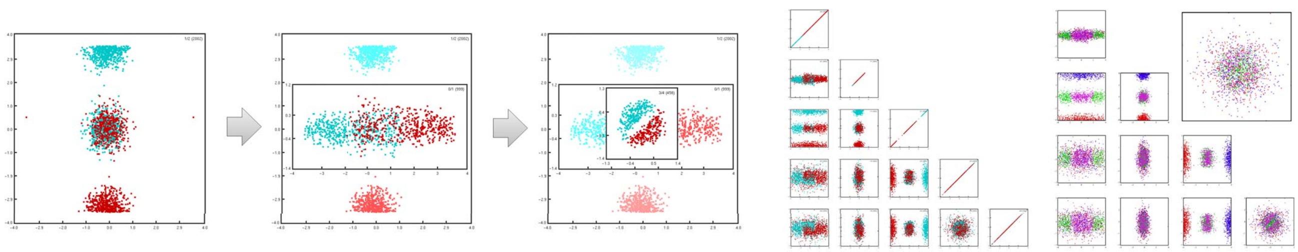
A Nested Hierarchy of Localized Scatterplots

Martin Eisemann, Georgia Albuquerque, Marcus Magnor
TU Braunschweig
Braunschweig, Germany
Email: {eisemann,georgia,magnor}@cg.cs.tu-bs.de



(a) (b) (c) (d) (e)

Fig. 1. **Example of a nested Hierarchy of Localized Scatterplots (HLSP).** (a) – (c) The user refines the visualization using our HLSP by inserting subplots revealing a local view on subsets of the data to separate the two classes (red/cyan). The HLSP defines a decision tree that can easily be derived even by inexperienced users directly from the visualization. (d) A Scatterplot Matrix (SPLOM) representation is insufficient to reveal any information on how to separate the classes. (e) Imitating the result of the HLSP with traditional brushing (using another tool) is insufficient and misses the clarity of the HLSP. The separating dimension is shown in the enlarged inset on the top right, the green and magenta data items correspond to the middle and most inner subplot in (c), respectively. The trivial plots along the diagonal of the SPLOM have been omitted.

***Abstract*—The simplicity and visual clarity of scatterplots makes them one of the most widely-used visualization techniques for multivariate data. In complex data sets the important information can be hidden in subsets of the data, often obscured in the typical projections of the whole dataset. This paper presents a new interactive method to explore spatially distinct subsets of a dataset within a given projection. Precisely, we introduce *a hierarchy of localized scatterplots* as a novel visualization technique that allows to create scatterplots within scatterplots. The resulting visualization bears additional information that would otherwise be hidden within the data. To aid the useful interactive creation of such a hierarchy of localized scatterplots by a user we display transitions between scatterplots as animated rotations in 3D. We show the applicability of our visualization and exploration technique for different tasks, including cluster detection, classification, and comparative analyses. Additionally, we introduce a new exploration tool which we call the *cross-dimensional semantic lens*. Our hierarchy of localized scatterplots preserves the visual clarity and simplicity of scatterplots while providing additional and easily interpretable information about local subsets of the data.**

***Keywords*-Data visualization;**

I. Introduction

The core activity of information visualization research is to develop novel visualization techniques specifically tailored to the characteristics of the investigated data. As in many cases these characteristics are unknown beforehand, scatterplots remain one of the most versatile and commonly used visual data representations. A typical scatterplot is a discrete bivariate visualization of pairwise data dimensions where single data items are represented as points in a cartesian coordinate system. Each of the orthogonal axes spanning this coordinate system is related to one dimension in the dataset, respectively. A third dimension can be added to create 3D scatterplots, but many real-world datasets have more dimensions, which cannot be represented at the same time. Scatterplot matrices (SPLOM) [1] overcome this limitation by plotting a matrix of all orthogonal data projections, giving a supposedly complete overview of the data. Dimensionality reduction techniques such as PCA [2] or MDS [3] are used to reduce a dataset to the, theoretically, information bearing dimensions and can therefore be used to find more expressive projections of a dataset. The problem is that neither a single projection of the whole dataset nor a series of projections, as in SPLOMs, may reveal the information required to correctly reason about the dataset and its characteristics due to occlusion or simply because such a projection does not exist if all data is projected at once. Additionally, with a non-orthogonal projection the coordinate axes loose their original semantic meaning which makes the result harder to interpret. Alternative visualizations for multidimensional datasets – such as parallel coordinates, dense pixel displays, or RadVis – have the benefit of more visual expressiveness at the cost of a loss in simplicity and intuitive understanding compared to standard scatterplots.

In this paper, we want to preserve the simplicity and understandability of standard scatterplots but extend their expressiveness to overcome the aforementioned limitations. We present a novel visualization technique consisting of a hierarchy of localized scatterplots. The common scatterplot technique is extended to contain subplots displaying a localized subset of the original data but with different projection axes. The subplots may recursively contain other subplots to

1530-1834/14 $31.00 © 2014 IEEE
DOI 10.1109/SIBGRAPI.2014.14

create a hierarchy of scatterplots, Fig. 1a–c.

The benefit of such a representation is that the relation to the dimensions displayed in the parent plots along the line of ancestors is not completely lost. The result is a visualization technique that conveys information about the *local*, multidimensional relations in a single scatterplot-like visualization, and allows to reason in more detail about the characteristics of the dataset. Additionally, this kind of visualization is especially beneficial for printing media, as it can convey the important information in a spatially more dense representation than e.g. SPLOMs. As the user is free to create and move the subplots in real-time, we can create a *cross-dimensional semantic lens* which allows to interactively look into a different projection of the datapoints underlying the subplot. Our technique can aid the analyst in a variety of different tasks, including occlusion reasoning, cluster analysis, classification tasks, or comparative analyses.

II. Related Work

In the following we will motivate our hierarchy of localized scatterplots. We need to mention that an exhaustive overview is far beyond the scope of this paper and we, therefore, can only sample the space of related work here.

Multidimensional Visualization: A taxonomy of primarily multidimensional visualization techniques can be found in [4] which proposes categories such as standard 2D/3D displays, geometrically transformed displays, dense pixel displays, stacked displays, and iconic displays. Scatterplots belong to the first category. Due to their simplicity and high visual clarity scatterplots are one of the most powerful and most widely used techniques for visual data exploration [5] and are incorporated into many multidimensional visualization tools such as GGobi, Tableau/Polaris or XmdvTool. Though the number of visualizable dimensions can be increased beyond the typical two by using glyphs integrating point color, shape or size, the spatial position remains the most distinctive and important feature for the data analysis. Scatterplot matrices [1] build upon this insight and arrange a series of scatterplots as a $m \times m$ matrix with the data dimensions on the rows and columns and each cell representing a single scatterplot projection according to the appropriate dimensions. With increasing dimensionality the individual scatterplots degrade to thumbnail like images with little information on the data distribution. The idea of the *grand tour* [6] is that a dataset is fully explored if all possible 2D projections have been seen, ignoring rotations. The practical applicability of this approach, however, is limited due to physical limitations such as display resolution, overplotting and timing constraints. Therefore, the typical 2D projections may reveal little about complex features, such as clusters, hidden within the dataset due to this overplotting. Also, visual ambiguities are not easily solvable within any global view of the dataset due to the down-projection onto two dimensions. The beauty of our approach is that we can solve many of these visual ambiguities by introducing nested scatterplots to refine the projection in the affected areas.

Dimensional Reduction and Multidimensional Projection: The aim of dimensional reduction techniques, sometimes also called multidimensional projections, such as PCA [2] or MDS [3], is to find the most interesting 2D projection planes within a dataset. Unfortunately, it has been shown that PCA is very sensitive to outliers and artifacts in the data [7]. t-SNE [8] is a more robust technique for non-linear embeddings that overcomes many of these drawbacks and nicely clusters “similar” data in the projection. Unfortunately, all dimensional reduction techniques have their own specific notion of “interest” and, therefore, cannot guarantee to find useful 2D projections, nor that such a projection exists. Even more important, non-orthogonal 2D projections of multidimensional data is beyond the understanding of casual information visualization users and can even pose a problem for experts. Our approach is simpler in the sense that we combine (nest) the easily interpretable orthogonal projections to display information beyond that of simple projections which can be a valid alternative to non-linear projections.

Occlusion: The projection of a large multidimensional dataset onto a 2D plane can easily result in cluttered or overplotted views due to limited resolution of the display or discretized data. To remedy this problem, sampling techniques can be used to reduce the amount of overplotting [9]. Distorting the projection space minimizes the degree of overlap [10] at the cost of aggravated interpretability for non-experts.

The mentioned techniques usually aim at optimizing the degree of overlap in the projected space of the scatterplot. Our assumption is that we can remedy the information loss of overplotting by locally changing the projection of the data to reveal interesting structures. If this projection is done in place, as we do in our approach, the rough relation to the original projection of the scatterplot is preserved. An out-of-place refinement for parallel coordinates and some other visualizations was proposed in [11].

Interactive Exploration: User-interaction is a powerful tool for a visual exploration of a multidimensional dataset if little or no information about the dataset is provided. In a visual exploration interactive visualizations are used for an exploratory data analysis [12]. Though automated methods to find suitable projections out of all possible combinations exist [13], [14], [15], they suffer from the fact that they usually require an already specified goal which might not be available. Additionally, these automated methods usually operate on the whole dataset and not on subsets. For this, user-interaction is mandatory.

The original PRIM-9 system [16] allows the user to manipulate and view 3D scatterplots of multidimensional datasets using a combination of projection, rotation followed by isolation and masking. The animation component within the system proved also useful in other applications and techniques [17], [18], as it allows a non-disruptive change between displayed dimensions that allow the user to track data points over time. Especially, if only one dimension at a time is smoothly changed the animated transition is easy to understand, as was shown in the rolling-the-dice tool [17]. Also, nesting visualizations within each other can increase the number of dimensions displayable on a 2D output device [19] or augment additional information into the plot [20]. We build upon the ideas of animation, isolation and nesting and extend it by a local analysis of the dataset within the same view. This allows us to integrate novel analysis tools such as the *cross-dimensional semantic lens* and to create a visually expressive view of the data that provides local insights across more than just two dimensions of the dataset in a single view.

Dynamic queries [21] allow for filtering operations which are integrated in most visualization tools. Query and filtering operations such as high-dimensional brushes [22] support the selection and filtering in data space using visual queries with direct feedback in the visualization itself. These filtering techniques form a backbone of high-dimensional data analysis, often combined in coordinated views showing the same data using different visualization techniques to get the best out of all [23]. The local subplots in our technique are related to the brushing technique as it allows to inspect a subset of the data in more detail. However, whereas with brushing, e.g. in a SPLOM, the visual connection to the selected data in other projections has to be established by the analyst, our approach improves the locality, as the new projection of the data is done in place.

In the following we describe our idea of a hierarchy of localized scatterplots (HLSP) where a standard scatterplot is augmented with localized views into other dimensions of the dataset. We will first give a short problem statement (Sect. III) and lay out the theoretical foundations of a HLSP (Sect. IV). We then describe a prototypical implementation of an exploration tool using our proposed approach (Sect. V). We analyze the applicability of our visualization and exploration scheme (Sect. VI) and, finally, conclude and give an outlook for future work (Sect. VII).

III. Problem Statement

Scatterplots are one of the most widely used techniques for visual data exploration. Given a dataset $\mathbf{D}$ with N data items $\mathbf{d}_i,\ i \in \{1, \ldots, N\}$, each data item consists of a m-dimensional vector $\mathbf{d}_i = (d_{i,1}, d_{i,2}, \ldots, d_{i,m})$. For simplicity, we will assume for the rest of the paper that each attribute $d_{i,j}$ is a real-valued entry. Additionally we will consider both, class-based and non-class-based datasets. In the former case, each data item is additionally assigned to a specific class which is typically color-coded in the visualization. A scatterplot defines a projection of $\mathbf{D}$ onto a lower-dimensional space, typically 2D. As simplicity and interpretability is of highest importance for us, we will concentrate only on orthographic projections here, as other projections are often difficult to understand [17], i.e., in a scatterplot each data item $\mathbf{d}_i = (d_{i,1}, d_{i,2}, \ldots, d_{i,m})$ is projected onto $\mathbf{d}'_i = (s_a \cdot d_{i,a}, s_b \cdot d_{i,b})$ with $a, b \in \{1, \ldots, m\}$ and s_a and s_b being scale factors depending on the display and size of the scatterplot. Therefore, the x-axis in the scatterplot denotes dimension a whereas the y-axis denotes dimension b, Fig. 2a. In a SPLOM all possible orthographic projections of $\mathbf{D}$ are conglomerated in a $m \times m$ matrix where each row and column represents a dimension of the dataset, though it is sufficient to show only the upper or lower half, due to symmetries, Fig. 2b. Unfortunately, orthographic projections of the whole dataset may not reveal all important information hidden within the dataset due to overplotting and occlusion. E.g. in Fig. 2b no 2D view exists that separates the two displayed classes (red and cyan). In contrast, using our HLSP, Fig. 2c we can create a view on the data that clearly separates the classes from each other. We will describe our HSLP in more detail in the following.

IV. Hierarchy of Localized Scatterplots

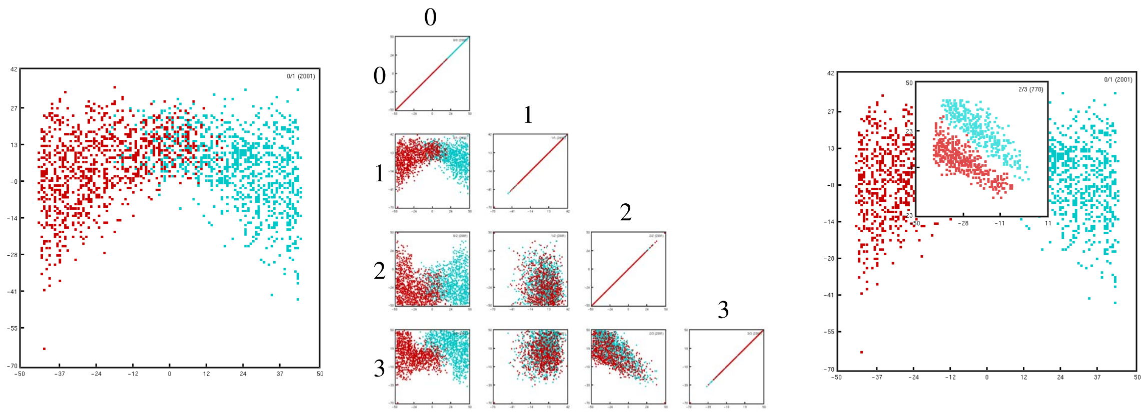
In this section we introduce our idea of a hierarchy of localized scatterplots. We will then present a visualization technique using the concept of a HLSP (Sect. IV-A).

A localized scatterplot (LSP) is similar to a traditional filtered scatterplot that displays only data items that are subject to certain user-defined restrictions. In our case these restrictions are min/max bounds on the data values of the dimensions of the dataset, i.e. rectangular subsets. The difference to filtered or conditioned scatterplots is that the LSP is an in-place augmentation of the scatterplot intending to strengthen the idea of focus and context instead of juxtaposing the selections.

We extend the idea of LSPs to a recursive formulation, introducing a hierarchy of LSPs. A hierarchy of LSPs can be formally described as a tree structure. Each node in this tree represents a LSP and each edge defines an additional restriction in terms of min/max bounds for the displayed data. The displayed dataset $\mathbf{D}'$ for each LSP consists of only those data items that fulfill the restrictions of all ancestors within the tree. The projection axis can be chosen at will for each scatterplot. Data normalization for better utilization of the display is also optional, as with classic scatterplots. Fig. 3 shows an example without data normalization. The defined structure resembles a simplified decision tree on the multidimensional data [24], [25]. Please note that we use the explicit tree structure only for explanatory purposes here and never present it to the user, as the information conveyed therein is already present in the HLSP itself.

A. Visualization

From the definition of HLSPs, we want to derive a visualization technique that helps the user in a visual exploration task to detect important local structures within the data. For this task we transform the inherently defined tree structure of the HLSP into a more suitable representation. If we restrict the min/max bounds at the edges in the tree to change only the dimensions equal to the projection axis of the parent node, then this defines a rectangular region R within the parent scatterplot. The scatterplot of the child node is then displayed within R with optionally changed projection axes. An example of such a visualization with two and three levels respectively is given in Fig. 2c and Fig. 1c. As the subset of hidden data behind each subplot is equal to the displayed data of the subplot, no information loss occurs.



(a) 2D Scatterplot (b) Scatterplot Matrix (SPLOM) (c) Hierarchy of Localized Scatterplots

Fig. 2. **A direct visual comparison.** From left to right: (a) A typical 2D scatterplot displaying dimensions 0 and 1, (b) A scatterplot matrix showing all orthographic projections, (c) Our proposed hierarchy of localized scatterplots. Here, a standard scatterplot is augmented with a local view on the data in the overlapping area. The display of different dimensions in the inset reveals information about the *local* internal structure of the dataset. The subplot displays exactly those data items occluded by the local window in the parent plot which is the same as (a). Additional subplots can be added as needed.

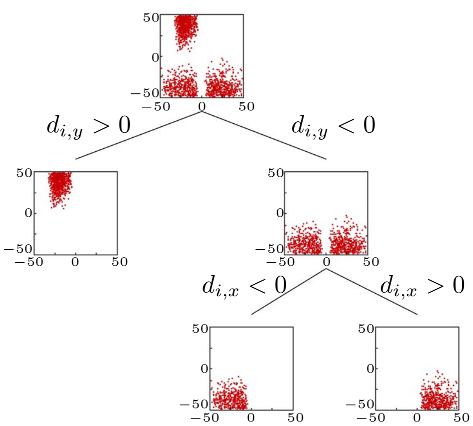
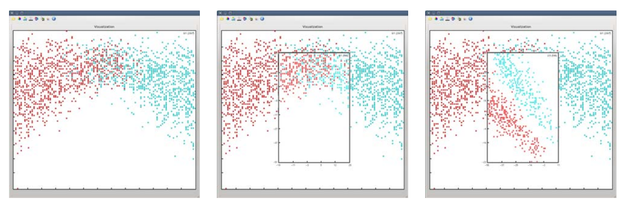


Fig. 3. **Tree representation of a NSP.** Each node, representing a localized subplot, inherits the min/max bounds from its ancestors, further limiting the displayed data set. In this example the projection axes are the same for all plots and the origin (0,0) is in the center of the projection.

V. Exploration Tool

We created a prototype implementation of our hierarchy of localized scatterplots to show how it could be used in practice. It is important to note that we do not claim that the concept of HLSPs should be used as a stand-alone visual analysis tool. We would rather suggest to use it as an add-on to other exploration tools such as Xmdv or GGobi. The proposed tool is mainly a simplistic example of how to use the concept of HLSP and should be regarded as such.

The main visual component of our tool consists of a simple scatterplot view with a menu bar at the top providing basic functionalities such as choosing the color and shape of the displayed data points, Fig. 4a. The user is free to assign any dimension of the dataset to either the x- or y-axis and can smoothly zoom in or out to inspect the plot in more detail. The currently displayed dimensions of an LSP is always displayed



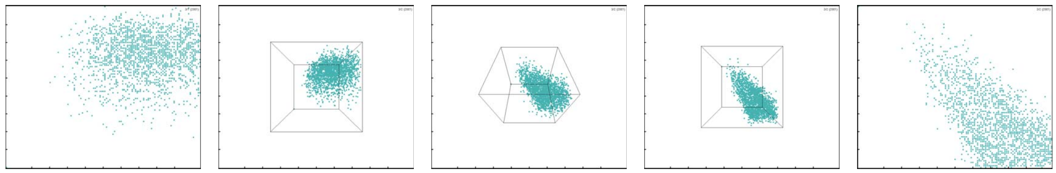
(a) (b) (c)

Fig. 4. **Exploration tool.** From left to right: (a) The main interface showing a simple scatterplot at the beginning. (b) A new LSP is created within the view, initially using the same dimension axis as the parent plot. (c) The projection axis of the LSP can be changed as desired.

in the top right corner, the tick marks at the bottom and left are automatically adjusted according to the zoom level. To create a localized scatterplot within the current view, the user can draw a rectangular selection to mark an area for further inspection. Based on the x and y edge positions of the selection tool within the current view the min/max bounds for the nested LSP are derived. The new LSP is created at the same position covering the selected area and displaying the subset of selected data items, Fig. 4b. The user can then again choose any projection axis for the subplot, Fig. 4c, and can continue the exploration. The current projection axes and number of selected points is displayed in the upper right corner of each LSP. To aid the user in creating the LSP at meaningul positions we support two strategies: animated 3D transitions and the *cross-dimensional semantic lens* tool.

A. Animated 3D Transition

We use the rolling-the-dice approach [17] to create animated smooth transitions between the different dimension as a 3D rotation. The benefit of such a transition is that the human mind can interpret the motion as shape [26] which helps to identify e.g. hidden clusters. We follow the three-stage transition proposed in [27]: transition into a perspective view



(a) (b) (c) (d) (e)

Fig. 5. **Transition between dimensions.** Overview of the animated transition used for changing the projection axes, here the y-axis is changed.

with normalization of the data, rotation, and backprojection. Given two currently viewed dimensions x and y and a vertical transition to a new dimension y', the individual steps are (see also Fig. 5):

- *Perspective projection:* The orthographic view of the xy plane is smoothly changed into a perspective view using the data values in the y' dimension as z-coordinates. For this we use a time-dependent projection matrix $\mathbf{P}$.

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & p & 1-p \end{pmatrix} \quad (1)$$

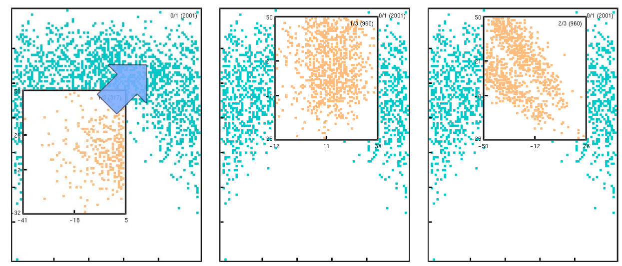
  With $p = 0$ we have a simple orthographic projection, with $p = 1$ we have a perspective projection matrix. Additionally, we smoothly scale the data during this first step of the animation to fit into a unit cube and move it along the z-axis, Fig. 5a and Fig. 5b.
- *Rotation:* The scatterplot is rotated by 90 degrees up or down, depending on whether the dimension index y' is larger or smaller than y. This causes the axis y' to replace y, Fig. 5c and Fig. 5d.
- *Backprojection:* The rotated cube is then smoothly back-projected to the new orthographic view inverting the process of perspective projection from the first step, Fig. 5e.

B. Cross-dimensional Semantic Lens

Besides the 3D rotation, we additionally propose a novel tool, the *cross-dimensional semantic lens* (CDSL). The CDSL is a natural extension to the HLSP which helps to quickly scan a dataset for interesting features based on the idea of a Magic Lens$^{\text{TM}}$ [28]. If little or nothing is known about the dataset it usually can become increasingly difficult to find local features of interest. The CDSL is basically a movable LSP that can be used for interactive local investigation of a dataset. The user creates a LSP at a desired position and size and specifies the dimensions to investigate in more detail. Using the mouse the user can drag the LSP across the parent LSP revealing a view into the new specified dimensions which is updated in real-time, Fig. 6. Once the user finds something interesting, the LSP can be adjusted as needed or the exploration can be continued choosing a different dimension for the CDSL.

VI. Applications

We now demonstrate the analytical capabilities of our proposed approach in three different application scenarios for typical multidimensional visual exploration tasks. In the first two scenarios we use synthetic data as a proof of concept and exemplify the suggested workflow. Then, we describe an explorative setting for a real-world scenario. Again, we want to stress that this is a prototypical stand-alone application and is only used here for explanatory purposes. For more complex and higher dimensional datasets the concept of HLSP should and can be integrated into already existing exploration tools. This, however, is beyond the scope of this paper and should be postponed for future work combined with a thorough user study which we deem hardly meaningful in the current state. Fruitful directions would be to use more sophisticated navigation tools, e.g. in the form of SPLOMs [17], or display each LSP as a SPLOM itself on demand to speed up the analysis. Here we focus on the basic interactions with a HLSP as an introduction to the concept itself.

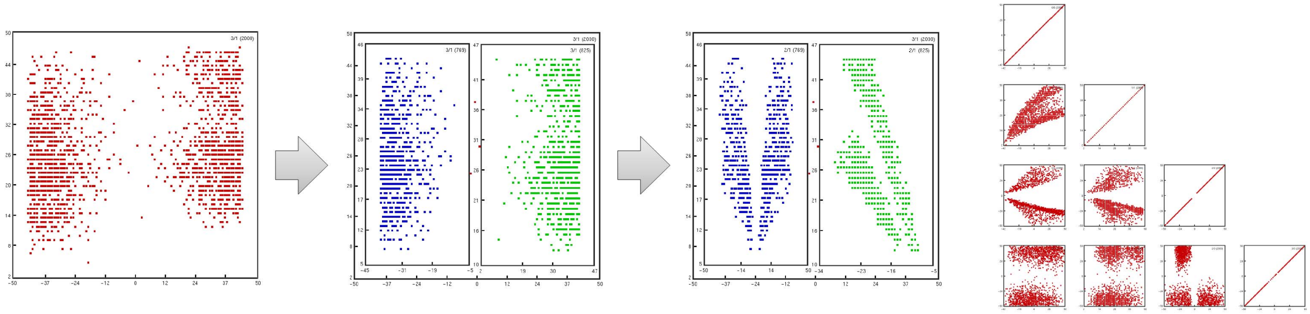


(a) (b) (c)

Fig. 6. **The *cross-dimensional semantic lens* (CDSL).** The CDSL tool allows interactive and local views into other dimensions of the dataset. (a)The user may freely move the LSP, denoted by the blue arrow, and change the projection dimensions as desired, here the user switched from (b) dimensions 1/3 in the local view to (c) dimensions 2/3. The local view is updated in real-time.

A. Application Scenario 1: Cluster Representation

We used a 2000 record sample of a synthetically created 4D dataset using the technique described in [29]. This data consists of four well separated subsets one of which is obscured in the orthographic projections of the whole dataset. A SPLOM of the data is shown in Fig. 7d. The user proceeds by scanning through the different orthogonal projections using our proposed scheme. Though no projection shows more than two subsets, the user can deduce by comparison of the projections that there seem to be at least three different subsets in the dataset, but to be sure the user analyzes the subsets in more detail. Once a view with a promising data separation is found,



(a) (b) (c) (d)

Fig. 7. **Cluster representation example.** (a) – (c) A typical exploration process using our HLSP, consisting of (a) finding a promising view, (b) creating additional LSPs, and (c) local investigation of the view creating additional LSPs. The four subsets are easily detectable with the HLSP visualization. (d) A SPLOM representation of the dataset for comparison. Non-of the projections reveal enough information to deduce the correct number of well separated subsets hidden in the dataset.

Fig. 7a, the user creates two LSPs, one for each well separated subset, Fig. 7b. Changing the color for each LSP increases the visual discriminability. Continuing the visual exploration for each of the separate LSPs, the user detects that actually both LSPs comprise two well separated subsets, Fig. 7c, which was not deducible from the orthographic projections in the SPLOM alone and is now conveniently displayed by our HLSP visualization.

B. Application Scenario 2: Classification

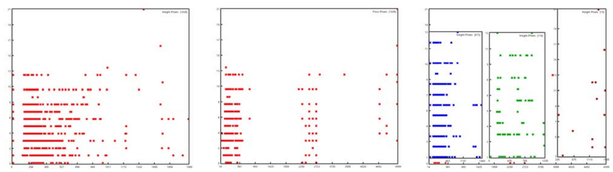
In this example we again used the technique described in [29] to create a 5D dataset comprising 2002 data items. The dataset consists of two classes entangled in a complex way so that no single 2D projection is able to separate them, Fig. 1d. Using our HLSP the user can isolate and further investigate local parts of the data, Fig. 1a–c. Each level of the HLSP creates an improved separation of the two classes, Fig. 1a and Fig. 1b, until a sufficient separation is found, Fig. 1c. Due to the direct visual connection established between each level of the HLSP, we can simply derive a decision tree for classification directly from the visualization. In this case classification can be performed as follows: For a data item $\mathbf{d}_i$, if $d_{i,2} > 2$ then $\mathbf{d}_i$ belongs to the blue class. If $d_{i,2} < -1.8$ then $\mathbf{d}_i$ belongs to the red class. Otherwise, if $d_{i,0} < -2.2$ then $\mathbf{d}_i$ belongs to the blue class, if $d_{i,0} > -1.8$ then $\mathbf{d}_i$ belongs to the red class. And finally, if not classified by the former decisions, $\mathbf{d}_i$ probably belongs to the blue class if $d_{i,4} > d_{i,3}$ otherwise it will belong to the red class. It is difficult to derive the same amount of information from the SPLOM itself, Fig. 1d, even when using brushing to separate similar data, Fig. 1e.

C. Application Scenario 3: Comparative Analysis

In a final application scenario we apply our technique for a comparative analysis using a real-world dataset for digital cameras containing 1039 cameras categorized in 13 dimensions, one being the name, the rest are categorical or numerical attributes. In this scenario we follow Mike, a visualization analyst, who wants to gain knowledge about the digital cameras available and which one would suit him best. Mike has no previous knowledge about the dataset and only a fuzzy understanding of what exactly he is looking for. All that he knows is that he wants a lightweight camera with a high resolution, but hasn't decided on how much he is willing to spend yet. After starting our exploration tool and loading the dataset, Mike assigns the dimension *weight* to the x-axis and *resolution* to the y-axis, Fig. 8a. Mike realizes that it is tedious to find a suitable camera, as the prices vary strongly in this view. Therefore, he switches the x-axis to *price* again and realizes that there seem to be roughly three largely different price ranges, Fig. 8b. Mike notices that the resolution does not seem to be related to the price range, so he decides that he would like to compare the weight correlated to the resolution of the camera for each of the three price ranges. Using the idea of LSPs, Mike creates a new view for each of the clusters within the visualization and changes the x-axis to *weight* for all three LSP, the y-axis remains at *resolution*, Fig. 8c. From his intuition Mike first investigates the high priced cameras, as he believes that quality correlates with the price. To his disappointment he realizes that weight roughly correlates with resolution in this price segment as the data items are only present in the lower right half of the plot. As weight is important for him, he decides not to choose any camera from this segment. As the low- and mid-priced cameras span all the same space of weight and resolution, Mike decides to choose a low-cost camera and maybe buy a high-priced camera once they get lighter because he is not willing to spend between $2,000$$ and $3,000$$ for a mediocre camera with which he might not be happy.

VII. Conclusion

In this paper we have proposed an extension to the classical scatterplot visualization to explore multidimensional datasets. Precisely, we allow to create local scatterplots within other scatterplots in a hierarchical fashion. The benefit of our approach is that the user is free to visually analyze subsets of the dataset and derive and display complex relationships from within the data in a simple and easy to understand manner



(a) weight/resolution (b) price/resolution (c) weight/resolution

Fig. 8. **Comparative analysis example.** (a) weight/resolution plot. (b) price/resolution plot. (c) weight/resolution plot using LSPs for a direct comparison.

similar to simple decision trees. Additionally, the technique allows to compare several subsets of the data at once. Our prototype implementation helps to visually analyze a dataset through zoom capabilities, animated transitions between scatterplots, and the cross-dimensional semantic lens.

Our developed exploration tool provides a first insight into the power and expressiveness of hierarchical scatterplots. We showed several application scenarios where HLSPs proved useful, also in comparison to traditional brushing in a SPLOM [30]. To truly evaluate the usefulness of our new approach we plan to conduct a thorough user study investigating HLSPs as an add-on in already well established visual analytics tools. While we allow zooming in the interactive application, using more than three levels of LSPs becomes a limiting factor for standard printing as the inner LSP are quickly reduced in size. An interesting direction for future work would be to combine our cross-dimensional semantic lens with quality metric algorithms [13], [14], [31] so that the displayed dimension is automatically updated to the best view according to the measure used.

Acknowledgment

The authors gratefully acknowledge funding by the German Science Foundation from project DFG MA2555/6-2 within the strategic research initiative on Scalable Visual Analytics.

References

[1] D. B. Carr, R. J. Littlefield, W. L. Nicholson, and J. S. Littlefield, "Scatterplot matrix techniques for large n," *Journal of the American Statistical Association*, vol. 82, pp. 424–436, 1987.

[2] I. Jolliffe, *Principal Component Analysis*. Springer, 2002.

[3] T. Cox and A. Cox, *Multidimensional Scaling, Second Edition*. Taylor & Francis, 2010.

[4] D. A. Keim, "Information visualization and visual data mining," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 1–8, 2002.

[5] E. R. Tufte, *The visual display of quantitative information*. Graphics Press, 1986.

[6] D. Asimov, "The grand tour: A tool for viewing multidimensional data," *SIAM Journal on Scientific and Statistical Computing*, vol. 6, no. 1, pp. 128–143, 1985.

[7] Y. Koren and L. Carmel, "Robust linear dimensionality reduction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 4, pp. 459–470, 2004.

[8] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[9] G. Ellis and A. Dix, "A taxonomy of clutter reduction for information visualisation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1216–1223, 2007.

[10] D. A. Keim, M. C. Hao, U. Dayal, H. Janetzko, and P. Bak, "Generalized scatter plots," *Information Visualization*, vol. 9, no. 4, pp. 301–311, 2010.

[11] J. F. Rodrigues, Jr., C. E. Cirilo, A. F. Prado, and L. A. M. Zaina, "Hierarchical visual filtering, pragmatic and epistemic actions for database visualization," in *ACM Symposium on Applied Computing*, 2013, pp. 946–952.

[12] D. A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler, "Challenges in visual data analysis," in *Proceedings of the conference on Information Visualization*. IEEE Computer Society, 2006, pp. 9–16.

[13] E. Bertini, A. Tatu, and D. A. Keim, "Quality Metrics in High-Dimensional Data Visualization: An Overview and Systematization," *IEEE Symposium on Information Visualization (InfoVis)*, vol. 17, no. 12, pp. 2203–2212, 2011.

[14] A. Tatu, G. Albuquerque, M. Eisemann, P. Bak, H. Theisel, M. Magnor, and D. Keim, "Automated analytical methods to support visual exploration of high-dimensional data," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 17, no. 5, pp. 584–597, 2011.

[15] J. H. Friedman and J. W. Tukey, "A projection pursuit algorithm for exploratory data analysis," *IEEE Trans. Comput.*, vol. 23, pp. 881–890, 1974.

[16] M. A. Fisherkeller, J. H. Friedman, and J. W. Tukey, "Prim-9: An interactive multi-dimensional data display and analysis system," in *ACM Pacific*, 1975, pp. 140–145.

[17] N. Elmqvist, P. Dragicevic, and J.-D. Fekete, "Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation," *IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis 2008)*, vol. 14, no. 6, pp. 1141–1148, 2008.

[18] G. G. Robertson, J. D. Mackinlay, and S. K. Card, "Cone trees: animated 3d visualizations of hierarchical information," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1991, pp. 189–194.

[19] S. K. Feiner and C. Beshers, "Worlds within worlds: metaphors for exploring n-dimensional virtual worlds," in *Proceedings of the 3rd annual ACM SIGGRAPH symposium on User interface software and technology*, ser. UIST '90, 1990, pp. 76–83.

[20] H. Piringer, R. Kosara, and H. Hauser, "Interactive focus+context visualization with linked 2d/3d scatterplots," in *International Conference on Coordinated & Multiple Views in Exploratory Visualization*, 2004, pp. 49–60.

[21] C. Ahlberg, C. Williamson, and B. Shneiderman, "Dynamic queries for information exploration: An implementation and evaluation," in *Proceedings of the ACM CHI92 Conference on Human Factors in Computing Systems*, 1992, pp. 619–626.

[22] A. R. Martin and M. O. Ward, "High dimensional brushing for interactive exploration of multivariate data," in *Proceedings of the IEEE Conference on Visualization*, 1995, pp. 271–278.

[23] J. C. Roberts, "State of the art: Coordinated & multiple views in exploratory visualization," in *International Conference on Coordinated and Multiple Views in Exploratory Visualization*, 2007, pp. 61–71.

[24] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[25] S. van den Elzen and J. van Wijk, "Baobabview: Interactive construction and analysis of decision trees," in *Visual Analytics Science and Technology*, 2011, pp. 151–160.

[26] S. Ullman, *The Interpretation of Visual Motion*. MIT Press, 1979.

[27] J. Heer and G. Robertson, "Animated transitions in statistical data graphics," *IEEE Transactions on Visusalization and Computer Graphics*, vol. 13, no. 6, pp. 1240–1247, 2007.

[28] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose, "Toolglass and magic lenses: the see-through interface," in *Proc. of SIGGRAPH*, 1993, pp. 73–80.

[29] G. Albuquerque, T. Löwe, and M. Magnor, "Synthetic generation of high-dimensional datasets," *IEEE Transactions on Visualization and Computer Graphics (TVCG, Proc. Visualization / InfoVis)*, vol. 17, no. 12, pp. 2317–2324, 2011.

[30] R. A. Becker and W. S. Cleveland, "Brushing scatterplots," *Technometrics*, vol. 29, no. 2, pp. 127–142, 1987.

[31] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan, "Selecting good views of high-dimensional data using class consistency," *Computer Graphics Forum (Proc. EuroVis)*, vol. 28, no. 3, pp. 831–838, 2009.