Device-based Adaptation of Visualizations in Smart Environments

Georg A. Fuchs, Conrad Thiede, Mike Sips, and Heidrun Schumann

Abstract—Smart environments are beginning to have a large impact to collaborative group work in business and science. The multi-user and multi-display character of these group work environments presents a novel challenge for information visualization, namely, the adaptation of graphical representations of data to specific target devices in the environment. In this paper, we discuss a general strategy for an automated device-based adaptation of visualizations. We report interesting preliminary results of our adaptation strategy for conventional scatterplots used within a service-oriented visualization framework.

Index Terms—Information Visualization, Display Adaptation, Smart Environment, Collaboration, Scatterplot.

1 Introduction

Smart environments facilitate collaborative work of a group of users, e.g., in the analysis of massive customer databases to achieve better business decisions. A typical device ensemble in a smart room environment consists of stationary devices such as desktop computers, projectors, light, or motion trackers, but also strives to integrate mobile devices such as laptops, PDA or smart phones which are often carried by the users. In contrast to classical meeting room environments, smart environments augment sensor devices to monitor the environment and its users to enable a "smart" interaction between the users and the environment.

These novel environments present a number of challenges for information visualization, namely, (a) to support different user goals and data sources, (b) to utilize multiple displays, and (c) to facilitate interaction among a group of users. In this paper, we consider the adaptation of graphical representations of data to specific target devices in smart environments. The adaptation of graphical representations gives rise to the following two visualization challenges:

- In collaborative work sessions, users usually share a visualization on a wall-sized display to analyze/discuss potentially interesting features of the data, but also use the same graphical representation on their personal output devices to look at the data. A smart room environment should allow a dynamic adjustment of the requirements such as the task at hand and the visualization needed to foster insight into the data, but also should support a interaction of the users with the environment such as moving around to join different subgroups of the users. Thus, a graphical representation of the data often needs to be distributed to different output devices.

- The diversity of output devices/display sizes is often quite high in smart environments. To maintain visual effectiveness of a graphical representation under different display sizes, i.e. important features of the data are faithfully communicated to the user, a visual interface should apply a *device-based adaptation* to the visual output.

To facilitate a smart interaction between the users and the environment, the adaptation of a visualization to a specific target device should be performed automatically. This requires suitable metrics to measure and assess the effectiveness of a visual representation for the current output device and task. In this paper, we focus on automatic device-based adaptation of visual representations to support the dynamic assignment of visualizations to varying display sizes.

- *G. Fuchs, C. Thiede and H. Schumann are with University of Rostock, E-mail: {georg.fuchs|conrad.thiede|heidrun.schumann}@uni-rostock.de.*
- *Mike Sips is with MPII Saarbrücken, E-mail: msips@mpi-inf.mpg.de.*

We review related work and discuss principal distribution scenarios in Section 2. As the main contribution, we propose a general adaptation strategy for automatic device-based adaptation of visualizations, and discuss the key challenges in Section 3. Section 4 presents preliminary results on our ongoing work on integrating this strategy for conventional scatterplots with a service-oriented framework for distributed visualizations. Section 5 concludes with a discussion of open research challenges and gives an outlook on future work.

2 Background & Related Work

In smart environments the devices form a loosely coupled network, which allows for the necessary communication to accomplish tasks in a coordinated fashion [1]. Specifically, this enables distribution of visual output among several output devices depending on the current situation, based on an automatic situation assessment [7]. However, multi-user collaboration in these environments is explicitly *not* limited to showing a single user's content on several displays. Instead, available output devices need to be assigned to visual representations according to the users' current situation and task requirements. Two principal distribution scenarios can be distinguished: multiple users work jointly on a single representation, or one or more users require individual visualization specific to their current task.

In the first scenario, all users view the same information representation that is either *distributed* to show simultaneously on different displays, or *split* to show sub-regions among several (neighboring) displays. The latter distribution scheme is useful if an array of small displays (such as PDA and TabletPC) is available to the user, or if the data set is extremely large. In these situations, using well-established concepts such as Overview & Detail and Focus & Context can help in exploring the data. Here, one "public" display (e.g. a whiteboard display) shows an overview representation of the data, while other "private" displays (e.g. Laptops) show additional data for regions of personal interest. In the second scenario, users require different information representations for their individual tasks. If there are fewer displays available than visual representations required, the available display space must be shared by *combining* representations. Note, both distribution scenarios require that visualizations are scaled to fit a particular display area on the target device.

Many approaches found in literature deal with scalable representations of graphical content such as video streams and vector graphics [8], 2D maps [4] or 3D virtual models [5], but do not address information visualization. Most adaptive visualization approaches, on the other hand, consider the properties of the data and the visualization goal (e.g., [6, 17]), but only few approaches adapt to the available resources on different target devices (e.g. [12]). Others address issues related to distributed visualization, e.g. multiple client platforms [12] or the use of web services as the output distribution mechanism [16]. In [13], the authors propose a more general approach for distributed visualization. It uses a service-oriented architecture (SOA) to generate visual representations in a distributed fashion, including mobile devices that can enter or leave the smart room's ensemble.

Fig. 1. Example for lost effectiveness of a conventional scatterplot due to transfer to a smaller display: the perceived number of classes may change from three to two.

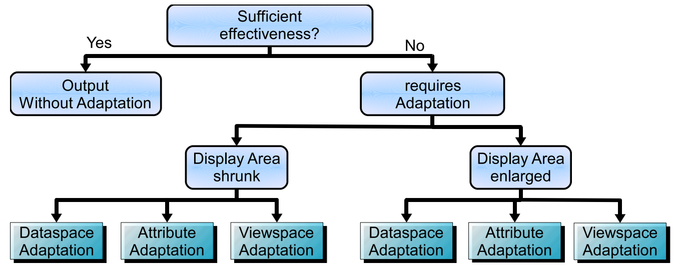


Fig. 2. Scheme of the adaptation process after a visual representation has been reassigned to a new display of different size.

These architectures support, in principle, dynamic multi-user and multi-display settings as outlined above. However, as the generated visual representations get dynamically reassigned to different displays, adaptation may become necessary to maintain their effectiveness (see Fig. 1). Thus, a distributed visualization architecture should incorporate suitable adaptation mechanisms to adapt graphical representations of data to specific output devices.

3 General Device-based Adaptation Strategy

The basic idea of our general device-based adaptation in this setup is illustrated in Figure 2. The first step is to estimate the visual effectiveness of the representation. This might be done either as an initialization step prior to the use of the output device or after a visual representation that is already displayed has been re-assigned to a different display. If the effectiveness score indicates the visual encoding becomes poor for the assigned display area, in a second step an appropriate adaption is selected to increase the visual saliency of potentially interesting features of the data.

For this general adaptation scheme to work, a number of questions need to be addressed. Namely, (a) how to evaluate visual effectiveness, (b) what methods and constraints exist for adaptation, and (c) what are the requirements for a suitable infrastructure to support both output distribution and device-based adaptation in smart environments.

3.1 Visual Effectiveness

The effectiveness of a graphical display to faithfully present potentially interesting features of data to the user can be degraded in either of two cases: (a) when the visual representation has been transferred to a display smaller than originally intended, or (b) to a much larger screen, e.g. a whiteboard display. In the first case, a problem occurs when too much data is displayed on too small displays. The resulting "visual clutter" is well researched (e.g. [2, 11]) and can have a significant impact on the effectiveness of a visual representation. However, there is also evidence that in some situations, upscaling a visual representation to a larger display area may also lead to a degradation in effectiveness as e.g. perception of overall data distribution and local densities change [2].

The definition of measures of goodness to score the visual effectiveness of a particular visualization is still a open research problem [9]. Tufte [15] proposes some measures to estimate the quality of 2D representations of static data, like the *data-to-ink* ratio or the *data density*, which take into account the size of the visual representation in relation to the amount of data displayed. Other approaches measure the overall "visual clutter" [2] or consistency [11] to evaluate how faithfully a visual representation communicates data characteristics. Moreover, to enable automatic adaptation reliable thresholds specifying the perceptual boundaries on what constitutes effective visualizations are required. This will necessitate user experiments to determine those thresholds, and to get a good estimate on the perceived quality of adapted visualizations.

3.2 Adaptation Options & Constraints

The visualization process can be understood as a pipeline of four data stages plus intra-stage and transform (inter-state) operators [3]: raw data (1^{st} stage) is transformed into analytical abstractions (2^{nd} stage), e.g. by calculating statistical moments, which are then further mapped to visual abstractions (3^{rd} stage), e.g. 2D points with position and color. Finally, the rendering process generates the image data (4^{th} stage). The first two stages constitute the *data space* that is transformed by the visualization's *mapping* into the *view space* comprised of the last two pipeline stages. This visualization model yields starting points for adaptation both in data space and in view space, depending on the modified pipeline stage. In addition, the mapping parameters of data values to visual attributes, i.e. the transformation from data to view space, can also be modified (attribute adaptation).

If a visual representation is transfered to a smaller display, the level of detail may need to be reduced. Here, adaptation of the representation in data space includes filtering or using a higher abstraction level (e.g. clusters or statistical aggregates) to reduce the amount of data items displayed. View space adaptation aims to reduce visual clutter (e.g., by employing density binning [10]). Contrary, on larger displays, more details can be shown, by selecting a lesser degree of abstraction (e.g. another level from hierarchical clustering) or adjusting filter settings accordingly.

Another question is which visual attributes and aggregations are eligible for adaptation. Visualization techniques encode data values to different visual elements and their associated attributes, thus requiring specific adaptation mechanisms based on the set of used attributes. Adaptation is also inherently task-dependent, i.e., what view space aggregations and abstractions of the raw data are admissible for a given visualization goal? Identifying outliers has different requirements for a visual representation than analyzing complex relations, for instance.

Providing adequate solutions to these questions is not trivial. As a proof of concept, we chose 2D scatterplots to derive concrete procedures from the general adaptation scheme (see Section 4, using the infrastructure described next.

3.3 Infrastructure

A suitable infrastructure generates device-driven visual representations of the data for the available output devices, utilizing computing devices in the smart room's device ensemble, and distributes these according to the current requirements of the users. We chose a service-oriented framework called SSC from [13] as the infrastructure for our experiments. It uses a visualization pipeline composed of distributed services implementing pipeline operators. Adaptation mechanisms can be integrated into this general framework through service parameterization, or through extensions of the basic pipeline with additional services, such as a filtering service to sample data prior to the mapping stage of the pipeline. Furthermore, in line with the principal distribution scenarios identified in Section 2, we implemented the following distribution mechanisms for visual representations into SSC to facilitate testing:

To enable *distribution* of a single visual representation to multiple devices, the final rendering stage of the pipeline is forked to multiple rendering services [13], one for each device. This allows device-based adaptation in view space and attribute adaptation on each device individually, while the earlier data stages are processed only once for all devices[1].

Splitting a visual representation across multiple displays is achieved by forking a second pipeline to render a detail view of the user-selected area on another device. Currently, the framework supports detail view of regions of interest that are interactively selected by the user.

The capability to *combine* multiple visualizations on a single display is provided by an *aggregator service* that partitions the physical display into a corresponding number of viewports. The visual representation for each viewport is generated by the respective visualization pipeline that feeds its output to the aggregator service.

The general framework augmented with these distribution mechanisms provides the basis for our experiments.

4 Demonstrating Example

As a starting point for our test implementation, we created a visualization pipeline with the SSC framework that defines the necessary operators to create a 2D scatterplot from multivariate data. We assume the data points have class labels assigned, and that the visualization goal is to communicate the class structure of the data. We further assume a suitable 2D projection is used that captures the high-dimensional class structure in the 2D scatterplot [11].

The scatterplot representations generated by this pipeline are then distributed to displays in the environment using SSC's distribution mechanisms as described in Section 2. We discuss implementation details for the different steps of the adaptation process as summarized by Figure 2 next.

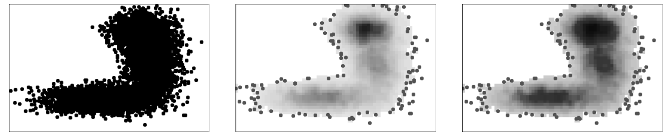
4.1 Efficiency Evaluation

The first step is to estimate the effectiveness of the (newly created or re-assigned) scatterplot. For this purpose, we use two measures. First, the class consistency score [11] is calculated. Second, the *visual density* of the scatterplot is determined to measure the impact of the new display area's size. Here we define the visual density of a scatterplot as the average ratio of cluster members to the screen space occupied by the cluster. The area is conservatively estimated by calculating the size in pixels of the convex hull of all points belonging to that cluster.

We established approximate thresholds for the two measures in an informal pilot study for the two principal cases of assigning a scatterplot to a smaller and a larger display area, respectively. In the first case, a drop of the consistency score and an increase in visual density can be observed. Low consistency scores ($60-80\%$) in conjunction with density values between 0.1 and 0.7 suggest the scatterplot display is saturated with points. At this point, clusters begin to mix visually (see Figure 1). This can be countered by suitable attribute adaptation, e.g. by using different shapes or by increasing the color contrast between points belonging to different clusters. When the consistency is even below 60% and visual density is above 0.7, mixing of clusters has become so severe that overplotting has likely occurred, and attribute adaptation does not help much to improve effectiveness. We propose to switch to a density plot in this situation since individual data points are no longer discernible anyway. The density representation at least allows the user to faithfully extract the cluster structure. We chose to integrate a binning approach (view space adaptation) further supplemented with an alternate color coding (attribute adaptation).

The second case – assignment to a larger display area – exhibits no drop in consistency. However, plots with a good consistency score ($> 80\%$) but visual density below 0.005 (i.e., only about one out of 200 pixels within the convex hull is set) describes a situation in which data points spread too much. An important finding of our preliminary user study is that the extraction of potentially interesting features by the human is biased toward low visual densities. In our scenario, the participants in the user started to identify sub-regions as individual clusters due to diminishing visual densities. This effect was also observed by Bertini and Santucci [2]. This can be addressed by encoding cluster membership into unused visual attributes (e.g. shapes, color) or by deliberately downscaling the representation to use only a fraction of the available area. To find appropriate adaptation strategies for this situation, however, is still an open research problem we did not yet pursue further. Therefore, the following subsections discuss examples for the three adaptation types (cf. Figure 2) specifically for the case of shrunk display sizes.

[1]Note that for adaptation in data space, the pipeline would have to be split into parallel services even sooner, at the corresponding data stage.



(a) (b) (c)

Fig. 3. **Results of density binning for a synthetic data set** – (a) standard scatterplot, (b) density binning with outlier preservation and linear color scale, (c) bin frequencies are mapped to logarithmic color scale.

4.2 View Space Adaptation

Our density binning approach borrows from [10], which has been proposed as a Focus & Context technique for crowded parallel coordinates. The basic idea can be summarized as follows. Both scatterplot axes are divided into b regular intervals. The resulting set of $b \cdot b$ bins represents a so called bin map and can be thought of as a 2D histogram of the data point distribution in view space. Every non-empty bin is represented as a rectangle in the adapted scatterplot, with the bin frequency color-coded into its fill color.

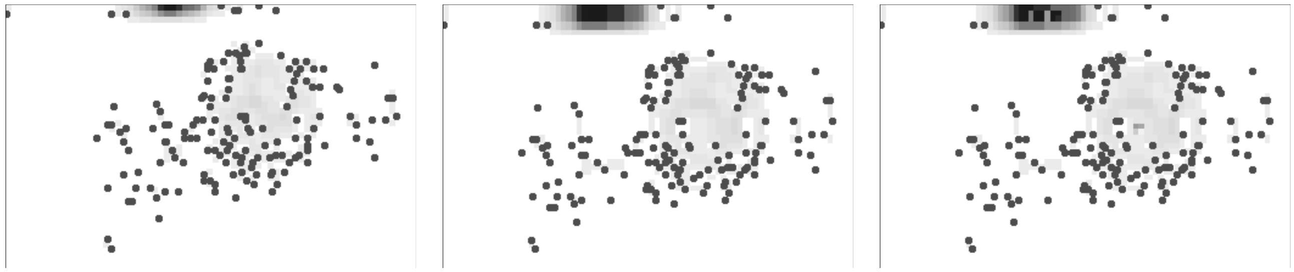
However, to faithfully extract the cluster structure, the user should be able to discern the cluster centers from the frequency representation. Ideally, each cluster should register as a high-frequency region in the plot that is visually distinguishable from peaks of neighboring clusters. To facilitate these properties, we introduce an extension of the approach based on the following ideas.

Automatic binning resolution adjustment: First, we adjust the bin resolution along the scatterplot axes with respect to cluster center locations to determine a good binning. Initially, the scatterplot area is partitioned into $b_x \times b_y$ bins according to a given starting bin size. Next we check if a bin contains more than one cluster center. The binning subdivision is then refined by increasing b_x (b_y) by 1, and the check is repeated. This subdivision continues by alternately increasing b_y/b_x until (a) all cluster centers are located in individual bins, or (b) predetermined bin size is reached. We found that a bin sizes between 5×5 pixels (starting value) to 2×2 pixels (minimum threshold) yield a good compromise between clutter reduction and faithful reproduction of clusters on small displays. (see Figure 3(a, b)).

Local magnification with sub-binning: After the bin size and the resulting bin frequencies have been determined, we optionally apply a rectangular fish-eye distortion aligned with the bin grid centered about those bins containing the cluster centers (see Figure 4(b)). This increase in screen space available for the clusters center regions allows sub-binning these regions. The sub-binning factor is thereby proportional to the magnification factor, e.g. a magnification of focused bins by factor two results in a two-fold subdivision of these bins. The locally increased bin resolution reproduces frequency variations around the cluster centers with higher fidelity and thus can improve visual separation of clusters with low separation (Figure 4(c)).

4.3 Attribute Adaptation

Moreover, instead of using the originally proposed, linear color map from [10], we use a logarithmic scale for bin colors. The skewed distribution of bin frequencies between dense centers of compact clusters and sparse regions where clusters mix suggests a non-linear color scale is better suited for this kind of data [14]. Figure 3(c) illustrates the difference between a linear and a logarithmic color scale. The latter scale



(a) (b) (c)

Fig. 4. **Use of rectangular fish-eye distortion** – (a) undistorted density plot, (b) bins containing cluster centers magnified, (c) magnified regions sub-binned. The increased density sampling rate in (c) reveals that the dense region at the top is actually comprised of three peaks (i.e., clusters). The location of the center peak within the low-density cluster at the middle-right is also emphasized.

assigns more gray levels to the low end of the frequency value range, thus further enhancing visibility of the peaks around cluster centers.

4.4 Data Space Adaptation

A third option for smaller displays is to reduce the number of data items prior to mapping them to the visual representation. This can be achieved by employing data sampling. To preserve the class structure, however, the sampling process should maintain local densities [2]. Although random sampling schemas are, in principal, of value in many adaptation situations, it is rather useless in our scenario. For this reason, implementation of a suitable density-preserving sampling service has not been pursued yet.

5 Summary & Discussion

Multi-user, multi-display settings in smart environments present novel challenges for information visualization. In particular, the varying sizes and capabilities of different output devices require device-based adaptation of generated visual representations. In this paper, we proposed a general strategy to guide this adaptation based on the notion of visual representation effectiveness and a visualization pipeline model. As a proof-of-concept, we implemented corresponding adaptation mechanisms for 2D scatterplots in the SSC framework [13].

We believe the general strategies proposed in this paper are valid and can be employed to many visualization techniques, however, *smart* device-based adaptation (i.e., minimal user intervention) requires more work. First, thresholds for consistency and visual density need further evaluation in controlled user experiments, which is the subject of our current efforts. This specifically includes cases with low visual densities (display size is too large for a given visualization), as this branch of the adaptation scheme (Fig. 2) was not pursued in detail so far. Also, the problem of meaningful effectiveness measures requires more research. Consistency is applicable only for scatterplots of clustered data. Metrics striving to capture the amount of visual clutter in visual representations [2, 11] seem promising candidates for a more generic effectiveness evaluation.

The distribution of visual representations is still work in progress as well. Smart distribution requires detection of the current user situation followed by inferring the individual goals as well as the group's intention, which is not the focus of our work. However, we plan to integrate these schemes with an existing inference module (cf. [7]) in the future.

Additionally, we plan to further investigate task-driven aspects of the adaptation process. A typical smart room scenario is a decision making process where several domain experts look at the same data, albeit with different goals and requirements to the visual representation. So far, we only considered a single visualization goal, namely communicating the class structure of multi-dimensional data.Note that the visualization goal respectively the user's current task have a direct impact on device adaptation. The task determines what data abstractions are permissible or how visual attributes should be modified e.g. through color coding. Using a suitable task description in the adaptation process would therefore allow to integrate different task-specific aspects in a single collaborative visualization on the same screen, rather than just juxtaposing several independent representations. Our initial studies in this direction included task-based adaptation of graphical content using enriched task models [5] and a task taxonomy for color coding [14].

References

[1] E. Aarts and J. Encarnação. *True Visions: The Emergence of Ambient Intelligence*. Springer, 2006.

[2] E. Bertini and G. Santucci. Quality metrics for 2d scatterplot graphics: Automatically reducing visual clutter. In *Proc. Smart Graphics (SG'04)*, volume LNCS 3031, pages 77–89. Springer-Verlag, 2004.

[3] E. H. Chi. A taxonomy of visualization techniques using the data state reference model. In *INFOVIS*, pages 69–76, 2000.

[4] J. Follin, A. Bouju, F. Bertrand, and P. Boursier. Management of multi-resolution data in a mobile spatial information visualization system. In *Proc. 4th Intl Conf. on Web Information Systems*, pages 92–99, 2003.

[5] G. A. Fuchs, M. Holst, and H. Schumann. 3D Mesh Exploration for Smart Visual Interfaces. In *Proc. 10th Intl. Conference on Visual Information Systems*, volume LNCS 5188, pages 80–91, 2008.

[6] I. Fujishiro, Y. Ichikawa, R. Furuhata, and Y. Takeshima. GADGET/IV: a taxonomic approach to semi-automatic design of information visualization applications using modular visualization environment. In *Proc. IEEE InfoVis*, 77-83, 2000.

[7] T. Heider and T. Kirste. Minimizing Cognitive Load by Automatic Display Management. In *Proc. Ubicomp Workshop'07*, 2007.

[8] M. Klima, P. Halabala, and P. Slavik. Semantic information visualization. In *Proc. 19th Intl. CODATA Conference*, 2004.

[9] C. North. Toward Measuring Visualization Insight. *IEEE Computer Graphics and Applications*, 26(3):6–9, 2006.

[10] M. Novotny and H. Hauser. Outlier-Preserving Focus+Context Visualization in Parallel Coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):893–900, 2006.

[11] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. In *Proc. EuroVis 2009*, volume 28 of *Computer Graphics Forum*, 2009.

[12] L. Skorin-Kapov, H. Komericki, M. Matijasevic, I. Pandzic, and M. Mosmondor. MUVA: a Flexible Visualization Architecture for Multiple Client Platforms. *Journal of Mobile Multimedia*, 1(1):3–17, 2005.

[13] C. Thiede, C. Tominski, and H. Schumann. Service-oriented information visualization for smart environments. In *International Conference Information Visualisation (IV)*, Barcelona, Spain, 2009.

[14] C. Tominski, G. Fuchs, and H. Schumann. Task-Driven Color Coding. In *Proc. 12th International Conference Information Visualisation (IV'08)*, London, 8 - 11 July 2008. IEEE Computer Society.

[15] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 2nd edition edition, May 2001.

[16] Y. Zhao, C. Hu, Y. Huang, and D. Ma. Collaborative Visualization of Large Scale Datasets Using Web Services. In *Proc. of Intl. Conf. on Internet and Web Applications and Services*, 2007.

[17] M. X. Zhou, M. Chen, and Y. Feng. Building a Visual Database for Example-based Graphics Generation. In *IEEE Information Visualization*, pages 23–30, 2002.