CAVIR: CORRESPONDENCE ANALYSIS IN VIRTUAL REALITY
WAYS TO A VALID INTERPRETATION OF CORRESPONDENCE ANALYSIS POINT CLOUDS IN VIRTUAL ENVIRONMENTS

[a]FREDERIK GRAFF, [b]ANDREA BÖNSCH, [c]DANIEL BÜNDGENS, [d]TORSTEN KUHLEN

[a]*RWTH School of Business & Economics, Templergraben 64, 52056 Aachen, Germany;* [b,c,d]*Virtual Reality Group RWTH Aachen University, Seffenter Weg 23, 52074 Aachen, Germany*

email: [a]*Frederik.Graff@org.rwth-aachen.de,* [b]*boensch@vr.rwth-aachen.de,* [c]*buendgens@vr.rwth-aachen.de,* [d]*kuhlen@vr.rwth-aachen.de*

The approach described in this article was funded by the Interdisciplinary Management Practice (IMP) Pathfinder Funding.

Abstract: Correspondence Analysis (CA) is frequently used to interpret correlations between categorical variables in the area of market research. To do so, coherences of variables are converted to a three-dimensional point cloud and plotted as three different 2D-mappings. The major challenge is to correctly interpret these plottings. Due to a missing axis, distances can easily be under- or overestimated. This can lead to a misinterpretation of data and thus to faulty conclusions. To address this problem we present CAVIR, an approach for CA in Virtual Reality. It supports users with a virtual three-dimensional representation of the point cloud and different tools for analysis and clustering. Besides, the free rotation of the entire point cloud enables the CA user to always have a correct view of the data.

Keywords: interaction techniques, user interfaces, exploratory data analysis, correspondence analysis, virtual reality, clustering, market research

1 Introduction

Correspondence Analysis (CA) is a descriptive dimensionality reduction method of multivariable statistics which allows a vivid graphical representation of complex correlations of two (or more) categorical variables as a point cloud in a (theoretically) three- (or more) dimensional space. Similarities between variables are converted into distances on three or more dimensions, and the positions of the variables, represented by points, are converted into coordinates. It is presumed that there is a dependency between rows and columns which can be explained by latent variables. Later on, the variables will be presented as a three-dimensional point cloud. The three axes can then intuitively be interpreted as these latent variables. CA is widely used in social science, psychology, medicine, and in the area of market research (Backhaus et al. 2006).

1.1 Key Concepts of Correspondence Analysis

The key concepts of CA, which are necessary for understanding the functionality and terminology of CAVIR, are briefly introduced[1]. The database consists of frequencies n_{ij} in a contingency table K with I rows R_i and J columns C_j (Table 1).

With $N = \sum_i \sum_j n_{ij}$ and $p_{ij} = \frac{n_{ij}}{N}$, $P = \begin{pmatrix} p_{11} & \cdots & p_{1J} \\ \vdots & \ddots & \vdots \\ p_{I1} & \cdots & p_{IJ} \end{pmatrix}$ is the matrix of relative frequencies.

Table 1: A *IxJ*-contingency table is the dataset the correspondence analysis is based upon.

	C_1	C_2	…	C_J	$\sum$
R_1	n_{11}	n_{12}	…	n_{1J}	r_1
R_2	n_{21}	n_{22}	…	n_{2J}	r_2
⋮			…		⋮
R_I	n_{I1}	n_{I2}	…	n_{IJ}	r_I
$\sum$	c_1	c_2	…	c_J	1

We define $r_i = \sum_{j=1}^{J} p_{ij}$ and $c_j = \sum_{i=1}^{I} p_{ij}$, as the masses of rows and columns. The frequencies in the cells are normalized with r_i and c_j respectively. As a result we get two matrices

$$R = \begin{pmatrix} \frac{p_{11}}{r_1} & \cdots & \frac{p_{1J}}{r_1} \\ \vdots & \ddots & \vdots \\ \frac{p_{I1}}{r_I} & \cdots & \frac{p_{IJ}}{r_I} \end{pmatrix} \text{ and } C = \begin{pmatrix} \frac{p_{11}}{c_1} & \cdots & \frac{p_{1J}}{c_J} \\ \vdots & \ddots & \vdots \\ \frac{p_{I1}}{c_1} & \cdots & \frac{p_{IJ}}{c_J} \end{pmatrix}.$$

Each row of R is called the i^{th} row profile of K. Each column of C is called the j^{th} column profile of K (Mortensen 2011). The columns, interpreted as J axes, put up a J-dimensional space C_J, which the row profiles should be plotted in, analogously the rows. Obviously, the J-/I-dimensionality causes problems, since plotting more than three-dimensional spaces makes an understanding of the mapping more difficult. Due to this, the number of dimensions of C_J and R_I is reduced to three (C^3, R^3) and both are integrated into one coordinate system, whereby three conditions have to hold:

1. The more similar the profiles are, the closer the points representing the respective row/column should lie to each other in the coordinate system. The distance is measured using the Euclidean distance.
2. The reduction of the overall variance of K overall p_{ij}, the so-called Total Inertia T, should be minimal, while T is given as $T = \sum_{i=1}^{I} \sum_{j=1}^{J} \frac{(p_{ij} - r_i \times c_j)}{r_i \times c_j}$.
3. The additive decomposition of T should be maximal (Mortensen 2011).

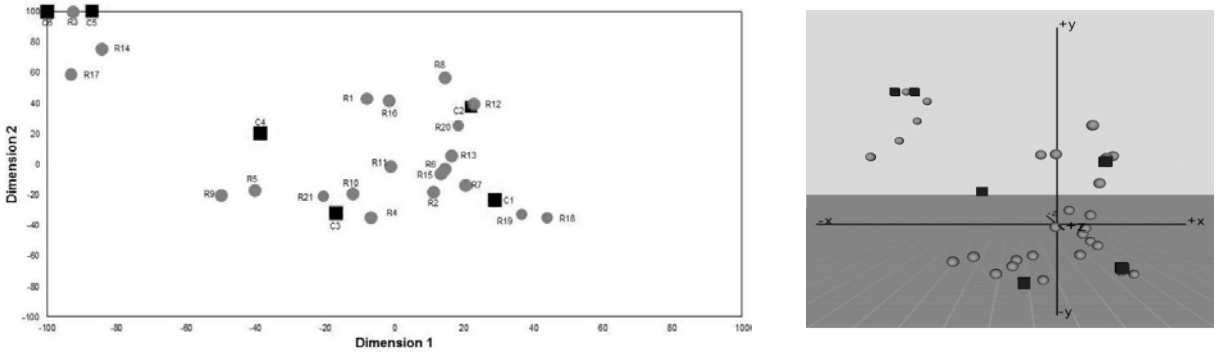
After further calculations we get *XYZ*-coordinates for each row and column, so points can easily be plotted (cf. fig. 1(a)). The axes of the integrated space can be interpreted as latent variables that explain the variance of the conditional frequencies.

1.2 Challenges of Plotting CA Point Clouds

The reason why CA is so widely used in social science and market research lies, i.a., in the relatively convenient way complex correlations between categorical variables can be intuitively interpreted using "mappings" (cf. fig. 1(a)). With a mapping, the researcher interprets the relative distances of the points from each other and their relative positions on the axes. Most often, one column point (■ in fig. 1(a)) makes up more or less the center of a sub-cloud consisting of several row points (● in fig. 1(a)). In market research, the column points usually represent variations of a product, brands, or firms, and the row points represent features, ratings, or (potential) purchasers (Backhaus et al. 2006). Column points will henceforth be referred to as "trait carriers", and row points as "traits". Sometimes, researchers use the mapping to cluster points graphically.

For a complete plotting of results, a coordinate system with three axes is required. This poses some difficulties, as a valid graphical representation may ultimately be two dimensional. Plotting software solutions (e.g. SPSS, mapwise, XGobi) therefore usually offer the option of plotting three different mappings, where one of the three axes is omitted. Hereby, both an *XY*-, an *XZ*- as well as a *YZ*-coordinate system are produced (Swayne et al. 1998), (Nenadic and Greenacre 2007). For the interpretation of results this may cause problems because only two axes can be considered simultaneously. This is why practitioners often choose the axes with the highest explained variance and ignore the third one and thereby neglect important distance information.

[1] For detailed information, see (Mortensen 2011), (Benzécri 1977) or (Backhaus et al. 2006); for a short introduction, see (Nenadic and Greenacre 2007).



(a) (b)

Figure 1: Comparison of the CA point cloud representation in a two-axis coordinate system with a simple 2D-mapping (layout similar to SPSS) (a) and in a three-axis coordinate system with our VR-based approach (b).

Figure 2 shows an example of how a missing dimension can cause invalidities in the graphical representation. In the tables 2 and 3 the respective coordinates and mean Euclidean distances are given. It becomes obvious that the distance between point F and point 5 is massively overestimated if only the axes *X* and *Y* are taken into account. The distance is reduced from 30,9 to 21,4 when axis *Z* is included into the calculation of the (mean) Euclidean distance. Analogously the distance between point C and point 2 is heavily underestimated in the 2D-mapping, because the mean Euclidean distance increases from 14,8 up to 56,8 when the *Z*-axis is included into the calculation.
This invalid representation of distances in the 2D-view can easily lead to a faulty clustering of data, which is dangerous. The aim of CA is (as discussed above) to show graphically which row points (traits) are associated with which column point (trait carrier) in the perception of potential customers. Faulty clustering so can lead to wrong interpretations of these associations and foster invalid conclusions in the manner that e.g. a point is included into one cluster but truly belongs to another one. In the worst case such errors can e.g. let a PR-campaign fail because traits of one product are stressed though they are – according to the customers perception – not in the least associated with this product.

Table 2: *XYZ*-coordinates of example points in figure 2

	Coordinates		
Point	**X**	**Y**	**Z**
F	45,9	8,3	-42,3
C	-24,8	6,2	100,0
5	6,0	-38,8	-59,6
2	-0,2	-10,4	-67,7

Table 3: Mean Euclidean distances of example points in figure 2

	Mean Euclidean Distance	
Points	**XY**	**XYZ**
F, 5	30,9	21,4
C, 2	14,8	56,8
C, 5	27,3	56,2

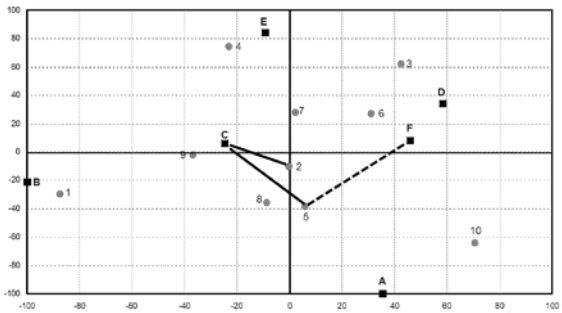


Figure 2: Underestimation (**—**) and overestimation (**- -**) of point distances in a 2D-mapping. Z-Axis not plotted. (Variance explained: x-Axis = .46 / y-Axis = .31 / **z-Axis = .22**)

To sum up, the main problems with non-three-dimensional plottings of a CA point cloud are:
1. Overestimation of distances
2. Underestimation of distances
3. Faulty clustering
4. Faulty interpretation

Several attempts to take account of all the three axes graphically have led to unsatisfactory results. Especially, the graphical summary of points into clusters is impossible when distance information is not realized validly as distances but rather as perspectives or as point characteristics.

1.3 Approach

In the work presented, the authors have developed CAVIR, a tool that enables the graphical analysis and interpretation of CA point clouds in a virtual environment (VE), providing a valid distortion-free impression of the spatial distances between points. To meet the challenges mentioned above, the key approaches to a solution are:
5. A virtual three-dimensional model of the CA point cloud
6. Free navigation through the VE
7. Arbitrary rotation opportunities of the point cloud to provide a valid impression of the spatial distances between points
8. Diverse options to display labels
9. Unhindered scaling of axes
10. Measuring of distances between points
11. Clustering of points

2 Related Work

Van Dam et al. already stated that the gap between the rate of data generation and the capacities to analyze this data is widening. As a possible solution they propose Immersive Virtual Reality (IVR) technology, which combines interactive visualization with immersive sensation (van Dam et al. 2000). Arns et al. integrated 3D-scatterplots into a CAVE-like VE (Arns et al. 1999). The authors performed a user study which contrasted their methods with the well-known XGobi system (Swayne et al. 1998). Interestingly, users performed better with the VE but felt more comfortable when working with the classical desktop-based tool.
Another approach that uses a “grand tour” view embedded in a VE to show raw data as well as clustering results is presented by Yang (Yang 1999). The same author specifically addresses the problem of overdrawing, and presents a solution based on direct volume visualization, namely a splatting approach for scatterplots and similar views (Yang 2003).
Very little research has yet been done on alternative ways of plotting the point cloud of a CA. Backhaus et al. give an introduction in the method itself and its mathematical procedures, pointing out that if more than two axes are needed to achieve a satisfactory explained variance, the researcher has to decide which two are the most important. These should be plotted. However, the authors admit that a two-axes-plotting is never possible without a loss of information (Backhaus et al. 2006). Whitlark and Smith propose chi-square residuals to

measure item distances (so-called attribute-brand relationships) and to thus escape the dimensionality problem. Previously they had stated: “Relying on a two-dimensional map may be risky. In our experience, it is rare to see a two-dimensional map tell a complete or even an accurate story” (Whitlark and Smith 2001).
The statistical software package R, as described by Nenadic et al., provides a non-immersive three-dimensional display of CA point clouds, which alleviates (but does not totally solve) the problem of perceptual distortion. Helpful analytical tools, such as brushing, labeling, distance measuring, and clustering, are not included in R (Nenadic and Greenacre 2007).

The first application of Virtual Reality (VR) on CA was done by Monmarché et al. In the context of a dermatological study, they plotted point clouds along three dimensions of skin characteristics using a stereoscopic display. Point clustering was ex ante implemented by a hierarchical cluster analysis (Monmarché et al. 2002). However, we want to enable the researcher to cluster the points manually within the model. In contrast to existing CA analyzing and visualizing tools, the approach presented here proposes an immersive 3D-display providing the researcher with a distortion-free impression of spatial distances as well as with new interaction methods for valid interpretations and clustering.

3 Technical Realization

CA is a complex process divided into several substeps, but in this paper we focus on one of the last steps, namely clustering and the supportive techniques. Due to the fact that the approach should be embedded into the workflow of CA experts, the format of our input and output files is determined by the notation of SPSS (Akin et al. 2009), one of the standard programs in the field of CA. This enables a quick and uncomplicated switch between both working areas.

Our initial situation is an input file containing a list of data points described by three attributes: one character **cMemshp** classifying the data point, one string **sDesc** giving a concrete point label and three numerical values defining the position of a point **P** in the 3D space.
The data points are divided into two sets, the traits **A_Trait** and the trait carriers **B_Carrier**. The data points’ membership of one of these two sets is encoded in *cMemshp*. A unique natural number indicates the inclusion in *A_Trait*; unique small letters, accordingly, in *B_Carrier*. To be able to cluster the data points meaningfully, users need to be able to easily distinguish both sets during exploration in the VE. For this purpose, different shapes, analogously to the representations in standard 2D-CA-programs, are used: a sphere is chosen for the traits and a box for the trait carriers (cf. fig. 1(b)). With this similarity, irritations on the user side will be avoided. Then, the representations are shown at the respective point *P*. A Cartesian coordinate system used as a reference frame (see 3.3) completes the basic set-up.
Due to the fact that the point cloud is relatively small compared to the complete VE, we added a tiled reference plane as a “ground floor” to give some spatial cues. This supports the user during exploration of the scene via navigation (Bowman et al. 2004). Here, we offer two techniques: a pointing-metaphor for free travel through the VE and a trackball metaphor for free rotation of the point cloud.

3.1 Interaction

In addition to navigation, different interaction techniques are offered. We scale the represented point clouds to enable an accurate selection with both techniques of a single point, even in tight clouds. To give some additional help by all selection tasks, the position of the input device is displayed by either a beam or a small sphere, as will be explained in section 4.
To enable an intuitive and well-defined handling, each technique provided is bound to one application mode. A menu hierarchy is used to for unhindered switching between the modes and their submodes during runtime. We choose a so-called *pie menu* (Callahan et al. 1988).

Besides the layout, we try to achieve an effective working by using just a small menu hierarchy. The first menu consists of four entries, showing the four available modes, presented in the following. Each entry has a submenu for mode specific tasks. To always indicate which mode is currently active, we show a tooltip that provides all the necessary information. Additionally, a color coding of the floor tile borders and the scene background is used. These colors are also linked to the respective circular menu’s entries to quickly ensure a correct handling.
Besides the different shapes, we add a color coding for an easy and quick identification of the points’ membership. Traits are visualized in red and trait carriers in green. This color association is reused as often as possible in the modes.

3.2 Labeling

Besides the general set-up, additional information is required for an effective clustering process. We use textual annotations, as shown in figure 3(a), which can be faded in or out for each point individually in the “label” mode. For simplification, we use the red-green coloring here, not only for the points but also for the labels.
We provide three label types, given in the initial input file, which are used to support the user during clustering.

1. The unique id *cMemshp* allows an explicit and a fast identification of data points.
2. Sometimes it may be useful to know what information a point represents, i.e. which product or key features it stands for. This information is contained in the attribute *sDesc*.
3. The explicit point position is also helpful during clustering to estimate point distances without using the distance mode, or to find outliers.

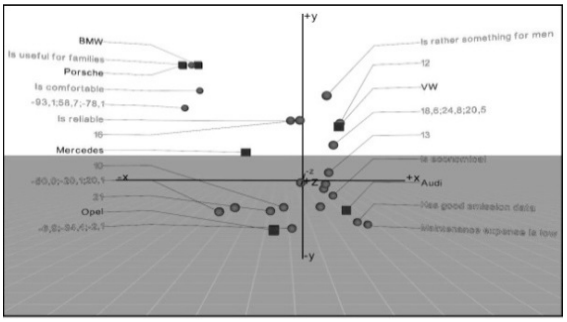
Users can switch between those label types via different entries in the circular menu.

3.3 Coordinate System

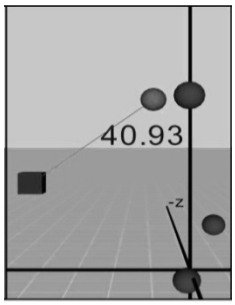
To quickly understand the distribution of the point cloud, we display a Cartesian coordinate system with labeled axes. Due to this, the origin of the point cloud can be clearly identified and this allows a first, rough classification of the distribution. To allow extraction of more information, the length of the individual positive and negative axis intercepts can be set to one of three predefined values using the mode “coordinate system”.

- To quickly understand the spatial distribution, the axes can be adapted to the *maximal absolute values* of the point cloud extended by a user specified value. Short axes imply small point distances, while large axes indicates an extensive distribution or outliers.
- In CA, points are often deemed to be outliers if they have at least one absolute value greater than a maximal bound. According to the concrete application, those values may be treated differently, e.g. they may be clamped to a special value. In our approach a maximal bound can be predefined by the user to enable the identification of outliers.
- In addition to the maximal bound, a minimal bound can be set. This value will be predefined by the user and used to quickly identify points in or over a special bound.

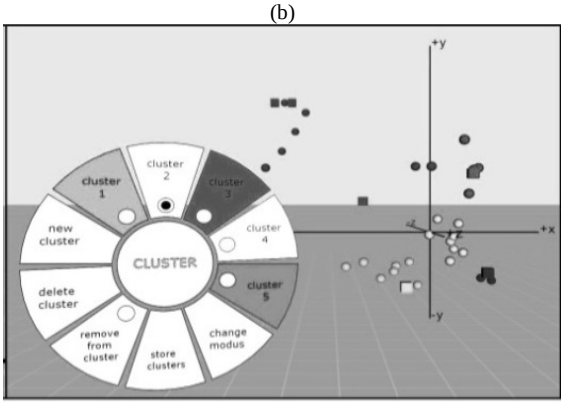
Besides scaling the axes, users can add an object-oriented bounding box to identify inliers and outliers more clearly.



(a)



(b)



(c)

Figure 3: (a) The label layout consists of two columns next to the point cloud and different labels are offered, which can be selected individually for each data point. (b) A measured distance is visualized via a connecting line and a centered text label. (c) One example to divide a given point cloud into five clusters, currently adding points to cluster 2.

3.4 Distance Measurement

The main aim of our approach is to facilitate clustering and to reduce errors during this process. One criterion used to find the correct cluster for a data point is comparing the distances between several points. This can be done in the mode "distance". Users are asked to select two different data points P_i and P_j, between which the Euclidean distance will be computed. The Euclidean distance itself is calculated based on the exact coordinates given in the input file.
Figure 3(b) illustrates the distance visualization: a connection line between both selected points, and a centered label with the computed distance. To maximize the clarity of which points are selected, all points are inked in gray in the beginning. After selecting a data point, it is recolored to red or green according to the set it belongs to.
Users can select a marked point again to deselect it, or just choose a third point P_k. In this case, the system will automatically deselect the first two points P_i and P_j. Owing to this, just one distance is shown at the same time.

3.5 Clustering

In the "clustering" mode, the whole interaction is managed by the circular submenu, shown in figure 3(c). Thereby, the compliance of two rules is always ensured:
R1: The total number of clusters is greater than zero and less or equal to |B carrier|.
R2: In each cluster there has to be at least one trait carrier.

Three interaction types are offered in this mode:
a) Users can add and delete clusters, complying with R1. For each cluster an individual menu entry is provided, containing a radio button.
b) Via brushing (Wright and Roberts 2005), points can be linked to an active cluster according to R2 or they can be removed from any cluster.
c) The clustering can be stored to the file system. The format of the output file is determined as dummy codes, to enable the analyst to go on working in her used surroundings after using the VE to cluster the point cloud.

If the user selects a cluster entry in the circular menu, linking of data points to the according cluster is activated. Adding points to the cluster is done by brushing: first focus a point, and then add it by clicking. To be able to comply to R2, the first point added to a new cluster has to be a trait carrier. We again use colors to show the set membership. All points that are not yet linked to a cluster are gray. Each cluster has a unique color, which is used for the cluster entry in the circular menu and the linked points. Traits that shall be linked to several clusters will be marked multicolored. Additionally, cluster membership is coded by a tube-like respectively colored silhouette. If the user adds a point to a wrong cluster, the circular menu provides an entry for deleting individual points from clusters. With this option, the points are recolored gray. If a single trait carrier in a cluster is deselected, all points of this cluster are deselected, too. With this automatic deselection, R2 is always ensured to hold.

4 Challenges of Different VR Displays

The aim of CAVIR is to accelerate the process of interpreting and clustering and thereby reduce the potential clustering errors described in section 1. Arns et al. (1999) already showed in a user study that VEs support this requirement, because a third dimension is available. Users are able to make better decisions, due to a better and not misleading view of the data (Whitlark and Smith 2001).

(a)

(b)

Figure 4: (a) Using CAVIR in a five-sided CAVE-like environment (here: aixCAVE at RWTH Aachen University) and (b) on a 3D monitor without head-tracking.

Our approach should be available for two different display system classes. On the one hand, we want to use a five-sided CAVE-like environment to enable collaboration with several people during the cluster process and the possibility to freely walk around the point cloud (fig. 4 (a)). On the other hand, we want to provide a semi-immersive low-cost and quick possibility to cluster the point cloud in the office in order to embed its use

into the normal workflow. Therefore, we decided to use a 3D monitor, as shown in figure 4(b).
The low-cost solution has multiple advantages. First, it is easy to handle for novice users, who are often not familiar with VR techniques. Second, all the input devices known from standard desktop systems can be used and thus the interaction can be maintained similarly to the usual work environment. Due to this, the learning phase before users can handle the application correctly is shortened.

5 Discussion and Future Work

We have presented a VR-based technique to ease valid interpretation and to realize clustering - one of the last steps in correspondence analysis (CA). The standard CA programs mostly provide 2D-views of the data, leading easily to misclustering and misinterpretations None offers a clustering option. In the virtual environment (VE) we have the advantage of showing the data three-dimensionally as a point cloud in a Cartesian coordinate system. Thus, a correct view is always provided and perceptual distortion avoided, which accelerates the clustering process and at the same time reduces the risk of invalid clustering and misinterpretation. Different navigation and interaction tools are used to support the clustering, which is done by linking points to certain clusters via brushing.
During the implementation process we had close contact to one CA expert, who pretested all features. However, we plan to evaluate our approach in a user study to identify issues which need to be addressed for improvement, especially concerning usability.

Literature:

1. Akin, R., Shape, R., de Veaux, R., Velleman, R.: *SPSS Manual for Business Statistics.* Pearson Education, 2009. ISBN 03-21571-36-3.
2. Arns, L., Cook, D., Cruz-Neira, C.: *The benefits of statistical visualization in an immersive environment*. Proceedings of IEEE VR, 88–95, 1999.
3. Backhaus, K., Erichson, B., Plinke, W., Weiber, R.: *Multivariate Analysemethoden. Eine anwendungsorientierte Einführung*, 11th ed. Berlin, Heidelberg: Springer, 2006. ISBN 35-40278-70-2.
4. Benzécri, J.-P.: *Histoire et préhistoire de l'analyse des données. Partie V : l'analyse des correspondances*. Les cahiers de l'analyse des données, 2(1), 9–40, 1977.http://www.numdam. org/item?Id=CAD_1977__2_1_9_0, (June 11, 2012).
5. Bowman, D. A., Kruijf, E., LaViola Jr., J., Poupyrev, I.: *3D user interfaces. Theory and practice*, 1st ed. Boston: Addison-Wesley, 2004. ISBN 0-201-75867-9.
6. Callahan, J., Hopkins, D., Weiser, M., Shneiderman, B.: *An Empirical Comparison of Pie vs. Linear Menus*. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 95–100, 1988.
7. Monmarché, N., Marteau, H., Gérard, J.-P., Guinot, C., Venturini, G.: *Interactive mining of multimedia databases with virtual reality*. Z. Pan (ed.) Proceedings of the Third International Conference on Virtual Reality, 478–484, 2002.
8. Mortensen, U.: *Einführung in die Korrespondenzanalyse*, http://uwemortensen.de/caneu1c.pdf, 2011, (June 11, 2012).
9. Nenadic, O., Greenacre, M.: *Correspondence Analysis in R, with Two- and Three-dimensional Graphics: The ca Package*. Journal of Statistical Software, 20(3), 1–13, 2007.
10. Swayne, D. F., Cook, D., Buja, A.: *XGobi: Interactive Dynamic Data Visualization in the X Window System.* Journal of Computational and Graphical Statistics, 7, 113–130, 1998.
11. van Dam, A., Forsberg, A., D. Laidlaw, LaViola Jr., J., Simpson, R.: *Immersive VR for scientific visualization: a progress report.* IEEE Computer Graphics and Applications, 20(6), 26–52, 2000.
12. Whitlark, D. B., Smith, S. M.: *Using Correspondence Analysis to Map Relationships*. Marketing Research, 13(3), 22–27, 2001.
13. Wright, M. A. E., Roberts, J. C.: *Click and Brush: A Novel Way of Finding Correlations and Relationships in Visualizations*. L. M. Lever, M. McDerby (eds.) EG UK Theory and Practice of Computer Graphics, 179–186, 2005.
14. Yang, L.: *3D Grand Tour for Multidimensional Data and Clusters*. D. J. Hand, J. N. Kok, M. R. Berthold (eds.) Proceedings of the Third International Symposium on Advances in Intelligent Data Analysis, 173–184, 1999.
15. Yang, L.: *Visual Exploration of Large Relational Data Sets Through 3d Projections and Footprint Splatting*. IEEE Transactions on Knowledge & Data Engineering, 15(6), 1460–1471, 2003.

Primary Paper Section: A

Secondary Paper Section: AE, AH