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, Group Work, Scatterplots

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 graphical representation of data for the current output device and task. Here we focus on automatic device-based adaptation of visual representations to support the dynamic distribution of visualizations to varying display sizes.

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

To enable the communication necessary to accomplish a task in a coordinated fashion, all devices in smart environments form a loosely coupled network (see [1]). In particular, this loosely coupled network of devices can distribute graphical representations of data to specific target devices in the environment. The distribution of visual content to different target device is based on the current state of the group work (analytical requirements of the users, subgroups of the users) which is often estimated by an automatic situation assessment [7]. However, multi-user collaboration in smart environments is explicitly *not* limited to present the visual content of the workspace of a single user on several displays. Two principal distribution scenarios can be distinguished: multiple users share the same visualization and output device, or one or more users require personalized instances of a visualization to accomplish their current task.

In the first scenario, a graphical representation is *distributed* to either several different output devices that present the same graphical representations simultaneously to all users or across an array of (neighboring) devices. The latter distribution scheme is useful if an array of small displays (such as PDA and TabletPC) is available to the user, or the data set is extremely large. In both situations, well-established concepts such as Overview & Detail and Focus & Context can help to explore the data. The basic idea is straightforward. A ”public” display (e.g. a whiteboard display) shows an overview visualization of the data while simultaneously ”private” displays (e.g. Laptops) show finer grain detail of regions of personal interest. In the second scenario, all devices are shared by the users through *combining* information if fewer displays are available than personalized visualizations required. Note, both distribution scenarios require that visualizations are scaled to fit a particular display size of the target device.

Many approaches found in literature deal with scalable representations of visual content such as video streams and vector graphics [8], 2D maps [4] or 3D virtual models [5], but to the best of our knowledge very little work has been done to specifically address information visualization. Most adaptive visualization approaches, on the other hand, only consider the properties of the data and the visualization goal (e.g., [6, 17]), but only few approaches take the available resources of target devices into account (e.g. [12]). Other visualization approaches 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, in-

---

- *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.*

Fig. 1. A distribution of a conventional scatterplot to both a large and a small display (or a scatterplot has been re-distributed from a large to a small display) often causes trouble with visual effectiveness: users may perceive a different number of clusters

cluding mobile devices that can enter or leave the device ensemble of a smart room.

These architectures support, in principle, dynamic multi-user and multi-display settings as outlined above. However, as the generated graphical representation gets dynamically distributed to different output devices/displays, adaptation may become necessary to maintain visual effectiveness (see Figure 1 for an illustration). 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 is shown in Figure 2. Our adaptation pipeline has two steps. The first step estimates the visual effectiveness of a given visualization. The visual effectiveness is estimated if a new visualization has been distributed to a target device, or an existing visualization has been distributed to a new target device. The second step selects an appropriate adaption strategy if the effectiveness score indicates that the visual encoding of the data is poor. 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 representation often cause trouble in two important situations: (a) the distribution of a graphical representation to a target device which display size causes visual clutter, or (b) the distribution of a graphical representation to a target device which display size causes low visual densities. Visual clutter refers to the problem of a potential loss of information when too much data is displayed on too small displays. Many approaches have been proposed to reduce visual clutter in graphical representations of data (e.g. [2, 11]). 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 single-out subsets of the clusters as clusters. In [2], the authors also report that perception of the overall data distribution and local densities changes significantly under low visual densities.

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 a visualization 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 potentially interesting features of the data are communicated to the user. Note, automatic adaptation requires reliable thresholds specifying the perceptual boundaries on what constitutes an effective visualizations. This will require extensive user experiments to determine those thresholds, and to get a good feel about the perceived quality of adapted visualizations.

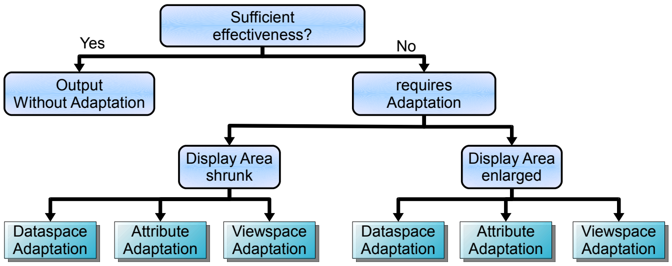

Fig. 2. The general device-based adaptation is based on an adaptation tree. The first step (root node) estimates the effectiveness of a graphical representation of data on a specific target device. The second step selects an appropriate adaptation strategy if the estimated effectiveness is poor.

3.2 Adaptation Options & Constraints

The visualization process is widely understood as a pipeline that consists of four data stages plus intra-stage and transformation (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 a *mapping* into the *view space* which are the last two stages. This visualization model yields starting points for adaptation in both the data space and the view space. 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 small display, the level of detail may need to be reduced. Here, adaptation in data space includes filtering or statistical aggregates of the data to reduce the amount of data items to be displayed. View space adaptation aims to reduce visual clutter (e.g., by employing density binning [10]). Contrary, more details can be shown on large displays (may down to raw data items).

Another question is which visual attributes and aggregations are eligible for adaptation. Visualization techniques encode attributes of the data as distinct visual variables. Thus, specific adaptation mechanisms over the set of attributes are required. Adaptation is also inherently task-dependent, i.e., what view space aggregation and abstraction levels are admissible for a given visualization goal? The identification of outliers, for example, states unique requirements for a visualizations. Providing adequate solutions to these questions is not trivial. To demonstrate the usefulness of our approach, we chose conventional scatterplots to derive particular procedures for device-based adaptations of scatterplots from the general adaptation scheme (see Section 4.

3.3 Infrastructure

A suitable infrastructure computes a graphical representation of the data for each output device in the environment; it utilizes the computing devices in the device ensemble of a smart room, and distributes the visual content to specific target devices according to the current requirements of the users.

We chose the service-oriented framework called SSC from [13] as the infrastructure for our experiments. It uses a visualization pipeline composed of pipeline operators which are distributed services. 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. Additionally, we implemented the following distribution mechanisms into SSC to facilitate testing:

- To enable distribution of a single visualization 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 space on each device, while the earlier data stages are processed only once for all devices[1].
- A distribution of a visualization across multiple displays is achieved by forking a second pipeline to render a finer grain details of a user-selected area on private devices. 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 APPLICATION EXAMPLE

We created a visualization pipeline based on the SSC framework to defines the necessary operators to create a conventional scatterplots for bivariate data. We assume the data points have class labels assigned, and that the visualization goal is to communicate a cluster structure to the user.

Scatterplots are then distributed to target devices in our smart environment using SSC's distribution mechanisms as described in Section 2. In the following, we discuss the implementation details for the different steps of the adaptation process as summarized by Figure 2.

4.1 Efficiency Evaluation

The first step is to estimate the effectiveness of a new or re-distributed scatterplot. For this purpose, we use two measures. Class consistency scores are used to estimate whether clusters start to mix in the scatterplot. Then the *visual density* of the scatterplot is computed to measure whether the display size of the target devices interferes with the effectiveness of the scatterplot. 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 of the convex hull of a cluster.

We conducted an informal pilot study for the two major adaptation scenarios (a) a scatterplot is distributed to a small display size and (b) a scatterplot is distributed to a large display size to determine reliable thresholds for the class consistency scores and visual density. The analysis of the results of our user study revealed the following important insights.

Low consistency scores ($60-80\%$) in conjunction with density values between 0.1 and 0.7 are a robust indication that the scatterplot display is saturated with points. At this point, clusters begin to mix (see Figure 1). This can be countered by suitable attribute adaptation, e.g. by using different shapes or by increasing the contrast between the 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).

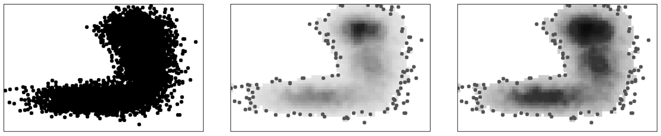
(a) (b) (c)

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.

Scatterplots 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 clusters are noise. Here, the user starts to single-out subregions of the clusters as clusters. Again, this can be addressed by increasing the contrast between clusters or by adjusting the scale of the scatterplot. To find appropriate adaptation strategies for this situation, however, is still an open research problem we did not yet pursue further.

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. The two axes of a scatterplot 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 2–D histogram of the data point distribution in view space. Every non-empty bin is represented as a rectangle in the adapted scatterplot with the each bin frequency is color-coded.

To faithfully extract the cluster structure, however, 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 the 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 scatter plot axes with respect to cluster center locations to determine a good binning. For this purpose, the view space of the scatterplot 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. If a bin contains more than one cluster center, this bin is then refined by increasing b_x (b_y) by 1. 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 (lower bound threshold) yield a good compromise between clutter reduction and a faithful reproduction of clusters on small displays. See Figure 3 for an illustration.

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. The focus point of each local fish-eye distortion is centered at the bins in the bin map that contain the cluster centers (see Figure 4(b)).

The increase in screen space around the local neighborhood of the cluster centers allows an additional sub-binning. The sub-binning factor is thereby proportional to the magnification factor, e.g. a magnification factor of 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 may improve visual separation of clusters (see Figure 4(c)).

4.3 Attribute Adaptation

In contrast to [10] which uses a linear color map, we use a logarithmic scale for the bin colors since on-linear color scale is better suited for skewed distributions of the bin frequencies [14]. Figure 3 illustrates the difference between a linear and a logarithmic color scale. In Figure 3(c), the non-linear color map assigns more gray levels to the low

---

[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.

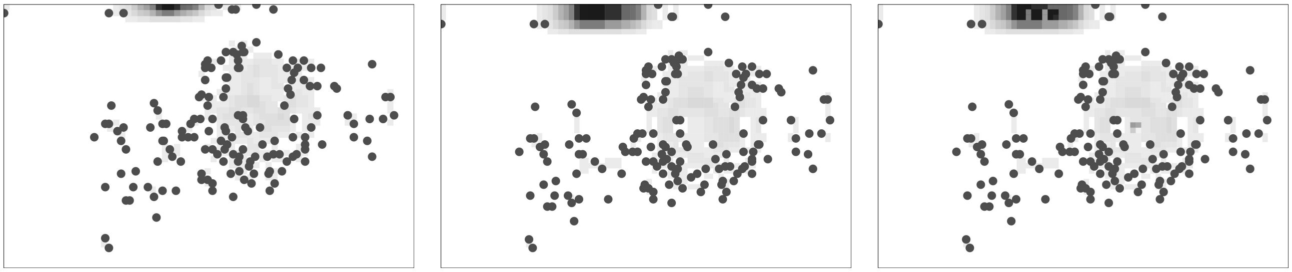
(a) (b) (c)

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

end of the frequency range, thus enhancing the visibility of the cluster centers.

4.4 Data Space Adaptation

A third strategy is to reduce the number of data items prior to their mapping to a small display. 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 and 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 graphical representations of data. In this paper, we proposed a general strategy to guide this adaptation based on the notion of visual effectiveness and a visualization pipeline model. As a proof-of-concept, we implemented corresponding adaptation mechanisms for conventional 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, smart device-based adaptation (i.e., minimal user intervention) requires further research. 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. Also, the problem of meaningful effectiveness measures requires more research. Consistency is applicable only for scatterplots and cluster structures. 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 content to specific target devices is still work in progress. Smart distribution requires detection of the current user situation followed by inferring 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 visualization. So far, we only considered a single visualization goal, namely communicating a cluster structure to the user. Note that the visualization goal resp. 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.