The Visual Computer (2018) 34:1087–1098
https://doi.org/10.1007/s00371-018-1516-8

ORIGINAL ARTICLE

Cross-table linking and brushing: interactive visual analysis of multiple tabular data sets

Rainer Splechtna[1] · Michael Beham[1] · Denis Gračanin[2] · María Luján Ganuza[3] · Katja Bühler[1] · Igor Sunday Pandžić[4] · Krešimir Matković[1]

Published online: 3 May 2018

Abstract
Studying complex problems often requires identifying and exploring connections and dependencies among several, seemingly unrelated, data sets. Those data sets are often represented as data tables. We propose a novel approach to studying such data sets using linking and brushing across multiple data tables in a coordinated multiple views system. We first identify possible mappings from a subset of one data set to a subset of another data set. That collection of mappings is then used to specify linking among data sets and to support brushing across data sets. Brushing in one data set is then mapped to a brush in the destination data set. If the brush is refined in the destination data set, the inverse mapping, or a back-link, is used to determine the refined brush in the original data set. Brushing and back-links make it possible to efficiently create and analyze complex queries interactively in an iterative process. That process is further supported by a user interface that keeps track of the mappings, links and brushes. The proposed approach is evaluated using three data sets.

Keywords Visual analytics · Interactive visual analysis · Multiple data sets analysis

1 Introduction

The visual data analysis process usually starts with the creation of a data set in the form of a data table. There are several ways how tables can be created from the same data set, depending on the analysis questions [2]. In this paper, we assume that a data table contains records (sometimes called data points or items) referred to as table rows. Each table row contains several attributes (table columns). Most visual analysis systems expect one such data table as basis for the analysis.

As the analysis tasks become more complex, analysts have to analyze hidden interplays between different data sets. Imagine an analyst who wants to see if weather characteristics influence real estate prices in an area or if there is a correlation between the number of influenza infections and the usage pattern of public transport in a city. The data used for such analysis is often stored in separate data tables with different structure (attributes/columns). Simultaneous analysis of several data tables supports deeper analysis.

We propose a novel way of interactive visual analysis of multiple data tables originating from different data sources. This approach does not rely on the availability of a (unique) key to link records from different data tables. Those records can, e.g., refer to houses in one data table, to counties in other, and to meteorological stations in another data table. Each data table can have a different structure.

We extend the well-known principle of linking and brushing to support multiple data tables by introducing data table links. A data table link can be established via single or multiple attributes. The link can be single-directional or multi-directional.

We also introduce *implicit brushing*, a direct mapping of explicit brushing (selected rows) in one data table to the corresponding rows in other linked data tables. The implicit

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s00371-018-1516-8) contains supplementary material, which is available to authorized users.

✉ María Luján Ganuza
mlg@cs.uns.edu.ar

[1] VRVis Research Center, Donau-City Str. 11, 1220 Vienna, Austria

[2] Virginia Tech, Blacksburg, VA 24060, USA

[3] VyGLab Research Laboratory, DCIC, UNS, San Andrés 800, 8000 Bahía Blanca, Argentina

[4] University of Zagreb, Unska 3, 10000 Zagreb, Croatia

brush can be refined in any data table. As the name suggests, the implicit brush has no explicit visual representation but still acts as a regular brush. A mechanism for links and brushes creation and maintenance is provided. We implemented the proposed approach in a Coordinated Multiple Views (CMV) system and evaluated the cross-table linking and brushing using three data sets: US meteorological stations monthly summary data, the California housing data set, and the Boston housing data set.

Dealing with multiple data tables in the databases and analysis domains is not unusual. However, as there are no unique common keys across all tables, the proposed approach defines links similar to general joins in the database techniques.

The newly proposed visual analysis methodology, enables the analyst to gain insight into hidden interplay among data sets by means of interactive visual analysis. The main contributions are: (1) a novel methodology for cross-table linking and brushing for data tables without unique key, (2) formalization of cross-table links for interactive visual analysis, (3) implicit brushing, and (4) evaluation of the newly proposed approach using three different data sets.

2 Related work

Linking and brushing is a well-known paradigm used in CMV systems. The goal is to show multidimensional data sets using different views and interactively select—brush—some data items in one view and see the corresponding items in all other views. The main idea was introduced by Fisherkeller et al. [9] in their PRIM-9 system (but they did not call it *brushing*). They used an operation for interactive selection of a region. Becker and Cleveland [1] introduced the term *brushing*.

Many tools support various brushing techniques. For example, Martin and Ward [19] introduced logical combinations of brushes using Boolean operations in the XmdvTool. Doleisch et al. [8] extended the concept and introduced a feature definition language to make the specification of complex brushes easier. Some tools support interactively building multiview visualizations using a simple shared-object coordination mechanism (Improvise [30]) or leveraging a conceptual model based on the relational database model (Snap-together [23]).

Weaver [31] used modularized cross-filtering, reused it across designs, and customized based on data types across multiple dimensions. Cross-filtered visualization has three basic elements: views, brushes, and switches. Brushing is used to select data items (not a region) and unique key. The requirement is that unique attribute values have individually brushable representations.

Shadoan and Weaver [26] analyzed the expressive power of existing visual querying systems and described a more flexible approach. Users can interactively construct queries as visual hypergraphs to explore n-ary conjunctive inter- and intra-dimensional relationships.

Interactive visual analysis and CMV is a proven concept for data analysis in different domains [2,8,13,14,27]. Many complex brushes have been proposed besides a basic rectangular brush in a scatter plot. Konyha et al. [13] introduced the line brush, a brush which selects curves that cross a user drawn line. An angular brush [12] selects only a subset of lines, in parallel coordinates, e.g., that have the slope in a certain range. A triangle brush [10] is a special brush for a barycentric view which takes data characteristics into account. Radoš et al. [25] introduce Mahalanobis and percentile brush which take data distribution into account. All of above-mentioned approaches deal with a single data set. We aim at analyzing multiple data sets by establishing links between them. There is not much research on the analysis of multiple data sets simultaneously.

Turkay et al. [28] deal with dual data sets. They derive dimensions space from the original space. They deal with two data tables, but the tables are dual representation of the same data. Liu et al. [17] proposed a network-based analysis of multiple data tables with a unique key which can be used to link the tables.

Relational databases [11] consist of multiple data tables linked using keys based on the relational model [5]. Queries are specified using Structured Query Language (SQL) [6], which incorporates relational algebra [5]. Virtual views are relations that could link and span over multiple data tables and are defined by a query over other relations. Those views are specified as join operations between data tables [11].

Visual Query Systems support spatio-temporal data queries [3]. General spatial composition frameworks allow us to model the graphical objects and the spatial relations of a large class of geospatial data [7]. Further, both visual specification of relational database queries and visual description of "ordinary" SQL queries could be presented [4]. We link data tables (like virtual views) not directly linked by a common key.

3 Proposed approach

Distributing data to several data tables is a common approach in relational databases [11] and supported by visualization tools [17]. A (possibly unique) primary key is assigned to each row (record) in a table to connect items from various data tables. A relationship between two tables is established when one table has a foreign key that references the primary key of the other table. Such a relationship between table records could be one-to-one, one-to-many and many-to-many.

Our data tables are not related in this way, i.e., keys are not used to connect records from different data tables. The tables

do not represent different aspects of the same data, the tables in our case represents different data sets. Still, we would like to explore correlations and hidden information in them.

A cross-table link supports complex analysis of multiple data sets. We must define which records from other tables correspond to selected records of a particular table. Although such an analysis is possible using conventional analysis and advanced SQL features, the newly proposed interactive cross-table linking and brushing makes the whole process more efficient and intuitive.

3.1 Formal model

We use relational algebra [5] to describe data tables links and to construct new relations from given relations. The relations are data stored in data tables, and constructed relations are answers to queries about the stored data. We start from the notion of the relational operations on bags, i.e., the same tuple of attributes values could appear in more than one data table row.

A data table represents a relation R where each of m rows is a tuple of n attributes $(A_1, \ldots, A_n)$. We then link data in different data tables (i.e., given relations) by matching a subset of rows in one data table to a subset of rows in one of more other data tables. The link is specified by constructing a new relation as a join or pairing of rows (tuples) that match in some way.

The simplest join is the *natural join* of two relations (data tables) R and S, denoted $R \bowtie S$, that match in some attributes common to R and S. The natural join consists of all combinations of tuples (rows) $r \in R$ and $s \in S$ paired based on the selected attributes. Figure 1a shows a simple example. Relation R has two tuples providing item inventory. Relation S has three tuples providing city population. Relation R and S share one attribute (column), $Count$. The resulting natural join using $Count$ attribute and tuples with the same value of 50 000, produces a single tuple in $R \bowtie S$.

However, when dealing with data sets from different tables, exactly matching the attribute values is not always feasible. Furthermore, an attribute in one data table could be expressed as a function of one or more attributes in other data table(s). In that case we use another form of join, the *theta-join* [11], where a condition C is used to construct a new relation, i.e., $R \bowtie_C S$. Figure 1b provides a simple example. Relation R has two tuples for item inventory. Relation S has three tuples for city population. Relation R and S share one attribute (column), $Count$. The resulting join using $Count$ attribute and the condition that the value of $Count$ in R is greater than or equal to the value of $Count$ in S produces three tuples in $R \bowtie_C S$.

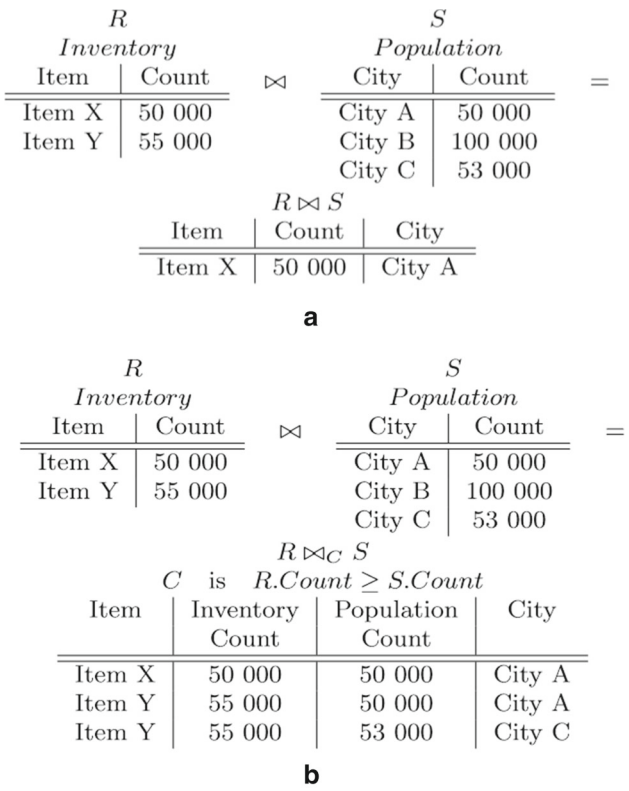


Fig. 1 **a** An example of a natural join. The common attribute is $Count$ and value 50 000 is present in both relations. **b** An example of a theta-join. The common attribute is $Count$ and condition is that the value of $Count$ in R is greater than or equal to the value of $Count$ in S

3.2 Implicit brush

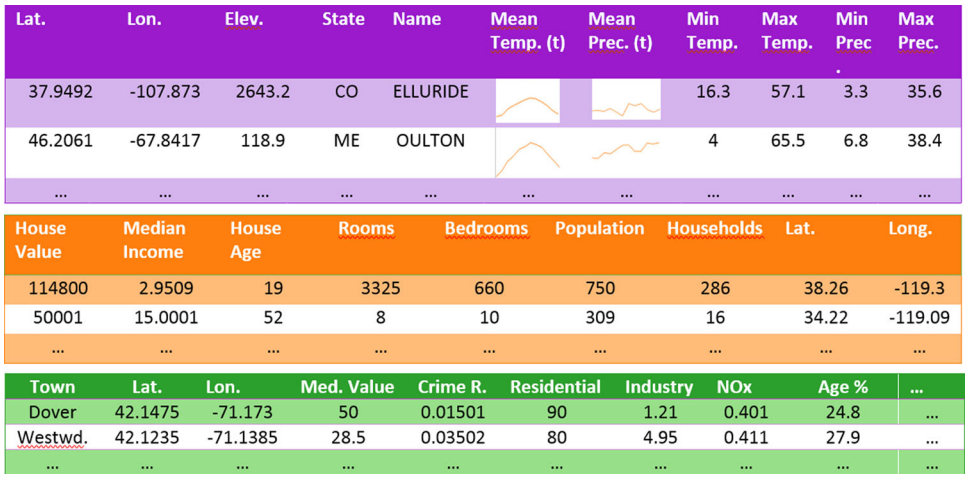
Sometimes the join is constructed using a subset of an existing relation, i.e., a subset R_s of an existing relation R is used (Eq. 1):

$$R_s \bowtie S \quad \subset \quad R \bowtie S \tag{1}$$

When a single data set is analyzed in a CMV system, the user draws a brush (selects a subset of R) and sees the corresponding items (linked data). The brush is usually displayed as a re-sizable and movable frame, line, or other shape depending on the brush.

When we use multiple tables, the data brushed in one data set $R_s \subset R$ results in linked data in other data sets. There is no graphical representation of the brush itself in the view container which shows the linked data set. Still, the items are highlighted. We call such a brush an implicit brush. Using the theta-join example in Fig. 1b and brushing the second tuple in R, $(Item\ Y,\ 55\,000)$, the implicit brush in S produces two tuples $(Item\ Y,\ 55\,000,\ 50\,000,\ City\ A)$ and $(Item\ Y,\ 55\,000,\ 53\,000,\ City\ C)$.

Finally, we can explore implicit brushing in the opposite direction (back-link). Let $S_{R_s \bowtie S}$ indicate all tuples in S that



Lat.	Lon.	Elev.	State	Name	Mean Temp. (t)	Mean Prec. (t)	Min Temp.	Max Temp.	Min Prec.	Max Prec.
37.9492	-107.873	2643.2	CO	ELLURIDE			16.3	57.1	3.3	35.6
46.2061	-67.8417	118.9	ME	OULTON			4	65.5	6.8	38.4
...	...	...	...	...	...	...	...	...	...	...

House Value	Median Income	House Age	Rooms	Bedrooms	Population	Households	Lat.	Long.
114800	2.9509	19	3325	660	750	286	38.26	-119.3
50001	15.0001	52	8	10	309	16	34.22	-119.09
...	...	...	...	...	...	...	...	...

Town	Lat.	Lon.	Med. Value	Crime R.	Residential	Industry	NOx	Age %	...
Dover	42.1475	-71.173	50	0.01501	90	1.21	0.401	24.8	...
Westwd.	42.1235	-71.1385	28.5	0.03502	80	4.95	0.411	27.9	...
...	...	...	...	...	...	...	...	...	...

Fig. 2 Three data sets used throughout the paper. The data sets contain scalar and time-series attributes. Each data set describes different items: US meteorological stations (meteorological data) depicted in purple, areas (California data) depicted in orange, and houses (Boston data) depicted in green

are part of the join operation. We can select a subset S_s of $S_{R_s \bowtie S}$ and then determine a subset of R_s that maps to S_s in the joint operation.

4 Multiple data tables and cross-table links

We use three data sets throughout the paper to illustrate multiple data tables analysis and cross-table links. The data sets are the US meteorological stations data set [22] (shown in purple), the California housing data set [24] (shown in orange), and the Boston housing data set [15] (shown in green). Each data set contains several scalar and categorical attributes. The meteorological data set also contains two time-series attributes—temperature and precipitation over a year. Every data set contains latitude and longitude attributes but the position has different meaning in each set.

In the US meteorological stations data set, the longitude and latitude correspond to the locations of meteorological stations. The California housing data set contains averaged data for larger areas where the position corresponds to the center of the area, not to a single object. In the Boston housing data set, the attribute values describe single houses.

The California housing and Boston housing data sets contain information about the houses, the households, and their neighborhood. In the California housing data set, the median income of the households (in dollars) and the mean number of people within the household give us information about the households. The housing price, number of rooms (and bedrooms) per units, and the house age is provided averaged per area. The neighborhood is described by the population, and the number of households [24].

In the Boston housing data set, the households are described by their percentage of people with lower income. For each house, information about the average number of rooms per dwellings and the full value property tax rate per $10,000 is provided. The crime rate, nitric oxides concentration, the proportion of owner occupied units build prior to 1940 provided, and distance to the five Boston employment centers give us information about the neighborhood [15]. Since California and Boston are two distinct geographical areas, no longitude–latitude pair exists across the data tables. Figure 2 shows the three data tables and some of their attributes.

Figure 3 shows the overall approach. We load two independent data sets, US meteorological stations (purple) and California census data (orange). The CMV system Comvis [20] is used to independently explore data sets and select: *1)* US meteorological stations in dry areas (purple) and *2)* California counties with old houses and just a few households—counties centroids are shown (orange).

After independent data set exploration, we can establish first links. We use longitude and latitude to cross-link two data sets. They are not unique keys, as area centers and US meteorological stations have different coordinates. Instead of searching for common keys, the longitude/latitude link will brush all items in the vicinity of the selected US meteorological stations.

We brush certain station characteristics (low precipitation, for example), and all areas in California whose centroids are close to the stains will be implicitly brushed. The influence radius can be set by user, we select 10 miles in our case. The scatter plot in the orange data set shows data for counties whose center is within 10 miles circle around meteorological stations in California with low precipitation values. Those counties have average house ages across the whole range, but they have rather low number of households. The data points are shown in purple (brush origin) in the orange data set.

Links can be refined (e.g., radius change or adding a new brush), new links can be added, and so on to continue data exploration. The exploration can be extended to include additional data sets (Sect. 5.2).

A cross-table analysis of such data might be illustrated by using the following example scenario. Analysis starts with

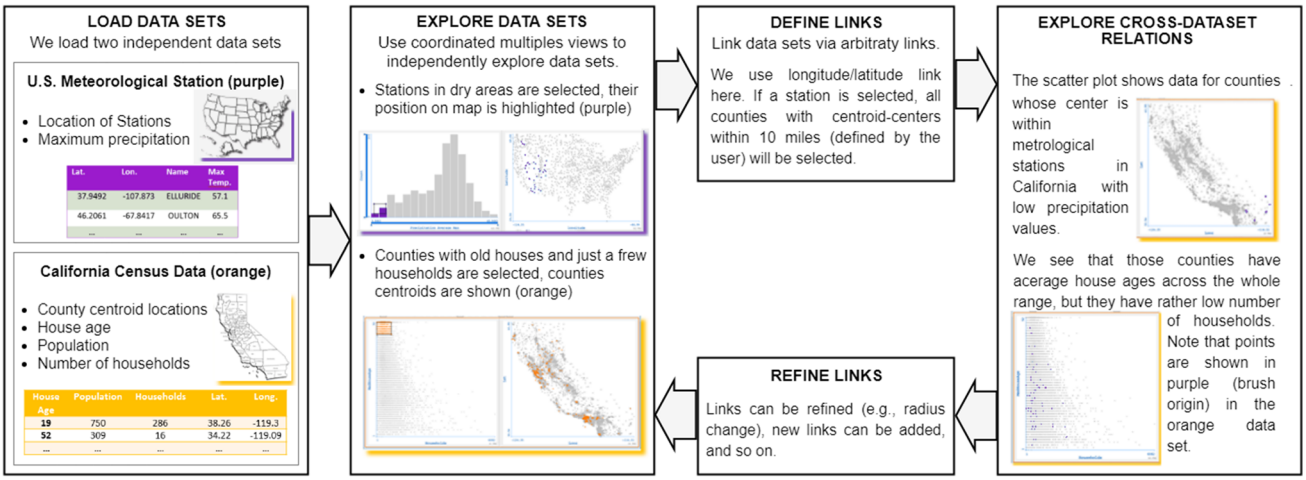


Fig. 3 The main principle of cross-table linking and brushing using two data sets, US data (purple) and California data (orange). Multiple, independent data sets are explored simultaneously, and, at the same time, the cross-data set relations can be explored. The user brushes some items in one data set and defines links toward other data set(s). There is no unique key in both data sets, links define arbitrary relations. In such a interactive and iterative process users can create complex queries efficiently and intuitively, and explore hidden relations between data sets

exploration of the meteorological data. An analyst checks meteorological stations in the areas with hot summers and dry winters. Once the stations are identified, the analyst wants to see the prices and number of rooms of houses in California that are located close to those stations, for example, within 15 kilometers. Once the areas in California are located, the houses with the same number of rooms in Boston can be selected and explored.

As stated above, we need to establish cross-table links between the data tables. Note that there is no unique key or some kind of identifier which would link the tables. Each data set is independent.

We can consider two data tables R and S to have m_R and m_S rows, respectively. They have attributes (columns) $(A_1^R, \ldots, A_{n_R}^R)$ and $(A_1^S, \ldots, A_{n_S}^S)$, respectively. The individual attributes A_i could have scalar, categorical, or time-series values.

Now we can establish cross-table links. The most basic case is one-to-one attribute link. If attributes A_i^R and A_j^S are linked, then for all rows b_S in S that are implicitly brushed (S_b), there is a brushed row b_R in R such that the value of A_i^R equals the value of A_j^S, i.e., $a_i^{R,b_R} = a_j^{S,b_S}$ (exploring implicit brush in the opposite direction). However, when comparing two different data sets, the exact match of values is not always feasible. Instead, we are using a condition C. There are numerous ways to specify the condition C.

4.1 Conditions

If an attribute is in both data tables, a threshold will be most often used. For example, if both data tables contain elevation, we can use it to link the data tables. Note that it is not a key, we use it to establish a cross-table link. If units of the elevation are same, a simple threshold ϵ may be used as a condition (Eq. 2).

$$\|a_j^{S,b_S} - a_i^{R,b_R}\| \leq \epsilon \tag{2}$$

A threshold can be specified as a value or percentage. If the units are not same, e.g., R contains elevation in feet and S in meters, a coefficient k in addition to the threshold could be used as a condition (Eq. 3).

$$\|a_j^{S,b_S} - k \times a_i^{R,b_R}\| \leq \epsilon \quad k = 3.28084 \tag{3}$$

In the same way it is also possible to create many-to-many cross-table links. For a location (specified by more than one attribute, e.g., x and y), conditions include region based, Manhattan Distance, Euclidean Distance, Mahalanobis Distance [18,21], latitude/longitude [29], and cosine similarity between two tuples.

- Region based: $|R.x - S.x| \leq \epsilon_x$ and $|R.y - S.y| \leq \epsilon_y$
- Manhattan Distance: $|R.x - S.x| + |R.y - S.y| \leq \epsilon$
- Euclidean Distance: $\sqrt{(R.x - S.x)^2 + (R.y - S.y)^2} \leq \epsilon$
- Mahalanobis Distance [18,21]: between two tuples $\mathbf{x} = (x_1, \ldots, x_n)^T$ and $\mathbf{y} = (y_1, \ldots, y_n)^T$ with covariance matrix $\mathbf{S}$ is: $\sqrt{(\mathbf{x} - \mathbf{y})^T S^{-1} (\mathbf{x} - \mathbf{y})}$
- Latitude and Longitude [29]: geodesic distance is the shortest distance between two points on the surface of an ellipsoid.

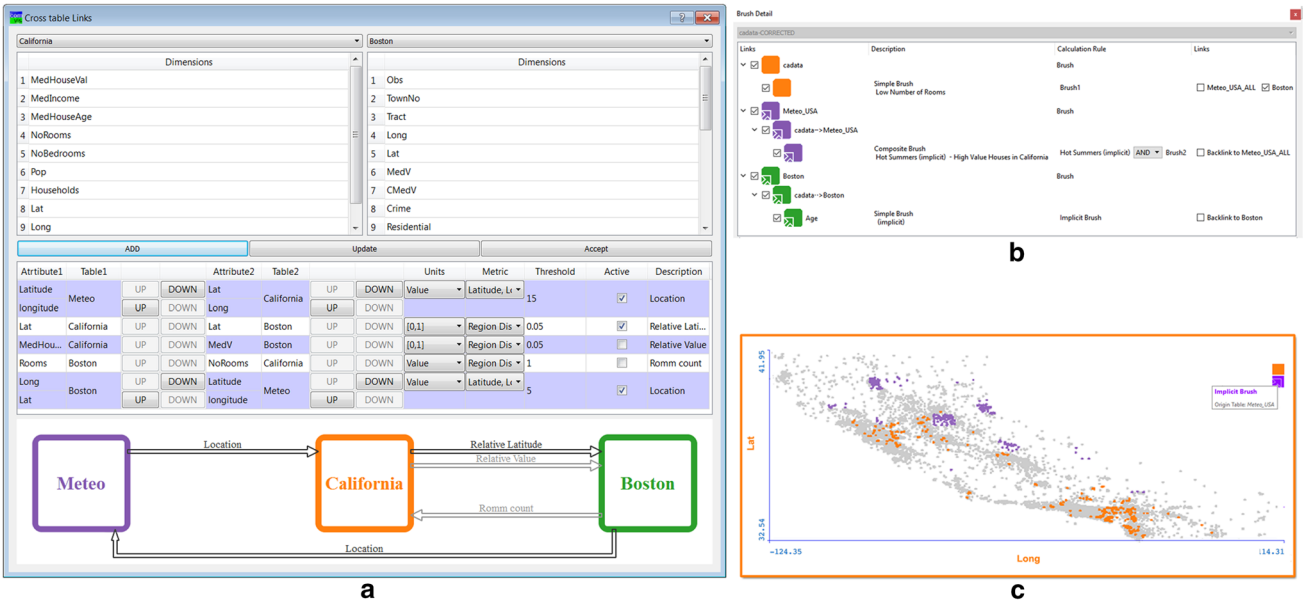


Fig. 4 **a** Proposed user interface for links definition. The user selects two data tables which should be linked (top part), and then selects attributes from each table. A new link is created and added to the table (middle part). The user can select metrics, units, add a description to the link, and enable it. The links are visualized (bottom part), so that users see which links exist and which are active (inactive links are grayed out). **b** The proposed user interface helps the users to keep track of brushes, links and back-links. We group brushes according to the data set which they influence. Such an interface is essential even for a moderate number of brushes and links. **c** In each view, brush legends provide more information about the brushes

– Cosine Similarity: between tuples $\mathbf{x} = (x_1, \ldots, x_n)^T$ and $\mathbf{y} = (y_1, \ldots, y_n)^T$ is:
$\cos\theta = \frac{\mathbf{x}\cdot\mathbf{y}}{|\mathbf{x}||\mathbf{y}|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$

Furthermore, we can use additional preprocessing before applying the conditions listed above:

– Normalization between 0 and 1: scalar attributes values a_i are modified so that:
$a_i' = \frac{a_i - \min a_i}{\max a_i - \min a_i}$
– Normalization between -1 and 1: scalar attributes values a_i are modified so that:
$a_i' = -1 + 2\frac{a_i - \min a_i}{\max a_i - \min a_i}$
– Sum over selected attributes: a sum of the values in tuple $\mathbf{x}$ for the selected attributes.
– Product over selected attributes: a product of the values in tuple $\mathbf{x}$ for the selected attributes.

4.2 User interfaces

Although the basic principle is relatively simple, keeping track of all the brushes and links is very challenging. During our numerous experiments, we realized that we need a way to support the user in this task. We have to externalize information about active brushes, active back-links, links themselves, etc.

As there are multiple views per data set visible on the screen (or hidden if tabbed version is used), each view container has a frame in a unique color. This color helps the user to link the views to data.

The first step in the exploration is links creation. We provide a user interface which guides the user through the process (Fig. 4a). The proposed interface allows the user to select two tables, and then to select which attributes of the tables will be linked. The link is then created. As we support different metrics, the user can now select which one is used. If normalization is necessary, it can also be selected. As it is necessary to keep tracks of the links, we provide a graphical way of depicting all links. Figure 4a shows an example with three data sets. The links are shown and the user can see which of them are active at the moment (light gray arrows for inactive ones, and black arrows for the active links). Providing such an interface is essential for cross-table linking and brushing. Easy access to links, and clear representation of all available links reduces cognitive load significantly. Our design for visual representation of links scales well for up to, approximately ten links per data set pair, and up to a handful of data sets. This seems as a reasonable limitation, as handling more links or data sets becomes infeasible. There is no explicit limitation regarding the number of items per data set.

As there is no representation of the linked brush, we provide a central place in the container to see all brushes. The implicit brushes are marked with an arrow symbol which suggests that the brush originates from a link. The brush control

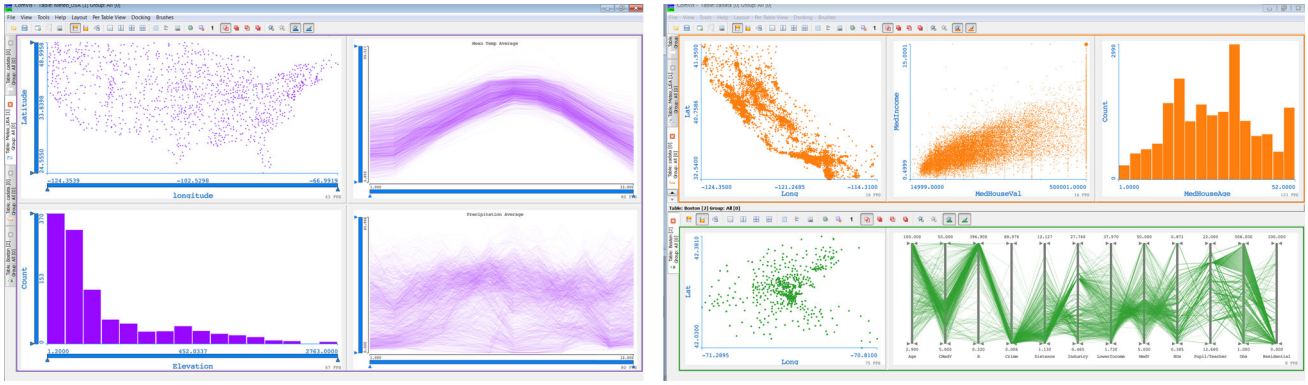


Fig. 5 Viewing the three example data sets. *Left* four views of the US meteorological stations data set (purple). *Right top* three views of the California data set (orange). *Right bottom* two views of the Boston data set (green)

can also be used to deactivate brushes if needed. The evaluation section provides numerous examples which illustrate this mechanism.

Figure 4b shows the developed user interface. It is a combination of a table and a tree like structure. For each data set, all brushes are grouped. In this way, the user can immediately see which brushes affect a data set. The California data views (they will have orange frames as described in Sect. 5) are influenced by orange brushes. This group consists of the brush originating from California, and from implicit brushes which originate from the two other data sets. Implicit brushes have different icons.

Furthermore, the interface can show brush description, and how the brush is composed. The user can activate back-links by employing the same interface. When multiple brushes and links are active, it is essential to have such an interface.

Additionally, each view contains a small legend of the brushes that are active in the view. This legend is composed of small squares located in the upper right corner of the view. Each square corresponds to a brush. When the user places the mouse over the brush legend, more information about the brush pops out (Fig. 4c).

Finally, we always use the color of the brushed data set to depict linked data. We do allow to further refine the brush in the linked view, which does not change the color of the brush. We assign orange to data set R and purple to data set S. Now, we brush some data in R, and linked data is depicted in orange in the purple container. The user can refine the implicit brush in the purple container. Resulting data is depicted in orange.

5 Evaluation

We evaluate the newly proposed approach using two scenarios with the three data sets. The authors of this manuscript have been involved in the design and in the evaluation of the proposed approach. Figure 5 shows the view configuration at the start of our analysis—on the left are four views for the US data set, on the right there are three views for the California data set (top), and two views for the Boston data set (bottom). We use additional views during the analysis when needed.

We use purple for the US meteorological stations data set, orange for the California data set, and green for the Boston data set to amplify cognition by externalization. The user immediately sees which data is shown, or which data set is a brush origin. An intuitive assignment of colors to data makes the analysis easier [16]. We asked ten colleagues, all visualization experts, which colors they would assign to the three data sets. Nine experts assigned colors as described above, and one suggested to switch US and California colors.

The US meteorological stations data set configuration contains a scatter plot of longitude and latitude (top left). The map of the USA is clearly recognizable. The histogram below the map shows the distribution of the elevation of the meteorological stations. Most of them have a rather low elevation, as expected.

The two curve views in the right half of the screenshot show temperature (top) and precipitation curves (bottom) for each station. We can clearly see high summer temperatures, and low winter temperatures. The precipitation curves show much more variety in patterns across all stations.

We use the scatter plot of the longitude and latitude in the other two data sets, as well (always in the left in the screenshots). In addition, we show a scatter plot of median house value and median household income, and a histogram showing median house age for the California data, and a parallel coordinates view with several parameters for the Boston data. The views can be shown on several monitors, or, if screen space is sparse, a tabbed version can be deployed. The following sections describe two scenarios, one with two links for two data sets, and a complex scenario, where three data sets and multiple, two-way links, are used.

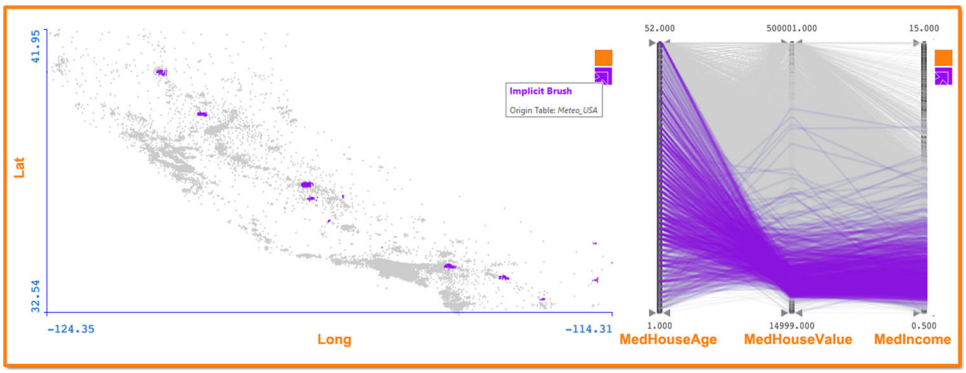


Fig. 6 An implicit brush in the California data set originating from the meteorological data set (purple) where hot summers are brushed. Note that brushed data is shown in purple in the orange data set as the brush originates from the purple data set. The brush itself is not visible here. Left: the scatter plot shows the regions in California which record hot summers. Right: parallel coordinates showing some house attributes for houses in these regions

5.1 Linking two data sets

We start our evaluation with a single link between the meteorological and the California data sets. What if an analyst wants to know, if houses in very hot areas in California have similar characteristics? We do have meteorological data for the meteorological stations, but we do not have them in the California housing data set. Just as in the case which is already described in Sect. 4 (Fig. 3), the first link is a 2D link, which links longitude and latitude values from two data sets.

We create the first link, and declare that the stations from one data set are considered linked with the areas in the other, if their distance is less than 16 kilometers (10 miles). Instead of dry places, we brush the high temperatures in summer in the meteorological data set.

The interpretation of the selected data could be expressed as: *The areas in California whose centroids are within 16 kilometers of meteorological stations which record high temperatures in summer.* The parallel coordinates view (Fig. 6 right) shows the values for house age, house value, and household income. Note that house age is evenly distributed across the whole range, but house value and household income are in the lower part of possible values for California.

Note also that we do not see the brush itself in the California configuration. The brushed data in the California data set (orange) is shown in purple, to help the user to perceive the origin of the brush, in this case the meteorological data set. Hence, the brush color represents an additional implicit brush indicator.

We brushed the areas close to the meteorological stations recording the highest temperatures from another data set without merging the tables. As a result, the number of brushed items in each data set is usually different. Several areas could be close to one meteorological station, and all of these areas are brushed.

The links from one data set to the other allow an advanced analysis. Just as the user can refine or broaden the selection in the origin data set, it is often required to drill down in the destination data as well. We would like to refine the brush in the linked data set.

We can refine the selection shown in Fig. 6 by selecting areas with high-value houses in the California data set. We simply brush the desired range in the parallel coordinates view (original selection shown in Fig. 6). Figure 7a shows the brushed areas in the scatter plot resulting from this brush refinement. The purple circles are added to mark the positions.

There are twelve such areas (we can list them in the details view, a simple table, not shown in the paper). Are there any attributes in the meteorological data set that explain this change? Are there some specific attributes in the set that are related to the high-value houses? We could check it by establishing a new link, from the California data toward the meteorological data. However, a more natural way would be to use the existing link and link the data back to the origin data set. Hence, we introduce a two-way link.

Based on the refinements done in the target table, the composite brush in the origin table is updated. This is done by introducing an implicit brush in the origin table, which represents the refinement in the target table. The brush detail view and the established color scheme help us to remember which brushes belong together.

The selection can be described as: *US meteorological stations recording hot summers that are in California and where the average house value is high within a 16 kilometer radius around meteorological station*. The existence of implicit brushes and active links does not prevent using a native brush in each view.

5.2 Complex links for three data sets

In order to illustrate a complex analysis, we create an additional link between the California and Boston data sets. We

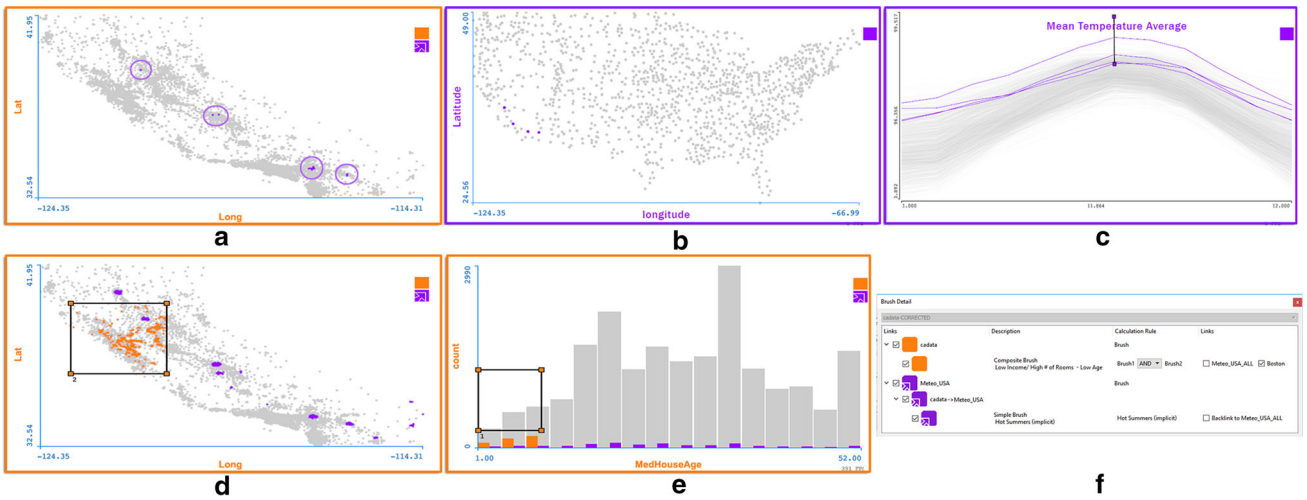


Fig. 7 Refining a brush using an implicit data link. **a** The scatter plot shows the result of a refinement of a selection based on an implicit brush shown in Fig. 6 (right). The purple brush is refined in the orange data set. The purple circles are added for illustration purposes only. **b** The back-link is active and meteorological stations are highlighted in the purple data set. **c** All views in the purple data set are refined now. The temperature curves are shown only for stations selected via back-link from the orange data set. **d** We can always introduce native brush in any view. Here, an orange brush is initiated in the orange data set. **e** Histogram shows both brushes, orange and purple. **f** The brush detail view for the example shown here

use the relative house value as the linking criterion. The use of the relative value partially eliminates real estate prices differences in the two data sets. The values in the California data set are significantly higher than in the Boston data set.

Let us see how hot summers areas in California correspond to Boston, via house value link. Both links are active now (meteorological to California, via longitude–latitude and California to Boston, via relative house value). Just as before, we brush the hot summers in the meteorological data set. The California data set is implicitly brushed and shows the purple selection.

Since we have a chain of links—meteorological to California and California to Boston—this selection is now also linked to the Boston data set and is also shown in purple there. So the selection seen in the Boston data set originates from the link to the California data set, but it is an implicit brush there, and we keep the brush color. The brush, which originates from California is orange. Finally, the brush from Boston is green.

The Boston data implicitly brushed in purple shows houses, which have similar values (in a relative scale) to the average house values in the areas in California which are close to the meteorological stations which record high temperatures in summer. Selecting such combinations is straightforward and intuitive when using the newly proposed cross-table linking and brushing. Achieving the same using conventional SQL commands and joined tables would be quite cumbersome. Figure 8 illustrates the above described case.

We first brush hot summers. We show the longitude/latitude scatter plot for California and a scatter plot of the median income and number of rooms (Fig. 8a). We brush low income, high number of rooms (orange brush). We then compare it with the purple brush. We see that the areas with low income and high number of rooms are further from the coast.

Figure 8b shows the Boston data. Three brushes are active, two implicit and one original brush. The purple brush arrived via two links and the orange brush arrived via a single link. As the same houses in Boston might belong to the same brush, we address the over-plotting problem. We split the bins vertically in the histogram, and offer three techniques for the scatter plot.

The simplest technique is to only show points selected by one brush. Another one is to jitter the points to avoid overlapping. In very dense areas, as in the central area in our example, the overlapping happens again. We can also show small multiples for each brush. Figure 8c shows such a case.

Each implicit brush can be refined in all data sets. These refinements only affect the origin data set and forward-linked data sets. If back-link is enabled (Sect. 3.2), the link origin data set is also affected. Just as forward-links, back-links also work across link chains.

Hence, a refinement in the Boston data set of the purple implicit brush will affect the original brush in the US meteorological stations data set if the back-links from Boston to California and California to meteorological data are activated. It will also affect the purple implicit brush in the California data set, which then consists of a forward-link

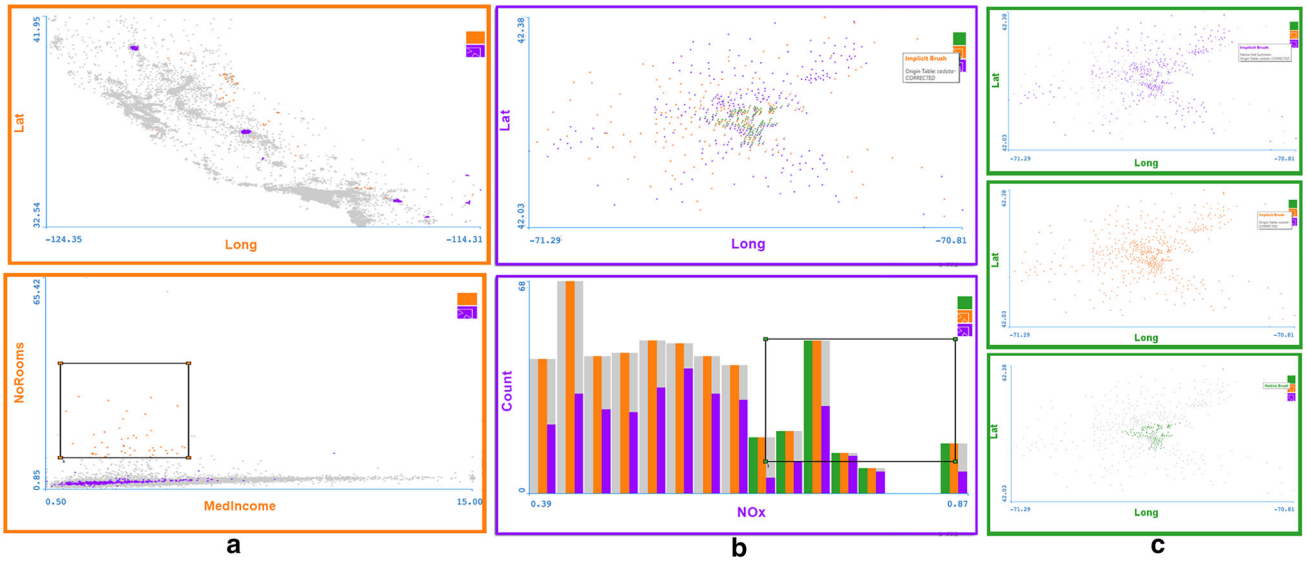


Fig. 8 Multiple links: **a** Views of the California data set showing the implicit brush from the meteorological data set (purple) and a native brush (orange) created in the bottom scatter plot. **b** Views of the Boston data set showing two implicit brushes—one from the meteorological data set (purple) and one from the California data (orange)—and one native brush (green) created in the histogram. The scatter plot uses jittered points for each brush to reduce over-plotting. The histogram splits the bins vertically to show the three brushes side-by-side. **c** An alternative view of the scatter plot shown in **b** where each brush is in a separate scatter plot to omit over-plotting

(from meteorological data) and a back-link (from Boston data) component.

This brush can be refined and that will affect the purple implicit brush in the Boston data set (via forward-linking) as well as the original composite brush in the meteorological data set (via back-linking). The brush details views and the consistent coloring help users to keep track of what is going on. In addition, users can name the brushes so that they reflect the selections. Names such as "Hot Summers" or "Hot Summers and Dry Springs" were often used during our evaluation.

6 Conclusion and future work

The proposed approach provides a well-defined framework for the concurrent analysis of multiple data sets. This approach allows a deep analysis of the hidden interplay between different data sets. We introduce a cross-table link to connect two data sets by relating one or more attributes in the first data table to one or more attributes in the second data table. The underlying formalism is based on relational algebra and the notion of natural join and theta-join operations. Applying an explicit brush in the first data set results in the implicit brush in the second data set. The joint operations are applied only on a subset of a data table.

Several cross-table links linking different data tables are usually created during an analysis session. The brushed data from one data table propagates through several tables via a sequence of links, i.e., a link chain. A cross-table link is also used in the reverse direction (back-link). These back-links allow showing the refinements of implicit brushes in data sets upstream the link chain. Consistent color-coding, icons, and brush track-keeping mechanism help users to identify implicit and explicit brushes. The evaluation was conducted using three data sets.

Future work will include additional metrics and data prefiltering. We expect that use cases in different domains require different metrics. A more formal user study with domain experts from different domains is an important subject of future research. Additionally, we plan to research how links can be efficiently combined, and provide a grammar and a link definition language. Furthermore, we will research how the results of a cross-table analysis can be reported. An efficient, intuitive and user-friendly communication of analysis results and workflows is an area of active research. The inclusion of results from a cross-table analysis certainly represents a great challenge.

Funding This study was funded by BMVIT, BMWFW, Styria, SFG and Vienna Business Agency in the scope of COMET (854174).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Becker, R.A., Cleveland, W.S.: Brushing scatterplots. Technometrics **29**(2), 127–142 (1987)
2. Card, S.K., Mackinlay, J., Shneiderman, B. (eds.): Readings in Information Visualization: Using Vision to Think. Interactive Technologies. Morgan Kaufmann, San Francisco (1999)
3. Cavalcanti, V.M.B., Schiel, U., de Souza Baptista, C.: Querying spatio-temporal databases using a visual environment. In: Proceedings of the Working Conference on Advanced Visual Interfaces, pp. 412–419. ACM, New York (2006)
4. Cerullo, C., Porta, M.: A system for database visual querying and query visualization: Complementing text and graphics to increase expressiveness. In: Proceedings of the 18th International Workshop on Database and Expert Systems Applications (DEXA 2007), pp. 109–113 (2007)
5. Codd, E.F.: A relational model of data for large shared data banks. Commun. ACM **13**(6), 377–387 (1970)
6. Date, C.J., Darwen, H.: A Guide to SQL Standard, 4th edn. Addison-Wesley Professional, Boston (1997)
7. Della Penna, G., Magazzeni, D., Orefice, S.: A general theory of spatial relations to support a graphical tool for visual information extraction. J. Vis. Lang. Comput. **24**(2), 71–87 (2013)
8. Doleisch, H., Gasser, M., Hauser, H.: Interactive feature specification for focus+context visualization of complex simulation data. In: G.P. Bonneau, S. Hahmann, C.D. Hansen (eds.) Proceedings of the Joint EUROGRAPHICS—IEEE TCVG Symposium on Visualization, pp. 239–248. The Eurographics Association (2003)
9. Fisherkeller, M.A., Friedman, J.H., Tukey, J.W.: PRIM-9: an interactive multidimensional data display and analysis system (1974). In: W.S. Cleveland, M.E. McGill (eds.) Dynamic Graphics for Statistics, chap. 3, pp. 91–110. CRC Press (1988)
10. Ganuza, M.L., Ferracutti, G., Gargiulo, M.F., Castro, S.M., Bjerg, E., Gröller, E., Matković, K.: The spinel explorer—interactive visual analysis of spinel group minerals. IEEE Trans. Vis. Comput. Graph. **20**(12), 1913–1922 (2014)
11. Garcia-Molina, H., Ullman, J.D., Widom, J.: Database Systems: The Complete Book, second edn, p. 07458. Prentice Hall, Upper Saddle River (2009)
12. Hauser, H., Ledermann, F., Doleisch, H.: Angular brushing of extended parallel coordinates. In: Proceedings of the 2002 IEEE Symposium on Information Visualization (INFOVIS 2002), pp. 127–130 (2002)
13. Konyha, Z., Matković, K., Gračanin, D., Jelović, M., Hauser, H.: Interactive visual analysis of families of function graphs. IEEE Trans. Vis. Comput. Graph. **12**(6), 1373–1385 (2006)
14. Kosara, R., Hauser, H., Gresh, D.L.: An interaction view on information visualization. In: Proceedings of the Eurographics State-of-the-Art 2003 (EG 2003), pp. 123–137 (2003)
15. Lichman, M.: UCI machine learning repository. http://archive.ics.uci.edu/ml. University of California, Irvine, School of Information and Computer Sciences (2013). Accessed 21 Jan 2018
16. Lin, S., Fortuna, J., Kulkarni, C., Stone, M., Heer, J.: Selecting semantically-resonant colors for data visualization. In: Proceedings of the 15th Eurographics Conference on Visualization, pp. 401–410. The Eurographics Association, Chichester, UK (2013)
17. Liu, Z., Navathe, S.B., Stasko, J.T.: Network-based visual analysis of tabular data. In: Proceedings of the 2011 IEEE Conference on Visual Analytics Science and Technology (VAST), pp. 41–50 (2011)
18. Mahalanobis, P.C.: On the generalised distance in statistics. Proc. Natl. Inst. Sci. India **II**(1), 49–55 (1936)
19. Martin, A.R., Ward, M.O.: High dimensional brushing for interactive exploration of multivariate data. In: Proceedings of the IEEE Conference on Visualization (Visualization'95), pp. 271–278 (1995)
20. Matkovic, K., Freiler, W., Gracanin, D., Hauser, H.: Comvis: a coordinated multiple views system for prototyping new visualization technology. In: Information Visualisation, 2008. IV'08. 12th International Conference, pp. 215–220. IEEE (2008)
21. McLachlan, G.J.: Mahalanobis distance. Resonance **4**(6), 20–26 (1999)
22. NOAA: land-based datasets and products. http://www.ncdc.noaa.gov/data-access/land-based-station-data/land-based-datasets. National Centers for Environmental Information (2018). Accessed 21 Jan 2018
23. North, C., Shneiderman, B.: Snap-together visualization: a user interface for coordinating visualizations via relational schemata. In: Proceedings of the Working Conference on Advanced Visual Interfaces, pp. 128–135. ACM, New York (2000)
24. Pace, R.K., Barry, R.: Sparse spatial autoregressions. Stat. Probab. Lett. **33**(3), 291–297 (1997)
25. Radoš, S., Splechtna, R., Matković, K., Đuras, M., Gröller, E., Hauser, H.: Towards quantitative visual analytics with structured brushing and linked statistics. Comput. Graph. Forum **35**(3), 251–260 (2016)
26. Shadoan, R., Weaver, C.: Visual analysis of higher-order conjunctive relationships in multidimensional data using a hypergraph query system. IEEE Trans. Vis. Comput. Graph. **19**(12), 2070–2079 (2013)
27. Splechtna, R., Matković, K., Gračanin, D., Jelović, M., Hauser, H.: Interactive visual steering of hierarchical simulation ensembles. In: Proceedings of the IEEE Conference on Visual Analytics Science and Technology (IEEE VAST 2015), pp. 89–96 (2015)
28. Turkay, C., Filzmoser, P., Hauser, H.: Brushing dimensions—a dual visual analysis model for high-dimensional data. IEEE Trans. Vis. Comput. Graph. **17**(12), 2591–2599 (2011)
29. Vincenty, T.: Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. Surv. Rev. **23**(176), 88–93 (1975)
30. Weaver, C.: Building highly-coordinated visualizations in improvise. In: Proceedings of the 2004 IEEE Symposium on Information Visualization, pp. 159–166 (2004)
31. Weaver, C.: Cross-filtered views for multidimensional visual analysis. IEEE Trans. Vis. Comput. Graph. **16**(2), 192–204 (2010)

Rainer Splechtna graduated in computer science at Vienna University of Technology in 2003. He is currently working as researcher at VRVis Research Center in Vienna. His research interests include virtual and augmented reality, scientific visualization, information visualization, and visual analytics with special focus on interaction techniques and visualization and analysis of multivariate time-dependent data.

Michael Beham studied Visual Computing at Vienna University of Technology. He graduated in 2015, and is currently working as researcher at VRVis Research Center in Vienna. His research focuses on information visualization and visual analytics with special focus on integrating different types of data, and interaction techniques.

Denis Gračanin received the BS and MS degrees in electrical engineering from the University of Zagreb, Croatia, in 1985 and 1988, respectively, and the MS and Ph.D. degrees in computer science from the University of Louisiana at Lafayette in 1992 and 1994, respectively. He is an Associate Professor in the Department of Computer Science at Virginia Tech. His research interests include virtual reality and distributed simulation. He is a senior member of ACM and IEEE and a member of AAAI, APS, ASEE and SIAM.

María Luján Ganuza graduated as Computer Science Engineer at Universidad Nacional del Sur (Bahía Blanca, Argentina) in 2006. She is currently working as a researcher at VyGLab Research Center in Bahía Blanca, Argentina. Her research interests include visualization of spatio-temporal data, scientific visualization and information visualization with special focus on interaction techniques. She teaches at Universidad Nacional del Sur, where she received her doctoral degree in 2018.

Katja Bühler graduated in Mathematics at University of Karlsruhe in 1996 and received a Ph.D. in Technical Sciences (Computer Science) from Vienna University of Technology in 2001. She continued her career in 2002 as senior researcher at VRVis and as external lecturer at Vienna University of Technology. In 2003, she became head of the Biomedical Image Informatics Group at VRVis and coordinates since 2010 the Research Area Complex Systems. She serves as chair, IPC member and reviewer for various international conferences and journals and is active member of international societies.

Igor Sunday Pandžić is a Professor at the Department of Telecommunications, Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. He leads the Human-Oriented Technologies Laboratory (HOTLab). He teaches undergraduate and postgraduate courses in the fields of virtual environments and communications. His main research interests are in the field of computer graphics and, more recently, computer vision, with particular interest in face analysis and animation and strong focus on applications of these technologies. He published five books and around 100 papers on these and related topics.

Krešimir Matković is a senior researcher at VRVis Research Center in Vienna. He is interested in extending visual analysis technology to challenging heterogeneous data, in particular to a combination of multi-variate data and more complex data types, such as functions, e.g. He also focuses his research on developing a structured model for visual analysis which supports a synergetic combination of user interaction and computational analysis. He teaches at TU Vienna where he received his doctoral degree and habilitation (in 1998 and 2015) and at University of Zagreb, where he received his graduate degree in 1994. He is a member of ACM, Eurographics, and IEEE Computer Society.