Journal of Visual Languages and Computing (2002) **13**, 601–622
doi:10.1006/S1045-926X(02)00026-5 available online at http://www.idealibrary.com on IDEAL®

The Role of Visual Perception in Data Visualization

MEHDI DASTANI

Institute of Information and Computing Sciences, Universiteit Utrecht, Padualaan 14, De Uithof 3584CH Utrecht, The Netherlands, e-mail: mehdi@cs.uu.nl

Received 12 July 2001; accepted 27 March 2002

This paper presents a perceptually motivated formal framework for effective visualization of relational data. In this framework, the intended structure of data and the perceptual structure of visualizations are formally and uniformly defined in terms of relations that are induced on data and visual elements by data and visual attributes, respectively. Visual attributes are analyzed and classified from the perceptual point of view and in terms of perceptual relations that they induce on visual elements. The presented framework satisfies a necessary condition for effective data visualizations. This condition is formulated in terms of a structure preserving map between the intended structure of data and the perceptual structure of visualization. © 2002 Published by Elsevier Science Ltd.

1. Introduction

THEORIES OF EFFECTIVE DATA VISUALIZATION aim to analyze and explain the relation between data and their effective visualizations [1–6]. This relation is essentially a denotation relation in the sense that visual elements and relations denote data elements and relations, respectively. For example, the number of cars produced by different factories can be visualized by a bar-chart. Each bar represents a car factory and its length represents the number of produced cars by that factory. The perceivable length relations between bars denote relations between the numbers of cars produced by different factories.

The effectiveness of a visualization depends on many conditions such as conventions, subjective user preferences, cultural settings, and the structural correspondence with its denoting data [4, 5, 7–9]. For example, the effectiveness of a temperature map in which temperature values are visualized by color hue values increases when conventional color hue values are used for high and low temperatures, i.e. red for high temperatures and blue for low temperatures. Although the satisfaction of some conditions enhance the effectiveness of visualizations, the satisfaction of other conditions are necessary for effective data visualization.

In general, we distinguish conditions for effective data visualization into necessary and sufficient conditions. This distinction is closely related to the well-known distinction between expressiveness and effectiveness [2, 3]. There are at least two reasons for not using the expressiveness/effectiveness distinction. First, the expressiveness criterion is defined with respect to a graphical language which we want to abstract from. The aim of this paper is to propose a general framework for effective data visualization, which is independent of graphical languages and thus not limited to their expressions. Second, the expressiveness criterion is claimed to be satisfied if and only if data are represented in the visual structure and the effectiveness criterion is claimed to be satisfied if the mechanism of the human visual system is taken into account [2]. We believe that the expressiveness criterion makes only sense if visual structures are perceivable by human viewers. This implies that both expressiveness and effectiveness criteria depend on the human visual system and that the expressiveness/effectiveness distinction is not a distinction between perceptually depending and perceptually not depending conditions.

In the context of effective data visualization, the structural correspondence condition is a necessary condition without which the effectiveness of visualizations cannot be guaranteed. This condition should be defined between the structure of data and the perceptual structure of its visualization. The emphasize on perceptual structure in this formulation is based on the assumption that the effectiveness of visualizations depend on the effective use of the capabilities of the human visual system to perceive visual structures. We argue that a theory of effective data visualization should therefore be based on theories of human visual system that describe human capabilities of perceiving visual relations. For example, the human capability to perceive relation between color hue values is limited to the identity relation. For this reason, a visualization in which the age of individuals is visualized by color hue values is not effective if the intended data relations are quantitative relations between ages.

In this paper, we focus on the structural correspondence condition for effective data visualization and propose a formal framework for effective data visualization, which satisfies the structural correspondence condition. This paper is organized as follows. In Section 2, we propose a process model for effective data visualization and identify the steps through which the structural correspondence between data and visualizations should be established. For this model, we discuss the class of input data structures, the projection step from data structures to perceptual structures, the layout of perceptual structures, and finally the necessary condition for effective data visualization. In Section 3, some general types of data attributes that are important for data visualization are introduced and a classification of attribute-based data structures is proposed. Similarly, various visual attributes are analyzed from the perceptual point of view and a classification of perceptual structures is proposed. In Section 4, we build on formal definitions of attributes and specify data and perceptual structures in a formal way. The structural correspondence condition for effective data visualization is then defined as a structure-preserving relation between data and perceptual structures. Finally, in Section 5, we conclude the paper and discuss future research directions.

2. A Process Model for Effective Data Visualization

The process of data visualization is considered here as the inverse of the interpretation process, i.e. the visualization process generates for an input data a visualization the interpretation of which results the original data. Since the effectiveness of a visualization can be measured in terms of the easiness and directness of acquiring its intended interpretation and because the interpretation of a visualization depends on human visual

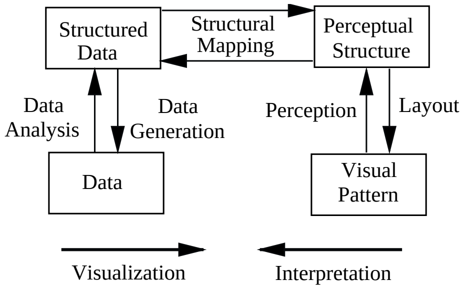

Figure 1. A process model for effective data visualization

perception, an effective visualization should strongly benefit from the capabilities of the human visual system. We formulate, therefore, the effectiveness of visualization as follows: *a visualization presents the input data effectively if the intended structure of the data and the perceptual structure of the visualization coincide*. Based on this view, a process model for effective data visualization [10] is schematically illustrated in Figure 1.

This process starts with an input data which is assumed to consist of data elements. The first step in data visualization is to determine the structure of data, i.e. how data elements are related to each other. This is not a trivial decision process since for example a set of integers may be related to each other nominally (i.e. integers are used as identifiers), ordinally (i.e. integers are used as ordinals), or quantitatively (integers are used as quantities). This step in the process of data visualization transforms the input data to what is called structured data. The second step in data visualization is to determine visual elements that represent data elements in such a way that the perceptual structure of the decided visual elements represents the structure of the data. We assume that the result of this step is not necessarily a drawable visualization, but an abstract perceptual structure. Therefore, the third and the last step in data visualization is the layout process which transforms the abstract perceptual structure to a drawable visualization.

On the other hand, the interpretation process starts with a visualization, possibly together with a legend. Since the input of the interpretation process is a visualization and because visualizations must be perceived before they can be understood as denoting data, the first interpretation step is perception. The process of perception determines perceivable relations among visual elements on the basis of their visual attribute values. As visual attribute values are assigned to visual elements by the layout process, perception is considered as the inverse of the layout process. Subsequently, visual elements are mapped to data elements. This mapping is possibly supported by a legend that determines the correspondence between visual and data attributes and their values. In the case of effective data visualization, the relations between data elements are structurally identical to the perceivable relations between their representing visual elements. Finally, given the structured data, the table of attribute values can be trivially generated. In the rest of this section, we elaborate on different steps of this process model.

2.1. Structuring the Input Data

In this paper, the input data for the visualization process is assumed to be nested relational data. This class of data can be represented as $m \times n$ data tables consisting of

Table 1. A data table describing the cooperation degrees between car companies

	company 1	*cooperation-degree*	*company 2*
d_1	d'_1	good	d'_2
d_2	d'_1	best	d'_3
d_3	d'_2	good	d'_1
d_4	d'_3	best	d'_1
d_5	d'_3	normal	d'_4
d_6	d'_4	normal	d'_3

m columns and n rows. A column A_i $(i = 1, \ldots, m)$ consists of values of one data attribute and a row B_j $(j = 1, \ldots, n)$ consists of m values each belongs to one of attributes $A_1, \ldots, A_m$. A row of a table is called a data entry[a]. The values of attributes $A_1, \ldots, A_m$ may themselves be rows of another $k \times l$ table that consists of $A'_1, \ldots, A'_k$ columns and $B'_1, \ldots, B'_l$ rows. In such a case, attributes $A_1, \ldots, A_m$ are called aggregated attributes. An attribute value, which is a single value, is called an atomic value.

For example, consider Table 1, which describes the cooperation degree between pairs of car companies. In this table, *company1* and *company2* are aggregated attributes since their values are rows from Table 2. This table describes the number of sold cars having a certain type and being produced in a certain country. In this data table, *car-type, sold-numbers,* and *country* are attributes that have atomic values. These two data tables can be combined into one single data table by means of aggregated attributes. Table 3 is obtained by combining Tables 1 and 2.

According to the proposed data visualization process model, the first step is to determine the intended structure of the input data. In general, we assume that an input data can be structured in many different ways and that the intended structure of the input data is determined by human users. We also assume that human users have access to different transformations that provide different structures of the input data [2].

Two types of data structures are distinguished. The structures of the first type are constituted by relations that are intended to exist among values of individual data attributes. These relations characterize the type of the corresponding attributes. For instance, values of the *sold-number* attribute are from the domain of integers and they may be intended to be related to each other by some arithmetical relations. These arithmetical relations characterize the type of the *sold-number* attribute. A data attribute together with its type is called a typed data attribute and a table that is defined on typed data attributes is called a typed data table. In a typed data table, relations that are defined on values of individual attributes are induced on data entries in the sense that two data entries are related in the same way as their attribute values are related. Relations on data entries may be induced by different data attributes. For instance, data entries in Table 2 may be considered as equivalent

[a]Many researchers in this field, such as Bertin [1] and Card *et al.* [2], have used tables as the format of the input data for the visualization process. However, they have used different terminology than used in this paper. For example, Bertin uses the term data *characteristics* for data *attributes* and data *objects* for data *entries*, while Card *et al.* use data *variables* for data *attributes* and data *cases* for data *entries*. The terminology used in this paper is motivated by the relational database terminology.

Table 2. A data table describing car companies in terms of car types, the number of sold cars, and the countries they produce cars

	car-type	*sold-number*	*country*
d'_1	vw	40 000	Germany
d'_2	toyota	50 000	Japan
d'_3	opel	30 000	Germany
d'_4	mazda	20 000	Japan

Table 3. Tables 1 and 2 are combined

company 1			*cooperation-degree*	*company 2*		
car-type	*sold-number*	*country*		*car-type*	*sold-number*	*country*
vw	40 000	Germany	good	toyota	50 000	Japan
vw	40 000	Germany	best	opel	30 000	Germany
Toyota	50 000	Japan	good	vw	40 000	Germany
Opel	30 000	Germany	best	vw	40 000	Germany
Opel	30 000	Germany	normal	mazda	20 000	Japan
Mazda	20 000	Japan	normal	opel	30 000	Germany

or non-equivalent according to the *car-type* attribute, while they are quantitatively related to each other according to the *sold-number* attribute.

The structures of the second type are constituted by binary relations that are defined on values of two different aggregated attributes. The values of these aggregated attributes should be the entries of one and the same table. In fact, the characterizing relation for each single aggregated attribute is the identity relation, but since the values of two aggregated attributes are from one and the same domain (i.e. the entries from one and the same table), two aggregated attributes together define a binary relation between domain elements. These binary relations are implicit in the data table and do not characterize any single attribute. For instance, the binary relation defined on the values of the *company1* and *company2* attributes in Table 1 can be represented as: $\{(d'_1, d'_2), (d'_1, d'_3), (d'_2, d'_1), (d'_3, d'_1), (d'_3, d'_4), (d'_4, d'_3)\}$. This binary relation relates data entries from Table 2. Binary relations can be characterized by their structural properties such as functional (injective, surjective, bijective), transitive, symmetric, and reflexive properties. In Table 1 the values of the *company1* and *company2* attributes define such a binary relation which is non-functional, symmetric, non-reflexive, and non-transitive. Notice that this type of structures can be extended by allowing n-ary relations between aggregated attribute values. In such a case, data tables consist of n aggregated attributes.

2.2. Projecting Structured Data to Perceptual Structure

The second step in visualizing data is the projection of the structured data into a perceptual structure. A perceptual structure is defined as a typed visual table which is basically

Table 4. A visual table for the data described in Table 1

	connection 1	*Size*	*connection 2*
v_1	v'_1	$size_2$	v'_2
v_2	v'_1	$size_3$	v'_3
v_3	v'_2	$size_2$	v'_1
v_4	v'_3	$size_3$	v'_1
v_5	v'_3	$size_1$	v'_4
v_6	v'_4	$size_1$	v'_3

a table constituted by typed visual attributes. The type of a visual attribute is determined by the relations that can be perceived among values of that attribute. As we will explain in the next section, the perceivable relation for each visual attribute is determined beforehand based on theoretical and empirical studies of human visual perception. The rows of a typed visual table will then be called visual entries.

The projection of the structured data into a perceptual structure consists of a bijective function from typed data attributes to typed visual attributes and another bijective function from rows of the data table (i.e. data entries) to rows of the visual table (i.e. visual entries). There are different functions that map typed data attributes to typed visual attributes. These different functions specify alternative visualizations of data. We assume that the second function does not map data attribute values to visual attribute values, but to variables that stand for visual attribute values. These variables will be instantiated with actual values by the layout process. As we will explain in Section 4, for effective data visualization the projection should be a structure-preserving projection. This can be guaranteed by ensuring that the first function maps each typed data attribute to a visual attribute with identical type, and that the second function maps data attribute values to visual variables in such a way that whenever two data attribute values are in a certain relation (characterizing the type of that data attribute), their corresponding visual variables are in the corresponding perceptual relation as well.

This projection specifies abstract visual entries that are related to each other by perceptual relations induced by the typed visual attributes. For example, consider Table 1 once again. The first function maps the *company1* data attribute to the *connection1* visual attribute, *company2* to *connection2*, *cooperation-degree* to *size*, *car-type* to *xpos*, *sold-number* to *ypos*, and *country* to *color-hue*. The second function maps data entries to abstract visual entries such that the structure preserving condition is satisfied. For instance, consider the data attribute *country* which is characterized by the identity relation. Whenever two values of the *country* attribute are identical, their corresponding *color-hue* variables are identical as well. This projection results typed visual tables illustrated in Tables 4 and 5.

2.3. The Layout of Visualization

The variables in the typed visual table should be instantiated with visual attribute values to generate a drawable visualization of data. However, the typed visual table consists of only those visual attributes to which a data attribute is projected. Therefore, it may be the case that some visual attributes, the values of which are necessary to generate a drawable

Table 5. A visual table for the data described in Table 2

	xpos	*ypos*	*color-hue*
v'_1	seg_1	pos_4	hue_1
v'_2	seg_2	pos_5	hue_2
v'_3	seg_3	pos_3	hue_1
v'_4	seg_4	pos_2	hue_2

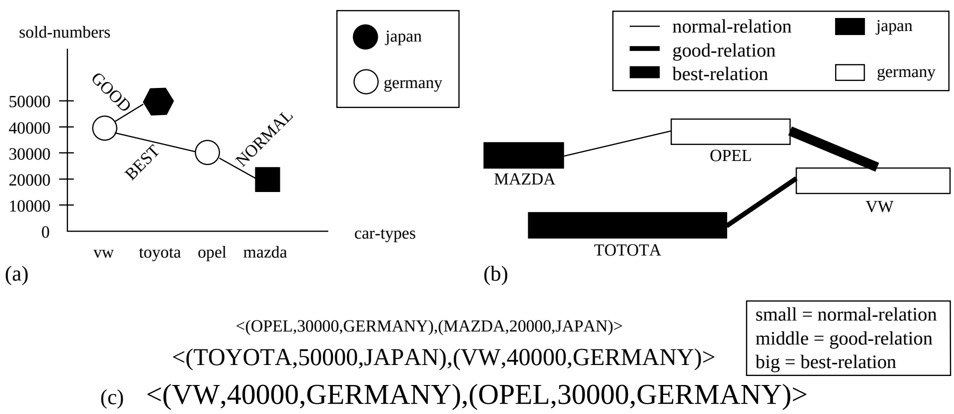

Figure 2. Visualizations of the data described by Table 1

visualization, are not used by the projection. For instance, if the position attribute is not used by a projection, then visual entries should receive position values in order to become drawable. These unused visual attributes will be called undecided visual attributes in contrast to the decided attributes to which data attributes are projected.

Consider the visualizations that are illustrated in Figure 2. These are three possible visualizations for the data represented in Tables 1 and 2. In visualization 2-A, shape and size are undecided visual attributes. Moreover, in visualization 2-B position and shape, and in visualization 2-C position, are undecided visual attributes. In visualization 2-C, the *label* attribute is decided for the *car-type, sold-number,* and *country* attributes. This example shows that more than one data attribute can be projected to a visual attribute. Of course, this does not hold for all visual attributes. For example, only one data attribute can be projected to the *color-hue* attribute since otherwise the effectiveness of visualization cannot be guaranteed. We assume that the projection function is defined based on this general visualization knowledge. Visualization 2-C shows also that a drawable visualization does not need to have values for all visual attributes. For example, in this visualization visual entries do not need shape values to become drawable. We assume that the combination of decided visual attributes determines which undecided visual attribute values should be specified for visual entries to become drawable.

The generation of values for decided and undecided visual attributes is a constraint satisfaction process. The constraints on the values of the decided visual attributes are imposed by the perceptual structure determined at the previous visualization step. Although the perceptual structure does not impose any constraints on the values of undecided visual attributes, these values cannot be generated arbitrarily since otherwise unwanted visual implicature [11] may occur. In the next section, we will discuss this issue in more details. The determination of values for decided and undecided visual attributes will be called the layout process.

It should be noted that the textual information inserted in these visualizations may either be the values of the *label* attribute or it may be a part of the interpretation function. In order to distinguish these two kinds of textual information, we use capital letters only to indicate that a textual information is a value of the visual *label* attribute.

2.4. Effective Data Visualization

A data table can be visualized in many different ways by deciding different visual attributes for each data attribute and by specifying different visual entries for each data entry. In Figure 3, two different visualizations for Tables 1 and 2 are illustrated. In visualization 3-A, the *xpos, ypos,* and *color-hue* attributes visualize the *car-type, sold-number,* and *country* attributes, respectively. Moreover, the links between visual elements, defined by the *connection1* and *connection2* attributes, visualize the *company1* and *company2* attributes, respectively. Finally, the *thickness* attribute of the links visualizes the *cooperation-degree* attribute. In visualization 3-B, the *shape, color-hue,* and *label* attributes visualize the *sold-number, country,* and *car-type* attributes, respectively. Like visualization 3-A, the links and their thickness visualize the companies and their degrees of cooperations, respectively. In visualizations 3-A and 3-B the inserted numbers are parts of the interpretation function (i.e. legend).

Although these two visualizations represent the same data, visualization 3-B is not an effective visualization if the quantitative relations between the numbers of sold cars are the intended data relations. In this visualization, the perceivable relation induced by the shape attribute is the identity relation which implies that one can only perceive that the numbers of sold cars are equal or not. Of course, one may deduce the intended quantitative data relations indirectly by means of the interpretation information that are inserted into visualization 3-B. For example, the interpretation values 20 000 and 40 000 implies that the second is twice the first. However, this inference is based on the knowledge of the perceiver about the structure of real numbers rather than being based on perceivable relations. The effectiveness of visualization is the ability to perceive data relations directly by structurally similar perceivable relations.

In order to illustrate the direct perception of relations, consider visualizations 4-A and 4-B (Figure 4). These visualizations are the same as visualizations 3-A and 3-B,

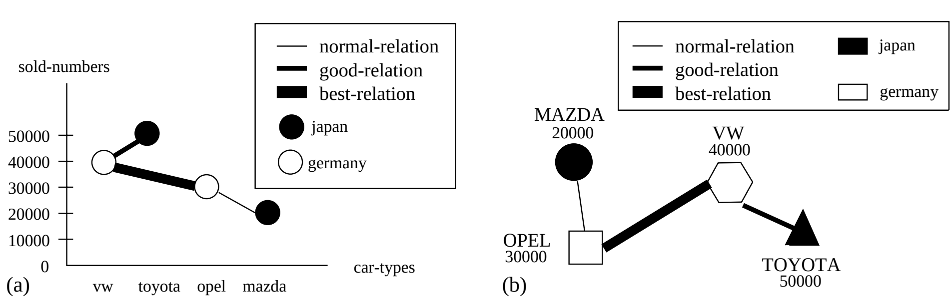

Figure 3. Alternative visualizations of the data described in Table 1

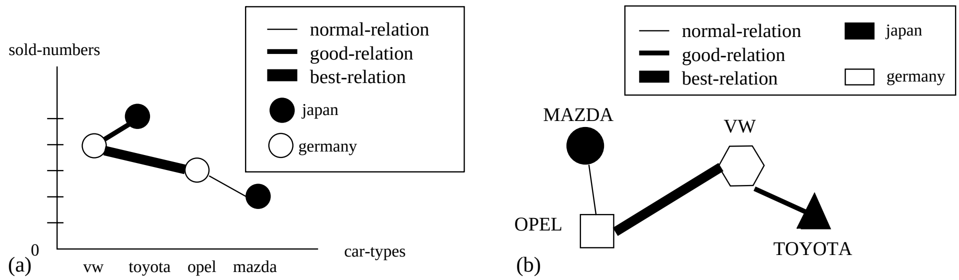

Figure 4. Visualizations without any interpretation values

respectively, except that in these visualizations the interpretation information are not presented. Knowing that the *shape* attribute visualizes the *sold-number* attribute in visualization 4-B, it is impossible to perceive any quantitative relation between visual elements and therefore it is impossible to infer any quantitative data relation. In contrast, knowing that the *ypos* attribute visualizes the *sold-number* attribute in visualization 4-A, it is easy to perceive quantitative relations between visual elements and therefore easy to infer quantitative data relations. In the rest of this paper, whenever we refer to the perceptual structure of visualizations we mean the structure of visualizations without using any interpretation information.

The above examples show that the choice of visual attributes for data attributes may influence the effectiveness of visualizations. However, the specifications of visual entries by the layout process may influence the effectiveness of visualizations as well. We argued that the layout process should specify values for both decided and undecided visual attributes. In the context of effective data visualization, the generation of values for the decided visual attributes is constrained by the perceivable relations. For example, in visualization 3-A where the *color-hue* attribute is decided for the *country* attribute, the layout process has generated identical *color-hue* values whenever their corresponding data attribute values are identical.

The generation of values for the undecided visual attributes may also influence the effectiveness of visualizations. For example, consider again visualization 2-A. Although this visualization represents the same information as visualization 3-A, it is not an effective visualization. The reason is that a human viewer perceives differences between shapes of visual elements and may infer that these differences visualize differences in the presented data. In order to avoid this type of unwanted visual implicatures, the values of undecided visual attributes should be specified in such a way to induce an identity relation on the involved visual elements. In this way, visual elements will be perceived as being identical to each other with respect to undecided visual attributes. This issue will be further explained in Section 4.3

3. An Attribute-based Classification of Structures

An important step in data visualization is the projection of data structures (typed data table) to perceptual structures (types visual table). In this section, a unifying approach is proposed to analyze different classes of data and perceptual structures. This approach is based on a formalization of attributes in terms of their characterizing relations. The advantage of this approach is that data and perceptual structures are both formalized in a uniform way such that the projection between them can be formulated in a formal and intuitive way. Based on this formalization a classification of data and perceptual structures is proposed.

In order to define attributes in a formal way, we use some notions from the *measurement theory* [12, 13]. The theory of measurement provides both a classification of attributes as well as their mathematical definitions by introducing different types of *measurement scales*. In fact, a measurement scale is a map from a structured set of elements to a structured set of values like the set of real numbers, the set of integers, or a set of strings. In order to model aggregated attributes, we also allow an attribute to map a structured set of elements to another structured set of elements. In the measurement theory, relational systems are used to represent structured sets.

Definition 1. A relational system is a pair $\langle A; R_1, \ldots, R_n \rangle$ where A is a set of elements, and $R_1, \ldots, R_n$ are relations defined on A.

We consider an attribute as a measurement scale, which map a structured set into another structured set.

Definition 2. An attribute is a homomorphism $\mathscr{H}$ from a relational system $\langle A; R_1, \ldots, R_n \rangle$ into a relational system $\langle B; S_1, \ldots, S_n \rangle$. The set A is the set of elements and the set B is either a set of elements or a set of attribute values such as the set of real numbers, the set of integers, or a set of strings.

In the context of data visualization, A can be either a set of visual elements or a set of data elements, $R_1, \ldots, R_n$ can be either perceptual relations or data relations, B is either a set of elements or a set of values, and $S_1, \ldots, S_n$ are characterizing relations defined on B. These characterizing relations are abstract mathematical relations such as $=$, $\leq$ and $+$. The homomorphism guarantees that the relations that an attribute induces on elements have identical structural properties as its characterizing relations. In the next subsections, we introduce data attributes and their corresponding data structures, followed by visual attributes and their corresponding perceptual structures.

3.1. Data Attributes

Below are five different types of data attributes that are relevant for data visualization.

Nominal attributes: A nominal attribute is a homomorphic map $\mathscr{H}$ from a relational system $\langle A; \approx \rangle$ into a relational system $\langle B; = \rangle$. Note that the homomorphic map assigns values such as real numbers or strings to elements of A.

Ordinal attributes: An ordinal attribute is a homomorphic map $\mathscr{H}$ from a relational system $\langle A; \preceq \rangle$ into the relational system $\langle B; \leq \rangle$. The homomorphic map matches the ordinal relation among attribute values with the ordinal relation among elements of A.

Interval attributes: An interval attribute is a homomorphic map $\mathscr{H}$ from a relational system $\langle A; \preceq, \circ \rangle$ into the relational system $\langle \mathscr{R}^k; \leq, \ominus \rangle$, where $\ominus$ is a quaternary metrical relation defined on real numbers. In the context of data visualization, we may use the quaternary metrical relation $\ominus$ defined as: $ab \ominus cd \Leftrightarrow |a-b| \leq |c-d|$. This attribute is often called the absolute interval attribute because we are interested in the absolute difference between pairs of elements.

Ratio attributes: A ratio attribute is a homomorphic map $\mathscr{H}$ from a relational system $\langle A;\ \preceq, \circ, Null \rangle$ into the relational system $\langle (\mathscr{R}_{\geq 0})^k; \leq, \oslash, 0 \rangle$, where $\oslash$ is a binary metric operator defined on real numbers and 0 is a zero-place operator identifying the zero number. Note that the zero number has the following property: $\forall e \in (\mathscr{R}_{>0})^k\ 0 \oslash e = 0$. This zero-place operator identifies the origin element. In the context of data visualization, we may use the binary metrical operator $\oslash$ defined as: $a \oslash b \Leftrightarrow a/b$ where $a \in (\mathscr{R}_{\geq 0})^k$ and $b \in (\mathscr{R}_{>0})^k$, i.e. b is not the absolute origin element 0 ($b \neq 0$).

Aggregated attributes: An aggregated attribute is a homomorphic map $\mathscr{H}$ from a relational system $\langle A; \approx \rangle$ into a relational system $\langle B; = \rangle$, where A and B are two distinct sets of data elements. This is in contrast with other attributes since the set B is the set of data elements instead of atomic values.

Although only interval and ratio attributes are introduced here, there may be many quantitative attributes, each of which uses different sets of arithmetical relations and properties of real numbers. This approach is general enough to define alternative data attribute types.

3.2. A Classification of Data Structures

The proposed five data attributes result in several classes of data structures. First, we distinguish between separable data structures that are constituted by characterizing relations of separate data attributes, and aggregated data structures that are based on two or more aggregated attributes. For example, Table 2 has a separable structure while Table 1 has an aggregated structure. Separable structures can be subdivided into qualitative and quantitative structures. Subsequently, qualitative data structures can be subdivided into nominal and ordinal structures and quantitative structures into interval and ratio.

Aggregated data structures can be subdivided into pure and mixed structures depending on the existence of non-aggregated attributes. The pure aggregated structures are constituted only by aggregated attributes, while mixed aggregated structures are also constituted by relations that characterize additional non-aggregated attributes. For example, Table 1 has a mixed aggregated structure since it is based on two aggregated the *company1* and *company2* attributes and the additional non-aggregated *cooperation-degree* attribute. Without this non-aggregated attribute, Table 1 would have a pure aggregated data structure. These data structures can be classified hierarchically as illustrated in Figure 5. The formal definitions of these data structures are given in Section 4.1.

3.3. Visual Attributes

A visual attribute maps a relational system $\langle A; R_1, \ldots, R_n \rangle$, where A is a set of visual elements, into a relational system $\langle B; S_1, \ldots, S_n \rangle$, where B is either a set of visual elements or a set of atomic visual attribute values. Since we are interested in perceptual characterizations of visual attributes, relations $S_1, \ldots, S_n$ for a visual attribute should characterize the perceivable relations that are induced on visual elements by that visual attribute. For example, the human visual system can identify the equality of shape values

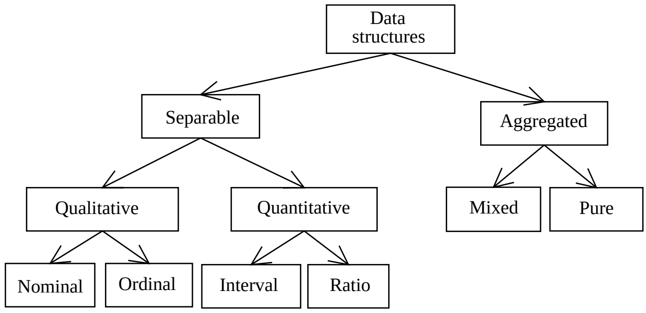

Figure 5. A partial classification of data structures

and perceives that a red square and a green square have the same shapes, while it cannot identify an ordering among a red circle and a red square. In the rest of this section, we study a number of visual attributes from the perceptual point of view.

3.3.1. Non-spatial Attributes

The non-spatial visual attributes are studied by Bertin [1][b]. These attributes include hue, saturation, brightness, size, shape, label, and texture.

Color-hue attribute: The color hue attribute can be defined as a homomorphic map from a relational system, consisting of a set of visual elements and an equivalence relation, into a relational system consisting of a set of color hue values and the equality relation. This definition of the color hue attribute assumes that the human visual system can only identify the equality of color hue values, but cannot order them[c].

Color-saturation attribute: The color saturation attribute is a homomorphic map from a relational system, consisting of a set of visual elements and an ordered equivalence relation, into a relational system consisting of the set of color saturation values and the ordering relation $\leq$. It is then assumed that both the identity and the ordering relations among color saturation values can be perceived.

Color-brightness attribute: The color brightness attribute is a homomorphic map from a relational system, consisting of a set of visual elements and an ordered equivalence relation, into a relational system consisting of the set of color brightness values and the ordering relation $\leq$. It is then assumed that both the identity and the ordering relations among color brightness values can be perceived.

Shape attribute: The shape attribute can be defined in a similar way as the color hue attribute. The range of the shape attribute is then a pair consisting of a set of shape values and

[b]In his book, *Graphics and Graphic Information Processing*, he uses the term 'visual variables' rather than 'visual attributes'.

[c]Of course, this is a narrow characterization of the color hue attribute. In fact, color hue values can in some cases be perceived as being ordered. For example, the rainbow colors (a certain sequence of the color hue values) are perceived as having an ordering structure. This ordering structure can be characterized as a ring structure that is constituted by an abstract mathematical relation like modulo relation. The characterization of the color hue attribute becomes even more difficult when we consider the fact that people (especially visual artists) can know that the purple hue value is between blue and red, orange between red and yellow, but have no awareness of what may be between orange and blue. Nevertheless, we assume that the properties of the color hue attribute can be specified by abstract mathematical relations, so that we can define this attribute as a homomorphic map between relational systems.

an equality relation defined on it. It is then assumed that one can perceive the equality of shapes but cannot perceive how one shape is related to another one[d].

Size attribute: The range of the size attribute can be defined as a pair consisting of a set of possible size values and an order relation defined on it. However, the equality of size values can be perceived by the human visual system only in special cases where the set of size values is a small finite set[e].

Label and Texture attributes: The label (texture) attribute is defined as homomorphic map for which the range consists of a set of label (texture) values. The relation for these relational systems is the equality relation.

3.3.2. Spatial Attributes

A rather different type of visual attributes concerns various uses of the extension of the space [14]. These attributes are called spatial attributes. The spatial attributes of multidimensional spaces interact. This implies that spatial attributes for multidimensional spaces may employ complex properties which cannot be defined as properties of individual one-dimensional spaces. Here, we consider only those properties of multidimensional spaces that can be defined in terms of properties of one-dimensional spaces. Therefore, we study different spatial attributes for one-dimensional space, henceforth 1D-space, and consider only those spatial attributes of the 2D-space or the 3D-space that can be defined by two or three spatial attributes of the 1D-space.

Segmentation attribute: The use of 1D-space by dividing it into subspaces is called segmentation. In this way, the total space is divided into a finite number of segments. The segmentation attribute can then be defined as a homomorphic map from a relational system, consisting of a set of visual elements and the equivalence relation, into a relational system consisting of a set of real numbers or names, each of which identifies a certain segment, and the equality relation defined on it. It is then assumed that the human visual system can perceive the space as consisting of a number of perceptually distinguishable segments.

Ordered segmentation attribute: One can also use the ordinal property of 1D-space and consider it as an ordered set of subspaces. This consideration results the ordered segmentation attribute, which can be defined as a homomorphic map from a relational system, consisting of a set of visual elements and an ordered equivalence relation, into a relational system consisting of a set of real numbers or names, each of which identifies a certain segment, and an order relation defined on this set.

Relative space attribute: The space can also be viewed in terms of its quantitative properties. The quantitative properties of the 1D-space can be used in two different ways: the first use of the quantitative properties of the 1D-space is by considering the space as a relative 1D-space consisting of positions in which no origin is defined. In this case, the space between two positions (interval) is a quantity that can be perceived and meaningfully used, e.g. two intervals can be compared to each other. The resulting visual attribute will be called relative space attribute. This attribute can be defined as a homomorphic

[d] If we consider only a specific set of shapes, we may perceive an ordering relation. For instance, in the case of polygons, shapes can be ordered according to the number of edges (nodes) involved.

[e] One can also argue that the size attribute is a quantitative attribute such that one can perceive that one square is twice as big as a second square. In this case, the domain and the range of the size attribute should contain some metric relation that defines a quantitative structure on the set of visual elements and the set of attribute values, respectively.

map from a relational system consisting of a set of visual elements and a set of relations consisting of an equivalence relation, an ordering relation, and a metrical relation. The range of the map is a relational system consisting of the set of real numbers, each of which identifies a position, on which the equality relation, the ordering relation and the absolute difference relation between pairs of real numbers are defined.

Absolute space attribute: The second use of the quantitative properties of the space is by considering the space as an absolute space in which one position is (explicitly or implicitly) defined to be the origin of the space. In this case, each position in the space is a quantity such that the proportionality between positions can be perceived and meaningfully used. The resulting visual attribute will be called absolute space attribute. This visual attribute can be defined as a homomorphic map from a relational system consisting of a set of visual elements and a set of relations consisting of an equivalence relation, an ordering relation, and a metrical relation. The range of the map is a relational system consisting of the set of real numbers, each of which identifies a position, on which the equality relation, the ordering relation and the proportionality relation are defined on real numbers.

Combined attributes: The separate analysis of one-dimensional spatial attributes suggests that some spatial attributes for two- and three-dimensional spaces can be defined as consisting of two and three one-dimensional spatial attributes, respectively. Given the above four spatial attributes as they are defined for 1D-space, there are 16 possibilities for two-dimensional spatial attributes. But of course, some of those combinations may yield identical results. For example, the two-dimensional spatial attribute consisting of the absolute space attribute and the ordered segmentation attribute may be the same as the two-dimensional spatial attribute consisting of the ordered segmentation attribute and the absolute space attribute. The perceptual structure of most visualizations is induced by a combination of visual attributes. An example of the combination of visual attributes is two-dimensional bar-chart. In a bar-chart, one 1D-space (e.g. x-axis) is structured by dividing it into various segments while the second 1D-space (e.g. y-axis) is structured by its metrical quantities.

3.3.3. Topological Attributes

A special type of visual attributes concerns various uses of topological properties of the space. The perceptual structures that are constituted by perceivable topological relations are for example, inside, outside, overlap, and connectedness. Topological relations are often used in visualizations like trees, flow charts, network diagrams, path diagrams, subway maps, and Venn diagrams. In these visualizations, the perceivable topological relations are either explicitly represented by visual elements such as arrow-connections and shaded areas, or implicitly represented by placing visual elements in relations such as inclusion, exclusion, or overlap. For example, arrow connections in visualization 6-A (Figure 6) represent topological relations between circles explicitly, shaded areas in visualization 6-B represent topological relations between closed-curves explicitly, and overlaps between disks in visualization 6-C represent topological relations implicitly.

Topological relations are assumed to be induced on a set of visual elements V by means of two or more topological attributes. The values of these topological attributes are elements of V. If topological relations are explicitly represented by some visual elements, like arrow-connections or shaded areas, then topological attributes are considered

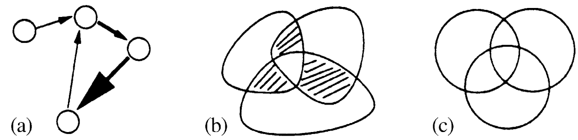

Figure 6. Examples of explicitly (A and B) and implicitly (C) represented topological relations

to be attributes of these visual elements. Visual elements that explicitly represent topological relations may also have other attributes such as size, color, and texture. Each topological attribute induces only the identity relation on visual elements that represent topological relations explicitly. For example, in visualization 6-A the arrows are defined on the basis of two topological attributes *Start-link* and *End-link* which assign two circles to each arrow and therefore relate two circles to each other. Each of these two attributes induces only the identity relation on arrows. The size values of these arrows induce an ordinal relation on them such that one perceives one arrow as thicker or thinner than another arrow. Similarly, in visualization 6-B the shaded areas are defined on the basis of two topological attributes *included-in-1st-element* and *included-in-2th-element*. In this way, a shaded area relates two closed-curves to each other.

In the case of implicitly represented topological relations, topological attributes generate a set of tuples, each of which relates visual elements to each other. These tuples do not represent any visual elements. For example, in visualization 6-C, two attributes *first-overlap-element* and *second-overlap-element* generate a set of binary tuples relating two disks implicitly.

An explicitly represented topological attribute is defined as a homomorphic map from a relational system, consisting of a set of visual elements (i.e. those that explicitly represent topological relations) and an equivalence relation, into a relational system consisting of a set of visual elements (i.e. those that are values of topological attributes) and the identity relation. Notice that explicitly represented topological relations, such as arrow-connections, can also be related to each other by means of visual attributes such as size attribute. Similarly, an implicit topological attribute is a homomorphic map from a relational system, consisting of a set of invisible elements and an equivalence relation, into a relational system consisting of a set of visual elements (e.g. Euler circles) and the identity relation. Although this treatment of implicit topological attributes is not intuitive since it assumes invisible elements, it has the advantage of modelling all visual attributes uniformly.

Topological relations may have various structural properties like being functional (injective, surjective, bijective), transitive, reflexive, and symmetric. Topological structures can be further subdivided into structures such as tree-structures, cyclic and acyclic graph-structures. A detailed classification of topological relations and structures is presented in [15].

3.4. A Classification of Perceptual Structures

The proposed visual attributes result in several classes of perceptual structures. First, we distinguish between spatial/non-spatial perceptual structures that are constituted by characterizing relations of spatial and non-spatial attributes, and topological structures that are based on two or more topological attributes. For example, Table 5 has a spatial/non-spatial structure while Table 4 has a topological structure. Spatial/non-spatial structures can be subdivided into continuous and discontinuous structures. Subsequently,

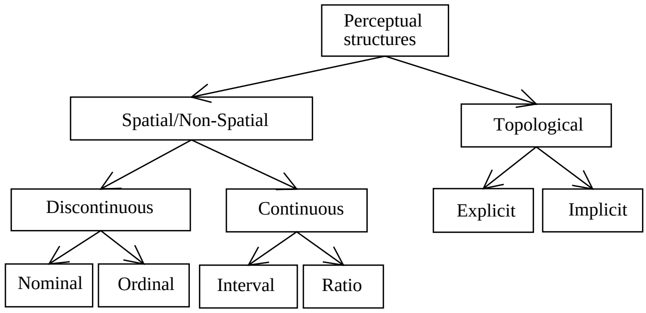

Figure 7. A partial classification of perceptual structures that are used in data visualization

discontinuous structures can be subdivided into nominal and ordinal structures and continuous structures into interval and ratio.

Topological structures can be subdivided into implicit and explicit structures depending on the existence of non-spatial attributes. Implicit topological structures are constituted only by topological attributes while explicit topological structures are also constituted by relations that characterize additional non-spatial attributes. For example, Table 4 has an explicit topological structure since it is based on the two topological *connection1* and *connection2* attributes and the additional non-spatial *size* attribute. Without this non-spatial attribute, Table 4 would have an implicit topological structure. This classification can be extended by including more specific types of perceptual structures. For example, explicit topological structures can be classified further into specific types such as graphs and trees.

The formal definitions of these perceptual structures are given in Section 4.1. These perceptual structures can be classified hierarchically as illustrated in Figure 7. Note the close similarity between this classification of perceptual structures and the classification of data structures illustrated in Figure 5.

This classification does not account for the interaction between visual attributes, which may influence the perceptual structure of visualizations [16, 17], Shepard-A. Of course, the interaction between visual attributes can be prevented by defining new attributes each of which is a constraint combination of the interacting attributes. For example, consider the *color-hue*, *color-saturation*, and *color-brightness* attributes, which are three interacting attributes. One may define a new *color-1* attribute for which the values are triples consisting of a hue, a saturation, and a brightness value. To prevent the interaction between these three attributes, the triples can be constraint in such a way that each triple consists of one and the same saturation value and one and the same brightness value. This implies that the use of one visual attribute in a visualization excludes the use of all interacting visual attributes in that visualization.

4. A Formal Framework For Data Visualization

In this section, we build on definitions from Section 3 and provide a formal description of the process model outlined in Section 2. In particular, we define two relational systems that are constructed based on data attributes and visual attributes, respectively. One relational system specifies the structure of the data and the second relational system specifies the perceptual structure of the visualization. The structural correspondence condition, which is a necessary condition for the effectiveness of visualizations, is then modeled by demanding a structure-preserving mapping between these two relational systems.

4.1. Structured Data and Perceptual Structures

As explained in Section 2.1, an $m \times n$ typed table consists of a set of attributes $A_1, \ldots, A_m$ and a set $D = \{d_1, \ldots, d_n\}$ of table entries. In this section, an $m \times n$ typed table is formally defined in terms of a combination of the mappings between relational systems where each mapping specifies one attribute of the table.

Definition 3. Let attributes $A_1, \ldots, A_m$ be defined as in Section 3, i.e.
$A_1 : \langle D; R_1^1, \ldots, R_k^1 \rangle \rightarrow < V^1; S_1^1, \ldots, S_k^1 >$
$\ldots$
$A_m : \langle D; R_1^m, \ldots, R_l^m \rangle \rightarrow < V^m; S_1^m, \ldots, S_l^m >$
These attributes share the set of elements D and can be combined to define a compound mapping as follows:
$\mathscr{M} : \langle D; R_1^1, \ldots, R_k^1, \ldots, R_1^m, \ldots, R_l^m \rangle \rightarrow \langle V^1 \times \ldots \times V^m; S_1^1, \ldots, S_k^1, \ldots, S_1^m, \ldots, S_l^m \rangle$ An $m \times n$ typed table based on attributes $A_1, \ldots, A_m$ is then the following relational system: $\langle D;\ R_1^1, \ldots, R_k^1, \ldots, R_1^m, \ldots, R_l^m \rangle$.

It is important to note that all attributes of one table share one and the same set of elements D. The value of attribute A_i for each element $d_j \in D$ is from the set V^i such that each element can be specified as an n-tuple of values from $V^1 \times \cdots \times V^m$. This definition can be used to define typed data tables as well as typed visual tables. For example, consider again Tables 1 and 2 describing the number of sold cars by different car companies in various countries and their cooperation degrees.

Let $\mathscr{D}' = \{d_1', \ldots, d_4'\}$ be the set of data entries of Table 2 consisting of values of the *car-type, sold-number,* and *country* attributes, and let $same - car$, $more - car$, $null$, and $quantity$ be relations defined on $\mathscr{D}'$. These relations are the identity relation between car types, the ordinal relation between amounts of cars, the zero amount, and the quantitative relation between amounts of cars, respectively. Let also $\mathscr{D} = \{d_1, \ldots, d_6\}$ be the set of data entries of Table 1 consisting of values of the *company1, cooperation-degree* and *company2* attributes, and let *better* and $same - company$ be relations defined on $\mathscr{D}$. These relations are the ordinal relation between cooperation degrees of car companies and the identity relation between car companies, respectively. Then, attributes of Tables 1 and 2 can be defined as the following homomorphic maps:

$$
\begin{array}{ll}
car\text{-}type : & \langle \mathscr{D}';\ same\text{-}car \rangle \rightarrow \langle \{vw, toyota, opel, mazda\}; = \rangle \\
sold\text{-}number : & \langle \mathscr{D}';\ more\text{-}car, quantity, null \rangle \rightarrow \langle \mathscr{R}^+; \leq, \oslash, 0 \rangle \\
 & where\ \forall a, b \in \mathscr{R}^+ if\ b \neq 0\ then\ a \oslash b = a/b. \\
country : & \langle \mathscr{D}';\ same - country \rangle \rightarrow \langle \{germany, japan\}; = \rangle \\
cooperation - degree : & \langle \mathscr{D};\ better \rangle \rightarrow \langle \{normal, good, best\}; \leq \rangle \\
company1 : & \langle \mathscr{D};\ same\text{-}company \rangle \rightarrow \langle \mathscr{D}'; = \rangle \\
company2 : & \langle \mathscr{D};\ same\text{-}company \rangle \rightarrow \langle \mathscr{D}'; = \rangle
\end{array}
$$

Note that *company*1 and *company*2 map elements from D to elements from D', which are themselves mapped to values by other data attributes. This is due to the involved aggregated data attributes. Tables 1 and 2 can now be represented as the following relational system:

$$\langle \mathscr{D} \cup \mathscr{D}';\ \textit{same-car}, \textit{more-car}, \textit{quantity}, \textit{null}, \textit{same-country}, \textit{better}, \textit{same-company} \rangle$$

In a similar way, Tables 4 and 5 can be represented as a relational system. Let $\mathscr{V}' = \{v'_1, \ldots, v'_4\}$ be the set of visual entries of Table 5 consisting of values of the *xpos, ypos,* and *color-hue* attributes, and let *same-seg, longer, null,* and *length* be relations defined on $\mathscr{V}'$. These relations are the identity relation between segments, the ordinal relation between positions, the zero position, and the quantitative relation between positions, respectively. Let also $\mathscr{V} = \{v_1, \ldots, v_6\}$ be the set of visual entries of Table 4 consisting of values of *connection1, size,* and *connection2* attributes, and let *thicker* and $same - element$ be relations defined on $\mathscr{V}$. These relations are the ordinal relation between the thickness of arrows and the identity relation between visual elements, respectively. Then, the attributes of Tables 4 and 5 can be defined as the following homomorphic maps:

$$\begin{array}{ll}
\textit{xpos}: & \langle \mathscr{V}';\ \textit{same-seg} \rangle \rightarrow \langle \{\textit{seq}1, \textit{seg}2, \textit{seg}3, \textit{seg}4\}; = \rangle \\
\textit{ypos}: & \langle \mathscr{V}';\ \textit{longer}, \textit{length}, \textit{null} \rangle \rightarrow \langle \mathscr{R}^+; \leq, \oslash, 0 \rangle \\
 & \textit{where}\ \forall a, b \in \mathscr{R}^+ \textit{if}\ \ b \neq 0\ \textit{then}\ a \oslash b = a/b. \\
\textit{color-hue}: & \langle \mathscr{V}';\ \textit{same-hue} \rangle \rightarrow \langle \{\textit{black}, \textit{white}\}; = \rangle \\
\textit{size}: & \langle \mathscr{V};\ \textit{better} \rangle \rightarrow \langle \{\textit{size}1, \textit{size}2, \textit{size}3\}; \leq \rangle \\
\textit{conection}1: & \langle \mathscr{V}; \textit{same-element} \rangle \rightarrow \langle \mathscr{V}'; = \rangle \\
\textit{connection}2: & \langle \mathscr{V}; \textit{same-element} \rangle \rightarrow \langle \mathscr{V}'; = \rangle
\end{array}$$

Tables 4 and 5 can now be represented as the following relational system:

$$\langle \mathscr{V} \cup \mathscr{V}';\ \textit{same-seg}, \textit{longer}, \textit{length}, \textit{null}, \textit{same-hue}, \textit{thicker}, \textit{same-element} \rangle$$

4.2. Structure-preserving Mapping

In the previous section, typed data tables and typed visual tables are defined as relational systems that are obtained by combining data and visual attributes, respectively. The visualization relation between a typed data table and a typed visual table can now be defined as a mapping between their corresponding relational systems. Although any mapping from the relation system of the data table to the relational system of the visual table provides a visualization, only a subset of them can be claimed to satisfy the structural correspondence condition. This condition will be satisfied by demanding a structure-preserving map. In fact, the structural correspondence condition will be satisfied if and only if there is an isomorphism between relational system that represents the typed data table and relational system that represents the typed visual table.

Definition 4. An isomorphism between $\langle \Delta_1; R_1, \ldots, R_n \rangle$ and $\langle \Delta_2; S_1, \ldots, S_n \rangle$ is a one-to-one and onto mapping between Δ_1 and Δ_2 and a one-to-one mapping between relations R_i and S_i (for $i = 1, \ldots, n$) which satisfies the following condition: If a relation R_i holds between two elements of Δ_1, the corresponding relation S_i holds between the corresponding elements of Δ_2, and if R_i does not hold between two elements of Δ_1, S_i does not hold between the corresponding elements of Δ_2.

The isomorphic mapping guarantees that each data entry corresponds to a visual entry, and whenever there exists an intended relation between two data entries their corresponding visual entries are related to each other by a structurally identical perceptual relation. The isomorphic mapping requires a correspondence between data relations and visual relations that have the same type. This condition is a partial reformulation of the effectiveness criterion, which states that in an effective visualization data attributes are visualized by visual attributes that have the same attribute type. It is important to note that the isomorphic mapping is a necessary but not a sufficient condition for effective visualization. In fact, other design factors such as cultural conventions and subjective preferences play important roles in the effectiveness of visualizations too.

4.3. Undecided Visual Attributes and Effective Visualization

The isomorphic relation is defined between a relational system that represents a typed data table and a relational system that represents a typed visual table. However, the typed visual table is only based on decided visual attributes. In Section 2.4, it is argued that the choice of values for undecided visual attributes play also an important role in the effectiveness of visualization. It is shown that a wrong choice of values of the undecided visual attributes may result unwanted visual implicatures, as illustrated in visualization 2-A.

In this section, we briefly elaborate on the choice of values for undecided visual attributes to prevent a certain class of unwanted visual implicatures. The unwanted implicatures that we aim to prevent are perceivable relations that do not represent any intended data relations. For example, consider visualizations that are illustrated in Figure 8. The difference between these two visualizations is the position value of the second box at the middle layer of the graph. In visualization 8-A, boxes 1,3, and 4 are perceived as being related to each other, but differentiated from box 2. If there is no difference between data elements that are visualized by boxes 1,3, and 4 on the one hand and the data element that is visualized by box 2 on the other hand, then the suggested relation between boxes 1,3, and 4 and their difference with respect to box 2 does not represent any data relation and therefore it forms an unwanted visual implicature. The same type of unwanted visual

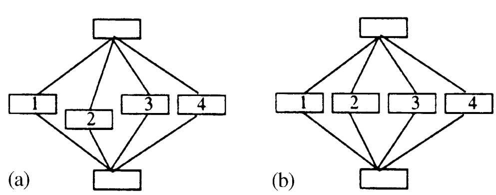

Figure 8. Boxes 1,2,3, and 4 in diagram A cause unwanted visual implicature while in diagram B they do not, assuming that they both visualize the same information and that diagram *B* visualize it correctly

implicature has also occurred in visualization 2-A where the perceivable relation between shapes of visual elements did not represent any data relation.

This class of unwanted visual implicatures can be avoided by demanding that each undecided visual attribute should only induce the identity relation on visual elements. Visual elements will then be perceived as being in one perceptual group with respect to undecided visual attributes. Many visual attributes such as color-hue, size, shape, and texture induce the identity relation on visual elements when they assign identical values to visual elements. For other visual attributes such as position attribute, visual elements cannot have identical values since otherwise visual elements will be placed on the top of each other. However, these visual attributes can induce the identity relation on visual elements if there exits perceptually motivated regularity between values that they assign to visual elements[f]. For example, in visualization 8-B, the equidistance regularity between position values induces the identity relation on the four boxes at the middle layer of the graph. A detail analysis of the classes of perceptually motivated regularities for different visual attributes can be found in [10, 18].

We assume a set of empirically validated and application-dependent perceptually motivated regularity relations for various visual attributes and define undecided visual attributes in terms of these regularity relations. In particular, undecided visual attributes are defined as attributes that induce the identity relation on visual elements by assigning values, which are related to each other by perceptually motivated regularity relation, to visual elements.

Definition 5. Let B be the set of values of the visual attribute α among which there exists a perceptually motivated regularity relation Reg. Then, α is an undecided visual attribute in a visualization if and only if it is a homomorphic map from the relational system $\langle A; \approx \rangle$, where A is the set of visual elements, into a relational system $\langle B; Reg \rangle$.

For example, consider again visualization 8-B and suppose that the *xpos* attribute is an undecided visual attribute. Let $\mathcal{N}$ be the set of integers considered as the position values on the x-axis and $+1$ be a perceptually motivated regularity relation. Then, the undecided visual attribute *xpos* can be defined as the following mapping:

$$xpos : \langle A; \approx \rangle \rightarrow \langle \mathcal{N}; +1 \rangle .$$

It is important to note that the regularity relations, which are supposed to induce the identity relation on visual elements, are context sensitive. An example of this context dependency is the Mueller–Lyer illusion, which is illustrated in Figure 9.

This example shows that the regularities of attribute values of visual elements (in this case horizontal lines) do not induce the identity relation on them such that human viewer cannot perceive them as being in one perceptual group. This effect is due to the presence of some other visual elements (in this case the attached $\rangle$ and $\langle$ elements). However, this

[f]There is a close relationship between Gestalt laws, which explain the perceptual groupings of visual elements, and the regularity principle. In fact, according to the psychologically motivated Structural Information Theory [18], regularity is the basic principle that governs human perceptual system such that all Gestalt laws can be explained in terms of the regularity principle.

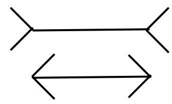

Figure 9. The Mueller–Lyer illusion

context problem is out of the scope of this paper since the aim was to formulate necessary conditions for effective data visualization and not sufficient conditions.

5. Conclusion and Future Research

The main focus of this paper was to formulate the structural correspondence condition, which is a necessary condition for effective data visualization. We argued that the correspondence between data and its visualization should be perceptually motivated in the sense that the intended structure of data should coincide with the perceptual structure of its visualization. To this end, we have studied perceptually motivated structures that can be used in visualizations to represent data structures. These perceptual structures are formally defined in terms of perceivable relations that are induced on visual elements by means of visual attributes. Based on the formal definitions of data and perceptual structures, we have formulated the structural correspondence condition as a structure-preserving map between data and perceptual structures.

In this paper, we have avoided the term 'visual languages', which is often used in the literature of data visualization [3, 19, 20]. In these studies, each visualization is considered as an expression or a sentence of a visual language. A visual language can be defined in our proposed framework as consisting of two relational systems, a data system and a visual system, and a mapping between them. The mapping between these relational systems represents the denotation relation between visualizations and data. Different pairs of relational systems specify different visual languages. Moreover, for each pair of relational systems, there are many mappings possible, each of which specifies a different visual language. A visual language may generate different visualizations by using either different number of elements from visual systems or by using different values for their visual attributes. Two different bar-charts that differ from each other in the number of bars or by their visual attribute values are considered as different expressions of one and the same visual language, namely the bar-chart language. Various kinds of maps, diagrams, graphs, and flow-charts can be considered as different visual languages. We leave the details of defining various visual languages in our framework for the future research.

An important issue which is not worked out here is the interaction between various visual attributes. These interactions are essential since they influence the perceptual structure of visualizations [16, 17]. In order to study the interaction of attributes, we may use a psychological notion called integrality of attributes. Two attributes are called integral when the perceptual structure is not induced by one or the other attribute separately, but rather by the overall similarities of visual elements caused by the combination of the integral attributes. For example, using the hue, the saturation, and the brightness attributes of visual elements, the induced structure is not by one of these three attributes, but it is induced by the overall similarities of elements which is caused by the hue, the saturation, and the brightness attributes together. In contrast, two attributes are called non-integral when the perceptual structure of elements is induced by one or the other attribute. For example, considering the shape and the texture attributes of visual elements, a perceptual structure may be induced by the shape attribute, by the texture attribute, or by both of them. In the last case, each attribute supports that perceptual structure separately. We have focused on perceptual structures that are defined in terms of non-integral attributes. A specification of structures that are defined in terms of integral attributes remains an open problem for the future research.

References

1. J. Bertin, (1981) *Graphics and Graphic Information-Processing.* Walter de Gruyter, Berlin New York.
2. S.K. Card, Mackinglay, J.D. & B. Shneiderman (1999) *Readings in Information Visualization-—Using Vision to Think*. Morgan Kaufmann, San Francisco.
3. J. Mackinlay (1986) Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics* **5,** 110–141.
4. E.R. Tufte (1983) *The Visual Display of Quantitative Information.* Graphics Press, Cheshire, CT.
5. E.R. Tufte (1990) *Envisioning Information.* Graphics Press, Cheshire, CT.
6. M. Twyman (1979) A schema for the study of graphic language. In: *Processing of Visible Language* (P.A. Kolers, M.E. Wrolstad, & H. Bouma, eds), Vol. 1. Plenum Press, New York.
7. W.J. Mitchell (1993) A computational view of design creativity. In: *Modeling Creativity and Knowledge-based Creative Design* (J.S. Gero & M.L. Maher, eds). Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 25–42.
8. M. Petre & T.R.G. Green (1993) Learning to read graphics: Some evidence that 'seeing' an information display is an acquired Skill. *Journal of Visual Languages and Computing* **4**, 55–70.
9. S.S. Stevens (1946) On the theory of scales of measurement. *Science* **103,** 667–680.
10. M. Dastani (1998) Languages of perception. Ph.D. thesis, Institute of Logic, Language, and Computation (ILLC), The Netherlands.
11. J. Marks & E. Reiter (1990) Avoiding unwanted conversational implicatures in text and graphics. *Proceeding AAAI*, Menlo Park, CA.
12. J. Pfanzagl (1968) *Theory of Measurement*. Physica-Verlag, Würzburg-Wien.
13. K. Reichenberger, T. Kamps & G. Golovchinsky (1995) Towards a generative theory of diagram design. In: *IEEE Proceedings on Information Visualization* (N. Gershon & S. Eick, eds), October 1995. IEEE Computer Society Press, Silver Spring, MD, pp. 11–18.
14. Y. Engelhardt, J. de Bruin, T. Janssen, & R. Scha (1996) The visual grammar of information graphics. *Artificial Intelligence in Design (AID'96), in the Workshop on Visual Representation, Reasoning and Interaction in Design*, 24–27 June, Stanford University, California.
15. T.M. Kamps (1997) A constructive theory for diagram design and its algorithmic implementation. Ph.D. thesis.
16. W.R. Garner (1974) *The Processing of Information and Structure*. Lawrence Erlbaum Associates, Potomac, MD.
17. R.N. Shepard (1991) Integrality versus separability of stimulus dimensions: from an early convergence of evidence to a proposed theoretical basis. In: *The Perception of Structure* (G.R. Lockhead & J.R. Pomerantz, eds). American Psychological Association, Washington, DC, pp. 53–71.
18. P. Van der Helm (1994) The dynamics of prägnanz. *Psychological Research* **56**, 224–236.
19. C.A. Gurr (1999) Effective diagrammatic communication: syntactic, semantic and pragmatic issues. *Journal of Visual Languages and Computing*, **10,** 4.
20. D. Wang, & H. Zeevat (1998) A syntax-directed approach to picture semantics. In: *Visual Language Theory: Towards a Human–Computer Interaction Perspective*. (B. Meyer & K. Marriott, eds) Springer-Verlag, Berlin, 307–324.