

A new Weaving Technique for Handling Overlapping Regions

Martin Luboschik Axel Radloff Heidrun Schumann

Institute for Computer Science
University of Rostock, Germany
{luboschik,radloff,schumann}@informatik.uni-rostock.de

ABSTRACT

The use of transparencies is a common strategy in visual representations to guarantee the visibility of different overlapping graphical objects, especially, if no visibility-deciding order is given (e.g., importance, depth). Alpha-blending, however, could generate new colors that are not specified by the given color scale and overlapping shapes may become difficult to be separated visually and the selection of specific elements would be difficult. In this paper, we present a new approach for representing overlapping regions: Instead of blending different colors, our weaving technique separates the original colors and shapes are easier to differentiate. Due to a deterministic weaving order, all overlapping objects are visible. We apply our approach to scatter plot visualizations to enhance the communication of overlapping clusters.

Categories and Subject Descriptors

H.5 [Information Interfaces and Presentation]: General; H.5.2 [Information Interfaces and Presentation]: User Interfaces—*Screen design*

General Terms

Design, Human factors.

Keywords

Weaving, information visualization, interaction, overlapping, transparency.

1. INTRODUCTION

A well-known problem of visual representations are overlapping graphical objects. Different methods are applied to solve this problem such as shrinking, jittering or distortions (see [6]). In particular, transparencies are used to make all colored objects contribute proportionally to the final rasterized image. However, when applying transparency some problems have to be considered (cp. [1]):

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AVI '10, May 25–29, 2010, Rome, Italy

Copyright 2010 ACM 978-1-4503-0076-6/10/05 ...\$10.00.

Interpretation of colors Combining transparent colors could generate new colors that are not defined by the used color scale. In this case, no semantic meaning can be associated with those regions. Otherwise, if colors of the color scale are generated, a predefined meaning will be associated.

Differentiation of colors The result of color blending depends on different aspects such as the used blending function, blending order and the blended colors. Extracting the underlying color fractions and blending parameters is difficult or impossible. Different settings could result in the same displayed color. To identify the 'topmost' color, adapted blending strategies (e.g., in [2, 19]) can be applied but the problem remains for the other colors.

Recognition of object shapes Transparent overlapping shapes visually generate geometrical intersection regions and hence new shapes (see Figure 1). Thus, the number of objects drawn at the display might be unclear and also which objects contribute to an overlapping region.

Figure 2 summarizes the described problems within a simple setup: Neither the colors nor the shapes are clearly distinguishable.

Besides visibility and perceivability of overlapping objects, there is a second closely related problem: interacting with overlapping graphical objects. In general, overlapped objects shall also be selected by picking. But it is often unclear which objects are currently beneath the pointing device and how to access them. This is a wide-spread problem that is present e.g. in design software or visualization applications.

A frequently used strategy is to allow picking only for the 'topmost' displayed object. To access *any* of the overlapping objects, a layer-selection mechanism (e.g., a list of layers) is generally integrated into the interface. This in turn is combined with additional interaction and implementation efforts. For example, advanced selection mechanisms (e.g., in [15]) for example modify the objects' position to de-overlap the objects. Such techniques lead to problems where positions encode concrete data values and thus should be fixed.

For both problems – displaying overlapping objects and the corresponding interaction – we propose the use of a technique that has been introduced as weaving (cp. [8]). Instead of blending overlapping objects, the objects are interweaved. The original weaving approach was not designed for representing arbitrary overlapping objects. Therefore, we extend

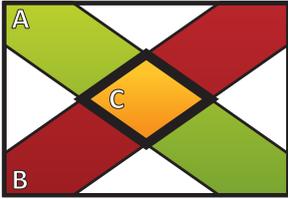


Figure 1: Three terms frequently used within this paper: C is the *overlapping region* of the *overlapping objects* A and B. The boundary of C is referred to as *overlapping border*.

that idea to use it as an alternative to transparent displays and moreover for interaction purposes. With our adapted weaving no new unspecified colors are displayed, the original shapes are kept and graphical objects do not disappear. Hence, the presented work proposes a new approach for handling overlapping objects.

The paper is organized as follows. In Section 2 we reflect on related work concerning transparency and weaving. In Section 3 we introduce two new weaving strategies with the aim to support the recognition of colors and shapes in overlapping regions. We apply the presented techniques to scatter plot displays to demonstrate the characteristics of our approach while representing information (Section 4.1). Moreover, we use our weaving approach for the selection of individual objects in overlapping regions (Section 4.2). A user study that evaluates the effectiveness of weaving during interaction is described in Section 5. This paper is concluded by a discussion of the proposed weaving in Section 6.

2. RELATED WORK

Overlapping graphical objects are a common problem in visual representations. In [4, 6] techniques have been summarized and categorized that address the problem of overlapping.

One of these techniques is transparency. It is simply applicable and a somehow 'natural' technique to handle overlapping. Thus, it is often applied in various scenarios (e.g., in [7, 13, 16]...). However, in [1] problems concerning the use of transparency are discussed. This motivates investigations that improve the use of transparencies and investigations that avoid transparency but maintain its strengths.

Techniques that improve transparency generally focus on color perception and try to make mixed colors separable. This is achieved either by modifying the colors to be blended (e.g., with techniques introduced in [10, 12]) or by modifying the blending function (e.g., in [2, 19]).

Techniques that avoid transparency try to generate images that are more expressive than transparent presentations by using alternative visual attributes. In [3] for example a method is shown, that uses colored grid cells in a scatter plots to visualize the contribution of different objects in overlapping regions without transparency. In [11] and [20] textures or stroke parameters respectively are used, to visualize different (overlapping) data attributes within an area.

An alternative approach called *color weaving*, allows for the visualization of overlapping attributes and has been presented in [18] to visualize multiscalar LIC-flow visualizations. This technique has been adapted for information visualization in [8] for the visualization of multiple attributes on map regions. Since our work is based on the idea of weaving, we will discuss this technique in the following section.

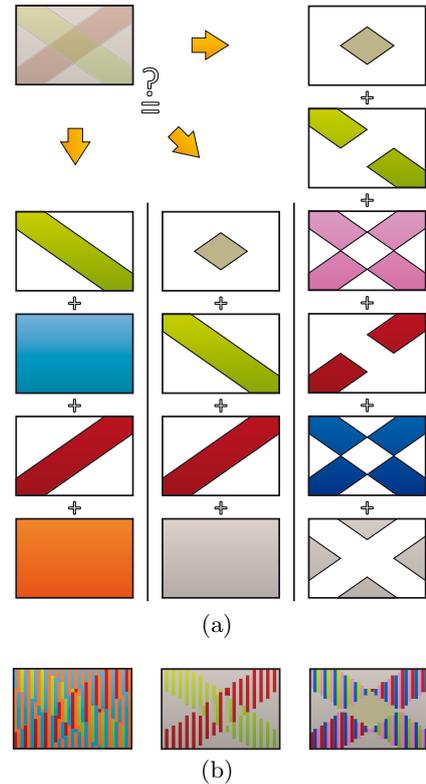


Figure 2: Three valid interpretations of a blended image (left top) to show the ambiguity of shape and color (a). With our weaving approach each of the different settings (each column) results in different images and thus avoids ambiguity (b).

3. REPRESENTING OVERLAPPING REGIONS BY WEAVING

In the following sections we present how weaving can be applied to represent overlapping regions. Therefore, we briefly introduce a formalism (Section 3.1), examine an existing weaving approach (Section 3.2), and finally introduce two new weaving strategies (Section 3.3).

3.1 The Method of Weaving

The basic principle of color weaving can be summarized as drawing the different, otherwise overlapping, colors next to each other [18]. This is done considering a given weaving strategy that defines a color pattern representing the overlapping objects within a delimited area. That distribution of a single pixel color information (composed of several objects) on a larger area is reminiscent of established halftoning and dithering-techniques. However, the aims of halftoning/dithering and weaving differ significantly: halftoning and dithering aim at generating a mixed perception of distributed colors whereas the idea of weaving is to visually separate the affected colors. In [8], based on this feature, weaving is used to visualize different data attributes within a single image area. We extend this approach by defining new weaving patterns. Before describing it, we briefly formalize the method of weaving in order to establish a common basis for specification and comparison.

Weaving assumes the n different objects to be given at n different unordered layers $L = l_1 \dots l_n$ ($n \in \mathbb{N}$). In contrast to blending that is generally a mathematical *ordered combination* of the different layers, in weaving the resulting color $c(x, y)$ is a *selection of one* color $c_{l_i}(x, y)$ from the n layers. Therefore, a selection function f_{sel} determines which layer l_i is chosen for the current position (x, y) :

$$\begin{aligned} c(x, y) &= c_{f_{sel}(x, y)}(x, y); \\ f_{sel}(x, y) &\in L. \end{aligned}$$

Since weaving generally depends on something to weave, it can be applied only in areas where overlapping occurs. Thus, weaving modifies the visual representation only within the corresponding regions. To realize that constraint the selection function is localized. We introduce the subset of *involved layers* $IL(x, y) \subset L$ which consists only of those layers that contain color information at position (x, y) :

$$\begin{aligned} IL(x, y) &= \{l_i \in L \mid c_{l_i}(x, y) \neq \text{empty}, 1 \leq i \leq n\} \\ f_{sel}(x, y) &\in IL(x, y). \end{aligned}$$

The concrete realization of f_{sel} determines the weaving strategy and in this way the visual output of the weaving approach – the weaving pattern.

We would like to explicitly point out that the method of weaving principally generates no new colors. Hence, all visible colors are well-defined in the used color scale and appear only at positions they are assigned to. Thus, when applying weaving, the ‘interpretation of colors’ problem is irrelevant and will, therefore, not be addressed in the remaining sections.

3.2 Random Weaving

The weaving strategy introduced in [8] handles map regions that are to be colored simultaneously according to six different attributes. For each pixel of a map region randomly one of the six colors is selected. Since that selection is done randomly we call this technique *random weaving*. The corresponding selection function is

$$f_{sel}(x, y) = \text{random}(IL(x, y)).$$

The benefit of *random weaving* as opposed to blending is a better recognition of the color and in this way of the attribute values. This result is emphasized within an encouraging user study [8]. Since that work’s focus was to make different attributes perceivable in one image, the problem of overlapping object contours was not examined.

We review the *random weaving* according to the problems of transparency shown in Section 1:

Differentiation of colors. The random distribution of different colors is similar to an equally distributed noise. This makes the identification of the different colors difficult since color perception depends on the size of uniformly colored connected areas. Especially for small noise blocks the separately displayed colors mix up in perception and result in a grayish looking area. The influence of block size on perception is also shown in [8] and thus larger blocks are proposed. Although the perception of a single color may be easier this way, the random distribution of different colors makes it difficult to recognize the number of involved layers at a glance.

Recognition of object shapes. Weaving improves the recognition of different overlapping objects. Thus, shape ambiguity as shown in Figure 2 should be reduced. But due to the random selection of colors, the overlapping borders become multicolored and shapes can be distorted (see Figure 3(a)). Enlarging the weaving blocks to ensure an improved color separation influences the shape recognition even more.

3.3 The Aligned Weaving Technique

To improve the differentiation of color, we extend the random weaving strategy by enlarging uniformly colored areas within the weaving patterns in a deterministic way and thus simplify color distinction. We align the weaving pattern to screen coordinates and thus assign the different layers to fixed pixel rows or columns.

In the following sections we introduce two weaving strategies that assign the individual layers l_i to separate pixel columns (a row-wise assignment is also possible) and illustrate how the problem of shape recognition and distinction can be addressed this way.

3.3.1 Stack-Aligned Weaving

To remove the grayish impression caused by a random distribution of colors we align the weaving pattern to the screen. For this, each pixel column is associated with a stack $\langle \dots \rangle_{\text{column}}$ of all layers L . Thereby the sequence of layers within the stacks circulates in consecutive columns: With n layers, the stack of pixel column 1 is $\langle l_1 \dots l_n \rangle_1$, of column 2 is $\langle l_2 \dots l_n l_1 \rangle_2$, of column 3 is $\langle l_3 \dots l_n l_1 l_2 \rangle_3$ and so on. This way, n different stacks are available and the $(n + 1)^{\text{th}}$ column is linked again to the first defined stack.

Our *stack-aligned weaving* ignores those layers of a current stack that do not contain an object at position (x, y) and selects the color of the topmost layer $l_{top} \in IL(x, y)$:

$$\begin{aligned} f_{sel}(x, y) &= l_{top} \mid l_{top} \text{ is topmost element in} \\ &\quad \langle \dots \rangle_{(x \bmod n)} \text{ that is } \in IL(x, y). \end{aligned}$$

Figure 3(b) shows the result of a (column wise) *stack-aligned weaving*. Although the consistent weaving pattern can be achieved in different ways, the stack metaphor illustrates the peeling of uninvolved layers.

Differentiation of colors. In *stack-aligned weaving*, the individual colors are easier to separate. Thus, the grayish impression found in *random weaving* is delayed. This is due to the enlarged uniformly colored areas. Moreover, the number of weaved objects can be easily determined by counting the differently colored columns within an overlapping region.

Recognition of object shapes. Due to a fixed assignment of layers to pixel columns, changes of shape are reduced to the appearing or disappearing of the corresponding color values. This causes a ‘growing’ of neighboring columns, if a layer disappears (see Figure 3(b)). Since this is the only visual change, overlapping borders may be difficult to extract. Moreover, the woven layers are actually of equal importance, but one of them becomes unintentionally more prominent due to a missing color of the neighbor column. Nevertheless, since the pattern width is fixed to n and layer-column assignment is fixed for the whole screen, it is easy to visually distinguish between present and absent layers. This way, it is easy to recognize the participating objects in overlapping regions.

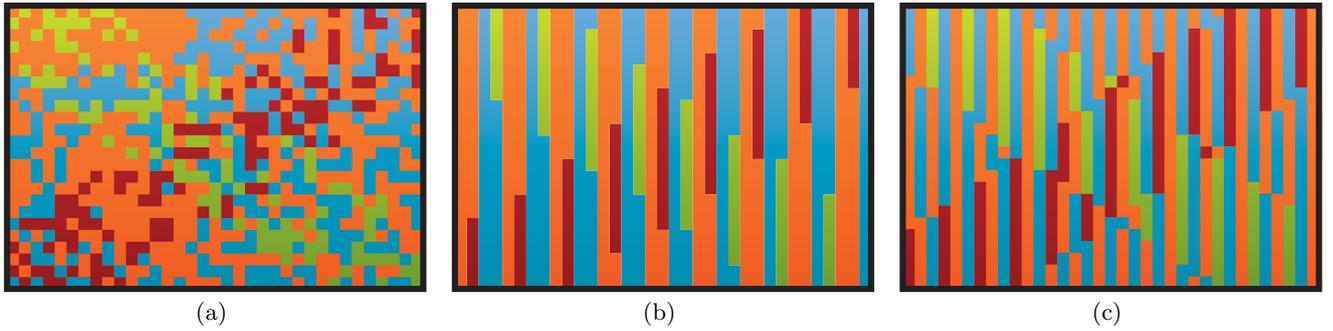


Figure 3: Detailed extracts from weaving the leftmost column in Figure 2. Principally no new colors are generated. In (a) *random weaving* is applied: Due to random distribution recognizing the colors is difficult and overlapping borders become distorted. In (b) *stack-aligned weaving* is applied: Colors are easier to distinguish but since edge information is perceivable only every n^{th} column, overlapping borders are difficult to recognize for small objects. Note that orange and blue columns are expanded in areas where the red and green columns are absent. In (c) *modulo-aligned weaving* is applied: Due to changing column alignments, overlapping borders are easier to recognize, colors are easy to distinguish but visual noise may be generated.

3.3.2 Modulo-Aligned Weaving

Stack-aligned weaving leads to an image, where edge information is only perceivable every n^{th} column. Hence, we present another weaving strategy to enhance the shape recognition. For this, the stacks per pixel columns do not contain all existing layers but only layers currently involved. To realize this weaving the $k = |IL(x, y)|$ involved layers are stored within a list $[[\dots]]_{\text{column}}$. The selection function is:

$$f_{\text{sel}}(x, y) = l_{\text{mod}} | l_{\text{mod}} \in IL(x, y) \text{ is} \\ m^{\text{th}} \text{ element in } [[\dots]]_x, m = (x \bmod k).$$

At column x , the m^{th} list entry is selected where $m = x \bmod k$. We refer to this method as *modulo-aligned weaving* that is shown in Figure 3(c). With the use of *modulo-aligned weaving*, the layers are assigned to their own columns similar to *stacked-aligned weaving*. In contrast to that, the column alignment changes from overlapping region to overlapping region and is not fixed for the whole screen.

Differentiation of colors. Although the uniformly colored regions are enlarged in contrast to *random weaving*, they are bound by the overlapping regions simultaneously. That means the uniformly colored regions are smaller than in *stack-aligned weaving*. Nevertheless, the individual colors are still distinguishable and no column becomes more prominent than others because of fixed widths. In contrast to *stack-aligned weaving*, the number of woven objects is more difficult to identify because the column alignment changes for every overlapping region. This may also lead to visual noise if the number of involved layers changes at high frequencies – for example if small objects are loosely distributed.

Recognition of object shapes. Since the layers are still assigned to pixel columns, the color information of one shape is visible every k^{th} column. But the alignment changes at overlapping borders and thus supports the recognition of these borders. Hence objects shapes are easier to perceive (see Figure 3(c)).

4. THE APPLICATION OF WEAVING

The weaving approach presented in this paper has been designed to depict overlapping objects. Our weaving is not designed for the application in scenarios where transparency is used as a visual attribute for, e.g., visualizing density (e.g., binning in parallel coordinates [14], density plots...) or deaccentuating uninteresting graphical objects (e.g., in [9, 17]...). In the following section we demonstrate the approach with the example of classified scatter plot displays to show its capabilities in presenting overlapping clusters. Thereafter, we use the approach for interacting with overlapping 2D objects.

4.1 Handling Overlapping in Scatter Plots

Information visualization generally deals with large amounts of data, often resulting in dense visualizations, frequently with overlapping graphical objects. Hence, transparencies are often used to visualize overlapping objects (e.g., in [4, 6, 13, 16]...) although ambiguous visualizations may result.

A basic and well known visualization technique is the scatter plot visualization. It is generally used to visualize the overall data distribution and the distribution of clusters. An ongoing topic concerning scatter plots is the distinction of clusters in overlapping regions and the identification of the clusters shape. Therefore, the common strategy is to assign separate colors to the cluster members. Since the data amount influences the amount of overlapping dots, the cluster shapes are often difficult to perceive (see Figure 4(a)). Moreover, the rendering order decides which dots are visible and 'topmost'. A rendering order update (e.g., after deleting or inserting new data points) may lead to significantly changing displays.

Therefore, we apply our new weaving technique to communicate the characteristics of classified data in the specific case of scatter plot displays. Our example is 10-dimensional climate dataset of approximately 40,000 data items and 8 clusters provided by Potsdam Institute for Climate Impact Research. For comparison we also applied transparency and *random weaving*. Figure 4(b) shows the transparent image with generated colors already defined within the color scale.

Hence, transparency manipulates the number of perceivable clusters and the shape of the clusters is not communicated well either. The random distribution of colors while applying *random weaving* results in effects known from dithering: the colors mix during perception the result somehow similar to transparency.

Scatter plots generally produce images with visual changes of high frequency. Thus, we applied the *stack-aligned weaving* as it generates a more smooth visualization and the cluster shape is easier to perceive due to less noise. Although the plot is dense, the shape of the visualized clusters and the dot colors (cluster membership) are perceivable (Figure 4(d)). No cluster becomes invisible due to overlapping. Since weaving applies only to overlapping regions, all dots outside these regions are clear to perceive. The weaved visualization of eight clusters is shown in Figure 4(d).

Some scatter plot visualizations alternatively approximate each cluster with a cluster-encapsulating hull (e.g., in [5, 16]) to improve the recognition of outliers and cluster shapes. As a result of overlapping complex shapes, several different (and even complex, see [16]) overlapping regions are generated. Thus, also in this case transparency may lead to ambiguous cluster shapes and numbers (see Figure 5(a)). The application of our weaving approach makes clear, which cluster shape contributes to which visible overlapping region. Thus the different clusters and their borders are distinguishable (see Figure 5(b)). Since the visualized hulls generate visual changes of lower frequency, the *modulo-aligned weaving* can be applied and results in more detailed edge information.

We applied the weaving approach also to other visualization techniques like parallel coordinates with similar successful results. Beside the presented benefits, weaving introduces artifacts that are akin to the moiré-pattern and result from frequency changes between neighboring overlapping regions. Although this artifacts are not visually pleasing, they influence color perception and shape recognition only marginal. In that the advantages of weaving overbalance aesthetical drawbacks. Nevertheless, countermeasures like alternating weaving patterns and color sorting may be helpful and are subject-matter of future work. Halftoning and dithering use local information to achieve convincing results and hence locally adapted weaving patterns may also be advantageous.

4.2 For Interaction

Representing overlapping objects is one problem, interacting with those objects is a further important step. Information visualization comprises interaction as an essential component (cp. [21]). In that, the *selection* of objects is important to modify parameters, to highlight data items in different views or to get detail information. Selecting single objects in 2D visualizations with overlapping graphical elements is difficult: It is generally unclear to the user how many objects are currently selectable (beneath the pointing device) and how the different objects could be accessed. A common strategy is to pick the 'topmost' element or to cycle through all selectable objects. Advanced mechanisms provide a two-phase selection (e.g. in [15]): The first step reveals the objects to select (e.g. by distortion, updating a list ...) and the second determines the selection. These mechanisms are generally more precise but require additional effort while interacting. In contrast selecting the 'topmost' object is simple but restricting the options of a user.

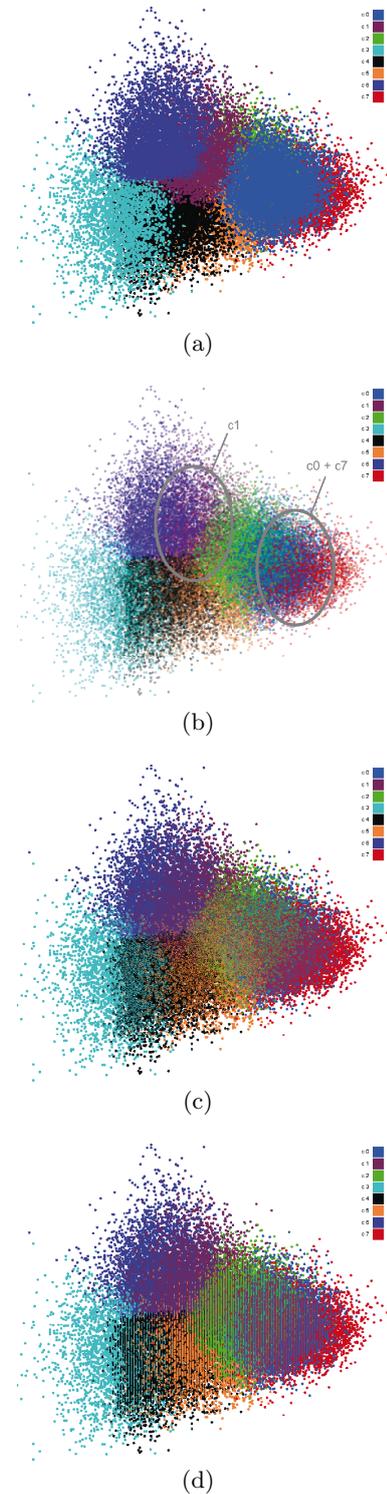


Figure 4: A standard scatter plot visualization of about 40,000 data items with 8 clusters. The clusters' shapes are difficult to perceive due to overlapping (a). Transparencies generate colors, that are already within the color scale resulting in ambiguous clusters (b). *Random weaving* generates the same mixed color impression during perception (c). The *stack-aligned weaving* keeps the colors separable and moreover reveals the clusters' shapes (d).

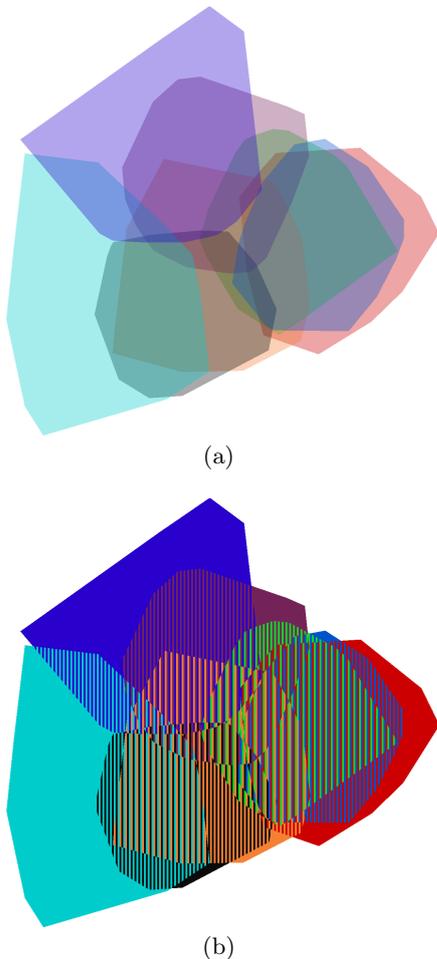


Figure 5: Cluster-encapsulating hulls to visualize the clusters' extent. Blending generates several shapes of different new colors (a). With *modulo-aligned weaving* the clusters become distinguishable and the user is facilitated to identify the shapes contributing to the single overlapping regions (b).

The basic problem within graphical interfaces consists of accessing the different overlapping objects by visual properties. Similar to the representation problem discussed, this requires a distinction of the overlapping objects. This is why weaving offers a new and effective alternative for picking any object from a set of overlapping objects.

In contrast to existing techniques, weaving distributes the single selectable objects to distinct neighboring pixel columns. Thus, each column represents only one associated object (or layer) and not the whole set. This is the key for using weaving for interaction purposes: Picking one of those columns with the pointing device displays the corresponding objects (e.g. the contour) and selects this object unambiguously. We tested this approach by selecting different clusters within the central overlapping region of the scatter plot display. In most cases it was easy to interact with different objects without annoying false selections. Moreover, the effort for interaction was reduced: The user selects objects directly like picking the 'topmost' but at the same time he is enabled to select from *all* layers (see Figure 6).

5. EVALUATING INTERACTION

Qualitative feedback from designers encouraged us to evaluate the aligned weaving techniques by a user study. The goal of our study was to get a first impression of time and correctness behavior of our interaction weaving approach compared to well established interaction techniques. Therefore we presented different overlapping shapes to a user. The users' task was to identify, pick and drag a given shape to a target region.

We performed a controlled user study with a fixed setting (display resolution: 1920×1200 , distance to screen: 0.8m, fixed light conditions). The test was accomplished in a window with a size of 500×500 . We presented randomly 5 or 8 overlapping shapes in the center of the window. The shapes were generated randomly and drawn with a randomly chosen color from a fixed color table. This setup was chosen to avoid learning effects. Within a target region we showed the shape to be moved by the user. After moving the correct shape into that target area it disappeared and the next shape to be moved was displayed. After completing one screen containing 5 or 8 shapes the next screen was presented. Within a two second break the next interaction technique was announced. For each screen the interaction technique was chosen randomly, each technique was used 10 times and applied 10 times per complexity (5 or 8 shapes).

We compared standard transparency with picking the topmost, a layer selection mechanism using a layers palette (like in e.g. Adobe Photoshop), *splatter* [15] and our weaving approaches. We had the following five hypotheses for the user study:

1. Transparency with picking the topmost will result in the least time to completion due to familiarization with this interaction but with the highest error rate.
2. Layers palette is the technique with the smallest error rate but with the highest time to completion.
3. The aligned weaving techniques have a time to completion similar to transparency and an error rate similar to layers palette.
4. The error rate of *splatter* is small but the time to completion is nearly as high as the time of layers palette.
5. In general: the less the time to completion the higher the error rate and vice versa. With aligned weaving both values can be kept small.

10 participants (non-experts, 4 female, 6 male, age 25–36) completed the test and thus 1000 samples were tested; 200 samples for each interaction technique. We recorded the time to completion and the false movements (error rate, movements that do not end inside the target area and shapes falsely moved into the target area). To get a first impression, a straight forward analysis has been performed. We disregarded extreme outliers ($3 \times IQR$) to get a meaningful average. The results are shown in Figure 7.

The average time and error rates we found differ from our as expectation, except of the layers palette. Transparency with picking the topmost was such laborious, that even this familiar technique caused a low performance concerning time. The unexpected values found in *splatter* are due to unexpected behavior of the participants while using this technique. In most cases, users were not aware of the

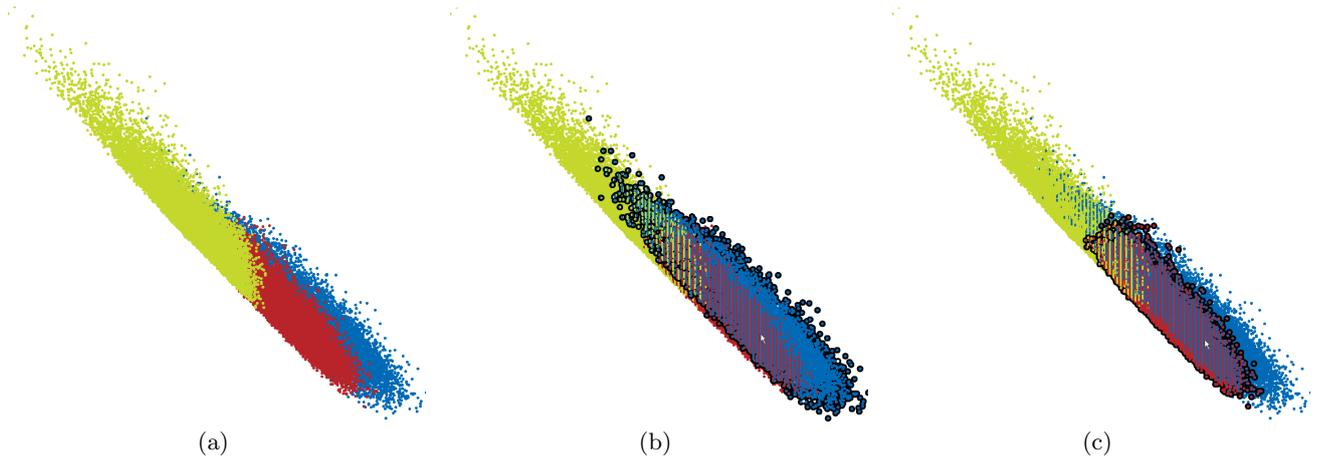


Figure 6: A standard scatter plot with 3 overlapping clusters. The shape of the red and blue cluster are not recognizable, due to overlapping (a). With applied *modulo-aligned weaving* the clusters are visible and moreover can be selected by their color fragments generated by weaving (b) and (c).

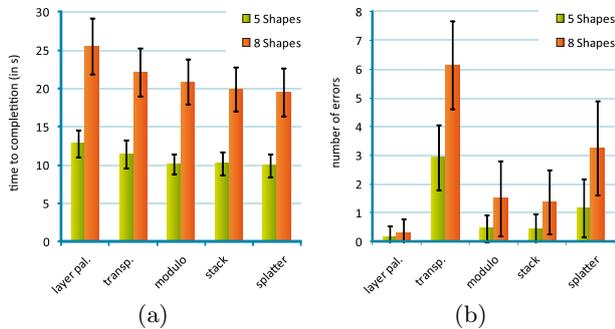


Figure 7: The results of our user study with time to completion (a) and belonging errors (b). Our aligned weaving approaches show medium times to completion at low error rates and hence pose a compromise.

possibility to *splat* the overlapping objects – despite of a familiarization phase. Thus, they dragged the solid shapes aside to find the correct shape resulting in high error rates. On the other hand, *splatter* was the fastest technique. Assuming that *splatter* was commonly used like ‘pick first’, this may show the distracting ambiguity feature of transparency – that shows worse results. Our weaving approaches performed nearly as expected. Surprisingly, *modulo-aligned weaving* was generally slower than *stack-aligned weaving* although the picking areas are smaller in the latter. The participants even reported that picking was difficult because of the thin weaving patterns in *stack-aligned weaving*. This result may be due to the additional distracting visual noise in the *modulo* approach (see Section 3.3.2).

Concerning time to completion, the standard deviation of our weaving approaches was marginally the lowest. Concerning error rates, only the layers palette performed constantly and nearly independent of complexity.

The results show that the aligned weaving could be a good compromise between time to completion and error rate. Furthermore, due to the informal interviews the stack-weaving

has been proven to be a usable technique to guarantee the visibility of covered objects. As aforementioned, this study’s aim was to get a first impression. We found, that comparing interaction techniques in step with actual practice while staying equitable to the different approaches is difficult.

6. CONCLUDING REMARKS

For sure, visual representations could benefit from applying weaving techniques. Due to the characteristics of weaving, only those colors that are defined by the used color scale are presented. Since our new weaving techniques define larger areas of equal color, also the differentiation of those colors is improved. In particular, the recognition of object shapes could be improved by our *modulo-aligned weaving*.

Additionally, the selection of individual objects in overlapping regions is improved: Picking a colored region within the weaving pattern selects the associated object and in this way supports the selection procedure. We would like to mention, that the improved selection mechanism in overlapping regions resulted in positive feedback from designers. They work regularly with overlapping objects inside their specific design-software. A first comparative study confirmed the effectiveness of our weaving selection.

The weaving technique is based on the combination of layers. This is a well-established method (e.g. in GIS) and thus it can easily be adapted for the use in arbitrarily 2D-applications.

However, the weaving technique as well as other techniques (e.g., transparency, shrinking, jittering, distortion (see [6])) has some constraints. The column width used during weaving has to be carefully determined: Enlarging the width delays the unwanted perceptually mixing of neighboring colors if too many layers are weaved. On the other hand, thin columns assure the visible representation even of small objects. This problem can be partially solved by the modulo weaving as it considers only the currently overlapping objects. Thus the number of layers to be represented by the weaving pattern is reduced.

Moreover, shape recognition can be influenced by weaving. Thin objects, that are parallel to the weaving pattern, could become invisible. For those objects the weaving pattern can

be modified to the use of rows instead of columns. In general, locally adapted patterns can be used, but the specification of those patterns requires the analysis of the scenario to be displayed. The use of metrics known from halftoning may be helpful.

Future investigations will examine the weaving approach as an interactive tool (e.g., as a lens) in 2D representations and moreover we are looking forward to investigate 3D environments and objects that are colored not uniformly. We are investigating countermeasures to reduce artifacts and to enhance the aesthetics of weaving. A study concerning only the representation of overlapping objects comparing transparency and weaving is outstanding.

7. ACKNOWLEDGMENTS

Axel Radloff is supported by a grant of the German National Research Foundation (DFG), Graduate School 1424, Multimodal Smart Appliance Ensembles for Mobile Applications (MuSAMA). We thank Dr. Thomas Nocke from the Potsdam Institute for Climate Impact Research for providing data.

8. REFERENCES

- [1] P. Baudisch and C. Gutwin. Multiblending: Displaying overlapping windows simultaneously without the drawbacks of alpha blending. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '04)*, pages 367–374, 2004.
- [2] J. Chuang, D. Weiskopf, and T. Möller. Hue-preserving color blending. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1275–1282, 2009.
- [3] H. Dolfig. A visual analytics framework for feature and classifier engineering. Master's thesis, University of Konstanz, 2007.
- [4] G. Ellis and A. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–1223, 2007.
- [5] 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*, 14(6):1141–1148, 2008.
- [6] S. Few. *Solutions to the Problem of Over-Plotting in Graphs*. Perceptual Edge - Visual Business Intelligence Newsletter, September/October 2008.
- [7] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of IEEE Visualization '99 (VIS'99)*, pages 43–50, 1999.
- [8] H. Hagh-Shenas, S. Kim, V. Interrante, and C. Healey. Weaving versus blending: A quantitative assessment of the information carrying capacities of two alternative methods for conveying multivariate data with color. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1270–1277, 2007.
- [9] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing of extended parallel coordinates. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, pages 127–134, 2002.
- [10] C. G. Healey. Choosing effective colours for data visualization. In *Proceedings of IEEE Visualization (VIS'96)*, pages 263–271, 1996.
- [11] C. G. Healey, L. Tateosian, J. T. Enns, and M. Remple. Perceptually-based brush strokes for nonphotorealistic visualization. *ACM Transactions on Graphics*, 23(1):64–96, 2004.
- [12] G. Kindlmann, E. Reinhard, and S. Creem. Face-based luminance matching for perceptual colormap generation. In *Proceedings of IEEE Visualization (VIS '02)*, pages 299–306, 2002.
- [13] K. T. McDonnell and K. Mueller. Illustrative parallel coordinates. *Computer Graphics Forum*, 27(3):1031–1038, 2008.
- [14] 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.
- [15] G. Ramos, G. Robertson, M. Czerwinski, D. Tan, P. Baudisch, K. Hinckley, and M. Agrawala. Tumble! splot! helping users access and manipulate occluded content in 2d drawings. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI'06)*, pages 428–435, 2006.
- [16] T. Schreck, M. Schüßler, K. Worm, and F. Zeilfelder. Butterfly plots for visual analysis of large point cloud data. In *Proceedings of the 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG'08)*, pages 33–40, 2008.
- [17] C. Tominski, P. Schulze-Wollgast, and H. Schumann. 3d information visualization for time dependent data on maps. In *Proceedings of the 9th International Conference on Information Visualisation (IV'05)*, pages 175–181, 2005.
- [18] T. Urness, V. Interrante, I. Marusic, E. Longmire, and B. Ganapathisubramani. Effectively visualizing multi-valued flow data using color and texture. In *Proceedings of IEEE Visualization (VIS'03)*, pages 115–121, 2003.
- [19] L. Wang, J. Giesen, K. T. McDonnell, P. Zolliker, and K. Mueller. Color design for illustrative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1739–1754, 2008.
- [20] C. W. William, W. Emigh, G. Liu, R. M. Taylor, J. T. Enns, and C. G. Healey. Oriented texture slivers: A technique for local value estimation of multiple scalar fields. In *Proceedings of Graphics Interface (GI2000)*, 2000.
- [21] J. S. Yi, Y. ah Kang, J. Stasko, and J. A. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, 2007.