Context-Preserving Visual Links

Markus Steinberger, Manuela Waldner, Marc Streit, Alexander Lex, and Dieter Schmalstieg

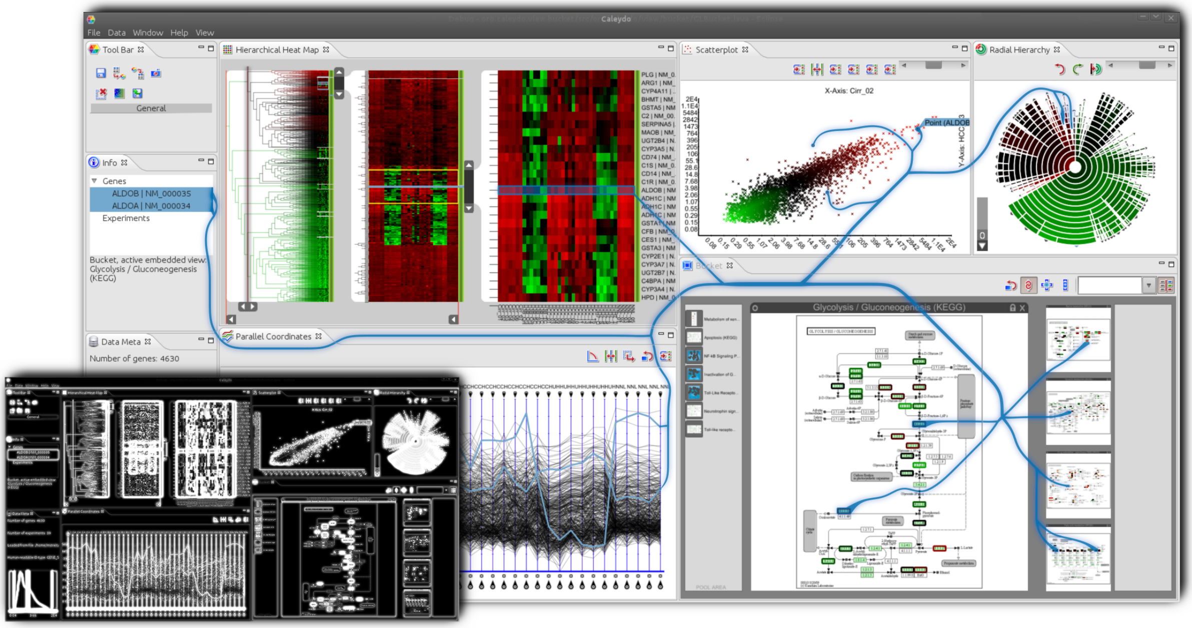


Fig. 1. Context-preserving visual links connecting related entities in an information visualization framework. Note that the links avoid occlusion of salient regions and, therefore, choose their routes along the window borders. The small inset shows the visual saliency derived from the base representation.

Abstract— Evaluating, comparing, and interpreting related pieces of information are tasks that are commonly performed during visual data analysis and in many kinds of information-intensive work. Synchronized visual highlighting of related elements is a well-known technique used to assist this task. An alternative approach, which is more invasive but also more expressive is visual linking in which line connections are rendered between related elements. In this work, we present context-preserving visual links as a new method for generating visual links. The method specifically aims to fulfill the following two goals: first, visual links should minimize the occlusion of important information; second, links should visually stand out from surrounding information by minimizing visual interference.

We employ an image-based analysis of visual saliency to determine the important regions in the original representation. A consequence of the image-based approach is that our technique is application-independent and can be employed in a large number of visual data analysis scenarios in which the underlying content cannot or should not be altered.

We conducted a controlled experiment that indicates that users can find linked elements in complex visualizations more quickly and with greater subjective satisfaction than in complex visualizations in which plain highlighting is used. Context-preserving visual links were perceived as visually more attractive than traditional visual links that do not account for the context information.

Index Terms—Visual links, highlighting, connectedness, routing, image-based, saliency.

◆

1 INTRODUCTION

A common task of modern knowledge workers is the identification of related pieces of information that may be scattered over a display screen. These pieces of information may be contained in a single view or distributed among several views or windows. In software development, for example, it is common to display multiple views, each of which shows zero to multiple instances of the same variable or method in a different context. During debugging, a developer may constantly switch between different views, first checking the context of a variable in one source file, then controlling the runtime value of the variable, and lastly looking for other instances referencing the same variable. Examples from the visualization domain are easy to find. Nearly every available multiple coordinated view system links the data shown in separate views. Figure 1 shows an example of a multiple coordinated view system that might be used by a biologist to explore gene expression values in different views and different contexts.

Real-world systems support such tasks by visually enhancing items related to the user's currently selected element. In this way, they actively guide the user's attention to these *highlighted* elements and make it easier for the user to distinguish important from unimportant information. Typical highlighting techniques use features that are efficiently perceived by humans. Examples are changing the color of an element or drawing a colored frame or background around it. The Eclipse Java IDE highlights all occurrences of a variable by drawing a colored frame around each occurrence when one is selected.

- *Markus Steinberger, Manuela Waldner, Marc Streit, Alexander Lex, and Dieter Schmalstieg are with Graz Univeristy of Technology, E-mail: {steinberger—waldner—streit—lex—schmalstieg}@icg.tugraz.at*

Manuscript received 31 March 2011 accepted 1 August 2011; posted online 23 October 2011; mailed on 14 October 2011.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

However, the power of traditional highlighting techniques is limited. For example, the colors used to visually discriminate items of interest may be difficult to perceive if the interface is cluttered with information presented in color. Similarly, frames may be too subtle to stand out from a cluttered display, and modifications to the surrounding information may be either too subtle to be perceived or too invasive, thereby reducing the legibility of contextual information. In addition, color is sometimes used as a visual variable to encode other information.

Because the use of colored frames as highlighting technique is problematic, several authors have recently investigated the use of *visual links* as an additional visual cue to show relations between multiple entities. We define visual links as *continuous shapes such as connection lines, curves, or surfaces which connect or surround multiple related pieces of information, thereby augmenting a base representation*. A base representation can be any kind of visual representation and includes images, visualizations, composed representations, multiple view systems, and multiple application windows.

The use of visual links is limited by the additional visual clutter they introduce, which can potentially occlude other information. To alleviate this problem, bundling strategies [6, 8, 9] have been employed to reduce the number of overall links on screen. However, bundling without the incorporation of knowledge of the visual information that is already present may still lead to occlusion of underlying content. Thus, the clever routing of links considering the information underneath has been identified as a potential direction for further research [23].

The primary contribution of this paper is a technique for **routing and adjusting visual links** that considers the information present in the original scene. Our technique minimizes the occlusion of important information when using visual links and makes visual links stand out from their surrounding base representation. To achieve this goal, we perform an image-based evaluation of the base representation. As a measure of how much a region stands out from its surrounding image regions, we apply an algorithm that computes visual saliency [10]. We use this measure for routing visual links, such that a minimal amount of (perceptually) important information is occluded. In addition, we optimize the amount of visual discrimination between the visual links and the base representation to increase the visibility of the links.

As a secondary contribution, we conducted a controlled **experiment that compared the use of three different highlighting techniques** to support typical visual search tasks of knowledge workers: simple color-coded or frame-based highlighting, conventional (straight) visual links, and context-preserving visual links. The results of this experiment show that visual links improve performance on tasks that involve finding related pieces of information on a cluttered display. The results also indicate that context-preserving visual links are the most visually attractive of the three techniques tested.

2 RELATED WORK

Highlighting using color, frames, or even synchronized blinking [22] is a cue-based focus-and-context technique [1]. In information visualization systems, the synchronized highlighting of multiple items is referred to as *brushing* [15]. Other cue-based focus-and-context techniques draw the user's attention to a focus object by reducing the visibility of the surrounding information. This goal is often accomplished by decreasing saturation [32], brightness [11, 32], or sharpness [12]. Zhai *et al.* [32] demonstrated that these methods are strongly preattentive and do not strongly affect the readability of the context information. Veas *et al.* [26] showed that image sequences can be modulated imperceptibly while still increasing the likelihood that users are looking at a particular item. With all of these techniques the highlighting of the focus object can be very subtle; in fact, the user is sometimes even not consciously aware of it. However, Hoffmann *et al.* [7] showed that users committed more errors and were more annoyed by a technique in which the context regions were darkened than by techniques in which the focus object was highlighting using frames or trails. Thus, explicit presentations of focus objects may be more desirable in the context of visual analysis.

All of the abovementioned focus-and-context techniques support the user in his or her visual search task of finding all occurrences of a selected entity within a single view or across multiple views. Previous research has shown that the performance and accuracy of visual search depends on a number of parameters, that include the number of target items and distractors [30] and the strength of basic features to support preattentive texture segmentation [25]. A target that stands out due to basic features such as color, motion, or orientation, can be discovered more efficiently [31].

Discovering multiple highlights in an information-dense and heterogeneous image is, in many cases, a 'serial' search [25]. In such searching a user must scan the whole image to make sure he or she did not miss any elements, which is tedious and time-consuming. By using visual links, these fragmented pieces of information are integrated into a single meta-visualization. Studies have shown that connectedness between elements is an even stronger grouping principle than proximity, color, size, or shape [18, 33]. This result means that when two items are connected, they are perceived as a group, irrespective of their other visual variables.

Another argument for use of visual links is that they can connect elements outside the small active visual field [29]. This fact is especially important when considering the trend to large, high-resolution displays that often significantly exceed the visual field. It has been shown that visual links are more efficient than simple frames in guiding attention to a single highlighted element on a large display [7].

Examples of the use of visual links in this context can be found in the work of Risch *et al.* [20], in the tree map overlays with graph links of Fekete *et al.* [4], in *Semantic Substrates* [23], in Streit *et al.*'s connections between pathways [24], in *Bubble Sets* [3], and in the *Matchmaker* technique [14]. In *VisLink* [2], generalized visual links are used to connect different visualization techniques, an approach also taken by Caleydo [13]. *Visual Links across Applications* [27] extends visual links to connect arbitrary desktop content. However, none of these visual linking techniques considers the base representation. Thus, depending on the visual content of the base representation, the use of visual links may lead to occlusion of valuable content or poor recognizability of the links themselves.

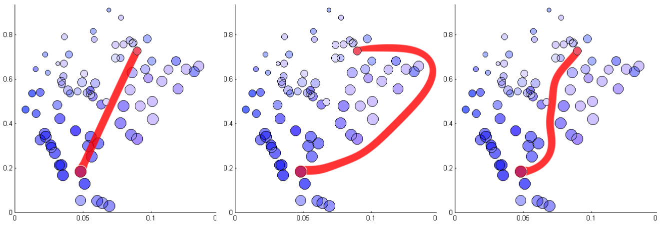
While straight connection lines are the most common form of links, curves, ribbons, and surfaces can also be used. The decision regarding which style to use depends on the content of the visualization. In visualizations with mainly symmetric content, curved connection lines stand out and are, therefore, easier to discriminate [7]. In addition, elements connected by smooth (curved) lines are more easily perceived as related [29, p.193]. Because they are smoothly routed around important content, our context-preserving visual links address these considerations.

3 OVERVIEW

Context-preserving visual links aim to address two major disadvantages of visual links. First, because visual links are a visually prominent highlighting technique rendered on top of the base representation, their use can lead to the occlusion of important content. Second, if the base representation is visually cluttered, visual links may be difficult to perceive. We address these problems by taking an image-based description of the base representation into account when constructing and rendering visual links. To preserve important context information, we introduce a routing algorithm to reduce overlap with important base representation content. We formulate visual links routing as an optimization problem in which four optimization criteria should be satisfied. These criteria are as follows:

- **O1** Visual links have **minimal length** and thus can be followed efficiently
- **O2** The amount of **occluded information** in the base representation **is minimal**
- **O3** Links are **visually distinguishable** from the base representation
- **O4** Multiple links in close proximity are **bundled** in such a way that context occlusion can be further reduced.

Because some of these criteria interfere with each other, finding an optimal route for visual links is a challenging task, even for simple cases. Consider the example of two related points at the opposite periphery of a scatterplot. The shortest link between two such points is a straight line across the scatterplot (Figure 2(a)). However, drawing this line leads to occlusion of a series of points in the visualization. On the other hand, bending the link around the entire scatterplot avoids occlusion at the cost of a long connection line (Figure 2(b)). Thus, the goal is to find a good trade-off among the four optimization criteria, as shown in Figure 2(c).



(a) (b) (c)

Fig. 2. Routing of context-preserving visual links: (c) trade-off between (a) occlusion of the base representation's content and (b) link length.

Despite optimized routing, it may not be possible to entirely avoid occlusion of important information in the base representation, which leads to two potential problems. First, if the links do not visually stand out, they may be difficult to perceive; second, occlusion leads to problems with the readability of the base representation's content. If occlusion cannot be avoided, we analyze the base representation and either adjust the links' appearance to the base representation or locally modulate the base representation around the links. The aim of this procedure is to make links visually distinct from the base representation and to ensure legibility of the text below the links.

In order to take into account the visual context of the base representation, we analyze the rendered image that forms the output of the visualization pipeline. This image-based analysis allows for a flexible employment of our technique in a variety of applications such as visualization frameworks, image viewers, or even the window manager. In Section 4, we will demonstrate how a thorough analysis of the base representation's rendered image can be used to ensure context-preserving routing of visual links; in Section 5 we will show how the visual representation of the links can be automatically adjusted to the background.

4 CONTEXT-PRESERVING LINKS ROUTING

In this section, we discuss the steps taken to meet the aforementioned optimization criteria for context-preserving routing of visual links. The algorithm begins by collecting information about the highlight regions (the set of image regions, each containing one of the possibly related items that should be emphasized) and the base representation (Section 4.1). This information is merged into a common *penalty map* that provides a unified image-based description of penalties for the links' routes on the base representation (Section 4.2). Based on this penalty map, the *routing* algorithm aims to find the shortest links with minimal penalty (Section 4.3).

4.1 Input Data

For any type of visual links routing, a basic set of input parameters (minimally, the position, shape, and color of the highlight regions and a specification of how they should be interconnected) is required. To describe interconnectivity, we use the notion of clusters. Multiple nodes forming a cluster will have a link originating from every highlight region; these links meet at one common point, the cluster point $\mathbf{p}_c$. Because the location of the cluster point $\mathbf{p}_c$ is initially unknown, determination of its optimal location is subject to the routing algorithm. Multiple clusters can be linked by connecting their individual cluster points as illustrated in Figure 9.

For conventional link routing, which does not take the base representation into account, the information about the appearance and location of the highlight regions and their connectivity is sufficient. However, to ensure context preservation, our routing algorithm additionally requires a description of the base representation, that is, the rendered image which is given by the output of the visualization pipeline.

4.2 Penalty Map

To generate context-preserving visual links using our technique, knowledge of which image areas are allowed to contain visual links and which regions should be avoided is necessary. We derive the penalty a link will receive for covering a certain point from the base representation image combined with information about the highlight regions and concurrently present linksets.

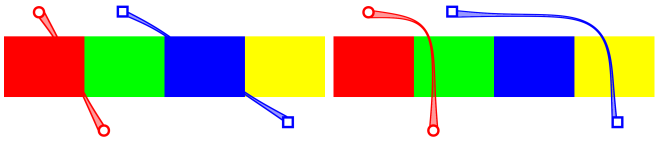
4.2.1 Importance Map

To satisfy the optimization criterion O2 presented in Section 3, we require a description of pixel-wise importance in the base representation. We, therefore, utilize the concept of importance maps [28]. Importance maps encode the importance of each pixel in an input image using a model of saliency-based visual attention [10]. Models of saliency take the biological properties of the human visual system into account [10] and can predict the likelihood of a user looking at a particular item [21]. This strategy assigns high importance to areas that are visually prominent compared to their surroundings; it works irrespectively of the background color. Using this technique, color disparities, intensity changes, and high-frequency textures will be assigned a high importance. Because important information is usually encoded using strong visual variables such as these, we found this perceptual measure sufficient to reliably identify important image content. Figure 1 shows the resulting importance map of a multi-view visualization system.

For our routing algorithm, the importance map is the foundation for occlusion avoidance. A low image importance (and therefore a low penalty) means that a link routed over this region will not cover any crucial information, whereas covering a region with high importance values implies a high probability of information occlusion.

4.2.2 Color Similarity

To satisfy O3, we additionally encode penalties into our map when visual discrimination of links with respect to the base representation cannot be guaranteed. As a measure of visual discrimination, we determine the color similarity between the predefined link color and the base representation's color at respective pixels. We analyze the colors in the CIE $L^*C^*_{ab}h_{ab}$ color space and determine color similarity as the inverse of the ΔE^*_{94} distance metric [19]. A high color similarity means that the link will hardly be distinguishable from its vicinity, while a low value indicates that link and context are sufficiently different. Thus, the higher the similarity, the higher the penalty added to the penalty map at the respective pixel. Figure 3 illustrates how color similarity influences links routing.



(a) (b)

Fig. 3. (a) Ignoring the link's color for route selection produces links that might be difficult to distinguish from the input image. (b) An additional penalty describing the similarity between the link's color and locally appearing image color is used to avoid areas of similar color.

Determining color similarity is necessary only if the link color is predefined and should not be modified. However, because connectedness is a strong visual variable, the color of links often does not encode any meaning. Thus, an alternative to routing links around similarly colored image regions is adjusting the links' color to the prevalent background color, as discussed in more detail in Section 5.

4.2.3 Highlight regions

So far, only visual features derived from the base representation have been considered in the penalty map. However, the highlight regions themselves are similarly important and should not be covered by visual links. In particular, if highlight regions are small, as in the example of Figure 4, they can be easily overlooked when occluded by a link.

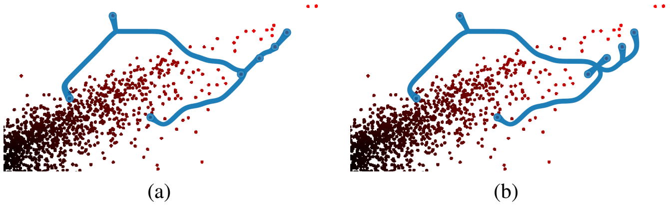

(a) (b)

Fig. 4. If other highlight regions are not considered during the route-finding process, undesired effects are generated. (a) Small highlight regions are occluded by the link and can be hardly recognized. (b) We add a blurred version of every highlight region to the penalty map to make links avoid other highlight regions, if possible.

We, therefore, increase the penalty at and near every highlight region by blurring its shape and inserting it into the penalty map with a high value. The blurring factor can be used to control the minimal distance between links and highlight regions. This effect is shown in Figure 9(a), in which regions of high penalty around highlight nodes are clearly visible.

4.2.4 Multiple Linksets

Similarly to the above-described avoidance of highlight regions, the routing algorithm should also take into account preexisting linksets. For instance, multiple linksets are required, if two or more set relationships should be emphasized concurrently [3] or if the system allows multi-selections. Treating these sets completely independently can lead to interferences, such as multiple crossings between links of different sets or partial overlap of link routes. To avoid this problem, the penalty map is incrementally extended as multiple linksets are routed. Given multiple linksets with predefined priorities, the highest prioritized set is routed according to the penalty map encoding the importance map, color similarities, and highlight regions of all linksets. For all lower prioritized sets, information about the previously generated sets is incorporated by adding their visual representations to the penalty map. Increasing the weight of higher prioritized linksets in the map also increases the detour subsequent linksets will take before they cross an existing link (*cf.* Figure 5).

4.3 Routing

We formulate route finding as an optimization problem in which link length is weighted against the accumulated penalties of the link derived from the penalty map. The cost function C for a set of highlight regions Ω forming a single cluster is defined as

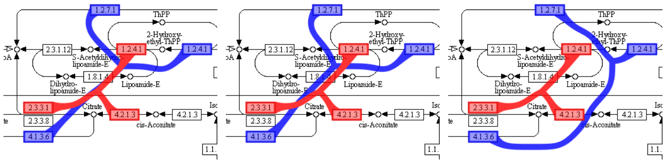
$$C = \sum_{n\in\Omega} \left(\alpha_L \int_{R_n} dR_n + \alpha_P \iint_{\mathbf{x}\in A_n} P(\mathbf{x})\, d\mathbf{x} \right). \qquad (1)$$

The first term of the weighted sum describes the length of a link's route R_n starting at node n and ending at the common cluster point $\mathbf{p}_c$. The second term describes the penalty accumulated for a link n; the penalty is formed by an integral over the area A_n made up by the link where $P(\mathbf{x})$ corresponds to the combined penalty map entry at point $\mathbf{x}$ as described in Section 4.2. The weights α_L and α_P control the trade-off between the link lengths (O1) and the amount of penalty by the base representation (O2 and O3). Increasing the value for α_L tightens the links, while increasing α_P forces the links to preserve more important context information.

In a subsequent step, the routing algorithm simplifies complex link structures by merging adjacent links to fulfill O4. This method is often referred to as 'edge bundling' [6, 8, 9]. Bundling merges segments of adjacent links, which lowers the overall link length and covered penalty. Mathematically, we describe the cost reduction $\tilde{\alpha}_i$ by lowering both α_L and α_P when n links share a common route by

$$\tilde{\alpha}_i = \left(s + \frac{1}{n - s/(s-1)} \right) \cdot \alpha_i, \qquad (2)$$

with $i \in \{L, P\}$ and where s is limited to $0 \le s < 1$ and controls how aggressively bundling should be enforced.



(a) (b) (c)

Fig. 5. (a) Multiple linksets with no knowledge of each other might overlap significantly. (b) Encoding the first set (red) with little penalty for the construction process of the second set (blue) lowers the overlap between the two sets. (c) Increasing the penalty forces the second set to accept greater detours before crossing the first set.

5 APPEARANCE ADJUSTMENTS

Let us assume that we can construct links that are optimal in terms of equations (1) and (2) with respect to the chosen weights. Although these links will be optimal according to the trade-off between the four optimization criteria, this fact does not guarantee that occlusions can be completely avoided. As examples, consider links connecting words in a text paragraph, points in the center of a scatterplot, or space-filling visualizations. To ensure sufficient visibility of the links and legibility of background information in such scenarios, we adjust the appearance of visual links to the base representation.

5.1 Link Color Selection

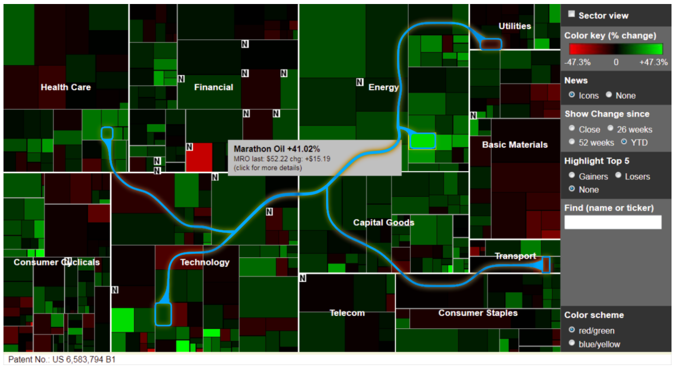
If the link color does not encode any predefined meaning, we can increase the link's visibility by adjusting its color. A visually easily distinguishable link color is automatically selected by maximizing the perceivable color difference between the link and its vicinity. We create a color histogram of the image regions along the link. The histogram is then smoothed, and its entries are considered as negative votes for the link color. The color with the lowest number of votes is selected as the link color. To permit more control over the link color selection, voting can also be applied against a predefined color palette instead of the entire color space.

5.2 Local Color Modulation

Determining a single optimal color for the entire link may not be sufficient to create links that can be visually discriminated from a cluttered background. In these cases, local color modulation of either the link's color or the base representation's color near the link can help increase the link's visibility. Local color modulation is applied during rendering by moving the link's color and the surrounding color on the base representation in opposite directions along their connection line in CIE $L^*C^*_{ab}h_{ab}$ space. Figure 6 shows an example in which the links' vicinity has been slightly modulated, leading to a halo-like glowing effect of the links. Because modifications to the base representations are often undesirable in information visualization, this step is optional.

5.3 Opacity Modulation

A well-known problem of cue-based focus-and-context techniques is their potentially negative impact on the legibility of background information. Consider, for instance, the example shown in Figure 7. In this example, even though links are mainly routed around important image regions, occlusion of text or scatterplot items cannot be completely avoided.



Base representation taken from `http://www.smartmoney.com/`

Fig. 6. Context-preserving visual links connect related elements in a space-filling treemap. The optimal link color has been automatically determined, and local color modifications make the links more visually distinct.

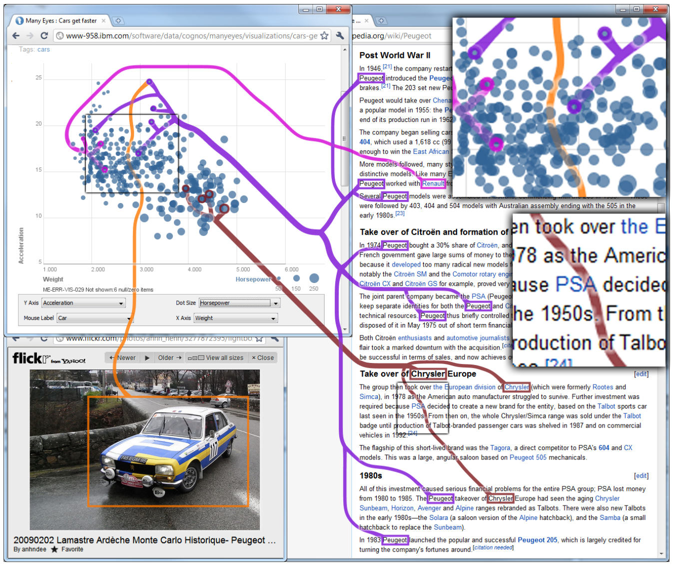


Fig. 7. Multiple linksets connecting heterogeneous information in a windowing system. Note that if occlusion cannot be avoided, the links' opacity is locally decreased to guarantee readability.

To avoid this problem, we apply additional appearance adjustments to visual links to guarantee the readability of text. To do so, we need to know which areas in the image contain text and which pixels form the characters. We rely on the simple assumption that text (or at least single words) is usually rendered in uniform color on a uniform background. We use this property to identify text directly from the input image using a histogram analysis. The input image is sub-divided into small patches, and color histograms are computed for all patches. If the histogram of an image patch is bimodal (*i.e.*, indicating two dominating colors), we assume that the patch contains text with the more prominent color representing the background color. The histogram analysis not only responds to text but to all areas that show a dominant background color with little additional information, such as the scatterplot in Figure 7.

In image patches identified as text or as other bimodal image data, we gradually reduce the link's alpha value near foreground pixels (*i.e.*, the non-dominant color in the bimodal histogram); this change makes the links appear to be rendered behind the text. An exemplary result is shown in the lower inset of Figure 7; in this example the text is still readable although a link with a similar color crosses relevant portions of the text.

6 IMPLEMENTATION

Our current system combines the advantages of both CPU and GPU to generate context-preserving visual links for interactive scenarios. Image operations are executed on the GPU using OpenGL and GLSL shaders, while sequential tasks are written in C++ and performed on the CPU. We assume that the base representation is generated with OpenGL and thus can be directly used as an input for shader-based saliency analysis [16], which yields the importance map. Using custom GLSL shaders, the penalty map is generated by extending the importance map to include color similarity, highlight regions, and information about other linksets. The penalty map is then down-sampled and transferred to the CPU, where links are routed according to the four optimization criteria. Afterwards, discretization artifacts are removed and the links are rendered using OpenGL; while at the same time, appearance adjustments are applied using custom GLSL shaders.

Using this pipeline, the overall system framerates remain interactive. The system performance on a Quad-Core 2.80 GHz CPU and NVIDIA GeForce GTX 480 is roughly given by 30 ms for the penalty map generation for an input image of 1280 × 1024 pixels, 400 ms for links routing of 20 highlight regions in one cluster uniformly distributed over the input image using a down-sampling factor of 8 for the penalty map and 200 ms for edge bundling. The other parts of the pipeline do not noticeably influence the system's performance.

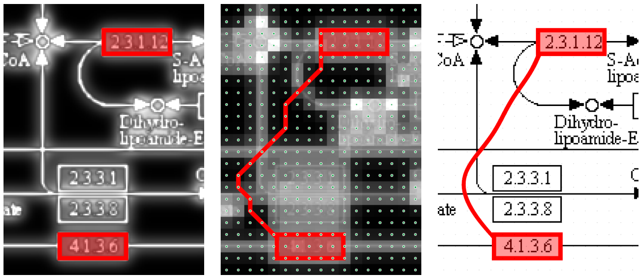
Because links routing is the most time-consuming and complex part of the algorithm, we focus on this step in the subsequent sections of this paper. As a first step, we construct a discrete version of the problem definition given in Equation (1) and discuss the applied simplifications used to make the problem feasible. We then split this setup into multiple shortest path problems and take their solutions to define the cluster point $\mathbf{p}_c$ and the optimal link routes for a single cluster. Finally, we consider connections between multiple clusters and edge bundling.

6.1 Optimization

The crucial process for the discretization of Equation (1) is the parameterization of the link routes. In our system, the link routes are defined by a variable number of anchor points, and the link-segments, which are the connections between these anchor points, are straight lines. We restrict these anchor points to a uniform grid G. Each grid cell corresponds to one pixel in the down-sampled penalty map. The distance from one anchor point to the other is further limited by the Moore neighborhood on G, *i.e.*, only connections to the eight surrounding grid points are possible (see Figures 8(a) and 8(b)). A single link's cost, which includes its length and accumulated penalty, can now be computed by the sum of all link-segment costs. We approximate the cost of a single link-segment by

$$S_i \approx \alpha_L \cdot l_i + 0.5 \cdot \alpha_P \cdot (P(\mathbf{r}_i) + P(\mathbf{r}_{i+1})) \cdot w \cdot l_i, \qquad (3)$$

where $P(\mathbf{r}_i)$ and $P(\mathbf{r}_{i+1})$ correspond to the two penalty map entries at the anchor points, w is the link's width, and l_i is the segment's length.



(a) (b) (c)

Fig. 8. (a) Closeup view of the penalty of the pathway shown in Figures 1 and 7 with two highlight regions. (b) A grid structure and a locally averaged penalty are used to simplify the problem. Every link is constructed of straight line segments only. (c) This coarse solution is smoothed in an additional step to generate visually pleasing context-preserving links.

We begin the search for optimal link routes with the only known information about the route: the links' starting points, which are defined by the locations of the highlight regions. To find the common cluster point $\mathbf{p}_c$, we compute the cost associated with reaching each grid point for every link and select the grid point with the minimal summed cost. To compute the best route for reaching any grid point, a single-source shortest path algorithm such as Dijkstra's algorithm [5] can be used. We, therefore, represent the grid as a graph in which every grid point defines a node with edges according to the Moore neighborhood. The edge weights are set using the approximation of the link segment cost (Equation (3)). The shortest path algorithm yields both, the cluster point and the optimal route for reaching this point.

To increase performance, we interleave the shortest path computations for all links that form a cluster by unifying their priority queues. In this way, we can assure that the cluster point is among the first nodes found that are reached by all links. If one such node is positioned at the front of the priority queue, there exists no other node with lower summed cost and the search can stop immediately.

6.2 Connecting Clusters

Thus far, we have only dealt with nodes that form a single cluster. In some scenarios, *e.g.*, in multi-view systems, it is desirable to form a cluster in every view and subsequently connect these sub-clusters to a main cluster. Because the sub-clusters are connected by links (*cf.* Figure 9), we treat them similarly to highlight regions and again use the aforementioned shortest path algorithm to find the links optimally connecting them. The main difference between this case and the previous scenario is that here, because the location of sub-cluster points can be influeced by other sub-clusters, the starting points for the links are also unknown (Figure 9(b-d)). To solve this problem, we first run the shortest path algorithm for all sub-clusters, which yields the cluster cost C_{sub} at each grid point. These costs are visualized for two clusters in Figure 9(a). Next, C_{sub} is used to weight the possible starting points for the computation of the main cluster, as shown in the following equation:

$$S_j(\mathbf{g}_k) = \frac{1}{\mathrm{B}} \cdot \sum_{i=1}^{N_{\text{sub}}} C_{\text{sub},i}(\mathbf{g}_k)/N_{\text{sub}}$$

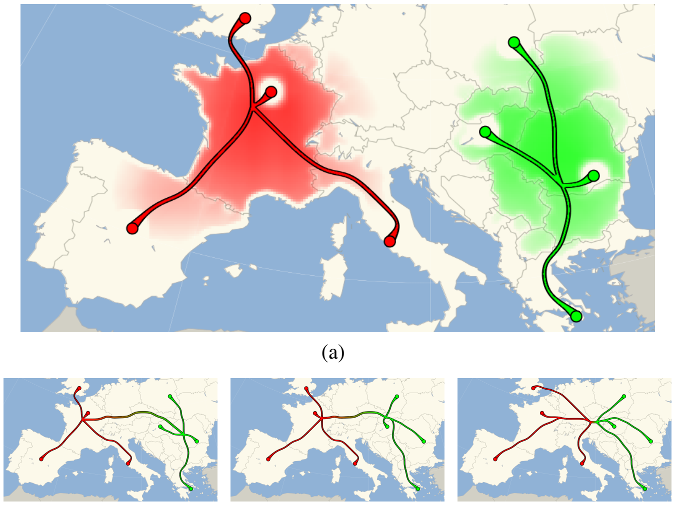
In this equation, S_j defines the initial cost at location g_k and N_{sub} corresponds to the number of links in sub-cluster j. We call B the bending factor because it influences how much the sub-cluster's links are bent towards the main cluster point (*cf.* Figure 9). If B is set to zero, the sub-cluster points remain unchanged. A high bending factor moves the sub-cluster points closer together.

To define the main cluster's link routes, we again use the information provided by the shortest path algorithm. However, in this case, the end points of the main cluster routes are not located at a highlight region but instead define the adjusted cluster points of the sub-clusters. Results from the previously computed sub-clusters' shortest path algorithm are used to construct the optimal routes connecting the adjusted sub-cluster points with their associated highlight regions.

6.3 Link Construction and Bundling

As a final step, edge bundling is applied. We could directly insert the α reduction, described by Equation (2), into the edge weights given in Equation (3). However, this process introduces interdependencies between individual links that cannot be covered by the shortest path algorithm. To keep the system interactive, we, therefore, employ bundling not during route finding but as an additional step that locally alters the routes given by the shortest path algorithm.

Starting at the cluster point, we compute the cost for all combinations of bundles in all eight directions, choosing the combination with the lowest cost as the new route. For all of the links routed along the same edge, we iteratively apply the same procedure to the next nodes until all bundles have been split. The link routes now consider all four optimization criteria. In a final step, we remove the straight line segments introduced by the discretization step by applying Gaussian smoothing as depicted in Figure 8(c).



(a)

(b) $B = 0.1$ (c) $B = 2$ (d) $B = 5$

Fig. 9. (a) Two linksets with their individual cluster costs encoded with color. Note how the cluster cost distribution is limited by the country outlines, which are treated as important information in this plain image. Clustering around the highlight regions (blank area around the nodes) is prevented by inserting them into the penalty map. (b) A very low bending factor B leaves the cluster points in place, when clusters are connected. (b-d) Increasing B pulls the sub-cluster points closer together.

It is often desirable to visualize how many links are bundled together, which we encode by a linearly increasing link width. This encoding helps the user identify in which direction he or she can expect the highest number of highlight regions and is especially advantageous for large displays and, therefore, long links. The technique is subtly applied in Figure 1 and is clearly visible in Figure 7.

7 APPLICATION SCENARIOS

As the primary application area for context-preserving visual links, we consider highlighting for complex visualizations. Our technique takes the base representation and the selected highlight regions as an input and renders the links on top of the base representation. With this general approach, a large variety of applications can be supported.

Visualization systems can apply context-preserving visual links on top of a single view to visually indicate multiple elements related to the user's current selection. For instance, in the space-filling treemap of Figure 6, a user investigates the prices of shares she has invested in.

As a more generalized approach, context-preserving visual links could be facilitated on a windowing system level utilizing an infrastructure similar to visual links across applications [27]. In this infrastructure, highlight regions are synchronized across multiple applications and collected by a central management instance. Links are then rendered on top of the desktop imagery by the window manager. Because importance maps of the entire desktop can be directly inferred from the window manager [28], context-preserving visual links can be extended across multiple application windows (*e.g.*, Figure 7) or across arbitrary GUI elements such as borders of individual views or unoccupied menu bars of a multi-view application (*e.g.*, in the *Caleydo* visualization framework [13] as shown in Figure 1).

Because our technique supports multiple concurrently visualized linksets (*cf.*, Figure 5), our concept of context preservation can also be applied to set visualizations [3] to encode different sets of items on top of visualizations with a fixed spatial encoding. Figure 7 shows an example in which links of different colors encode car brands across multiple application windows. The scatterplot visualizes the relationship between the cars' weights and their acceleration. Using context-preserving links, the properties of cars of the different brands are clearly visible. In the right window, highlights in the text help the user infer how different car brands are related historically.

Despite their general utility, context-preserving visual links are not suitable for all types of visualizations because the strong curvature of the links caused by context preservation of the routing algorithm may be interpreted as semantic meaning. For instance, encoding semantically related geographic locations on a map using our irregularly curved links may be misunderstood as an attempt to represent a street route connecting two locations. Similarly, a link carefully routed between two data clusters in a scatterplot may increase the perceived distance between the rendered data points and could suggest some kind of subdivision. Another problematic scenario is the use of context-preserving links in free-floating views used in 3D visualization systems such as *VisLink* [2] or *Caleydo's Bucket* [13]. Because our routing algorithm relies on a 2D representation of the output image, context-preserving links may interfere with the user's depth perception.

8 USER STUDY

We conducted a controlled experiment to evaluate the effectiveness of context-preserving visual links for visual search in complex visualizations. We compared context-preserving visual links to two other highlighting techniques, conventional color-coded or framed elements and straight visual links without occlusion avoidance.

We recruited 18 participants (aged 22 to 30, 14 males, 4 females) from local universities. The participants were from the fields of computer science, economics, medicine, and art history. Ten participants indicated that they had experience with visual data analysis.

8.1 Highlighting Techniques

We compared three different highlighting techniques:

Highlighting (H) draws frames around focus elements or renders these elements in distinct colors. This widely adopted technique served as a baseline condition in our experiment.

Visual links (L) are naïvely routed, straight lines that visually connect elements highlighted as in technique H. Connection lines are bundled according to the algorithm given in Section 6.3. However, the base representation is not taken into account for links routing.

Context-preserving visual links (CL) use our algorithm, which satisfies the optimization criteria, as proposed in Section 4.

The top row of images in Figure 10 shows an example of the three highlighting techniques applied to the same base representation. The appearance adjustments described in Section 5, were applied to all three techniques if the highlight regions were evaluated as subjectively difficult to perceive in a pilot study.

8.2 Task and Apparatus

Users were asked to perform a modified visual search task. In a conventional visual search task, the user looks for a specific target item within multiple distractors; a single instance of the target is usually present in about 50% of the presented images [30]. In our experiment, users were required to identify multiple targets within complex images that were visually enhanced by one of the highlighting techniques described above. This task represents a typical visual search task of knowledge workers, who need to scan related elements in multiple views or items related to their current selection in a single visualization. To assess whether users identified all the highlighted elements in the image, we asked them to count as accurately and quickly as possible the number of elements found. Users were instructed that accuracy was considered more important than task completion time.

Because we did not wish to limit our technique to a single scenario, we presented users with 16 different use case images for each technique. These images included scatterplot matrices, biochemical pathways, satellite images, and treemaps as well as multiple application windows containing text, images, and maps (Figure 10). In addition, we varied the number of highlighted elements across these images from 5 to 12, with two occurrences each.

Images were presented to the participants on a 19" monitor with 1280 x 1024 pixels. Users were seated 70 cm from the monitor. The participants' eye movements were captured using an SMI monitor-mounted eye tracker operated at 60 Hz. To limit the users' head movements and reduce physical effort, a chin rest was provided.

To assess the amount of context preservation for each of the highlighting techniques, we used various software benchmarks to evaluate the resulting images. We measured the amount of occluded information by comparing the base representation's importance map with the resulting importance map including the highlights. In addition, we calculated the increase of visual clutter in the image compared to the plain base representation using two measures of visual clutter introduced by Rosenholtz *et al.* [21], feature congestion (FC) and subband entropy (SE). The accumulated measures across all 16 images show that H causes the least occlusion increase in the base representation and that L causes the most ($O_h = 0.8\%, O_l = 1.97\%, and O_{cl} = 1.28\%$). The clutter measures of Rosenholtz *et al.* [21] indicate that H slightly decreases visual clutter compared to the plain base representation ($FC_h = -0.77\%$ and $SE_h = -0.36\%$) and that both linking techniques lead to a clutter increase; CL causes more measurable clutter ($FC_{cl} = 3.57\%$ and $SE_{cl} = 2.5\%$) than L ($FC_l = 2.52\%$ and $SE_l = 1.23\%$).

8.3 Design and Procedure

The study was conducted as a within-subjects experiment with three experimental conditions (highlighting technique) and 16 repetitions (images) for each condition. For each trial, we measured task completion time and correctness. Users loaded a new image by pressing a key and were asked to press the same key again as soon as they counted all occurrences; this press caused a blank background to be displayed. Then, the user reported the number of highlighted elements to the experimenter, who noted it. Task completion time corresponded to the duration the image was displayed. As a measure of correctness we determined the deviation of the reported numbers from the actual number of highlights. Results from the 16 repetitions were accumulated.

Prior to the experiment, users were tested for color-blindness and given a warm-up period with 18 images. After each condition, the users were required to assess their subjective satisfaction with the highlighting technique on a questionnaire containing seven questions. Upon completion of the experiment, they were asked to assess their overall preference and to participate in a semi-structured interview.

The sequence of the conditions was counter-balanced and the sequence of the 16 images per condition was randomized. We used the same images for each technique. To avoid learning effects due to the within-subjects design, we varied the location of highlight regions in the three conditions while keeping the distribution of distances between the highlight regions approximately constant.

8.4 Hypotheses

The major goal of the experiment was the assessment of the effectiveness of visual links and of context preservation of links on visual search task performance. It should be remembered that context information was not required for the visual search task and that the aim of the experiment was *not* to assess the amount of context preservation provided by the different highlighting techniques or the effect of such preservation on context information perception. To evaluate this aspect, a more complex visual analysis task would have been requiring. We therefore evaluated context preservation solely based on software benchmarks (*cf.* Section 8.2).

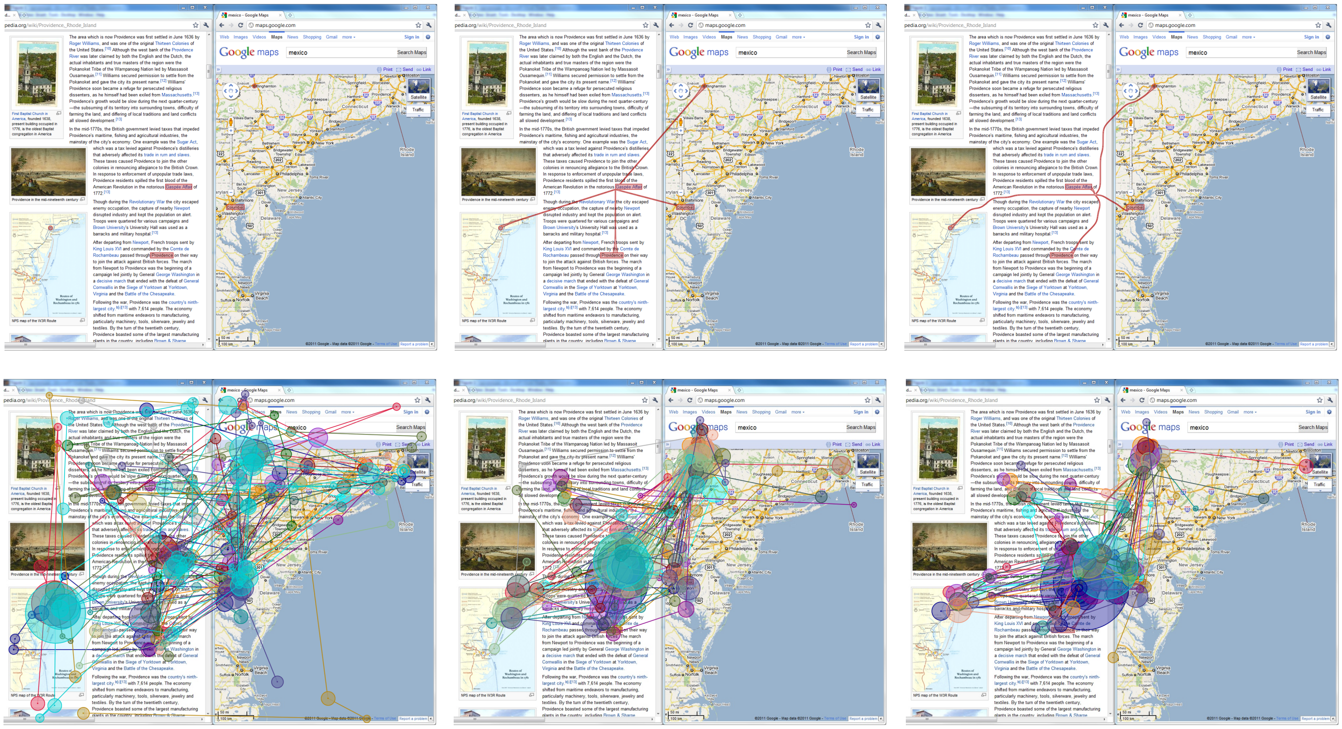
We formulated the following three hypotheses for this experiment:

[H1] *Visual links lead to a better performance than conventional highlights.*

Because connectedness is a strong visual variable, we expect visual links to outperform conventional highlighting. Although this hypothesis does not directly address the core aspect of this paper, this question has, to the best of our knowledge, never been evaluated using multiple highlighted elements in complex visualizations. Corroboration of this hypothesis is necessary to fully motivate our approach of using visual links to effectively highlight related items for visual analysis tasks.

[H2] *Context-preserving visual links do not have a negative impact on performance.*

Context preservation of visual links leads to longer links with more complex shapes. Following such a complex link probably requires more mental effort than following a straight line. However, we think



(a) (b) (c)

Fig. 10. The three highlighting techniques evaluated in the experiment: (a) simple highlighting **(H)**, (b) straight visual links **(L)**, and (c) context-preserving visual links **(CL)**. The top row shows one set of images presented during the experiment; the bottom row illustrates the corresponding recorded gaze plots for the 18 participants, each of which is represented by a different color. The dots indicate fixations, size encodes the fixation duration (dwell time), and the connecting lines show the gaze paths (saccades). While simple highlighting creates irregular search patterns, both linking techniques strongly guide the user's gaze.

that the easier recognizability resulting from better discrimination from the background information due to occlusion avoidance outweighs this additional effort, resulting in a performance comparable to that obtained using naïvely routed, straight links.

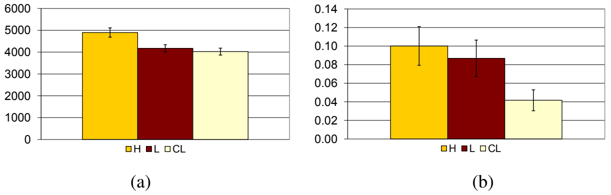
[H3] *Context-preserving visual links have a positive impact on user satisfaction*.
Although the context information in the base visualization is not required for our visual search task, we assume that the user acceptance of context-preserving visual links is higher because the appearance of such links is adapted to the base representation. We hypothesize that the decreased amount of occlusion that is due to the optimized routing subjectively outweighs the higher measurable visual clutter.

8.5 Results

We measured the time users needed to count the highlighted elements in an image (defined by the time period an image was displayed), the correctness of the reported highlight element numbers, and subjective measures, which were addressed by evaluation questionnaires. Performance (*i.e.*, task completion time and correctness) measures were evaluated using repeated measures ANOVA ($\alpha = .05$) with Bonferroni-adjusted post-hoc comparisons. Questionnaire answers, which were given on a seven-point Likert scale, were analyzed using Friedman non-parametric tests ($\alpha = .05$) for main effects and post-hoc comparisons using Wilcoxon Signed Rank tests with Bonferroni adjustments. Performance results are illustrated in Figure 11 and questionnaire results are given in Figure 12.

8.5.1 Performance

We found a significant main effect for task completion time ($F_{2,34} = 13.994, p < .001$). Post-hoc comparisons revealed that completion time was significantly higher with H ($t_h = 4897ms$) than with either linking technique ($t_l = 4176ms$ and $t_{cl} = 4024ms$), as illustrated in Figure 11(a). The difference in completion time between L and CL was not significant.



(a) (b)

Fig. 11. Performance results: average and standard error of (a) task completion time (ms) and (b) error rate (average deviation of the reported number from the actual number of highlights per image).

Because participants were instructed that correctness was more important than speed, the error rate was generally low. On average, users had a misestimation of 0.07 elements per image across all techniques; the percentages of erroneous trials were 13.2%, 12.5%, and 6.9% for H, L, and CL, respectively. The main effect for correctness was borderline significant ($F_{2,34} = 3.572, p = .039$); we found no significant differences in Bonferroni-adjusted post-hoc comparisons. Figure 11(b) gives the measured error rates.

8.5.2 Subjective Assessment

The questionnaire item assessing subjective speed (*I could find highlighted elements very quickly*, $\chi^2(2) = 20.548, p < .001$) was rated significantly higher for the linking techniques than for H. The questionnaire item assessing subjective correctness (*I am sure I did not miss any highlighted element*, $\chi^2(2) = 18.681, p < .001$) was rated highest for CL, and the difference was significant with respect to H. Users assessed the mental demand (*The task was very mentally demanding*, $\chi^2(2) = 9.692, p = .008$) as higher for H than for L and

CL. We did not find any differences in the assessments of the background visibility (*The background (non-highlighted) information was clearly visible*, $\chi^2(2) = 2.607, p = .272$) or the amount of visual clutter (*The highlighting technique introduced a high amount of visual clutter*, $\chi^2(2) = 6.632, p = .036$). However, the ratings of visual attractiveness (*The highlighting technique was visually pleasing*, $\chi^2(2) = 23.831, p < .001$) show that users assessed CL as more visually pleasing than either L or H. The usefulness (*The highlighting technique would be beneficial for my every day computer work.*, $\chi^2(2) = 23.825, p < .001$) was assessed as significantly lower for H than for either of the linking techniques. Similarly, the overall preference scores assigned after the experiment show a higher acceptance of the two linking techniques than of H. On average, all questionnaire items received a higher assessment for CL than for L as illustrated in Figure 12.

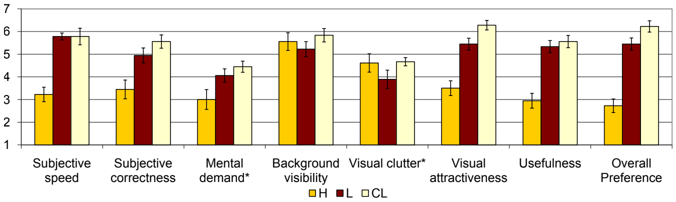


Fig. 12. Questionnaire results (average and standard error) on a 7-point Likert scale, in which higher is better. Results marked with * have been inverted.

8.5.3 User Feedback and Eye-Tracking Data

In the interview, four users mentioned they disliked H because of an inability to clearly see highlighted elements on cluttered background. Others reported that highlighted elements were easy to miss; thus, simple highlighting was described as *'more mentally demanding'* than the linking techniques. Five users reported that they appreciated visual links in general because clustered elements were easier to count, and no sequential scan of the image was necessary.

An exploratory analysis of the recorded eye-tracking data supports this informal user feedback. The fixation patterns in the gaze plots shown in Figure 10 illustrate how users sequentially scanned the entire display. As illustrated in the exemplary gaze plot, gazes were unstructured in H (Figure 10(b)). In contrast, the visual search area for the two linking techniques was much smaller. In particular, fixations on the cluster points are clearly visible.

The feedback on context-preserving visual links obtained in the interviews was generally positive. Seven users mentioned that context-preserving visual links look *'cooler'*, *'better'*, or *'more organic'*. Two users described them as more *'cluttered'*. Five users explicitly mentioned that they appreciated that context-preserving visual links do not cover background (text in particular). One user said *'they are well visible and the background is also still visible, so they are easy to follow'*.

8.5.4 Discussion of Results

Based on the results of our experiment, two of our hypotheses are fully supported, and one is partially supported, as described below.

[H1] *Visual links lead to a better performance than conventional highlights*.
The higher task completion time and the (mostly) lower subjective assessment of simple highlighting compared to both linking techniques supports this hypothesis. This trend is also reflected in the informal user feedback in which highlighting was described as more mentally demanding as well as in the eye tracking data where more unstructured scan paths are evident with simple highlighting than with the visual linking techniques. Previous work [7] has shown that combining frames and trails is more effective in guiding a user's attention to a *single* target at a far distance than the use of frames alone. Complementing this work, we were able to show that adding trails is also beneficial for locating *multiple* targets on a conventional monitor where no items are placed outside the user's field of view.

[H2] *Context-preserving visual links do not have a negative impact on performance*.
Because we found no difference between the two linking techniques in task completion time or correctness this hypothesis is supported. Our results show that even though context information is not required for a simple search task, adding context preservation to visual links neither increases the perceived complexity of a visual search task nor measurably influences task performance.

[H3] *Context-preserving visual links have a positive impact on user satisfaction*.
Based on our questionnaire results, this hypothesis is partially supported. We did not find any differences in the subjective assessment of mental demand, in the perceived background visibility, or in the overall user preference. Interestingly, we also did not find any difference in the subjective evaluation of visual clutter, even though software measures by Rosenholtz *et al.* [21] indicate that context-preserving links cause the most pronounced clutter increase. However, there was a difference in the assessment of visual attractiveness; context-preserving links were rated significantly higher than straight links. Because it can strengthen user engagement, visual attractiveness is an important factor [17].

Several aspects of information perception in a visual display environment could not be captured in the course of our experiment. These aspects include the impact of context preservation on the perceived context information, the suitability of context-preserving visual links for different visualization techniques, and the suitability of such links for concurrently visualizing multiple linksets. In the future, we plan to investigate more specific task areas. We also plan to conduct long-term observations of knowledge workers using more realistic visual analysis tasks to evaluate more thoroughly the effects of context preservation.

9 CONCLUSIONS AND FUTURE WORK

In this paper, we present a technique for augmenting visual links on top of a base representation with the main objective of preserving the underlying context information. Using an image-based description of the base representation, the routing and visual appearance of links is optimized to avoid occlusion of important information and to permit the links to visually stand out from the overlaid representation. By performing the highlighting procedure as an additional step at the end of the visualization pipeline, our technique can be applied to a wide range of visual analysis scenarios.

The results of a user study support our hypothesis that context preservation does not have a negative impact on either the subjective or quantitatively measured task performance in a simple visual search task involving complex visualizations. In contrast, subjective feedback suggests that users perceive context-preserving visual links as more attractive than straight links or simple highlights. Irrespective of their form factor, our experiment also showed a clear benefit of visual links compared to simple highlighting without any line connections for visual search tasks.

In the future, we aim to analyze further visual variables in addition to color, including curvature, line width, or texture, to optimize visual discrimination. We also plan to utilize modern graphics processing units to increase performance and provide context-preserving visual linking as a lightweight tool at the window manager level, facilitating an infrastructure similar to the one proposed in [27]. In this way, existing visualization frameworks and other software applications could easily incorporate the linking functionality as a valuable extension in a minimally invasive manner.

ACKNOWLEDGMENTS

This work was funded in part by the Austrian Research Promotion Agency (FFG) through the BRIDGE projects *Deskotheque* (822716) and *InGeneious* (385567) as well as the Caleydoplex project (P22902) granted by the Austrian Science Fund (FWF).

REFERENCES

[1] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys (CSUR)*, 41(1):1–31, 2008.

[2] C. Collins and S. Carpendale. VisLink: revealing relationships amongst visualizations. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '07)*, 13(6):1192–1199, 2007.

[3] C. Collins, G. Penn, and S. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '09)*, 15(6):1009–1016, 2009.

[4] J. Fekete, D. Wang, N. Dang, A. Aris, and C. Plaisant. Interactive poster: Overlaying graph links on treemaps. In *Proceedings of the IEEE Symposium on Information Visualization Conference Compendium (InfoVis '03)*, pages 82–83. IEEE Computer Society Press, 2003.

[5] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596–615, 1987.

[6] E. R. Gansner, Y. Hu, S. North, and S. Carlos. Multilevel agglomerative edge bundling for visualizing large graphs. In *Proceedings of the IEEE Symposium on Pacific Visualization (PacificVis '11)*. IEEE Computer Society Press, 2011.

[7] R. Hoffmann, P. Baudisch, and D. S. Weld. Evaluating visual cues for window switching on large screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*, pages 929–938. ACM Press, 2008.

[8] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '06)*, 12(5):741–748, 2006.

[9] D. Holten and J. van Wijk. Force-Directed edge bundling for graph visualization. *Computer Graphics Forum*, 28(3):983–990, 2009.

[10] L. Itti, C. Koch, and E. Niebur. A model of Saliency-Based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1254–1259, 1998.

[11] A. Khan, J. Matejka, G. Fitzmaurice, and G. Kurtenbach. Spotlight: directing users' attention on large displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*, page 791798. ACM Press, 2005.

[12] R. Kosara, S. Miksch, and H. Hauser. Semantic depth of field. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '01)*, pages 97–104. IEEE Computer Society Press, 2001.

[13] A. Lex, M. Streit, E. Kruijff, and D. Schmalstieg. Caleydo: Design and evaluation of a visual analysis framework for gene expression data in its biological context. In *Proceeding of the IEEE Symposium on Pacific Visualization (PacificVis '10)*, pages 57–64. IEEE Computer Society Press, 2010.

[14] A. Lex, M. Streit, C. Partl, K. Kashofer, and D. Schmalstieg. Comparative analysis of multidimensional, quantitative data. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '10)*, 16(6):1027–1035, 2010.

[15] A. R. Martin and M. O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Proceedings of the IEEE Conference on Visualization (Vis '95)*, page 271. IEEE Computer Society Press, 1995.

[16] E. Mendez, S. Feiner, and D. Schmalstieg. Focus and context in mixed reality by modulating first order salient features. In *Smart Graphics*, volume 6133 of *Lecture Notes in Computer Science*, pages 232–243. Springer, 2010.

[17] H. L. O'Brien and E. G. Toms. The development and evaluation of a survey to measure user engagement. *Journal of the American Society for Information Science and Technology*, 61(1):50–69, 2010.

[18] S. Palmer and I. Rock. Rethinking perceptual organization: the role of uniform connectedness. *Psychonomic Bulletin and Review*, 1(1):2955, 1994.

[19] C. I. E. Publication. Industrial Colour-Difference evaluation. 116-1995, CIE, 1995.

[20] J. Risch, R. May, S. Dowson, and J. Thomas. A virtual environment for multimedia intelligence data analysis. *IEEE Computer Graphics and Applications*, 16(6):33–41, 1996.

[21] R. Rosenholtz, Y. Li, and L. Nakano. Measuring visual clutter. *Journal of Vision*, 7(2), 2007.

[22] J. Seo and B. Shneiderman. Interactively exploring hierarchical clustering results. *Computer*, 35(7):80–86, 2002.

[23] B. Shneiderman and A. Aris. Network visualization by semantic substrates. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '06)*, 12(5):733–740, 2006.

[24] M. Streit, M. Kalkusch, and D. Schmalstieg. Interactive visualization of metabolic pathways. In *Poster Compendium of the IEEE Conference on Visualization (Vis '07)*. IEEE Computer Society Press, 2007.

[25] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980.

[26] E. Veas, E. Mendez, S. Feiner, and D. Schmalstieg. Directing attention and influencing memory with visual saliency modulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM Press, 2011.

[27] M. Waldner, W. Puff, A. Lex, M. Streit, and D. Schmalstieg. Visual links across applications. In *Proceedings of the Conference on Graphics Interface (GI '10)*, pages 129–136. Canadian Human-Computer Communications Society, 2010.

[28] M. Waldner, M. Steinberger, R. Grasset, and D. Schmalstieg. Importance-Driven compositing window managment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM Press, 2011.

[29] C. Ware. *Information visualization : perception for design*. Morgan Kaufman, San Francisco CA, second edition, 2004.

[30] J. M. Wolfe. Visual search. In H. E. Pashler, editor, *Attention*, pages 13–74. Psychology Press, 1998.

[31] J. M. Wolfe. What can million trials tell us about visual search? *Psychological Science*, 9(1):33–39, 1998.

[32] S. Zhai, J. Wright, T. Selker, and S. Kelin. Graphical means of directing user's attention in the visual interface. In *Proceedings of the Conference on Human-Computer Interaction (INTERACT '97)*, pages 59–66. Chapman & Hall, 1997.

[33] C. Ziemkiewicz and R. Kosara. Laws of attraction: From perceptual forces to conceptual similarity. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '10)*, 16(6):1009–1016, 2010.