visualizations differently.
- Task requirements – The purpose of visualization impacts how it is designed.
- Data characteristics – Changing datasets may necessitate different visualization approaches.

1.7.4 The Role of Visualization in Data Analysis

Visualization is often part of larger processes such as:

- Exploratory Data Analysis (EDA) – Discovering insights within raw data.
- Knowledge Discovery – Identifying patterns and relationships.
- Visual Analytics – Combining visualization with computational models for deeper analysis.

The process of generating visualizations follows a structured pipeline, which varies depending on the purpose.

1.7.5 The Visualization Pipelines

The Computer Graphics Pipeline

Computer graphics follow a sequence of steps to render visual representations (Figure 1.8). These steps include:

- Modeling – Creating 3D models with defined surfaces and vertices.
- Viewing – Positioning a virtual camera to determine perspective.
- Clipping – Removing unseen or partially visible elements.
- Hidden Surface Removal – Eliminating obscured polygons.
- Projection – Converting 3D objects into 2D representations.
- Rendering – Applying colors, textures, and lighting effects.

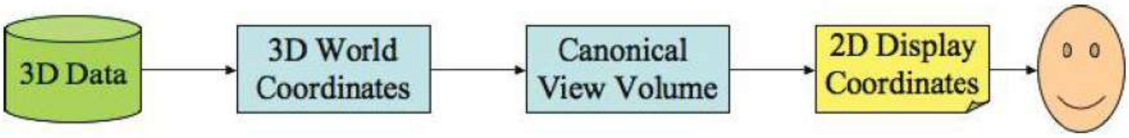


Figure 1.8 The graphics pipeline

A variation of this process, ray tracing, involves casting rays from the camera to simulate realistic lighting and reflections.

The Data Visualization Pipeline

The data visualization pipeline follows a structured approach to transform data into meaningful visual representations (Figure 1.9):

- Data Modeling – Structuring data for easy access and modification.
- Data Selection – Identifying relevant subsets of data for visualization.
- Data-to-Visual Mapping – Converting numerical or categorical data into graphical elements.
- Scene Parameter Settings – Defining colors, lighting, and other attributes.
- Rendering – Generating the final visualization, including labels and annotations.

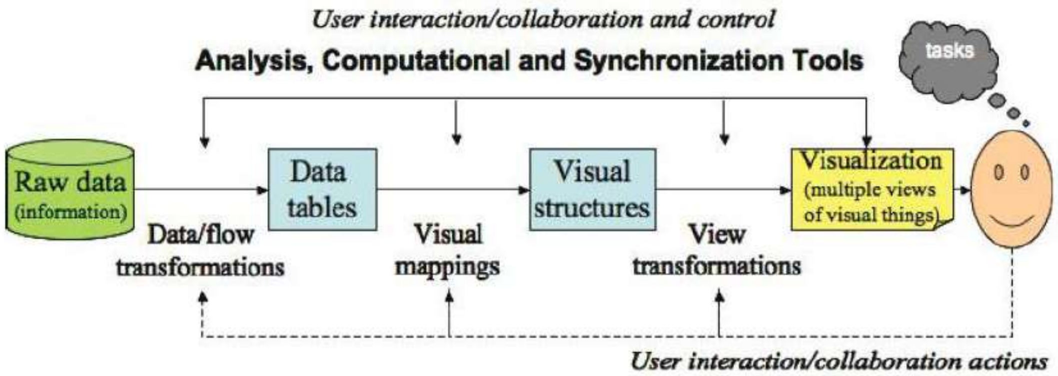


Figure 1.9 An example of the visualization pipeline, which, despite its variations, consistently converts data into an internal computer representation before rendering it visually on the screen.

The Knowledge Discovery Pipeline

The knowledge discovery process (data mining) follows a pipeline similar to visualization but focuses on generating models (Figure 1.10):

- Data Collection – Acquiring structured and unstructured data.
- Data Cleaning & Integration – Filtering and aggregating data.
- Data Mining – Applying algorithms to extract insights.
- Pattern Evaluation – Assessing the validity and reliability of discovered patterns.
- Visualization & Presentation – Communicating findings effectively.

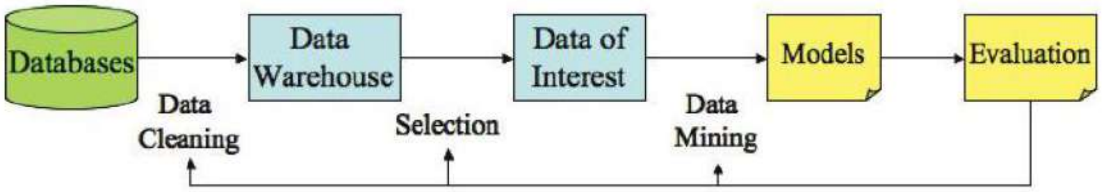


Figure 1.10 Knowledge discovery pipeline

1.7.6 The Role of Perception in Visualization

Human Perception and Visualization

Effective visualization must consider human perceptual capabilities. If a visualization is poorly designed, it may lead to misinterpretation, as seen in optical illusions like Figure 1.11 (Shepard's multi-legged elephant illusion).

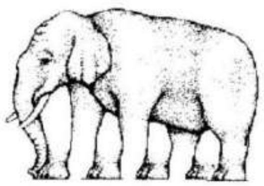


Figure 1.11 How many legs are there for this elephant?

Key perceptual concepts include:

- Color Perception – Some users, particularly those with color blindness, may struggle with certain color schemes.
- Motion Perception – Motion can enhance focus but may also distract if used excessively.
- Preattentive Processing – The ability to quickly identify visual elements such as:
    - Color differences
    - Line orientation
    - Complex patterns

Gestalt Principles in Visualization

The Gestalt School of Psychology outlines rules that influence how we perceive visual patterns:

- Proximity – Elements close together appear related.
- Similarity – Similar shapes and colors suggest relationships.
- Continuity – Aligned elements create a sense of flow.
- Closure – Incomplete shapes are mentally completed.
- Figure-Ground Perception – Distinguishing foreground from background.

Understanding these principles helps in designing intuitive visualizations.

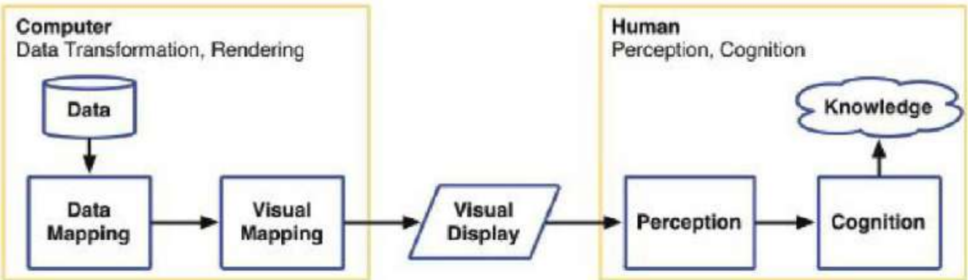


Figure 1.12 A proposed model for visualization pipeline

1.8 The Role of Cognition in Visualization

The visualization process is designed to synthesize visual representations of data, focusing on the user and their interaction with the visualization. Several key questions arise when considering cognition in this context:

- What does the user perceive in a visualization?
- What information is easily understood, missed, or retained?
- How long can the user remember the displayed information?

To address these questions, it is essential to consider cognitive processes beyond basic perception. Figure 1.12 presents a more balanced visualization pipeline, further explored later.

1.9 Pseudocode Conventions

Throughout this text, pseudocode is provided to illustrate key algorithms without unnecessary details related to user interaction, graphics rendering, or data management. The following global variables and functions are assumed in the pseudocode environment:

Global Variables

- data: Represents the working data table, containing only numeric values.
- m: Denotes the number of dimensions (columns) in the data table, indexed using j.
- n: Represents the number of records (rows) in the data table, indexed using i.

Functions

- Normalize(record, dimension, min, max): Maps values within a specified range (default: 0 to 1). Different normalization techniques (e.g., logarithmic, square root) may be applied based on context.
- Color(color): Sets the color state of the graphics environment.
- MapColor(record, dimension): Assigns a color based on a normalized value and global color map.
- Circle(x, y, radius): Draws a circle centered at (x, y) with the specified radius.
- Polyline(xs, ys): Renders a polyline using given x- and y-coordinates.
- Polygon(xs, ys): Fills a polygon with the current color state.

Geographic Visualization Functions

- GetLatitudes(record), GetLongitudes(record): Retrieve latitude and longitude arrays for geographic data.
- ProjectLatitudes(lats, scale), ProjectLongitudes(longs, scale): Convert latitude and longitude values into visualization coordinates.

Graph and 3D Surface Data Functions

- GetConnections(record): Returns an array of connected record indices.

Arrays are indexed starting from zero.

1.10 Understanding Scatterplots

A scatterplot is one of the fundamental visualization techniques, offering an intuitive representation of relationships between data variables. It follows the Cartesian coordinate system and is widely used for analyzing data trends and patterns.

Scatterplot Implementation

The following pseudocode outlines how to generate a scatterplot where each data record is represented by a circle with varying location, color, and size:

```
Scatterplot(xDim, yDim, cDim, rDim, rMin, rMax)
for each record i:
    x ← Normalize(i, xDim)
    y ← Normalize(i, yDim)
    r ← Normalize(i, rDim, rMin, rMax)
    MapColor(i, cDim)
    Circle(x, y, r)
```

Example Data Set

A dataset of 428 vehicles from 2004 provides an example of scatterplot applications. It includes attributes such as price, weight, horsepower, fuel efficiency, and size. A partial dataset is shown in Table 1.1, which categorizes vehicles by attributes like sedan, sports car, or SUV.

Table 0.1 A partial table of car and truck data, which can be viewed either as row-based (vehicles) or column-based (vehicle attributes).

Vehicle Name	Sedan	Sports	SUV	Wagon	Minivan	Pickup	AWD	RWD	Price
Acura 3.5 RL 4dr	1	0	0	0	0	0	0	0	43755
Acura MDX	0	0	1	0	0	0	1	0	36945
Suzuki XL-7 EX	0	0	1	0	0	0	0	0	23699

Variables in the Dataset:

a. vehicle name (text 1–45 characters);
b. small, sporty, compact or large sedan (1=yes, 0=no);
c. sports car? (1=yes, 0=no);
d. sport utility vehicle? (1=yes, 0=no);
e. wagon? (1=yes, 0=no);
f. minivan? (1=yes, 0=no);
g. pickup? (1=yes, 0=no);
h. all-wheel drive? (1=yes, 0=no);
i. rear-wheel drive? (1=yes, 0=no);
j. suggested retail price, what the manufacturer thinks the vehicle is worth, including adequate profit for the automaker and the dealer (U.S. dollars);
k. dealer cost (or “invoice price”), what the dealership pays the manufacturer (U.S. dollars);
l. engine size (liters);
m. number of cylinders (=-1 if rotary engine);
n. horsepower;
o. city miles per gallon;
p. highway miles per gallon;
q. weight (pounds);
r. wheel base (inches);
s. length (inches);
t. width (inches).

Exploring the Data with Scatterplots

While tabular data helps with initial analysis, visualizing relationships is much more effective. Consider a scatterplot of horsepower versus city MPG for Toyota vehicles (Figure 1.13). The vehicle category is color-coded, revealing a nearly linear relationship between horsepower and fuel efficiency.

This pattern is further tested using data from Kia (Figure 1.14) and Lexus (Figure 1.15), demonstrating a trend: increased horsepower correlates with lower MPG. A broader dataset visualization (Figure 1.16) allows us to explore general trends and anomalies.

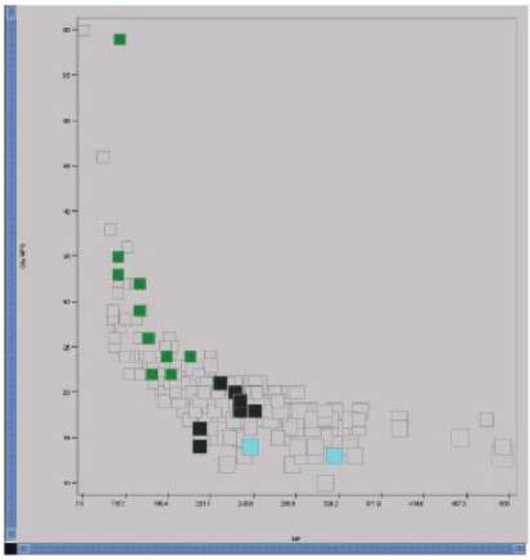


Figure 1.13 A scatterplot displaying horsepower against city MPG for Toyota vehicles, with vehicle class represented by color.

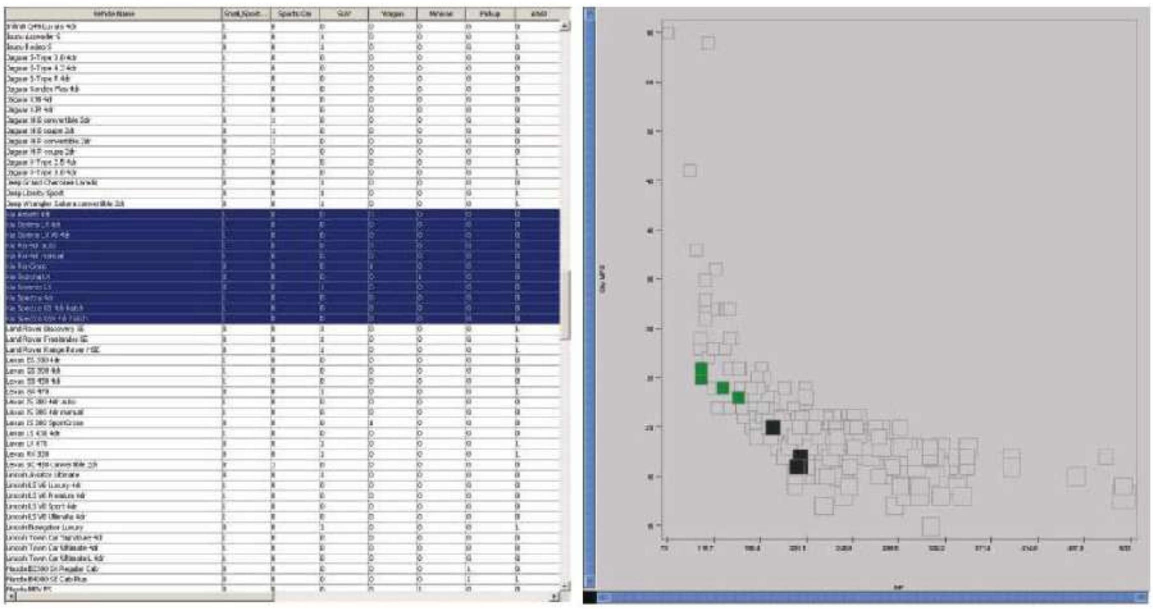


Figure 1.14 A table and scatterplot of Kia vehicles, showing a clear linear relationship in the data.

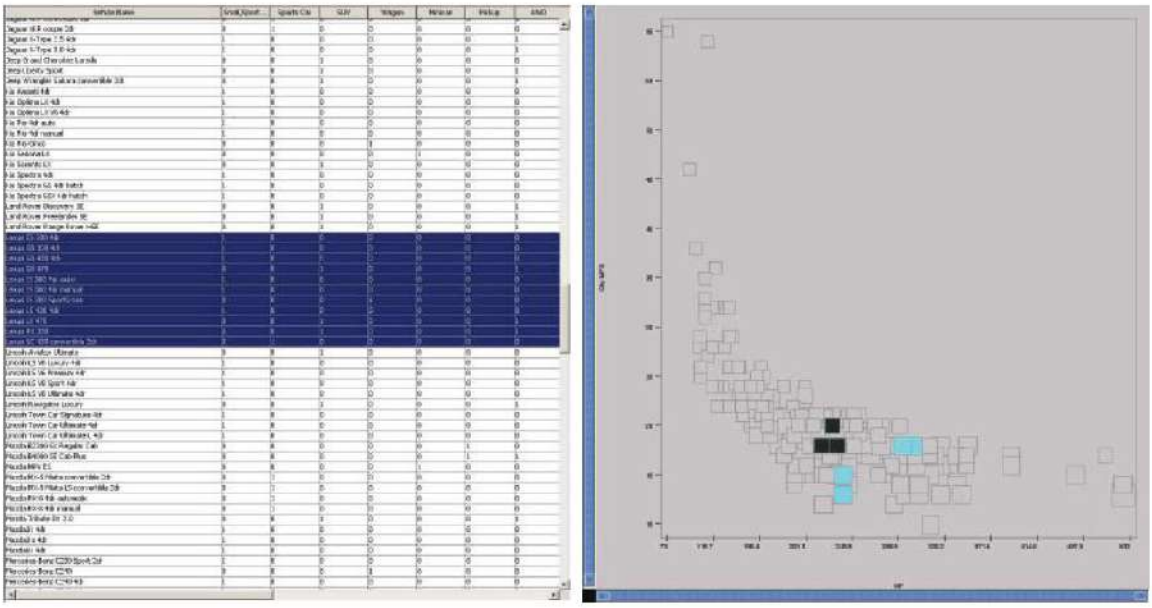


Figure 1.15 A table and scatterplot of Lexus vehicles, indicating that the hypothesis is not supported.

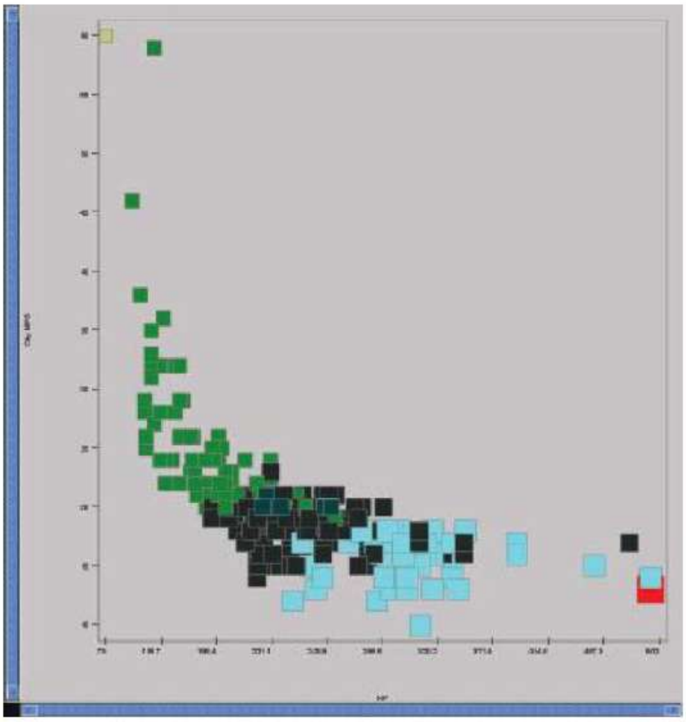


Figure 1.16 A scatterplot of all vehicles, offering numerous insights to explore.

1.11 The Role of the User in Visualization

When analyzing scatterplots, users must ask various questions to extract meaningful insights:

1. Are the observed trends applicable to the entire dataset or specific subsets?
2. How many records have missing values, and how does missing data impact conclusions?

3. What general patterns can be derived from the dataset?
    - What is the distribution of MPG?
    - Are there identifiable trends or groups?

While scatterplots offer valuable insights, they also have limitations. For example, Figure 1.16 lacks a legend, and overlapping points may obscure data. Understanding these drawbacks is crucial for improving visual analysis techniques.

In computer graphics, users typically engage at two key points in the visualization pipeline: either in creating the model of a scene to be rendered or in observing the final results. In many cases, users may have interactive control over parameters such as camera angles and animation settings. Since rendered scenes often simulate a three-dimensional world, human perception is generally well-equipped to interpret them. The main goal of such rendered images is to accurately convey the scene's contents, including object interactions and movements.

However, in data visualization, the user plays a role in multiple stages of the process, not just at the start or end. The purpose of the visualization greatly influences the extent of user involvement. Visualizations can generally be categorized based on their intended purpose:

Exploratory Visualization

The user analyzes a dataset to understand its contents and identify patterns or specific features. This type of visualization helps users discover relationships within the data that may not be immediately apparent.

Confirmatory Visualization

The user has already formulated a hypothesis based on computational analysis and wants to verify it through visualization. For example, if a hypothesis suggests a relationship between two variables, a confirmatory visualization can provide evidence to support or refute it.

Presentation Visualization

This type of visualization is designed to communicate specific findings to an audience. The visualization is refined with labels, color enhancements, and clear annotations to emphasize key insights.

An extension of presentation visualization, interactive presentations allow users to explore data independently. These are commonly found in web-based visualizations, where users can zoom in, filter information, or manipulate data points. However, creating effective interactive visualizations requires careful design to ensure users do not get lost in excessive complexity.

Most people are familiar with presentation visualizations, such as bar charts and line graphs, since they are commonly used in reports, news articles, and education. Until recently, most visualizations were static images due to the high effort required to produce them. However, advancements in computing power and software tools have made it easier to create interactive and exploratory visualizations.

Exploratory visualization is particularly useful in data-intensive fields such as genomics.

Each visualization type has specialized tools. Presentation visualizations often rely on software like Photoshop for precise design adjustments, while exploratory and interactive visualizations use more advanced computational tools for real-time data manipulation.

1.12 Data Foundations

Since every visualization starts with the data that is to be displayed, a first step in addressing the design of visualizations is to examine the characteristics of the data. Data comes from many sources; it can be gathered from sensors or surveys, or it can be generated by simulations and computations. Data can be raw (untreated), or it can be derived from raw data via some process, such as smoothing, noise removal, scaling, or interpolation. It also can have a wide range of characteristics and structures.

A typical data set used in visualization consists of a list of n records, $(r_1, r_2, \ldots, r_n)$.

Each record r_i consists of m (one or more) observations or variables, $(v_1, v_2, \ldots, v_m)$. An observation may be a single number/symbol/string or a more complex structure (discussed in more detail later in this chapter). A variable may be classified as either independent or dependent. An independent variable iv_i is one whose value is not controlled or affected by another variable, such as the time variable in a time-series data set. A dependent variable dv_j is one whose value is affected by a variation in one or more associated independent variables. Temperature for a region would be considered a dependent variable, as its value is affected by variables such as date, time, or location. Thus, we can formally represent a record as:

$$r_i = \left(iv_1, iv_2, \ldots, iv_{\{m_i\}}, dv_1, dv_2, \ldots, dv_{\{m_d\}}\right)$$

where m_i is the number of independent variables and m_d is the number of dependent variables. With this notation, we have $m = m_i + m_d$.

In many cases, we may not know which variables are dependent or independent. We can also think of data as being generated by some process or function. In this case, the independent variables would be considered the domain of the function, and the dependent variables would be the range of the function, as for each entry in the domain there is a single unique entry in the range. Note that, in general, a data set will not contain an exhaustive list of all possible combinations of values for the variables in its domain.

1.12.1 Types of Data

In its simplest form, each observation or variable of a data record represents a single piece of information. We can categorize this information as being ordinal (numeric) or nominal (nonnumeric). Subcategories of each can be readily defined.

Ordinal: The data take on numeric values:

- Binary—assuming only values of 0 and 1;
- Discrete—taking on only integer values or from a specific subset (e.g., {2, 4, 6});
- Continuous—representing real values (e.g., in the interval [0, 5]).

Nominal: The data take on nonnumeric values:

- Categorical—a value selected from a finite (often short) list of possibilities (e.g., red, blue, green);
- Ranked—a categorical variable that has an implied ordering (e.g., small, medium, large);
- Arbitrary—a variable with a potentially infinite range of values with no implied ordering (e.g., addresses).

Another method of categorizing variables is by using the mathematical concept of scale.

Scale: Three attributes that define a variable's measure are as follows:

- Ordering relation, with which the data can be ordered in some fashion. By definition, ranked nominal variables and all ordinal variables exhibit this relation.
- Distance metric, with which the distances can be computed between different records. This measure is clearly present in all ordinal variables but is generally not found in nominal variables.
- Existence of absolute zero, in which variables may have a fixed lowest value. This is useful for differentiating types of ordinal variables. A variable such as weight possesses an absolute zero, while bank balance does not.

Scale is an important attribute to examine when designing appropriate visualizations because each graphical attribute that we can control has a scale associated with it. Ideally, the scale of a data variable should be compatible with the scale of the graphical entity or attribute to which it is mapped, though it is somewhat dependent on the task to be performed with the visualization.

The type of data also determines the operations that can be applied to the data. While the equality and inequality operators (=, <>) can be applied to any type of data, comparison operators (<, >, <=, >=) can only be applied to ranked nominal and ordinal data types, and mathematical operators and distance computations (+, −, *, /) can only be applied to ordinal data types.

1.12.2 Structure within and between Records

Data sets have structure, both in terms of the means of representation (syntax), and the types of interrelationships within a given record and between records (semantics).

Scalars, Vectors, and Tensors

An individual number in a data record is often referred to as a scalar. Scalar values, such as the cost of an item or the age of an individual, are often the focus for analysis and visualization. Multiple variables within a single record can represent a composite data item. For example, a point in a two-dimensional flow field might be represented by a pair of values, such as a displacement in x and y. This pair, and any such composition, is referred to as a vector. Other examples of vectors found in typical data sets include position (2 or 3 spatial values), color (a triplet of red, green, and blue components), and phone number (country code, area code, and local number). While each component of a vector might be examined individually, it is most common to treat the vector as a whole.

Scalars and vectors are simple variants on a more general structure known as a tensor. A tensor is defined by its rank and by the dimensionality of the space within which it is defined. It is generally represented as an array or matrix. A scalar is a tensor of rank 0, while a vector is a tensor of rank 1. One could use a 3×3 matrix to represent a tensor of rank 2 in 3D space, and in general, a tensor of rank M in D-dimensional space requires D^M data values. An example of a tensor that might be found in a data record would be a transformation matrix to specify a local coordinate system.

Geometry and Grids

Geometric structure can commonly be found in data sets, especially those from scientific and engineering domains. The simplest method of incorporating geometric structure in a data set is to have explicit coordinates for each data record. Sometimes the geometric structure is implied, assuming some form of grid exists, and the data set is structured such that successive data records are located at successive locations on the grid.

Nonuniform, or irregular, geometry is also common. For example, in simulating the flow of wind around an airplane wing, it is important to have data computed densely at locations close to the surface, while data for locations far from the surface can be computed much more sparsely.

Other Forms of Structure

Examples of various structured data include MRI scans, CFD simulations, financial records, CAD models, remote sensing data, census data, and social network structures, each with unique attributes and connectivity relationships.

1.12.3 Data Preprocessing

In most circumstances, it is preferable to view the original raw data. In many domains, such as medical imaging, the data analyst is often opposed to any sort of data modifications, such as filtering or smoothing, for fear that important information will be lost or deceptive artifacts will be added. Viewing raw data also often identifies problems in the data set, such as missing data or outliers that may be the result of errors in computation or input. Depending on the type of data and the visualization techniques to be applied, however, some forms of preprocessing might be necessary. Some common methods for preprocessing data are briefly discussed later in this chapter. The interested reader is directed to the many fine textbooks dedicated to these topics for more in-depth coverage.

Metadata and Statistics

Information regarding a data set of interest (its metadata) and statistical analysis can provide invaluable guidance in preprocessing the data. Metadata may provide information that can help in its interpretation, such as the format of individual fields within the data records. It may also contain the base reference point from which some of the data fields are measured, the units used in the measurements, the symbol or number used to indicate a missing value, and the resolution at which measurements were acquired. This information may be important in selecting the appropriate preprocessing operations and in setting their parameters.

Various methods of statistical analysis can provide useful insights. Outlier detection can indicate records with erroneous data fields. Cluster analysis can help segment the data into groups exhibiting strong similarities. Correlation analysis can help users eliminate redundant fields or highlight associations between dimensions that might not have been apparent otherwise.

The most common statistics about data include the mean:

$$\mu = \frac{1}{n}\sum_{i=0}^{n}(x_i)$$

and the standard deviation.

$$\sigma = \sqrt{\sum (x_i - \mu)^2}$$

The most common statistical plot is the distribution of data, often represented in the form of a histogram.

Missing Values and Data Cleansing

One of the realities of analyzing and visualizing "real" data sets is that they often are missing some data entries or have erroneous entries. Missing data may be caused by several reasons, including, for example, a malfunctioning sensor, a blank entry on a survey, or an omission on the part of the person entering the data. Erroneous data is most often caused by human error and can be difficult to detect. In either case, the data analyst must choose a strategy for dealing with these common events.
Some common strategies include:

- Discard the bad record – Removing data records with missing or erroneous values is common but can lead to significant information loss.
- Assign a sentinel value – A designated out-of-range value (e.g., -5 in a 0–100 scale) can indicate missing values while keeping the record.
- Assign the average value – Missing values are replaced with the average of that variable. This keeps overall statistics stable but may obscure outliers.
- Assign value based on nearest neighbor – The missing value is estimated using the most similar record in the dataset.
- Compute a substitute value (Imputation) – Advanced statistical techniques (e.g., model-based imputation) generate probable values for missing data.

In all cases where a value is substituted, it is critically important to preserve the fact that this value is an approximation and ensure that visualizations communicate this uncertainty.

Normalization
Normalization is the process of transforming a data set so that the results satisfy a particular statistical property. A simple example of this is transforming the range of values a particular variable assumes so that all numbers fall within 0.0 to 1.0.
For a variable with a range d_{min} to d_{max}, the normalization formula is:

$$d_{normalized} = {(d_{original} - d_{min})}\Big/{(d_{max} - d_{min})}$$

Other transformations include:

- Square root normalization:

$$d_{sqrt-normalized} = \left(\sqrt{d_{original}} - \sqrt{d_{min}}\right) \Big/ \left(\sqrt{d_{max}} - \sqrt{d_{min}}\right)$$

- Logarithmic normalization:

$$d_{log-normalized} = \left(\log d_{original} - \log d_{min}\right) \Big/ \left(\log d_{max} - \log d_{min}\right)$$

These transformations help when data distributions are highly skewed. Additionally, bounding values (e.g., limiting extreme outliers) can improve data visualization clarity.

Segmentation

Segmentation involves dividing data into meaningful categories. In an MRI dataset, for instance, raw pixel values might be segmented into bone, muscle, fat, and skin categories.
A common segmentation algorithm follows split-and-merge refinement:

similarThresh = similarity measure indicating two regions have similar characteristics
homogeneousThresh = uniformity measure indicating a region is too nonhomogeneous

```
do {
    changeCount = 0
    for each region:
        - Compare with neighboring regions to find most similar
        - If within similarity threshold, merge regions
        - Evaluate homogeneity of merged region
        - If too nonhomogeneous, split into two subregions
    changeCount++
} until changeCount == 0
```

Challenges include undersegmentation (regions are too coarse) and oversegmentation (too many small regions). Refining similarity and homogeneity thresholds can help.

Sampling and Subsetting

Data may need to be resampled to a different resolution. Methods include:

a. Linear Interpolation

Linear interpolation is a method used to estimate the value of a variable at an intermediate point between two known values. Given a variable d at two locations, A and B, we can determine its value at a third point, C, by calculating the relative position of C between A and B. This percentage is then applied to adjust the value proportionally based on the difference between the values at A and B.

If the points lie along the x-axis, the relationship can be expressed mathematically as:

$$\frac{(x_C - x_A)}{(x_B - x_A)} = \frac{(d_C - d_A)}{(d_B - d_A)}$$

Rearranging for d_C:

$$d_C = d_A + (d_B - d_A) * \frac{(x_C - x_A)}{(x_B - x_A)}$$

This method is similar to normalization transformations. To generalize beyond a single axis, we use parametric equations. By defining both position and value changes parametrically, we can compute the value at C in any dimension:

$P(t) = P_A + V\,t$, where $V = P_B - P_A$.

Setting $P(t)$ equal to P_C, we solve for t and use it in:

$d(t) = d_A + U\,t$, where $U = d_B - d_A$.

b. Bilinear Interpolation

Bilinear interpolation extends linear interpolation to two dimensions. It is commonly used when estimating a variable at a point (x, y) within a uniform grid. If (x, y) coincides with a grid point, the value is directly retrieved. Otherwise, we determine the four surrounding grid points and interpolate in two stages: first along one axis, then along the other.

If the grid is spaced uniformly, the bounding grid points are:

(i, j) , $(i + 1, j)$, $(i, j + 1)$, and $(i + 1, j + 1)$

where i and j are the largest whole numbers less than x and y, respectively.

First, we interpolate horizontally. Let s be the relative position of x between i and $i + 1$. We compute the interpolated values at (x, j) and $(x, j + 1)$. Next, we interpolate vertically using the relative position t of y between j and $j + 1$:

$$d_{\{x,y\}} = d_{\{x,j\}} + t \times \left(d_{\{x,j+1\}} - d_{\{x,j\}}\right).$$

Ultimately, $d_{\{x,y\}}$ is a weighted average of the four surrounding grid values.

c. Nonlinear Interpolation

Linear interpolation provides smooth local changes, but abrupt variations may appear at grid points due to discontinuities in the first derivative. To improve smoothness, higher-order polynomial interpolation techniques are used. Quadratic and cubic splines are commonly employed in graphics for interpolating surface positions using control points and blending functions.

A widely used method is the Catmull-Rom spline, which ensures that the curve passes through control points. Given four control points (p_0, p_1, p_2, p_3), the cubic curve is defined as:

$$q(t) = 0.5 \times (1.0\ t\ t^2\ t^3) \times \begin{pmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{pmatrix} \times \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix}$$

or

$$q(t) = 0.5 \times ((2 \times p_1) + (-p_0 + p_2) \times t + (2 \times p_0 - 5 \times p_1 + 4 \times p_2 - p_3) \times t^2 + (-p_0 + 3 \times p_1 - 3 \times p_2 + p_3) \times t^3).$$

This method results in smoother transitions while preserving data values at grid points.

When reducing dataset size, subsampling (selecting a subset) may be used, but care must be taken to avoid losing significant patterns. Averaging neighborhoods or feature-preserving techniques are often preferable.

Dimension Reduction in Data Visualization

When dealing with high-dimensional data, visualization techniques often struggle to effectively represent all dimensions. To address this, dimension reduction methods are employed to simplify the dataset while preserving crucial information. This process can be achieved either manually, by selecting the most significant dimensions, or through computational techniques. Popular methods include:

- Principal Component Analysis (PCA)
- Multidimensional Scaling (MDS)
- Kohonen Self-Organizing Maps (SOMs)
- Local Linear Embedding (LLE)

Each of these techniques transforms high-dimensional data into a lower-dimensional form, highlighting key features like clusters, patterns, and outliers. However, these methods often yield varying results based on initial configurations, parameter settings, and computational variations.

a. Principal Component Analysis (PCA)

PCA is one of the most widely used techniques for dimension reduction. It transforms data into a new set of dimensions (principal components) that are linear combinations of the original attributes. These new dimensions are sorted based on their ability to explain data variance. The steps for PCA computation are:

1. Identify a line that best spreads the n-dimensional data—the first principal component.
2. Select another line perpendicular to the first that maximizes spread—the second principal component.
3. Repeat until the required number of principal components is determined.

Formal Steps in PCA Computation:

1. Normalize the data by subtracting the mean of each dimension.
2. Compute the covariance matrix.
3. Determine the eigenvectors and eigenvalues of the covariance matrix.
4. Rank eigenvectors based on their eigenvalues in descending order.
5. Select the top eigenvectors corresponding to the desired number of dimensions.
6. Construct a transformation matrix using these eigenvectors.
7. Convert the original data into the new space using matrix multiplication.

Example: In Figure 1.17, PCA is applied to a four-dimensional dataset, reducing it to two dimensions and visualizing it using star glyphs. The result demonstrates PCA's effectiveness in clustering similar data points.

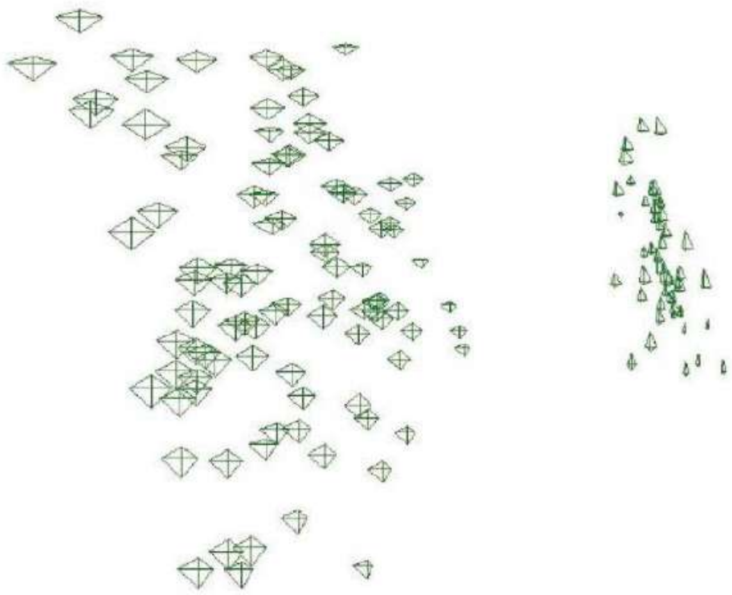


Figure 1.17 The Iris dataset is shown with star glyphs, positioned using the first two principal components. Each glyph's four lines represent different variables, revealing clear clustering.

b. Multidimensional Scaling (MDS)

MDS aims to create a low-dimensional representation of data that preserves the inter-point distances as accurately as possible. The goal is to minimize the difference between:

- dij: Distance between points in the original high-dimensional space.
- δij: Distance between points in the reduced space.

Steps for MDS Computation:

1. Compute pairwise distances in the original n-dimensional space.
2. Assign initial positions for data points in the lower-dimensional space.
3. Calculate stress, a measure of discrepancy between distances.
4. If the stress is below a threshold, stop; otherwise, adjust points.
5. Modify each point's position based on computed direction vectors.
6. Repeat the process until an optimal configuration is reached.

MDS can sometimes get stuck in local optima, requiring multiple runs with different initial positions or methods like simulated annealing to escape poor solutions. Some visualization tools even use PCA to initialize MDS, reducing computation time.

c. Other Dimension Reduction Techniques

Several non-linear techniques also exist, such as:

- Self-Organizing Maps (SOMs): Neural network-based technique for clustering high-dimensional data.
- Local Linear Embedding (LLE): Retains local neighborhood relationships in the reduced space.

Mapping Nominal Dimensions to Numbers

Many datasets contain nominal variables (e.g., car brands, product names). Mapping these variables into a numerical format for visualization requires careful handling to avoid artificial relationships.

Methods for Mapping Nominal Data:

- Using colors or shapes: Effective for distinguishing categories.
- Applying statistical similarity: If two categories exhibit similar distributions, they can be assigned nearby numerical values.
- Correspondence analysis: A statistical method for mapping categorical data into numerical space.

Aggregation and Summarization

When dealing with large datasets, grouping similar data points can simplify analysis. Aggregation involves two key steps:

1. Grouping similar points based on distance, clustering techniques, or partitioning methods.
2. Representing these groups using statistical measures such as means, cluster size, or data distribution.

In Figure 1.17, the Iris dataset is visualized using parallel coordinates, illustrating both original data and aggregated clusters.

Smoothing and Filtering

To reduce noise and enhance data clarity, smoothing techniques like convolution and exponential smoothing are used:

- Convolution-based smoothing: Applies weighted averaging to neighboring values.
- Exponential smoothing: Assigns greater weight to recent data points while reducing the influence of older ones.

Formula for exponential smoothing: where α is the decay factor controlling the influence of past values.

Raster-to-Vector Conversion

Raster-based data (e.g., images) can be converted into vector-based representations for improved analysis and transformation. Techniques include:

- Thresholding: Assigns pixel regions based on intensity levels.
- Region growing: Merges similar neighboring pixels into clusters.
- Boundary detection: Identifies edges by applying convolution filters.
- Thinning: Reduces broad features to single-pixel width structures for better representation.

Summary of Data Preprocessing

Applying dimension reduction and preprocessing techniques enhances data visualization, enabling better pattern recognition and analysis. However, transparency about these transformations is crucial to prevent misinterpretations. Users must be informed about the modifications applied to ensure accurate data insights.

1.13 Understanding Data Sets for Visualization

Data visualization becomes more meaningful when users understand the context of the data being represented. While visualizations can be appreciated without knowing the data, having background information enhances interpretation. Table 1.2 gives various data sets used throughout this book and provides sources where additional data sets can be found.

For more data sets, refer to:

- U.S. Government Open Data Portal: www.data.gov
- European Union Open Data Portal: open-data.europa.eu

Table 0.2 Datasets for visualization

#	Dataset	Description	Format	Source & Link
1	DJIA Data (djia-100.xls)	100+ years of daily closing values	Excel	Analyze Indices http://www.analyzeindices.com/dow-jones-history.shtml
2	Colorado Elevation (colorado_elev.vit)	2D elevation grid (400×400)	Binary	OpenDX http://www.opendx.org/
3	Flow Field (uvw.dat)	3D vector field (turbulent flow)	Plain text	Worcester Polytechnic Institute (J. Lu & G. Tryggvason) http://www.me.wpi.edu/Tryggvason
4	U.S. City Temperature (city_temp.xls)	Avg. January temps for 56 cities	Excel	CMU StatLib http://lib.stat.cmu.edu
5	Head CT Scan (CThead.zip)	3D scalar CT scan (113 slices)	Binary	Stanford Graphics http://graphics.stanford.edu/data/voldata/
6	Multivariate Data (cars.xls, detroit.xls, cereal.xls)	Various datasets for analysis	Excel	CMU StatLib http://lib.stat.cmu.edu
7	Health Data	UNICEF/CDC health statistics	Excel	OpenIndicators http://www.openindicators.org
8	VAST Contest Data	Synthetic heterogeneous datasets	Various	VAST Archive (University of Maryland) http://hcil.cs.umd.edu/localphp/hcil/vast/archive/index.php
9	U.S. Census (counties.csv)	Demographics for 3,137 counties	CSV	U.S. Census Bureau / OpenIndicators http://www.openindicators.org/data
10	Iris Data (iris.csv)	Plant species measurements	CSV	UCI Machine Learning Repository http://archive.ics.uci.edu/ml/datasets/Iris

1.14 Human Perception and Information Processing

This section explores human perception and the ways in which images and graphics are interpreted. Initially, perception studies focused on the capabilities of the human visual system. Over time, cognitive processes and pattern recognition became central areas of study. This document details both approaches, providing insights into human perception.

1.14.1 Understanding Perception

Humans perceive data, but the exact mechanisms behind perception remain complex. Visualizations convey data that we interpret, but ensuring that these representations are universally understood is a challenge. The study of perception helps refine how information is presented, ensuring clarity and minimizing misinterpretation.

Perception is commonly defined as the process of recognizing, organizing, and interpreting sensory information. This process involves multiple senses—sight, hearing, touch, smell, and taste—with vision and hearing being the most extensively studied. Simply put, perception enables us to construct a mental representation of our environment, though this representation is not an exact replica of reality. The brain makes assumptions to resolve ambiguities in sensory data based on context and prior knowledge.

1.14.2 Visual Illusions and Misinterpretation

Visual representations are sometimes misinterpreted because they do not align with our perceptual system or are deliberately designed to be misleading. Illusions highlight how perception can be distorted. For example, Figure 1.18 depicts objects that seem feasible at first glance but are physically impossible. These illusions demonstrate how our perception system fills in gaps, often leading to incorrect conclusions.

Another example is the Hermann grid illusion (Figure 1.19), where black squares appear at the intersections of white spaces. These perceptual effects can be reduced by focusing directly on an intersection, highlighting the dynamic nature of visual processing. Similar effects are seen in Figure 1.20, where seemingly bent lines and non-existent shapes emerge due to forced interpretations by our visual system.

The study of perception aims to identify the mechanisms behind these illusions and improve the accuracy of visual data representation. By understanding these principles, we can design better visualizations that minimize misinterpretation.

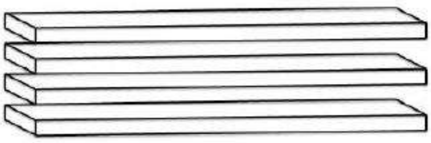


Figure 1.18 Four ≠ Three Illusion: the object is impossible to construct since the left side appears to have four boards while the right side has only three.

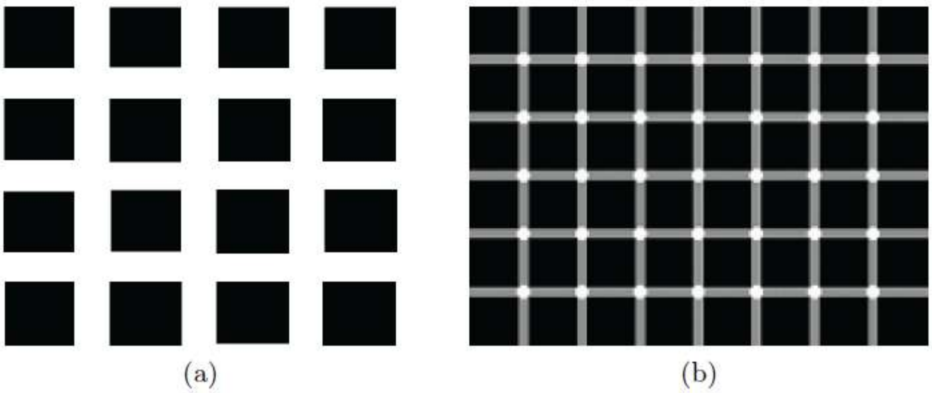
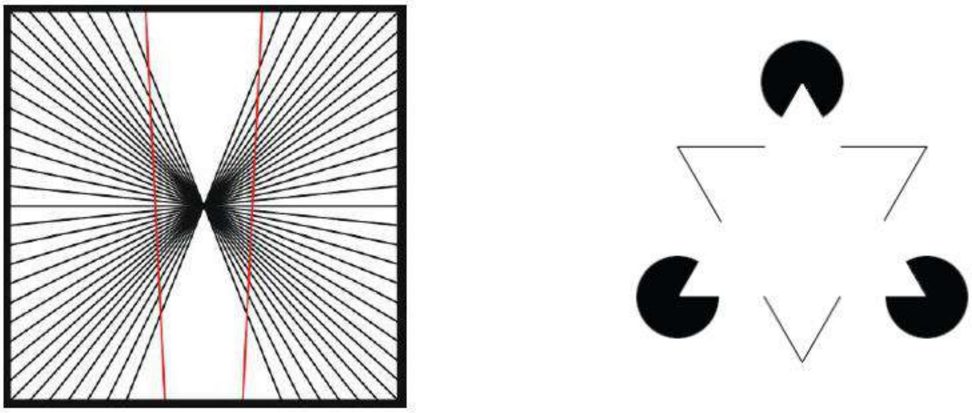


Figure 1.19 Hermann Grid Illusion (a) As you gaze at the image, illusory black spots appear at the intersections. (b) A variation of (a), creating an even more dynamic and engaging visual effect.



(a) (b)

Figure 1.20 (a) Hering Illusion: The red lines appear curved, but they are actually straight. You can verify this using a straight edge.

1.14.3 The Science Behind Perception

To analyze perception, two main approaches are used:

1. Measurement-Based Approaches: These involve assessing sensory responses, such as comparing line lengths or recognizing objects in complex scenes.
2. Model-Based Approaches: These aim to explain how perception occurs, using theoretical frameworks to predict future perceptual responses.

Both approaches are interlinked—measurements help refine models, while models guide new measurements. Ignoring perceptual principles can result in poor visualization design, as seen in illusions that create unintended distortions. Factors like color perception and three-dimensional interpretation also play critical roles in ensuring effective visualization.

1.14.4 Physiology of Vision

The Role of Photoreceptors

The human visual system captures and processes light scattered from objects, forming a two-dimensional representation on the retina. This is made possible by photoreceptors, small sensory cells that respond to light waves.

The Visible Spectrum

Visible light is a narrow section of the electromagnetic spectrum, ranging from approximately 380 nm (near ultraviolet) to 700 nm (near infrared) (Figure 1.21). Individual differences, such as color blindness, affect the ability to perceive specific wavelengths.

Beyond light itself, physical properties of objects influence perception. Variations in wavelength, object geometry, illumination, and reflectance properties all impact how an observer interprets their surroundings.

Understanding these physiological and perceptual principles allows us to design better visual representations, ensuring clarity and accuracy in data interpretation.

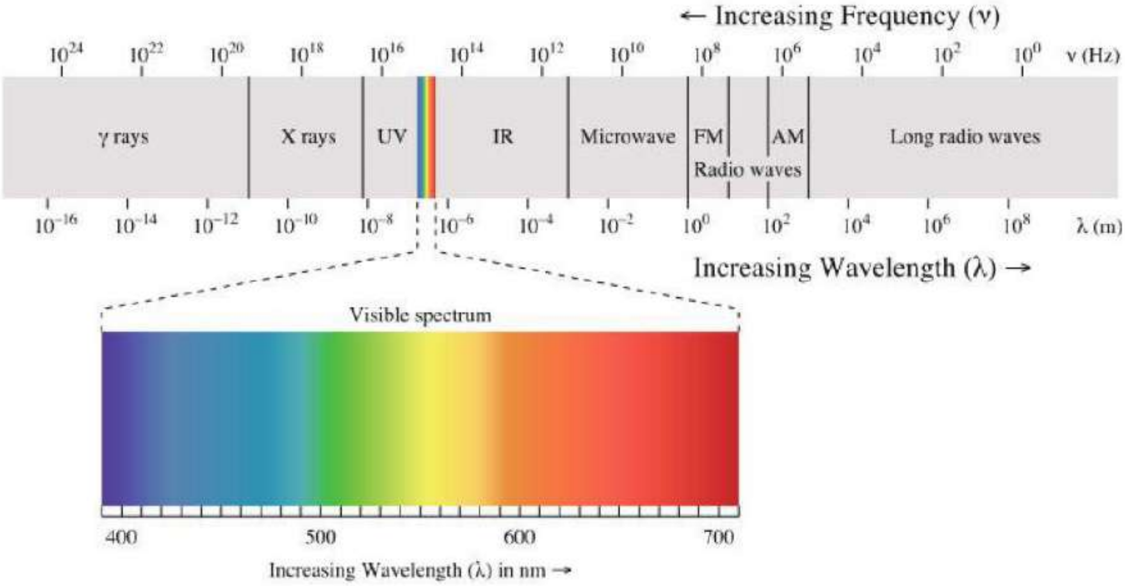


Figure 1.21 The electromagnetic spectrum with an expanded visible light spectrum

1.15 The Human Visual System

The human eye is a remarkable organ, allowing us to perceive the world around us. Despite its complexity, its structure is relatively simple. The eye functions as a fluid-filled sphere containing light-sensitive cells, connected to the brain via the optic nerve and controlled by six motion-regulating muscles. The key components involved in processing light include the cornea, iris, pupil, lens, and retina.

1.15.1 Structure and Function of the Eye

Lens System and Muscles

The six muscles surrounding the eye are responsible for its movement, allowing us to focus on different objects. These muscles stabilize images by making continuous small adjustments, even though we do not consciously perceive these movements. Unlike engineered optical systems where such motions might be seen as imperfections, in the human visual system, they enhance visual perception (Figure 1.22).

The eye's optical system functions similarly to a dual-lens camera. The key components include:

- Cornea: The eye’s outermost layer, which protects internal structures and helps focus incoming light.
- Pupil: A circular opening in the iris that regulates the amount of light entering the eye, much like a camera aperture.

- Iris: A colored ring of muscles that controls the pupil’s size.
- Lens: A flexible, crystalline structure that adjusts its shape to focus light on the retina, allowing for clear vision at different distances. With age, the lens loses elasticity, reducing its ability to focus on nearby objects.

Once light passes through these structures, it reaches the retina, where it is converted into neural signals.

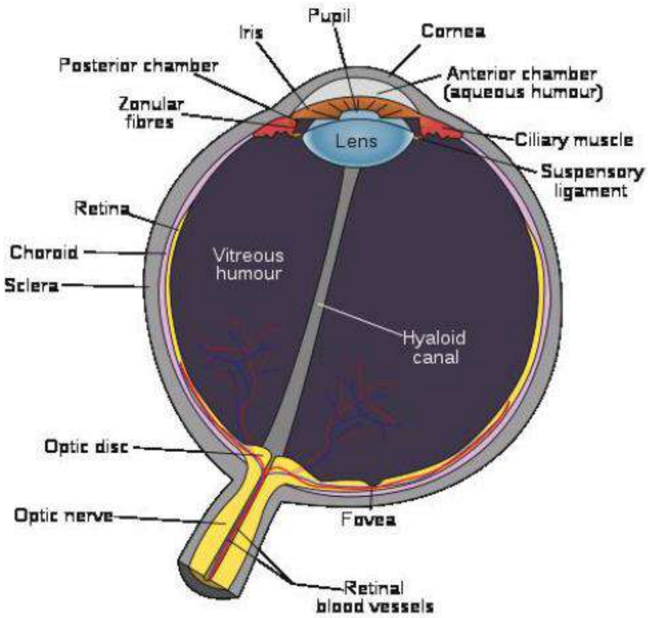


Figure 1.22 Horizontal cross-section of the human eye, viewed from above.

1.15.2 The Retina: The Light-Sensitive Layer

The retina contains specialized photoreceptors responsible for detecting light and color. These photoreceptors are divided into two types:

Rods and Cones

- Rods: Highly sensitive to light, they enable vision in low-light conditions (scotopic vision) but do not perceive color.
- Cones: Function in bright light (photopic vision) and are responsible for color vision. There are three types:
    - S-cones (Short wavelength): Sensitive to blue light.
    - M-cones (Medium wavelength): Sensitive to green light.
    - L-cones (Long wavelength): Sensitive to red light.

The highest concentration of cones is in the fovea, the central region of the retina, responsible for sharp and detailed vision. Rods are more abundant in the peripheral retina, which aids in detecting motion and objects in dim lighting.

The Blind Spot

The blind spot is an area of the retina where the optic nerve exits the eye, devoid of photoreceptors. This can be identified through a simple experiment: by focusing on a specific point while moving closer or farther from an image, a small object in the peripheral vision disappears.

Image Processing in the Retina

The retina performs initial image processing before transmitting information to the brain. Since the optic nerve contains fewer fibers than the total number of photoreceptors, compression and pre-processing of visual data occur within the retina. The connections between photoreceptors and neurons help in reducing redundant information and enhancing contrast and detail.

1.16 Visual Processing in the Brain

1.16.1 The Pathway from Eye to Brain

The brain plays a critical role in visual perception. The optic nerves from both eyes cross at the optic chiasma, where half of the nerve fibers switch to the opposite hemisphere of the brain. This helps in depth perception and combining images from both eyes into a unified vision.

1.16.2 Role of Neurons in Image Processing

Neurons in the retina and brain process visual signals using frequency-modulated pulses. The retina has four layers of neurons that refine the initial input before transmitting it to the brain:

- Horizontal cells: Connect multiple photoreceptors, refining the image.
- Bipolar cells: Transmit signals from photoreceptors to other layers.
- Amacrine cells: Link bipolar and ganglion cells, enhancing contrast.
- Ganglion cells: Send processed visual information to the brain via the optic nerve.

Since the retina develops from brain tissue during embryonic growth, it serves as an extension of the central nervous system, allowing sophisticated preprocessing before images reach the brain.

1.17 Visual Perception and Adaptation

1.17.1 Sensitivity to Light and Motion

- The eye detects ratios of light intensity, rather than absolute brightness, which helps in adaptation and contrast sensitivity.
- Rods are sensitive to flickering lights at high frequencies, up to a threshold where the flicker is perceived as constant illumination.

1.17.2 The Air Force Experiment

Experiments have demonstrated the eye's remarkable speed in processing images. The U.S. Air Force conducted a test where pilots viewed an aircraft image flashed for 1/220th of a second in a dark room. Pilots could not only perceive the afterimage but also identify the aircraft type, highlighting the brain's rapid processing ability.

The human visual system is a sophisticated network of structures working together to provide sight. From the cornea and lens focusing light, to the retina converting it into neural signals, and the brain interpreting the information, each part plays a crucial role. Understanding these processes helps us appreciate the complexity and efficiency of human vision.

1.18 Understanding Eye Movements and Their Role in Perception

1.18.1 Eye Movement

Eye movement plays a crucial role in perception, enabling humans to interpret scenes and images effectively. It also explains visual illusions, such as the perception of black dots in certain optical illusions. Various types of eye movements contribute to scene interpretation and focus shifting.

Types of Eye Movements

a. Smooth Pursuit Movements

As the name suggests, smooth pursuit movements allow the eyes to track a moving object smoothly without abrupt jumps. This type of movement occurs when following an object in motion.

Example: Hold your forefinger at arm’s length, focus on it, and move your arm left and right. Your eyes will smoothly follow your fingertip. These movements are also called conjugate eye movements, meaning both eyes move in a coordinated manner, maintaining equal angles in all directions.

b. Vergence Eye Movements

Vergence movements involve non-conjugate movements, meaning each eye moves in a different direction. This results in different angles relative to the face’s normal position.

Example: Moving your finger closer to your face while keeping your focus on it forces your eyes to turn inward, creating a vergence movement. This movement also helps merge depths in visual illusions.

c. Saccadic Eye Movements

Saccadic movements occur when the eyes rapidly shift focus between multiple points of interest. These movements can reach speeds of up to 1000 degrees per second, bringing objects into focus in as little as 25 milliseconds. Once a target is located, the eyes hold their position.

The selection of focal points occurs in the frontal region of the cerebral cortex, where the brain determines which visual elements require attention. This selection process is influenced by various factors and can appear somewhat random.

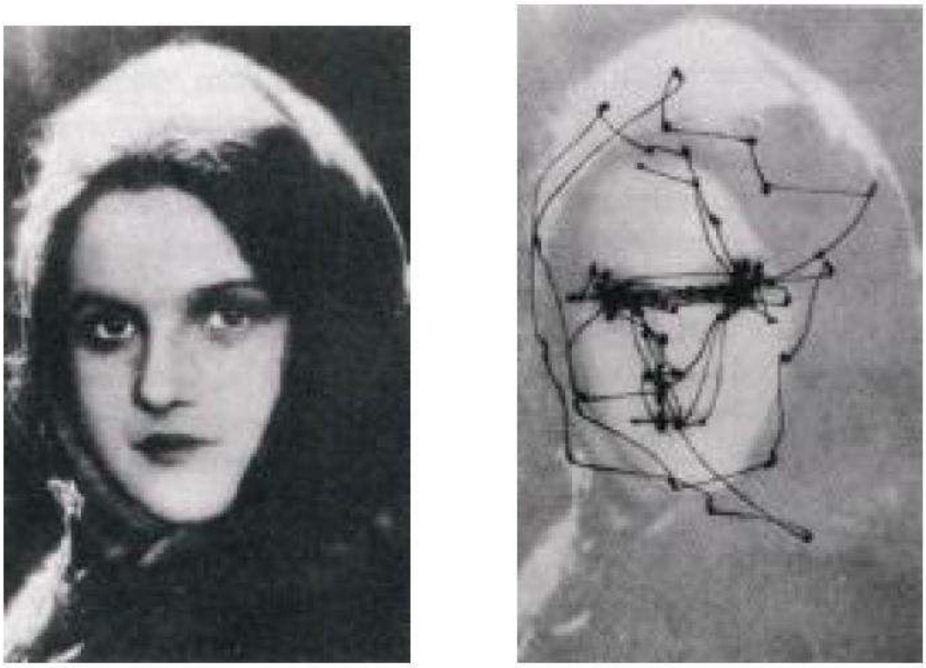
d. Saccadic Masking

During rapid saccadic movements, the brain filters out blurred images, ensuring a seamless visual experience. This process, known as saccadic masking or saccadic suppression, prevents perception of the motion blur between two focused points. Instead, the brain constructs a continuous flow of meaningful visual information by combining the static images from the points where the eye has stopped.

1.18.2 Eye Tracking and Visual Focus

Studies in marketing and psychology have explored how eye movements influence attention and focus. Researchers use eye-tracking technology to analyze where individuals direct their gaze when viewing images or advertisements.

For instance, when examining a face, as shown in Figure 1.23(a), the eye naturally follows a specific path, leading to concentrated points of focus. The tracking results in Figure 1.23(b) illustrate these areas of high visual interest.



(a) (b)

Figure 1.23 (a) The face used to study eye tracking. (b) The results of the tracking of the gaze

By understanding the different types of eye movements and their effects on perception, researchers can develop strategies to optimize visual presentations, whether in advertising, design, or user interface development.

1.19 Perceptual Processing

Perceptual processing is the mechanism through which sensory information moves from basic, low-level awareness to more complex cognitive understanding. A widely accepted model of information processing illustrates this flow, showing how memory plays a role in post-processing (Figure 1.24). However, this model only partially captures reality, as perception can be both automatic (preattentive) and controlled (attentive).

- **Preattentive perception** occurs almost instantly, within approximately 250 milliseconds (ms), without conscious effort. Some visual elements "pop out" due to preconscious visual processes.
- **Attentive perception** is more deliberate, involving short-term memory. It helps organize and interpret visual stimuli into structured objects, enabling tasks such as recognizing an emergency exit.

The process begins with detecting low-level attributes, which are then structured into higher-level perceptions, ultimately integrating memory for complex tasks.

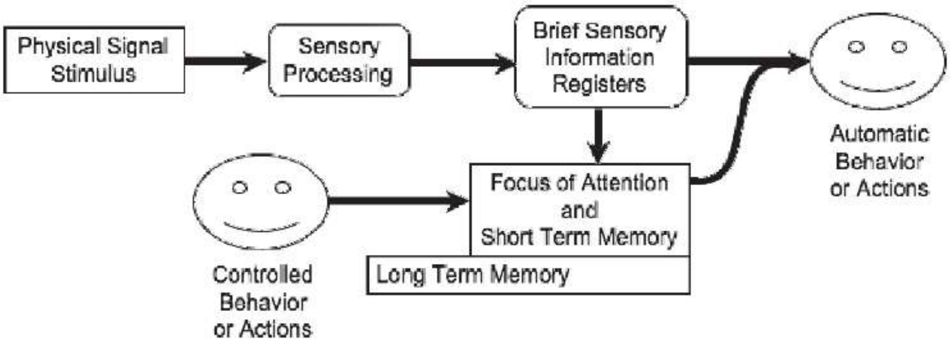


Figure 1.24 Traditional Model of Sensory Data Processing for Cognition

1.19.1 Preattentive Processing

Researchers have extensively studied how the human visual system interprets images. Early findings revealed a set of visual properties that are detected rapidly and effortlessly by the low-level visual system. These properties were initially labeled as "preattentive" since they seemed to be processed before focused attention was engaged. Although we now know that attention influences even early visual processing, the term "preattentive" remains useful in describing the speed and ease of recognizing these properties.

Tasks that can be completed within 200-250 ms without eye movement are considered preattentive. Since eye movements require at least 200 ms to initiate, this means the visual system can analyze an entire scene in parallel. For example, detecting a red circle among blue circles happens effortlessly.

Examples of Preattentive Processing

1. Hue Differentiation:
   When a red circle appears among blue circles, it is immediately noticeable due to its unique color.
2. Shape Differentiation:
   A red circle among red squares is quickly identified because of its curved shape.

3. Conjunction Search:
   When the target object shares features with distractors (e.g., a red circle among red squares and blue circles), preattentive processing fails. Instead, viewers must conduct a slower, serial search.

Visual Features in Preattentive Processing

For effective visualization, the display should take advantage of the visual system's capabilities while avoiding interference that could obscure important information. Common preattentive visual features include:

- Length, width, size
- Curvature, number, terminators
- Intersection, closure
- Hue, intensity, flicker
- Direction of motion, stereoscopic depth
- Lighting direction and 3D depth cues

Key perceptual attributes linked to these features include:

- **Luminance and Brightness:** Luminance refers to the measured light intensity, while brightness is the perceived light level.
- **Color:** Hue and intensity variations aid in distinguishing objects.
- **Texture:** Differences in surface patterns help define regions.
- **Shape:** The geometric properties of objects impact recognition.

Preattentive Visual Tasks

Psychologists have identified several tasks that rely on preattentive processing:

1. Target Detection:
   Viewers quickly determine whether a unique object is present in a display
2. Boundary Detection:
   A texture boundary is easily identified when two groups of elements have distinct visual properties.
3. Region Tracking:
   Objects with unique features can be tracked as they move.
4. Counting and Estimation:
   Viewers can quickly count or estimate the number of elements with distinct features.

By leveraging preattentive visual features, effective data visualization can highlight critical patterns and trends without overwhelming the viewer.

Theories of Preattentive Processing

Several theories have been proposed to explain how preattentive processing occurs in the visual system. This section discusses four well-known models: Feature Integration Theory, Texton Theory, Similarity Theory, and Guided Search Theory. Additionally, we briefly cover the phenomenon of Postattentive Vision, which highlights how prior exposure to a scene does not necessarily help a viewer answer detailed questions about its content.

a. **Feature Integration Theory (Anne Treisman)**

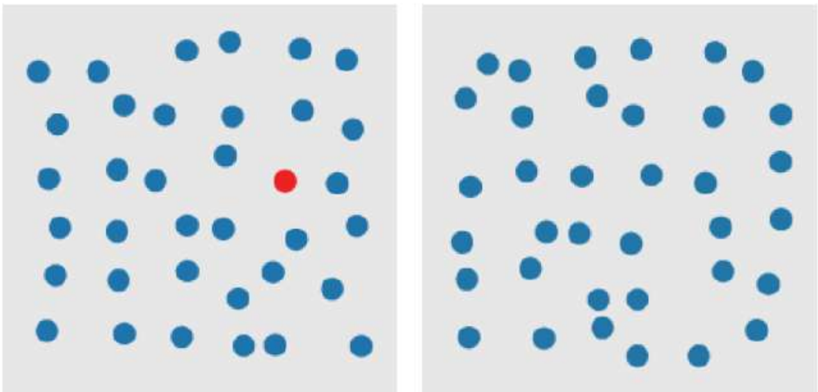
Anne Treisman pioneered research on preattentive processing, focusing on two key questions:

1. **Which visual properties are detected preattentively?** She referred to these properties as **preattentive features**.
2. **How does the human visual system process these features?**

Treisman's Experiments

Treisman conducted experiments using target detection and boundary detection to classify preattentive features:

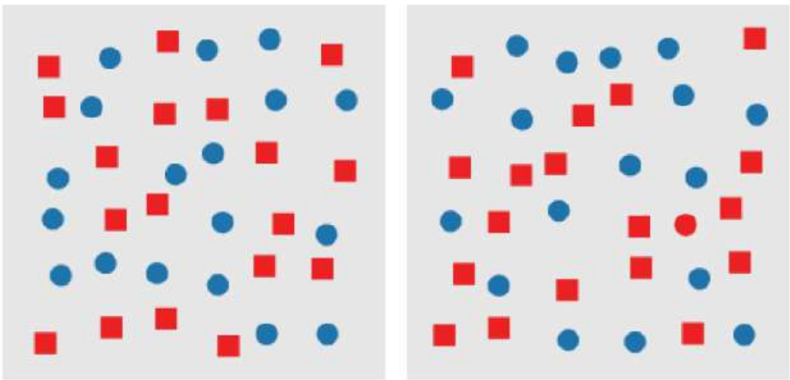
- **Target Detection:** Participants identified whether a target object was present among distractors (Figures 1.25, 1.26).
- **Boundary Detection:** A group of target elements with unique features was placed among distractors to test whether the boundary was detectable (Figure 1.27).



(a) Target is present in a sea of blue circle distractors.

(b) Target is absent.

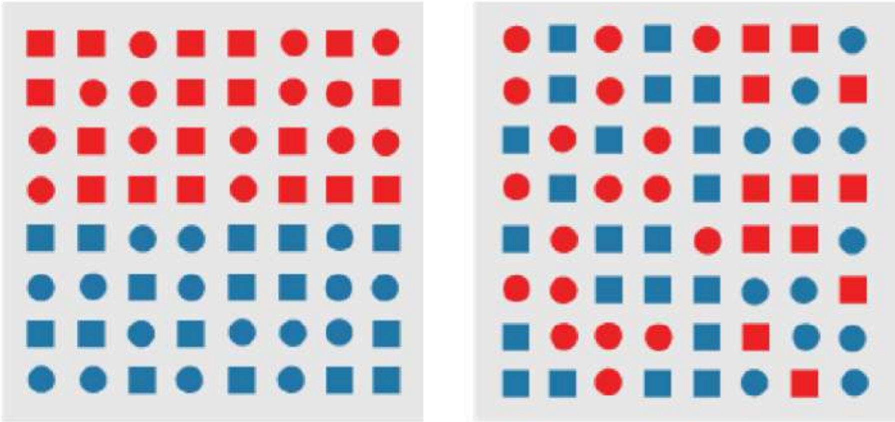
Figure 1.25 An example of searching for a target red circle based on a difference in curvature.



(a) Target is absent in a sea of red square and blue circle distractors.

(b) Target is present.

Figure 1.26 An example of a conjunction search for a target red circle



(a) A boundary defined by a unique feature hue (red circles and red squares on the top, blue circles and blue squares on the bottom) is preattentively classified as horizontal.

(b) A boundary defined by a conjunction of features (red circles and blue squares on the left, blue circles and red squares on the right) cannot be preattentively classified as vertical.

Figure 1.27 An example of a boundary detection, from Treisman's experiments

Performance in preattentive tasks was measured using:

- **Response Time Model:** Viewers completed tasks as quickly as possible while maintaining accuracy. If task completion time remained constant regardless of the number of distractors, the feature was considered preattentive.
- **Accuracy Model:** A display was shown for a fixed duration (typically 200-250 ms) before being removed. If viewers could accurately complete the task regardless of the number of distractors, the feature was deemed preattentive.

Key Findings

- Some visual features are **asymmetric** in detection.
  - A sloped line among vertical lines is detected preattentively.
  - A vertical line among sloped lines is **not** detected preattentively.
- Background distractors influence detection efficiency.
- Features are encoded into **feature maps**, which register specific properties like color, orientation, and shape (Figure 1.28).
- **Unique features** can be detected in parallel, while **conjunction targets** require serial searching in the **master map of locations**.
- **Feature differences influence search time:**
  - When a target is highly distinct from distractors, search time is fast.
  - As similarity increases, search time slows.

Later research showed that some conjunction searches (e.g., involving motion, depth, and color) can still be preattentive under certain conditions.

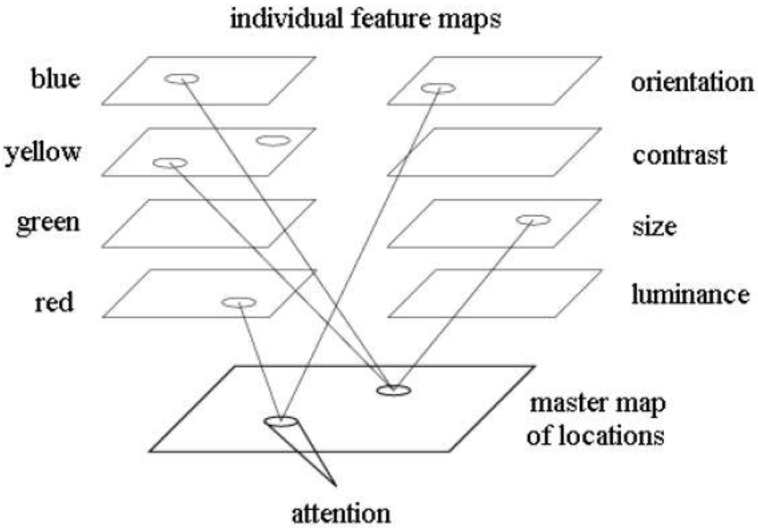


Figure 1.28 Treisman's Feature Integration Model for Early Vision: Individual feature maps can be accessed to detect specific attributes, while focused attention operates through a serial scan of the master location map.

b. **Texton Theory (Bela Julesz)**

Bela Julesz contributed to our understanding of texture perception and preattentive processing. His research initially focused on statistical texture analysis but later led to the Texton Theory.

Key Concepts

- The human visual system detects textons, fundamental visual elements grouped into three categories:
  1. **Elongated blobs** (e.g., line segments, ellipses) with properties such as hue, width, and orientation.
  2. **Terminators** (endpoints of line segments).
  3. **Intersections** (where lines cross).
- Differences in textons or their density can be detected preattentively, but their spatial arrangement cannot.

Experimental Findings

- **Texture Segregation:** Observers can preattentively detect texture boundaries based on texton differences (Figure 1.29).
- Even when objects appear different in isolation**,** if they share the same textons, they become difficult to distinguish in a texture**.**

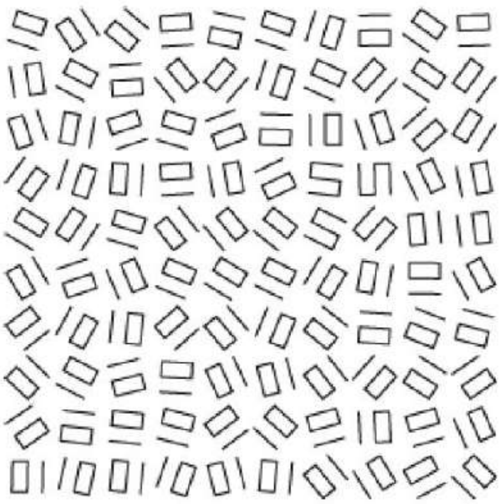


Figure 1.29 A target group of b-textons is difficult to detect in a background of a-textons when a random rotation is applied.

Julesz's findings reinforced the idea that preattentive processing occurs in parallel, while focused attention is required for positional details.

c. **Similarity Theory (Quinlan & Humphreys, Duncan & Humphreys)**

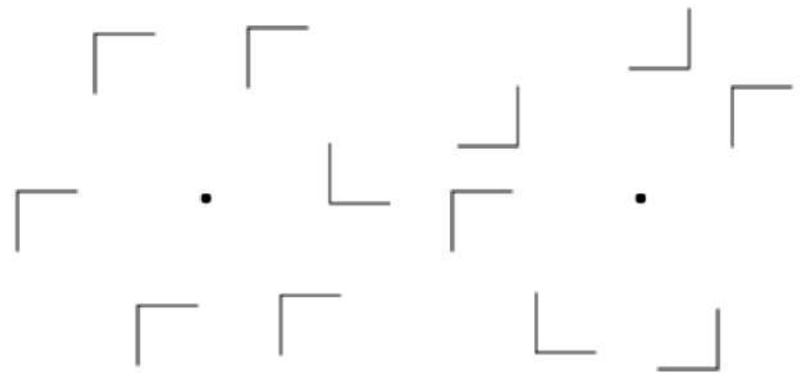
This theory challenges the parallel vs. serial dichotomy, suggesting that search time depends on similarity rather than feature uniqueness.

Key Findings

- **Search efficiency depends on:**
    1. **T-N Similarity (Target vs. Nontarget):**
        - When the target is highly similar to distractors, search becomes difficult.
    2. **N-N Similarity (Nontarget vs. Nontarget):**
        - When distractors are very different from each other, search becomes harder because no single feature dominates.

Experimental Results

- **High N-N Similarity:**
    - If distractors are uniform, the target stands out easily (Figure 1.30a).
- **Low N-N Similarity:**
    - If distractors vary significantly, the target is harder to detect (Figure 1.30b).



(a) High N-N (nontarget-nontarget) similarity allows easy detection of target L.

(b) Low N-N similarity increases the difficulty of detecting the target L.

Figure 1.30 Example of N-N similarity affecting search efficiency for a target shaped like the letter L

This model suggests that search time increases when T-N similarity is high or when N-N similarity is low.

d. **Guided Search Theory**

Guided Search Theory builds on Feature Integration Theory, proposing that attentional guidance can enhance visual search efficiency. It introduces:

- **Bottom-up Processing:** Rapid, preattentive feature detection.
- **Top-down Processing:** Goal-directed attention that prioritizes relevant features.

Search efficiency improves when both processes work together.

e. **Postattentive Vision**

After seeing a scene, viewers often fail to recall detailed information. This suggests that:

- Pre-exposure does not guarantee better recognition.
- Attention plays a crucial role in encoding visual details.

Table 0.3 Summary of Preattentive processing theories

Theory	**Key Idea**	**Search Type**	**Key Researchers**
Feature Integration	Features are processed in parallel, but combinations require serial search.	Parallel & Serial	Anne Treisman
Texton Theory	Textons (blobs, terminators, intersections) drive perception.	Parallel	Bela Julesz
Similarity Theory	Search difficulty depends on target-distractor similarity.	Continuous Spectrum	Quinlan, Humphreys, Duncan
Guided Search	Attention is guided by both bottom-up and top-down processes.	Parallel & Guided	Wolfe

Understanding these theories helps in designing effective visual displays, data visualizations, and user interfaces, optimizing information retrieval through efficient preattentive processing.

1.20 Visual Search Theories and Perceptual Processing

1.20.1 Duncan and Humphreys' Three-Step Theory of Visual Selection

Duncan and Humphreys proposed a hierarchical model of visual selection, where the visual field is segmented into structural units. These units share common properties such as spatial proximity, hue, shape, or motion, and can be further divided into smaller subunits.

Key Concepts:

1. Hierarchical Representation:
    - The visual field is divided into structural units, which are described by properties like spatial location, hue, texture, and size.
    - This segmentation occurs in parallel across the visual scene.
2. Resource Allocation and Template Matching:
    - Since visual short-term memory has limited capacity, a resource allocation mechanism is used.
    - A template of the target information is available, and each structural unit is compared to it.
    - More resources are allocated to units that match the template better.
3. Search Efficiency and Similarity Effects:
    - If a structural unit does not match the template well, related units can be rejected efficiently**,** speeding up the search.
    - The probability of access to visual short-term memory depends on the degree of match with the template.
    - **Target-Distractor (T-N) Similarity:** Higher similarity increases competition for resources, slowing down the search.
    - **Distractor-Distractor (N-N) Similarity:** Lower similarity prevents efficient rejection of distractors, also increasing search time.

1.20.1 Guided Search Theory (Jeremy Wolfe)

Jeremy Wolfe expanded on earlier theories with the Guided Search Theory, which integrates bottom-up and top-down influences during visual search.

Key Concepts:

1. Feature Maps and Activation Maps:
    - Like Treisman's theory, Wolfe proposed that the early visual system decomposes images into feature maps (e.g., color, orientation).
    - Each feature (e.g., red, blue, steep, shallow) is categorized separately within its respective map.
2. Bottom-Up vs. Top-Down Activation:
    - **Bottom-Up Activation:** Measures how different an element is from its neighbors (e.g., a red shape among green ones stands out).
    - **Top-Down Activation:** User-driven search that prioritizes specific feature values (e.g., searching for "blue").
    - These activations combine to form an activation map**,** where higher peaks attract attention first.
3. Search Mechanisms and Task Dependence:
    - **Pop-Out Search:** When a target is highly different from distractors, it stands out regardless of distractor count.
    - **Conjunction Search:** When a target shares features with distractors, search relies more on **top-down control** and is slower.
    - **Similarity Effects:**
        - **Low N-N similarity** increases bottom-up activation, making distractors more visually salient.
        - **High T-N similarity** reduces the target's distinctiveness, making it harder to find.

1.20.2 Postattentive Vision and Change Blindness

Jeremy Wolfe also explored postattentive vision, questioning what happens to objects once attention shifts away from them.

Key Findings:

1. No Accumulation of Detailed Visual Representations:
    - Contrary to intuition, visual memory does not retain a rich representation of previously attended objects**.**
    - When attention moves elsewhere, the perceptual representation remains unchanged—no additional detail is saved.
2. Search Efficiency and Memory Constraints:
    - Repeated exposure to a scene does not improve search performance unless objects are familiar and stored in long-term memory.
    - Conjunction targets (those defined by a combination of features, such as shape and color) cannot be **preattentively** detected and must be actively searched.
3. Experimental Evidence:
    - **Traditional Search:** A target is identified first via text, then searched for in an image.
    - **Postattentive Search:** The scene is previewed before the target is specified. However, results showed no improvement in search speed**,** indicating no benefit from prior exposure.

- **Repeated Search and Memory Search:**
  - Searching the same scene multiple times or using letters instead of shapes yielded no speed advantage.
  - Memorizing a set of letters before searching did not improve performance.

Implications for Visualization Design:

- Since novel displays are not stored in long-term memory**,** repeated exposure does not help users search more efficiently.
- Effective visualization tools should use preattentive processing methods to highlight important data without requiring memory-based searching**.**

1.21 Feature Hierarchy

1.21.1 Multidimensional Visualization and Data-Feature Mapping

One effective strategy for multidimensional visualization is to assign distinct visual features to different data attributes. This mapping allows multiple data values to be presented simultaneously within a single image. However, an essential requirement of this approach is to ensure that the data-feature mapping does not cause visual interference, which can obscure information within the display.

1.21.2 Visual Interference in Feature Mapping

Interactions between different visual features can lead to visual interference, where one feature masks another, making it harder to perceive the intended information. A simple example of this is the conjunction search, where identifying a target based on multiple features becomes difficult due to overlapping attributes.

Another critical form of interference results from a feature hierarchy in human vision. Research has shown that the visual system prioritizes certain features over others depending on the task.

1.21.3 Hue-on-Form Feature Hierarchy

- Boundary Detection: The human visual system favors color over shape when identifying boundaries.
- Effect of Background Variations: Variations in background color interfere with the ability to recognize individual shapes and the spatial patterns they form. However, when color is held constant, shape patterns become immediately visible.
- Asymmetry in Interference: While changes in hue affect shape recognition, variations in shape do not significantly impact the perception of color patterns.

Illustrative Examples:

- Figure 1.31: A horizontal hue boundary is easily detected when form remains constant, whereas a vertical hue boundary is still visible even when form varies in the background.
- Figure 1.32: A vertical form boundary is identifiable when hue is held constant, but a horizontal form boundary is obscured when hue varies randomly.

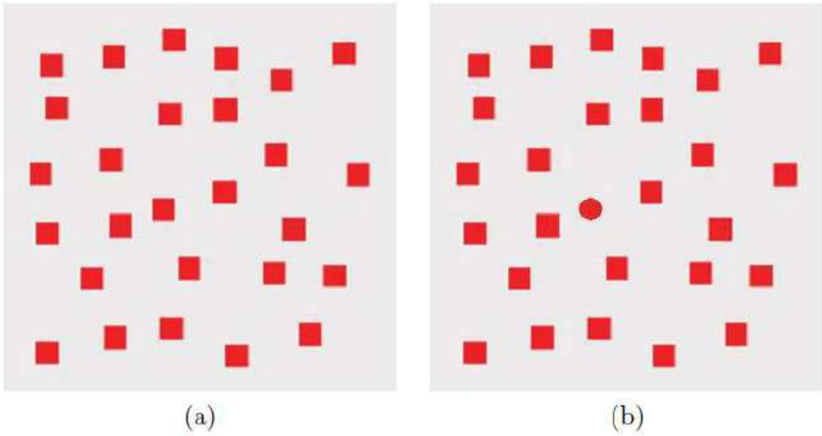


Figure 1.31 Hue-on-Form Feature Hierarchy Example: (a) A horizontal hue boundary is preattentively detected when form is constant; (b) a vertical hue boundary is detected when background form varies randomly.

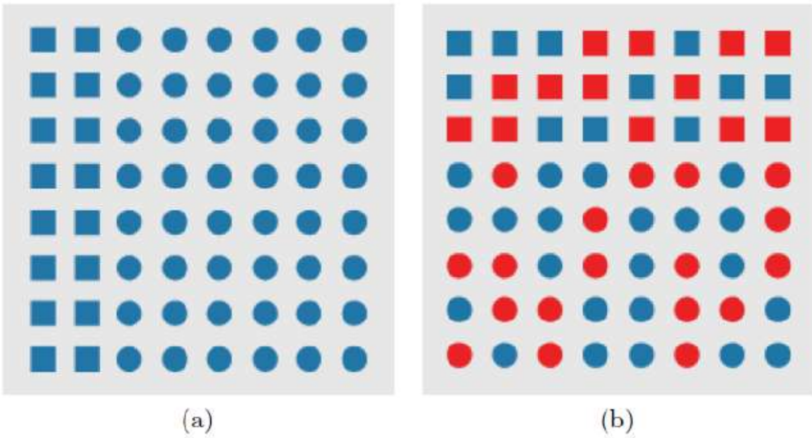


Figure 1.32 Hue-on-Form Feature Hierarchy Example: (a) A vertical form boundary is preattentively detected when hue is constant; (b) a horizontal form boundary is not detected preattentively when hue varies randomly.

1.21.4 Other Feature Hierarchies

- Luminance-on-Hue: Research by Callaghan suggests that luminance takes precedence over hue in perception.
- Hue-on-Texture Interference: Random variations in hue interfere with the identification of texture patterns, whereas variations in texture do not affect hue perception.

1.21.5 Implications for Data Visualization

The most important data attributes should be displayed using the most salient visual features to ensure clear perception. It is also critical to avoid using secondary attributes in ways that obscure primary data, ensuring that key insights remain visible and easily interpretable.

1.22 Change Blindness

1.22.1 Understanding Change Blindness

Recent research in visualization explores ways to create visually salient images based on principles of human perception. Unlike a camera that captures an image passively, human vision functions as a dynamic, ongoing construction process. Rather than forming a complete, static representation of a scene, vision selectively processes elements relevant to the viewer's current task.

1.22.2 Limitations of Human Vision

One of the most significant discoveries in visual perception is change blindness, the phenomenon where individuals fail to notice significant changes in a visual scene when there is an interruption, such as a blink, an eye movement, or a blank screen.

1.22.3 Factors Influencing Change Blindness

- Attentional Focus: Viewers are often blind to changes unless they directly focus on the altered object.
- Expectations and Goals: What we perceive is shaped by what we expect to see rather than simply the raw visual input.
- Preattentive vs. Attentive Processing: Some features can be detected without focused attention (preattentive), while others require active search and recognition.

1.22.4 Examples of Change Blindness Experiments

- Simons and Levin’s Movie Experiment: In a short film, the main actor was switched during a cut scene. Despite this drastic change, nearly two-thirds of viewers failed to notice the replacement. Most subjects retained details only from the first actor, highlighting the role of first impressions in perception.
- Real-World Experiment: An experimenter asking for directions had a basketball in hand. As a group of students walked by, one took the ball. Many pedestrians did not notice the missing object until they were directly asked about it, suggesting that details are often stored but not consciously accessed unless prompted.

1.22.5 Theories Explaining Change Blindness

1. Overwriting Hypothesis: New images replace previous ones in memory, with only abstracted information retained.
2. First Impression Hypothesis: Viewers form an initial impression of a scene and do not update it unless explicitly guided to do so.
3. Nothing Is Stored Hypothesis: Once perception is complete, the details are not stored, and the world itself serves as the memory store.
4. Everything Is Stored, But Not Compared Hypothesis: Details may be stored but remain inaccessible until triggered by a significant change.
5. Feature Combination Hypothesis: Elements from an initial view are combined with new details in a way that may mask significant changes unless they contradict the viewer’s expectations.

1.22.6 Implications for Visualization Design

- Directing Attention: Effective visualizations must guide the viewer's eye toward critical areas to prevent important data from being overlooked.
- Avoiding Sudden Changes: Changes in a visualization should be designed to be noticed, avoiding interruptions that can trigger change blindness.
- Leveraging Preattentive Features: Key information should be encoded in salient visual features (e.g., color, size, orientation) to improve detection and awareness.

By understanding the principles of feature hierarchy and change blindness, we can create better visual representations that align with human perceptual capabilities, ensuring clear and effective data communication.

1.23 Perception in Visualization

Visualization is significantly influenced by human perception, enabling effective data representation. The three perceptually motivated multidimensional visualizations are:

1. E-commerce Auctions Visualization:
    - X-axis: Time
    - Y-axis: Auction (each row represents a separate auction)
    - Towers: Bids by different agents
    - Color: Agent ID
    - Height: Bid price
    - Width: Bid quantity
2. CT Scan of an Abdominal Aortic Aneurism:
    - Yellow: Artery
    - Purple: Aneurism
    - Red: Metal tines in stents supporting the artery wall
3. Painter-like Weather Visualization over the Rocky Mountains:
    - Color: Temperature (dark blue for cold, bright pink for hot)
    - Orientation: Precipitation (tilting right for heavier rainfall)
    - Coverage: Wind speed (less background showing for stronger winds)
    - Size: Pressure (larger strokes for higher pressure)

The following sections explore perceptual properties used in visualization, including color, texture, motion, and memory issues.

1.23.1 Color

Color is a primary feature in visualization and can be enhanced through perceptual design.

Types of Color Scales:

- Simple scales: Rainbow spectrum, red-blue or red-green ramps, gray-red saturation scales.
- Perceptually optimized scales: Maintain consistent perceptual differences rather than relying on RGB distances.

Key Color Properties:

- Perceptual Balance: Uniform differences along the color scale.
- Distinguishability: Ensuring each color is equally recognizable.
- Flexibility: Selecting colors from a wide range.

Perceptual Color Models:

- CIE LUV, CIE Lab, and Munsell models improve perceived color differentiation.
- Nonlinear mappings (logarithmic or exponential) emphasize particular ranges.
- Healey's Color Selection Technique: Colors are selected along a perceptually balanced spiral in CIE LUV space.

1.23.2 Texture

Texture, like color, comprises multiple perceptual dimensions that enhance visualization.

Texture Properties:

- Regularity, directionality, contrast, size, and coarseness contribute to automatic segmentation and classification.
- Perceptual texture elements (pexels) can encode multiple attributes, varying in size, density, and regularity.
- Stick-figure icons and Gabor filters use texture for multidimensional data representation.

Application in Visualization:

- Surface perception: Anisotropic textures that follow principal curvature directions improve 3D object visualization.
- Texture mapping: Effective in medical and geographic visualizations.

1.23.3 Motion

Motion is a perceptually salient feature widely used in dynamic visualizations.

Key Motion Properties:

- Flicker Frequency: Distinguishable differences range from 2-5% in central vision and 100% in peripheral vision.
- Velocity Perception: Faster objects require less time to detect changes.
- Grouping by Motion: Coherent movement allows users to distinguish elements rapidly.

Applications in Visualization:

- Animated Flow Visualization: Glyphs and particles represent fluid dynamics.
- Motion-Based Data Grouping: Helps isolate data clusters.
- Notification Systems: Motion cues are more effective than color or shape changes.

1.23.4 Memory Issues

Memory plays a crucial role in visualization perception.

Types of Memory:

1. Sensory Memory:
    - High-capacity, rapid processing (<200 msec).
    - Used for preattentive perception (e.g., typing, playing instruments).
2. Short-Term Memory:
    - Limited capacity (5-9 chunks).
    - Effective visualization uses grouping and repetition.
3. Long-Term Memory:
    - Complex and limitless but slow.
    - Strengthened by mnemonics and association techniques.

Example of Memory in Reading:

Text Scrambling Experiment:

- "Aoccdrnig to a rscarhee at Cigdmabre Uinervtisy..." demonstrates how we process words holistically rather than letter-by-letter.
- However, excessive scrambling disrupts comprehension.

Memory aids in rapid scanning, pattern recognition, and cognitive chunking, crucial for effective visualization interpretation.

1.24 Introduction to Metrics in Human Perception

Humans have limitations in accurately perceiving distinct visual and auditory stimuli. Understanding these limitations is crucial in designing effective data visualizations. Researchers have studied how many levels of different stimuli can be distinguished without error, influencing the way data is presented visually and aurally.

Key questions explored in this field include:

- What graphical entities can be accurately measured by humans?
- How many distinct entities can be used in a visualization without confusion?
- With what level of accuracy do we perceive various graphical primitives?
- How do we combine primitives to recognize complex phenomena?
- How should color be used to present information?

1.24.1 Resource Model of Human Information Processing

Channel Capacity in Human Perception

George Miller (1956) introduced the concept of channel capacity from information theory, measuring how much information a human can accurately perceive and report.

- Each sensory stimulus (visual, auditory, taste, touch, smell) has a channel capacity measured in bits.
- If errors begin when more than 8 levels of a phenomenon are tested, the channel capacity is 3 bits.
- Channel capacity follows an asymptotic behavior—increasing levels beyond a certain point increases errors without adding useful information.

1.24.2 Absolute Judgment of 1D Stimuli

Several experiments have quantified human ability to judge absolute levels of stimuli:

Table 0.4 Channel Capacity for Different Stimulus Types

Stimulus Type	**Researcher**	**Distinct Levels Perceived**	**Channel Capacity (bits)**
Sound Pitches	Pollack	6 levels	2.5 bits
Sound Loudness	Gardner	5 levels	2.3 bits
Salinity (Taste)	Beebe-Center	4 levels	1.9 bits
Position on a Line	Hake/Gardner	10-15 levels	3.25 bits
Sizes of Squares	Eriksen/Hake	4-5 levels	2.2 bits
Color Hue	Eriksen	10 levels	3.1 bits
Touch (Vibration Location, Intensity, Duration)	Gelard	Various (4–7 levels)	-
Line Geometry (Length, Orientation, Curvature)	Pollack	2.6–3.3 bits	-

These findings indicate that humans can reliably perceive about 6-7 levels of a single-dimensional stimulus before errors occur.

1.24.3 Absolute Judgment of Multidimensional Stimuli

Researchers have examined whether combining multiple stimuli increases perception capacity. Unfortunately, results show that while more information is communicated, individual stimuli become harder to distinguish.

Table 0.5 Measured vs. Expected Channel Capacity for Combined Stimuli

Combined Stimuli	Researcher	Expected Capacity (bits)	Measured Capacity (bits)
Dot Position in a Square (2D Positioning)	Klemmer/Frick	6.5	4.6
Salinity and Sweetness (Taste Mixing)	Beebe-Center	3.8	2.3
Loudness and Pitch (Hearing)	Pollack	4.8	3.1
Hue and Saturation (Color Perception)	Halsey/Chapanis	5.3	3.6
Size, Brightness, and Hue (Visual Features)	Eriksen	7.6	4.1
Multiple Sound Features (6 Auditory Dimensions)	Pollack/Ficks	7.2	150 discernible combinations

1.24.4 Relative Judgment in Perception

Effectiveness of Graphical Attributes

William Cleveland and colleagues examined graphical perception by ranking different visualization techniques based on error rates. Their findings support bar charts and scatterplots as effective tools, while pie charts are less effective due to difficulty in judging area and angles.

Ordered Perceptual Accuracy (Least to Most Error):

1. Position along a common scale
2. Position along identical, nonaligned scales
3. Length
4. Angle/slope
5. Area
6. Volume
7. Color hue, saturation, density

1.24.5 Expanding Perceptual Capabilities

To enhance data visualization effectiveness, the following strategies can be employed:

- Relative judgment: Instead of requiring absolute measurement, use comparisons (e.g., adding grid lines and axis markers).
- Increasing dimensionality: Adding another stimulus can enhance capacity, though only up to a limit (~10 dimensions).
- Sequential presentation: Presenting information in steps may improve the ability to extract and remember details.

1.24.6 Relationship to Immediate Memory

Studies on short-term memory show an average recall capacity of 7 items. This suggests that while absolute judgment is limited by information bits, memory recall is limited by item count.

- Example: A person can remember 6 monosyllabic words as easily as 6 multisyllabic words.

1.24.7 Role of Recoding in Expanding Perception

Recoding is the process of restructuring information into fewer, more meaningful chunks.

- Example: Learning Morse code starts with dots and dashes, then progresses to letters, then words.
- Recoding is also observed in music, dance, and artificial intelligence (compilation in machine learning).
- Chunking improves memory and processing efficiency, helping humans retain and interpret more information effectively.

1.25 Cognition in Information Visualization

1.25.1 Understanding Human Cognition in Visualization

The human cognition framework for information visualization directly relates to cognitive processes. This framework enhances insight, reasoning, and understanding when analyzing visual data.

Key Leverage Points in Visual Analysis

Several leverage points can harness, influence, and measure human cognition in visualization:

1. **Exogenous and Endogenous Attention** – The ability to focus attention externally (stimulus-driven) or internally (goal-driven).
2. **Chunking** – Grouping information into meaningful units for better processing.
3. **Reasoning with Mental Models** – Using internal representations of real-world phenomena to interpret data.
4. **Analogical Reasoning** – Drawing comparisons between similar structures to understand new concepts.
5. **Implicit Learning** – Unconscious acquisition of knowledge through repeated exposure to patterns.

These elements contribute to how effectively users derive insights from visual representations.

Human Cognition and the Visualization Pipeline

A broader overview of human cognition in visualization is illustrated in Figure 1.33, which extends the visualization pipeline introduced previously. This expanded model integrates cognitive processes, emphasizing their role in information interpretation and decision-making.

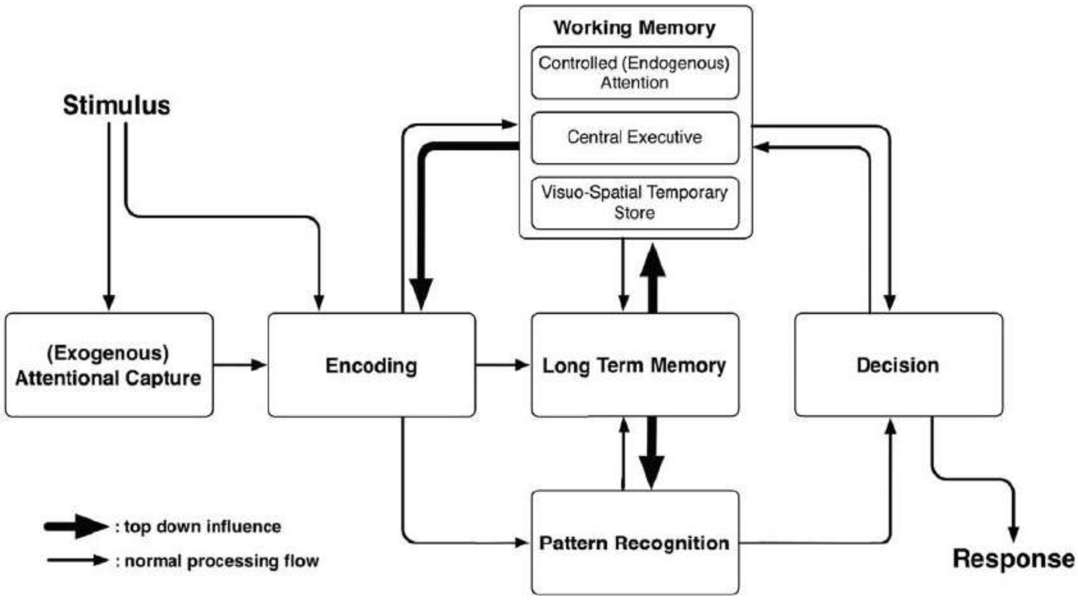


Figure 1.33 A dual-process framework underpins human cognition, guiding reasoning and decision-making in information visualization. The process flows left to right, from stimulus to decision, involving encoding, working memory, pattern recognition, long-term memory,

Summary

This unit introduced the fundamental principles of data visualization, its importance, history, and relationship with other fields. Additionally, it explored how human perception and cognition affect data interpretation. Understanding these foundations is essential for effective visual communication.

Keywords

- Data Visualization
- Scientific Visualization
- Information Visualization
- Perception in Visualization
- Cognitive Processing

Learning Outcomes

At the end of the unit, the learner would be able to:

- Demonstrate an understanding of data visualization and its significance.
- Explain the historical development and evolution of visualization techniques.
- Distinguish between scientific visualization and information visualization.
- Analyze the impact of perception and cognition on visualization.
- Apply appropriate visualization techniques to different types of data.

Exercises

Objective Questions

1. Which of the following is NOT a key goal of visualization?
   a) Exploration
   b) Hypothesis Confirmation
   c) Data Encryption
   d) Presentation
2. What is the primary purpose of the data-to-visual mapping step in the visualization pipeline?
   a) Structuring data for modification
   b) Converting data into graphical elements
   c) Removing unseen elements from a 3D model
   d) Applying algorithms to extract patterns
3. Which of the following best describes raw data?
   a) Data that has undergone preprocessing
   b) Data that has been generated by simulations
   c) Data that is untreated and unprocessed
   d) Data that has been visualized
4. A dependent variable is one that:
   a) Is not influenced by any other variables
   b) Is controlled or independent of changes
   c) Varies based on one or more independent variables
   d) Is always categorical
5. Which of the following is an example of an ordinal data type?
   a) Names of different countries
   b) RGB color codes
   c) Temperature readings in Celsius
   d) ZIP codes
6. In the context of data scales, which attribute allows data to be ordered but does not guarantee equal distances between values?
   a) Distance metric
   b) Ordering relation
   c) Absolute zero
   d) Normalization
7. A vector in a data record typically represents:
   a) A single scalar value
   b) A ranked categorical variable
   c) A composite data item with multiple values
   d) A random numerical value
8. Which part of the eye is responsible for adjusting its shape to focus light on the retina?
   a) Cornea
   b) Pupil
   c) Lens
   d) Iris
9. What type of photoreceptor in the retina is responsible for detecting color in bright light?
   a) Rods

b) Cones
c) Ganglion cells
d) Bipolar cells

10. Which eye movement allows smooth tracking of a moving object?
    a) Vergence movements
    b) Saccadic movements
    c) Smooth pursuit movements
    d) Saccadic masking

Short Questions

1. Why is visualization important in the context of large datasets?
2. How does the visualization pipeline differ from the computer graphics pipeline?
3. What is the role of perception in the design of effective visualizations?
4. What is the difference between raw data and derived data?
5. Define independent and dependent variables with an example.
6. What are the three attributes that define a variable's measure in terms of scale?

Long Questions

1. Discuss the evolution of visualization techniques from early maps to modern interactive visualizations.
2. Compare and contrast the visualization pipeline with the knowledge discovery pipeline.
3. Explain how an understanding of preattentive processing can improve the design of data visualizations.

- Explain the different types of data, including ordinal and nominal, with examples.
- Discuss the concept of scalars, vectors, and tensors in data representation. Provide examples of each.

1. • What is data preprocessing, and what are some common techniques used for handling missing values?

Collaborative Learning Tasks

1. **Group Discussion:** Discuss examples of visualizations that effectively communicate complex information and explain why they are successful.
2. **Design Challenge:** In small groups, design a visualization to represent a given dataset, considering the principles of perception and cognition.

Case Studies and Questions

1. John Snow's Cholera Map

- **Task:** Research John Snow's map of cholera deaths in London.
- **Questions:**

    - How did this visualization contribute to understanding and addressing the cholera outbreak?
    - What visual elements made it effective?

2. Medical Imaging

- **Scenario:** Consider an MRI dataset where raw values need to be segmented into categories like bone, muscle, and skin.
- **Questions:**
    - What preprocessing steps might be necessary before visualizing this data?
    - How would segmentation and interpolation techniques improve the visualization?

3. Change Blindness

- **Task:** Use the change blindness paradigm to show how viewers often miss significant changes in a scene if their attention is not directed to the changing elements.

Programming Exercises

1. Write a function to normalize a given set of numbers to a range between 0 and 1.
2. Implement a program to perform data aggregation by grouping data points based on proximity.

Study Tips

- **Active Reading:** Highlight key concepts and examples as you read.
- **Visual Thinking:** Practice translating data into visual forms and consider the best way to represent different types of information.
- **Real-World Examples:** Look for visualizations in your daily life and analyze their effectiveness.
- **Hands-On Practice:** Apply data preprocessing techniques using real datasets.
- **Visualize the Process:** Create diagrams to illustrate how each preprocessing step transforms the data.
- **Experimentation:** Test different visual properties and their impact on perception.
- **Iterative Design:** Use feedback to refine visualizations based on perceptual principles.

Self-Assessment Tasks

- **Concept Mapping:** Create a concept map linking key terms and concepts from the unit.
- **Reflection:** Write a short reflection on how your understanding of visualization has changed after completing this unit.
- **Technique Application:** Choose a dataset and describe which preprocessing techniques would be most appropriate and why.
- **Design Critique:** Select a visualization and write a detailed critique of its design based on perceptual principles.
- **Perceptual Test:** Create a simple perceptual test and administer it to a few people.

FURTHER READINGS

- Matthew O. Ward, Georges Grinstein, and Daniel Keim. *Interactive data visualization: foundations, techniques, and applications*. AK Peters/CRC Press, 2nd Edition, 2015.
- Stephen Few, *Now you see it: simple visualization techniques for quantitative analysis*. Analytics Press, 1st Edition, 2009.
- Ben Fry, *Visualizing data: Exploring and explaining data with the processing environment*. " O'Reilly Media, Inc.", 1st Edition, 2007.

NPTEL VIDEOS

- *Data Visualization | Introduction to Learning Analytics* - ***https://www.youtube.com/watch?v=UjYzNhBVIvY***
- *Introduction to Data Visualisation Analysis _ Part 1* - ***https://www.youtube.com/watch?v=qdnM8Fpvdqc***
- *Visualizing categorical and numerical data* - *https://youtu.be/axVBD1j-OnY*

References

The contents of this chapter have been prepared based on the reference book:

Matthew O. Ward, Georges Grinstein, and Daniel Keim. Interactive data visualization: foundations, techniques, and applications. AK Peters/CRC Press, 2nd Edition, 2015.

All the figures have been referred from this text book.

UNIT 2

VISUALIZATION FOUNDATIONS

2.1 Introduction

Visualization plays a crucial role in understanding and analyzing data by transforming raw information into meaningful graphical representations. It enhances human cognition, facilitates data interpretation, and supports decision-making. This unit explores fundamental concepts, historical perspectives, and techniques used in visualization, with a special focus on geospatial data representation.

2.2 Learning Objectives

- Understand the key components of the visualization process.
- Explore different techniques for visualizing one-, two-, and three-dimensional data.
- Analyze the impact of visual variables on perception and interpretation.
- Study historical developments and taxonomies in data visualization.
- Learn geospatial data visualization methods and their applications in Geographic Information Systems (GIS).

2.3 Overview

Data visualization is the process of converting raw data into graphical formats to enhance understanding. Effective visualization helps identify patterns, trends, and relationships within data. This unit covers the visualization process, the role of graphical symbols, historical advancements, classification frameworks, and techniques for representing various types of data, including spatial and dynamic data.

2.4 The Visualization Process in Detail

We have now covered the beginning and end of the visualization pipeline, focusing on data input and how perception helps interpret images. One fundamental visualization, the scatterplot, has been explored, but many other techniques exist. To study them effectively, we categorize methods into taxonomies. First, we review the visualization pipeline, followed by an overview of different visualization techniques.

Defining a visualization requires following key steps in the pipeline. This structured process allows for user interaction at any stage, with multiple visualizations often displayed simultaneously. The process includes several transformations that modify the data.

2.4.1 Data Preprocessing and Transformation

The first step ensures raw data is usable. It involves mapping data to fundamental types, handling missing values, and managing large datasets through sampling, filtering, aggregation, or partitioning.

2.4.2 Mapping for Visualizations

Once data is processed, it is mapped to visual elements like geometry, color, and sound. Poor visual representation can mislead interpretations, as seen in Figure 2.1(a), where a bar chart is incorrectly used. A scatterplot, as in Figure 2.1(b), offers better accuracy.

Expressiveness and effectiveness are crucial in visualization. Expressiveness measures how well a visualization presents only the necessary information, while effectiveness evaluates interpretation speed and cost.

2.4.3 Rendering Transformations

This final stage converts geometry into an image using a graphics API. It involves selecting viewing parameters, shading techniques for 3D graphics, and device-specific transformations. The choice of rendering tools impacts visualization quality.

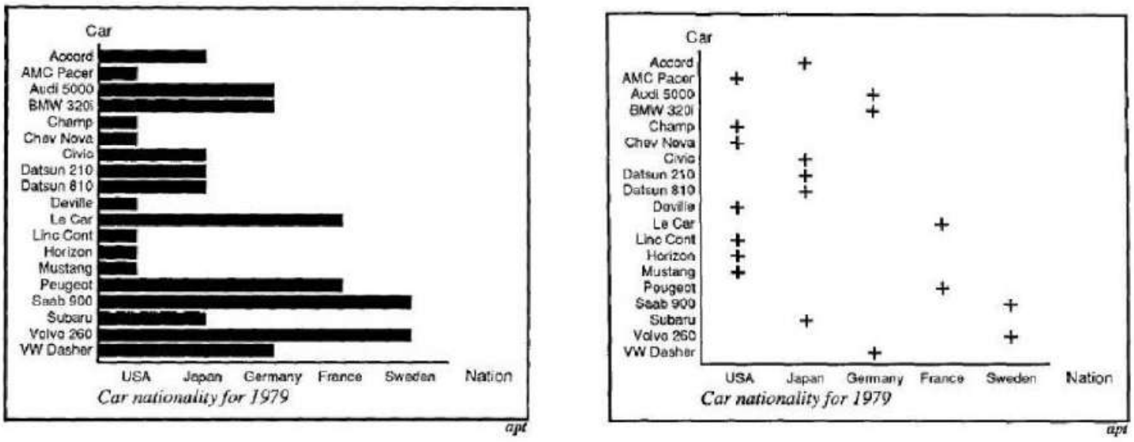
2.4.4 Measuring Visualization Quality

Metrics help assess visualization efficiency. Two key measures are:

- **Expressiveness**: The ratio of displayed information to intended information ($0 \leq M_{exp} \leq 1$). A perfect visualization has $M_{exp} = 1$.
- **Effectiveness**: The ability to interpret data quickly and accurately. Defined as $M_{eff} = 1 / (1 + interpret + render)$, where a higher value indicates better performance.

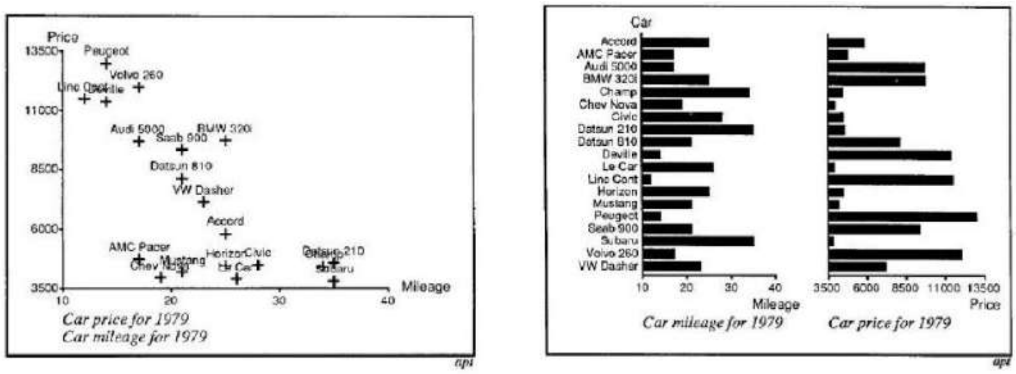
Figures 2.2(a) and 2.2(b) compare visualizations of car prices and mileage. While both display the same information, effectiveness varies. Bar charts (Figure 2.2(b)) allow easy identification of best mileage, but scatterplots (Figure 2.2(a)) provide better query-answering capabilities.

These principles ensure that visualizations effectively convey information while minimizing misinterpretation.



(a) (b)

Figure 2.1 Poor use of a bar chart. (b) Better use of a scatterplot.



(a) (b)

Figure 2.2 A scatterplot (a) using a plus symbol is effective for answering complex queries but can be slower for single-variable analysis. A bar chart (b) clearly presents cost and mileage but lacks flexibility for more detailed queries.

2.5 Semiology of Graphical Symbols

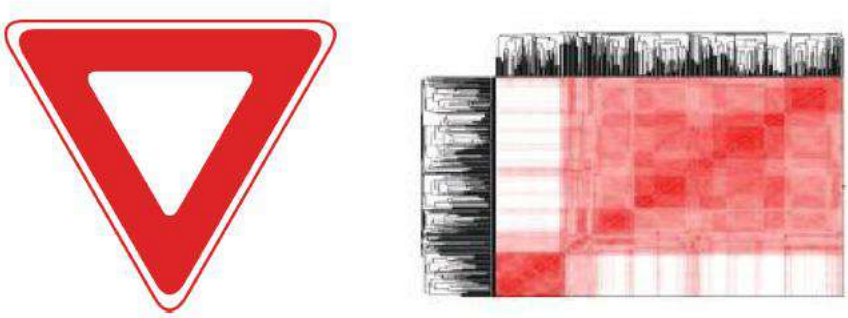
Semiology is the study of how graphical symbols convey meaning. Graphical symbols, such as arrows or labels, are essential components of visualizations. Understanding their design and perception helps in creating more effective visuals.

2.5.1 Symbols and Visualizations

Symbols can be instantly recognizable (Figure 2.3a) or require interpretation (Figure 2.3b). Effective symbols must be:

- **Easily identifiable** without excessive cognitive effort.
- **Mapped correctly** to data, ensuring patterns in the visualization match patterns in the dataset.
- **Ordered logically**, as incorrect ordering can mislead viewers.

For example, in a matrix representation (Figure 2.4), symbol size represents relationship strength, making patterns clear.



(a) (b)

Figure 2.3(a) Symbol with obvious meaning. (b) Representation with complex meaning.

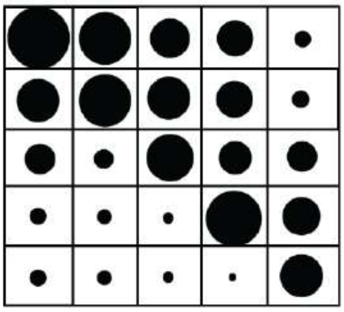


Figure 2.4 Matrix representation of a set of relationships between nodes in a graph. The size represents the strength of the relationship

2.5.2 Features of Graphics

Graphics typically function in three dimensions (x, y, and an additional variable like size or color for z). They must:

- Enable discovery of groups and orders in data.
- Maintain clarity when representing complex relationships.
- Follow perceptual rules ensuring the interpretability of visual elements.

Figure 2.5 illustrates how human vision identifies dominant features (e.g., a tree) rather than individual elements.

2.5.3 Rules of Effective Graphics

To create meaningful graphics, follow these principles:

1. Graphics should highlight patterns and relationships in data.
2. Three-dimensional construction (x, y, z) enhances interpretation.
3. Ordering and classification within the visualization improve clarity.
4. Overcomplicating a graphic with unnecessary dimensions reduces effectiveness.

5. Graphics must be easily readable and interpretable.

Analyzing a graphic follows a three-step process:

1. **Preattentive perception** – Identifying object groups.
2. **Cognitive processing** – Understanding relationships between groups.
3. **Detailed analysis** – Exploring unique cases or patterns.

Supporting analysis, such as clustering, can improve perception by guiding the viewer to meaningful insights.

Figure 2.5 The tree is recognized as the dominant feature of the image rather than its individual components.

2.6 The Eight Visual Variables

The use of graphics to convey information requires an understanding of graphic primitives and their attributes. These primitives, known as marks, form the basis for encoding data visually. Mapping different data values to distinct marks and their properties is fundamental in visualization. However, marks alone do not create meaningful representations, as overlapping marks can obscure information. Instead, the arrangement of marks within a display space plays a crucial role in revealing data patterns. Once the positioning and type of marks are determined, additional graphical attributes, such as size, color, and orientation, can be applied to encode further information.

There are eight primary visual variables: position, shape, size, brightness, color, orientation, texture, and motion. Adjusting these variables enhances a visualization's effectiveness. However, the final image must be interpreted by the human visual system, which has perceptual limitations.

2.6.1 Position

Position is the most fundamental visual variable, determining the placement of graphical elements in one-, two-, or three-dimensional space. Spatial arrangement plays a key role in how information is perceived. Effective visualization distributes graphical elements across the display space to maximize clarity and prevent overlaps. Poor positioning results in overlapping elements, reducing the information conveyed. Example visualizations are shown in Figure 2.6.

Choosing positioning variables helps analyze data distributions, detect trends, and recognize patterns. Perceptual principles like proximity and symmetry, reveal clusters within data. Scaling techniques, such as linear and logarithmic scaling, adjust variable ranges to highlight relationships.

Linear and logarithmic scales apply to individual data variables, while projections map multiple variables to a lower-dimensional space, commonly used in cartography and nonspatial data.

Beyond positioning, additional components like axes enhance interpretability. Axes provide reference points, displaying tick marks and labels that indicate data intervals. Axis titles describe the mapped data variable.

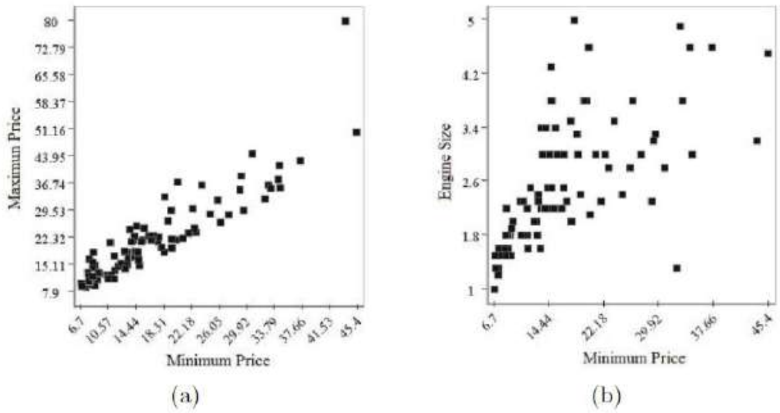


Figure 2.6 example visualizations: (a) a linear relationship between minimum and maximum car prices in 1993 models, and (b) a weak correlation between minimum price and engine size.

2.6.2 Mark

The second visual variable, the mark, refers to basic graphical elements like points, lines, areas, and volumes. These elements serve as building blocks for visualizations. Choosing visually distinct marks prevents confusion and enhances readability, especially in large datasets. Figure 2.7 shows how different shapes distinguish car types in a scatterplot. Marks should have similar areas and complexity to maintain visual balance.

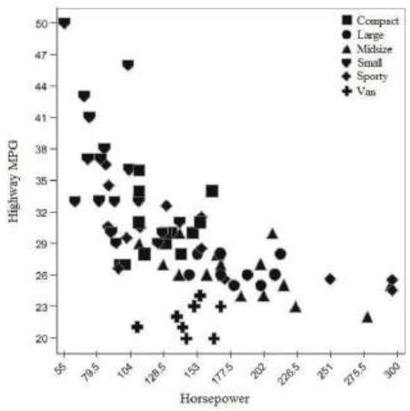


Figure 2.7 This figure illustrates shapes differentiating car types in a highway MPG vs. horsepower plot, revealing clusters and outliers.

2.6.3 Size (Length, Area, and Volume)

Size, the third visual variable, determines a mark's relative dimensions, as shown in Figure 2.8. It effectively represents interval and continuous data due to its ability to show gradual variations. However, when applied to categorical data, distinguishing between similar-sized marks becomes difficult, limiting its use for datasets with many categories.

Size works well for points, lines, and curves. However, when applied to larger graphical elements, precise comparisons become harder, making size differences more qualitative than quantitative.

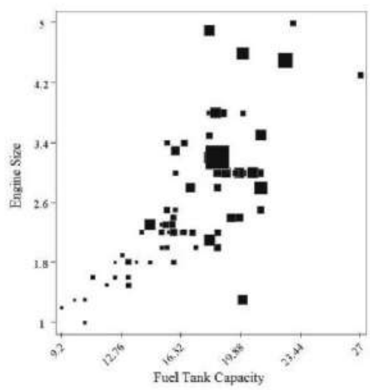


Figure 2.8 This figure visualizes the 1993 car models dataset, mapping engine size to fuel tank capacity, with size representing the maximum price.

2.6.4 Brightness

Brightness, or luminance, is the fourth visual variable, modifying marks to encode data. While it spans a full numerical range, human perception has limitations in distinguishing subtle differences. As a result, brightness is often used for relative comparisons in interval and continuous data or for categorical distinctions when applied using a simplified brightness scale.

Perceptually linear brightness scales improve clarity by ensuring noticeable differences between levels. Figure 2.9 provides an example of brightness used to encode data.

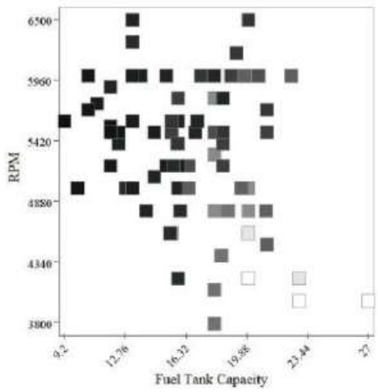


Figure 2.9 the 1993 car models dataset visualized using brightness to represent car width, where darker points indicate wider vehicles.

2.6.5 Color

Color, the fifth visual variable, is defined by hue and saturation. Hue represents the dominant wavelength, while saturation determines intensity relative to gray.

Effective color use in visualizations involves mapping data values to distinct colors. This typically requires defining color maps that assign value ranges to specific hues and saturation levels. Color maps work well for interval and continuous data by providing smooth value gradients.

For categorical or low-cardinality interval data, manually selecting distinct colors improves readability. Resources like www.colorbrewer2.org offer guidance on effective color selection. Figure 2.10 presents visualization with color.

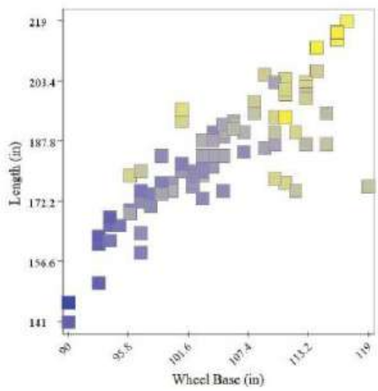


Figure 2.10 This figure visualizes the 1993 car models dataset, using color to represent car length. Blue indicates shorter cars, while yellow represents longer ones, plotted against wheelbase on the y-axis.

2.6.6 Orientation

Orientation refers to the direction in which a graphical mark is rotated in relation to data. It plays a key role in iconographic representations and relies on preattentive vision. However, orientation is only effective for marks with a distinct axis of symmetry. For instance, a triangle or arrow can clearly indicate direction, while a circle remains unchanged under rotation. Visualization techniques often use orientation to encode values by adjusting the angle of marks, enabling intuitive interpretation of directional trends. Some applications include mapping wind directions or depicting movement within datasets. For instance Figure 2.11 showcases an example visualization utilizing orientation to convey additional information.

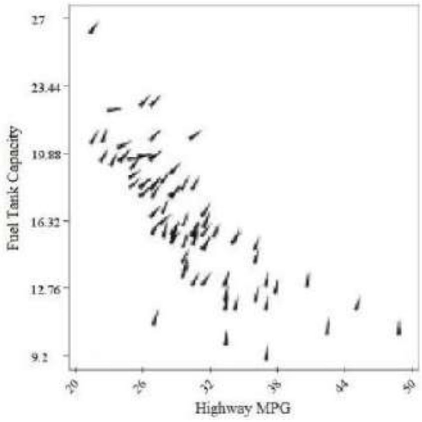


Figure 2.11 This figure visualizes the 1993 car models dataset, mapping highway MPG against fuel tank capacity, while mark orientation represents midrange price.

2.6.7 Texture

Texture is a combination of multiple visual variables, including marks, color, and orientation. It enhances graphical representation by differentiating elements through pattern variations. Dashed and dotted lines are common examples of texture in linear features, helping distinguish categories or trends. Textures can be applied to surfaces, regions, or polygons in both 2D and 3D visualizations. In 3D graphics, textures can simulate real-world surfaces by altering color variations and geometric properties. Effective use of texture enhances contrast and provides additional layers of information, making data more visually distinct. Figure 2.12 displays an example visualization incorporating texture to convey additional information.

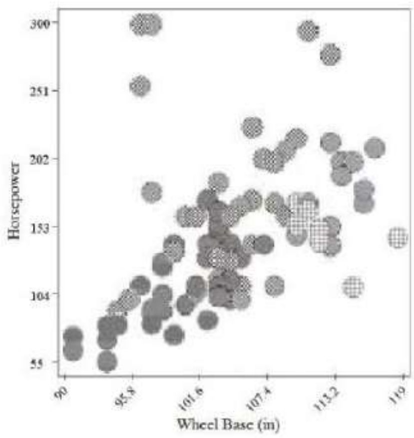


Figure 2.12 the 1993 car models dataset, plotting wheelbase against horsepower while using texture to differentiate car types.

2.6.8 Motion

Motion is a dynamic visual variable that conveys information through changes over time. It can modify other visual variables by altering position, brightness, or size. For example, moving elements can highlight outliers or indicate trends, while blinking effects can emphasize important data points. Directional movement can also provide insights into flow or transition patterns. Motion-based visualizations are commonly used in animations, simulations, and interactive data displays to enhance perception and understanding of changes in datasets.

2.6.9 Effects of Visual Variables

Different visual variables serve specific purposes in data visualization. They can be categorized based on how they structure and communicate information:

- **Selective variables** help distinguish groups visually (e.g., size, brightness, texture, color, and orientation).
- **Associative variables** ensure elements have equal visibility for categorical data (e.g., texture, color, orientation, and shape).
- **Ordinal variables** naturally order values for ranking or quantitative comparisons (e.g., texture, size, and brightness).
- **Proportional variables** represent relative differences in magnitude (e.g., size, orientation, and brightness).
- **Separating variables** improve visibility by distinguishing elements (e.g., texture, color, orientation, and shape).

By selecting the appropriate visual variables, data representations become clearer, improving comprehension and pattern recognition.

2.7 Historical Perspective

The field of visualization has seen numerous contributions aimed at formalizing its principles. Despite various advancements, a universally accepted language to describe visualizations remains elusive. Researchers have developed models to establish a structured foundation for visualization techniques. This section explores key historical contributions toward this goal.

2.7.1 Bertin (1967) - Semiology of Graphics

Jacques Bertin introduced *Sémiologie Graphique* (1967), a seminal work that defined the principles of graphical representations as a semiology—a science of sign systems. He distinguished between:

- **Content:** The information to encode.
- **Container:** The properties of the graphical system.

Bertin identified graphical primitives based on perceptual attributes:

- **Marks:** Points, lines, and areas.
- **Positional Variables:** Two planar dimensions.
- **Retinal Variables:** Size, value, texture, color, orientation, and shape.

These components formed the foundation for graphical visualization.

2.7.2 Mackinlay (1986) - APT (A Presentation Tool)

Mackinlay developed APT, an automated system for designing graphical presentations of relational data. He introduced **graphical sentences**, defined as collections of tuples containing objects and locations. His work built upon Bertin's graphical vocabulary while extending it to include:

- **Positional Variables:** 1D, 2D, and 3D representations.
- **Temporal Variables:** Animation.
- **Retinal Variables:** Color, shape, size, saturation, texture, and orientation.

Mackinlay proposed **composition algebra**, allowing primitive graphical languages to form complex presentations. He also defined two main principles:

- **Expressiveness:** The ability to represent the desired information.
- **Effectiveness:** Utilizing the display medium efficiently for human perception.

2.7.3 Bergeron and Grinstein (1989) - Visualization Reference Model

Bergeron and Grinstein introduced a conceptual visualization pipeline with four stages:

1. **Data Identification:** Defines the data source and structure.
2. **Model Transformation:** Projects the source data into a usable form.
3. **View Specification:** Maps transformed data into visual representations.
4. **Association:** Generates graphical outputs, including visual and auditory stimuli.

This model provided a structured approach to visualization design.

2.7.4 Wehrend and Lewis (1990) - Visualization Catalog

Wehrend and Lewis created a catalog of visualization techniques categorized by:

- **Objects:** Problem types grouped by target domains.
- **Operations:** Visualization techniques addressing specific goals.

Although an automated system was not developed, the catalog served as a foundation for future visualization knowledge bases.

2.7.5 Robertson (1990) - Natural Scene Paradigm

Robertson proposed the Natural Scene Paradigm, leveraging human perception of real-world environments for data visualization. He defined **natural scene views** as:

- **Spatial Surfaces:** 2D or 3D representations.
- **Visual Properties:** Surface height, material, density, phase, and wetness.

By aligning data attributes with perceptual priorities, this approach aimed to enhance intuitive data interpretation.

2.7.6 Roth (1991) - Visage and SAGE

Roth and his team developed **Visage**, a user-interface for information exploration, incorporating **SAGE**, an intelligent graphics presentation tool. Key contributions included:

- **Information-Centric Design:** Direct interaction with data elements.
- **Two Object Types:**
  - **Elements:** Data representations.
  - **Frames:** Containers for marks.

SAGE extended Mackinlay's work by introducing additional **composition operators**, such as merging display edges and network nodes for multi-relational graphs.

2.7.7 Casner (1991) - BOZ

Casner developed BOZ, an automated system for designing graphical presentations tailored to specific tasks. It replaced logical tasks with perceptual equivalents, allowing users to extract insights visually rather than through computation. The BOZ framework consisted of five components:

1. **Logical Task Description:** Expressing tasks in logical notation.
2. **Perceptual Operator Substitution:** Mapping logical tasks to perceptual operators.
3. **Perceptual Data Structuring:** Organizing visual elements for effective representation.
4. **Perceptual Operator Selection:** Choosing optimal visualization techniques.
5. **Rendering:** Displaying the final visualization.

BOZ's approach prioritized perceptual efficiency, enhancing the effectiveness of graphical presentations.

2.7.8 AutoVisual by Beshers and Feiner (1992)

AutoVisual is an automatic system for designing visualizations within the n-Vision visualization system. The n-Vision system implements the worlds-within-worlds technique, which recursively defines subspace coordinate systems. It is structured as a hierarchy of interactors composed of four key components:

- Encoding objects
- Encoding spaces
- Selections
- User interface

Key Features of AutoVisual

Visualization Task Specification:

AutoVisual allows users to define a set of visualization tasks through task operators and task selections. These operators enable users to explore, search, and compare data relations.

Graph Types and Display Generation:

The system uses predefined "ready-made" graph types within the worlds-within-worlds framework to generate visual representations.

Expressiveness and Effectiveness:

AutoVisual adapts Mackinlay's expressiveness and effectiveness criteria to support interactive visualizations.

Optimized Rendering for Interactivity:

The system considers rendering time for each encoding technique, excluding those that negatively impact interactive performance. This ensures smooth user interaction within the environment.

By incorporating these features, AutoVisual enhances visualization flexibility while maintaining real-time responsiveness.

2.7.9 Senay and Ignatius (1994) - VISTA

Senay and Ignatius extended Mackinlay's work with a focus on scientific data visualization. They developed VISTA (Visualization Tool Assistant), a knowledge-based system for visualization design. VISTA, like Bertin and Mackinlay's models, implements composition rules to create complex visualizations from simple techniques.

Visualization Pipeline and Knowledge Classification

They define the visualization pipeline with three key sub-processes:

- **Data Manipulation** - Prepares raw data for visualization.
- **Visualization Mapping** - Converts data into graphical representations.
- **Rendering** - Displays the visual representation.

VISTA classifies system knowledge into five categories:

- Data Characterization
- Visualization Vocabulary
- Primitive Visualization Techniques
- Composition Rules
- Visual Perception Rules

Visualization Marks and Composition Rules

VISTA defines two types of marks:

- **Simple Marks**: Points, lines, areas, and volumes.
- **Compound Marks**: Contour lines, glyphs, flow ribbons, and particles.

VISTA encodes data using three variation methods: **positional, temporal, and retinal.** Five composition rules are used to combine visualization techniques for displaying multidimensional data:

- **Mark Composition** - Pairs marks from different techniques.
- **Composition by Superimposition** - Overlays one mark set on another.
- **Composition by Union** - Merges marks using a set union.
- **Composition by Transparency** - Adjusts opacity to blend techniques.
- **Composition by Intersection** - Finds common elements and superimposes them.

2.7.10 Hibbard (1994) - Lattice Model

Hibbard introduced a lattice model for visualization that emphasizes transformations from data to displays. He considers visualization as a mathematical function mapping a set of data objects (U) to a set of display objects (V), denoted as $D: U \rightarrow V$.

Lattice Representation

- **Data Model (U)**: Represents data objects ordered by approximation to mathematical objects.
- **Display Model (V)**: Uses Bertin’s graphical primitives to order displays by information content.
- **Graphical Primitives**: Includes position, size, color, texture, orientation, and shape.

This model extends expressiveness criteria by ordering visual displays based on their information content.

2.7.11 Golovchinsky (1995) - AVE (Automatic Visualization Environment)

AVE is an automatic graphical presentation system that designs diagrams from basic components reflecting data relationships.

Visualization Process in AVE

The process includes four key stages:

- **Query** - Defines database views to retrieve relevant data.
- **Analysis** - Groups data tuples based on their relationships.
- **Constraint Network Generation** - Maps relationships to graphical structures.
- **Geometric Layout** - Positions objects to satisfy graphical constraints.

AVE generates tree and graph-based visualizations using rectangles for nodes and lines/arrows for relationships.

2.7.12 Card, Mackinlay, and Shneiderman (1999) - Spatial Substrate

Card, Mackinlay, and Shneiderman introduced a reference model for mapping data to visual form.

Key Transformations

1. Data Transformations - Convert raw data into structured information.

2. Visual Mappings - Assign graphical attributes to data.
3. View Transformations - Adjust the display for user interaction.

Spatial Substrate and Axes Types

The spatial substrate organizes information using axes:

- Unstructured (no axis)
- Nominal (categorical subdivisions)
- Ordinal (ordered categories)
- Quantitative (numerically measurable)

Techniques for Expanding Information Representation

- Composition of Axes - Orthogonal placement to create a 2D space.
- Alignment of Axes - Repetition of axes for comparison.
- Folding of Axes - Extends an axis in another dimension.
- Recursion on Axes - Subdivides space repeatedly.
- Overloading of Axes - Reuses display space.

2.7.13 Kamps (1999) - EAVE

EAVE (Extended Automatic Visualization Engine) expands AVE by incorporating **user preferences** alongside data characteristics and graphical knowledge.

Three-Phase Diagram Design Process

- **Data Classification** - Identifies relationship types.
- **Graphical Resource Allocation** - Assigns graphical representations.
- **Layout Phase** - Optimizes readability using procedural techniques.

EAVE applies formal concept analysis to model binary relationships and logical dependencies.

2.7.14 Wilkinson (1999) - Grammar of Graphics

Wilkinson introduced a formal grammar for statistical graphics, defining visualizations through compositional elements.

Key Stages of Graphic Creation

- Specification - Defines visual representation rules.
- Assembly - Combines graphical components.
- Display - Renders the final visualization.

Seven Core Specifications

Table 2.1 Wilkinson's seven specifications

Specification	Description
Data	Defines variables and datasets
Trans	Performs data transformations
Frame	Establishes spatial structure
Scale	Defines data scaling rules
Coord	Determines coordinate system
Graph	Represents data as graphical marks
Guide	Provides labels and annotations

Wilkinson's model emphasizes data-to-graph mappings and algebraic transformations to define visualization layouts.

2.7.15 Hoffman (2000) Table Visualizations

Hoffman's formal model for Table Visualizations was the first attempt to define a generalized space of data visualizations. It aimed to encapsulate primitive-graphic properties of individual techniques and infer their combinations within a geometric layout. The research focused on four visualization techniques:

- Survey plots
- Scatterplots
- RadViz
- Parallel coordinates

Dimensional Anchor Graphical Parameters

Hoffman introduced nine graphical parameters that define various aspects of these techniques as given below:

Parameter	Description
P1	Size of the scatterplot points
P2	Length of the perpendicular lines extending from individual anchor points in a scatterplot
P3	Length of the lines connecting scatterplot points associated with the same data point
P4	Width of the rectangle in a survey plot
P5	Length of the parallel coordinate lines
P6	Blocking factor for the parallel coordinate lines
P7	Size of the RadViz plot point

P8	Length of the spring lines extending from individual anchor points of a RadViz plot
P9	Zoom factor for the spring K constant

Visualization Generation

Table visualizations represent two-dimensional data tables using dimensional anchors—graphic curves that define mathematical axes for data dimensions. Arranging these anchors and their parameters within a geometric structure creates different visualization techniques.

Instead of distinguishing marks from retinal variables, Hoffman's model treats the vector of graphics parameters as input for visualization rendering. This approach enables generating both predefined and new visualizations by modifying parameter combinations. Using the grand tour technique, Hoffman demonstrated a range of visualizations applied to datasets exploring compound mechanisms of action.

2.8 Taxonomy

A classification system, or taxonomy, is a structured way of organizing and categorizing similar objects, often in a hierarchical manner. In the field of visualization, taxonomies help define relationships between data, visualization techniques, tasks, and interaction methods. This section provides an overview of various taxonomies from the literature. Many other researchers have proposed similar classifications, but this serves as a representative selection.

2.8.1 Keller and Keller (1994)

Visualization Goals Classification Keller and Keller, in their book *Visual Cues*, classify visualization techniques based on the type of data being analyzed and the associated user tasks. The data types they consider include:

- Scalar (or scalar field)
- Nominal
- Direction (or direction field)
- Shape
- Position
- Spatially extended region or object (SERO)

Additionally, they identify various tasks that users may perform in a visualization setting. Although some tasks are interconnected, their list serves as a foundational guide for designing visualizations. The identified tasks include:

- Identify: Determine key characteristics of an object.
- Locate: Determine an absolute or relative position.
- Distinguish: Recognize differences without needing full identification.
- Categorize: Group objects into predefined classes.
- Cluster: Organize similar objects together.
- Rank: Establish an order or hierarchy among objects.
- Compare: Identify similarities and differences.
- Associate: Establish relationships between objects.
- Correlate: Determine direct connections, such as causal relationships.

They categorized over 100 visualization techniques based on these classifications, making their approach useful for evaluating visualization tools and methods, despite its age.

2.8.2 Shneiderman (1996)

Data Type and Task Classification Shneiderman proposed a taxonomy focusing on data types and the tasks analysts perform when extracting insights. His data types include:

- One-dimensional linear
- Two-dimensional map
- Three-dimensional world
- Temporal
- Multidimensional
- Tree
- Network

For tasks, he emphasized user interactions that facilitate data analysis:

- **Overview:** Gain a broad view of the dataset, such as fisheye browsing in networks.
- **Zoom:** Focus on specific details by enlarging sections of a display.
- **Filter:** Remove unneeded elements to refine search results.
- **Details-on-demand:** Display more information for selected elements, such as through pop-ups.
- **Relate:** Show connections between different elements.
- **History:** Allow users to undo actions, replay steps, and refine their approach.
- **Extract:** Export data for use in other applications, such as saving to a file or sending via email.

Shneiderman suggested that effective visualization tools should support most or all of these tasks for an intuitive user experience.

2.8.3 Keim (2002) Information Visualization Classification

Keim proposed a taxonomy that categorizes visualization systems based on three dimensions: data types, visualization techniques, and interaction/distortion methods. His classification shares some similarities with Shneiderman’s approach but introduces a new classification for visualization techniques.

Data Type Classification

- One-dimensional data (e.g., temporal data, stock prices, text documents)
- Two-dimensional data (e.g., maps, charts, newspaper layouts)
- Multidimensional data (e.g., relational tables, spreadsheets)
- Text and hypertext (e.g., web documents, news articles)
- Hierarchies and graphs (e.g., network traffic, system dynamics models)
- Algorithm and software (e.g., execution traces, memory dumps)

Visualization Technique Classification

- Standard 2D/3D displays (e.g., bar charts, line graphs, scatter plots)
- Geometrically transformed displays (e.g., scatterplot matrices, projection pursuit techniques)
- Iconic displays (e.g., Chernoff faces, star icons, color icons)
- Dense pixel displays (e.g., recursive patterns, circle segments)
- Stacked displays (e.g., treemaps, cone trees, hierarchical axes)

Interaction and Distortion Technique Classification

- Dynamic projection (e.g., XGobi, XLispStat, ExplorN)
- Interactive filtering (e.g., dynamic queries, InfoCrystal, Polaris)
- Interactive zooming (e.g., TableLens, IVEE/Spotfire)
- Interactive distortion (e.g., fisheye views, hyperbolic visualizations)
- Interactive linking and brushing (e.g., scatterplots, bar charts, parallel coordinates)

2.9 Visualization Techniques for Spatial Data

Spatial data visualization, a key aspect of scientific visualization, involves data that includes spatial or spatiotemporal attributes. These attributes help in both constructing and interpreting visual representations, as they allow for an intuitive mapping of data properties to graphical elements. Since human vision constantly processes real-world imagery, visualizing spatial data on a screen aligns naturally with our perceptual abilities. However, digital visualization differs in several ways:

- Unlike real-world perception, digital displays are constrained by two-dimensional, discrete, and lower-resolution projections.
- Computer screens allow exploration of real or simulated phenomena at any scale.
- Visualization tools enable dynamic adjustments of contrast, lighting, resolution, and density.
- Users can navigate and explore spaces that may be inaccessible in real life.
- The ability to add or remove elements interactively helps in reducing clutter and enhancing context.

When creating a spatial data visualization, a key decision is how to map the spatial attributes of the data to screen coordinates. This process may involve transformations such as scaling, rotation, translation, shearing, or projection. After positioning the spatial elements, additional data attributes must be visually encoded using color, texture, size, shape, or other graphical properties.

The following content explores various techniques for visualizing spatial data, organized by data dimensionality. Lower-dimensional techniques are often used to represent projections or subsections of higher-dimensional data, and we will examine multiple examples of this approach.

2.10 One-Dimensional Data

One-dimensional spatial data is typically collected through sequential sampling along a path in space. A common example is a drill-hole sample, where measurements of mineral content and ore grade are recorded based on depth. This form of sampling is often referred to as a probe when used to explore higher-dimensional structures.

Visualization Methods for One-Dimensional Data

- **Line Graphs**:
    1. Spatial data is mapped to one axis (e.g., x-axis), while data values are mapped to the other axis (e.g., y-axis), forming a line graph.
    2. Example: Tracking temperature variations along a river's course.
- **Color Bars**:
    1. The data values determine the color of a mark or region along the spatial axis, creating a continuous representation of data intensity.
    2. Example: Visualizing the concentration of pollutants in an air quality dataset.
- **Scaling and Transformation**:
    1. Data values need to be scaled to fit within display constraints, ensuring effective visualization. This includes reserving space for axes, labels, and legends.
    2. Example: Mapping elevation changes along a hiking trail to a vertical axis while adjusting the scale for better readability.

Algorithm for Drawing a Line Graph

To generate a visualization, the following parameters must be defined:

- **Data Range:** `datamin` (minimum value) and `datamax` (maximum value) represent the range of data values.
- **Data Count:** `datacount` represents the total number of data points to be displayed.
- **Display Space:** The visualization occupies a rectangular display area, defined by `(xmin, ymin, xmax, ymax)`.

A basic algorithm for rendering a line graph using a function `drawLine(x1, y1, x2, y2)` is as follows:

```
Draw-Line-Graph(data, dataCount, xMin, xMax, yMin, yMax)
    dataMin ← computeMin(data, dataCount)
    dataMax ← computeMax(data, dataCount)
    xFrom ← xMin
    yFrom ← worldToScreenY(data[0], dataMin, dataMax, yMin, yMax)
    for i ← 1 to dataCount
    do  xTo ← worldToScreenX(i, dataCount, xMin, xMax)
        yTo ← worldToScreenY(data[i], dataMin, dataMax, yMin, yMax)
        drawLine(xFrom, yFrom, xTo, yTo)
        xFrom ← xTo
        yFrom ← yTo

worldToScreenX(index, dataCount, xMin, xMax)
return (xMin + index * (xMax − xMin)/dataCount)

worldToScreenY(value, dataMin, dataMax, yMin, yMax)
```

```
return (yMin + (value - dataMin) * (yMax - yMin)/(dataMax - dataMin))
```

Coordinate Transformations

To map the data values to screen coordinates, we use transformation functions that scale and shift values appropriately.

```
worldToScreenX(index, dataCount, xMin, xMax)
return (xMin + index * (xMax - xMin) / dataCount)
```

This function evenly spaces the points along the X-axis based on their index in the data set.

```
worldToScreenY(value, dataMin, dataMax, yMin, yMax)
return (yMin + (value - dataMin) * (yMax - yMin) / (dataMax - dataMin))
```

This function scales the data values to fit within the Y-axis range.

Mathematical Basis

The coordinate transformations are derived from the formula for mapping a value from one range to another:

$$(A_i - A_{min}) \Big/ (A_{max} - A_{min}) = (B_i - B_{min}) \Big/ (B_{max} - B_{min})$$

Where:

- A_i is the original value in the data range $[A_{min} \rightarrow A_{max}]$
- B_i is the mapped value in the screen range $[B_{min} \rightarrow B_{max}]$

This formula ensures that values are proportionally mapped between coordinate systems.

Generalized X Mapping for Arbitrary Ranges

If users want to specify a subset of the data for display, the `worldToScreenX` function can be modified to support an arbitrary index range:

```
worldToScreenX(index, indexMin, indexMax, xMin, xMax)
return (xMin + (index - indexMin) * (xMax - xMin) / (indexMax - indexMin))
```

This version allows mapping a selected range of indices rather than assuming all data starts from zero.

Handling Multivariate One-Dimensional Data

When multiple variables exist per data point, the following strategies can be used:

- **Juxtaposition**: Stacking multiple line graphs to compare different variables.
- **Superimposition**: Plotting multiple variables on the same graph using different colors, line styles, or markers.

- **Stacked Bar Graphs**: Each bar is divided into segments representing different variables, making it easier to see proportions.

2.11 Two-Dimensional Data

Two-dimensional data is visualized by mapping spatial attributes to display attributes. Common visualization methods include:

Image Representation

A single data value at each location is mapped to color, with interpolation used for intermediate pixels. Figure 2.13 illustrates this method.

Figure 2.13 An image from a tomographic data set.

Rubber Sheet Representation

Data points are mapped to heights in three dimensions, forming a triangulated surface. Figure 2.14 shows an example with elevation data.

Figure 2.14 A Rubber Sheet visualization of elevation data for the southeast U.S.

Cityscape Representation

Three-dimensional objects, like boxes, represent data attributes such as height and color. Figure 2.15 depicts air traffic density over the U.S.

Scatterplot Representation

Markers are placed at locations, with size, shape, or color controlled by data values. Unlike images, no interpolation is performed.

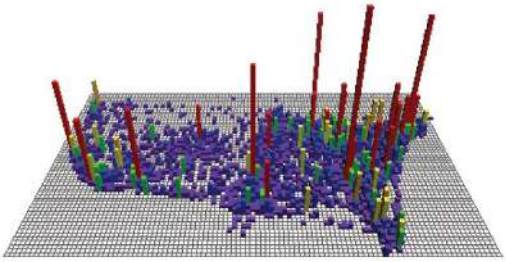


Figure 2.15 A cityscape showing the density of air traffic over the United States at a particular

Map Representation

Maps visualize linear, area, and point features. Roads are drawn as connected coordinates, lakes as closed contours, and landmarks as symbols. Figure 2.16 presents an example.

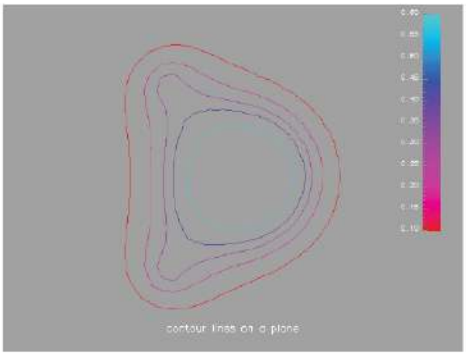


Figure 2.16 Contour lines from a sectional image of a hydrogen molecule.

Contour or Isovalue Map

Contours indicate boundaries where data transitions above or below a threshold. Multiple isovalues are displayed using color, line style, or labels. Figure 2.17 demonstrates this method.

2.11.2 Expanding Two-Dimensional Visualizations

For multivariate data, techniques include:

Juxtaposition

Stacking multiple 2D visualizations into a 3D representation provides an overview but can

Superimposition

Layering multiple data sets in one visualization, such as colored stacked blocks, can represent multiple variables. However, excessive layers can reduce clarity. Maps commonly use this method.

2.11.3 Probing Two-Dimensional Data

Subset visualizations provide additional insights:

Frequency Histograms

Histograms display data frequency within subranges. Proper partitioning is crucial to preserving key data features.

Row and Column Aggregations

Summarizing data along rows and columns identifies feature boundaries. Measures like average, max, or min can be visualized using color bars, line plots, or bar graphs.

Linear Probes

A one-dimensional probe through a dataset visualizes sampled values. Given two points, P_1 and P_2, the probe follows:

$P(t) = P_1 + t(P_2 - P_1)$, where $0 \leq t \leq 1.0$

Sampled coordinates are interpolated, and values are visualized as a one-dimensional dataset.

2.12 Three-Dimensional Data

Three-dimensional spatial data can be represented as discrete samples of a continuous phenomenon or as a structure best described using vertices, edges, and polygons. Many scientific and engineering visualizations combine these representations, such as airflow analysis around a wing or stress distribution in a mechanical part.

2.12.1 Visualizing Explicit Surfaces

An explicit surface is defined in two primary ways:

1. A list of three-dimensional vertices, edges (connections between vertices), and planar polygon patches defined by indices into the edge list.
2. A set of parametric equations defining x-, y-, and z-coordinates of surface points, along with a grid structure (triangular or rectilinear) for computing edges and patches.

Example - Representation of a Unit Cube:

- **Vertices:**
    - vertex[0] = (0., 0., 0.)

- vertex[1] = (0., 0., 1.)
- vertex[2] = (0., 1., 1.)
- vertex[3] = (0., 1., 0.)
- vertex[4] = (1., 0., 0.)
- vertex[5] = (1., 0., 1.)
- vertex[6] = (1., 1., 1.)
- vertex[7] = (1., 1., 0.)

- **Edges:**
  - edge[0] = (0, 1)
  - edge[1] = (1, 2)
  - edge[2] = (2, 3)
  - edge[3] = (3, 0)
  - edge[4] = (0, 4)
  - edge[5] = (1, 5)
  - edge[6] = (2, 6)
  - edge[7] = (3, 7)
  - edge[8] = (4, 5)
  - edge[9] = (5, 6)
  - edge[10] = (6, 7)
  - edge[11] = (7, 4)
- **Faces:**
  - face[0] = (0, 1, 2, 3)
  - face[1] = (8, 9, 10, 11)
  - face[2] = (0, 5, 8, 4)
  - face[3] = (1, 6, 9, 5)
  - face[4] = (2, 7, 10, 6)
  - face[5] = (3, 4, 11, 7)

Ensuring consistent edge orientation is crucial for correctly computing surface normals, which should point outward from the object's interior.

Example - Parametric Definition of a Unit Cylinder:

- **Top:** $y = 1.0$, $x = \cos\theta$, $z = \sin\theta$, $0 \leq \theta \leq 2\pi$
- **Bottom:** $y = 0.0$, $x = \cos\theta$, $z = \sin\theta$, $0 \leq \theta \leq 2\pi$
- **Sides:** $y = h$, $x = \cos\theta$, $z = \sin\theta$, $0 \leq \theta \leq 2\pi$, $0 \leq h \leq 1.0$

Adjusting the step size of θ controls the smoothness of the curved surface.

Different attributes in visualization may be associated with:

- **Vertices:** Representing temperature or stress at a joint.
- **Edges:** Representing strength in a chemical bond.
- **Faces:** Representing terrain type in a map region.

2.12.2 Visualizing Volume Data

Volume elements (voxels) are the three-dimensional equivalents of pixels and are used in data acquired via sensors or simulations. Visualization techniques include:

- **Slicing Techniques:** Extract a two-dimensional slice of the volume data using a cut plane.
- **Isosurface Techniques:** Generate surfaces corresponding to a specified data value.
- **Direct Volume Rendering:** Compute pixel values by casting rays through the volume.

For isosurfaces, such as those generated by the Marching Cubes algorithm, resampling is necessary to determine where the data matches the chosen isovalue, often occurring between data points. A similar process is used in contour map generation for two-dimensional data. In slicing operations, particularly when the slicing plane does not align with an axis, resampling ensures evenly spaced pixels. Additionally, interpolation plays a key role when dealing with nonuniformly spaced data points. Direct volume rendering also relies on resampling, especially when sampling data values along a ray. However, parallel projections along major axes generally do not require resampling. These processes, along with coordinate system transformations, are essential for visualizing spatial data.

2.12.3 Slicing Volume Data with Cut Planes

A common approach in three-dimensional data visualization is to reduce the dataset to a lower-dimensional subset. One technique for achieving this is through cut planes, where a plane slices through a data block, and the intersected data is mapped onto the plane for display.

A basic implementation of this technique restricts the cut plane's orientation so that its normal aligns with one of the data axes. Users can specify a row, column, or depth within the data block, and the corresponding slice is displayed using traditional two-dimensional visualization methods. Animating slice selection can enhance user understanding by showing relationships between adjacent slices.

For arbitrarily oriented cut planes, each voxel intersected by the plane may contribute to one or more pixels. Different resampling approaches include:

- Sampling data on a regular grid on the cut plane.
- Assigning the nearest voxel value to a cut plane pixel.
- Combining contributions from multiple voxels with weightings inversely proportional to their distance from the voxel center to the cut plane.

To define a cut plane, six parameters are required: three positional coordinates and three to specify the plane's normal. Since interactive manipulation of these parameters can be challenging, users often first set the orientation and then adjust depth using a slider.

Variations of this technique include:

- Nonplanar slices.
- Consecutive slices with varying orientations to remove data blocks.
- Stacked slices displayed simultaneously.
- Orthogonal slices displayed together.

2.12.4 Isosurface Extraction Using Marching Cubes

The Marching Cubes algorithm, developed by Lorensen and Cline in 1987, is a method for rendering isosurfaces in volumetric data. The technique involves defining a voxel (cube) by the values at its eight corners. When one or more corners have values below a user-defined isovalue while others are above, a portion of the isosurface must pass through the voxel. The algorithm determines which cube edges are intersected by the isosurface and constructs triangular patches to form the surface representation.

Algorithm Details

The algorithm consists of two major steps:

1. Determining how to construct surface sections for individual cubes.
2. Identifying intersections along cube edges to create triangular patches.

Since each cube corner can be classified as inside or outside the isosurface, 256 possible configurations exist. However, due to symmetry, only **15 unique configurations** need to be considered (Figure 2.17).

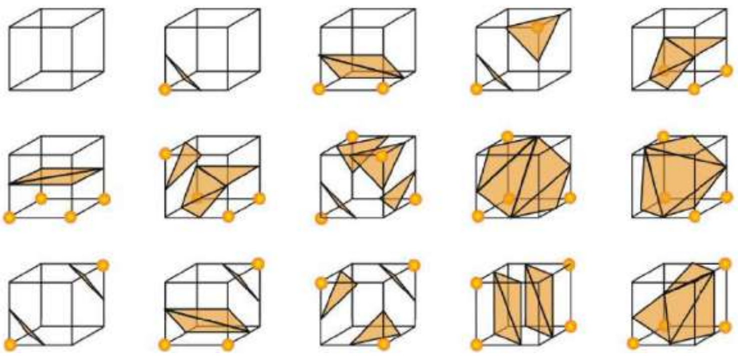


Figure 2.17 The 15 voxel configurations for Marching Cubes.

For instance:

- A cube with one corner below the isovalue forms a single triangle.
- A cube with two corners below the isovalue has three possible configurations depending on edge positions.
- A cube with three corners below the isovalue also has three configurations, including an "L" shape.
- A cube with four corners below the isovalue has seven unique configurations, varying by shared edges.

Each nontrivial configuration results in 1 to 4 triangles being added to the isosurface. The vertex positions can be interpolated along edges or approximated by placing them at edge midpoints, with interpolation yielding smoother surfaces.

The Marching Cubes Algorithm

```
MarchingCubes(IsoValue)
1. for EachCell
2.   if Isosurface Passes through Cell
3.     Classify each vertex as inside or outside
4.     Determine index of the 15 Cell Types
```

```
5.      Get edge list from table[index]
6.      Interpolate the edge location
7.      Create Small Polygons for Surface within Cell
```

Processing applies this method across the entire volume. Slabs of data consisting of two slices can be processed, with edge intersections shared between adjacent cubes. This edge-sharing mechanism improves efficiency and enhances surface continuity.

Problems and Alternatives

A key issue with Marching Cubes is the high memory requirement for storing surface data, as each boundary cube can generate up to four facets. Solutions include:

- Sharing vertices and edges.
- Merging coplanar patches into larger facets.
- Fitting parametric surfaces to boundary points (complex for intricate geometries).

Another challenge arises when voxel data has voids due to acquisition methods, requiring interpolation or exclusion strategies.

2.12.5 Direct Volume Visualization Techniques

Direct volume rendering generates images without constructing three-dimensional polygons. Instead, pixel values are computed individually, either by ray casting or voxel projection.

Basic Process

Rendering volumetric data involves transforming voxel positions into the viewing coordinate system, similar to traditional 3D graphics. The user defines:

- A view reference point.
- View direction.
- Image dimensions.
- Perspective projection parameters.

Once voxel positions are set, two main approaches can be used:

1. **Forward Mapping** – Projects each voxel onto the plane of projection, determining affected pixels.
2. **Inverse Mapping (Ray Casting)** – Sends rays through pixels into the volume, sampling values along the ray path.

Challenges in Forward and Inverse Mapping

In Forward Mapping, challenges include:

1. Resolving multiple voxels influencing the same pixel.
2. Handling pixels with no direct voxel mapping.
3. Addressing voxel projections landing between pixels.

In Inverse Mapping, issues involve:

1. Selecting an appropriate sampling rate.
2. Interpolating values at sampled points.
3. Combining values along the ray.

Solutions include resampling techniques, compositing methods, and voxel influence weighting.

Compositing in Volume Rendering

Compositing determines pixel values by integrating voxel contributions along the viewing ray. A voxel with color c_i and opacity o_i contributes:

$$\mathrm{I}(\mathrm{x},\mathrm{y}) = c_i * o_i * \prod_{j=0}^{i-1}(1 - o_j)$$

This considers accumulated transparency from the projection plane to the voxel. Variants include back-to-front compositing and early termination for efficiency.

Transfer Functions and Lighting

Assigning opacity and color values to voxels is known as classification, often guided by transfer functions that emphasize regions of interest. Users may set these functions based on data analysis or interactive exploration.

Lighting and shading are computed using gradient estimation since explicit surface normals are unavailable. The gradient direction serves as a proxy for surface orientation. Common gradient estimation techniques include:

- **Intermediate difference operator**: Uses voxel value differences from neighbors.
- **Central difference gradient estimator**: Uses values from both sides of the voxel.
- **3D Sobel operator**: Uses all 26 neighboring voxels for detailed estimation.

2.12.6 Implicit Surfaces

The traditional approach to modeling surfaces in computer graphics involves using parametric equations to define points on the surface. These points can then be connected to form polygonal meshes, making it easy to perform transformations and compute surface normals. An alternative method is the use of implicit representations, where a surface is defined as the zero contour of a function of two or three variables. Implicit representations are particularly useful in operations such as blending and metamorphosis. They are also widely used in data visualization to convey sparse data sets and fields of influence. By utilizing implicit surfaces, scattered data points can be merged into higher-level surfaces and solids.

Metaballs, or blobby objects, represent a specific type of implicit modeling technique. Each metaball can be viewed as a particle surrounded by a density field, where the density (or influence) decreases as the distance from the particle increases. A surface is formed by defining an isosurface through this density field, with the surface being closer to the particle at higher isosurface values. One of the key advantages of metaballs is their smooth blending, which occurs by summing the influence fields of multiple metaballs.

The mathematical formulation of metaballs involves defining an equation that specifies the influence of a particle at any given point. Blinn proposed an exponentially decaying field using a Gaussian function, where the influence is given by b^{-ar} where r is the distance from the particle to a field location. To improve computational efficiency, the squared distance was used instead to eliminate the need for square root calculations.

Wyvill et al. further refined the approach by defining a cubic polynomial that ensures the influence is 1.0 at r = 0.0 and 0.0 when r equals the radius of influence R. This function is given as:

$$C(r) = 2r^3/R^3 - 3r^2/R^2 + 1$$

To avoid square root calculations, the function was reformulated in terms of r^2 and R^2:

$$C(r) = ar^6/R^6 + br^4/R^4 + cr^2/R^2 + 1$$

where a = -0.444444, b = 1.888889, and c = -2.444444.

To visualize implicit surfaces, a density field is computed over a grid, taking contributions from surrounding data points. Once generated, rendering techniques such as the Marching Cubes algorithm or direct volume rendering can be applied. Examples of implicit surface renderings include smoothly merging spheres and complex structures composed of multiple metaballs with varying influences.

2.13 Dynamic Data

Flow visualization focuses on displaying the dynamic behavior of fluids such as liquids and gases. The study of flow dates back to Leonardo Da Vinci, who illustrated the movement of particles in fluids. Over time, advancements in laboratory techniques and photography improved the precision of flow visualization. More recently, computational fluid dynamics (CFD) has enabled scientists to simulate and analyze flow behavior under various conditions. The outcome is often a 2D or 3D grid of velocity vectors, which can be analyzed to identify turbulence, vortices, and other structural features.

Flow data can be categorized into different structures. A static field consists of a single, unchanging velocity field, whereas a time-varying field may involve fixed positions with changing vector values or both changing positions and vectors, such as rotating turbine blades or pitching airfoils. The latter category is referred to as unsteady flow.

2.13.1 Definitions

Several key definitions are relevant to flow visualization:

- **Pathline**: The trajectory of a particle released into a flow field, observed over multiple time instances.

- **Streakline**: The simultaneous positions of particles continuously released from a single location.
- **Timeline**: The positions of a batch of particles released at the same time, observed at an instant.
- **Streamline**: A line through the velocity field that is tangent to the velocity at every point.
- **Steady flow**: A flow field that does not change over time, where streaklines, pathlines, and streamlines coincide.
- **Particle advection**: The computation of particle motion through a flow field.
- **Vorticity**: The curl of the velocity field, representing the magnitude and direction of angular velocity at each point.

2.13.2 Mathematics of Particle Advection

Particle advection involves computing the motion of particles between time steps in a flow field. The process requires two time steps of flow and grid position data. The following pseudocode outlines the general procedure for computing streaklines:

```
1. Read the first two time steps of flow and grid data.
2. For each time step t:
     o  Compute new positions of particles.
     o  Store updated positions if they remain in the field.
     o  Release a new particle at the seed location.
     o  Read the next time step of flow data.
     o  If the grid is moving, read the next time step of grid data.
```

Particle advection typically employs a Runge-Kutta integration method, which uses a predictor-corrector algorithm. The integration process interpolates velocity values at arbitrary locations, allowing accurate motion calculations.

2.13.3 Visualization Techniques

Many methods for visualizing flows have been developed over the years. The simplest form of flow visualization is to display the velocity field data itself, either as displacement vectors using arrow glyphs (see Figure 2.18) or as magnitude scalar values using image, surface, or volume visualization techniques (mapping values to color, size, or position). The number and placement of the displayed components is a crucial factor in conveying the important features of the data; showing too many flow field components can lead to significant occlusion, while using too few elements raises the potential for missing features. While the simplest solution is to allow the user to interactively change the density of elements (often specified by a plane or vector in space, along with a sampling rate), recent research has focused on automated placement of the displayed flow field components by analyzing the data and identifying regions where potentially interesting flow is occurring.

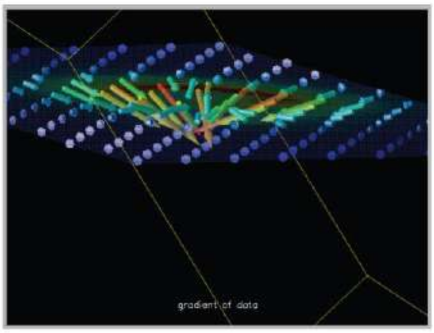


Figure 2.18 A storm cloud visualization containing glyphs showing wind direction and strength. (Image generated with OpenDX.)

The next most common technique is the generation of streamlines based on a static velocity field. The user selects seed locations (often along a line or at two-dimensional grid locations), and computes a path for each seed point through the field, maintaining a continuous tangent to the flow field. Besides using lines to indicate the streams, we can use planar or solid objects, such as ribbons and tubes. Other attributes of the field, such as magnitude or vorticity, can now be mapped to other attributes of the stream-ribbons or stream-tubes, such as color, size, or twist (see Figure 2.19).

Streaklines are often represented as a continuous stream of particles emanating from a discrete set of points and flowing through the field (see Figure 2.20). Individual points in a given trace (all particles coming from a particular location belong to the same trace) may be identified by color-coding to help distinguish related points that get separated when entering areas of high velocity.

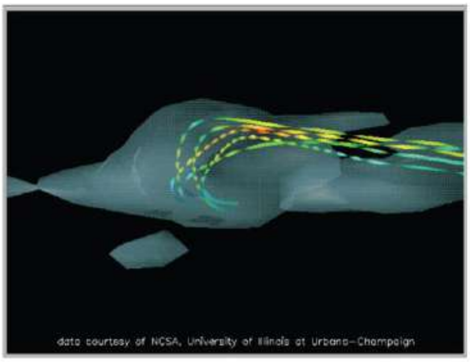


Figure 2.19 Flow data visualized using ribbons, with vorticity mapped to twist. (Image generated with OpenDX.)

Figure 2.20 An example of streaklines: Sparks from a Campfire. (Image Courtesy Wikimedia Commons.)

One more recent technique for flow visualization is called a streamball, which is based on the use of metaballs or blobby/soft objects. This is an implicit surface based on a field created by computing the influence of seed points on each location in the field. In effect, each seed point will influence a certain part of space, and locations in space can be influenced by multiple points. What this means is that if we use each particle along a streamline or streakline to influence a spherical segment of the field, locations under the influence of multiple particles will have continuous, smooth transitions from one particle to the other. This can form both tubes and surfaces, depending on how close the particles and streamlines are to each other.

2.13.4 Line Integral Convolution

An interesting approach to vector field visualization was developed by Cabral and Leedom in 1993. The method, called line integral convolution (LIC), uses a random field and a vector field with the same height and width to generate a dense display of flow information (see Figure 2.21). Basically, every pixel in the resulting image is a weighted average of a sequence of adjacent pixels in the random field along a linear path centered on the given pixel and following the streamline going through the pixel.

Figure 2.21 The texture field on the left is combined with the vector field in the middle to

Figure 5.14. The texture field on the left is combined with the vector field in the middle to create the line integral convolution on the right. (Images generated with LicFactory.)

More formally, the process follows these steps:

```
1. Given a vector field V and a texture image T.
2. Read and store V and T, computing the maximum vector magnitude.
3. For each point (i, j) in the result image:
  a. Compute, via interpolation, the flow direction and magnitude at this point.
  b. Advect L points forward and backward, stopping at boundaries. Step size
  depends on normalized vector magnitude.
  c. Given 2L + 1 positions, extract corresponding values from T.
  d. Convolve a list of texture points, normalizing based on the shape of the
  convolution kernel to maintain good range of color/intensity values.
```

This process continues until the length of the streamline centered on the selected point is equal to 2L. Again, the texture points are summed and normalized. They may also be weighted using a Gaussian filter to allow a tapering effect at the ends of the filter kernel. The resulting convolution, while much more computationally expensive than the simple straight-line version, is much more accurate for flow lines with a small radius of curvature. An interesting effect can be obtained by animating this computation.

2.14 Combining Techniques

Many effective visualizations are combinations of two or more techniques. Each technique has strengths and weaknesses, and by minimizing occlusion, a combined visualization can leverage multiple strengths. Many problems require simultaneous analysis of multiple data sets for informed results. For example, weather forecasting integrates air and surface temperature, wind speed, and humidity. This section explores visualizations that combine methods, emphasizing design and interpretation considerations.

2.14.1 Slice Plus Isosurface

In Figure 2.22, an isosurface from a medical data set is combined with an orthogonal slice of the same data. The isosurface is mapped to one color, while the slice uses a separate color ramp. The isosurface conveys surface structure, which is difficult to obtain from volume slicing alone. However, it only provides information on a single value within the volume, without indicating the distribution of other values or the gradient. The slice provides detailed two-dimensional information, especially with an appropriate color assignment. It highlights uniform regions and areas of significant change. Additionally, nested regions of a particular value range can be seen in the slice, whereas the isosurface generally only displays the outermost surface.

Key considerations in designing this visualization:

- Rapid and intuitive modification of the isosurface value is essential.
- The slice's position and orientation should be easily adjustable along three axes, with an option for animation.
- Camera position and orientation control allow better visualization.
- Careful color assignment is crucial to distinguish between the slice and isosurface and prevent misinterpretation.
- The user should be able to hide either visualization component or adjust opacity using a slider.

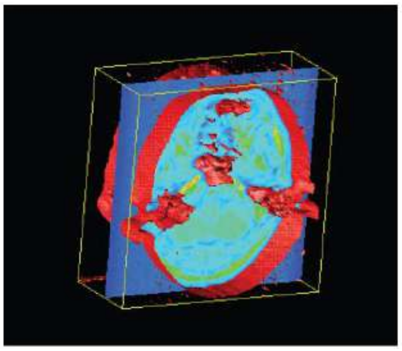


Figure 2.22 A medical volume dataset visualized using an isosurface and a 2D slice. (Image created with OpenDX)

2.14.2 Isosurface Plus Glyphs

Isosurfaces effectively depict three-dimensional surfaces but do not convey other data aspects. Glyphs, such as arrow glyphs, illustrate magnitude and direction of change within a dataset, either as gradients in static data or flow in dynamic data. Glyphs can be positioned near the isosurface or controlled separately.

In Figure 2.23, a storm cloud visualization includes an isosurface representing water density and arrow glyphs indicating wind direction and magnitude. A cut plane further details water density. By adjusting glyph positions, calm and turbulent regions can be identified. Modifying the isovalue changes the cloud's shape and position, allowing wind field interactions to be explored.

Design considerations include:

- Interactive control over isosurface value, glyph base position, and viewing position.
- Ability to vary glyph density.
- Scaling glyph sizes.
- Coloring glyphs to provide additional information or redundancy.
- Computing glyph base positions based on regions of interest in vector or water density fields.

Enhancements could include using a cut plane or rubber sheet to display variability in fields while maintaining context with a translucent isosurface. Streamlines could further aid flow field understanding.

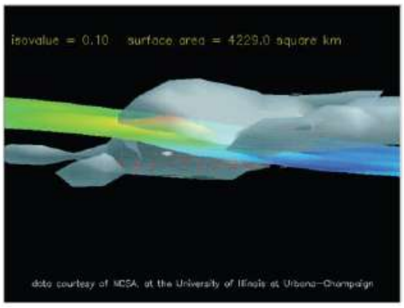


Figure 2.23 A storm cloud visualization featuring an isosurface and a cut plane representing water density, along with glyphs indicating wind direction and strength. (Image created with OpenDX.)

2.14.3 Rubber Sheet Plus Contour Lines and Color

Redundant data mappings improve visualization effectiveness. A rubber sheet conveys a two-dimensional field as a height field, revealing peaks and valleys in data. This creates a virtual landscape that leverages users' intuitive understanding of topography (Figure 2.14). Color mapping to elevation (Figure 2.24) facilitates identification of widely separated regions of similar height.

Each addition to the visualization enhances data understanding while maintaining the importance of viewing data in multiple ways. Selectively hiding and showing components allows users to incrementally refine their mental model of the dataset's contents.

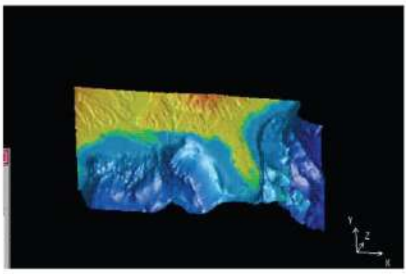


Figure 2.24 The same dataset as in Figure 2.14, with color representing elevation.

2.15 Techniques for Visualizing Geospatial Data

Geospatial data is distinct from other data types because it represents real-world locations of objects or phenomena. This data is widely used in areas such as credit card transactions, telecommunications, environmental monitoring, and demographic studies. This chapter explores geospatial visualization, often called geovisualization, and introduces essential concepts like map projections and visualization techniques for different spatial data types. While geographic information systems (GIS) and cartography cover extensive methods, this chapter provides a foundational understanding from a visualization perspective.

2.16 Representing Spatial Data

Spatial datasets are created by collecting samples of real-world phenomena across two dimensions, often consisting of discrete observations of continuous variables. Various fields rely on spatial analysis, such as global climate studies (e.g., temperature, precipitation, wind speed), environmental monitoring (e.g., pollution levels, CO2 emissions), social studies (e.g., unemployment rates, education), customer behavior analysis, telecommunications, financial activities, and crime mapping.

Because spatial data includes location attributes, mapping them onto a two-dimensional plane is the most intuitive visualization method.

Maps simplify real-world data by representing it as points, lines, and areas. Additional characteristics such as size, shape, texture, color, and orientation enhance the information conveyed. The U.S. Geological Survey (USGS) defines maps as structured representations of points, lines, and areas based on a coordinate system and nonspatial attributes. MacEachren describes geovisualization as a technique that leverages human visual perception to reveal spatial relationships effectively.

2.16.1 Types of Spatial Phenomena

Spatial data can be classified based on its spatial properties:

- **Point-Based Phenomena:** These have no spatial extent and are represented using longitude and latitude coordinates along with descriptive attributes. Examples include cities, oil wells, and individual buildings.
- **Line-Based Phenomena:** These are one-dimensional, possessing length but no width. They consist of a series of longitude and latitude coordinates and represent features such as roads, country borders, and communication networks.

- **Area-Based Phenomena:** These two-dimensional phenomena include both length and width and are represented by enclosed coordinates. Examples include lakes, parks, and administrative regions.
- **Surface-Based Phenomena:** These are sometimes considered "two-and-a-half-dimensional" as they include length, width, and elevation. They are represented by longitude, latitude, and height coordinates.

2.16.2 Map Types

Maps can be classified based on data properties (qualitative vs. quantitative, discrete vs. continuous) and graphical representation (points, lines, surfaces, volumes). Common types include:

- **Symbol Maps** – Display nominal point data.
- **Dot Maps** – Represent ordinal point data.
- **Land Use Maps** – Illustrate nominal area data.
- **Choropleth Maps** – Represent ordinal area data.
- **Line Diagrams** – Show nominal or ordinal line data.
- **Isoline Maps** – Depict ordinal surface data.
- **Surface Maps** – Represent ordinal volume data.

A dataset can be visualized through different map types. For example, aggregating point data into areas can transform a dot map into a choropleth map. Similarly, density surfaces can be derived from dot maps and visualized as isoline or surface maps. Cartograms adjust the size of regions based on the number of points within them.

Interactive geovisualization allows users to explore spatial data dynamically. Unlike traditional cartography, users can manipulate map displays and classifications interactively. Tools like CommonGIS enable users to perform queries, link maps, and integrate spatial representations with statistical charts (e.g., bar graphs, line graphs) and multidimensional visualizations such as parallel coordinates (see Figure 2.25).

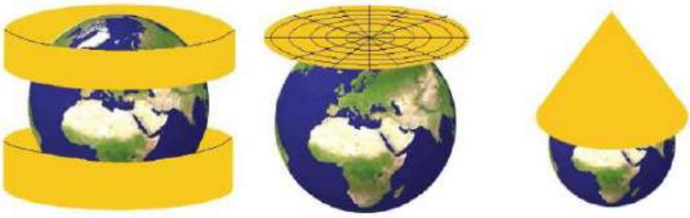


Figure 2.25 Cylinder, plane, and cone projections

2.17 Understanding Map Projections

Map projections convert three-dimensional locations on the globe into two-dimensional representations. A projection function transforms longitude (λ) and latitude (ϕ) into screen coordinates (x, y):

$$\Pi : (\lambda, \phi) \rightarrow (x, y)$$

Longitude ranges from -180° to 180° (negative for west, positive for east), while latitude spans from -90° to 90° (negative for south, positive for north). Various projection types are designed for specific visualization needs:

- **Conformal Projections** – Preserve angles and shapes locally but distort areas.
- **Equal-Area Projections** – Maintain accurate area proportions while distorting shape and angles.
- **Equidistant Projections** – Preserve distances from a reference point or line.
- **Gnomonic Projections** – Represent great circles as straight lines to show the shortest paths.
- **Azimuthal Projections** – Maintain accurate directions from a central point and exhibit radial symmetry.
- **Retroazimuthal Projections** – Ensure correct directional relationships between locations and a fixed reference point.

Projections can also be categorized based on their projection surfaces.

- **Cylindrical Projections:** Project the globe onto a cylinder, often preserving angles.
- **Plane Projections:** Azimuthal projections that map the sphere onto a flat plane.
- **Conic Projections:** Use a conical surface to preserve distances along meridians.

2.17.1 Commonly Used Map Projections

1. **Equirectangular Projection** (see Figure 2.26): A basic cylindrical projection with evenly spaced meridians and parallels. It does not preserve angles or areas precisely.
    - Formula: $x = \lambda, \quad y = \phi$
2. **Lambert Cylindrical Projection** (see Figure 2.27): An equal-area cylindrical projection suitable for world maps.
    - Formula: $x = (\lambda - \lambda_0) * cos\phi_0, \quad y = sin\phi \, / \, cos\phi_0$
3. **Hammer-Aitoff Projection** (see Figure 2.28): A modified azimuthal equal-area projection with an elliptical appearance.
    - Formula: $x = (2\sqrt{2}\, cos\phi\, sin(\lambda/2)) \, / \, (1 + cos\phi\, cos(\lambda/2))$

$$y = (\sqrt{2}\, sin\phi) \, / \, (1 + cos\phi\, cos(\lambda/2))^{1/2}$$

4. **Mollweide Projection** (see Figure 2.29): A pseudocylindrical equal-area projection representing the globe as an ellipse.
    - Formula: $x = (2\sqrt{2}(\lambda - \lambda_0)\, cos\theta) \, / \, \pi$

$$y = \sqrt{2} sin\theta, \qquad where\ 2\theta + sin(2\theta) = \pi\, sin\phi$$

5. **Cosinusoidal Projection** (see Figure 2.30): A simple equal-area projection with strong local accuracy.
    - Formula: $x = (\lambda - \lambda^0) * cos\phi, \qquad y = \phi$
6. **Albers Equal-Area Conic Projection** (see Figure 2.31): Uses two standard parallels to minimize distortions.

$$n = \frac{\cos\beta_1 + \cos\beta_2}{2}$$

$$p = \sqrt{\frac{4}{n} * \sin\frac{\frac{\pi}{2} - \varphi}{2} + \frac{4}{n^2} * \left(\sin\frac{\beta_1}{2}\right)^2 * \left(\sin\frac{\beta_2}{2}\right)^2}$$

- Formula: $x = p / sin(n * \lambda), y = -p / cos(n * \lambda)$

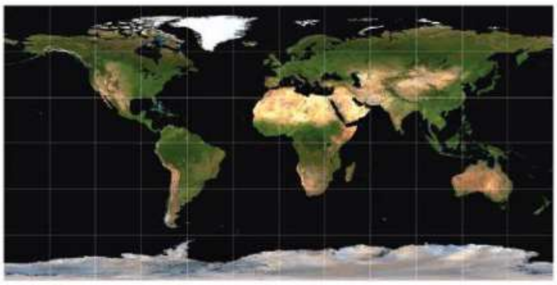


Figure 2.26 Equirectangular projection

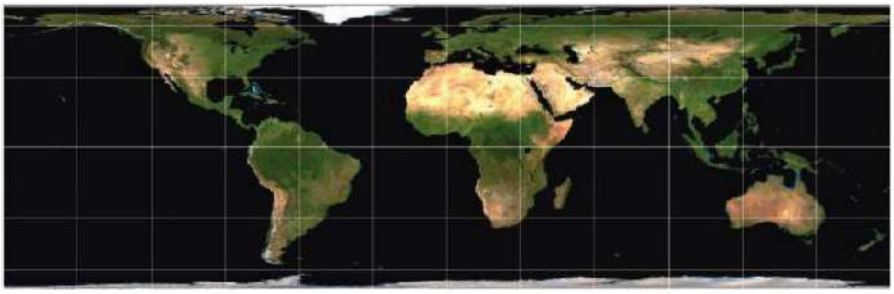


Figure 2.27 Lambert cylindrical projection

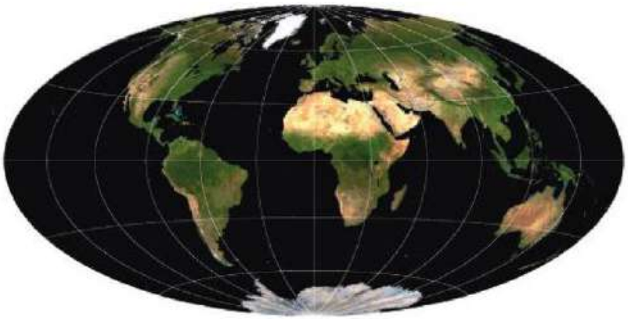


Figure 2.28 Hammer-Aitoff projection

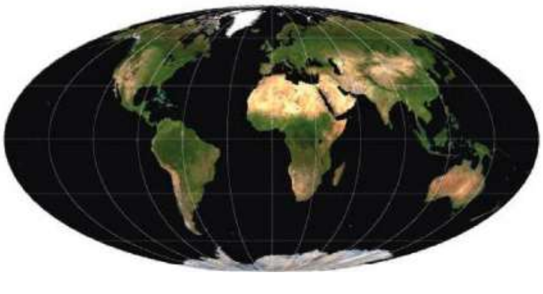


Figure 2.29 Mollweide projection

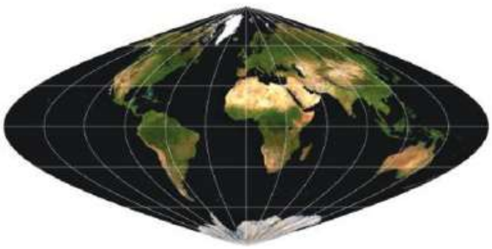


Figure 2.30 Cosinusoidal projection

Figure 2.31 Albers equal-area conic projection

2.18 Visual Variables in Spatial Data Representation

Maps use different visual variables to communicate spatial information (see Figure 2.32):

- **Size** – Adjusting symbol sizes or line widths.
- **Shape** – Differentiating symbols or area patterns.
- **Brightness** – Using varying brightness levels.
- **Color** – Encoding data with different colors.
- **Orientation** – Modifying directionality of symbols or patterns.
- **Spacing (Texture)** – Adjusting element density.
- **Perspective Height** – Employing 3D perspectives.
- **Arrangement** – Organizing patterns within areas.

Effective map design is crucial for accurate representation. Decisions related to class divisions, data normalization, and aggregation significantly influence visualization outcomes.

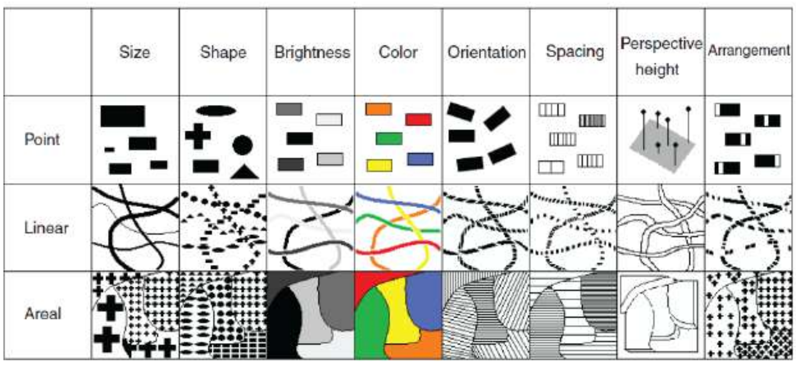


Figure 2.32 Visual variables for spatial data

2.19 Visualization of Point Data

Point data are discrete but may describe continuous phenomena like temperature measurements. Designers must decide how to display data: discrete vs. continuous and smooth vs. abrupt.

2.19.1 Dot Maps

Dot maps visualize point phenomena by placing symbols at specific locations. Quantitative parameters can be mapped to symbol size or color. Circles are common, but squares or bars may also be used. Correct symbol scaling is crucial, as perceived size can be distorted. Overplotting occurs in dense areas, making data difficult to interpret (Figure 2.33). Solutions include aggregating data in 2.5D visualizations, such as In3D and ArcView, or displaying data as bars in MineSet and Swift 3D. However, occlusion in 3D can obscure data.

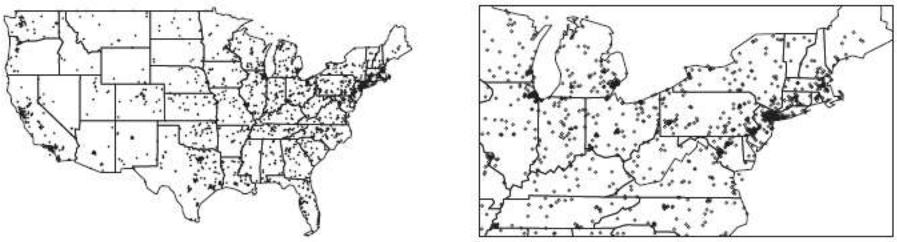


Figure 2.33 USA dot map: every circle represents the spatial location of an event. Even in the

2.19.2 PixelMaps

PixelMaps reposition overlapping pixels instead of aggregating data. A quadtree-like structure partitions the dataset into subregions, ensuring data points fit within available space. A pixel placement algorithm positions overlapping points in nearby empty locations (Figure 2.34). However, in dense areas, repositioning depends on data order.

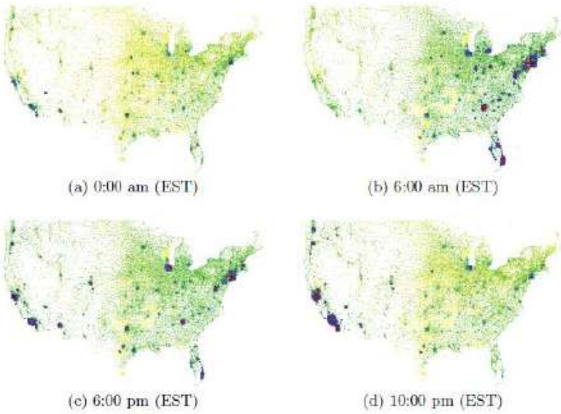


Figure 2.34 The images illustrate U.S. telephone call volume at four different times throughout a single day. The approach involves placing the initial data points in their accurate positions while arranging overlapping points in nearby available spots.

2.19.3 Visualization of Line Data

Line data represent spatial relationships using segments between geographic points. Parameters can be mapped to width, pattern, color, and labels. Unlike graph drawing, node positions are fixed in geospatial applications.

2.19.4 Network Maps

Network maps depict connectivity using color, shape, and width coding. Systems like SeeNet allow interactive exploration, while AT&T's SWIFT-3D integrates 2D/3D views to analyze network data (Figure 2.35). However, overlap issues persist in dense areas.

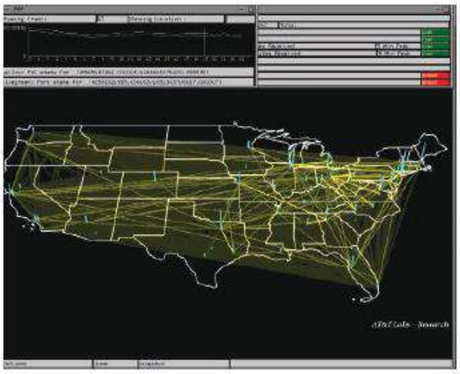


Figure 2.35 Swift-3D

2.19.5 Flow Maps and Edge Bundling

Curved lines reduce clutter in network maps. Stanford flow maps cluster nodes and reroute flows, while edge bundling groups edges based on a hierarchy, reducing visual noise). These techniques improve clarity in visualizing connections.

2.20 Visualization of Area Data

Thematic maps are essential for visualizing area-based phenomena. The most common types include:

1. **Choropleth Maps**: These maps encode attribute values using colored or shaded regions. They assume uniform distribution of attributes within regions (Figure 2.36). If the attribute distribution does not align with the region partitioning, alternative techniques like dasymetric maps are used.
2. **Dasymetric Maps**: These maps partition areas based on the distribution of the variable rather than predefined regions (Figure 2.37).
3. **Isarithmic Maps**: These maps display contours of continuous phenomena. Variants include:
   - **Isometric Maps**: Contours derived from real data points (e.g., temperature readings at specific locations).
   - **Isopleth Maps**: Contours generated from regional measurements, such as population density.

4. **Cartograms**: These distort region sizes to reflect statistical values, creating unique geometric distortions.

Additionally, area data can be visualized using discrete symbols or points, such as proportional symbols or dot density maps.

(a) (b)

Figure 2.36 Thematic maps: (a) A choropleth map depicting the U.S. election results from the 2008 presidential race between Obama and McCain. (b) An isarithmic map representing the number of photographs taken on Mainau Island using a heatmap, where colors transition

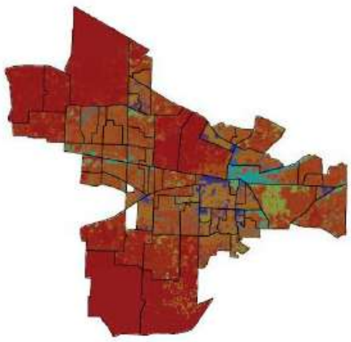


Figure 2.37 A dasymetric map showing the population distribution in Beaverton Creek, Oregon,

2.21 Choropleth Maps

Choropleth maps represent area data using shaded polygons with closed contours. They highlight spatial distributions of geographic attributes. Key considerations for choropleth maps include:

- Normalization of data
- Color mapping

However, choropleth maps may misrepresent data by emphasizing large, sparsely populated regions over smaller, densely populated ones (e.g., U.S. Census Demographics).

2.22 Cartograms

Cartograms address the limitations of choropleth maps by resizing regions based on statistical variables. Applications include:

- Population demographics
- Election results

- Epidemiology

2.22.1 Types of Cartograms

1. **Noncontinuous Cartograms** (Figure 2.38(a))
    - Satisfy area and shape constraints but do not preserve topology.
    - Polygons are drawn inside original regions.
    - Smaller polygons remain constrained in size.
2. **Noncontiguous Cartograms** (Figure 2.38(b))
    - Scale all polygons independently.
    - Preserve shape but distort topology.
    - May result in loss of overall map structure.
3. **Circular Cartograms** (Figure 2.38(c))
    - Ignore polygon shapes, using circles instead.
    - Often relax area and topology constraints.
4. **Continuous Cartograms** (Figure 2.38(d))
    - Retain topology while adjusting area and shape.
    - Maintain resemblance to the original map.

Continuous cartograms balance shape and area constraints, making them more recognizable but computationally complex.

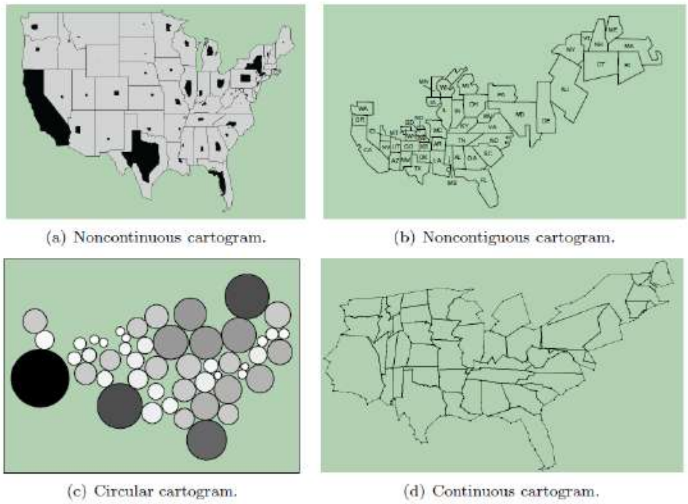


Figure 2.38 Different types of cartograms

2.22.2 The Continuous Cartogram Problem and the CartoDraw Algorithm

The continuous cartogram problem is defined as a map deformation challenge where a given planar polygon mesh must be adjusted so that each region's area matches a predefined value while preserving the overall shape as much as possible.

The Cartogram Problem

Given:

- A planar polygon mesh P consisting of polygons $p_1, p_2, \ldots, p_k$.
- A set of values $X = \{x_1, x_2, \ldots, x_k\}$ with $x_i > 0$ and $\sum x_i = 1$.
- Let $A(p_i)$ denote the normalized area of polygon p_i, where $A(p_i) > 0$ and $\sum A(p_i) = 1$.

Objective: Find a topology-preserving polygon mesh P' that minimizes the function:

$$f(S, A) = \omega \sum_{i=1}^{k} s_i + (1-\omega) \sum_{i=1}^{k} a_i$$

where:

- $S = \{s_1, s_2, \ldots, s_k\}$ with $s_i = d_S(p_i, p_i')$ (Shape Error)
- $A = \{a_1, a_2, \ldots, a_k\}$ with $a_i = d_A(x_i, A(p_i))$ (Area Error)
- ω is a weighting factor where $0 \leq \omega < 1$.

CartoDraw Algorithm

The iterative optimization process follows:

$$\text{CartoDraw}(P, \tilde{X}, SL)$$

repeat
    $\text{AreaError} = \text{AreaDist}(P, \tilde{X})$
    for each $(sl \in SL)$ do
        $P' = \text{Scanline}(P, \tilde{X}, sl)$
        if $(\text{ShapeDist}(P, P') < \epsilon_S)$ AND
            $\left(\text{AreaDist}(P, \tilde{X}) - \text{AreaDist}(P', \tilde{X}) > \epsilon_A\right)$ then $P = P'$
    until
        $\left(\text{AreaDist}(P, \tilde{X}) \leq \text{AreaErrorThreshold}\right)$ OR
        $\left(\text{AreaError} - \text{AreaDist}(P, \tilde{X}) \leq \epsilon_A\right)$ OR
        $(\text{IterationCount}{+}{+} \geq \text{MaxIterationCount})$.

Since this problem is computationally complex, various algorithms have been proposed to generate contiguous cartograms. Some approaches, such as radial expansion methods and nonlinear magnification techniques, result in excessive polygon deformation.

CartoDraw Algorithm

The CartoDraw algorithm provides an efficient solution by iteratively repositioning polygon vertices using scanlines. The key steps include:

1. **Input Processing:**
   - The map’s medial axes are identified.
   - Polygons that must expand or contract are determined.
2. **Iterative Adjustments:**
   - Scanlines are processed to incrementally adjust polygon boundaries.
   - If changes reduce total area error without excessive shape error, they are made permanent.

- The process repeats until the area error falls below a threshold or a maximum iteration count is reached.

3. **Results:**
   - Figure 2.39(a) shows the U.S. map with medial axes, where polygons requiring expansion are red, and those requiring contraction are blue.
   - Figure 2.39(b) illustrates local transformations performed by the CartoDraw algorithm.

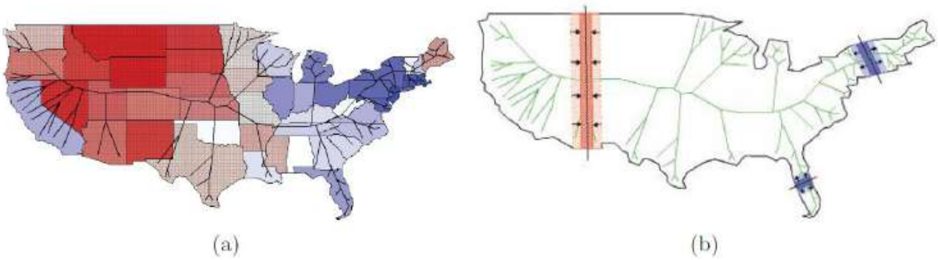


Figure 2.39 Medial Axes CartoDraw Algorithm: (a) U.S. map with medial axis and polygons

2.22.3 The Rectangular Cartogram Problem and the RecMap Algorithm

Concept

Rectangular cartograms approximate map regions using rectangles, ensuring areas are proportional to given statistical values. The goal is to maintain:

- Proximity to the original geographic positions.
- Adjacency relationships between regions.
- Proper aspect ratios of rectangles.

RecMap Algorithm

The RecMap algorithm constructs rectangular cartograms through heuristic optimization, adjusting placement and proportion to optimize constraints. A notable example is the U.S. population cartogram (Figure 2.40), which accurately represents population sizes while preserving region adjacencies.

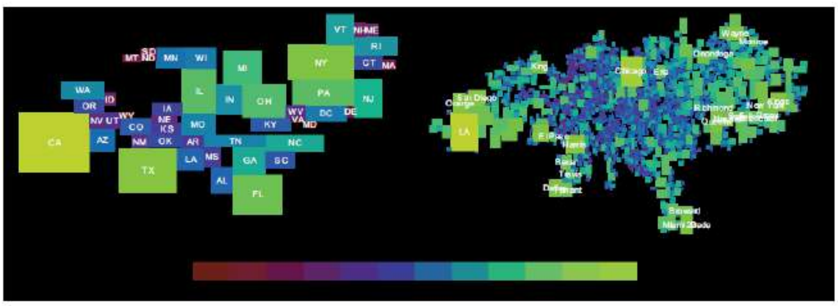


Figure 2.40 A rectangular U.S. population cartogram on the state and county level. The area

2.23 Other Issues in Geospatial Data Visualization

Effective spatial visualizations rely on cartographic techniques such as map generalization and map labeling.

2.23.1 Map Generalization

Map generalization selects and abstracts information to enhance perception in visualizations. It is crucial when creating small-scale maps from detailed large-scale ones. Since generalization is task-dependent, it highlights key elements while maintaining geographic accuracy.

Common techniques include:

- **Simplifying points** – Removing or merging irrelevant or indistinct points.
- **Simplifying lines** – Smoothing fluctuations, removing bends, and collapsing dual lines.
- **Simplifying polygons** – Preserving essential shapes while reducing unnecessary details.

2.23.2 Map Labeling

Map labeling places textual or symbolic labels near points, lines, and polygons. Although it may seem simple, it requires specialized algorithms for optimal placement. Methods include heuristics and optimization techniques like local search, greedy algorithms, and simulated annealing. Examples of labeling software include Label-EZ by MapText and Maplex by ESRI.

2.23.3 GIS and Geospatial Visualization

Geographic Information Systems (GIS) provide tools for spatial data exploration, enabling dynamic queries, geographic comparisons, and spatial analysis. Advancements in web technologies (e.g., AJAX, Google Maps API) have led to highly interactive visualization tools that improve public awareness of social, economic, and environmental issues.

A growing trend in GIS is integrating temporal data visualization to track spatial changes over time. Additionally, ensuring semantic interoperability through ontologies helps unify data from different sources, improving GIS functionality and data reuse.

Summary

This unit introduces the fundamental principles of data visualization, focusing on its role in enhancing data comprehension and decision-making. It explores the visualization process, graphical representation techniques, and the impact of visual variables on perception. Additionally, it provides a historical perspective on visualization advancements and taxonomies. Special emphasis is placed on geospatial data visualization, including various mapping techniques and Geographic Information Systems (GIS).

Keywords

Data visualization, graphical representation, visual variables, geospatial data, GIS, mapping techniques, perception, historical perspectives, taxonomies, decision-making.

Learning Outcomes

By the end of this unit, learners will be able to:

- Identify and describe the key components of the visualization process.
- Compare and apply different techniques for visualizing one-, two-, and three-dimensional data.
- Assess the impact of visual variables on human perception and data interpretation.
- Examine historical advancements and classification frameworks in data visualization.
- Utilize geospatial data visualization techniques in the context of Geographic Information Systems (GIS) applications.

Exercises

Objective Questions

1. Which of the following is a key component of the visualization process?
   a) Data Collection
   b) Data Transformation
   c) Mapping and Rendering
   d) All of the above
2. What type of data visualization technique is commonly used for one-dimensional data?
   a) Bar Chart
   b) Scatter Plot
   c) Heatmap
   d) Choropleth Map
3. Which of the following is NOT a visual variable?
   a) Position
   b) Orientation
   c) Motion
   d) Temperature
4. Who developed the Grammar of Graphics framework for visualization?
   a) Jacques Bertin
   b) Edward Tufte
   c) Leland Wilkinson
   d) Ben Shneiderman
5. What is the primary purpose of a choropleth map?
   a) Representing categorical data
   b) Showing density variations using colors
   c) Visualizing network connections
   d) Displaying three-dimensional surfaces
6. Which technique is used for direct volume visualization?
   a) Marching Cubes
   b) Isosurface Extraction
   c) Ray Casting
   d) Dot Maps
7. What does the term "geospatial data" refer to?
   a) Data related to financial markets

b) Data representing geographic locations
c) Data stored in tabular format
d) Data with undefined coordinates

8. In which year did Mackinlay introduce APT (A Presentation Tool) for visualization?
a) 1967
b) 1986
c) 1994
d) 2002
9. What is a key advantage of interactive visualizations?
a) They require less computational power
b) They allow users to explore and manipulate data dynamically
c) They are always easier to interpret
d) They remove the need for statistical analysis
10. What is the function of map generalization in geospatial visualization?
a) Adding excessive detail to a map
b) Abstracting and simplifying map features for better readability
c) Labeling every point on a map
d) Using only satellite images for mapping

Answers:

1. d) All of the above
2. a) Bar Chart
3. d) Temperature
4. c) Leland Wilkinson
5. b) Showing density variations using colors
6. c) Ray Casting
7. b) Data representing geographic locations
8. b) 1986
9. b) They allow users to explore and manipulate data dynamically
10. b) Abstracting and simplifying map features for better readability

Short Questions

1. What are the main steps involved in the visualization process?
2. How do visual variables influence data interpretation?
3. What is the significance of Bertin's Semiology of Graphics in data visualization?
4. Explain the difference between one-dimensional, two-dimensional, and three-dimensional data visualization.
5. What are the key challenges in geospatial data visualization?

Long Questions

1. Discuss the role of visual perception in data visualization. How do different visual variables affect the way people interpret graphical data?
2. Compare and contrast various geospatial data visualization techniques. Provide examples of their real-world applications.

3. Explain different methods used for visualizing three-dimensional data. Discuss their advantages, limitations, and areas of application.

Programming Exercises

1. Modify `DrawLineGraph()` to render a color bar instead of a line graph. The color bar should be based on a given color ramp spanning a specified range (`colormin` to `colormax`). Ensure the width of the color bar is determined by the available screen width, assuming the number of data points does not exceed it. Allow the height of the rectangle to be set by the user.
2. Develop a program that enhances `DrawLineGraph()` by implementing data subsampling. If the number of data points (`dataCount`) exceeds the number of available pixels in the drawing area (`xMax - xMin`), the program should automatically reduce the data points to fit within the available space.
3. Implement a program that loads a three-dimensional volume data set and enables the user to select and view a specific slice. The visualization should use a grayscale color map with 256 intensity levels to represent the data.
4. Develop a script to process the TIGER-System (Topologically Integrated Geographic Encoding and Referencing) geographic polygon dataset from the U.S. Census (`TGR06001.RT2`). The script should extract polygon data and convert it into the following format for each polygon:

```
-121764253|+37160714
-121746453|+37611800
-121746709|+37611300
NA—NA
```

   The `NA—NA` line should indicate the end of each polygon. The necessary TIGER files can be accessed at `ftp://ftp2.census.gov/geo/tiger/TIGER2013`.

5. Use the R-project function “polygon” to draw the extracted polygons of Project 1.

Collaborative Learning Tasks

1. **Comparing Visualization Techniques:**
   - Form small groups and research different visualization techniques for one-, two-, and three-dimensional data.
   - Each group will present a specific technique, explaining its strengths, limitations, and best use cases.
   - Discuss as a class which techniques work best for different types of data and why.
2. **Analyzing Misleading Visualizations:**
   - Each group will find a real-world example of a misleading visualization (e.g., distorted graphs, inappropriate color schemes, incorrect scales).
   - Identify the flaws and discuss how they impact data interpretation.
   - Propose redesigns that improve accuracy and clarity.
3. **Building a Geospatial Visualization Project:**
   - Collaborate on a project where each group selects a real-world problem (e.g., urban traffic, climate change, disease outbreaks).
   - Use GIS tools or mapping software to visualize relevant data.

- Present findings and discuss how visualization aids in decision-making.

Case Studies and Questions

1. **Case Study:** Examine a real-world GIS-based disaster management system. How does visualization contribute to emergency response?
2. **Case Study:** Analyze a famous misleading visualization (such as a distorted map projection or an improperly scaled chart). What errors are present, and how can they be corrected?
3. **Case Study:** Review the development of a well-known visualization tool, such as Tableau or Power BI. How has it evolved to enhance user interaction and data comprehension?

Study Tips

- Focus on understanding different visualization techniques rather than just memorizing them.
- Practice analyzing real-world visualizations and identifying their strengths and weaknesses.
- Experiment with visualization tools like D3.js, Python libraries, or GIS software to gain hands-on experience.
- Collaborate with peers to discuss interpretations and improve comprehension.
- Use historical perspectives to understand how visualization principles have evolved over time.

Self-Assessment Tasks

1. Explain the role of mapping techniques in geospatial data visualization.
2. Identify and differentiate between various types of visual variables.
3. Create a simple bar chart using any visualization tool and evaluate its effectiveness.
4. Compare different geospatial visualization techniques and list their advantages and disadvantages.
5. Reflect on how visualization impacts decision-making in real-world applications.

2.24 FURTHER READINGS

- Matthew O. Ward, Georges Grinstein, and Daniel Keim. Interactive data visualization: foundations, techniques, and applications. AK Peters/CRC Press, 2nd Edition, 2015.
- Ben Fry, Visualizing data: Exploring and explaining data with the processing environment. " O'Reilly Media, Inc.", 1st Edition, 2007.
- Bertin, J. (2010). Semiology of Graphics. ESRI Press.
- Bertin, J. (1981). Matrix Theory of Graphics. Information Design Journal, 2(2), 59-72.
- Wilkinson, L. (2005). The Grammar of Graphics. Springer.
- Chi, E. H. (2000). A Data State Reference Model for Information Visualization. IEEE Symposium on Information Visualization.
- Wright, H. (2007). Introduction to Scientific Visualization. Springer.
- Telea, A. (2008). Data Visualization: Principles and Practice. A. K. Peters.
- Hansen, C. D., & Johnson, C. R. (Eds.). (2005). The Visualization Handbook. Elsevier.

2.25 NPTEL VIDEOS

- Introduction to R - https://youtu.be/rl1oUlLtR6k
- Data Visualization – Full Course - https://youtu.be/i0QSlGJ3jGU?si=sYPtJEfFi9RIGRBZ
- Introduction to Data Visualisation Analysis _ Part 1 - https://www.youtube.com/watch?v=qdnM8Fpvdqc

References

The contents of this chapter have been prepared based on the reference book:

Matthew O. Ward, Georges Grinstein, and Daniel Keim. Interactive data visualization: foundations, techniques, and applications. AK Peters/CRC Press, 2nd Edition, 2015.

All the figures have been referred from this text book.

CONTENTS

UNIT 3

DESIGNING EFFECTIVE VISUALIZATION

3.1 Introduction

Visualization is a powerful tool for interpreting and communicating complex data. Effective visualizations enable users to gain insights, identify patterns, and make informed decisions. However, poorly designed visualizations can lead to misinterpretation, confusion, or even manipulation of data. This unit explores the principles of designing effective visualizations, the key steps involved, common pitfalls to avoid, and methods for evaluating different visualization techniques.

3.2 Learning Objectives

By the end of this unit, learners should be able to:

1. Understand the fundamental principles of effective visualization.
2. Apply key steps in designing intuitive and informative visual representations of data.
3. Identify and mitigate issues that lead to misleading or ineffective visualizations.
4. Evaluate visualization techniques based on user tasks, data characteristics, and usability metrics.
5. Implement best practices for using color, labels, legends, and aesthetics in visualization.
6. Compare different visualization methods through benchmarking and structured evaluation.

3.3 Overview

This unit begins by exploring the characteristics of effective visualizations and the essential steps in designing visualizations. Topics such as intuitive mappings, selecting appropriate views, balancing information density, and using color effectively are discussed. The unit also addresses common problems in visualization, including misleading representations, chart junk, and issues related to raw versus derived data. Finally, it introduces techniques for comparing and evaluating visualization methods, considering factors like user needs, data properties, and benchmarking approaches. By the end of this unit, learners will gain a deeper understanding of how to create meaningful, accurate, and aesthetically appealing visualizations.

3.4 Effective Visualizations

An effective visualization should efficiently and accurately convey the intended information to the target audience while considering its purpose—whether for exploration, confirmation, or presentation. Given a dataset, numerous methods exist for mapping data components to graphical attributes, and various interactive tools can be provided to users. Selecting the most suitable techniques is a complex task.

A visualization may fail for several reasons. It could be overly complex or confusing for the intended audience, or essential data might be distorted, hidden, or lost during the mapping process. Other deficiencies include the absence of tools for modifying views or controlling color maps. Even aesthetics play a role in the effectiveness of a visualization; an unappealing design may discourage engagement. Any interference with the transmission of information hinders the visualization's effectiveness.

3.5 Steps in Designing Visualizations

Creating a visualization involves mapping data fields to graphical attributes, selecting and implementing methods for modifying views, and determining the appropriate volume of data to display. Additional elements, such as labels and color keys, are crucial for interpretation and should be integrated seamlessly. Aesthetic appeal is another important factor to consider. Each stage in the design process presents specific issues that designers must address.

3.5.1 Intuitive Mappings from Data to Visualization

For an effective visualization, it is crucial to consider data semantics and user context. When mappings align with users' domain-specific mental models, interpretation becomes more straightforward. Consistency in design reduces misinterpretation and accelerates understanding. For instance, Figure 3.1 illustrates a scatterplot using planetary images to depict the relationship between distance from the sun and orbital duration.

Spatial attributes like longitude and latitude are often mapped to screen positions, as humans naturally associate placement in visualizations with real-world positions. Similarly, animation provides an intuitive representation of temporally related data, allowing users to observe variations over time.

Some mappings become intuitive within specific contexts. For example, temperature is frequently mapped to color, with red or white signifying higher temperatures. In disciplines like cartography and geology, color plays a specialized role, such as land-use classification or stratigraphic layering. Similarly, height or line length is effective for displaying scalar values, such as temperature readings on a thermometer.

A critical consideration is the compatibility between data scale and graphical representation. Ordered data (e.g., age) should be mapped to ordered graphical attributes (e.g., length), while categorical data (e.g., country of origin) should not be represented using ordered attributes.

Though intuitive mappings are generally preferred, unconventional mappings can reveal hidden patterns in data. For instance, using color to represent time along a streakline can expose variations in particle speeds that might otherwise go unnoticed. A recommended approach is to use intuitive default mappings while allowing user customization for exploratory tasks.

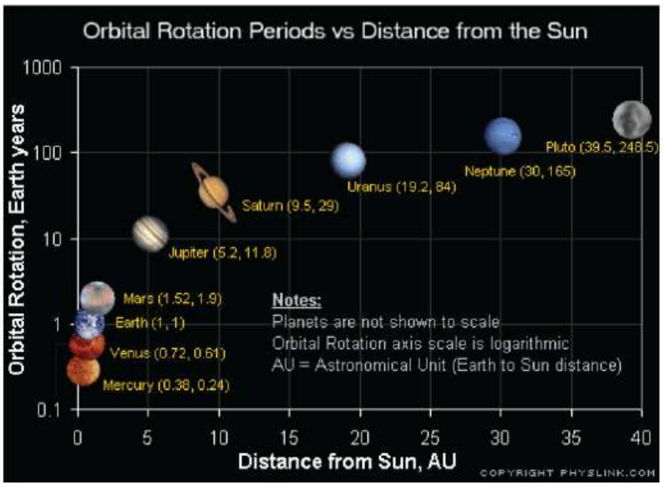


Figure 3.1 Utilizing intuitive scatterplot symbols to represent the distance of planets from the Sun in relation to the duration of their orbits. (Image source: http://www.physlink.com.)

3.5.2 Selecting and Modifying Views

A single view is often insufficient for complex datasets. Effective visualization design anticipates the most useful views and provides intuitive controls for customization. The choice of views depends on the dataset and the associated task, with each view requiring clear labeling and easy accessibility.

Key view modification categories include:

- **Scrolling and zooming**: Necessary when the entire dataset cannot be displayed at the desired resolution.
- **Color map control**: Users should be able to select different palettes or customize individual colors.
- **Mapping control**: Switching between different visualizations of the same data can highlight hidden features (Figure 3.2).
- **Scale control**: Adjusting value ranges before mapping helps emphasize critical data points.
- **Level-of-detail control**: Enables users to switch between varying levels of abstraction (Figure 3.3).

User-friendly implementations of these modifications enhance visualization effectiveness. Direct manipulation techniques, such as using mouse gestures for zooming and panning, are generally preferable.

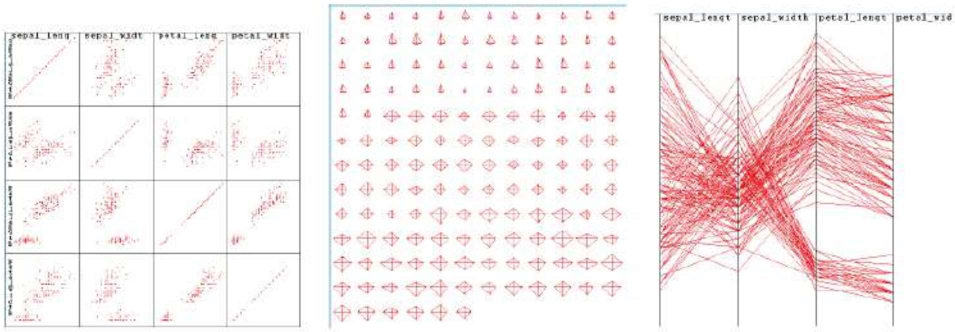


Figure 3.2 Three visual representations of the Iris dataset: scatterplot matrix, star glyphs, and parallel coordinates. (Image source: XmdvTool.)

Figure 3.3 Different levels of detail in maps. (Images courtesy of Google Maps © 2008 Google; map data © 2008 NAVTEQ™.)

3.5.3 Information Density—Balancing Too Much and Too Little

Deciding how much information to display is a key challenge. Two extreme situations should be avoided:

1. **Excessive graphics**: Some visualizations use unnecessary graphical elements to present minimal data, such as gender distribution, which could be effectively conveyed using a simple number.
2. **Overloaded visualizations**: Displaying too much information can cause confusion, making it difficult for viewers to focus on key insights.

Solutions include:

- Allowing users to enable or disable display components as needed.
- Using multiple screens or panels to distribute information efficiently.
- Filtering large datasets to highlight relevant details.
- Adjusting scales to balance data distribution and avoid congestion in specific areas.

3.5.4 Keys, Labels, and Legends

A common issue in visualizations is the lack of sufficient supporting information. Essential elements include:

- **Captions** explaining displayed data and mappings.
- **Grid and tick marks** to define numeric ranges.
- **Axis labels** with appropriate units.
- **Symbol keys** for easy interpretation.
- **Color legends** when color carries significance (Figure 3.4).

Poor grid or tick mark placement can obscure data, leading to clutter. Likewise, inconsistent grid spacing can confuse users. Careful selection of displayed value ranges prevents misinterpretation. Consistency in labeling across multiple frames or windows is crucial to prevent confusion.

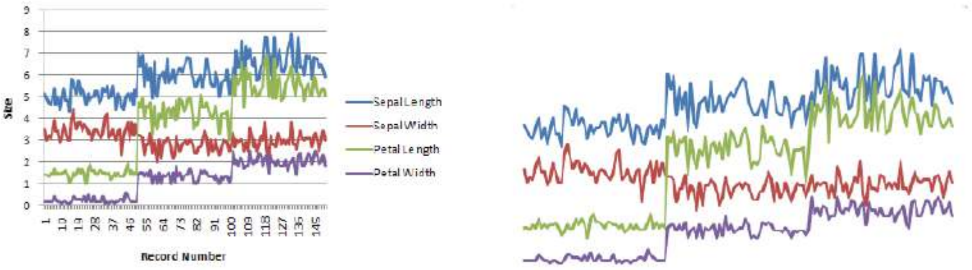


Figure 3.4 A complex visualization with and without captions/ticks/legends.

3.5.5 Using Color with Care

Color is often misused in visualizations, leading to misleading or ineffective presentations. Color perception is context-dependent and varies among individuals, including those with color vision deficiencies. Guidelines for effective color usage include:

1. Limiting the number of numeric color levels (Figure 3.5).
2. Using redundant mappings, such as both color and size, to reinforce data representation (Figure 3.6).
3. Ensuring both hue and lightness vary for better differentiation (Figure 3.7).
4. Providing labeled color keys.
5. Using semantically meaningful colors where possible.

Some designers advocate starting with grayscale visualizations before adding color to enhance effectiveness. While color enhances visual appeal, it should be used judiciously to ensure clear communication.

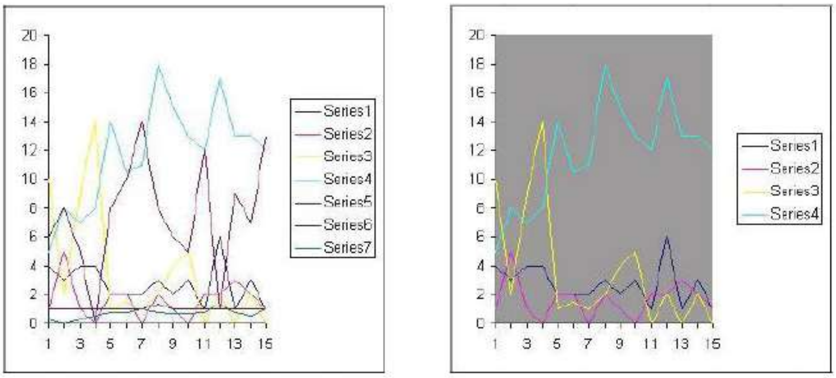


Figure 3.5 Too many colors versus a moderate number of colors

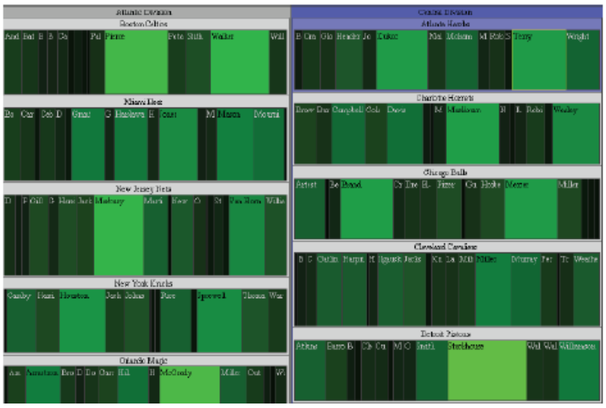


Figure 3.6 Treemap of basketball statistics, with points per game redundantly mapped to color

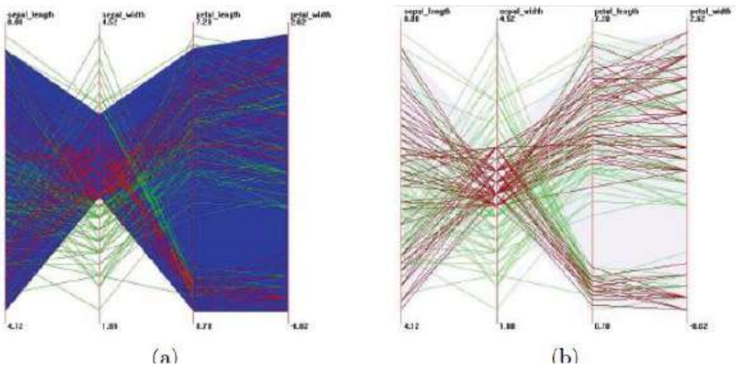


Figure 3.7 (a) Changing hue; (b) changing both hue and saturation

3.5.6 The Importance of Aesthetics

Once a visualization effectively conveys information (function), the final step is to assess aesthetics (form). The best visualizations are both informative and visually appealing, inviting deeper engagement. Conversely, poor aesthetics can detract from communication.

Guidelines from art and design communities help create attractive visualizations:

- **F**ocus: The most important elements should be emphasized to guide viewer attention (Figure 3.8).
- Balance: Effective use of screen space keeps key components central without biasing toward borders (Figure 3.9).
- Simplicity: Avoid overcrowding or unnecessary graphical effects. Iteratively removing elements and assessing the impact helps retain only essential details (Figure 3.10).

Many examples of poor visualizations exist, often featuring random, distracting patterns. Designers should assess aesthetics before presenting results and leverage existing design literature (Figure 3.11).

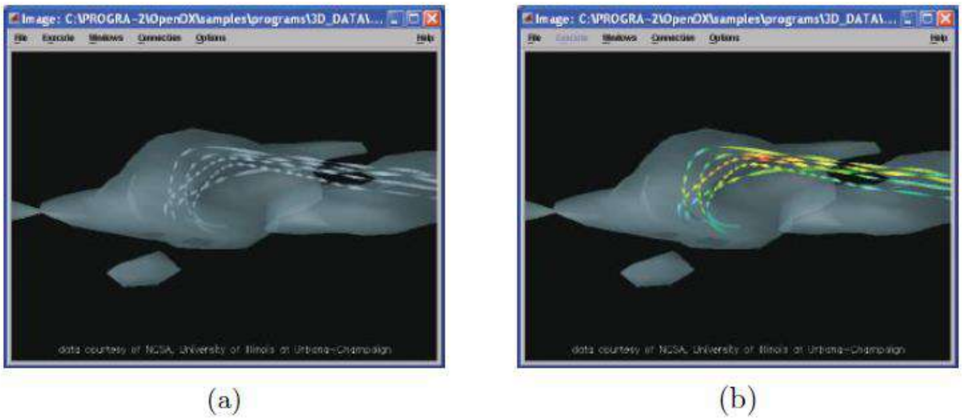


Figure 3.8 (a) Subdued streamlines vs. (b) highlighted streamlines from OpenDX.

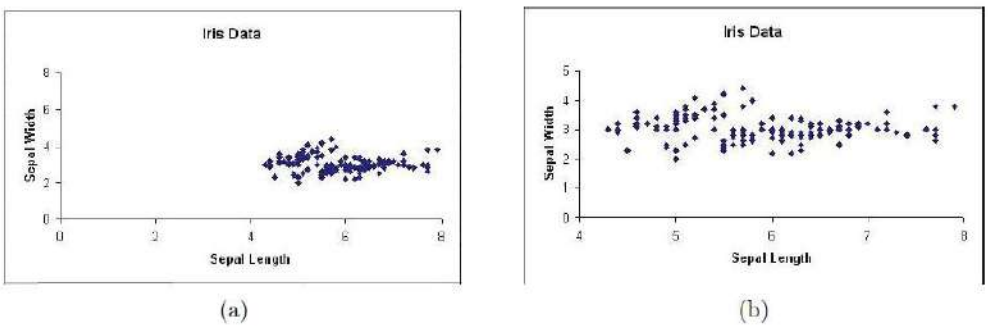


Figure 13.12. (a) Everything to one side vs. (b) balanced between left and right.

Figure 3.9 (a) Everything to one side vs. (b) balanced between left and right.

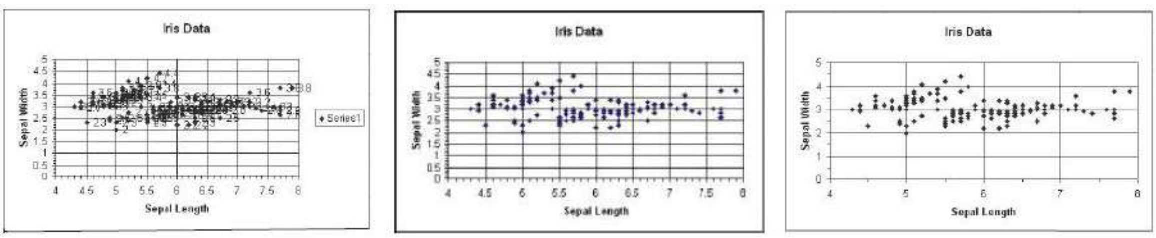


Figure 3.10 Progression from a cluttered chart to a simplified chart.

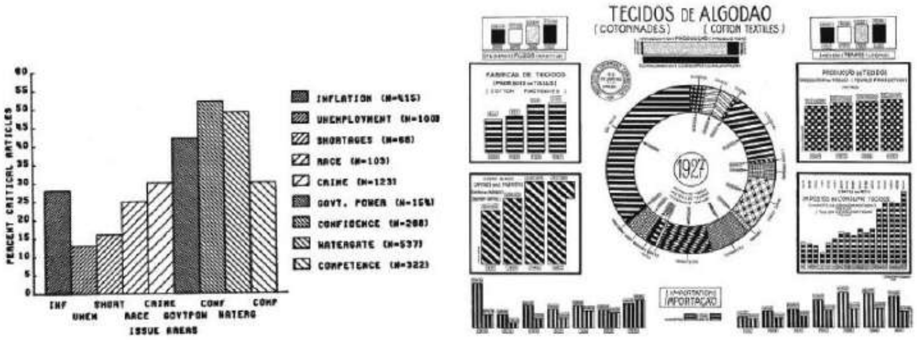


Figure 3.11 Examples of poorly designed visualizations. In both cases, the random and distracting cross-hatchings could be improved by using shades of gray for better visual clarity

3.6 Problems in Designing Effective Visualizations

Despite following best practices, common issues can arise, often stemming from decisions about what to visualize and how. Some problems distort data, leading to misinterpretation, while others obscure real data behind excessive graphics. Addressing these issues improves clarity and honesty in visual representation.

3.6.1 Misleading Visualizations

A fundamental rule of visualization is accuracy, yet distorted data is often used to manipulate audiences. "Viz lies" appear everywhere, from academic journals to corporate reports. Some common misleading techniques include:

- **Data scrubbing**: Selectively removing outliers to support a particular argument can distort findings. Outliers should be removed only if they are proven data errors and with transparency (Figure 3.12).
- **Unbalanced scaling**: Scaling can highlight patterns but may also exaggerate or diminish trends. Tufte's "lie factor" describes the disparity between actual data changes and their depiction (Figure 3.13).
- **Range distortion**: Unexpected axis adjustments can mislead viewers. While modifying baselines to optimize space is valid, it should be clearly indicated if it deviates from standard expectations (Figure 3.14).
- **Abusing dimensionality**: Errors increase with dimensional complexity. Mapping scalar values to volume can mislead interpretations. Simpler representations often work better.

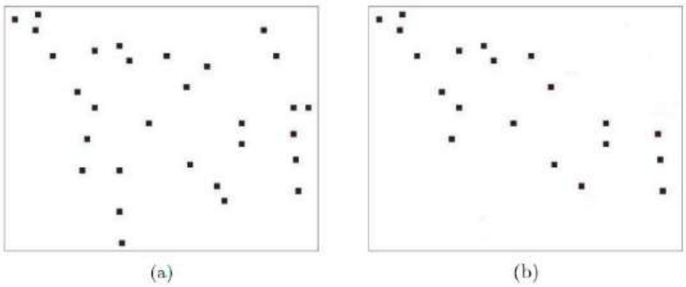


Figure 3.12 The issue with data scrubbing: (a) raw data displaying no apparent correlation, and (b) scrubbed data misleadingly suggesting a false correlation.

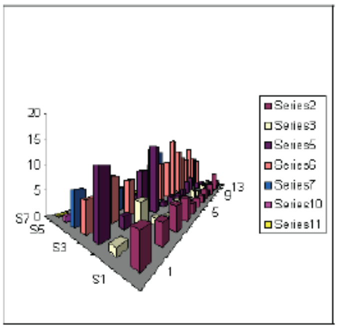


Figure 3.13 Viz Lies: Perspective skews size perception, making closer objects appear disproportionately larger.

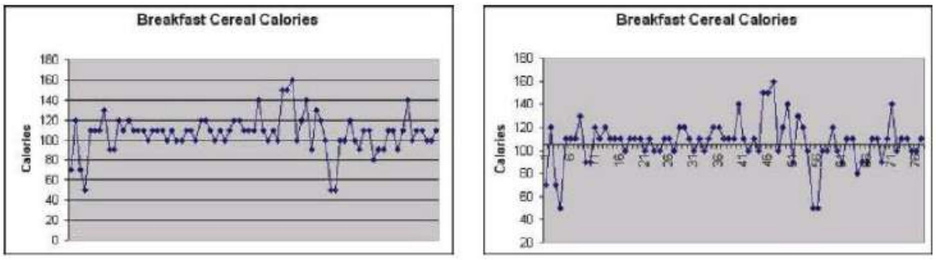


Figure 3.14 Plotting data with different baselines

3.6.2 Visual Nonsense—Comparing Apples and Oranges

Visualizations should convey meaningful relationships. Combining unrelated datasets can create misleading patterns, such as correlating stock market trends with sunspot occurrences (Figure 3.15). Ensuring logical combinations prevents misinterpretation.

Factors to consider:

- **Temporal and spatial compatibility**: Comparing data across different times and locations should be justified.
- **Consistent units**: Data measured in different units (e.g., price per volume vs. price per weight) should be normalized for valid comparisons.
- **Ordered vs. categorical data**: Applying trends or curves to unordered data (e.g., company names) lacks meaning and should be avoided.

Thoughtful consideration ensures logical and accurate visual representations.

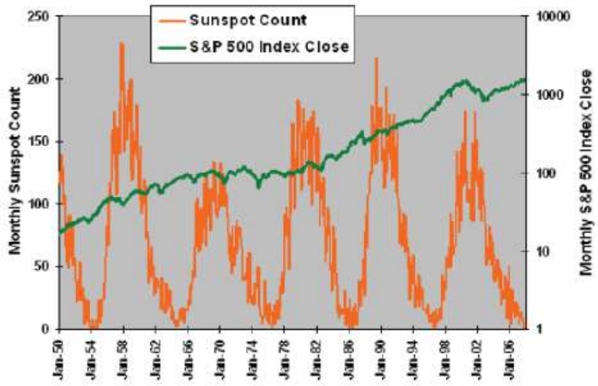


Figure 3.15 An irrelevant correlation: A plot comparing sunspot occurrences to the S&P 500 Index. (Image source: http://www.cxoadvisory.com/blog/internal/blog4-07-09/.)

3.6.3 Losing Data in Chart Junk

Excessive non-data elements, known as "chart junk," can clutter visualizations. While labeled grids and tick marks help with quantitative assessment, excessive use can obscure key insights. Because visualizations are dynamic, users should have options to adjust supplementary information. For qualitative overviews, clarity should be prioritized, while quantitative analysis may require additional tools. A good approach balances usability, allowing users to toggle elements based on their needs.

3.6.1 Raw versus Derived Data

A common approach in data visualization is to apply analytic models such as curve or surface fitting to create smoother, more visually appealing representations. However, this can distort the truth, leading to misleading assumptions and conclusions. In some cases, raw data is discarded entirely in favor of a smooth approximation, forcing viewers to trust that the derived model accurately represents the original data. This trust may be misplaced, especially when statistical fitting algorithms are applied without scrutiny. A best practice is to present both the raw data and the fitted model while allowing users to emphasize or filter out either as needed (Figure 13.16).

Another method of data refinement is resampling, where raw data from a sparse or randomly distributed grid is transformed into a denser or regularly spaced grid. This technique can enhance visualization quality, making it resemble continuous sampling. However, it may also create the false impression of a larger dataset than actually exists. The more densely the data is resampled, the greater the risk of misinterpretation—unless the observed phenomenon exhibits minimal variability.

For instance, Figure 13.17 illustrates global temperature monitoring stations, highlighting large gaps where no data is collected. Resampling in such cases can produce inaccurate conclusions, such as interpolating data from only a few stations to represent an entire region, potentially leading to false claims about temperature trends over time.

Insufficient sampling presents additional challenges. As seen in Figure 13.18, uniform interpolation without considering data characteristics can overlook critical features. The left image in the figure applies uniform sampling, while the right image utilizes contour information to add sample points in areas with significant variation.

Users must always have access to the raw dataset and be informed of any modifications, such as scrubbing, smoothing, or resampling. In fields like radiology, data filtering is often avoided to prevent the loss of crucial signals mistaken for noise. Consequently, it is essential to provide visualization options that allow users to examine the raw data before interacting with any derived versions, ensuring an accurate representation of the original information.

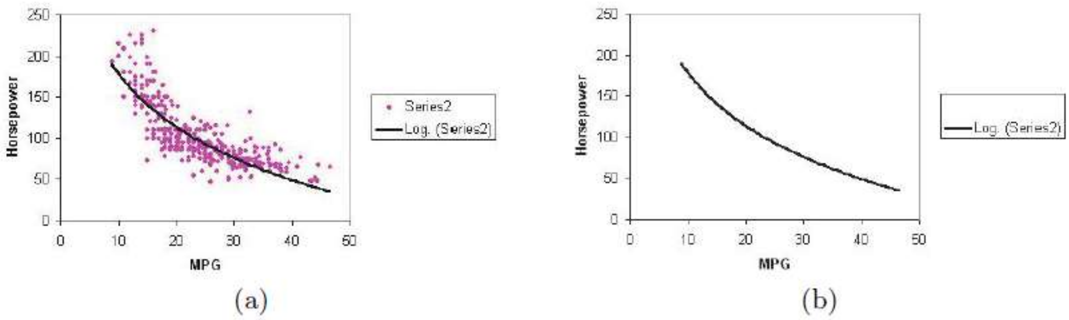


Figure 3.16 (a) Raw data plot with fitted curve; (b) only fitted curve

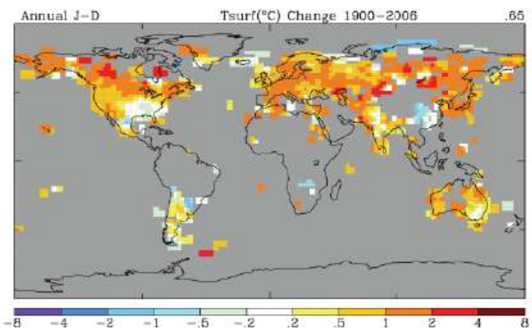


Figure 3.17 Limited global temperature data could lead to inaccurate values across most of the planet when interpolated.

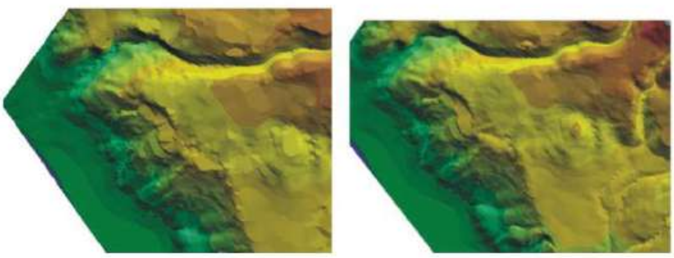


Figure 3.18 Variations in sampling and interpolation of the same dataset: Certain details visible in the right image are absent in the left image.

3.6.2 13.2.5 Absolute versus Relative Judgment

As discussed in Chapter 3, human perception has limitations in making absolute judgments of visual stimuli. Visualizations that require users to precisely interpret graphical attributes such as position, length, and color may lead to errors in understanding. To mitigate this issue, visualizations should favor relative judgment over absolute judgment or restrict the number of distinct values assigned to graphical attributes.

Techniques such as bounding boxes, grids, and tick marks help shift the cognitive load from absolute judgment to relative comparison. By referencing these structures, users can more effectively gauge the approximate value of a data point in relation to known levels. Another useful approach involves using residuals—subtracting values from their means—to simplify the measurement process into a binary decision of whether a value is above or below a reference level (Figure 3.19).

Figure 3.19 Some examples of absolute versus relative judgment. (Image courtesy of Michael

3.7 Comparing and Evaluating Visualization Techniques

A common challenge faced by users of visualization tools is determining the most suitable technique for their specific problem. Several factors influence this decision, including the tasks to be accomplished, data characteristics, and user expertise in visualization. Additionally, developers often struggle with assessing the effectiveness of their visualization techniques compared to existing ones or determining whether modifications improve or degrade their system.

Conducting formal evaluations of visualization techniques is complex and time-consuming. However, assessing and comparing these techniques is essential to ensure their effectiveness. The following sections explain the key components necessary for evaluation, including user tasks, user characteristics, data characteristics, and visualization attributes. It also describes a systematic approach for benchmarking visualization techniques.

3.7.1 User Tasks

A proper assessment of a visualization technique requires identifying specific tasks that users aim to perform. Keller and Keller categorize these tasks as follows:

- **Identify**: Recognizing an object based on its characteristics (e.g., detecting fractures in an X-ray).
- **Locate**: Determining an object's position (e.g., finding the point of maximum stress in structural analysis).
- **Distinguish**: Differentiating objects based on certain attributes (e.g., identifying elevations above a threshold).
- **Categorize**: Classifying objects into distinct types (e.g., identifying different land cover types).
- **Cluster**: Grouping similar objects based on relationships; segmentation involves separating dissimilar objects.
- **Rank**: Arranging objects in order (e.g., chronological or numerical ranking).
- **Compare**: Analyzing similarities and differences without necessarily ranking them (e.g., identifying variations between datasets).
- **Associate**: Establishing relationships between objects (e.g., linking temperature and location on a weather map).
- **Correlate**: Identifying causal or reciprocal relationships between objects (e.g., studying interest rates and economic growth).

The effectiveness of a visualization technique depends on its ability to accomplish these tasks with accuracy. Considerations such as error tolerance, resolution, and acceptable misclassification rates must be factored into the evaluation.

3.7.2 User Characteristics

User expertise significantly influences the effectiveness of visualization techniques. Users can be classified based on:

- **Familiarity with the domain**: Experience in the field of data being visualized.
- **Familiarity with the task**: Experience in performing specific tasks using visualization.
- **Familiarity with the data**: Prior exposure to similar datasets.
- **Familiarity with the visualization technique**: Experience using the specific visualization method.
- **Familiarity with the visualization environment**: Experience with the software or tool implementing the visualization.

Each factor influences how well users interpret visualizations. Evaluations should consider a user group representative of the intended audience.

3.7.3 Data Characteristics

The nature of the dataset greatly impacts visualization effectiveness. Evaluations should consider:

- **Type**: Homogeneous (e.g., numerical) or heterogeneous (e.g., text and numbers) data types.
- **Size**: Ranges from small datasets to large-scale records.
- **Dimensionality**: Number of variables; may involve projections or subsets.
- **Number of parameters**: Assessing the effectiveness of visualization with univariate or multivariate datasets.
- **Structure**: Simple (e.g., grids) or complex (e.g., hierarchical structures).
- **Range**: Testing values across their entire range.
- **Distribution**: Uniform or nonuniform distribution of values.
- **Real vs. synthetic data**: Synthetic data allows controlled tests, but real data ensures practical applicability.

Comprehensive evaluation requires testing across diverse datasets, though practical constraints may limit this approach. Developing annotated data repositories for benchmarking would benefit visualization researchers and developers.

3.7.4 Visualization Characteristics

Evaluation of visualization techniques must consider:

- **Computational performance**: Speed of rendering across varying dataset sizes.
- **Memory performance**: Resource requirements for generating visualizations.
- **Data limitations**: The upper and lower bounds of dataset size and complexity.
- **Degree of occlusion**: The extent to which parts of the visualization obscure others.
- **Degree of complexity**: Learning curve and required user adjustments.
- **Degree of usability**: Ease of interaction and interpretation.
- **Degree of accuracy**: Success rate and error analysis in performing tasks.

3.7.5 Structures for Evaluating Visualizations

Several evaluation methods exist within human-computer interaction research that apply to visualization systems:

- **Usability tests**: Assess visualization effectiveness based on efficiency, engagement, error tolerance, and ease of learning.
- **Expert reviews**: Involves evaluations by visualization or domain experts.
- **Field tests**: Long-term usability tests in real-world scenarios.
- **Case studies and use cases**: Demonstrating real or simulated examples of visualization application.

Each method offers different insights into visualization effectiveness. A well-rounded evaluation typically combines multiple approaches to ensure reliability and practical relevance.

3.7.6 Benchmarking Procedures

Benchmarking is a systematic approach for evaluating the performance of an object or a set of objects based on specific attributes. These attributes can range from quantitative measures, such as speed, to qualitative aspects, such as user experience. While quantitative benchmarks yield numerical results and can often be conducted without human subjects, qualitative benchmarking typically involves human participants, making it more complex due to user variability. Despite these differences, both approaches follow similar steps:

1. **Formulating a Hypothesis**: A benchmarking process begins with a precise hypothesis, such as "Algorithm A executes faster than Algorithm B" or "Visualization Technique X enables easier identification of tumors compared to Technique Y." A well-defined hypothesis should be specific, incorporating elements like user type, task, and data complexity. For example, "System A allows novice users to more effectively identify clusters in data sets with 5–10 dimensions and 1000 to 10,000 data points than System B."
2. **Designing the Experiments**: The design of benchmarking experiments should ensure that only one attribute varies at a time. For computational speed tests, all algorithms should run on the same hardware with the same data set and level of optimization. For usability studies, subjects should perform identical tasks that target the usability attribute, avoiding interference from variables like color schemes or hardware differences.
3. **Executing the Experiments**: Proper execution is crucial to obtaining reliable results. Human subjects should receive uniform training regarding procedures, response formats, and time constraints. A sample size of at least 15–20 participants with similar backgrounds is recommended to achieve statistical significance. For non-human subject experiments, conditions like computational load and time of execution should be controlled to minimize external influences.
4. **Analyzing the Results and Validating the Hypothesis**: The results of the benchmarking study must be analyzed to determine whether the hypothesis is supported, refuted, or inconclusive. Statistically significant findings are essential to establishing credibility. For quantitative variables, results can be plotted against variables such as data set size. For qualitative factors, each variable should be analyzed separately to identify dominant trends or clear patterns.

3.7.7 An Example of Visualization Benchmarking

Ward and Theroux conducted a series of experiments to evaluate the strengths and weaknesses of three multivariate visualization techniques—scatterplot matrices, parallel coordinates, and star glyphs—when applied to cluster analysis and outlier detection. The structure and results of their experiments are described below.

Outlier Detection and Measurement Experiments

- **Stage 1**: A quantifiable definition for outliers was developed using an algorithm based on one-dimensional projection regression. This algorithm calculated the fit for each point within each dimension, determining outliers based on deviations exceeding a threshold value.
- **Stage 2**: Both real and simulated data sets containing outliers were generated. Some data sets varied the separation between outliers and main clusters to test user sensitivity

(Figures 3.21 and 3.22). Six experiments were conducted and repeated for each visualization technique.

- **Stage 3**: Nineteen computer science graduate students with limited exposure to data visualization received an hour of training before being shown 18 images of data sets containing 0 to 6 outliers. Participants had a one-minute limit per image and were required to:
  1. Identify whether an image contained outliers.
  2. Mark the suspected outliers.
  3. Estimate the degree of separation on a five-point scale (marginal to extreme).
- **Stage 4**: Results were analyzed to determine the effectiveness of each visualization technique in detecting and measuring outliers. Performance was assessed based on correct and incorrect identifications, along with the average error in separation estimation.

Cluster Detection and Measurement Experiments

- **Stage 1**: A cluster definition and classification algorithm were developed, assigning probabilities of cluster membership to each data point.
- **Stage 2**: Both real and simulated data sets containing clusters were obtained. Simulated data sets varied in cluster number, size, orientation, and relative positioning, with some containing additional noise (Figure 3.20). Sixteen experiments were conducted for each visualization method.
- **Stage 3**: The same 19 graduate students received an hour of training before analyzing 48 images containing 0 to 4 clusters. Participants had a one-minute limit per image and were required to:
  1. Determine if an image contained clusters.
  2. Highlight perceived clusters.
  3. Estimate cluster size on a five-point scale (small to very large).
- **Stage 4**: Results were assessed based on the percentage of correctly and incorrectly identified clusters and errors in size estimation.

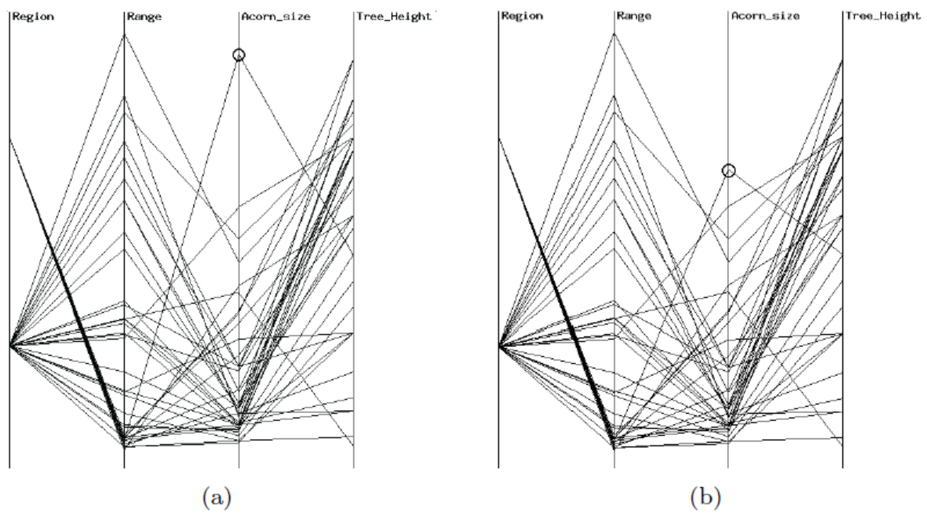


Figure 3.20 Parallel coordinates representation of a dataset on acorn attributes, highlighting a single outlier (circled in the acorn size dimension): (a) in its original position and (b) with its distance artificially reduced (Image © 1997 IEEE.)

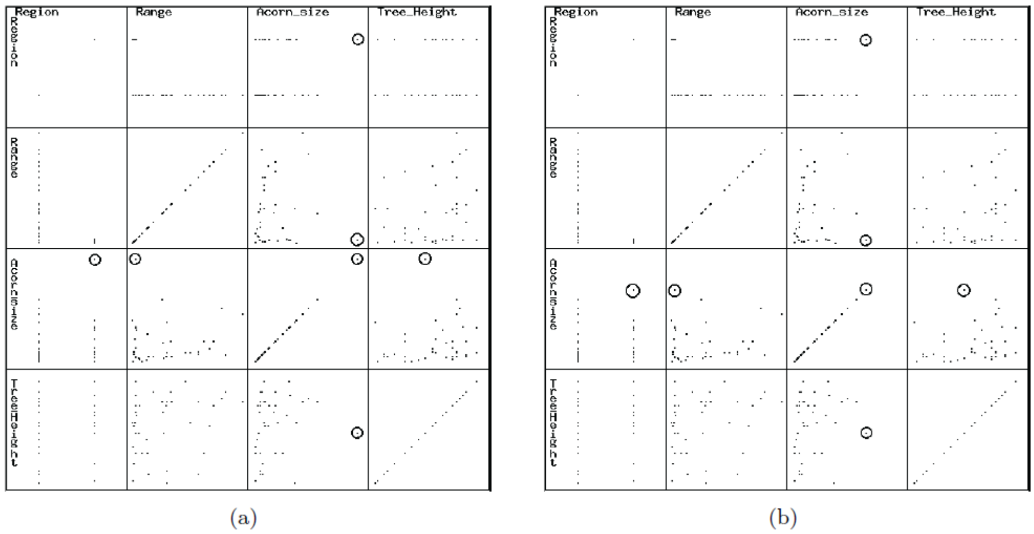


Figure 3.21 Detecting outliers using scatterplot matrices: The same dataset as the previous figure, now visualized through scatterplot matrices (Image © 1997 IEEE.)

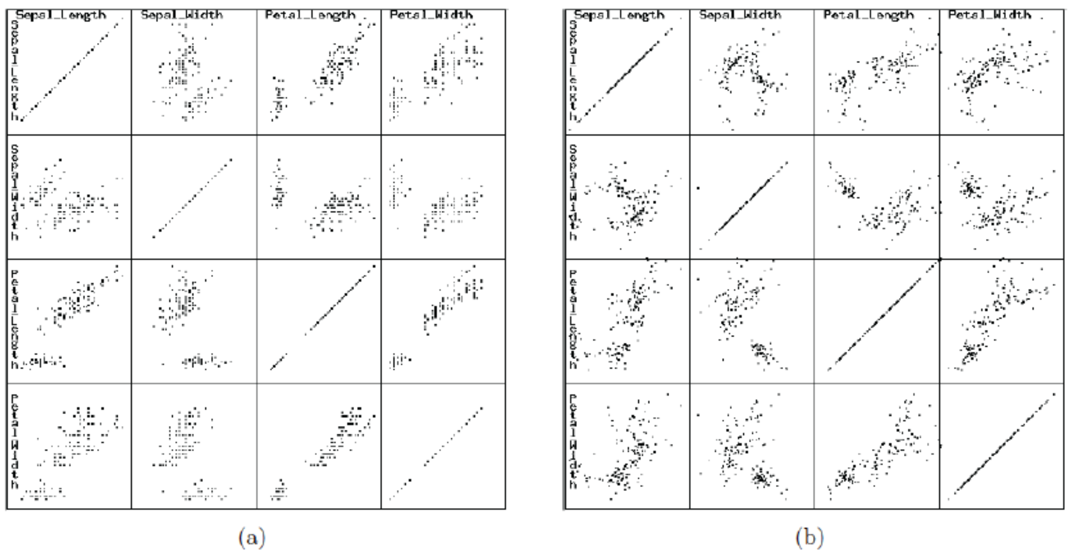


Figure 3.22 Iris dataset illustrating varying clustering levels: (a) the original data and (b) with a 40% added noise factor (Image © 1997 IEEE.)

Results

Over 4000 data points were collected across four key areas: cluster identification, cluster size assessment, outlier identification, and outlier separation estimation. The study found that scatterplot matrices generally performed best, followed by star glyphs and parallel coordinates. Each method exhibited specific strengths and weaknesses:

- **Scatterplot Matrices**:
  - Less effective for overlapping clusters.
  - Less accurate in assessing large cluster sizes.
  - Less effective when outliers fell between clusters.

- **Star Glyphs**:
    - Best for identifying internal outliers (within clusters).
    - Poor for differentiating non-outliers.
    - Effective for conveying outlier separation.
    - Suitable for overlapping clusters and moderate-sized clusters.
- **Parallel Coordinates**:
    - Effective for distinguishing non-outliers.

Significant differences emerged in specific areas. For instance, scatterplots had only 27 false positive cluster identifications, while glyphs and parallel coordinates recorded 126 and 159, respectively. However, correctly identified clusters were similar across techniques (78% for scatterplots, 74% for glyphs, and 71% for parallel coordinates).

Summary

Effective visualizations transform complex data into meaningful insights by following principles of clarity, accuracy, and usability. This unit explored the essential steps in designing visualizations, including selecting appropriate chart types, applying color effectively, and ensuring intuitive mappings. It also addressed common pitfalls, such as misleading visualizations and excessive visual clutter, which can distort data interpretation. Additionally, methods for evaluating visualization effectiveness were discussed, focusing on usability, user needs, and benchmarking. By understanding these principles, learners can create visual representations that enhance data comprehension and decision-making.

Keywords

Data Visualization, Information Design, Chart Types, Data Representation, Visual Perception, Misleading Visuals, Benchmarking, User-Centered Design, Aesthetic Principles, Interactive Visualizations.

Learning Outcomes

Upon completing this unit, learners will be able to:

1. Define and explain the principles of effective visualization.
2. Apply appropriate visualization techniques based on data characteristics.
3. Identify and correct misleading visual elements in charts and graphs.
4. Design visually appealing and informative visualizations using best practices.
5. Evaluate visualization effectiveness through user feedback and benchmarking methods.
6. Optimize visual elements such as labels, legends, and colors to enhance readability and understanding.

Exercises

Objective Questions

1. What is the primary goal of an effective visualization?
   a) To use the most advanced graphical tools
   b) To efficiently and accurately convey information

c) To include as many colors as possible
d) To eliminate the need for data interpretation

2. Which of the following is NOT a common reason why a visualization may fail?
a) Overly complex design
b) Distorted or hidden data
c) Absence of interactive tools
d) Use of ordered attributes for ordered data
3. Why is consistency in design important when mapping data to visualization?
a) It makes the visualization look more attractive
b) It reduces misinterpretation and improves understanding
c) It allows for random color choices
d) It forces users to interpret the data in only one way
4. Which of the following is NOT a recommended method for modifying views in a visualization?
a) Scrolling and zooming
b) Randomly changing color palettes
c) Level-of-detail control
d) Scale control
5. What is a major risk when using color in visualizations?
a) Colors make the visualization look unprofessional
b) Colors can mislead if not chosen carefully
c) Colors have no impact on data interpretation
d) Using too few colors makes the visualization useless
6. Which of the following is NOT a common misleading technique in visualizations?
a) Data scrubbing
b) Unbalanced scaling
c) Using interactive filters
d) Abusing dimensionality
7. Why is range distortion problematic in data visualization?
a) It makes data interpretation easier.
b) It introduces unnecessary data points.
c) It misleads viewers by modifying axis baselines without proper indication.
d) It eliminates redundant information.
8. What is the primary concern with resampling raw data?
a) It always increases accuracy.
b) It can create the illusion of more data than actually exists.
c) It removes important statistical relationships.
d) It reduces computational efficiency.
9. Why should visualizations prioritize relative judgment over absolute judgment?
a) Absolute judgment is more reliable for precise interpretation.
b) Relative judgment helps users compare values more effectively.
c) Relative judgment eliminates the need for graphical elements.
d) Absolute judgment simplifies data visualization.
10. Which factor should be considered when comparing datasets for visualization?
a) The presence of visual effects
b) The availability of multiple color schemes
c) Consistency in measurement units
d) The aesthetic appeal of the visualization

Answers:

1. b) To efficiently and accurately convey information
2. d) Use of ordered attributes for ordered data
3. b) It reduces misinterpretation and improves understanding
4. b) Randomly changing color palettes
5. b) Colors can mislead if not chosen carefully
6. c) Using interactive filters
7. c) It misleads viewers by modifying axis baselines without proper indication.
8. b) It can create the illusion of more data than actually exists.
9. b) Relative judgment helps users compare values more effectively.
10. c) Consistency in measurement units

Short Questions

1. What is the significance of intuitive mappings in data visualization?
2. Why is balancing information density important in visualization design?
3. What is "chart junk," and how can it affect data interpretation?
4. What is "chart junk," and how does it affect the effectiveness of data visualization?
5. Explain the risks associated with using only derived data instead of raw data in a visualization.

Long Questions

1. Explain the importance of aesthetics in visualization and discuss three key design principles that enhance visual appeal.
2. Discuss misleading visualizations and describe three common techniques that can distort data interpretation.
3. Discuss the key problems that can arise when designing effective visualizations. Include examples of misleading visualization techniques and how they can distort data interpretation. Also, explain the importance of maintaining transparency in visual representation.

Exercises

1. Identify at least three issues present in the visualization depicted in Figure 3.11
2. Provide suggestions for improving each of the visualizations shown in Figure 3.11.
3. Create a table that outlines the advantages and disadvantages of different evaluation methods for visualization tools. You may need to review some of the suggested literature to gather insights. Consider whether certain strategies complement each other, where the strengths of one method offset the weaknesses of another. Identifying such pairings could provide a more comprehensive assessment of a visualization technique than relying on a single evaluation approach.
4. Compare qualitative and quantitative evaluation methods for visualization tools. In what scenarios would each approach be more suitable? Provide examples of situations where a combination of both methods might offer a more comprehensive assessment.

Collaborative Learning Tasks

Task 1: Evaluating Visualization Effectiveness

Instructions**:**

1. Form small groups and choose a type of visualization (scatterplot, bar chart, heatmap, etc.).
2. Use the provided real-world dataset (e.g., climate data, stock market trends) to assess how well the visualization represents the information.
3. Evaluate your assigned visualization based on clarity, accuracy, aesthetics, and any potential for misleading the audience.
4. Discuss within your group and prepare a short presentation on your findings. Include suggestions for improvement.

Task 2: Designing an Interactive Dashboard

Instructions**:**

1. Form teams and choose a dataset related to social, environmental, or economic issues.
2. Design a dashboard using D3.js or another visualization tool, incorporating interactive elements like zooming, filtering, and color mapping.
3. Ensure your design makes the data easy to interpret and provides insights for users.
4. Present your dashboard to the class, explaining your design choices and how they enhance user experience.

Task 3: Identifying and Fixing Misleading Visualizations

Instructions**:**

1. Work in groups to analyze a set of misleading visualizations provided to you. These may include improper scaling, manipulated axes, or excessive visual clutter.
2. Identify why the visualization is misleading and propose a revised version that presents the data more accurately.
3. Create a before-and-after comparison and explain your modifications.
4. Share your revised visualizations with the class and discuss their impact on audience interpretation.

Task 4: Identifying and Rectifying Misleading Visualizations

Instructions**:**

1. Form groups of 3-4 members.
2. Each group will find three examples of misleading visualizations from online sources, research papers, or reports.
3. For each visualization, analyze the following:
    - What makes the visualization misleading?
    - What data distortion techniques are used (e.g., truncated axes, improper scaling, omitted data)?
    - What impact could the misleading visualization have on decision-making?
4. Propose a corrected version of each visualization using appropriate design principles.
5. Present the findings to the class, explaining the corrections made.

Case Studies

Case Study 1: Misleading COVID-19 Graphs

Scenario: During the COVID-19 pandemic, many graphs published in the media misrepresented infection trends due to inconsistent scaling, omitted baselines, or distorted Y-axis intervals.

Discussion Questions:

- What elements in these graphs make them misleading?
- How could the data have been represented more accurately?
- Why is ethical data visualization important in public health communication?

Task: Analyze examples of misleading COVID-19 graphs, discuss your observations in a group, and suggest improvements.

Case Study 2: The Impact of Color in Data Representation

Scenario: A weather forecasting agency tested different color schemes to represent temperature data. One version used a high-contrast palette (red for high, blue for low), while another used a single hue with varying saturation levels.

Discussion Questions:

- Which color scheme do you think is more effective in conveying temperature variations?
- How does color perception influence data interpretation?
- What considerations should be made to ensure accessibility for colorblind users?

Task: Compare the color schemes and suggest improvements for better readability and accessibility.

Case Study 3: Analyzing a Failed Business Strategy through Data Visualization

Scenario:

A retail company launched an aggressive discount campaign to increase sales. The marketing team created a series of visual reports showing increased sales volume. However, after six months, the company faced severe financial losses, and executives discovered that the discount strategy was not sustainable.

Task:

1. Analyze the provided sales and financial data visualizations.
2. Identify potential visualization flaws that may have contributed to poor decision-making.
3. Discuss how more effective data presentation could have prevented the financial loss.
4. Propose improved visualizations to present a clearer picture of profitability.

Study Tips

1. **Understand the Basics First** – Start with fundamental concepts like types of data (quantitative vs. categorical), visualization techniques, and chart selection.
2. **Learn by Doing** – Practice coding visualizations using D3.js or other tools. Modify existing charts to see how changes impact data representation.
3. **Analyze Existing Visualizations** – Look at charts from news articles, research papers, and dashboards. Identify their strengths and weaknesses.
4. **Focus on Design Principles** – Study color theory, layout, and accessibility to create clear, effective visualizations.
5. **Work on Real-World Datasets** – Use open datasets (e.g., climate data, financial data) to create meaningful visualizations.

Self-Assessment Tasks

1. Analyze an existing data visualization and identify its strengths and weaknesses.
2. Select a dataset and determine the most suitable chart type for representing it. Justify your choice.
3. Create a scatter plot using any visualization tool and evaluate its effectiveness in representing relationships between variables.
4. Compare different color schemes used in data visualization and discuss their impact on readability and interpretation.
5. Reflect on the role of interactivity in data visualization and provide examples of how it enhances user experience.

3.8 FURTHER READINGS

- Matthew O. Ward, Georges Grinstein, and Daniel Keim. Interactive data visualization: foundations, techniques, and applications. AK Peters/CRC Press, 2nd Edition, 2015.
- Ben Fry, Visualizing data: Exploring and explaining data with the processing environment. " O'Reilly Media, Inc.", 1st Edition, 2007.
- Colin Ware, Information visualization: perception for design. Morgan Kaufmann, 2019.

3.9 NPTEL VIDEOS

Introduction to Classification case study – Python for Data Science - https://youtu.be/B2J1Gtyn02U?si=vjWyENZnhr_RBsWC

Case study on Regression – Python for Data Science - https://youtu.be/g4NKNvUUPiE?si=P1olay6vQdfCbj08

References

The contents of this chapter have been prepared based on the reference book:

Matthew O. Ward, Georges Grinstein, and Daniel Keim. Interactive data visualization: foundations, techniques, and applications. AK Peters/CRC Press, 2nd Edition, 2015.

All the figures have been referred from this text book.

CONTENTS

UNIT 4

INFORMATION DASHBOARD DESIGN

4.1 Introduction

Dashboards play a crucial role in data visualization by providing a centralized interface to monitor, analyze, and interpret key information effectively. A well-designed dashboard enables users to gain insights quickly, facilitating informed decision-making across various domains. This unit explores the principles of information dashboard design, covering essential strategies, display media, and graphical elements that contribute to clear and efficient communication of data.

4.2 Learning Objectives

- Understand the fundamental concepts of data dashboards and their significance.
- Identify key features and characteristics of an effective dashboard.
- Apply best practices in dashboard design to enhance clarity and usability.
- Evaluate and select appropriate visual elements such as graphs, icons, and tables for effective information display.
- Organize and structure information to ensure meaningful and actionable insights.

4.3 Overview

Information dashboards serve as essential tools for data monitoring and analysis, allowing users to track performance indicators, trends, and critical metrics. This unit provides an in-depth understanding of dashboard components, covering various graph types, icons, text, and other visual elements used in their design. Additionally, the unit delves into strategies for optimizing dashboard layouts, ensuring information is well-structured and easily interpretable. The importance of maintaining consistency, enhancing usability, and incorporating interactivity in dashboards is also explored.

4.4 Introduction to Data Dashboards

A data dashboard is a tool used for visualizing key performance indicators (KPIs) and other important metrics. Similar to an automobile dashboard that displays speed, fuel level, and other vital information, a data dashboard provides decision-makers with a quick overview of an organization's performance. These dashboards consolidate and present data from various sources in a structured manner, allowing organizations to monitor operations, track progress, and make informed decisions efficiently.

A **data dashboard** can be defined as an interactive and graphical representation of key data points, metrics, and trends. It consolidates information from multiple sources into a single view, allowing users to monitor performance, identify patterns, and make data-driven decisions. Dashboards often include elements like charts, graphs, tables, and filters, enabling users to explore data dynamically.

4.4.1 Key Features of a Data Dashboard:

- **Real-time or Periodic Updates** – Displays the latest data for accurate decision-making.
- **Interactivity** – Includes filters, slicers, or drill-down options for deeper analysis.
- **Multiple Data Visualizations** – Uses bar charts, line graphs, pie charts, heatmaps, etc.
- **Customizability** – Can be tailored to focus on specific KPIs (Key Performance Indicators).
- **User-Friendly Interface** – Designed for easy interpretation without technical expertise.

4.5 Strategies for Effective Dashboard Design

Simplicity is the fundamental principle of dashboard design. The data should be displayed as clearly as possible while avoiding unnecessary decorations. A well-designed dashboard effectively presents large amounts of diverse information within a limited space without losing clarity.

4.6 Characteristics of a Well-Designed Dashboard

The main challenge of dashboard design is to fit a large volume of meaningful data into a single screen while maintaining clarity. Similar to an aircraft cockpit, dashboards must be designed carefully to ensure all essential information is visible at a glance. However, unlike cockpit design, many dashboard creators lack formal design training. Proper dashboard design can help businesses avoid costly miscommunications and inefficiencies.

The concept of simplification is crucial, as eloquent communication is often achieved through clarity. Dashboards should focus on presenting essential information in a condensed and well-organized manner using concise visual mechanisms. A well-designed dashboard should:

1. Be exceptionally well-organized.
2. Provide condensed information, primarily through summaries and exceptions.
3. Be specific to and customized for the audience and objectives.
4. Use concise media that clearly communicate data and messages, even in small spaces.

Dashboards should provide an overview of key metrics, similar to a car dashboard, which displays speed, fuel level, and other vital indicators. However, they should not attempt to present every detail, as too much information can reduce readability. Instead, dashboards should highlight critical data and allow users to access further details when necessary.

4.6.1 Condensing Information via Summarization and Exception

To fit large datasets into a dashboard, information should be condensed using summarization and exceptions:

1. **Summarization -** This involves reducing a large dataset into key values, such as sums and averages. Other statistical measures, such as distribution and correlation, may also be used when necessary.

2. **Exceptions** - Instead of displaying all data points, dashboards should highlight only those values that fall outside the normal range, signaling potential problems or opportunities.

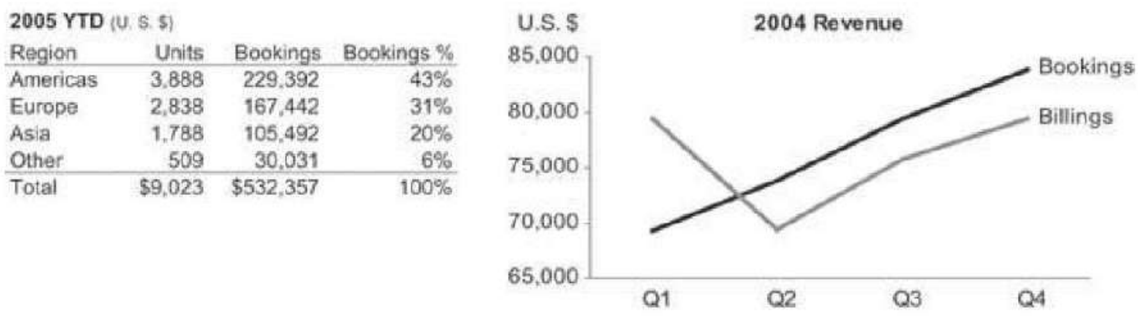
Customization is essential for effective dashboards. The displayed information should align with the audience's needs and terminology. Data should also be expressed with appropriate precision to avoid overwhelming users with unnecessary details. For example, financial executives may only need rounded values, whereas accountants might require exact figures.

The chosen display mechanism must communicate data effectively in the least amount of space. While decorative elements may seem visually appealing, they should not compromise clarity. The selection of display media should be guided by two principles:

- The display should be the best way to present a particular type of information found in dashboards.
- The display must remain effective even when scaled down to fit within the dashboard's limited space.

4.7 Key Goals in the Visual Design Process

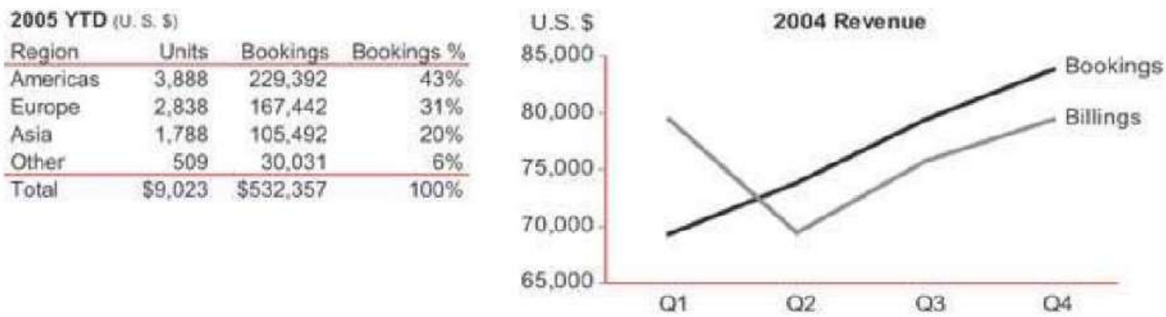
Edward R. Tufte's concept of the **data-ink ratio** is useful for dashboard design. In printed graphics, some ink represents data, while the rest serves non-data purposes. The goal is to maximize the proportion of ink that conveys meaningful information. This principle can be adapted to digital dashboards by minimizing non-data pixels. Figure 4.1 presents two representations of quantitative data: one as a table and the other as a graph. Both are designed with minimal non-data ink to ensure clarity. In Figure 4.2, the same table and graph are shown again, but this time, the non-data ink is highlighted in red. Non-data pixels, such as unnecessary decorations, borders, and grid lines, should be minimized to enhance readability. Take a close look at the dashboard in Figure 4.3 and identify elements that do not contribute to the actual data representation.



2005 YTD (U. S. $)

Region	Units	Bookings	Bookings %
Americas	3,888	229,392	43%
Europe	2,838	167,442	31%
Asia	1,788	105,492	20%
Other	509	30,031	6%
Total	$9,023	$532,357	100%

Figure 4.1 A table and a graph showing both data and non-data ink.



2005 YTD (U. S. $)

Region	Units	Bookings	Bookings %
Americas	3,888	229,392	43%
Europe	2,838	167,442	31%
Asia	1,788	105,492	20%
Other	509	30,031	6%
Total	$9,023	$532,357	100%

Figure 4.2 The same table and graph with non-data ink highlighted in red.

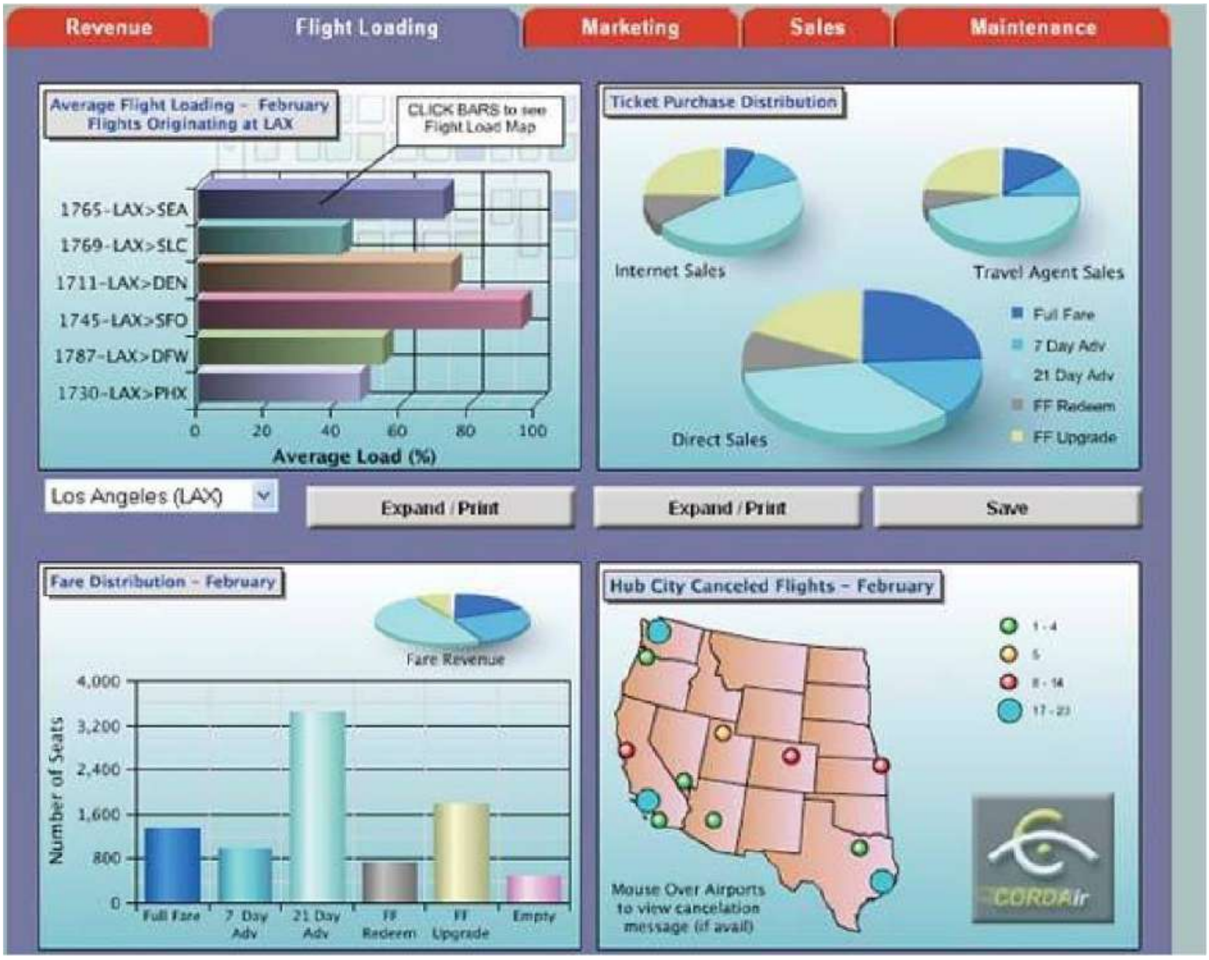


Figure 4.3 This dashboard displays an excessive amount of non-data pixels.

Tufte's principle states:

- Data-ink ratio = data-ink / total ink used in the graphic
- The goal is to maximize the data-ink ratio within reason, ensuring that every visual element contributes to meaningful communication.

Applying this to dashboards, the primary goals in visual design are:

- Reduce non-data pixels - Eliminate unnecessary visual elements.
- Enhance data pixels - Make essential data stand out clearly.

4.7.1 Reducing Non-Data Pixels

The process of minimizing non-data pixels involves two steps:

1. **Eliminate unnecessary non-data pixels** - Remove decorative elements that do not enhance communication.
2. **De-emphasize and regularize the remaining non-data pixels** - Ensure necessary non-data elements do not overshadow the data itself.

Eliminating Unnecessary Non-Data Pixels

Dashboard design is an iterative process, requiring multiple revisions to achieve optimal clarity. Some common non-data elements that should be eliminated include:

- **Decorative graphics** - Remove elements that serve no informational purpose.
- **Unnecessary color variations** - Avoid using different colors unless they convey meaning.
- **Borders** - Replace with white space when possible.
- **Fill colors** - Use neutral backgrounds instead of color-coded sections.
- **Gradient backgrounds** - Replace with solid colors.
- **Grid lines in graphs** - Remove when they do not enhance readability.
- **Grid lines in tables** - Use white space instead of grid lines.
- **Alternating row colors in tables** - Only use if necessary for readability.
- **Complete borders around graphs** - Use axes instead of enclosing the data region.
- **3D effects in graphs** - Avoid unless the third dimension represents actual data.
- **Unnecessary visual embellishments** - Ensure all elements contribute to data clarity.

By following these principles, dashboards can effectively convey critical information while remaining visually clear and easy to interpret.

Deemphasizing and Regularizing Non-Data Pixels

Not all non-data pixels can be eliminated without affecting the usability of a dashboard. Some are necessary for structure, organization, or readability. When data is densely packed, elements like lines or fill colors may be required to separate sections when white space alone is insufficient. Instead of removing these essential non-data pixels, they should be visually muted so that they do not draw attention. The focus should always remain on the data itself, making the dashboard's design nearly invisible. Non-data pixels should be just visible enough to serve their purpose but no more.

Examples of useful non-data pixels include:

1. **Axis lines in graphs** - These help define the data region but should be subtle.
2. **Lines, borders, or fill colors** (**Figure 4.4**) - Used when white space is not enough to separate sections but should be as minimal as possible.
3. **Grid lines in graphs** - Only necessary when they help compare specific data points.
4. **Grid lines and fill colors in tables** - Useful when white space alone does not adequately separate columns or rows.
5. **Fill colors in alternating table rows** - Helpful for scanning across rows but should be applied lightly.

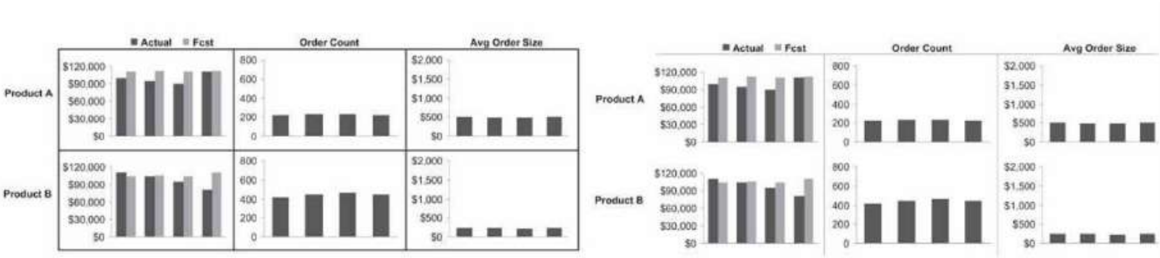


Figure 4.4 Thin lines can effectively separate adjacent sections of a display while maintaining a minimal visual impact.

The visibility of non-data pixels can be reduced using light, low-saturation colors (e.g., light gray) and thin stroke weights. Additionally, regularizing non-data pixels, such as keeping axis lines uniform across all graphs, prevents unnecessary distractions.

Navigation and data selection elements, such as buttons and dropdowns, are also non-data pixels. While important, they should not dominate the dashboard. Placing them in less prominent areas, such as the bottom-right corner, and muting their visual prominence ensures they do not compete with data. **Figure 4.5** highlights a dashboard where excessive space is given to navigation elements. Similarly, instructions that users only need once should not take up prime dashboard space; instead, they can be placed on a separate screen or as pop-ups when necessary (**Figure 4.6**).

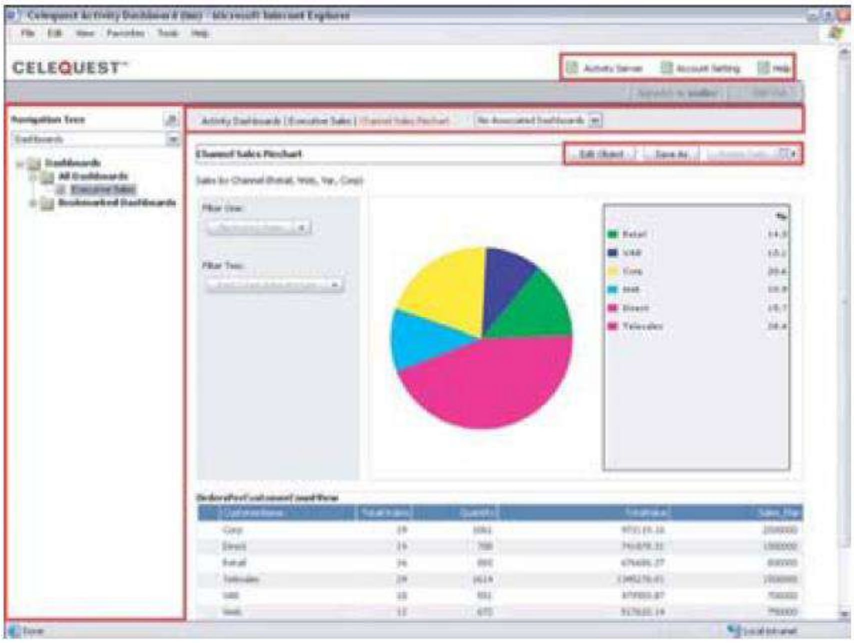


Figure 4.5 This dashboard allocates excessive space and emphasis to navigational and data selection controls.

Figure 4.6 The area highlighted in red shows that this dashboard occupies valuable space for instructions that could have been presented separately, such as through a pop-up menu or an additional screen.

4.7.2 Enhancing Data Pixels

After reducing non-data pixels, the next step is to enhance the data pixels. This involves two key actions:

1. Eliminate unnecessary data pixels
2. Highlight the most important data pixels that remain

Eliminating Unnecessary Data Pixels

A dashboard should not be overloaded with every possible piece of information. While it may seem beneficial to include as much data as possible, doing so can reduce clarity. Instead, focus only on essential information. Removing unnecessary data naturally increases the prominence of what remains.

Unnecessary data can be reduced through:

- **Summarization**: Instead of displaying detailed transaction-level sales data, consider summarizing it at a relevant level (e.g., per quarter, per region, or per month).

- **Managing by Exception**: Only show data when an issue or opportunity arises. For instance, displaying staff expenses only when they exceed a threshold avoids unnecessary clutter.
- **Multi-foci displays**: Useful for historical data, where recent information is more detailed, while older data is summarized (**Figure 4.7**). For example, showing daily data for the current month, monthly data for the past year, and yearly data for previous years provides relevant context without overwhelming detail.

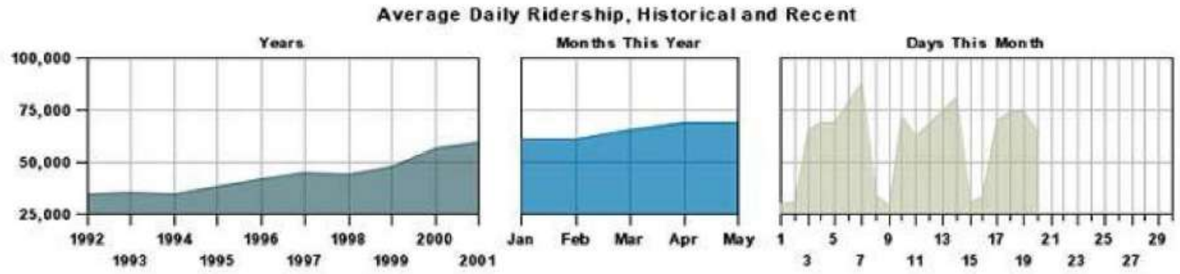


Figure 4.7 These three time-series graphs represent public transportation rider statistics at three levels of detail: daily for the current month, monthly for the current year, and yearly for the past 10 years.

Highlighting Important Data Pixels

All data on a dashboard is important, but some data is more critical than others. Key data falls into two categories:

1. **Always important information -** Data that should consistently be emphasized.
2. **Contextually important information -** Data that requires attention only in specific situations, such as alerts.

Different techniques are used to highlight these categories:

- **Static highlighting**: Always-important data should be placed in high-visibility areas. Figure 4.8 shows that the top-left and center regions naturally attract the most attention. Figure 4.9 illustrates how valuable dashboard space should not be wasted on logos or decorative elements.

- **Dynamic highlighting**: Contextually important data should be emphasized dynamically using visual attributes. Visual attributes, other than location on the screen, can be effectively manipulated in a dynamic manner to highlight critical information on a dashboard. These attributes can be used to emphasize data that is important only at specific times or to highlight consistently significant data when prime screen locations are already occupied.

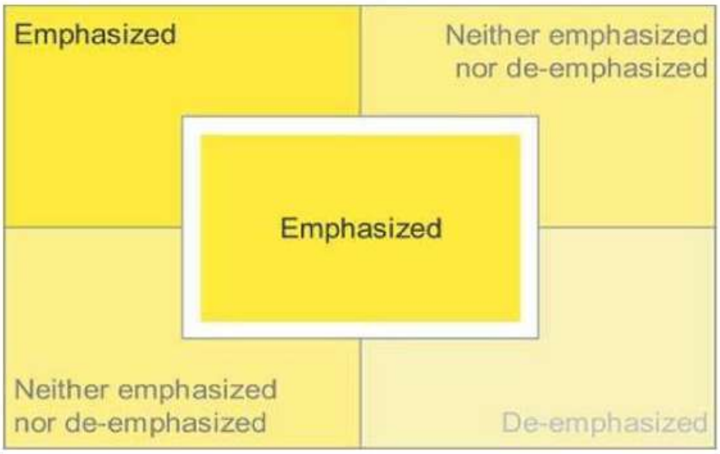


Figure 4.8 Various regions of a dashboard carry different levels of visual emphasis, influencing how information is perceived.

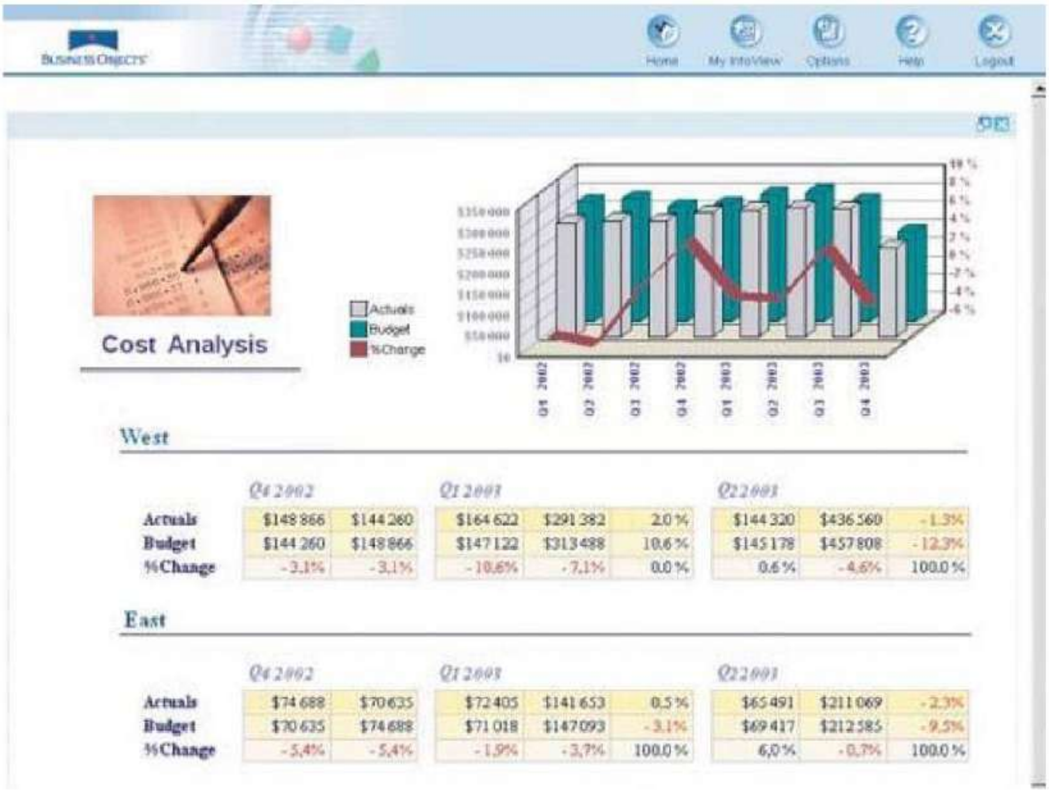


Figure 4.9 The most important space on this dashboard is occupied by a company logo and unnecessary decorative elements.

Effective Use of Visual Attributes

Several visual attributes can be utilized dynamically to draw attention to important data. Two key approaches to highlighting information include:

- Using expressions of visual attributes that are greater than the norm, such as increased brightness or darker colors.
- Using expressions that contrast with the norm, such as blue text when the standard is black or gray.

Contrast, rather than magnitude, is often enough to make information stand out, as human visual perception is highly sensitive to differences and seeks to assign meaning to them.

Some visual attributes naturally appear stronger or more intense when emphasized. These include:

- **Color Intensity**: A darker or more saturated hue is perceived as more significant than a lighter or less saturated version.
- **Size**: Larger elements stand out more prominently than smaller ones.
- **Line Width**: Thicker lines draw more attention than thinner lines.

Other attributes create emphasis by differing from the surrounding norm. These include:

- **Hue**: Any hue distinct from the norm will be noticeable.
- **Orientation**: Elements positioned differently from the norm stand out.
- **Enclosure**: Items enclosed by borders or surrounded by a fill color become prominent.
- **Added Marks**: Additional symbols, such as circles or checkmarks, placed next to data points enhance visibility.

Color is a particularly effective means of highlighting due to its ease of manipulation in dashboard software. It allows for dynamic adjustments based on predefined data conditions.

A useful technique involves using an added mark with a distinct color to draw attention to text-based information. For instance, symbols such as circles, checkmarks, or asterisks placed next to critical items enhance visibility. Additionally, varying the intensity of a single color to represent different levels of importance or urgency is more effective than using multiple colors, as this method remains accessible to those with colorblindness.

Figure 4.10 illustrates this principle by demonstrating how simple symbols and varying color intensities can dynamically highlight data.

Metric	Actual	Variance
Revenue	$913,394	+$136,806
Profit	$193,865	-$73,055
Avg Order Size	$5,766	-$297
On Time Delivery	104%	+4%
New Customers	247	-62
Cust Satisfaction	4.73 / 5	+0.23

Figure 4.10 Basic symbols combined with different color intensities can be used to dynamically emphasize data.

4.8 Effective Dashboard Display Media

Dashboards must consolidate a vast amount of information onto a single screen, ensuring quick readability without compromising clarity or omitting essential details. To achieve this, they must employ display media that effectively communicate data within limited space. Every section of a dashboard should use the most suitable and efficient display medium, selected, customized, or even created specifically for dashboards.

4.9 Selecting the Best Display Medium

The choice of display medium depends on the nature of the information, the intended message, and the audience's needs. Since dashboards present various types of data, multiple display media should be utilized appropriately. The primary question to address is whether to use text, graphics, or a combination of both.

Text is processed sequentially, word by word. While individual reading speeds vary, text is best suited for conveying precise numerical information. For example, presenting a year-to-date (YTD) expense of **$487,321** is most effectively done with text rather than graphics. A simple table like the following suffices:

Table 4.1 Simple table

YTD Expenses	$487,321

When additional context is necessary, such as comparing actual expenses against a target, structured text remains effective:

Table 4.2 Table showing comparison

	Actual	**Target**
YTD Expenses	$487,321	$450,000

While textual representation is excellent for looking up specific values, it does not effectively convey trends or patterns. For instance, to observe changes over time, a graphical representation like a line chart is more suitable.

A predominantly text-based dashboard, as shown in Figure 4.11, is useful for quick lookups but limits comparative analysis. To improve usability, tables should be structured to facilitate scanning, as demonstrated in Figure 4.12, where measures are grouped logically. Replacing color coding with text formatting such as bold, black, or gray ensures accessibility for color-blind users.

A well-designed dashboard should integrate both text and graphics to balance precision with visual clarity. The appropriate choice depends on what needs to be conveyed, how the data will be used, and how efficiently it can be interpreted.

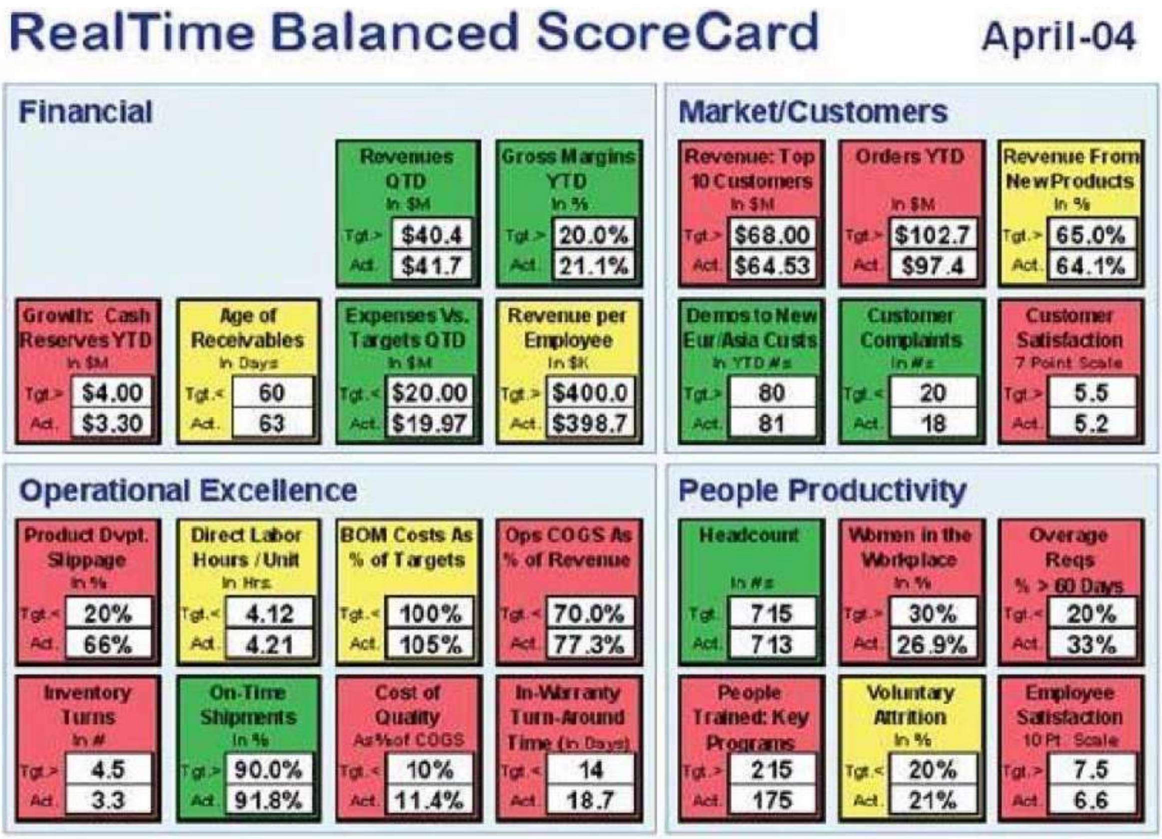


Figure 4.11 A predominantly text-based dashboard

RealTime Balanced Scorecard (as of April 4, 2003)

Financial

Metric	Actual	Target
Revenue QTD (millions)	$41.7	$40.4
Gross Margins YTD (%)	21.1%	20.0%
Growth: Cash Reserves YTD (millions)	**$3.3**	**$4.0**
Age of Receivables (days)	63	60
Expenses vs. Targets QTD (millions)	$20.0	$20.0
Revenue per Employee (thousands)	$398.7	$400.0

Market Customers

Metric	Actual	Target
Revenue: Top 10 Customers (millions)	**$64.5**	**$68.0**
Orders YTD (millions)	**$97.4**	**$102.7**
Revenue from New Products (%)	64.1%	65.0%
Demos to New Eur/Asia Customers (count)	81	80
Customer Complaints (count)	18	20
Customer Satisfaction (7-point scale)	**5.2**	**5.5**

Operational Excellence

Metric	Actual	Target
Product Development Slippage (%)	**66.0%**	**20.0%**
Direct Labor Hours per Unit (hours)	4.21	4.12
BOM Costs as % of Target (%)	105%	100%
Ops COGS as % of Revenue (%)	**77.3%**	**70.0%**
Inventory Turns (count)	**3.3**	**4.5**
On-Time Shipments (%)	91.8%	90.0%
Cost of Quality as % of COGS (%)	**11.4%**	**10.0%**
In-Warranty Turnaround Time (days)	**18.7**	**14.0**

People Productivity

Metric	Actual	Target
Headcount (count)	713	715
Women in the Workplace (%)	**26.9%**	**30.0%**
Overage Requisitions > 60 days (%)	**33%**	**20%**
People Trained: Key Programs (count)	**175**	**215**
Voluntary Attrition (%)	21%	20%
Employee Satisfaction (10-point scale)	**6.6**	**7.5**

Figure 4.12 Redesigned version of the text-based dashboard in Figure 4.8, structured into tables for improved data lookup.

4.10 An Ideal Library of Dashboard Display Media

After determining whether to use text, graphics, or both, the next step is selecting the most appropriate graphical format. A poorly chosen graph can obscure rather than clarify data. The ideal library of dashboard display media consists of six categories:

- Graphs
- Images
- Icons
- Drawing objects
- Text
- Organizers

4.11 Graphs

Given that most dashboard data is quantitative, graphs are the predominant display medium. The following types of graphs are particularly effective:

- Bullet graphs
- Bar graphs (horizontal and vertical)
- Stacked bar graphs
- Combination bar and line graphs
- Line graphs
- Sparklines
- Box plots
- Scatter plots
- Treemaps

4.11.1 Bullet Graphs

Bullet graphs, designed specifically for dashboards, provide a compact alternative to traditional gauges and meters. Unlike circular gauges, which consume excessive space and often lack clarity, bullet graphs maximize space efficiency while maintaining clear data representation.

Figure 4.13 shows a simple bullet graph, which consists of the following components:

1. A bar representing the actual value
2. A comparative marker (such as a target)
3. Background shading to indicate qualitative ranges (e.g., good, satisfactory, poor)

Figure 4.14 shows the same bullet graph, this time with each of its components identified.

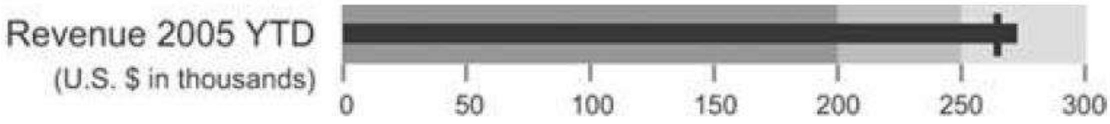


Figure 4.13 simple horizontally oriented bullet graph

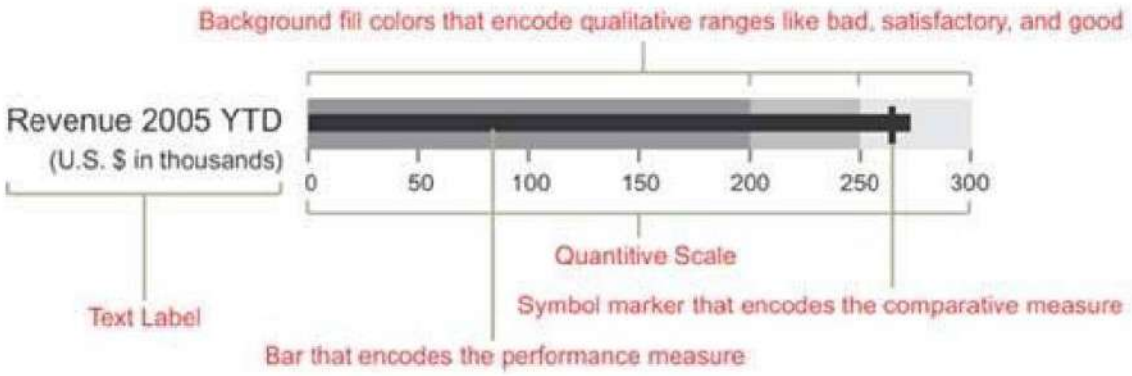


Figure 4.14 A simple bullet graph with each of its components labeled

Bullet graphs can also accommodate multiple comparative measures. Figure 4.15 demonstrates a version with two comparative markers, differentiated by stroke weight.

To ensure accuracy, the scale of a bullet graph should always start at zero. If the key measure is best represented by something other than a bar, alternative visualization methods should be explored while maintaining clarity and efficiency.

Dashboards must utilize display media that maximize clarity and insight while conserving space. The choice of text or graphics—and the specific graphical medium selected—should always be dictated by the data's purpose and the audience's needs.

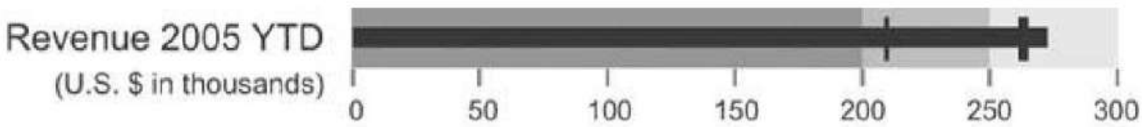


Figure 4.15 This bullet graph presents two comparisons, differentiated visually by varying stroke weights.

4.11.2 Bar Graphs

Bar graphs are designed to display multiple instances of one or more key measures. Unlike bullet graphs, which show a single instance, bar graphs effectively present data related to categories such as regions or departments. A typical example, as seen in Figure 4.16, illustrates two key measures—bookings and billings revenue—segmented by sales regions.

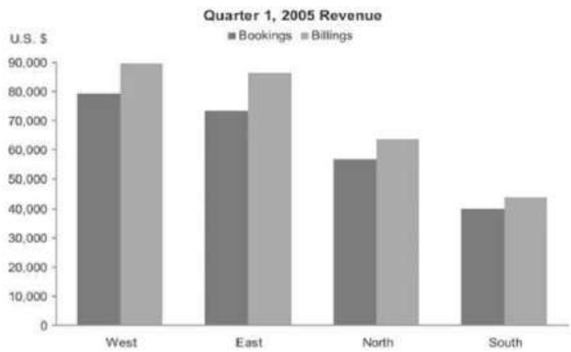


Figure 4.16 A typical bar graph.

The term "bar graph" refers to all graphs that use bars to encode data, regardless of whether they are oriented vertically or horizontally. To decide whether to use bars or lines in a graph, it is essential to understand three types of categorical scales:

1. **Nominal Scales:** These consist of discrete items that belong to a common category but do not relate in a specific order. Examples include regions (e.g., The Americas, Asia, and Europe) and departments (e.g., Sales, Marketing, and Finance).
2. **Ordinal Scales:** These contain items with an inherent order but do not represent quantitative values. Examples include rankings such as "A, B, C" and "poor, average, excellent."
3. **Interval Scales:** These include ordered items that represent quantitative values. They transform a continuous range into smaller sequential ranges of equal size, such as income brackets from 55 to 80 grouped into five equal ranges.

Figure 4.17 illustrates each type of categorical scale.

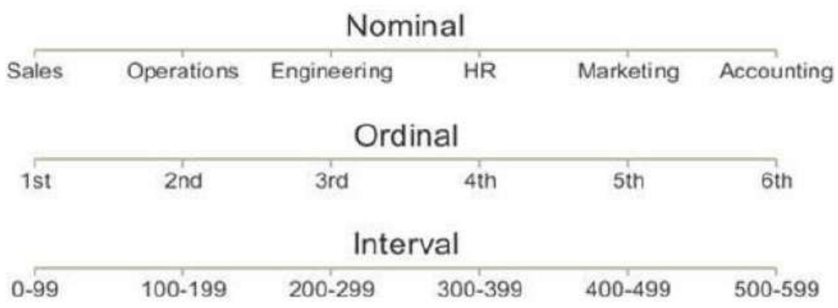


Figure 4.17 The three types of categorical scales found in graphs

Bar graphs should always be used instead of line graphs for data along a nominal or ordinal scale. The weight of bars emphasizes individual values, making comparisons easy by assessing bar heights. In contrast, line graphs highlight overall trends by connecting values, which is suitable only for interval scales.

Line graphs work well for interval scale data, but sometimes bar graphs are preferable when emphasizing individual values, adjacent comparisons, or part-to-whole relationships. Figure 4.18 contrasts the same interval data displayed as both a bar graph and a line graph, showing the differing emphasis of each.

Since bar graphs emphasize individual values, they facilitate easy comparisons between adjacent values. Even when depicting part-to-whole data, bar graphs are superior to pie charts.

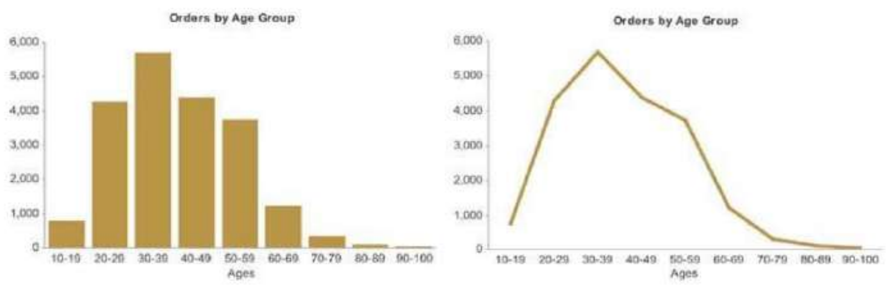


Figure 4.18 These two graphs—a bar graph and a line graph—represent the same data but emphasize different aspects of it.

4.11.3 Stacked Bar Graphs

Stacked bar graphs, a variation of bar graphs, can be useful but are often misused. They should not be used to display a single series of part-to-whole data, as seen in Figure 4.19. A standard bar graph is more effective for this purpose.

Stacked bar graphs are beneficial when showing multiple instances of a whole and its parts, with an emphasis on the whole. Figure 4.20 provides an example of quarterly sales revenue subdivided by sales channel. However, changes in the distribution are difficult to discern, except for the segment at the bottom of the bars. For greater clarity, two separate graphs—one for the whole and one for the parts—or a combination bar and line graph can be used, as shown in Figure 4.21.

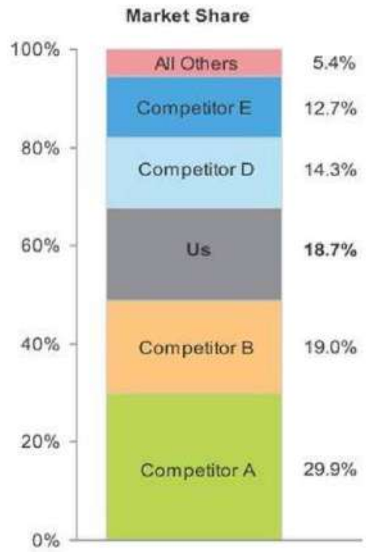


Figure 4.19 A stacked bar graph is not the best way to display a single series of part-to-whole data.

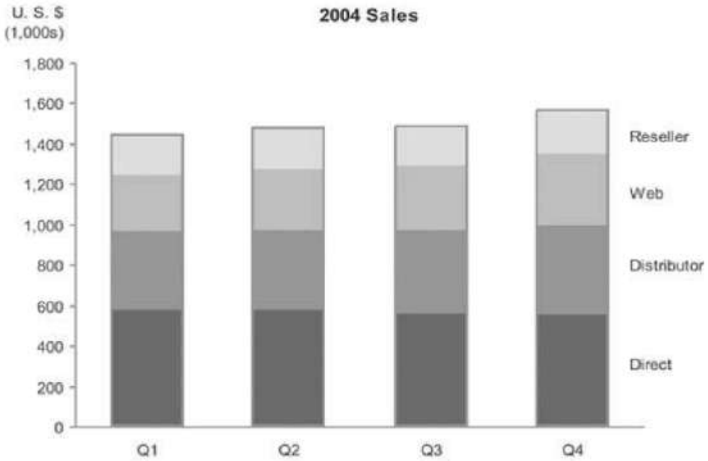


Figure 4.20 A stacked bar graph is only useful when displaying multiple instances of a whole (e.g., total sales) and its components (e.g., sales by channel) across different periods (e.g., each quarter), with a stronger focus on the whole rather than its parts.

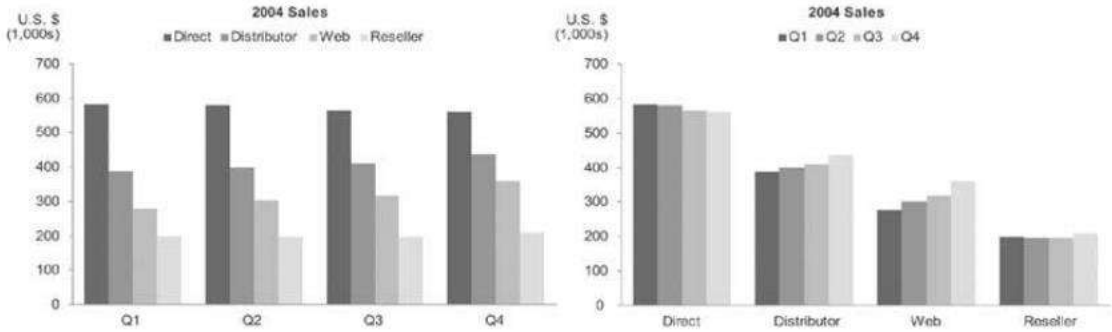


Figure 4.21 These bar graphs illustrate changes in the distribution of sales across the four channels more clearly than the stacked bar graph in Figure 4.17

4.11.4 Combination Bar and Line Graphs

Bars and lines should only be combined when some data require bars (emphasizing individual values) and others require a line (emphasizing trends). A common use case is displaying revenues and expenses with bars while showing profit trends with a line, as seen in Figure 4.22.

A combination bar and line graph can also effectively display part-to-whole relationships over time. Figure 4.23 shows quarterly revenue by sales channel with a line for total revenue, using separate quantitative scales for bars and the line to optimize space.

An exception to the rule against using lines on non-interval scales is the Pareto chart. This chart, named after Vilfredo Pareto, combines bars to show individual values and a line to depict cumulative totals. Figure 4.24 demonstrates a Pareto chart ranking sales representatives by revenue, with cumulative sales forming a meaningful, connected sequence. This visualization effectively illustrates the 80:20 rule, which suggests that a small portion of contributors often account for the majority of the outcome.

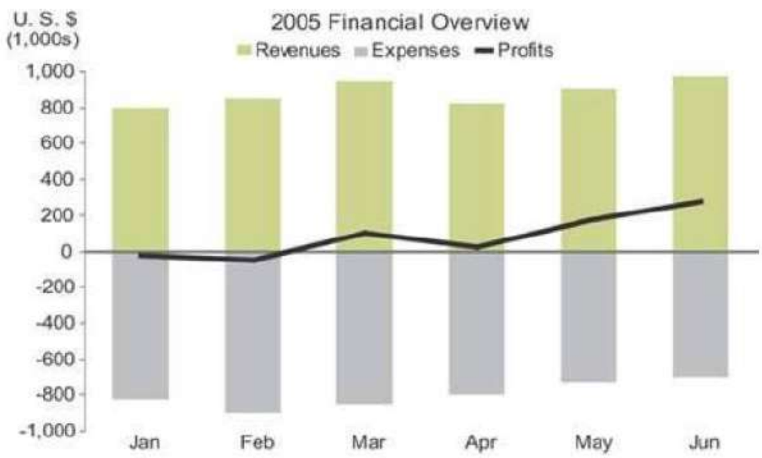


Figure 4.22 This graph uses bars and a line to emphasize monthly revenues and expenses while also showcasing the overall profit trend.

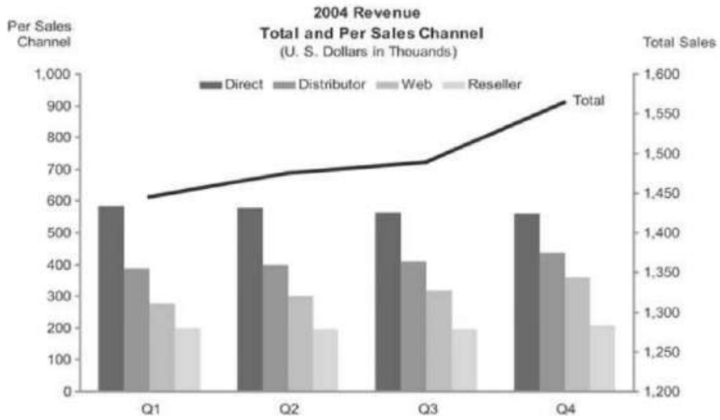


Figure 4.23 This combination bar and line graph represents quarterly revenue by sales channel using bars and total revenue using a line.

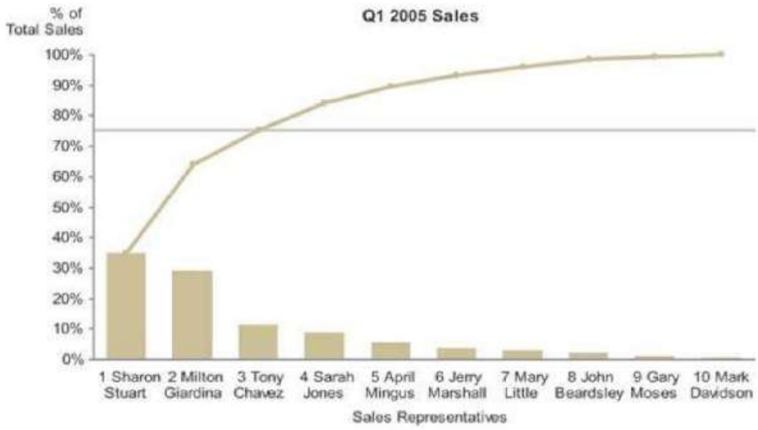


Figure 4.24 This Pareto chart represents sales revenue by sales representatives using bars and cumulative revenue using a line.

4.11.5 Line Graphs

Line graphs are excellent for illustrating patterns in data, such as trends, fluctuations, and relationships between datasets. They are particularly useful in dashboards for quickly conveying the overall movement of time-series data. Figure 4.25 contrasts the same time-series data in a bar graph and a line graph, highlighting how line graphs better reveal data trends. Unlike bar graphs, line graphs do not need to start from zero and can focus on the relevant data range for more detail.

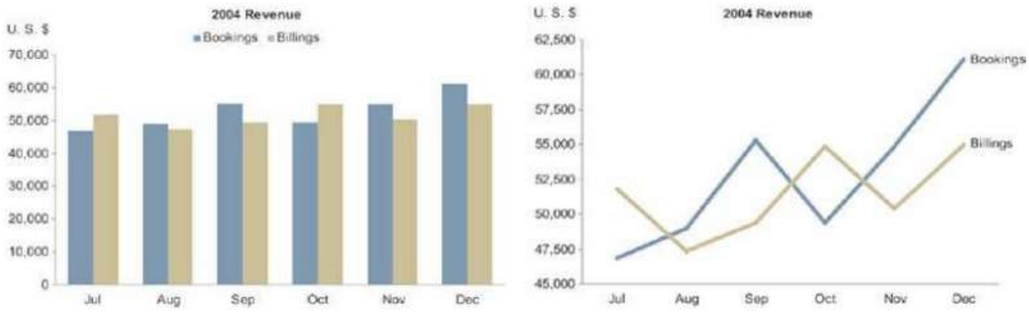


Figure 4.25 Two graphs display the same time-series data: a bar graph on the left and a line graph on the right. The line graph makes the overall data pattern more visible.

4.11.6 Sparklines

Sparklines, introduced by Edward R. Tufte, provide compact and efficient visualizations of time-series data. **Figure 4.26** presents an example of a sparkline showing a 12-month history of an account balance.

Tufte describes sparklines as "data-intense, design-simple, word-size graphics." They are ideal for dashboards and reports that require condensed data visualization. Unlike standard graphs, sparklines omit the quantitative scale, focusing instead on historical context. This makes them suitable for quick assessments rather than precise numerical comparisons.

Sparklines can be enhanced with additional elements, such as light gray rectangles to highlight specific values, as seen in **Figure 4.27**.

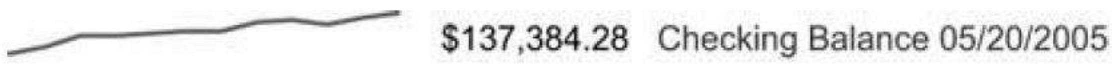


Figure 4.26 A simple sparkline that displays the 12-month history of a checking account balan

Figure 4.27 This sparkline displays 30 days of manufacturing defect history compared to the acceptable range

4.11.7 Box Plots

The box plot is a graphical representation introduced in the 1970s by John Wilder Tukey, a mathematician specializing in data visualization. This type of graph effectively displays the distribution of values across an entire range, providing key insights into data dispersion.

Summarizing a dataset using a single measure, such as the sum or average, may not always be sufficient. Understanding how values are distributed is crucial, such as in the case of employee salaries within different salary grades. A single median value, as illustrated in Figure 4.28, does not provide enough information about the overall distribution. The median represents the middle value but does not indicate whether values are concentrated at the lower, middle, or upper end of the salary range.

Figure 4.29 illustrates six different salary distributions within the same salary range, all sharing the same median salary. However, each distribution pattern varies significantly, highlighting the limitations of using the median alone.

A basic representation of salary distributions using range bars also lacks insight into how values are distributed within the range.

A more informative method, illustrated in Figure 4.30, combines range bars with median markers. When the median is closer to the lower end, it indicates that most salaries fall into the lower half of the range, and vice versa. This technique provides more information about data distribution but still has limitations.

Tukey's box plot builds upon this concept by adding more statistical information. In a typical box plot, represented in Figure 4.31, a rectangular box represents a range of values. The top and bottom of the box indicate the first quartile (Q1) and third quartile (Q3), respectively. The median is marked inside the box, while "whiskers" extend to the smallest and largest values within a set limit. Outliers are represented as individual points beyond the whiskers.

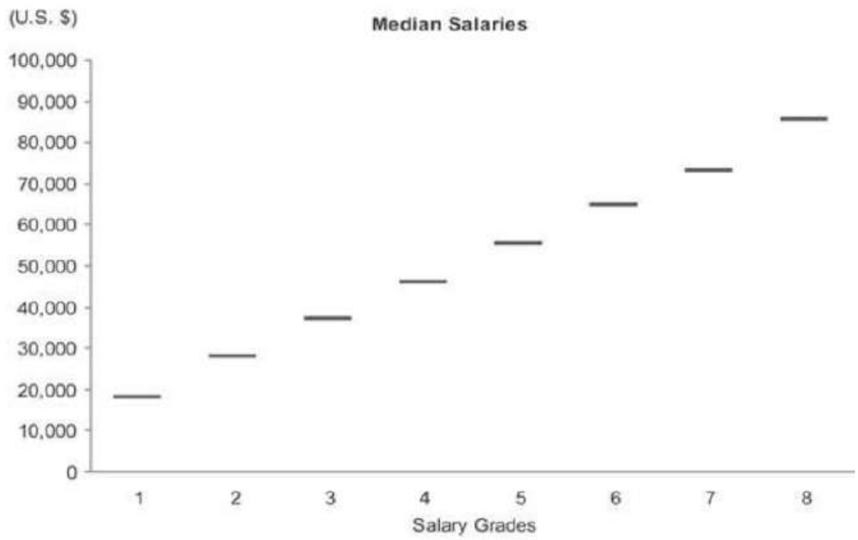


Figure 4.28 This graph displays employee salaries per salary grade as a single median value for each grade.

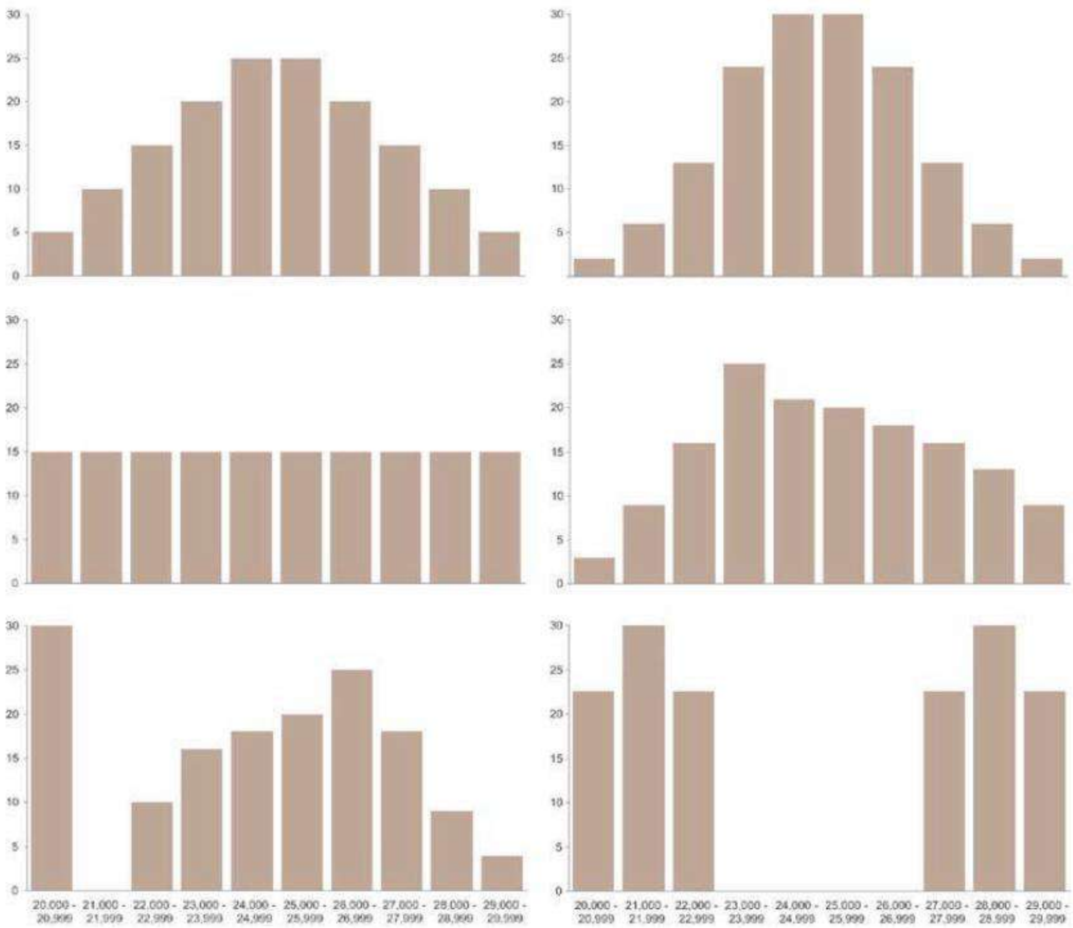


Figure 4.29 Six examples illustrating different distributions of salaries with the same median value. The vertical axis represents the number of employees, while the horizontal axis shows salary ranges.

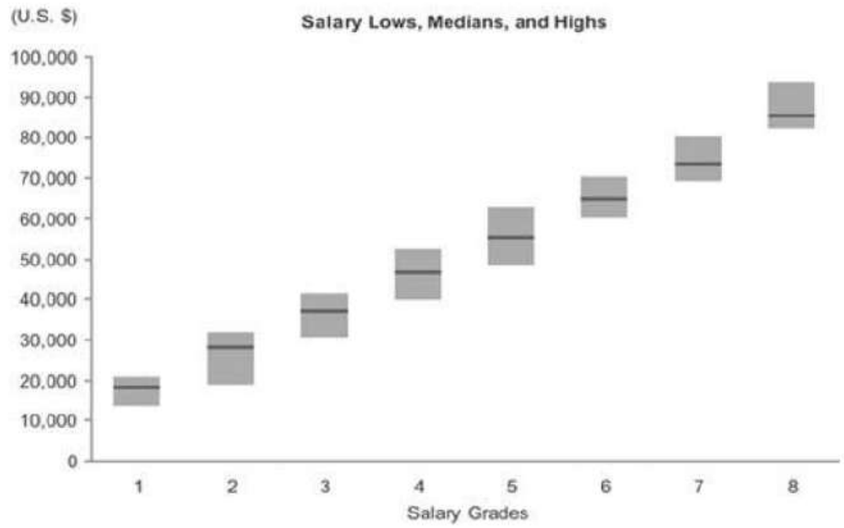


Figure 4.30 This graph combines range bars with data points to mark the medians as well as the high and low salaries in each

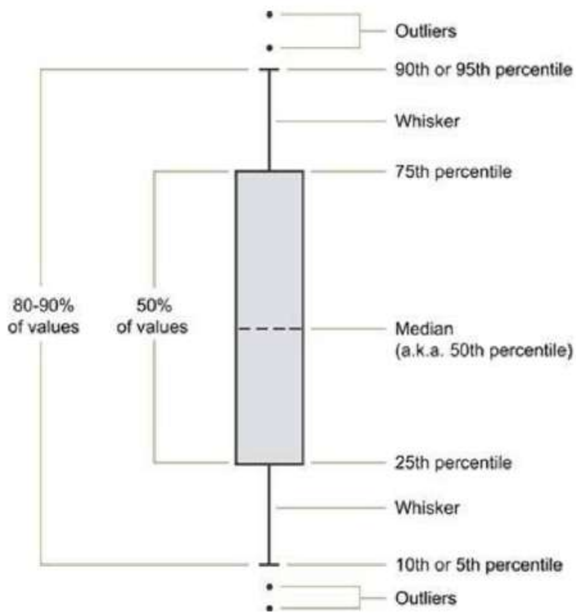


Figure 4.31 An individual box plot with whiskers.

4.11.8 Scatter Plots

A scatter plot effectively displays the correlation between two quantitative variables, showing whether and how strongly they are related. For example, Figure 4.32 illustrates the correlation between the number of broadcast ads and sales revenue over 24 months.

Key insights from this scatter plot include:

- A correlation exists, as changes in ad frequency generally correspond to changes in sales revenue.
- The correlation is positive, meaning that increased ads are associated with higher sales.
- The correlation is strong, indicated by the close clustering of points around the trend line.

A scatter plot can also separate data into different categories. In Figure 4.33, the correlation between sales revenue and two ad types—radio and television—is displayed separately. The television ads show a stronger correlation with sales revenue than radio ads.

While three-dimensional scatter plots exist to represent three variables, they are generally not recommended for dashboards due to their complexity.

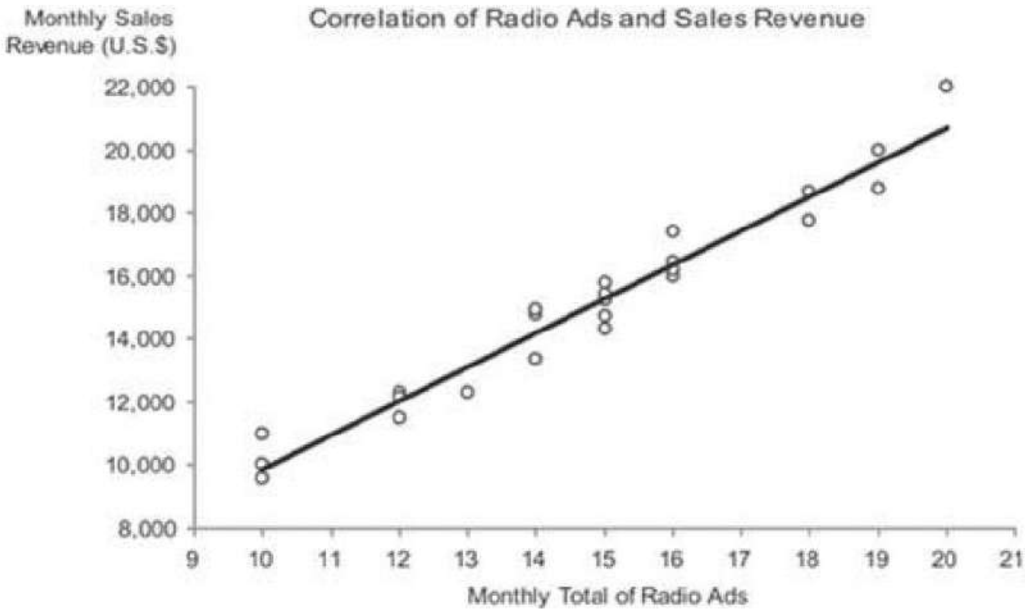


Figure 4.32 This scatter plot shows the correlation between the number of broadcast ads and sales revenue over a 24-month period.

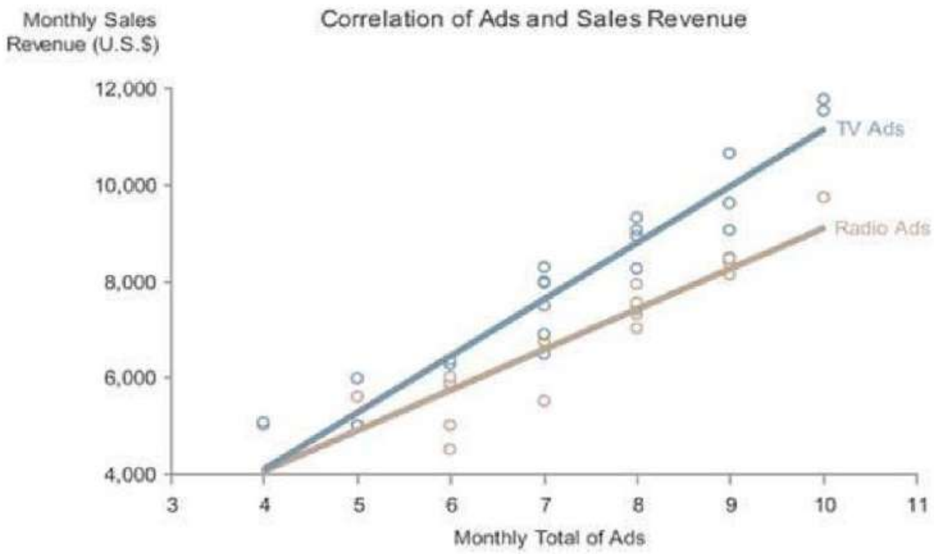


Figure 4.33 This scatter plot illustrates the correlation between the number of radio and television ads and their corresponding sales revenue over a 24-month period.

4.11.9 Treemaps

Treemaps were developed in the 1990s by Ben Shneiderman from the University of Maryland to efficiently display large hierarchical or categorical datasets. They maximize available space by filling it with contiguous rectangles, where the size of each rectangle represents a quantitative variable. An additional variable can be encoded using color.

Treemaps are not meant for fine quantitative comparisons but are useful for identifying specific trends. Figure 4.34 demonstrates a treemap displaying sales revenue by region, with rectangle size representing revenue and color indicating percentage of quota achieved. This format quickly highlights underperforming or high-performing regions.

A common color scheme for treemaps involves varying hues, such as using red for values below quota and green for values above. However, using a single hue with varying intensity can sometimes be more effective for nuanced interpretation.

Treemaps often include interactive features, allowing users to drill down into specific categories for further insights. This makes them valuable tools for dashboards, provided they are designed appropriately for their intended purpose.

Figure 4.34 This treemap, generated using Treemap 4.3 software from the University of Maryland's Human-Computer Interaction Lab (HCIL), visualizes sales data, including revenue and percentage of quota, by region.

4.11.10 Final Thoughts on Graphs

Certain commonly used graphs are excluded from this collection due to their inefficiency or complexity. The pie chart, for example, is often used to show part-to-whole relationships but is less effective than bar graphs. Figure 4.35 compares a pie chart and a bar graph displaying the same data, demonstrating that bar graphs allow for easier interpretation and comparison.

Another type of graph that can be problematic is the area graph, which suffers from occlusion, as illustrated in Figure 4.36. In this example, overlapping data hides important information.

Radar graphs, which plot quantitative values along radial axes, are another example of a less effective visualization. As shown in Figure 4.37, a radar graph can be harder to interpret compared to a bar graph. They are only useful when the categorical scale naturally fits a circular arrangement, such as hours of the day.

While many graph types exist, dashboards should prioritize clarity, efficiency, and ease of interpretation. Selecting the most effective visualization techniques ensures that the displayed data conveys meaningful insights without unnecessary complexity.

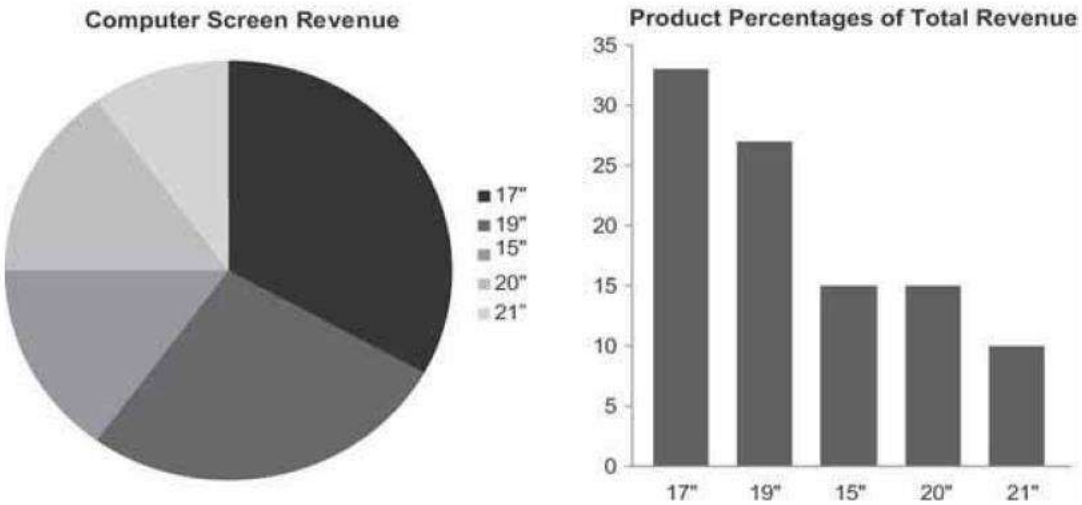


Figure 4.35 This pie chart and bar graph represent the same part-to-whole data, but the bar graph makes it easier to interpret and compare values.

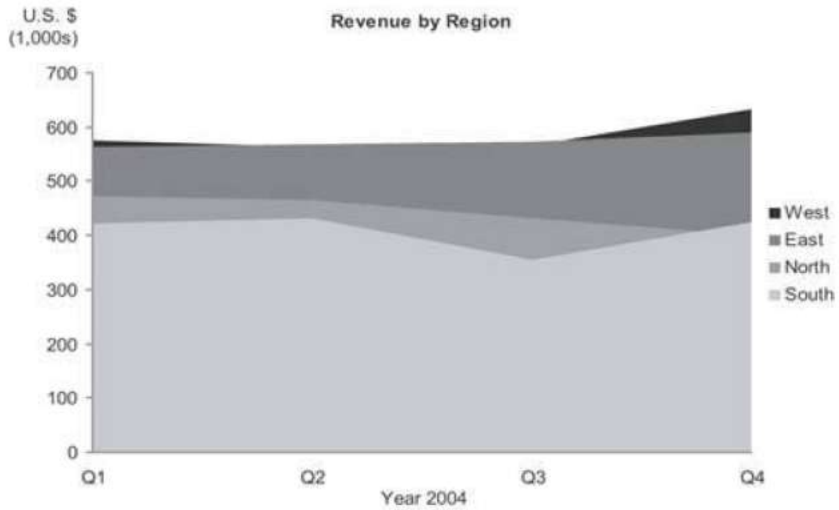


Figure 4.36 Area graphs can suffer from the problem of occlusion.

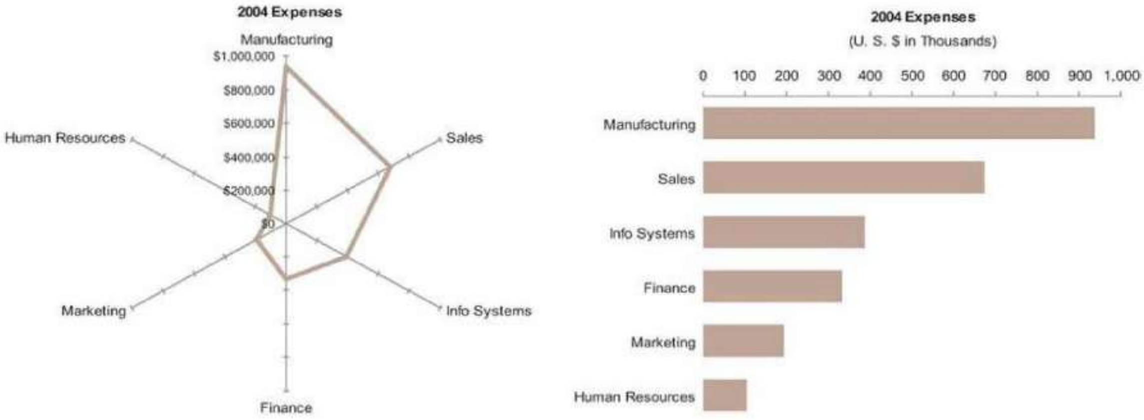


Figure 4.37 This radar graph (left) and bar graph (right) represent the same expense data. In the radar graph, departments are positioned along the circumference, with expenses measured along radial axes extending from the center. The bar graph provides a clearer and faster way to interpret the data.

4.12 Icons

Icons are simple images that convey a clear and straightforward meaning. Only a few icons are necessary on a dashboard, and the most useful ones typically indicate the following:

1. Alert
2. Up/Down
3. On/Off

4.12.1 Alert Icons

Alert icons are used to draw attention to specific information, especially when something requires immediate attention. An effective alert icon should be simple and highly noticeable. It is best to limit the number of alert levels to one or two, as having multiple variations can be overwhelming and reduce effectiveness.

A common approach to alerts is the traffic light metaphor, using green, yellow, and red to indicate different statuses. However, this method may not be effective since a green indicator serves little purpose when everything is functioning correctly. Instead, an alert icon should appear only when attention is needed, making it more noticeable.

A simple shape such as a circle or square works best for alert icons. If multiple alert levels must be represented, it is better to use a single shape and vary its color intensity rather than using distinct icons. Using traffic light colors can be problematic for individuals with color blindness, as shown in Figure 4.38, which illustrates how these colors appear to people with color vision deficiencies. A better approach is to use different intensities of a single hue, such as light red and dark red.

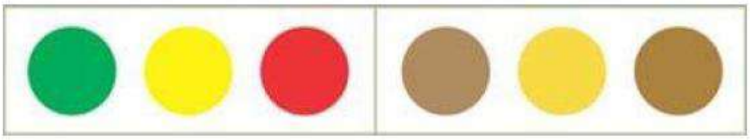


Figure 4.38 The icons on the right simulate what someone who is color-blind would see when looking at those on the left.

4.12.2 Up/Down Icons

Up/down icons indicate whether a measure has increased or decreased compared to a previous value or a target. These icons are commonly used for financial data to represent changes in stock prices, profits, and other key metrics.

A conventional way to display these changes is through an arrow or triangle pointing up or down. Color may also be used to enhance visibility, with green indicating positive change and red indicating negative change. However, this approach may not be effective for color-blind individuals. Instead, using colors with significant intensity differences, such as saturated red for a negative trend and pale green for a positive trend, can ensure better accessibility. Figure 4.39 presents examples of simple up/down icons.

Figure 4.39 Simple Up and Down icons

4.12.3 On/Off Icons

On/off icons serve as indicators to differentiate certain items from others. They are useful for marking significant items in a list, identifying priority tasks, or indicating current status in a schedule.

Common icons for on/off indicators include checkmarks, asterisks, and Xs, as shown in Figure 4.40 To maintain consistency and clarity, it is recommended to select a single icon style and use it consistently across the dashboard.

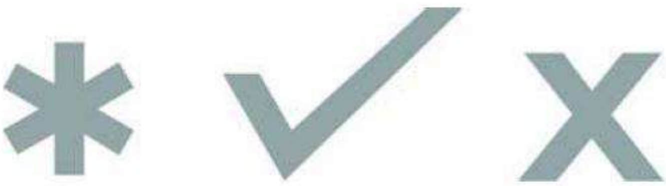


Figure 4.40 Sample on/off icons.

4.13 Text

Even highly graphical dashboards require some textual information. Text is essential for categorical labels on graphs and for displaying single measures clearly. When presenting an isolated value without comparison, text is often more effective than a graph. Dashboards should include simple text boxes to accommodate these values where necessary.

4.14 Images

Images such as photographs, illustrations, and diagrams are rarely necessary on a dashboard. However, they may be useful in specific contexts. For example, a trainer's dashboard could include photos of attendees, a maintenance worker's dashboard might highlight areas needing repair, and a police dashboard could display a map of recent incidents.

4.15 Drawing Objects

Drawing objects help visualize relationships between pieces of information. For example, they can illustrate sequential steps in a process or hierarchical structures such as an organization chart.

Simple shapes like rectangles and circles can represent different entities, while lines and arrows can indicate relationships and direction. **Figure 4.41** provides example of how drawing objects can clarify relationships within data.

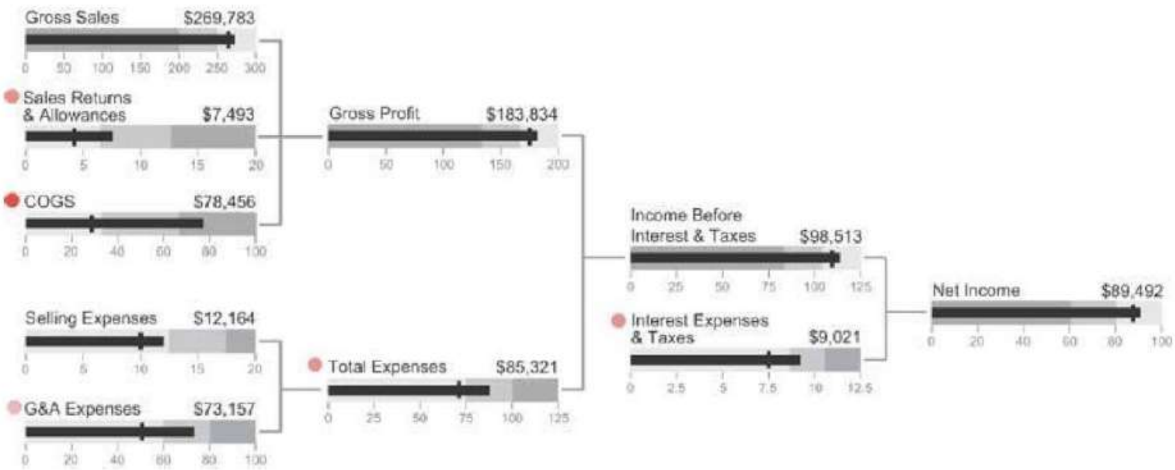


Figure 4.41 Basic drawing elements can help illustrate the relationships between different components of net revenue.

4.16 Organizers

Data often needs to be arranged systematically to enhance clarity. Three primary methods for organizing dashboard information include:

1. Tables
2. Spatial Maps
3. Small Multiples

4.16.1 Tables

Tables arrange data into rows and columns, making it easy to read and compare information. They can also structure graphs, icons, and images for better organization.

4.16.2 Spatial Maps

Spatial maps associate data with physical locations and are useful when the meaning of the data is tied to geography. For instance, maps can display employee absenteeism across store locations, revealing patterns that might not be obvious otherwise.

Another common spatial map is a building floor plan, which can help visualize temperature variations across different sections of a facility. Figure 4.42 illustrates how spatial maps should only be used when necessary to avoid unnecessary complexity.

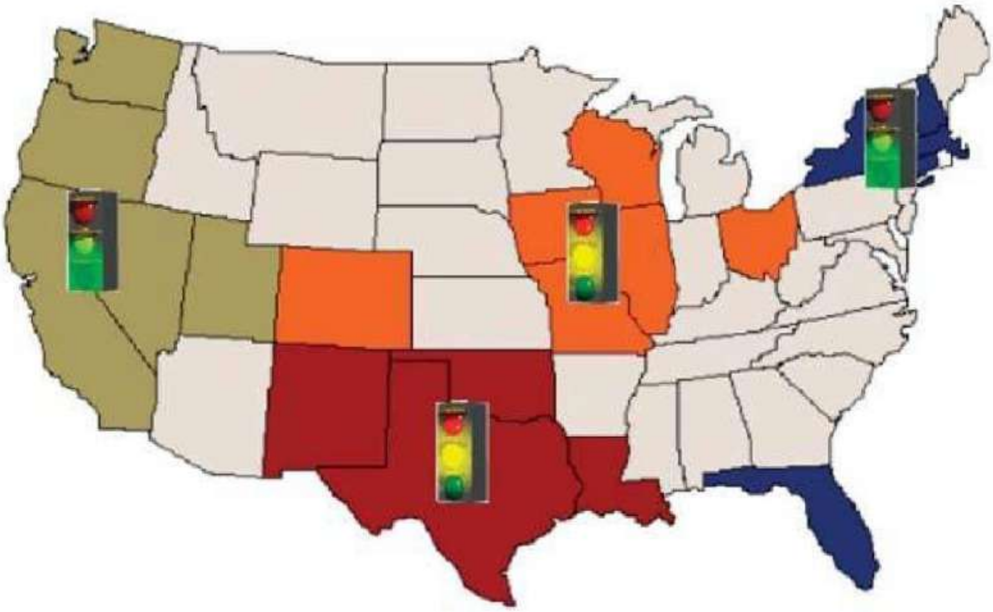


Figure 4.42 Spatial map example.

4.16.3 Small Multiples

Edward Tufte's concept of "small multiples" involves displaying multiple versions of the same graph with variations along a single dimension. This approach is useful when presenting multiple related datasets in a compact space.

For example, a revenue chart showing bookings and billings across four regions can be effectively represented as a small multiple arrangement (Figure 4.43).

An intelligent dashboard tool should allow users to easily configure small multiples by selecting data dimensions, graph axes, and layout preferences. However, as of now, few dashboard applications offer an automated way to generate small multiples efficiently.

Figure 4.43 This series of horizontally aligned small multiples displays revenue split between three sales channels.

4.17 Designing Effective Dashboards

Beyond selecting appropriate display media and minimizing non-data pixels, several other design aspects must be considered to ensure dashboards are user-friendly and fully support the viewer's need to interpret information effectively. Implementing key design strategies helps blend visual elements into a functional and aesthetically pleasing display.

4.18 Organizing Information for Meaningful Use

Information should not be placed randomly on a dashboard. The arrangement of data plays a crucial role in its usability. Consider the following principles:

- Organize groups based on business functions, entities, and use.
- Position related items close to each other.
- Use minimal visual elements to delineate groups.
- Facilitate meaningful comparisons.
- Avoid unintentional, misleading comparisons.

4.18.1 Grouping by Business Functions, Entities, and Use

A logical way to structure data is by grouping it according to business functions (e.g., sales, budgeting, logistics), entities (e.g., departments, projects), or data usage (e.g., revenue-expenditure comparisons). These groupings should reflect how the information is typically used within the organization.

Some users, such as CEOs, require a more holistic view that integrates different data points based on overall business impact rather than strict departmental divisions. Arranging data in a structured sequence can enhance readability and efficiency in information retrieval.

4.18.2 Positioning Related Items Together

Items that belong to the same category should be placed close to one another. This proximity makes relationships immediately apparent and allows for quick interpretation of related data.

4.18.3 Using Minimal Means to Delineate Groups

Visual elements like grid lines, borders, and background colors add non-data pixels that can be distracting. The best way to separate groups is through white space, which allows for clear segmentation without adding unnecessary visual clutter.

However, in high-density dashboards where space is limited, subtle borders can be used for separation. Figure 4.44 illustrates how white space or light borders can effectively delineate groups.

Product	Units Sold	Actual Revenue
Shirts	938	187,600
Blouses	1,093	114,765
Pants	3,882	62,112
Skirts	873	36,666
Dresses	72	2,088
Total	6,858	$403,231

Region	Units Sold	Actual Revenue
North	2,263	133,066
South	1,920	112,905
East	1,303	76,614
West	754	44,355
Canada	618	36,291
Total	6,858	$403,231

Channel	Units Sold	Actual Revenue
Direct	2,057	120,969
Distributor	1,921	119,903
Reseller	1,783	104,840
OEM	1,097	64,519
Total	6,858	$403,231

Warehouse	Units Sold	Actual Revenue
Virginia	2,537	149,195
California	1,920	112,905
Texas	1,372	80,646
Calgary	1,029	60.485
Total	6,858	$403,231

Product	Units Sold	Actual Revenue	Region	Units Sold	Actual Revenue
Shirts	938	187,600	North	2,263	133,066
Blouses	1,093	114,765	South	1,920	112,905
Pants	3,882	62,112	East	1,303	76,614
Skirts	873	36,666	West	754	44,355
Dresses	72	2,088	Canada	618	36,291
Total	6,858	$403,231	Total	6,858	$403,231
Channel	Units Sold	Actual Revenue	Warehouse	Units Sold	Actual Revenue
Direct	2,057	120,969	Virginia	2,537	149,195
Distributor	1,921	119,903	California	1,920	112,905
Reseller	1,783	104,840	Texas	1,372	80,646
OEM	1,097	64,519	Calgary	1,029	60.485
Total	6,858	$403,231	Total	6,858	$403,231

Figure 4.44 The four tables at the top are distinctly separated using white space, while the ones at the bottom, placed closer together, are divided with light borders.

4.18.4 Encouraging Meaningful Comparisons

Data gains significance when compared with relevant benchmarks. To promote useful comparisons, consider the following techniques:

- Display multiple related measures in a single table or graph.
- Position comparable items near each other.
- Use consistent colors to indicate related data.
- Include comparative values like percentages, ratios, or variances.

Figure 4.45 shows examples where multiple measures are combined in a single graph to facilitate comparisons, while Figure 4.46 demonstrates how comparative values can directly enhance understanding.

Figure 4.45 Combining multiple measures in a single graph to encourage comparisons.

Product	Units Sold	Actual Revenue	% of Total	Forecast Revenue	% of Fcst
Product A	938	187,600	47%	175,000	107%
Product B	1,093	114,765	28%	130,000	88%
Product C	3,882	62,112	15%	50,000	124%
Product D	873	36,666	9%	40,000	92%
Product E	72	2,088	1%	50,000	4%
Total	6,858	$403,231	100%	$445,000	91%

Figure 4.46 Using comparative values to directly support comparisons

7.1.5 Avoiding Misleading Comparisons

Not all data displayed on a dashboard is meant to be compared. Without careful design, viewers might draw unintended connections.

For example, Figure 4.47 shows a dashboard where inconsistent color usage leads to confusion. To prevent this, avoid using the same colors for unrelated metrics and ensure that items meant to be interpreted separately are spaced apart.

Figure 4.47 This dashboard inadvertently encourages meaningless comparisons.

4.19 Maintaining Consistency for Quick Interpretation

Inconsistencies in visual presentation can cause confusion and slow down comprehension. To avoid this:

- Ensure that similar data is always displayed in the same format.
- Use a uniform style for visual elements, such as axes and labels.
- Select the most effective data display format and apply it consistently.

Avoid varying visual elements purely for variety's sake. If different sections of a dashboard present similar types of relationships (e.g., time-series data comparisons), they should use the same type of graph.

4.20 Enhancing Aesthetic Appeal for Better Usability

Aesthetic appeal plays a role in usability by making dashboards more engaging and less visually fatiguing. Donald Norman's book *Emotional Design* highlights that well-designed, visually pleasing interfaces enhance comprehension and engagement.

4.20.1 Choosing Colors Wisely

Poor color selection is a common issue in dashboard design. To improve readability:

- Use bright colors sparingly to highlight key information.
- Prefer subdued colors, like natural earth tones, for most elements.
- Employ slightly off-white backgrounds to reduce harsh contrasts.

Figure 4.48 illustrates effective use of color to enhance dashboard readability.

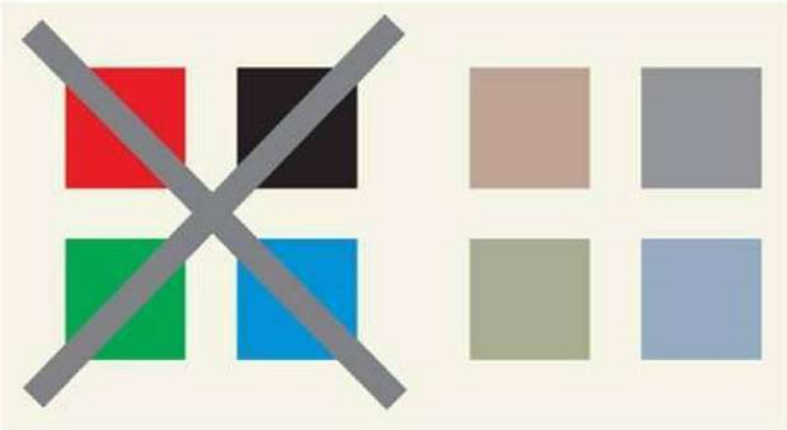


Figure 4.48 Use bright colors only for highlighting specific data, while keeping most elements in subdued tones. A slightly off-white background helps reduce harsh contrast with foreground colors.

4.20.2 Using High Resolution for Clarity

High-density dashboards require sharp, clear visuals to prevent eye strain and make scanning efficient. Blurry or low-resolution graphics slow down interpretation and should be avoided.

4.20.3 Selecting Readable Fonts

Dashboard text should be legible and free from unnecessary embellishments. Use:

- Clear, simple fonts that are easy to read quickly.
- A consistent font throughout the dashboard.
- Different font sizes or styles only where necessary, such as headings.

Figure 4.49 demonstrates fonts that are effective for dashboard readability.

Fine Legibility		Poor Legibility	
Serif	Sans-Serif	Serif	Sans-Serif
Times New Roman	Arial	Script	Gill Sans Ultra
Palatino	Verdana	Broadway	Papyrus
Courier	Tahoma	Old English	Tempus Sans ITC

Figure 4.49 Examples of some fonts that are easy to read and some that are not.

4.21 Designing Dashboards as Interactive Launch Pads

Dashboards often serve as an overview rather than a complete data source. They should be designed for interaction, allowing users to access more detailed information when needed. Key interactive elements include:

- **Drilling down** into detailed data.
- **Filtering and slicing** to focus on specific aspects of the data.

To ensure smooth interaction:

- Enable users to click directly on data points for further details.
- Maintain consistent interaction methods throughout the dashboard.

For example, clicking a bar in a revenue chart could open a breakdown by region, or hovering over a point in a line graph could display the exact value. Consistency in interaction ensures intuitive navigation and minimizes confusion.

4.22 Testing Your Design for Usability

Even the most well-designed dashboards can be met with resistance due to users' preconceived notions of how they should look. It is important to guide users in forming expectations based on expert input rather than independent assumptions. To achieve this, present them with a single, well-thought-out prototype that represents the most effective design. This prototype should serve as the foundation for discussions on potential refinements. Avoid offering multiple alternative designs, as users may understand their functional needs but lack expertise in visual design. As the designer, it is your responsibility to lead this process.

Perfection is unlikely on the first attempt, regardless of experience. Therefore, usability testing is essential. The best way to evaluate the design's effectiveness is to test it with actual users who will interact with the dashboard in real-world scenarios. Populate the dashboard with real data and observe how users interpret and engage with it. If the design incorporates new display methods unfamiliar to users, provide a brief explanation of how they function and the rationale behind their selection.

If the design process has been thorough and users are focused on improving their workflow rather than clinging to habitual methods, usability testing typically results in minor adjustments rather than significant redesigns. While exceptions exist due to individual preferences, a well-designed dashboard generally receives a positive reception.

4.23 Case Studies

This section explores dashboards that demonstrate clear and efficient communication through well-informed design. The sample dashboards cover two distinct business scenarios

1. **Sales Dashboard** – Designed for a sales manager to track sales performance and opportunities (**strategic**).
2. **Marketing Analysis Dashboard** – Helps a marketing analyst evaluate the website's marketing performance (**analytical**).

4.23.1 Sample Sales Dashboard

Sales dashboards are among the most frequently implemented dashboards, second only to executive dashboards. Since sales activities are crucial for business success, sales managers must continuously monitor performance. Rapid changes in market conditions, competitive pressures, and emerging opportunities require a flexible and responsive sales strategy. A well-designed sales dashboard serves as an essential tool to support these needs.

Key Metrics in a Sales Dashboard

When designing a sales dashboard, the first step is identifying the most relevant metrics for sales managers. The selected measures should provide real-time insights into sales performance. Some commonly included metrics are:

- **Sales revenue**
- **Sales revenue in the pipeline** (expected revenue categorized by probability)
- **Profit**
- **Customer satisfaction rating**
- **Top 10 customers**
- **Market share**

It is important to note that this list is not definitive. The data included in a sales dashboard should be customized based on the specific needs of the business and its users. The primary goal of this example is to demonstrate how visual design principles can be applied effectively.

Key Considerations in Dashboard Design

For each metric, several decisions need to be made to ensure clarity and usability:

- **Level of Summarization:** Should the data be presented at an aggregated or detailed level?
- **Unit of Measurement:** What unit best represents the data (e.g., currency, percentage, count)?
- **Contextual Information:** What additional data should be included to enhance interpretation?
- **Display Method:** Which visualization format best conveys the information?
- **Importance Ranking:** How critical is this metric compared to others?
- **Viewing Sequence:** In what order should the data be presented to facilitate logical analysis?
- **Comparison Needs:** Which other metrics should be available for comparison?

If designing a sales dashboard for a specific team or individual, user feedback should guide these choices. However, in general examples, assumptions based on industry knowledge can be made to create an effective prototype.

Features of an Effective Sales Dashboard

A well-structured sales dashboard follows certain design principles to enhance readability and usability. The example dashboard in Figure 4.50 demonstrates these principles:

- **Minimal Use of Color:** The dashboard primarily uses grayscale tones, with selective colors like red for alerts. This ensures that attention is drawn only to critical data points.
- **Prime Screen Positioning:** The most important data, such as key metrics, is placed in the upper-left section of the dashboard, where users naturally focus first.

- **Compact and Efficient Visuals:** To accommodate a large amount of data without clutter, space-efficient visualizations such as sparklines and bullet graphs are used.
- **Dual Representation of Key Metrics:** Some figures are displayed both graphically and in text format to support quick insights and detailed analysis.
- **Stacked Bar Charts for Pipeline Revenue:** Combining actual and anticipated revenue in a stacked format provides an intuitive view of projected earnings relative to targets.
- **Use of White Space for Organization:** Instead of unnecessary borders, grid lines, or background fills, white space is used to separate sections clearly.
- **Minimal On-Screen Instructions:** Rather than cluttering the display with excessive instructions, a single help button provides guidance when needed.

While this dashboard demonstrates strong design principles, alternative designs can offer further improvements. Comparing different sales dashboards can provide valuable insights into effective visualization techniques.

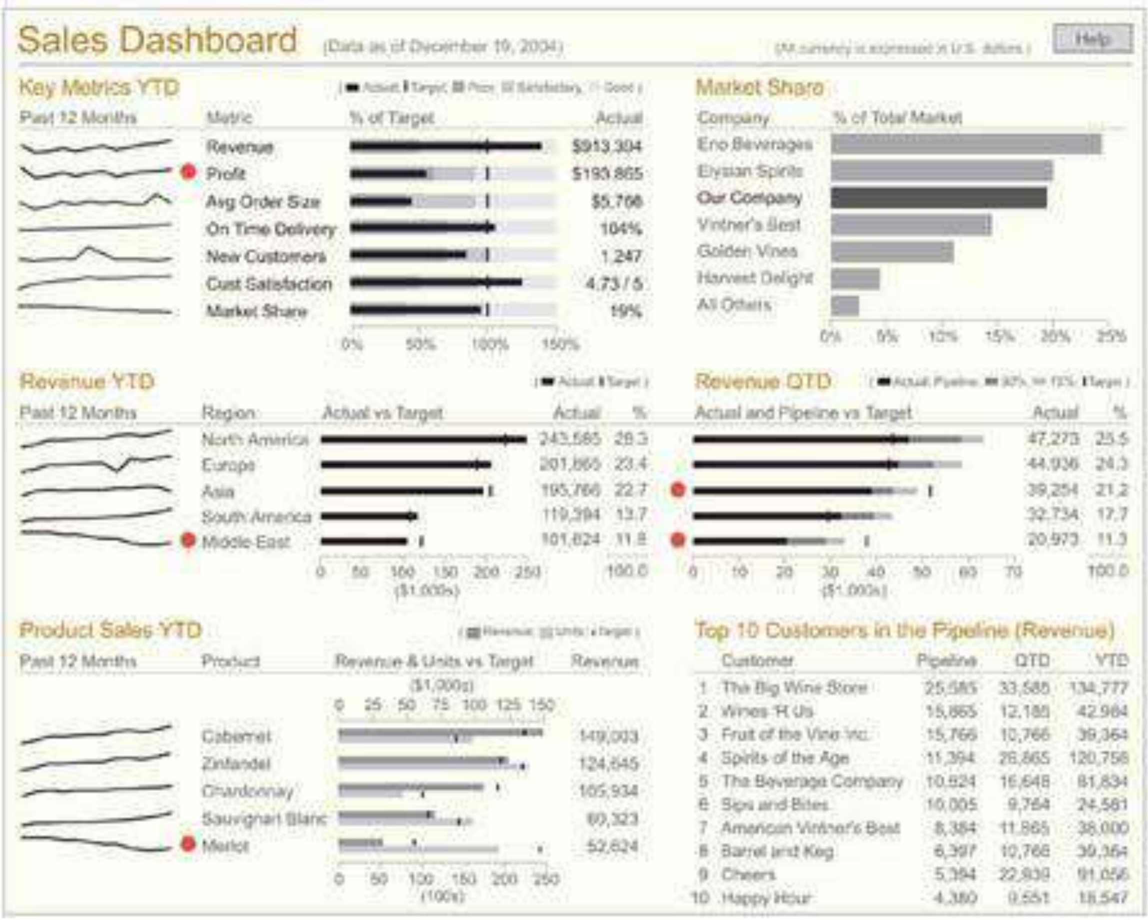


Figure 4.50 A sample sales dashboard with good design principles

Several alternative sales dashboards were designed to meet the same requirements. Examining them reveals both strengths and weaknesses, reinforcing the importance of well-applied design principles.

Example 1: Text-Based Dashboard (Figure 4.51)

This version relies primarily on text, with color coding (green, light red, and vibrant red) to indicate performance levels. However, several issues reduce its effectiveness:

- **Heavy Reliance on Text:** Text-based dashboards require users to read through data sequentially, slowing down information processing. Visual dashboards enable parallel processing, which is faster and more efficient.
- **Lack of Graphical Comparisons:** Comparing actual performance to targets requires mental calculations. Using visual indicators such as bar charts or bullet graphs would make these comparisons more intuitive.
- **Misaligned Numerical Data:** Center-justified numbers make vertical comparisons difficult. Right-justified alignment improves readability.
- **Missing Key Measures:** Important metrics like pipeline revenue and top 10 customers are absent, limiting the dashboard's usefulness.
- **Equal Emphasis on All Quarters:** The dashboard gives equal focus to all quarters, while sales managers are likely more interested in the current quarter's performance.
- **Ineffective Use of Color for Distinctions:** While the bright red highlights for poor performance stand out, the more subdued red and green shades may be difficult to distinguish, especially for color-blind users. Additionally, poor-performing measures are displayed against a dark red background, making them harder to read.

SALES DASHBOARD - 19 December 2004

Performance: Good / Satisfactory / Poor

		Actual Q1	Actual Q2	Actual Q3	To Date Q4	Forecast Q4	Target Q4
Revenue Total		$ 154,057	$ 165,158	$ 199,738	$ 206,264	$ 225,205	$ 215,000
by Region	*North America*	$ 78,963	$ 78,138	$ 91,176	$ 91,441	$ 100,197	$ 107,500
	Europe	$ 30,811	$ 33,032	$ 39,948	$ 41,253	$ 45,374	$ 43,000
	Asia	$ 28,877	$ 37,472	$ 48,641	$ 52,944	$ 57,380	$ 39,775
	South America	$ 3,041	$ 3,435	$ 4,206	$ 5,035	$ 5,738	$ 5,375
	Middle East	$ 12,365	$ 13,081	$ 15,767	$ 15,592	$ 16,515	$ 19,350
by Product	*Cabernet*	$ 28,430	$ 30,228	$ 35,053	$ 38,728		$ 38,700
	Zinfandel	$ 13,876	$ 10,164	$ 17,876	$ 18,664		$ 19,350
	Merlot	$ 25,440	$ 24,977	$ 28,955	$ 28,865		$ 36,550
	Chardonnay	$ 68,634	$ 64,025	$ 104,063	$ 107,610		$ 98,900
	Sauvignan Blanc	$ 17,677	$ 35,763	$ 13,790	$ 12,399		$ 21,500
Profit		$ 31,999	$ 36,749	$ 42,431	$ 46,685		$ 53,750
Avg Order Size		405	421	435	449		430
Market Share		23%	20%	19%	17%		27%
Customer Satisfaction		3.18	2.95	2.82	2.67		3.29
On-Time Delivery		83%	73%	65%	68%		95%
New Customers		346	430	447	468		450

Figure 4.51 This text-based sample sales dashboard could be improved.

Example 2: Graph-Heavy Dashboard (Figure 4.52)

This version incorporates more visual elements but still exhibits several design flaws:

- **Excessive Grid Lines:** Both tables and graphs include heavy grid lines that add unnecessary visual clutter. If needed, these lines should be muted to avoid overpowering the data.
- **Unnecessary Visual Effects:** Drop shadows on bars, lines, and pie charts add decorative elements without enhancing readability.

- **Overuse of Color:** Bright colors throughout the dashboard make it visually overwhelming and detract from the most important data.
- **Lack of Numerical Context:** All figures are expressed as percentages. While this might be useful for performance tracking, actual sales figures would provide additional valuable insights.
- **Inefficient Use of Pie Charts:** A bar chart with ranked values would be a more effective way to display regional revenue distribution than a pie chart.
- **Limited Trend Analysis:** While a 12-month revenue history is shown, there are no regional or product-based trend comparisons, which could provide deeper insights.

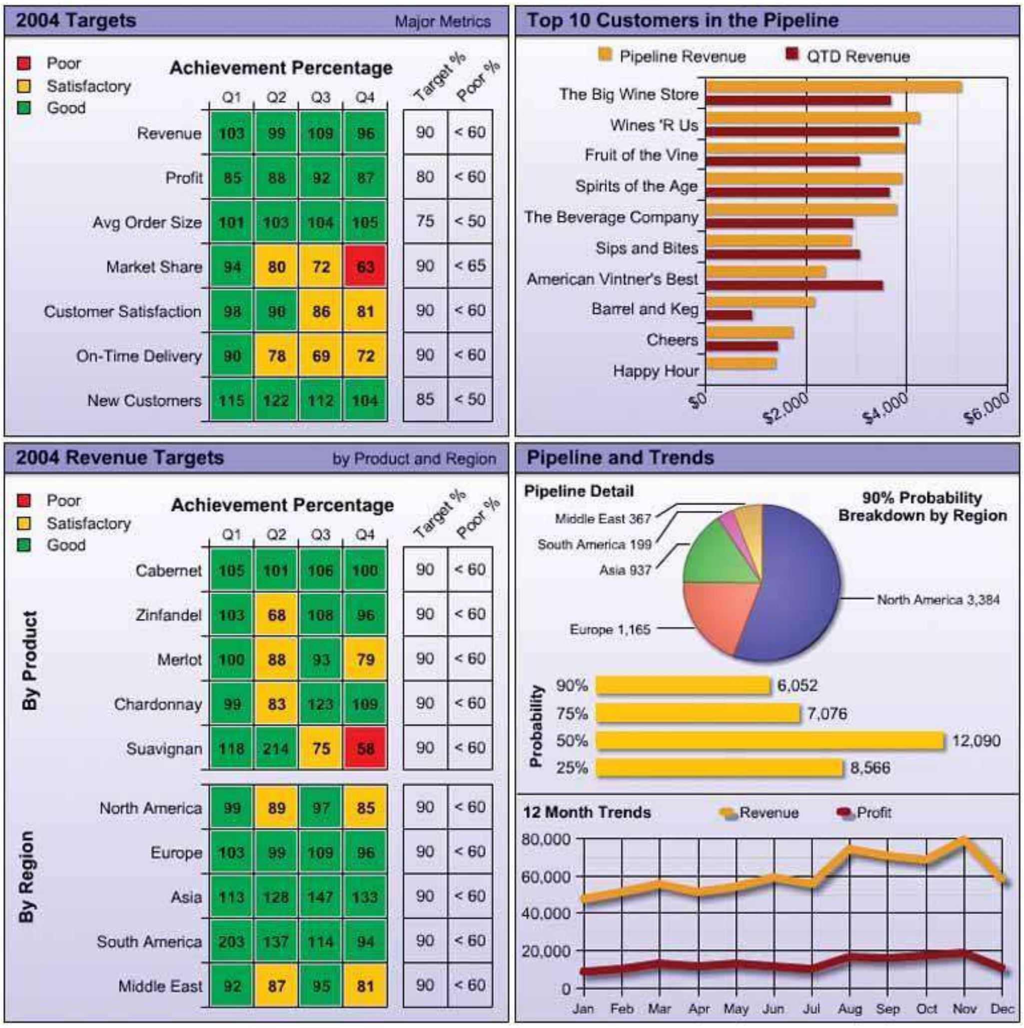


Figure 4.52 An example of a bad design

By analyzing these examples, it becomes clear that effective dashboard design prioritizes simplicity, readability, and meaningful data visualization. Applying these principles ensures that a sales dashboard delivers immediate insights while avoiding unnecessary distractions.

4.23.2 Sample Marketing Analysis Dashboard

A marketing analysis dashboard serves as a tool for monitoring key performance metrics related to a company's marketing efforts. Unlike other dashboards, this type of dashboard is specifically designed to support analytical tasks. It helps analysts identify patterns, problems, and opportunities that can improve marketing strategies. Ideally, it should not only present data but also serve as a starting point for deeper analysis by providing access to additional data and tools.

This example focuses on an analyst responsible for monitoring customer behavior on a company's website. Her primary goal is to identify obstacles preventing customers from completing purchases and to discover opportunities for increasing sales through better product placement and marketing strategies. The dashboard includes the following key data points:

- **Number of visitors** (daily, monthly, and yearly)
- **Number of orders placed**
- **Number of registered visitors**
- **Number of times individual products were viewed**
- **Instances where products displayed together were rarely purchased together**
- **Instances where products not displayed together were frequently purchased together**
- **Referral sources that led to the highest number of site visits**

The top section of the dashboard provides an overview of website performance over time, highlighting missed opportunities and ineffective marketing efforts. The visitor data is segmented into three time intervals to offer varying levels of detail. Recent data is displayed with greater granularity, while older data is summarized to focus on long-term trends.

Ranking and Text-Based Insights

Many elements of this dashboard are organized using ranking-based structures, which is common for dashboards that highlight critical conditions—both positive and negative. The majority of the ranked information is displayed in text format rather than as graphical content. Since the primary goal is to inform the analyst about areas that require attention, text descriptions provide the necessary context effectively.

Unlike performance dashboards, where graphical displays allow for rapid scanning, this type of dashboard requires the analyst to read and interpret the data. The inclusion of an item in a ranked list already signifies its importance, making additional graphical alerts unnecessary.

Figure 4.53 illustrates an example of a web marketing analysis dashboard, demonstrating these principles in action.

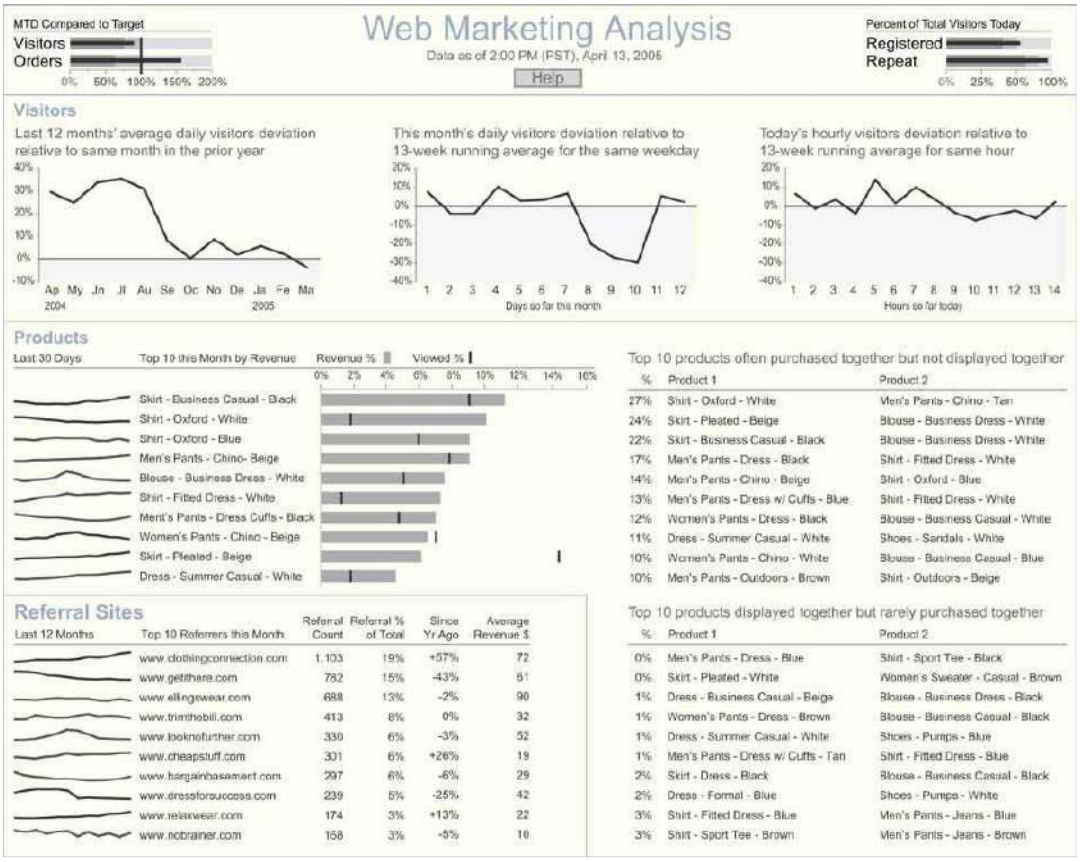


Figure 4.53 A sample web marketing analysis dashboard.

Summary

This unit covered the principles of information dashboard design, emphasizing clarity, usability, and visual appeal. It explored key design strategies, effective use of graphs, icons, and text, and the importance of structured organization for meaningful insights. The role of interactive dashboards and usability testing was also highlighted, with case studies providing real-world applications.

Keywords

Information Dashboard, Data Visualization, Dashboard Design, User Experience, Graphical Representation, Interactivity

Learning Outcomes

Upon successful completion of this unit, learners will be able to:

- Define and explain the concept of data dashboards and their applications.
- Describe the key characteristics and strategies for designing an effective dashboard.
- Utilize appropriate graphical elements to enhance data visualization.
- Differentiate between various dashboard display media and their optimal use cases.
- Implement best practices in organizing dashboard information for better readability and usability.

Exercises

Objective Questions

1. What is the primary purpose of a data dashboard?
   a) To replace traditional reports
   b) To provide decision-makers with a quick overview of key metrics
   c) To store raw data for future analysis
   d) To eliminate the need for data analysts
2. Which of the following is NOT a key feature of a data dashboard?
   a) Real-time or periodic updates
   b) Interactive elements like filters and drill-downs
   c) Automatic decision-making based on AI models
   d) Multiple data visualizations, such as charts and tables
3. What is the main goal of reducing non-data pixels in dashboard design?
   a) To make the dashboard visually appealing
   b) To maximize the data-ink ratio for better clarity
   c) To eliminate the need for user interaction
   d) To ensure all information is displayed in one view
4. How can a dashboard highlight important data effectively?
   a) By using excessive colors and animations
   b) By increasing the brightness, size, or contrast of key elements
   c) By displaying all available data at once
   d) By using complex 3D graphs for every metric
5. When should text-based representation be preferred over graphical displays in dashboards?
   a) When comparing data trends over time
   b) When presenting precise numerical values
   c) When analyzing complex relationships between multiple variables
   d) When creating an interactive and engaging dashboard
6. What is the primary advantage of using bullet graphs in dashboards?
   a) They use bright colors to attract attention
   b) They provide a compact alternative to traditional gauges
   c) They are easier to create than other graphs
   d) They eliminate the need for comparative markers
7. When should bar graphs be used instead of line graphs?
   a) When data represents trends over time
   b) When data involves a nominal or ordinal scale
   c) When exact numerical values are not necessary
   d) When displaying part-to-whole relationships over time
8. What is the main limitation of stacked bar graphs?
   a) They cannot display part-to-whole relationships
   b) They make it difficult to discern changes in distribution
   c) They require multiple colors to be effective
   d) They take up more space than other graphs
9. Why are pie charts often considered less effective than bar graphs?
   a) They require additional labeling to be understood
   b) They do not allow for part-to-whole comparisons
   c) They make it harder to compare individual values accurately
   d) They are more visually complex than scatter plots

10. What is a key characteristic of sparklines?
    a) They include a quantitative scale for precise measurements
    b) They are compact visualizations of time-series data
    c) They use multiple colors to differentiate trends
    d) They emphasize individual values rather than trends

Answers:

1. b) To provide decision-makers with a quick overview of key metrics
2. c) Automatic decision-making based on AI models
3. b) To maximize the data-ink ratio for better clarity
4. b) By increasing the brightness, size, or contrast of key elements
5. b) When presenting precise numerical values
6. b) They provide a compact alternative to traditional gauges
7. b) When data involves a nominal or ordinal scale
8. b) They make it difficult to discern changes in distribution
9. c) They make it harder to compare individual values accurately
10. b) They are compact visualizations of time-series data

Short Questions

1. Why should a data dashboard be interactive?
2. What is the impact of excessive non-data pixels on dashboard clarity?
3. How do sparklines differ from standard graphs?
4. What are the key elements of a bullet graph?
5. Why are treemaps useful for dashboards?

Long Questions

1. Explain the importance of dashboard customization and provide examples of how different users may require different dashboard designs.
2. Discuss the role of visual attributes in dashboard design and their impact on data interpretation.
3. Discuss the importance of selecting the appropriate graphical format in dashboards.
4. Explain the advantages and disadvantages of stacked bar graphs and alternative methods for displaying similar data.

Collaborative Learning Tasks

Task 1: Evaluating Dashboard Effectiveness

Instructions:

1. Form small groups and select an existing data dashboard from a website or application.
2. Assess the dashboard based on clarity, accuracy, aesthetics, and interactivity.
3. Discuss within your group and identify areas for improvement.
4. Present your findings along with suggestions for redesign.

Task 2: Comparative Visualization Analysis
Instructions:

1. Select a dataset and visualize it using different types of graphs (e.g., bar chart, line graph, pie chart).
2. Analyze which visualization conveys the data most effectively.
3. Present a summary of your findings and justify your preferred visualization method.

Task 3: Evaluating Visualization Effectiveness
Instructions:

1. Form small groups and choose a type of visualization (scatter plot, bar chart, heatmap, etc.).
2. Use the provided real-world dataset (e.g., climate data, stock market trends) to assess how well the visualization represents the information.
3. Evaluate your assigned visualization based on clarity, accuracy, aesthetics, and potential for misleading the audience.
4. Discuss within your group and prepare a short presentation on your findings. Include suggestions for improvement.

Task 4: Redesigning a Misleading Visualization
Instructions:

1. Search online for an example of a misleading data visualization.
2. Analyze why it is misleading (e.g., improper scaling, omitted context, confusing labels).
3. Redesign the visualization to present the data more accurately.
4. Share your redesigned version and explain the changes made.

Case Studies

Case Study 1: Improving Dashboard Usability
Scenario:
A company has been using a text-heavy dashboard that employees find difficult to interpret.
Tasks:

1. Identify the key challenges employees might face with the existing dashboard.
2. Suggest strategies for improving usability and comprehension.
3. Propose a new layout incorporating better visual elements.

Case Study 2: Real-Time Hospital Dashboard
Scenario:
A hospital needs a real-time dashboard to monitor patient vitals and staff availability.
Tasks:

1. List the key features that should be included in the dashboard.
2. Discuss potential challenges in designing a real-time healthcare dashboard.
3. Suggest ways to ensure data accuracy and user-friendliness.

Case Study 3: Analyzing COVID-19 Data

Scenario: A health organization needs to track and visualize COVID-19 cases across different regions. They have data on daily infections, recoveries, and deaths.

Tasks:

1. Identify the best types of visualizations for displaying trends in COVID-19 data.
2. Discuss how different visualization choices impact public understanding.
3. Recommend an improved visualization strategy for better public awareness.

Case Study 4: Sales Performance Analysis

Scenario: A retail company wants to analyze its sales performance across various locations. The dataset includes monthly revenue, customer demographics, and product popularity.

Tasks:

1. Determine the most effective visualization techniques for analyzing sales performance.
2. Suggest ways to improve business decisions based on visualized data insights.
3. Create a mock visualization and justify its effectiveness.

Study Tips

1. Focus on simplicity when designing dashboards to enhance readability.
2. Use summarization and exceptions to highlight critical insights without overwhelming users.
3. Choose the right display medium based on the data type and intended message.
4. Practice identifying the best chart types for different data sets.
5. Use visualization tools like Excel, Tableau, or Python libraries (Matplotlib, Seaborn) to experiment with different visualizations.
6. Analyze real-world data visualizations from reports or news articles to understand effective and misleading practices.

Self-Assessment Tasks

Task 1: Dashboard Analysis
Instructions:

1. Analyze an existing dashboard and list five improvements based on best practices.
2. Justify your recommendations with examples.

Task 2: Data-Ink Ratio Application
Instructions:

1. Select an existing graph or chart.
2. Apply Tufte’s data-ink ratio principle to improve its effectiveness.
3. Present the before-and-after version along with an explanation of the changes.

Task 3: Selecting the Right Visualization

Instructions:

1. Given a dataset with sales data over five years, select the best visualization type.
2. Justify your choice based on clarity and effectiveness.

4.24 FURTHER READINGS

- Stephen Few, Information Dashboard Design. Oreilly, 1st Edition, 2006.
- Matthew O. Ward, Georges Grinstein, and Daniel Keim. Interactive data visualization: foundations, techniques, and applications. AK Peters/CRC Press, 2nd Edition, 2015.
- Ben Fry, Visualizing data: Exploring and explaining data with the processing environment. " O'Reilly Media, Inc.", 1st Edition, 2007.

4.25 NPTEL VIDEOS

Data Visualization – Full Course - https://youtu.be/i0QSlGJ3jGU?si=sYPtJEfFi9RIGRBZ

Data Visualization & Story Telling with Tableau - Practical Applications - https://www.youtube.com/live/7IO9d1zFs7U?si=qujelqmHcx7OJUKu

References

The contents of this chapter have been prepared based on the reference book:

• Stephen Few, Information Dashboard Design. Oreilly, 1st Edition, 2006.

All the figures have been referred from this text book.

CONTENTS

UNIT 5

VISUALIZATION SYSTEMS

5.1 Introduction

Data visualization plays a crucial role in transforming raw data into meaningful insights. It enables users to comprehend complex information through graphical representations, making it easier to detect patterns, trends, and outliers. Visualization systems are designed to handle various data types and support different analytical techniques to facilitate data-driven decision-making. With advancements in computing and visualization technologies, modern visualization systems integrate multiple techniques to provide comprehensive data analysis solutions.

5.2 Learning Objectives

After completing this unit, learners will be able to:

1. Understand the fundamental concepts and significance of visualization systems.
2. Classify visualization systems based on data type and analysis type.
3. Explore various toolkits and libraries used for data visualization.
4. Examine text analysis and visualization techniques.
5. Identify key research directions and challenges in visualization.

5.3 Overview

This unit introduces the core principles of visualization systems and their applications in data representation. It covers systems categorized based on data types, such as scientific, multivariate, and graph data, as well as those classified by analysis type, including statistical and spatio-temporal visualization. Modern integrated visualization systems, along with essential toolkits like Prefuse, VTK, and Weave, are explored in detail. Additionally, commonly used libraries such as D3.js, QGIS, and Google Maps are discussed. The unit also delves into research directions, addressing key challenges related to data issues (scale, static vs. dynamic data, spatial vs. non-spatial data), cognitive factors (perception, memory, and learning), system design (interaction techniques, integration of computational methods), evaluation, hardware constraints, and domain-specific applications. By the end of this unit, learners will gain a comprehensive understanding of visualization systems and their role in effective data analysis and interpretation.

5.4 Visualization Systems

The following sections provide an overview of various data and information visualization systems and toolkits. The focus is primarily on freely available software to help students explore visualization technologies. While this chapter covers a selection of tools, there are many other excellent options available. Some commercial software with similar functionalities is also mentioned. Note that the URLs provided were accurate at the time of publication but may have changed, and some tools might no longer be freely distributed.

5.5 Systems Based on Data Type

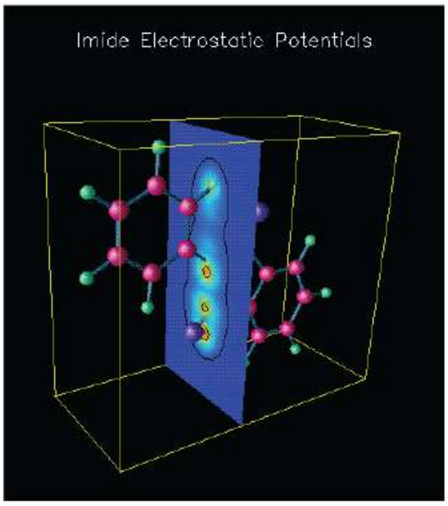
5.5.1 Scientific Data

OpenDX, formerly known as IBM Visualization Data Explorer, is an extensible visualization environment used primarily for scientific and engineering data analysis. A distinguishing feature of OpenDX is its visual programming interface, where users create custom visualizations by assembling components in the Network Editor. Components are linked to enable data communication. The modules in OpenDX are categorized as follows:

- **Import and Export:** Handles loading and saving data in various formats.
- **Flow Control:** Implements loops and conditional execution.
- **Realization:** Maps data to renderable entities such as isosurfaces, grids, and streamlines.
- **Rendering:** Controls attributes like lighting, cameras, and clipping.
- **Transformation:** Applies filtering, mathematical functions, and sorting.
- **Interactor:** Includes widgets like file selectors, menus, sliders, and button boxes.

OpenDX components are pure functions without internal states, and their color changes during execution to indicate progress. Errors highlight the problem-causing component. Each module has an interface for setting parameters, apart from its connections, which are set using point-and-click operations.

OpenDX provides various interactors for setting thresholds, modifying color maps, and controlling execution. An iterator module allows for the animation of visualization sequences. Users can combine different data sources and mappings into a single visualization (Figure 5.1). While OpenDX includes a comprehensive set of modules, users can develop custom modules using its software development kit (SDK). Originally designed for UNIX/Linux, OpenDX can also run on Windows and Mac with an X-server. The software, including source code and documentation, can be downloaded from the OpenDX website. Similar tools include the commercial AVS and the public-domain SCIRun.



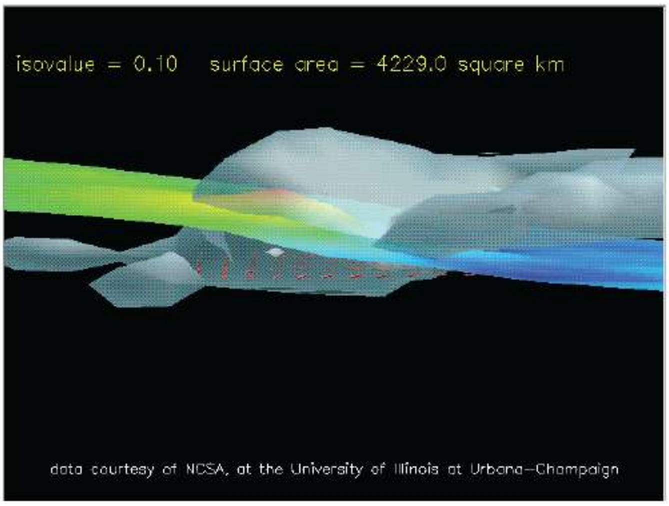


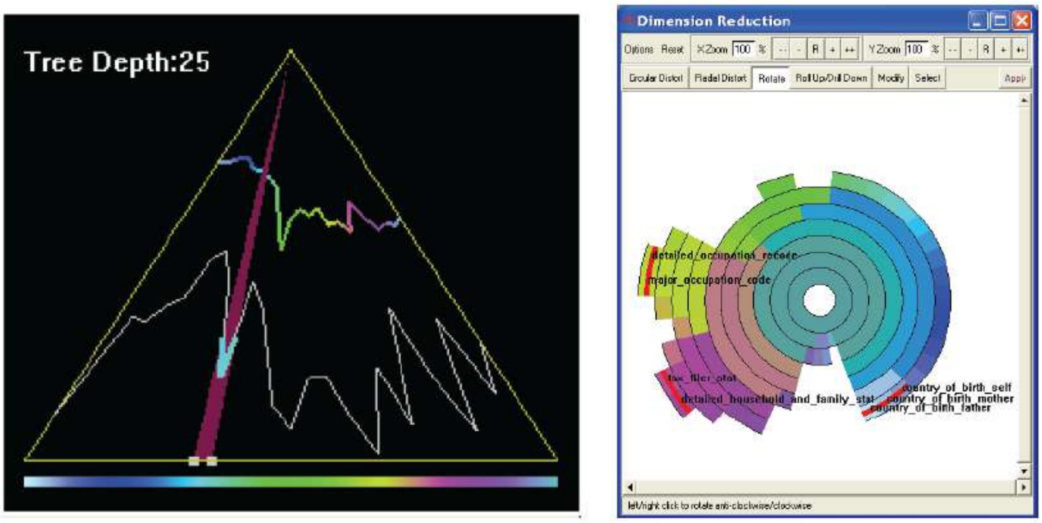
Figure 5.1 Examples of visualizations that can be generated using OpenDX include: one that overlays a slice of a potential field onto a geometric model, and another that integrates temperature, moisture levels, and wind patterns to analyze the characteristics of a storm cloud

5.5.2 Multivariate Data

XmdvTool, developed at Worcester Polytechnic Institute (WPI), is a public-domain visualization tool that integrates five methods for multivariate data visualization: scatterplot matrices, star glyph scatterplots, parallel coordinates, dimensional stacking, and pixel-oriented techniques. The tool features an N-dimensional brush for selection and highlighting across multiple views, allowing users to isolate and analyze data subsets.

Enhancements to XmdvTool include hierarchical parallel coordinates for handling large data sets, where data clusters are represented with varying opacity. Other additions include structure-based brushing and interactive hierarchical displays (Figure 5.2(a)), which allow users to navigate and manipulate hierarchical data structures. The tool also provides a visual hierarchical dimension reduction (VHDR) framework for organizing dimensions into meaningful subspaces (Figure 5.2(b)). Additional functionalities include distance quantification classing (DQC) for handling nominal variables and tools for dimension reordering to reduce visual clutter.

XmdvTool, first released in 1994, is currently in version 7.0. It supports various file formats, including Excel and Oracle databases, and is used in domains such as earth sciences, bioinformatics, and network analysis. The software is available for Windows and UNIX/Linux on its official project website. Commercial alternatives include Spotfire and Tableau Software, which offer more advanced features.



(a) (b)

Figure 5.2 Hierarchical structure visualizations in XmdvTool include: (a) a structure-based brush, enabling navigation and selection within a data hierarchy, and (b) InterRing, which allows users to group data dimensions and choose subsets or averaged values for display

5.5.3 Graph Data

GraphViz is a graph layout tool developed at AT&T Research. Unlike other visualization tools, GraphViz is primarily script-driven, allowing users to specify graphs in the DOT language and apply various layout algorithms. The system supports a wide range of input methods, layout techniques, and output formats for integration into documents, web pages, and applications.

A sample DOT script is provided below:

```
digraph G {
size="6,6";
node [shape=circle,fontsize=8];
rankdir=LR;
st9 -> st9 [label="11/1"];
st9 -> st10 [label="10/1"];
st8 -> st8 [label="10/1"];
st8 -> st0 [label="00/-"];
st7 -> st8 [label="10/1"];
st7 -> st7 [label="00/1"];
st6 -> st6 [label="01/1"];
st6 -> st0 [label="00/-"];
st5 -> st6 [label="01/1"];
st5 -> st5 [label="11/1"];
st4 -> st4 [label="01/1"];
st4 -> st0 [label="00/-"];
st3 -> st4 [label="01/1"];
st3 -> st3 [label="00/1"];
st2 -> st9 [label="11/1"];
st2 -> st7 [label="00/1"];
st2 -> st2 [label="01/1"];
st10 -> st10 [label="10/1"];
st10 -> st0 [label="00/-"];
st1 -> st5 [label="11/1"];
st1 -> st3 [label="00/1"];
st1 -> st1 [label="10/1"];
st0 -> st2 [label="01/-"];
st0 -> st1 [label="10/-"];
```

```
st0 -> st0 [label="00/0"];

}
```

This script defines a directed graph with attributes for nodes and edges. The output graph is shown in **Figure 5.3**, which demonstrates four different layout techniques:

- **Dot:** Uses a layered approach to direct edges uniformly.
- **Neato:** Implements a spring model based on multidimensional scaling.
- **Circos:** Arranges nodes in a circular layout, useful for communication networks.
- **FDP:** Applies a force-directed model with multigrid heuristics for handling large graphs.

GraphViz is compatible with Windows, UNIX, and Mac. The official GraphViz website provides documentation and sample graphs. A commercial alternative with similar functionality is Tom Sawyer Software.

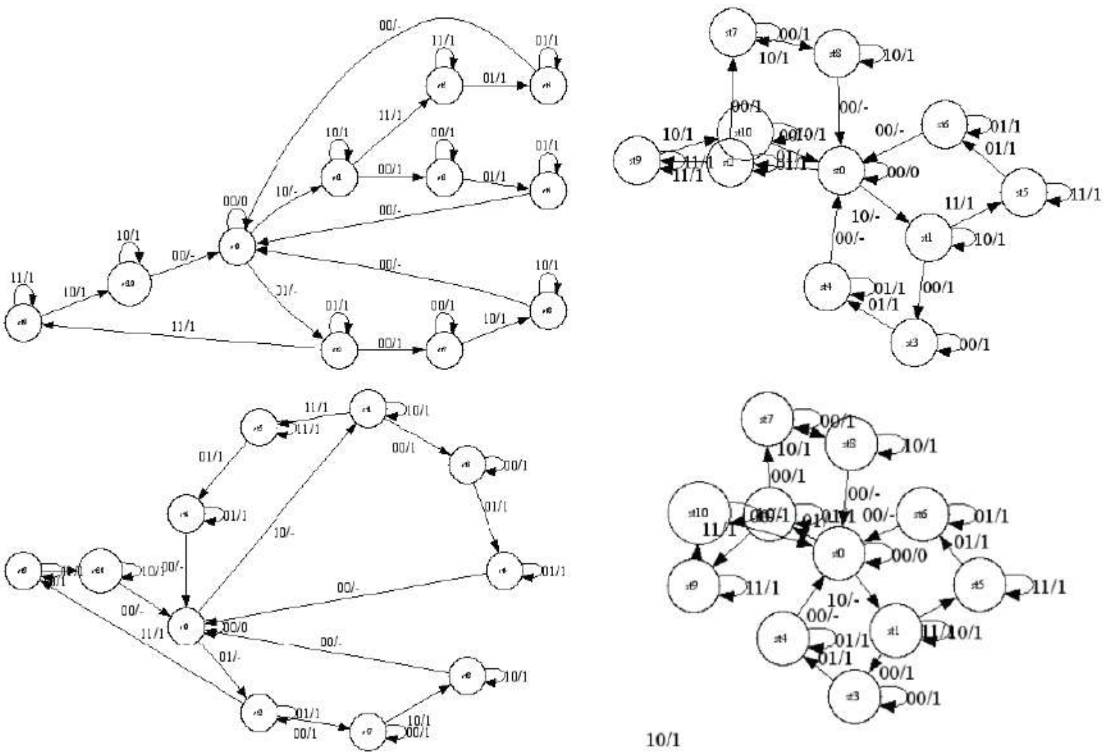


Figure 5.3 Sample outputs from GraphViz, showing the same graph with four different layouts

5.6 Systems Based on Analysis Type

5.6.1 Statistical Visualization

GGobi is an interactive tool designed for multivariate data visualization and analysis. Originally developed by Deborah Swayne, Dianne Cook, and Andreas Buja at Bellcore in the early 1990s, GGobi has undergone continuous evolution with contributions from various developers. The software supports a range of visualization techniques, including:

- **Scatterplots and Scatterplot Matrices** – Used for exploring relationships between multiple variables.
- **Bar Charts and Graphs** – Represent categorical and numerical data effectively.
- **Parallel Coordinates** – Useful for visualizing high-dimensional data.

Each visualization type is accompanied by a dedicated control panel that allows users to customize display settings. GGobi utilizes **color linking** to synchronize data across multiple views. Users can select a specific data dimension to control the color, then use an interactive histogram to refine the assigned color ranges (see Figure 5.4). Options include defining bins by either width or count, ensuring a flexible approach to data grouping.

One of GGobi's most powerful features is its **grand tour functionality**, which provides a dynamic exploration of data through projection space. This feature allows users to view data from multiple perspectives by navigating through different projections. Users can control the speed of the transitions, pause to examine specific data features, and adjust viewing parameters accordingly.

Additional functionalities in GGobi include:

- Integration with the R statistics package for advanced statistical analysis.
- Support for graph drawing techniques, including radial layouts (see Figure 5.5).
- Imputation methods for handling missing values.
- Dimension reduction techniques, such as Principal Component Analysis (PCA) and Multidimensional Scaling (MDS).

GGobi is compatible with **Windows, Mac, and Linux**, utilizing the GTK graphics package. The software, along with documentation and support, is available on the GGobi website.

Other notable statistical visualization tools include:

- **SPSS** (commercial) – A comprehensive tool for statistical analysis and data visualization.
- **SAS** (commercial) – Offers extensive statistical modeling and visualization capabilities beyond standard charting techniques.

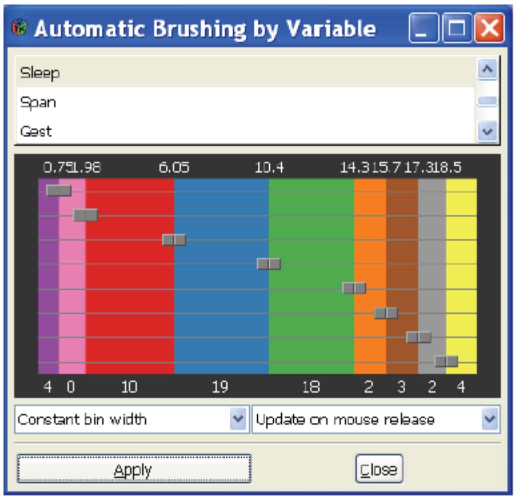


Figure 5.4 Automated color brushing in GGobi. Users can assign any data dimension to control the color, and can adjust the ranges and colors used

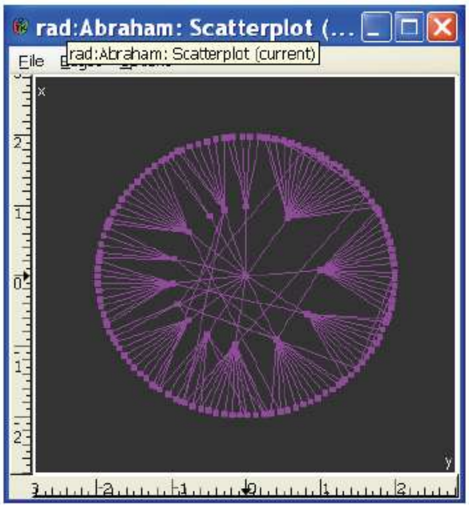


Figure 5.5 An example of graph drawing in GGobi.

5.6.2 Spatio-Temporal Visualization

Macrofocus InfoScope is an advanced interactive tool designed for visually exploring spatio-temporal data. It integrates multiple views, including geographic, thematic, and graphical representations, to enhance data interpretation (Figure 5.6)

- Geographic View – Provides both global and local perspectives of spatial data, allowing users to zoom and focus using set buttons or shift-clicking on locations (see Figure 5.6).
- Theme View – Organizes data points based on similarities using an optimized spring-based MDS algorithm (see Figure 5.7).
- Graphical View – Displays data using parallel coordinates or tabular formats, offering flexibility in data representation (see Figure 5.8).

A key strength of InfoScope is its **linked brushing capability**, which highlights selected data points across all views. Users can apply up to four brushes to isolate specific records for comparative analysis. Additional filtering options allow users to adjust value ranges dynamically, dimming irrelevant data for clearer insights. **Color encoding** further enhances the visual distinction between variables, enabling quick thematic mapping.

Animated transitions are effectively incorporated into InfoScope's interface. These include:

- **Smooth zooming** – Maintains context while exploring different regions.
- **Adjustable parallel coordinates spacing** – Facilitates in-depth data comparison.
- **Theme transitions** – Helps users identify stable versus varying relationships between different thematic representations.

InfoScope is freely available for exploring predefined datasets; however, importing custom data requires purchasing the commercial version from Macrofocus's website.

Other notable spatio-temporal visualization tools include:

- **GRASS GIS** (freeware) – A widely used geographic information system for spatial data analysis.
- **ArcGIS** (commercial) – A comprehensive GIS platform for mapping and spatial analytics.
- **ERDAS IMAGINE** (commercial) – A tool for remote sensing and image analysis.

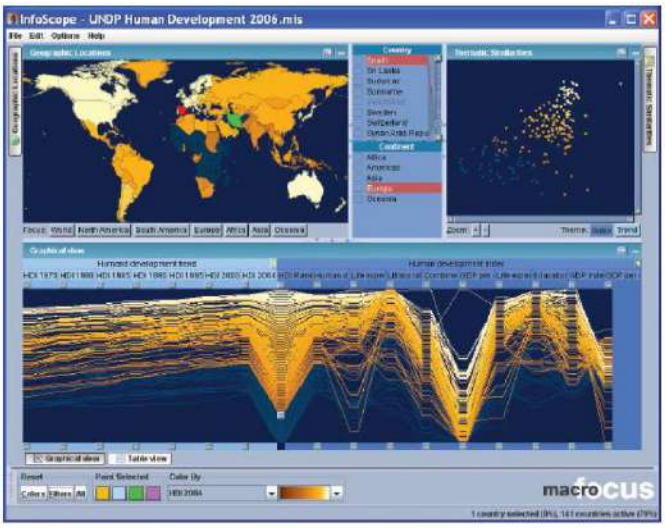


Figure 5.6 A sample InfoScope screen.

Figure 5.7 The Geographic View in InfoScope allows users to focus on specific areas using set buttons or by shift-clicking on a map location.

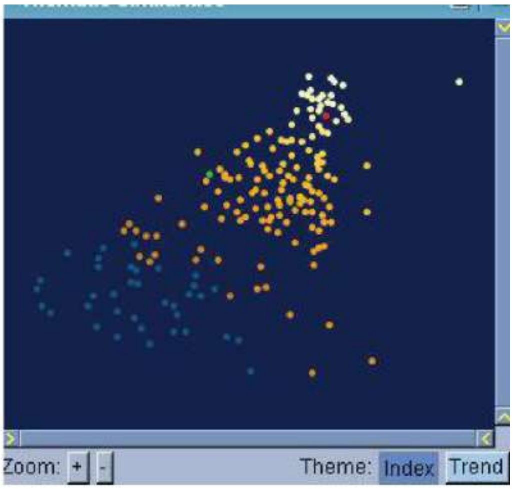


Figure 5.8 The Theme View in InfoScope enables users to choose from various layouts that illustrate relationships between data records

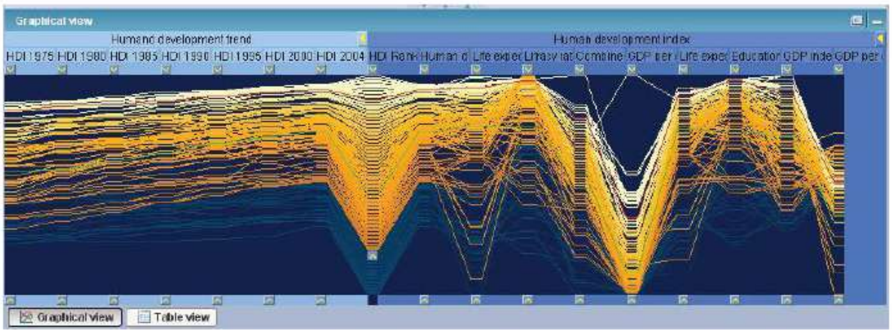


Figure 5.9 The Graphical View in InfoScope features a parallel coordinates plot of the data. Lower values in one of the dimensions have been filtered, causing those data records to appear with dark coloring across all views

5.7 Text Analysis and Visualization

Jigsaw is a text visualization tool developed by John Stasko and his students at the Georgia Institute of Technology. It is designed to explore entities such as people, places, dates, and money, along with the relationships between them. Jigsaw provides various views to present information, including calendar, list, graph, scatterplot, text, and timeline views. These views are shown in Figure 5.10. Each view is displayed in a separate window, updating automatically based on user queries.

All views are linked by default, ensuring that actions performed on a document or entity in one view trigger updates in others. Users have the option to toggle listening for updates in any window. Jigsaw is often used across multiple screens to maximize its analytical potential. This tool is particularly useful for analyzing large collections of data, with applications in military intelligence, law enforcement, and journalism. However, while Jigsaw helps in identifying critical documents and connections, it does not replace the need for reading documents but rather optimizes the selection process.

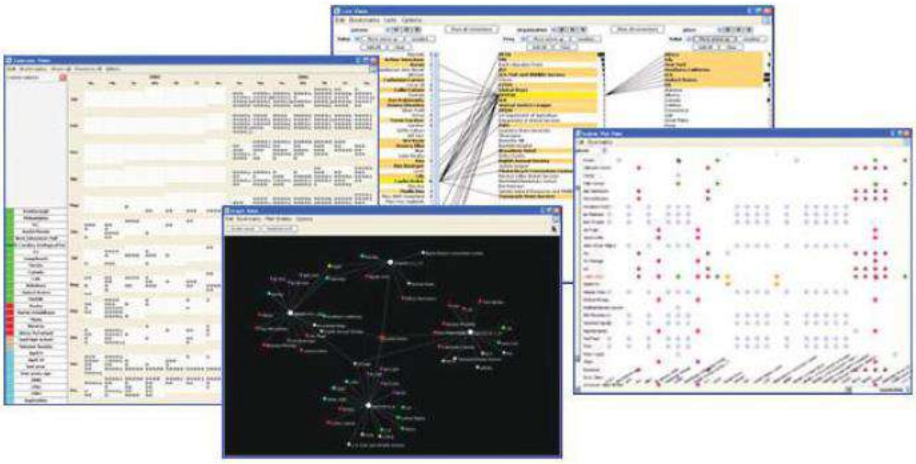


Figure 5.10 Examples of different views in Jigsaw include the list view, scatterplot view, graph view, and timeline view, arranged in a clockwise order starting from the top.

5.7.1 Modern Integrated Visualization Systems

Tableau is a commercial software package initially developed by Pat Hanrahan and his students at Stanford. It supports data analysis through modern interactive visualizations. Users can import data from various formats and generate visual representations, including graphs, scatterplots, bar charts, pie charts, and maps. Tableau recognizes data types like geographic information and automatically processes them for visualization.

The software offers interactive dashboards that update from data sources and allows users to export annotated visualizations in PDF format. Web publishing tools are also included. Figure 5.11 presents an example of linked displays in Tableau, showing gas field production changes from 2004 to 2005 alongside the paths of hurricanes Rita and Katrina. This example illustrates how data integration and perceptual design enhance analytical insights.

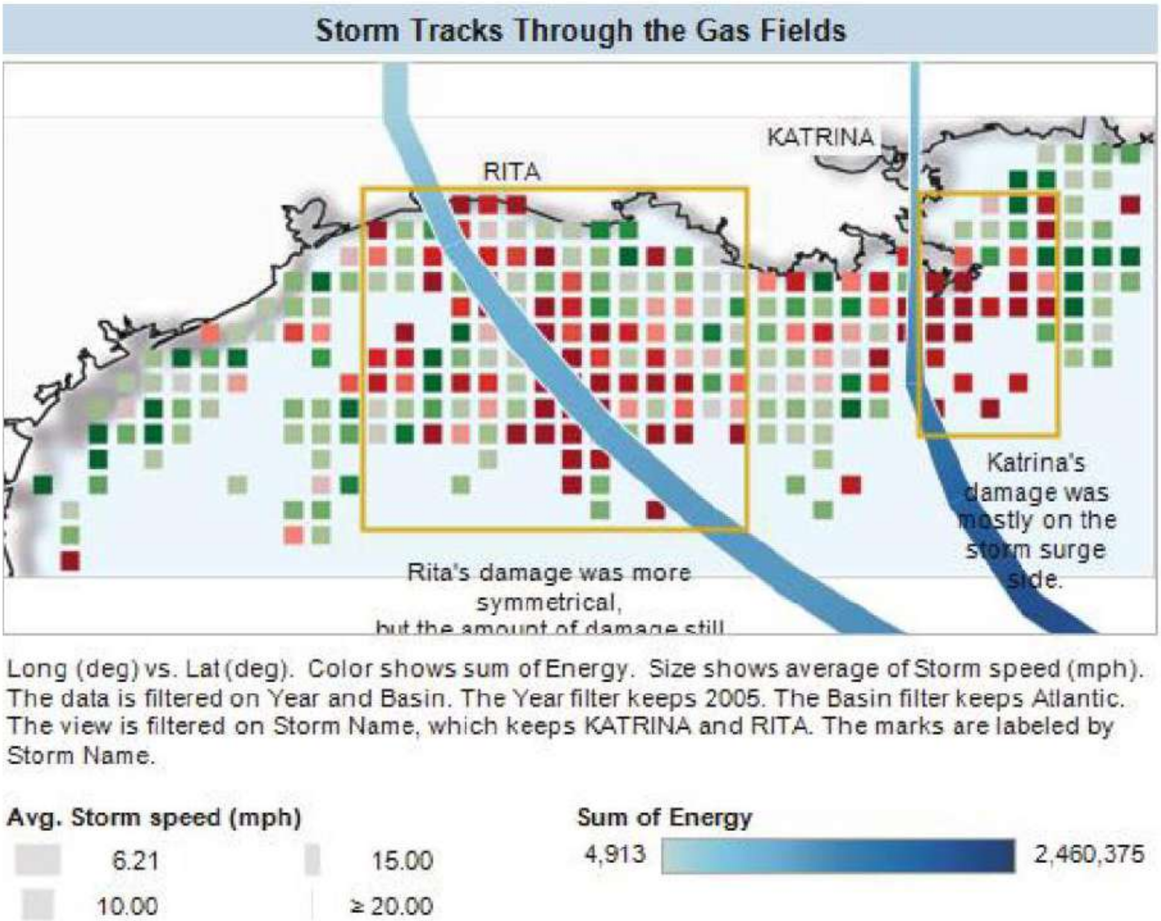


Figure 5.11 An example of linked displays and tables in Tableau shows overlaid paths colored according to hurricane strengths. Significant focus has been given to the interface design and perceptual considerations.

5.8 Toolkits

5.8.1 Prefuse

Prefuse is a toolkit for building visualization applications, available in two components: a Java-based library and an ActionScript-based component called Prefuse Flare. The Java component provides interfaces for developing both standalone and web-based visualization applications, while Prefuse Flare is designed for creating visualizations in Adobe Flash Player. Prefuse supports multiple data structures and visualization techniques, including:

- Table, graph, and tree data structures
- Layout, color, size, and shape encodings
- Distortion and animations
- Interactive operations and dynamic queries
- Text search and SQL-like query language
- Physical force simulation for layouts
- Multiple views (overview+detail, small multiples)

Prefuse follows the information visualization reference model proposed by Ed Chi, dividing the process into data transformations, visual mappings, and visual transformations. Figures

5.12, 5.13, 5.14 showcase various applications built with Prefuse, including network visualizations, document collections, and keyword analysis tools.

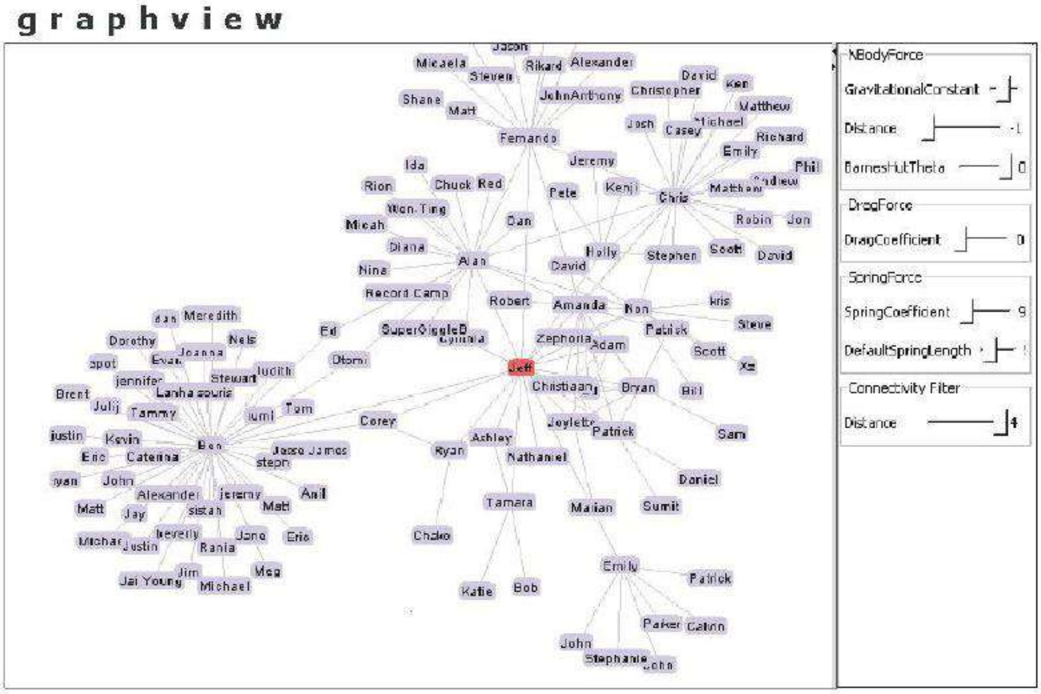


Figure 5.12 An example of a graph generated with Prefuse

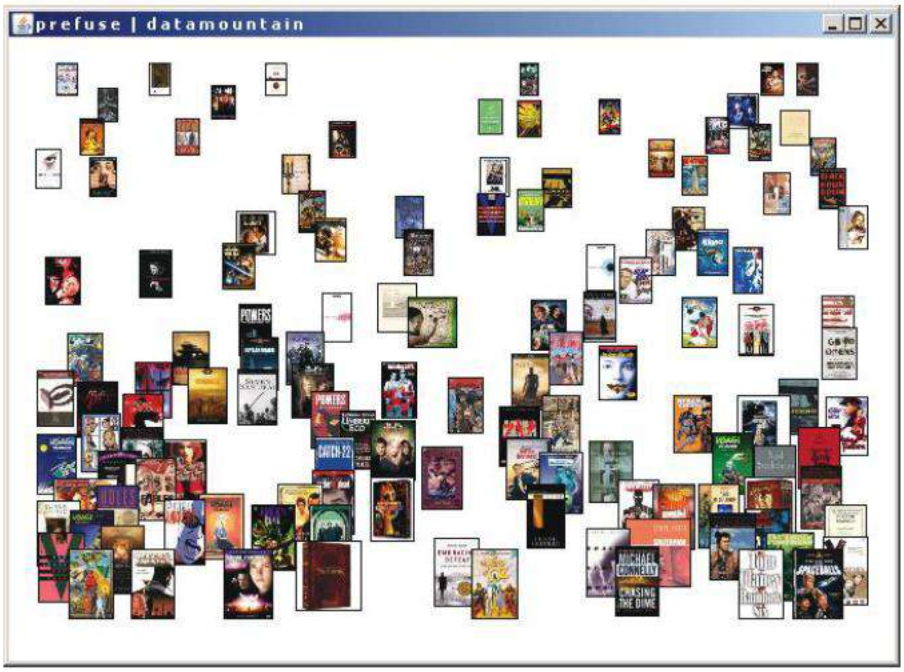


Figure 5.13 Thumbnail view of a document collection using Prefuse

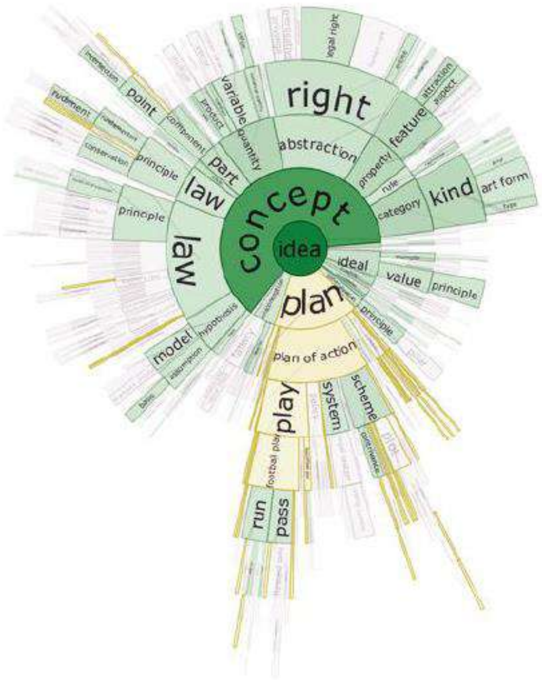


Figure 5.14 Examining keywords of a document and their relationships using Prefuse.

5.8.2 The Visualization Toolkit (VTK)

VTK is an open-source toolkit for creating 3D visualizations, including graphics, modeling, imaging, and both scientific and information visualization. Written in C++, it provides wrappers for other languages and supports interactive UI tools, annotation, and parallel computation. VTK follows a data-flow paradigm, allowing transformations as data moves through modules. It supports structured and unstructured grids, polygonal data, and imaging techniques.

Programming with VTK is similar to OpenGL but provides higher-level abstraction, making it easier and less error-prone. Figure 5.15 illustrate airflow visualization over a delta aircraft demonstrating VTK's capabilities. Kitware, the company behind VTK, has built several applications on top of it, including ITK (image registration and segmentation), ParaView (distributed data visualization), and VolView (interactive volume visualization).

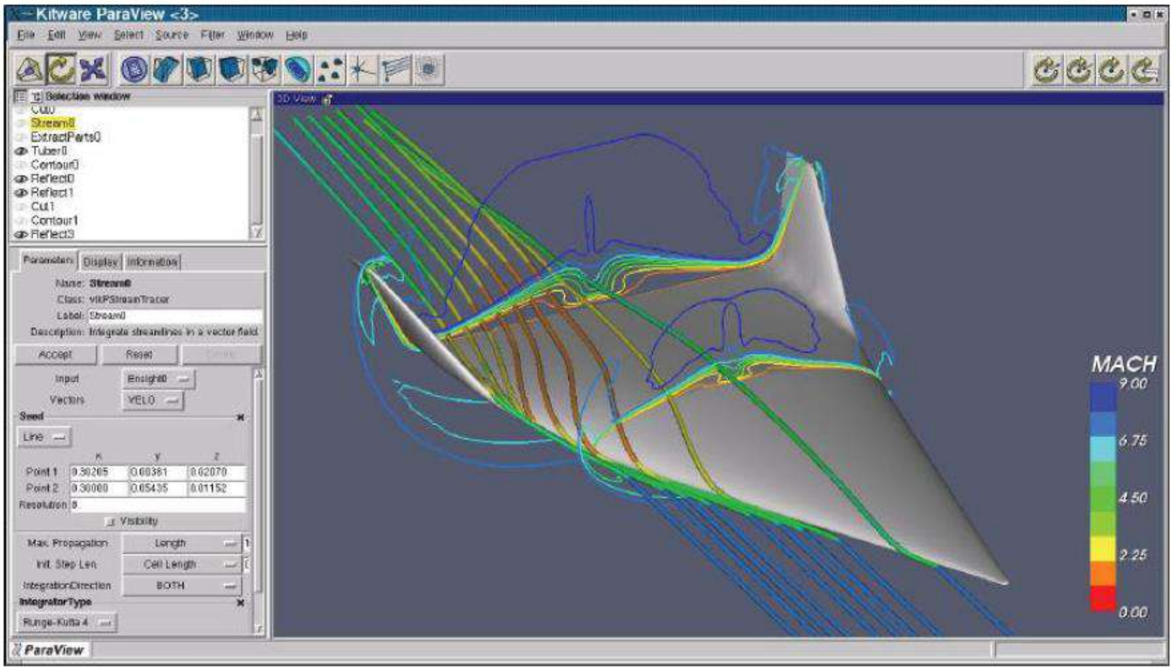


Figure 5.15 Airflow streamlines over a delta aircraft. (Image from Kitware, http://www.vtk

5.8.3 Weave

Weave is an open-source, web-based visualization toolkit developed by Georges Grinstein and his students at the University of Massachusetts Lowell. It was funded by the Open Indicators Consortium. Weave supports various data sources, including SQL databases, spreadsheets, and open data platforms like CKAN and data.gov.

Weave provides geographic visualizations (as shown in Figure 5.16) and supports multiple visualization types, including scatterplots, bar charts, pie charts, heatmaps, and parallel coordinates. Users can access and compare data from different locations, interact with multiple visualizations in a browser, and collaborate with other users. Weave also supports annotation, storytelling, and session history, allowing users to customize and persist preferences.

Technologies used in Weave’s client interface include Adobe Flex, JavaScript, and Flex SharedObjects. The software enables organizations to set up their own secure servers, facilitating remote data access and collaborative analysis across locations. Some public implementations of Weave include MetroBostonDataCommon.org and RiDataHub.org.

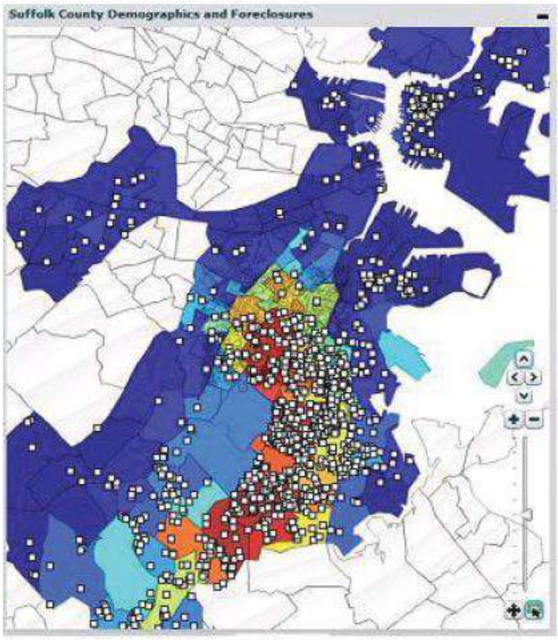


Figure 5.16 A visualization created with Weave shows foreclosures as dots and represents black population percentages using color in the Boston area.

5.9 Libraries

5.9.1 D3.js

D3, short for Data-Driven Documents (see Figure 5.17), was initially developed as a successor to Protovis by Bostock, Heer, and Ogievetsky of the Stanford Visualization Group. It is a framework designed for creating and controlling interactive visualizations that run in web browsers. D3 follows a functional coding style and is compatible with modern browsers. The core library has minimal requirements, including JavaScript, the W3C DOM API, Scalable Vector Graphics (SVG), HTML5, and Cascading Style Sheets (CSS) standards.

D3 enables embedding interactive and dynamic visualizations within an HTML page and facilitates data-driven transformations of documents. Its primary goal is to manipulate the Document Object Model (DOM) while ensuring better compatibility and reusability of the library. With D3.js, developers can utilize prebuilt JavaScript functions to select elements, create SVG objects, style them, and add transitions, dynamic effects, or tooltips.

D3 allows binding data sets to SVG objects using simple functions to generate a variety of visualizations. The data formats it supports include JSON, but developers can write JavaScript functions to process other formats and extend its functionality.

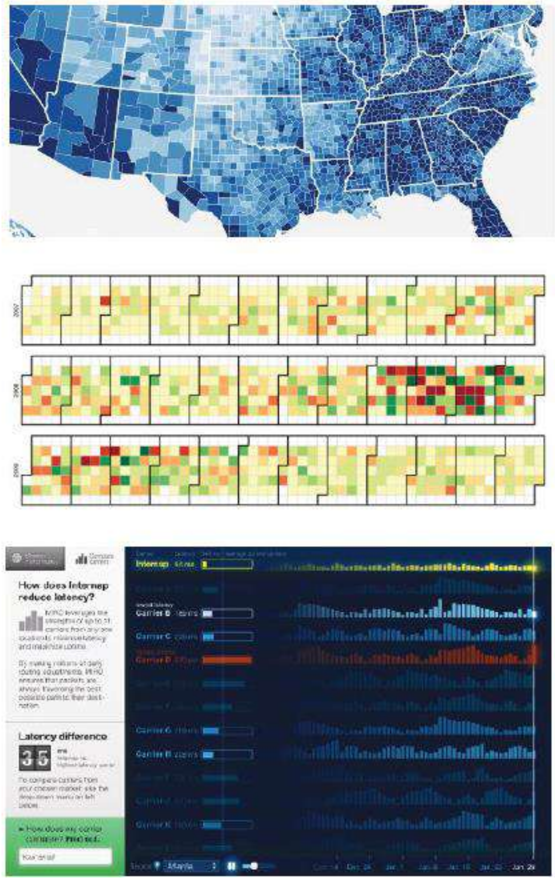


Figure 5.17 Some example visualizations generated with D3.js

5.9.2 QGIS

QGIS is an open-source geographic information system (GIS) and an official project of the Open Source Geospatial Foundation (OSGeo). It is cross-platform, running on Linux, Unix, macOS, Windows, and Android, and supports a wide range of vector, raster, and database formats. QGIS provides tools for generating maps on the web (see Figure 5.18).

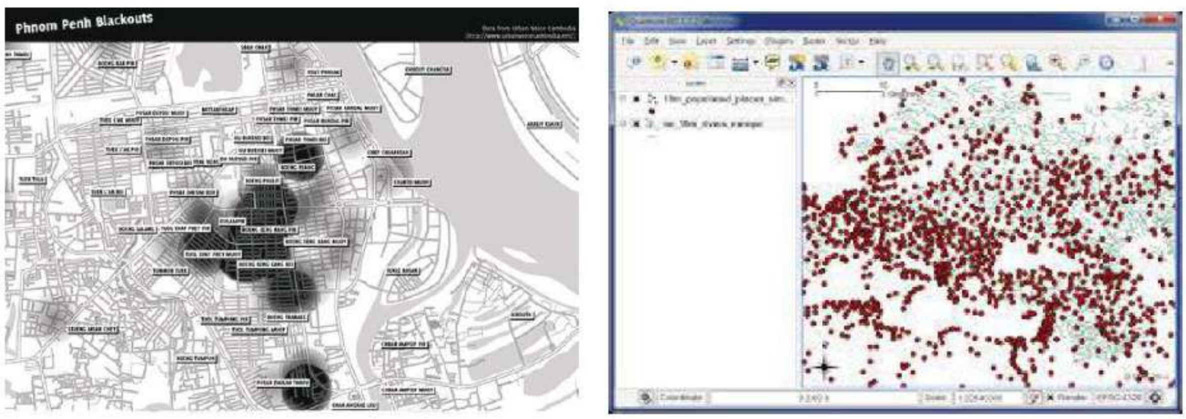


Figure 5.17 Some example visualizations generated with QGIS

5.9.3 Google Maps

Google Maps is a web-based application that offers various map-related services, including directions for driving, walking, and mass transit, as well as real-time traffic information and aerial views (see Figure 5.19). Users can interact with maps by adding layers, such as traffic and directions, and choosing from different map styles.

Although Google Maps is free to use, it remains a proprietary application, and certain features may require authentication, such as creating a profile or obtaining a license. Google Maps provides functionalities that enhance application development, including voice-guided navigation, restaurant recommendations based on location, and public transportation schedules. It functions similarly to a mashup library, allowing integration with other data sources and applications.

Figure 5.18 Sample Google Map visualizations

5.9.4 Circos

Circos is a radial graph visualization tool developed by Martin Krzywunski at Canada's Michael Smith Genomic Sciences Center (see Figure 5.20). Originally designed for genomic data visualization, Circos has since been applied to various domains, with its primary strengths remaining in biological data representation.

Circos visualizes relationships between records within a circular layout, where arcs indicate connections between different points. This approach enables a compact and structured representation of data. With appropriate ordering, similar to RadViz, Circos can help in identifying groupings and patterns within the dataset.

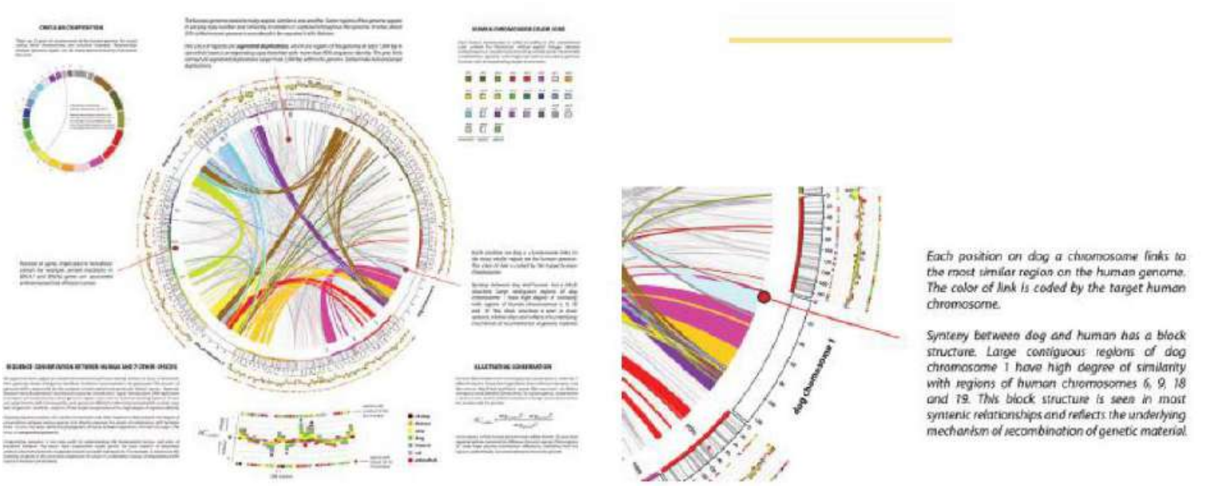


Figure 5.20 Circos visualization with zoomed in portion (from http://circos.ca/guide/genomic/img/circos-conservation.png)

5.10 Research Directions in Visualization

Visualization has become a well-established field, leading researchers to explore its future directions. Several research agendas have been published in recent years, including:

- Chris Johnson's list of top problems in scientific visualization [1]
- Chaomei Chen's list of unsolved problems in information visualization [2]
- MacEachren and Kraak's report on geovisualization research challenges [3]
- The NVAC research and development agenda for visual analytics [4]
- The NIH/NSF visualization research challenges report [5]
- The Grand Challenge in Information Visualization panel [6]

The following sections identify common themes within these reports, providing an overview of key research directions in visualization.

5.11 Issues of Data

Many current and future research efforts focus on expanding the types of data that can be effectively visualized. Key issues include:

5.11.1 Scale

Handling increasing data sizes is a major challenge. Scientific data, both acquired and simulated, has been growing exponentially, surpassing Moore's Law. While past research focused on small datasets, modern challenges involve gigabytes and terabytes of data. Techniques such as clustering, sampling, and abstraction are being explored for large-scale data visualization, including graph visualization, genetic sequence analysis, and flow simulations.

5.11.2 Static vs. Dynamic Data

Most visualization techniques have assumed static data, but there is growing interest in visualizing dynamic data. Streaming data sources require real-time analysis due to both urgency and storage limitations. Analysts are interested in tracking data changes over time rather than viewing isolated snapshots.

5.11.3 Spatial vs. Non-Spatial Data

Many application areas involve both spatial and non-spatial data. To enable effective analysis, researchers are integrating spatial visualization techniques from scientific visualization with non-spatial approaches from information visualization.

5.11.4 Nominal vs. Ordinal Data

Most graphical attributes used in visualization, such as position, size, and color, are quantitative, whereas many datasets contain nominal data (e.g., names, addresses). Effective mappings must be developed to avoid misleading interpretations. Solutions include using perceptually neutral color schemes and deriving numeric mappings that preserve data relationships.

5.11.5 Structured vs. Unstructured Data

Data varies in structure, from highly organized numerical tables to unstructured text. Intermediate forms, such as emails containing both structured (sender, time) and unstructured (message body) components, present unique challenges. Research focuses on extracting useful information from unstructured data and using structured components for indexing. While text analysis has advanced, much work remains.

5.11.6 Time as a Variable

Time is a fundamental data attribute. Time-based visualization methods include volume visualization over time and interactive time controls. Spatio-temporal databases are gaining prominence as more data becomes publicly available. Researchers seek methods to analyze time as an additional variable, uncovering temporal patterns in data.

5.11.7 Data Quality and Uncertainty

Visualization systems often assume complete and reliable data, but real-world datasets frequently contain missing or erroneous values. Incomplete data may result from defective sensors or manual entry errors. Researchers are developing visualization techniques that incorporate uncertainty, enabling analysts to assess data quality in decision-making.

5.12 Issues of Cognition, Perception, and Reasoning

Visualization research must consider how human cognition and perception influence data interpretation. While some work has leveraged Gestalt principles, more study is needed on how users solve problems with interactive visual displays.

5.12.1 Higher-Level Visual Analysis

Rather than simply identifying trends, visualization should support users in forming mental models of entire phenomena. Analysts must integrate multiple insights to cope with complex data, using visualization to explore relationships and confirm hypotheses.

5.12.2 Visualization in Learning

Visualization can enhance learning by accommodating different cognitive styles, such as concrete vs. abstract or sequential vs. random learning. Effective educational visualization tools must be designed based on learning theories.

5.12.3 Memory and Visualization

Visualizations can aid memory by structuring intermediate results and supporting evidence on-screen. This parallels note-taking, which reduces reliance on memory recall. Research is needed to optimize visualization for memory retention.

5.13 Issues of System Design

Developing effective visualization tools requires seamless integration of computational and interactive visual analysis. While some visualization systems incorporate computational methods like clustering and statistical modeling, true synergy between visual and computational techniques remains an open challenge.

5.13.1 Integrating Visual and Computational Methods

Certain tasks benefit more from human cognition, while others are best handled computationally. For example, analysts can use visual overviews to guide parameter selection for filtering and clustering algorithms. Computational models, in turn, refine visual exploration by identifying patterns in large datasets.

5.13.2 Advanced Interaction Techniques

Interaction methods have evolved more slowly than visualization technologies. Many believe existing interaction paradigms do not align well with analytical tasks. Novel techniques are needed to help users explore causality, conduct "what-if" analyses, and interpret underlying data-generating processes.

5.13.3 Democratization of Data

Visualization tools have traditionally required expert users. However, with the rise of interactive visualization in media and online platforms, there is growing demand for user-friendly visualizations accessible to the general public.

5.13.4 Towards a Science of Visualization

Current visualization development relies on experience, heuristics, and pragmatic research. Efforts are underway to automate visualization generation based on dataset characteristics and analytical tasks. Evaluation metrics and systematic frameworks could formalize visualization as a scientific discipline.

5.14 Issues of Evaluation

In the early days of visualization research, rigorous evaluation was rarely performed. The assumption was that visualization was inherently better than no visualization, and qualitative comparisons of images sufficed. More recently, formal evaluation processes have gained traction, enabling quantification of improvements and validation of visualization's benefits in analysis and decision-making. Despite these advances, several key questions remain:

- How important are aesthetics in designing visualizations, and how can they be measured?
- How can we leverage human perceptual and cognitive limitations to improve visualizations?
- How do we measure the benefits of visual analysis compared to traditional computational methods?
- What usability metrics are most important for different user categories (novices vs. experts) and domains?
- How do we quantify information content, distortion, or loss in a visualization and improve accuracy?
- What are the trade-offs between long, longitudinal studies with a few users versus short-term studies with many users?
- What mix of domain and visualization knowledge is necessary for effective tool design?

These evaluation challenges will continue to shape research. Competitions such as the VAST challenges contribute to progress, but evaluation remains crucial in moving visualization from ad hoc methods to a structured science. Drawing from fields like human-computer interaction and sociology can help validate visualization techniques.

5.15 Issues of Hardware

Technological advancements require reassessing visualization applications to maximize their benefits. Several hardware trends influence visualization:

Hand-held displays: Mobile phones, tablets, and other portable devices present opportunities for interactive data representations. Potential applications include real-time maintenance visualization for aircraft and buildings, crisis management for emergency responders, and rapid risk assessments for border security. The challenge is to design effective visual solutions within limited screen space and interactivity options.

Display walls: Large-scale multi-panel displays (10–30 feet in each direction) are increasingly used in control centers and data analysis. Simply porting desktop solutions to large displays is suboptimal; redesigning environments to effectively arrange different views and information types can enhance analysis.

Immersive environments: Virtual and augmented reality systems have been used for applications such as virtual wind tunnels, architectural walkthroughs, and medical simulations. Low-latency rendering remains a challenge. Devices like Google Glass and augmented reality systems offer new visualization possibilities in both real and projected environments.

Graphical processing units (GPUs): Graphics hardware has advanced faster than general-purpose CPUs, driven by the gaming industry. Visualization research increasingly focuses on adapting algorithms to GPU architecture, requiring significant redesigns rather than direct porting from CPUs.

Interaction devices: New input technologies expand visualization possibilities. Force-feedback devices aid in virtual surgery training, while game console controllers (e.g., Wii wands) offer gesture-based interaction. Head and eye tracking, interactive tables, and brain-computer interfaces present further opportunities for intuitive visual interactions.

5.16 Issues of Applications

Advancements in visualization often stem from specific application domains before being generalized. Research issues can be categorized into domain-specific challenges and cross-domain adaptations.

Depth-based innovations: Effective visualization tools require a deep understanding of the target domain. Designers focusing solely on data formats risk creating tools that are ineffective for domain experts. Conversely, domain specialists without visualization expertise may produce suboptimal designs. Successful visualization systems typically arise from close collaboration between both groups. Overcoming barriers such as time constraints, geographic distance, and institutional hurdles is crucial. Encouraging domain experts by demonstrating productivity gains and novel insights can improve adoption. Seamless integration with existing analysis tools also enhances acceptance.

Breadth-based innovations: Visualization has applications across society, as information overload increases while display devices become ubiquitous. Graphical data representation is replacing textual communication in areas such as weather forecasting, stock market trends, and health statistics. Expanding visualization to new domains requires identifying appropriate representations and interaction metaphors. Additionally, many datasets exist in unstructured formats, limiting their usability. Advances in visualizing unstructured data and automated data structuring will unlock further applications.

Summary

This unit provides an in-depth exploration of visualization systems, focusing on their classification, applications, and key challenges. It discusses different data and analysis types, essential toolkits and libraries, and emerging research directions in visualization. Additionally, it highlights the role of cognition, perception, system design, evaluation, and hardware considerations in visualization. The unit emphasizes the significance of visualization in extracting insights from complex data and supports the development of effective data-driven decision-making strategies.

Keywords

Visualization systems, data representation, scientific data, multivariate data, graph data, statistical visualization, spatio-temporal visualization, text analysis, toolkits, data issues, cognition, system design, evaluation, hardware constraints.

.Learning Outcomes

Upon completing this unit, learners will be able to:

1. Explain the role and importance of visualization in data analysis.
2. Differentiate between various visualization system classifications.
3. Identify and describe key visualization toolkits and libraries.
4. Apply visualization techniques for textual data analysis.
5. Discuss emerging trends and challenges in the field of visualization.

Exercises

Objective Questions

1. Which of the following is NOT a component of OpenDX?
a) Import and Export
b) Flow Control
c) Parallel Coordinates
d) Rendering

2. What is the primary programming language used for developing VTK?
a) Python
b) Java
c) C++
d) JavaScript

3. Which software is known for its N-dimensional brushing feature in multivariate data visualization?
a) OpenDX
b) XmdvTool
c) GraphViz
d) Tableau

4. Which of the following best describes the primary purpose of a heatmap in data visualization?
a) Displaying categorical data distribution
b) Representing hierarchical relationships
c) Showing intensity variations across a dataset
d) Comparing trends over time

5. What is the main advantage of using interactive visualizations over static visualizations?
a) They are easier to create
b) They provide more detailed information through user interaction
c) They require less computational power
d) They eliminate data bias

6. Which of the following is NOT a common challenge in data visualization?
a) Data distortion due to improper scaling
b) Misleading interpretations from poorly designed visuals
c) Increased data accuracy through visualization
d) Overloading users with too much information

7. What is one of the primary challenges in handling increasing data sizes in visualization?

a) Reducing color complexity
b) Managing gigabytes and terabytes of data
c) Avoiding interactive features
d) Eliminating text annotations

8. Which factor is essential for democratizing data visualization?

a) Increased complexity in interfaces
b) Exclusive tools for experts
c) User-friendly visualizations accessible to the public
d) Reducing interactive capabilities

9. What is one of the primary issues in evaluating visualization effectiveness?

a) Measuring aesthetic appeal only
b) Quantifying benefits compared to traditional computational methods
c) Avoiding human cognitive considerations
d) Reducing usability studies

10. What is a challenge of visualizing nominal data?

a) Lack of numerical attributes
b) Over-reliance on spatial positioning
c) Excessive use of animation
d) Difficulty in identifying patterns

Answers:

1. c) Parallel Coordinates,
2. c) C++,
3. b) XmdvTool
4. c) Showing intensity variations across a dataset
5. b) They provide more detailed information through user interaction
6. c) Increased data accuracy through visualization
7. b) Managing gigabytes and terabytes of data
8. c) User-friendly visualizations accessible to the public
9. b) Quantifying benefits compared to traditional computational methods
10. a) Lack of numerical attributes

Short Questions

1. What are the main visualization techniques supported by GGobi?
2. Describe the significance of the Network Editor in OpenDX.
3. What are the primary views available in Jigsaw for text visualization?
4. How does visualization aid in memory retention?
5. What are some challenges in integrating structured and unstructured data in visualization?

Long Questions

1. Explain how InfoScope integrates multiple views for spatio-temporal visualization and describe its key features.
2. Compare and contrast GraphViz and Prefuse in terms of their approach to graph visualization.
3. Explain the significance of evaluating visualization tools. What are some key questions researchers consider when assessing their effectiveness?

Collaborative Learning Tasks

Task 1: Exploring an Open-Source Visualization Tool

Instructions**:**

1. Form a group and choose an open-source visualization tool (such as D3.js or VTK).
2. Research its features, applications, and advantages.
3. Create a presentation summarizing your findings and explaining its potential use cases.

Task 2: Analyzing Data Using Visualization Software

Instructions:

1. Work with your peers to analyze a dataset using Tableau or GGobi.
2. Compare different visualization techniques and assess their impact on data interpretation.
3. Discuss your observations and present your insights to the class.

Task 3: Designing a Climate Data Dashboard
Instructions**:**

1. Collaboratively design a dashboard integrating multiple visualization techniques to display climate data.
2. Justify your selection of visualization methods based on data clarity and effectiveness.
3. Share your dashboard and explain your design choices.

Task 4: Data Quality Analysis
Instructions:

1. Investigate how uncertainty is represented in visualization.
2. Identify different techniques used to indicate missing or erroneous data.
3. Present real-world applications where uncertainty visualization is crucial.

Task 5: Exploring Interaction Techniques
Instructions:

1. Choose an advanced interaction technique (e.g., gesture-based, eye tracking, or brain-computer interfaces).
2. Discuss its potential applications and limitations in visualization.
3. Present your findings and suggest improvements for usability.

Case Studies

Case Study 1: Scientific Data Visualization

A research team is working with climate data, requiring an interactive tool to visualize temperature variations and wind patterns over time. They are considering OpenDX and InfoScope. Analyze the strengths and weaknesses of both tools and recommend the best choice.

Case Study 2: Text Analysis with Jigsaw

A news agency wants to use Jigsaw to analyze large text corpora for investigative journalism. Explain how Jigsaw's linked views and entity relationships can assist journalists in uncovering hidden connections.

Case Study 3: Visualization for Crisis Management
Emergency responders require real-time visualization tools for disaster management. They need to analyze weather patterns, evacuation routes, and resource distribution efficiently. How can mobile and large-scale display solutions improve their decision-making process?

Study Tips

1. Experiment with different visualization tools by working on small projects to understand their functionalities.
2. Compare multiple tools for similar visualization tasks to grasp their unique strengths and limitations.

3. Practice interpreting visualizations critically to identify insights and potential biases in data representation.
4. Focus on key research challenges in visualization, such as data scale, uncertainty, and interaction techniques.
5. Use diagrams and flowcharts to understand the relationship between different visualization approaches.

Self-Assessment Tasks

Task 1: Creating a Visualization

Instructions:

1. Choose a dataset and create a visualization using any free tool mentioned in this section.
2. Explain your choice of visualization method and justify its effectiveness.

Task 2: Analyzing a Dataset Using GraphViz or Prefuse

Instructions:

1. Analyze a real-world dataset using GraphViz or Prefuse.
2. Write a short report on your findings and challenges encountered.

Task 3: Developing a Concept Map

Instructions:

1. Create a concept map connecting different visualization techniques discussed in this section.
2. Describe how these techniques interrelate and their appropriate applications.

Task 3: Evaluating a Visualization Tool

Instructions:

1. Pick a visualization tool and assess its strengths and weaknesses.
2. Compare it to another tool based on usability, features, and effectiveness.
3. Present your evaluation with supporting examples

5.17 FURTHER READINGS

- Matthew O. Ward, Georges Grinstein, and Daniel Keim. Interactive data visualization: foundations, techniques, and applications. AK Peters/CRC Press, 2nd Edition, 2015.
- Ben Fry, Visualizing data: Exploring and explaining data with the processing environment. " O'Reilly Media, Inc.", 1st Edition, 2007.
- Colin Ware, Information visualization: perception for design. Morgan Kaufmann, 2019.

5.18 NPTEL VIDEOS

- Data Visualization: Types, tools and technologies - http://kcl.digimat.in/nptel/courses/video/109102392/L28.html
- Introduction to Data Visualisation Analysis - https://youtu.be/qdnM8Fpvdqc?si=t6V1zlv32KOpLx_l

References

1. Chris R. Johnson. “Top Scientific Visualization Research Problems.” IEEE Comput. Graph. Appl. 24:4 (2004), 13–17.
2. Chaomei Chen. “Top 10 Unsolved Information Visualization Problems.” IEEE Comput. Graph. Appl. 25:4 (2005), 12–16.
3. A. M. MacEachren and M.-J. Kraak. “Research Challenges in Geovisualization.” Cartography and Geographic Information Science 28:1 (2001), 3–12.
4. K. A. Cook and J. J. Thomas. Illuminating the Path: The Research and Development Agenda for Visual Analytics. Los Alamitos, CA: IEEE Computer Society Press, 2005.
5. C. R. Johnson, R. Moorhead, T. Munzner, H. Pfister, P. Rheingans, and T. S. Yoo, editors. NIH-NSF Visualization Research Challenges Report. Los Alamitos, CA: IEEE Computer Society Press, 2006.
6. G. Grinstein, D. Keim, and T. Munzner. “Grand Challenges in Information Visualization.” Panel from VisWeek 2008, Columbus, OH, October 20, 2008. Report available at http://www.cs.uml.edu/~grinstei/ GrandChallengesPanelVisWeek2008.pdf.

The contents of this chapter have been prepared based on the reference book:

Matthew O. Ward, Georges Grinstein, and Daniel Keim. Interactive data visualization: foundations, techniques, and applications. AK Peters/CRC Press, 2nd Edition, 2015.

All the figures have been referred from this text book.