Chapter 2
Creating Visual Representations

In this chapter, we'll take a closer look at the process of generating an artifact of visual representation, or rather the mechanism that creates a visual representation from a certain number of data, using specific computer processes. Without delving too far into technical details, we'll describe this process through a model that we will use as a reference for the interactive visual representation. Furthermore, we will present some common techniques for visualizing linear data structures.

2.1 A Reference Model

Let's imagine that we have at our disposal a collection of data on which we'd like to carry out explorative analysis to identify any possible unknown tendencies or relationships. How can we go about a creating visual representation from this data? As always, good design is the key to success in applications of this kind. Before tackling this delicate and extremely important aspect, however, we'll find out which tools information technology puts at our disposal to realize visual representations.

Computers can help us greatly and, if we don't want to attempt designing everything from scratch, these days there are many varieties of visualization software that can provide us with a complete series of visual templates. But how do these programs work?

Software dedicated to the creation of visual representation of abstract data, even if they differ greatly among themselves, all follow a generation process that can be outlined in Fig. 2.1. Let's take raw data as our starting point, or rather abstract data provided by the world around us. As we saw in the previous chapter, we speak of abstract data when these data don't necessarily have a specific connection with physical space. For example, they may deal with people's names, the prices of consumer products, voting results, and so on. These data are rarely available in a format that is suitable for treatment with automatic processing tools and, in particular, visualization software. Therefore, they must be processed appropriately, before being represented graphically.

R. Mazza, *Introduction to Information Visualization*,
DOI: 10.1007/978-1-84800-219-7_2,

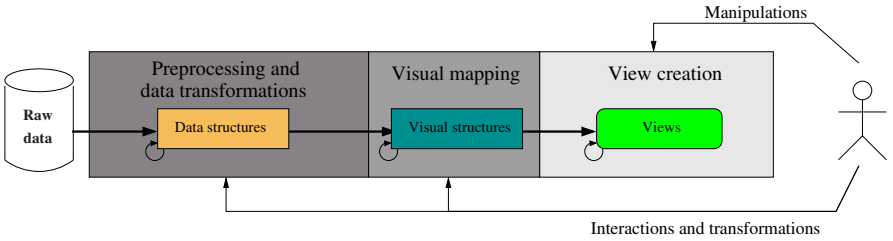

Fig. 2.1 The process of generating a graphical representation.

The creation of a visual artifact is a process that we can model through a sequence of successive stages:

1. preprocessing and data transformations,
2. visual mapping,
3. view creation.

We will describe each of these stages through an example, showing how data are transformed from the original format through to the creation of the visual representation.

2.1.1 Preprocessing and Data Transformations

We use the term "raw" to describe data supplied by the world around us. They can be data generated by tools, like the values of some polluting agents taken from a monitoring station during pollution testing. They can also be generated and calculated by appropriate software, such as weather forecast data. They may even be data linked to measurable events and entities that we find in nature or the social world, like the number of inhabitants or birth rates of the cities in a specific state. In each case, these collections of data (known as *datasets*) are very rarely supplied to us with a precise logical structure. To be able to process these data using software, we have to give them an organized logical structure. The structure usually used for this type of data is tabular—the arranging of data in a table—in a format appropriate for the software that must receive and process them. Sometimes the input data are contained in one or more databases and are, therefore, already available in electronic format and with a well-defined structure. In this case, the raw data correspond to the data located in the databases, and the phases of preprocessing and elaboration involve extracting these data from the database and converting them into the structured format used by the visualization software.

We'll show a concrete example, taken from [43]. Let us assume we want to study how people communicate in a discussion forum—the Internet-based communication tools that allow users to converse through an exchange of messages. The users can write a message on the forum, which all other users of the service can read.

Anyone can reply to the message, thus creating an environment of interactive discussion. Imagine having to carry out an analysis on data relative to the number of messages read and written in a discussion forum. Suppose, for instance, that we wish to quickly single out both the most active users (or, rather, those who read and write a high number of messages in the forum), as well as the users who silently read all of the messages and don't take an active part in the discussion. The tools that offer this type of service usually record every action carried out by the system's users in an appropriate file: the *log* file.

A typical log file of the discussions could have this format:

```
                          .
                          .
                          .
                          .
                          .
[Tue 1 March 2005, 10:22 AM] Luigi "add post"
[Tue 1 March 2005, 10:26 AM] Orazio "view discussion"
[Tue 1 March 2005, 11:02 AM] Luigi "add post"
[Tue 1 March 2005, 02:02 PM] Enzo "view discussion"
[Tue 1 March 2005, 02:04 PM] Enzo "view discussion
                          .
                          .
                          .
                          .
                          .
```

This file will be the source of row data in our system. The preprocessing phase should convert these data into a tabular format.

The data structures can also be enriched with additional information or preliminary processing. In particular, **filtering** operations to eliminate unnecessary data and **calculations** for obtaining new data, such as statistics to be represented in the visual version, can be performed; furthermore, we can add **attributes** to the data (also called *metadata*) that may be used to logically organize the tabular data. The intermediate data structure, of the example we are processing, could therefore look like the following:

User	Read	Posted
Enzo	90	10
Giorgio	134	20
Luigi	89	3
Michele	14	0
Orazio	117	13

In this structure in particular, we have filtered out some information, such as the date and time of each logged event, as they are irrelevant to the current problem. The attributes *read* and *posted* are calculated from the data featured in the log file.

2.1.2 Visual Mapping

The key problems of this process lie in defining which visual structures to use to map the data and their location in the display area. As we have already mentioned, abstract data don't necessarily have a real location in physical space. There are some types of abstract data that, by their very nature, can easily find a spatial location. For example, the data taken from a monitoring station for atmospheric pollution can easily find a position on a geographic map, given that the monitoring stations that take the measurements are situated in a precise point in the territory. The same can be said for data taken from entities that have a topological structure, such as the traffic data of a computer network. However, there are several types of data that belong to entities that have no natural geographic or topological positioning. Think, for example, of the bibliographic references in scientific texts, of the consumption of car fuel, or of the salaries of various professional figures within a company. This type of data doesn't have an immediate correspondence with the dimensions of the physical space that surround it.

We must therefore define the visual structures that correspond to the data that we want to represent visually. This process is called *visual mapping*. Three structures must be defined [8]:

1. spatial substrate,
2. graphical elements,
3. graphical properties.

The **spatial substrate** defines the dimensions in physical space where the visual representation is created. The spatial substrate can be defined in terms of axes. In Cartesian space, the spatial substrate corresponds to x- and y-axes. Each axis can be of different types, depending on the type of data that we want to map on it. In particular, an axis can be *quantitative*, when there is a metric associated to the values reported on the axis; *ordinal*, when the values are reported on the axis in an order that corresponds to the order of the data; and *nominal,* when the region of an axis is divided into a collection of subregions without any intrinsic order.

The **graphical elements** are everything visible that appears in the space. There are four possible types of visual elements: points, lines, surfaces, and volumes (see Fig. 2.2).

The **graphical properties** are properties of the graphical elements to which the retina of the human eye is very sensitive (for this reason, they are also called *retinal variables*). They are independent of the position occupied by a visual element in spatial substrate. The most common graphical properties are size, orientation, color, texture, and shape. These are applied to the graphical elements and determine the properties of the visual layout that will be presented in the view (see Fig. 2.3).

In terms of human's visual perception, not all graphical properties behave in the same way. Some graphical properties are more effective than others from the viewpoint of quantitative values. Cleveland and McGill [11] carried out a study to evaluate the accuracy with which people are able to perceive quantitative values mapped to different properties, graphical elements, and spatial substrates. The study defined

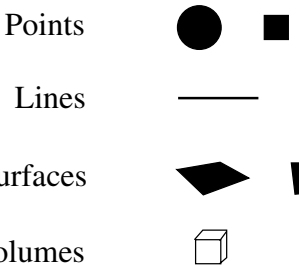

Fig. 2.2 Examples of graphical elements.

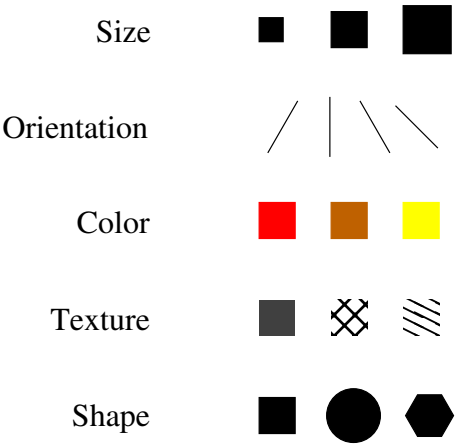

Fig. 2.3 Examples of graphical properties.

a classification that is reported in Fig. 2.4, from which we can deduce that spatial positioning is one of the most accurate ways to perceive quantitative information. The chosen mapping has to make the most important conceptual attributes also become perceptively accurate.

Color has to be given particular attention. In fact, color is the only graphical property in which perception can depend on cultural, linguistic, and physiological factors. Some populations, for example, use a limited number of terms to define the entire color spectrum (in some populations, there are only two words to describe the colors: black and white). It is therefore possible that two people from different cultures may use diverse terminology to identify the same color or may even have different perceptions, given that they might not have a specific term for identifying a determined color on a cognitive level. Studies on perception [65] have demonstrated that, even taking the cultural differences into account, the colors that can be considered primary are white, black, red, green, yellow, and blue. These are the only colors that have the same name all over the world and, consequently, are the colors that must be chosen when it is necessary to map a category attribute to a maximum of six colors. Colin Ware [65] suggests limiting any mapping of categor-

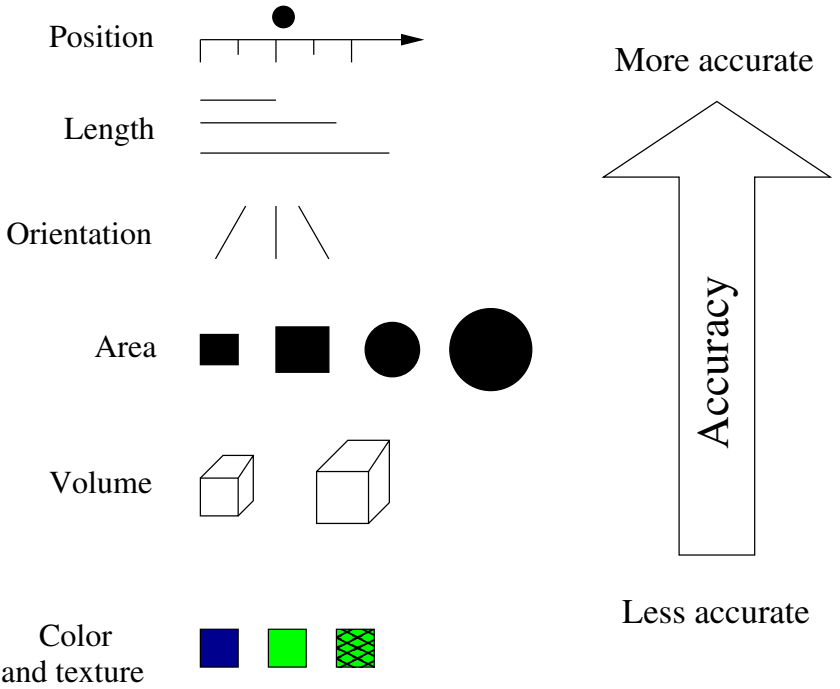

Fig. 2.4 Accuracy in the perception of quantitative values for some graphical and spatial elements.

ical attributes to these six primary colors, but, if necessary, it is possible to extend the list by adding pink, brown, cyan, orange, and purple. To represent quantitative attributes, or where there is an ordering of values, the use of primary colors is not advisable, because (1) there might not be enough primary colors, and (2) our culture does not adopt any convention on the ordering of colors (does blue come before or after yellow?). A clever idea might be to use a convention on a color scale, to be clearly explained in the application (from green to red, for example), or to vary the color intensity to codify the various levels of values (Fig. 2.5).

It is also necessary to bear in mind that a large percentage of the population (in Australia, 8% of males and 0.4% of females) has a particular ocular visual perception problems: *daltonism*.[1] People who suffer from this condition are generally unable to distinguish between red and green, or (less frequently) between yellow and blue. It is therefore important to consider that there are some people with this visual defect and to develop applications in which it is possible to change the color mapping.

Let's return to the example that we had been studying. To continue with the process of generation, we have to associate a visual structure with which to map the

[1] The term "daltonism" originates from the name of the English physicist John Dalton (1766–1844), who was the first to study this defect.

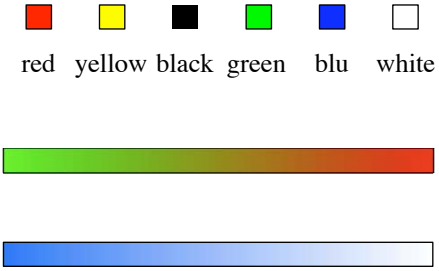

Fig. 2.5 Use of primary colors to define categorical attributes (top) and color scales to define ordinal attributes (bottom).

data that we wish to represent, to the data structures. In the specified case, we have three attributes to represent:

Attribute	Data type
user	categorical
read	quantitative
posted	quantitative

We can resolve to map the attributes *read* and *posted* to the x- and y-coordinates on a Cartesian axis. Since it deals with quantitative data, the mapping can be carried out without any problems. This constitutes the **spatial substrate**. We then choose to represent each element individually in the spatial substrate with a point-type **graphical element**. The graphical element will be square-shaped and colored blue. In this way, we have defined the **graphical property** that will contain the element to be represented in the picture. We also decide to add a further graphical element, comprising a textual tag that contains the values of the attribute *user*, using the same spatial substrate as previously defined. We have therefore completed the visual mapping for all of the attributes in play.

2.1.3 Views

The views are the final result of the generation process. They are the result of the mapping of data structures to the visual structures, generating a visual representation in the physical space represented by the computer. They are what we see displayed on the computer screen. Figure 2.6 represents a possible view for the example we are dealing with.

The visual representation allows for efficient responses to the questions we posed on the analysis of discussions, recognizing who posts the most messages on the forum and who, on the other hand, reads messages without actively participating in

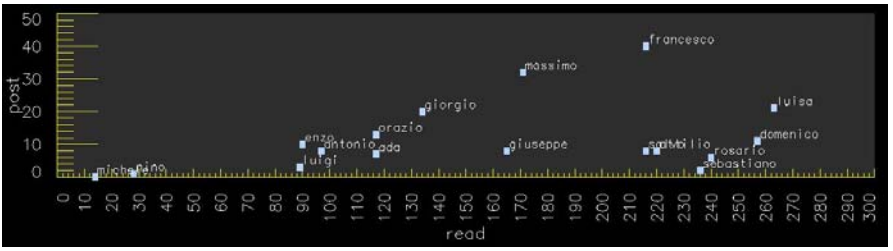

Fig. 2.6 A possible visual representation of the data collected from a discussion forum.

the discussions. In Fig. 2.6, we can identify the users *francesco* and *massimo*, who have been the most active both in posting new messages on the forum and also in reading. The users *rosario* and *sebastiano* have, instead, read many messages but have participated very little with their own messages. Finally, we can immediately single out the users *michele* and *nino,* who have been inactive in both reading and submitting new messages.

The views are characterized by a difficult inherent problem, a quantity of data to be represented that is too large for the available space. This is a problem that we come across rather frequently, given that very often real situations involve a very large amount of data (at times even millions of items). In these cases, when the display area is too small to visibly support all the elements of a visual representation, certain techniques are used, including zooming, panning, scrolling, focus + context, and magic lenses. These techniques will be discussed in more detail in Chapter 7.

2.2 Designing a Visual Application

A generation process, such as the one described previously, should be preceded by good design. Correct design is the key to the success of this type of application. Many prototypes developed in the context of scientific research don't even define what type of user the visualization model is addressing or the purpose of its development.

The main problem in designing a visual representation lies in creating visual mapping that, on the one hand, faithfully reproduces the information codified in the data and, on the other, facilitates the user in the predetermined goal. As we already discussed in Section 1.8, there is no way to know, given a collection of abstract data, which type of visual representation is suitable for such data. This depends on the nature of the data, the type of user it's designed for, the type of information that has to be represented, and its use, but also on the creativity, experience, and ability of the representation's designers. In these cases, the most precious and important information comes to us from potential users of the visual application, those who will use the system and ordain its success or failure. Believe it or not, most authors of works of visual representation of information don't carry out preliminary research

with the users of the system to understand their actual needs, or only afterwards do they effectuate empirical evaluation, when the application prototype has been developed.

The procedure to follow, when creating the visual representations of abstract data, can be outlined in the following steps:

1. **Define the problem** by spending a certain amount of time with potential users of the visual representation. Identify their effective needs and how they work. This is needed to clearly define what has to be represented. Why is a representation needed? Is it needed to communicate something? Is it needed for finding new information? Or is it needed to prove hypotheses? It is necessary to bear in mind the human factors specific to the target audience that the application will address and, in particular, their cognitive and perceptive abilities. This will influence the choice of which visual models to use, to allow users to understand the information.
2. **Examine the nature of the data to represent**. The data can be *quantitative* (e.g., a list of integers or real numbers), *ordinal* (data of a non numeric nature, but which have their own intrinsic order, such as the days of the week), or *categorical* (data that have no intrinsic order, such as the names of people or cities). A different mapping may be appropriate, according to the data type.
3. **Number of dimensions**. The number of dimensions of the data (also called *attributes*) that we need to represent very importantly determines the type of representation that we use. The attributes can be *independent* or *dependent*. The dependent attributes are those that vary and whose behavior we are interested in analyzing with respect to the independent attributes. According to the number of dependent attributes, we have a collection of data that is called *univariate* (one dimension varies with respect to another), *bivariate* (there are two dependent dimensions), *trivariate* (three dependent dimensions), or *multivariate* (four or more dimensions that vary compared to the independent ones).
4. **Data structures**. These can be *linear* (the data are codified in linear data structures like vectors, tables, collections, etc.), *temporal* (data that change in time), *spatial* or *geographical* (data that have a correspondence with something physical, such as a map, floorplan, etc.), *hierarchical* (data relative to entities organized on hierarchy, for example, genealogy, flowcharts, files on a disk, etc.), and *network* (data that describe relationships between entities).
5. **Type of interaction**. This determines if the visual representation is *static* (e.g., an image printed on paper or an image represented on a computer screen but not modifiable by the user), *transformable* (when the user can control the process of modification and transformation of data, such as varying some parameters of data entry, varying the extremes of the values of some attributes, or choosing a different mapping for view creation), or *manipulable* (the user can control and modify some parameters that regulate the generation of the views, like zooming on a detail or rotating an image represented in 3D). The model represented in Fig. 2.1 illustrates at which levels of the process these types of interactions come into play.

The elements just described, to be considered during the design stage, are summarized in Table 2.1.

Problem	Data type	Dimensions	Data structure	Type of interaction
Communicate	Quantitative	Univariate	Linear	Static
Explore	Ordinal	Bivariate	Temporal	Transformable
Confirm	Categorical	Trivariate	Spatial	Manipulable
		Multivariate	Hierarchical	
			Network	

Table 2.1 Variables to consider when designing visual representations.

Each of the possible options described here can point to the use of a specific technique. Furthermore, correct design should also define suitable tools for assessing the effects of the proposed representations on the users' performance. Evaluation is discussed in Chapter 8. In the following sections, we will illustrate the most common types of representation, keeping in mind the most distinctive aspect, which is the number of dimensions. We'll start by looking at linear data. Other data organizations (spatial, hierarchical, and network structures) will be discussed in next chapters.

2.3 Visual Representation of Linear Data

A collection of data is defined as **univariate** when one of its attributes varies with respect to one or more independent attributes. Let's suppose that we have to analyze the gross national product (GNP) realized by some nations in 2000. A tabular version, as shown in Table 2.2, is the most efficient form for immediately identifying the GNP of one of the featured nations. Basically, a specific nation and its corresponding GNP can be singled out immediately from the alphabetically ranked list. It may be interesting, however, to compare the GNP of one nation with that of another. For this particular task, the tabular version featured here is not the ideal solution, and it is better to explore other types of representation.

One possible graphical form is the *single-axis scatterplot* (Fig. 2.7 on the left). It consists of representing a single-axis spatial substrate and positioning a visual element according to the value of the dependent attribute, which in this case is represented by a circular shape to which a label is also attached. We can immediately make out which nations have the highest and lowest GNP, while some groupings are clear. For example, it is instantly noticeable how the GNP of Brazil and Spain differ very little, while Germany has a notably higher GNP than the following nation, France.

Nation	GNP
Argentina	284.2
Brazil	601.7
Canada	713.8
France	1308.4
Germany	1870.2
Italy	1074.8
New Zealand	52.2
Poland	166.5
Portugal	106.5
Spain	561.8
Switzerland	246.2
The Netherlands	370.6

Table 2.2 Gross national product (GNP) of some nations in 2000. Values expressed in billions of USD. Source: The World Bank, *World Development Indicators 2005*.

Another very common form of univariate representation is the bar chart. The data in Table 2.2 can also be represented using the bar chart shown in Fig. 2.7 on the right. Scatterplots and bar charts are two relatively common forms of visual representation of information. Their popularity is due to the fact that they deal with two very simple shapes, immediately clear and understandable. Through the scatterplot, we are able to instantly take in the global distribution of GNP all along the values axis, while the bar chart allows us to make very efficient comparisons between the different nations. On the other hand, only in very rare cases are the data that need to be

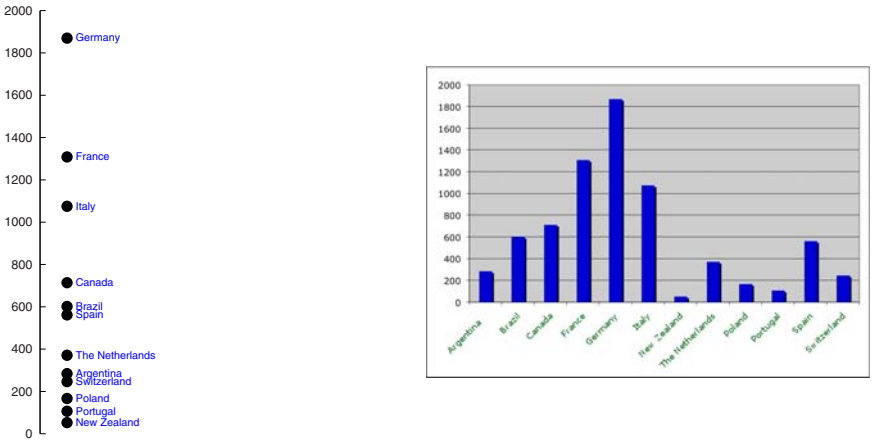

Fig. 2.7 The GNP of various nations visually represented by a single-axis scatterplot (left) and a bar chart (right).

Nation	Export	Import
Argentina	10.9	11.5
Brazil	10.7	12.2
Canada	46.1	40.3
France	28.5	27.3
Germany	33.8	33.4
Italy	28.3	27.3
New Zealand	35.9	34.2
Poland	27.8	34.4
Portugal	31.5	42.8
Spain	30.1	32.4
Switzerland	45.6	39.9
The Netherlands	67.5	62.2

Table 2.3 Total import and export values of some nations in 2000. Values expressed in percentages of the GNP. Source: The World Bank, *World Development Indicators 2005*.

analyzed univariate, and so we are concerned with analyzing the cases in which the number of dependent attributes is greater than one.

When the number of dependent attributes is two, we speak of **bivariate** data representation. Let's suppose that we need to examine the total value of the overseas import and exports goods of these nations. In this case also, the values can be reported in a table (see Table 2.3). However, to have an effective vision of the import and export distribution of these nations, we can represent these values on a two-axis scatterplot (Fig. 2.8).

This type of representation has very high expressive power, given that the most important data (import and export) are mapped onto the axes and, as we have seen in Section 2.1.2, they are the most accurate visual way to perceive quantitative infor-

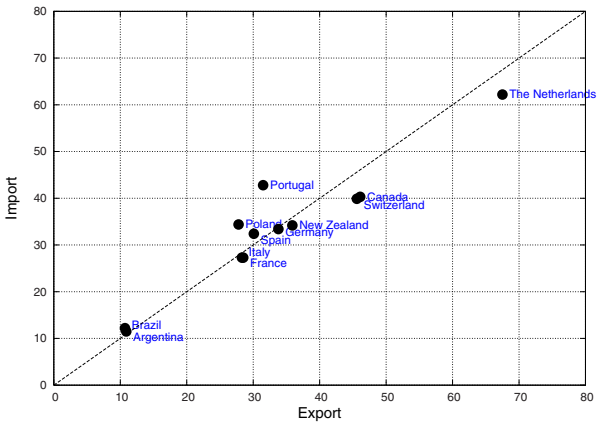

Fig. 2.8 Two-dimensional scatterplot that compares import and export values.

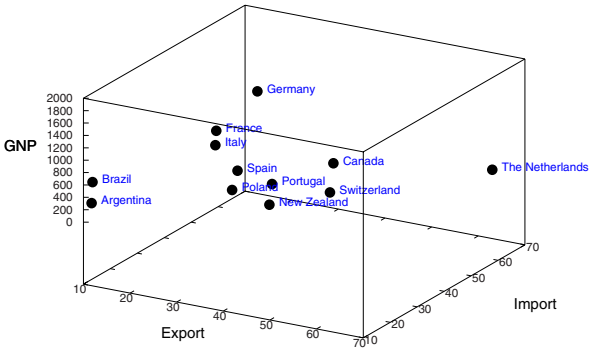

Fig. 2.9 A three-dimensional scatterplot.

mation. In the picture, the median line is indicated, leading us to immediately notice how the nations of Portugal, Poland, Spain, Brazil, and Argentina had a much higher import-based economy in 2000.

The term **trivariate** representation is used when three dependent attributes vary with respect to one or more independent attributes. This case becomes complicated, since the two spatial dimensions we have used until now to map the dependent attributes are not sufficient. Because we live in a three-dimensional world, we are well used to observing objects represented in three-dimensional spaces. Therefore, a very natural thing to do is to extend a scatterplot to include a third dimension, which we represent through perspective. Figure 2.9 provides us with an example of a scatterplot in which we have grouped the GNP and import and export values of the previously featured nations.

Representations like Fig. 2.9 typically present two types of problems. First of all, in three-dimensional (3D) representations, *occlusion* problems can occur, meaning there is a possibility that some graphical elements are "hidden" behind the elements in front. Second, it is difficult to identify the position of the graphical elements with respect to the axes. For instance, in Fig. 2.9, it is very difficult to understand whether France or Canada has a higher import value.

There are various strategies for solving these types of problems, which are intrinsic to all 3D representations, such as rotating the image to reveal the occluded objects or identify the values associated with each axis. Another solution could be using a two-dimensional scatterplot and mapping an attribute using other graphical properties, like color or the dimension of the graphical elements. For example, in Fig. 2.10 the third attribute is mapped to the area of the graphical elements or to a color scale. In this way, the third dimension is sacrificed, but, on the other hand, we have a clearer and more precise visual representation. Nevertheless, the occlusion problem still remains.

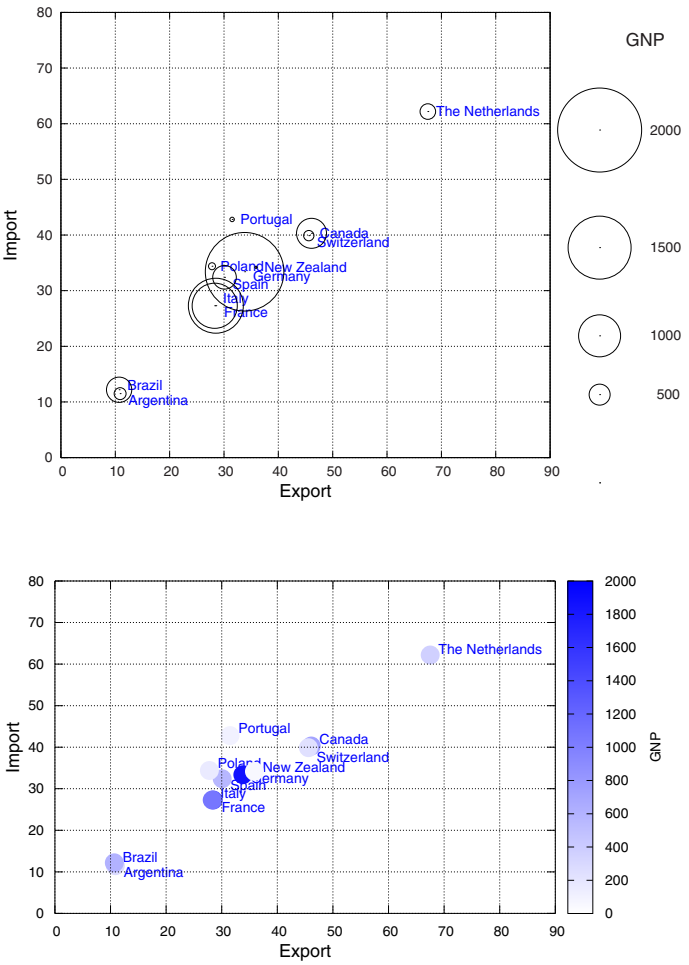

Fig. 2.10 A two-dimensional scatterplot with GNP mapped by the area of the circle (top image) and by a color scale (bottom image).

2.4 2D vs. 3D

In the previous section we looked at a situation in which a collection of trivariate data is represented by two-and three-dimensional spatial views. We have also demonstrated how 2D representations are clearer and more precise than 3D, due to some intrinsic problems that afflict the 3D views. However, we are used to representing the images on a two-dimensional screen, so that the third dimension is simulated by using perspective. Furthermore, some empirical studies have shown that 3D representations increase *cognitive load*, or the user's mental effort to cor-

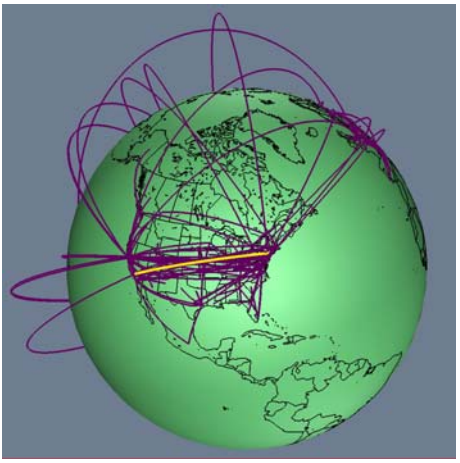

Fig. 2.11 Graphic representation of the topology of the Internet network MBone on a terrestrial scale. Image reproduced with the permission of Tamara Munzner and © 1996 IEEE.

rectly interpret the data represented. Does this mean that 3D representations are to be avoided in any case? Not always.

As a general rule, we can say that 2D representations should be preferred over 3D. 3D representations are to be used in limited and particular cases. One case in which 3D representation works wonderfully is, for example, when there is a need to represent an object in movement, or when the data to be represented have a three-dimensional spatial component, like the Earth or the structure of a molecule. Figure 2.11 visually represents the topology of the data transmission network used by the Internet in multicast modality (MBone, *multicast backbone*) [45]. The image is represented through VRML technology, which allows a user to visualize and explore the globe interactively to understand the structure of this network's topology in every part of the planet. Certainly, this is a very effective representation of a table of Internet IP addresses and numbers. Thanks to the usage of VRML technology, the interactivity allows the best use of this 3D representation.

2.5 Conclusion

In this chapter, we have presented a reference model that describes the procedure that generates interactive visual representations from data, by means of a pipeline of three stages: preprocessing and data transformations, visual mapping, and view creation. Each of these was analyzed in detail with a practical example. We saw how some operations, such as the choice of graphical elements and properties to be used in the visual mapping stage, are crucial and depend on the experience and ability of the system's designer. We suggested a procedure to follow when designing

visual applications. Finally, we introduced some examples of visual representations for univariate, bivariate, and trivariate data.