ViSizer: A Visualization Resizing Framework

Yingcai Wu, Xiaotong Liu, Shixia Liu, *Member, IEEE*, and Kwan-Liu Ma, *Fellow, IEEE*

Abstract—Visualization resizing is useful for many applications where users may use different display devices. General resizing techniques (e.g., uniform scaling) and image resizing techniques suffer from several drawbacks, as they do not consider the content of the visualizations. This work introduces ViSizer, a perception-based framework for automatically resizing a visualization to fit any display. We formulate an energy function based on a perception model (feature congestion), which aims to determine the optimal deformation for every local region. We subsequently transform the problem into an optimization problem by the energy function. An efficient algorithm is introduced to iteratively solve the problem, allowing for automatic visualization resizing.

Index Terms—Resizing, Visualization Framework, Perception, Focus+Context, Nonlinear Least Squares Optimization

1 Introduction

Research into visualization resizing is becoming particularly important with the advance of collaborative visual analysis, in which users might use different display devices. Although modern visualization methods can regenerate a new visualization if the display changes, the common regeneration methods do not always work. Some methods such as tag clouds and force-directed graph algorithms may regenerate a totally different layout to fit the new display, which is unacceptable in a collaborative application. As a result, a generic resizing framework is needed to efficiently produce consistent visualizations, such that embedded useful patterns in the resized visualizations can still be revealed as effectively as the original. Additionally, such a framework can relieve the burden of designing a visualization, as the developers no longer need to consider the re-scaling problem.

There are several possible solutions to resizing a visualization. One simple approach is uniform scaling. Unfortunately, this would not work if the visualization is resized to a different aspect ratio. To tackle this problem, the visualization can simply be cropped to ensure that the uniform scaling can coincide with the new aspect ratio. However, this method may discard important or useful context information. Some other straightforward solutions also often fail to produce desired results. For instance, an alternative approach for graph layout resizing is to scale node coordinates homogeneously, maintain their visual size, and use a fast overlap removal mechanism [11]. Unfortunately, this approach does not work when the target display is too small to hold all graph nodes without shrinking some of them. Moreover, the links between the graph nodes are likely to be occluded by the dense graph nodes that are relatively large in the target display.

This work presents a perception-based framework, *ViSizer*, for effectively resizing a visualization for any display using an image warping approach [35]. The majority of image resizing methods such as seam carving [1] keep important regions unchanged, leading to failure when the region sizes are larger than the target image sizes. In contrast, the optimized scale-and-stretch method [35] can address this problem by scaling important regions uniformly and deforming homogeneous context. ViSizer employs a similar deformation scheme, but it is much more flexible. It can be viewed as a multi-focus+context visualization technique by allowing users to explicitly specify the expected scaling factors for the regions of interest in the target visualization.

Importantly, ViSizer employs a new perception-based significance measure designed for visualization. The measure can estimate the visual clutter magnitude and guide the resizing process to avoid compressing visually cluttered items. A new energy function is defined based on the measure to transform the resizing problem into a nonlinear least squares optimization problem. The optimization problem can then be solved by an efficient iterative algorithm.

The major contributions of this work are as follows:

- Study a new problem of how to effectively resize a visualization for any display.
- Transform the visualization resizing problem into an optimization problem with a novel perception-based energy function.
- Design and develop a generic framework for automatic visualization resizing.

2 Related Work

Image resizing methods can be generally classified as discrete or continuous methods [31]. Discrete meth-

- *Yingcai Wu and Kwan-Liu Ma are with the Department of Computer Science, University of California, Davis, 2121 Kemper Hall, One Shields Avenue, Davis, CA 95616. E-mail: {ycwu,ma}@cs.ucdavis.edu*
- *Xiaotong Liu and Shixia Liu are with the Microsoft Research Asia, Beijing, China. E-mail: fallingstar621@gmail.com, Shixia.Liu@microsoft.com.*

ods, i.e., seam carving [1], resize an image by judiciously inserting or removing left-to-right or top-to-bottom seams. Continuous methods [35], [36] associate an image with a grid and resize the image by deforming the grid non-homogeneously. These techniques are not optimal for visualization resizing. First, visualizations have special layouts with interactive visual elements rather than static pixels. Acceptable deformation in images may be viewed as a serious distortion to the layouts. For example, non-homogeneous deformation of words in a word cloud may decrease their readability. Second, visualization resizing is more constrained by visual clutter - the state in which excess and disorganized items degrade visual task performance. This performance degradation is due to the difficulty in recognizing or searching for an item interfering with other surrounding items, especially when the item spacing is small [33].

Automatic resizing methods based on constraints [17] are commonly used for resizing objects in Graphical User Interfaces (GUIs) due to their expressiveness. However, they could become difficult to use when there are many GUI objects or when the constraints are too complex to specify. Manually-authored methods such as Artistic Resizing [10] are also widely employed to resize GUI objects based on provided examples, thus allowing a designer to customize the resizing behaviors. They are primarily employed for resizing a limited number of relatively simple graphics objects in GUIs. In contrast, ViSizer mainly aims at resizing a visualization with a large number of visual items that are usually distributed irregularly.

Our method is an automatic resizing technique based on nonlinear least squares optimization. Compared with the automatic GUI resizing techniques, our method automatically formulates the constraints by the perception-based clutter measure and thus it does not require manually-specified constraints. Our method can also be regarded as a manually-authored technique because users are allowed to customize the resizing by manually specifying regions of interest and assigning expected scaling factors for the regions. Therefore, it takes advantage of both automatic and manually-authored resizing techniques.

Visual Clutter is an important factor for designing an effective visualization and user interface. Baldassi et al. [3] showed that visual clutter misleads users to problematic judgments and to more confidence in erroneous decisions. Researchers traditionally measured visual clutter based on information density or the number of elements [32]. Some researchers argued that this traditional method was not a good measure of clutter, because the number of elements can be ill-defined [27]. Our framework uses the feature congestion method based on local feature variance [27] because it is effective for predicting clutter and is much more efficient than other quantitative methods.

Data Abstraction can be used to adapt visualizations to devices of small displays (e.g., mobile devices). Various data abstraction techniques [12], such as clustering [14], point/line displacement [9], and dimensional reordering [26], have been proposed to reduce information density for alleviating the visual clutter problem. Elmqvist and Fekete [13] presented a general hierarchical aggregation model for information visualization. These techniques can simplify visualizations created on large-screen devices. As a result, the visualizations may be adapted to devices of small screens. However, they inevitably discard information and are likely to fail when resizing to different aspect ratios. Moreover, what information is to be discarded solely relies on either the user's ability to navigate a view with less clutter or the heuristic rules embedded in a visualization [27].

Focus+Context Visualization such as Fisheye [15] is a popular solution for visualizing data on mobile devices [19]. Sarkar and Brown [29] explored a Fisheye lens method for viewing and browsing graphs. A metaphor called rubber sheet stretching was also introduced to visualize graphs within small display areas [30]. Carpendale et al. [5], [6] presented a new magnification technique using a three-dimensional pliable surface. Keahey and Robertson [20] introduced efficient techniques to combine multiple transformations. Image warping techniques were used to deform a street-level map to fit the associated schematic map [4]. Jenny and Hurni [18] employed a deformation method to visually analyze the planimetric and geodetic accuracy of the old map. Munzner et al. [25] used Fisheye to ensure landmark visibility and constant frame rates for scalable tree comparison.

These Focus+Context techniques are useful in visualization, but they may lead to target acquisition problems and impaired spatial comprehension [7]. Zanella et al. [38] suggested using grids and shading to tackle these problems. Our method can be regarded as a focus+context technique, but we novelly apply the technique in resizing visualizations. It allows users to specify the expected scaling factors of regions of interest. Furthermore, the important regions are uniformly scaled, and the distortions are distributed across the whole visualization rather than only the local regions as handled in the existing techniques. Guided by a perception-based significance map, ViSizer can also minimize the chance of task performance degradation caused by visual clutter.

3 Design Methodology

This section presents our methodology for designing a resizing framework. We start from investigating the challenges raised by several use scenarios, and then discussing the design constraints and visual variables for the framework. Our approach, the flexible distortion control mechanism, as well as the system overview are subsequently presented.

3.1 Design Challenges

There are several typical scenarios where our visualization resizing framework is useful.

- Several users collaborate on one visual analytics project using computing devices with different display sizes and aspect ratios.
- A user facing a large (e.g., wall-sized) display uses her hand-held device for visual item selection.
- A user may have different computing devices with different displays to work at different places.

These scenarios present a few challenges for designing an effective resizing approach. First, the resized visualization must be consistent with the original one. Visualization inconsistency may convey incorrect information, mislead the discussion in the collaboration, or even draw a wrong conclusion. This poses a challenge to some visualization techniques such as tag clouds and graph layout methods that usually create different layouts for different displays. Second, the technique must be efficient. In a collaboration scenario, a new user may join in the collaboration at any time and the visualization under discussion should be resized to fit his display in real time, such that he can start the collaboration immediately. Therefore, the algorithms that require excessive time to regenerate layouts is inappropriate for the application. Third, the method should naturally support multi-focus+context visualization as well as necessary visual cues for users to comprehend the geometric distortion. Usually, a user using a hand-held device does not have an appropriate display to show the original visualization and a deformed version is needed. Finally, it should avoid introducing additional clutter when a visualization is resized to a smaller display.

3.2 Design Constraints

Different visualizations usually have different constraints and requirements on their use of space. It is difficult or even impossible to design a generic resizing framework that can suit all different visualizations. This work mainly focuses on non-space-filling visualizations such as word clouds, graphs, and scatterplots. Space-filling visualizations such as treemaps have more strict spatial and geometric constraints than non-space-filling visualizations. For instance, radial space-filling visualizations have strict circular layouts, thus limiting the flexibility of using geometric deformation to fit a certain aspect ratio. This prevents us from utilizing empty or unimportant regions for preserving significant regions. Furthermore, the spatial and geometric constraints are quite different, which presents a big obstacle to creating a general framework. Therefore, our framework primarily aims at non-space-filling visualizations.

3.3 Visual Variables

Visual items such as points and lines are the basic elements for creating a visualization. Each visual item owns a set of visual variables such as color and position to encode multidimensional information of a data item. Visual variable encodings are considered as a basis for visualization. The effectiveness of the resizing framework highly depends on which visual variable encodings are modified in the resizing process and whether or not these encodings are preserved after resizing. This requires that we should carefully determine which spatial visual variables are to be changed or to be preserved in the resizing process.

Spatial visual variables, such as position, area, length, angle, slope, density, and shape [23], have spatial properties, which can be changed more or less by geometric deformation. In contrast, non-spatial visual attributes such as color and texture are invariant to deformation. As the framework utilizes geometric deformation to resize a visualization, spatial visual variables of the visual items are modified. This can result in undesired distortion to the original visualization and may mislead a user to draw a wrong conclusion in quantitative visualizations.

Our framework focuses on visualization tasks where users merely need to discern data patterns such as distributions in scatterplots rather than interpret data values quantitatively and accurately. Additionally, it provides a method to facilitate user collaboration and interactions in visualizations shown in different displays. We argue that in these tasks changing spatial visual variables to a certain extent is allowed when data patterns are preserved in significant regions. To simplify the discussion, in this work we mainly change visual variables: position, area, or both to resize a visualization. All other visual variables remain the same during the resizing process.

A scatterplot, for example, uses x- and y- positions (or coordinates) to encode two-dimension information. In many qualitative visualization tasks, the information is transformed from higher dimensional space by, for example, multidimensional scaling techniques (see Fig. 8). The scatterplot is used to show only the overall pattern and trend of the pattern. It is unnecessary to accurately and quantitatively interpret the positional visual variable. Therefore, changing the positional visual variable is allowed to maintain the overall pattern. This observation can also apply to other visual variables such as area in some scenarios, where there is no need for accurate and quantitative interpretation of the visual variables. For example, when a user facing a large (e.g., wall-sized) display uses her hand-held device to select graph nodes, it would be reasonable to enlarge the area where the important nodes locate to facilitate selection.

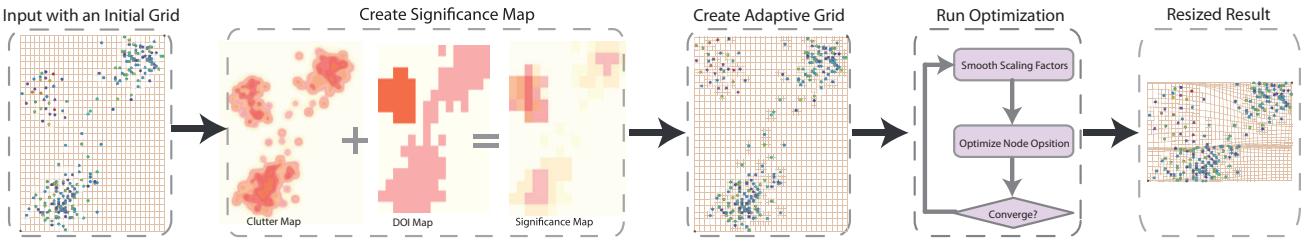


Fig. 1. System overview: ViSizer first creates a significance map based on the degree of interest (DOI) map and the visual clutter map, and produces a significance-aware grid; it then searches for an optimal transformation to the grid and adjusts the visualization with the deformed grid.

3.4 Method and Distortion Control

We design an efficient and flexible resizing framework with a seamless integration of a multi-focus+context mechanism for non-filling-space visualizations. By changing the spatial variables, the framework enables different levels of distortion in the resized results to meet different user requirements.

- Distortion-free: all visual items except the empty space in the non-space-filling visualization are uniformly scaled. In other words, we modify the positions of visual items, such that the empty space is greatly compressed while the relative positions of the visual items are preserved. This is particularly useful for the tasks of visualizing overall data patterns rather than quantitative analysis.
- Controlled distortion (multi-focus+context visualization): the visual items will be deformed based on the expected scaling factors specified by users. In particular, we change both the positions as well as the areas of the visual items to fit a new display while allowing for the multi-focus+context effect.

Furthermore, the framework uses background grids to facilitate users' comprehension of the geometric distortion to improve the accuracy, as suggested by other researchers [22], [38].

The primary benefit of this framework is that it is flexible and can meet different resizing requirements. Users can determine whether distortion is allowed or not. By measuring the visual clutter in the original visualization, the framework can avoid compressing the cluttered regions in the resized result. Additionally, it can also relieve the burden of visualization designers for handling the re-scaling problem.

3.5 System Overview

Fig. 1 shows an overview of ViSizer. ViSizer employs a grid-guided resizing optimization scheme. It partitions a visualization with a grid, then iteratively adjusts the grid in an judicious manner under some constraints to achieve an optimal deformation of the grid. Finally, the visualization can be resized according to the deformed grid by forward mapping. We choose to deform the grid rather than the visualization in the optimization because of the efficiency and flexibility of the grid-guided method. The efficiency is achieved through the iterative optimization scheme widely used in image warping and resizing, while the flexibility is achieved by the energy function associated with the grid-guided optimization method. Moreover, the grid can provide sufficient visual cues for a viewer to comprehend the deformation.

ViSizer includes two parts: pre-processing and optimization. In the pre-processing part, a significance map, a combination of a DOI map and a visual clutter map, is created to encode the significance value of every quad in the grid. Next, a significance-aware or adaptive grid is created based on the significance map to reduce linearization artifacts and to approximate the nonlinear deformation better. In the optimization part, the resizing problem is transformed into a nonlinear least squares optimization problem through an energy function based on the significance map, quad deformation, and edge bending. ViSizer solves the optimization problem iteratively to find a good solution. The scaling factor for every grid quad will be adjusted at each iteration to minimize the potential distortion. The iteration repeats until a certain convergence condition is reached, i.e., all vertex movements are very small in the current iteration. Finally, the optimization generates a deformed grid and it is utilized to adjust the visualization accordingly.

4 Pre-processing

In pre-processing, the framework first associates an input visualization with a uniform grid used for warping the visualization. The input visualization consists of a bitmap image of the visualization and all visual items of the visualization. It then creates a significance map for encoding the significance of different regions in the visualization. Finally, the grid is adjusted to be significance-aware, which means that more important regions are covered by more quads.

4.1 Significance Measure

The significance measure is an image-based measure and is a core part of the resizing framework. It is used to create a significance map for guiding the significance-aware grid adjustment and to determine

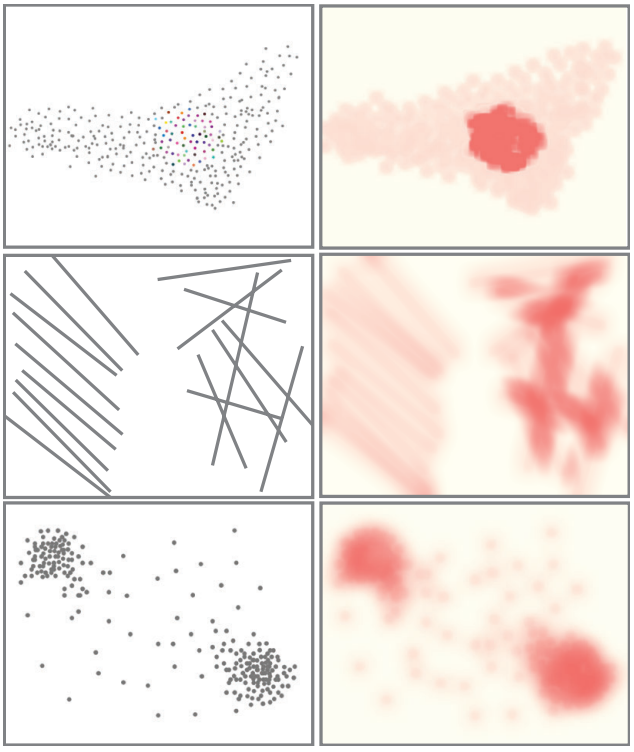


Fig. 2. Left to right: illustrative examples (left) and their clutter maps (right) for showing the detected color clutter, orientation clutter, and density clutter (from top to bottom).

the vertex movements in the optimization process. The significance of each local region can be estimated by the measure based on the DOI and the magnitude of clutter of the visual items in the region. Only quads that are both locally important and cluttered should be protected against distortion.

4.1.1 Degree of Interest

Degree of interest (DOI) was first introduced by Furnas [15] to indicate that visual items in visualization have different levels of importance. Clearly, DOI is application-specific and different applications may have different definitions. With appropriately defined DOI, important regions can be differentiated from less important regions, which allows us to distribute distortion to less important regions. For example, a DOI map for a scatterplot is shown in Fig. 1; it assigns a higher level of importance to the top left cluttered region. For simplicity and clarification, in scatterplots (except Fig. 1), we view all visual items equally important. In word clouds and graphs, the importance of a visual item is assigned based on the size of the item (word or graph node).

4.1.2 Clutter Estimation

The DOI map is used to preserve regions of interest in visualization resizing. However, relying only on the DOI is insufficient for determining the shrinking or stretching operations of a visualization. This is because some regions may become crowded with excess, unorganized visual items when a user repeatedly resizes the visualization. As a consequence, the visualization would be cluttered and the performance of visual tasks, such as visual searching, could be degraded [33]. Fig. 3(f) shows an example in which visual clutter becomes severe when the words in the red ellipses get closer and closer. To tackle this problem, a quantitative measure of visual clutter estimation is introduced. In this scenario, the regions with high magnitudes of clutter should not be shrunk to avoid being even more cluttered.

Our framework employs an efficient method called *Feature Congestion* [27] to estimate the clutter magnitude in every local region. This method can produce an image called *clutter map* with the same resolution of the visualization for revealing the clutter magnitude at every pixel. It uses the level of feature congestion to indicate the degree of clutter in an image. The congestion level can be measured by a statistical saliency model based on the observation that unusual items are usually salient [27]. Whether or not an item is unusual depends on how different the feature vector of the item is from the local distribution of other feature vectors. A feature vector is composed of the color, the luminance-contrast, and the orientation of the item. Thus, the statistical saliency for a feature vector X can be evaluated by the Mahalanobis distance [24] as

$$\Delta = \sqrt{(X-\mu)^T S^{-1}(X-\mu)} \qquad (1)$$

where μ and S denote the mean matrix and covariance matrix of the local feature distribution, respectively.

This model uses a set of covariance ellipsoids, determined by the covariance matrix S, in the feature space to represent the local feature distribution. With the model, the difficulty of adding the new important item to a local area can be simply measured by the size of the local covariance ellipsoid represented by S. Given a type of feature space with a limited volume, such as color gamut, the larger size of the local ellipsoid indicates less space for adding a new salient item. This is because the item has to be outside of the ellipsoid to ensure that it appears to be unusual to the existing items inside the ellipsoid. In other words, there is little feature space excluding the large local ellipsoid for choosing an appropriate feature vector for a new salient item. Thus, the feature space is likely to be congested with a large covariance local ellipsoid.

We follow the procedure in [28] for quantitatively estimating the degree of visual clutter across an image. Interested readers can refer to [28] for more details about the procedure. With the method, the system can successfully identify cluttered regions due to the color clutter, the orientation clutter, the density clutter clutter (see Fig. 2), or their combination (see the clutter map in Fig. 1).

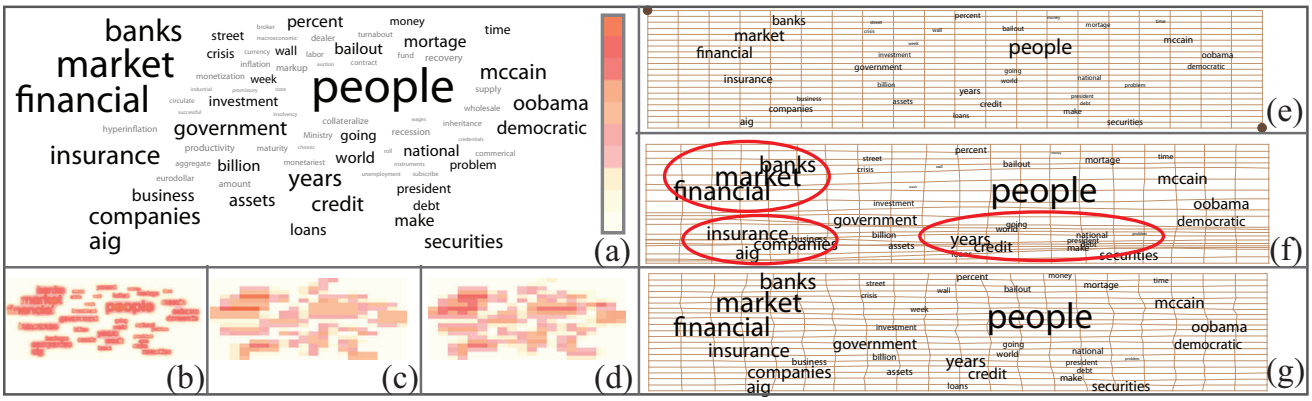


Fig. 3. (a) Word cloud where the tiny words in grey are filtered out before resizing. (b) Visual clutter map. (c) DOI map. (d) Significance map. (e) Uniformly scaled word cloud. (f) and (g) Results resized using the DOI map and the significance map, respectively.

4.1.3 *Significance Map*

The significance map is used to guide the later optimization process (see Section 5) for visualization resizing. The goal of the optimization is to shrink or stretch a visualization to fit any display, whereas the clutter magnitude in the visualization should not be increased and regions with high degrees of interest should be preserved. Therefore, the significance map W can be set up by combining the DOI map DOI and the clutter map C as $W = DOI * C$. The average of pixel significance within quad q is computed as the significance w_q for the quad. Furthermore, w_q is normalized such that $0 \leq w_q \leq 1$ (a larger value indicates higher significance).

4.2 Significance-Aware Grid and Adaptive Grid

The initial uniform grid can simplify the implementation and support faster performance. However, the uniform grid places an equivalent number of quads in every local region in spite of the significance of the regions, leading to linearization artifacts in the optimization process (see Fig. 4(b), where the words in the right side are shrunk too much). To reduce the artifacts and better approximate the optimal deformation, we adjust the initial grid to ensure that significant regions are covered by more quads than less important regions. The resulting grid is called a significance-aware grid. Two types of significance-aware grids are derived, such that the proposed framework is applicable to most visualizations. The first type is to directly deform the initial grid to attract the quads of less important regions to those of significant regions. It is adapted from the method in [35] by optimizing the following energy function:

$$\sum\nolimits_{\{i,j\}\in E} \sqrt{1+w_{ij}} \cdot (v_i - v_j)^2 \qquad (2)$$

where E is a set of edges in the grid, v_i and v_j are the positions of nodes i and j, and w_{ij} is the average weight of the quads that share the edge $\{i, j\}$. In an optimal scenario in which the energy is the minimum, the nodes in the interior of the significant regions become closer, thus attracting the surrounding nodes to the regions (see Fig. 4 (c) and (f)). Compared with the energy function in [35], our energy function uses $\sqrt{1+w_{ij}}$ rather than $1+w_{ij}$ to prevent a quad attracting too many neighboring nodes (see Fig. 4 (d) and (g)). We also tested with other choices such as w_{ij} and w_{ij}^2, and found that $\sqrt{1+w_{ij}}$ produced better results in general. This optimization problem is a nonlinear least squares optimization problem and can be solved iteratively to approximate the optimal node positions of the grid. This approach is simple for implementation and does not require changing the grid topology. In addition, because this is done only once in the preprocessing step, the resizing performance remains the same.

However, this method may introduce linearization artifacts (the important words inside the red ellipses in Fig. 4(g) are too small). To tackle this problem, an adaptive grid is used as a second type of significance-aware grid. The basic idea is simply to use a quad tree to partition the visualization and ensure that significant regions are covered by more quads. Fig. 4(h) shows the result by the adaptive grid (Fig. 4(e)). The implementation becomes more complicated compared with the first grid type, but the results look better.

5 Optimized Resizing

This section describes the resizing algorithm of ViSizer. It is adapted from a continuous image warping method [35]. Our method is different from the image warping method in three aspects. First, a completely different significance map is derived to guide the optimization process. Second, the energy function is tailored for visualizations (Section 5.2). Third, besides automatically finding an optimal scaling factor for a focus region, our method also allows users to specify an expected one for the region (Section 5.3).

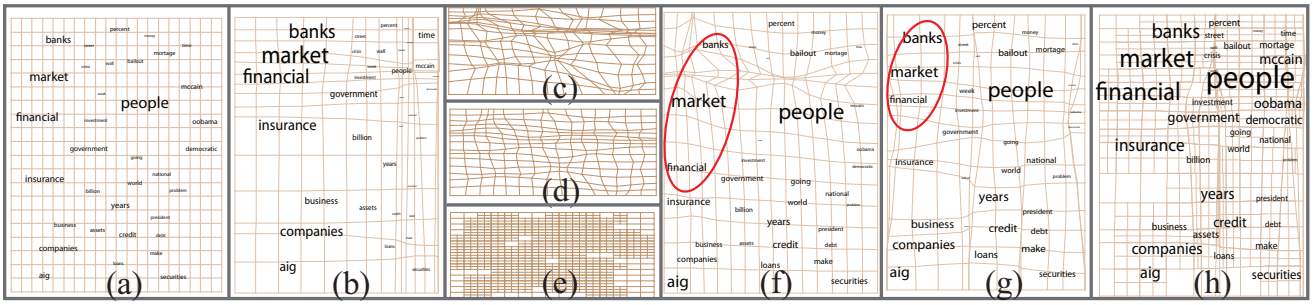


Fig. 4. Results obtained by resizing Fig. 3(a) horizontally. (a) Uniformly resized result. (b) Bad result produced by the initial grid where the words on the right are overly compressed. (c) and (d) Significance-aware grids of Fig. 3(a) created using the energy functions defined in [35] and in Equation (2), respectively. (e) Adaptive grid of Fig. 3(a). (f)-(h): Results produced by the grids shown in (c), (d), and (e), respectively.

Given a visualization, we place a grid on it and denote the grid by $G = \{V, E, F\}$, where V, E, and F represent the nodes, edges, and quads of G, respectively. Let n_v, n_e, and n_f be the numbers of nodes, edges, and quads. We have $V = [v_0^T, v_1^T, \cdots, v_{n_v}^T]^T$, and $v_i \in \Re^2$ represents the node position of node i. The optimization is to transform G judiciously for creating a new grid $G' = \{V', E', F'\}$, so as to ensure the uniform scaling of salient quads and the distortion distribution to other quads without increasing visual clutter. A new visualization can then be created by space interpolation according to the new grid G'.

5.1 Non-Linear Least Squares Optimization

An energy function is used to transform the resizing problem to an optimization problem. We follow [35] and define a quad deformation term for quantifying the non-uniform quad distortion and an edge bending term for evaluating the edge bending. In the following, we briefly describe the energy function and more details can be found in [35].

The quad deformation energy is defined for a quad to ensure that it is uniformly scaled. Given a quad f_k, its uniformly deformed version is $f_k' = s_k f_k$ where s_k is a 2×2 uniform scaling matrix. We mathematically formulate the energy for all quads F in a least-squares system as $||W_F F' - W_F S F||^2$ where $F = [f_0, f_1, \cdots, f_{n_f-1}]^T$, and W_F and S are the matrices whose element at the u^{th} row and the v^{th} column is defined as

$$S_{uv} = \begin{cases} s_k & \text{if } u = v \\ 0 & \text{otherwise} \end{cases} \quad W_{F,uv} = \begin{cases} \sqrt{\omega_{f_k}} & \text{if } u = v = k \\ 0 & \text{otherwise} \end{cases}$$

The quad can be represented as $f_k^T = q_k V$ where q_k is a $4 \times n_v$ matrix and its element at the u^{th} row and the v^{th} column is defined as

$$q_{k,uv} = \begin{cases} 1 & \text{if } v = i \\ -1 & \text{if } v = j \\ 0 & \text{otherwise} \end{cases}$$

where i and j are the node indices of the u^{th} edge of f_k. We let $Q = [q_0^T, q_1^T, \cdots, q_{n_f^3-1}^T]^T$, the quad set F can be derived by $F = QV$. Therefore, the total quad deformation energy is

$$||W_F Q V' - W_F S Q V||^2 \tag{3}$$

The edge bending energy is used to preserve the grid edge orientation. Given an edge $e_k \in E$, its uniformly scaled version is $e_k' = l_k e_k$ where l_k is a 2×2 uniform scaling matrix. The energy for all the edges E can be defined in a least-squares system as $||W_E E' - W_E L E||^2$ where $E = [e_0^T, e_1^T, \cdots, e_{n_e-1}^T]^T$, and L and W_E are matrices whose element at the u^{th} row and the v^{th} column is defined as

$$L_{uv} = \begin{cases} l_k & \text{if } u = v \\ 0 & \text{otherwise} \end{cases} \quad W_{E,uv} = \begin{cases} \sqrt{\omega_{e_k}} & \text{if } u = v = k \\ 0 & \text{otherwise} \end{cases}$$

Let $e_k = v_i - v_j = h_k V$, where h_k is a $1 \times n_v$ vector and its v^{th} element $h_{k,v}$ can be defined as

$$h_{k,v} = \begin{cases} 1 & \text{if } v = i \\ -1 & \text{if } v = j \\ 0 & \text{otherwise} \end{cases}$$

Let $H = [h_0^T, h_1^T, \cdots, h_{n_e-1}^T]^T$, we can obtain $E = HV$. Therefore, the total edge bending energy is

$$||W_E H V' - W_E L H V||^2 \tag{4}$$

The optimal nodes positions V' of the grid can be approximated by minimizing the total energy:

$$\underset{V', S, L}{\text{argmin}} \; ||W_F(QV' - SQV)||^2 + ||W_E(HV' - LHV)||^2$$

It can be viewed as an over-determined system $AV' = b(V)$, where $A = [Q^T W_F^T, H^T W_E^T]^T$ and $b(V) = [V^T Q^T S^T W_F^T, V^T H^T L^T W_E^T]^T$. Hence, we minimize

$$\underset{V'}{\text{argmin}} \; ||AV' - b(V)||^2 \tag{5}$$

It is a non-linear least squares optimization problem that can be approximated by iteratively updating the node positions. In our experiments, the initial guess

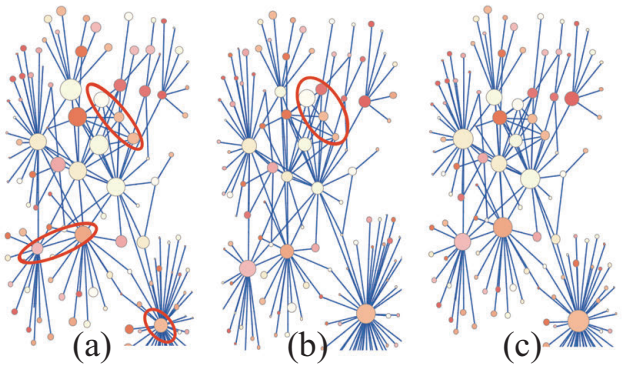


Fig. 5. (a) Result without any smoothing. (b) and (c) Results in which the scaling factors are equalized using the original and our adapted energy functions, respectively. (c) is the best result because the relative sizes of the nodes (indicated by the red ellipses in (a)) are preserved better than those of (a) and (b).

V^0 is obtained by homogeneously resizing the original visualization to the target display. See [21] for a more detailed description for solving the problem.

5.2 Quad Transformation Smoothing

The scaling factors in each local region must be smoothed out to prevent the distortion of an important object caused by the different scaling factors of the surrounding regions. Fig. 5(a) shows a resized visualization without quad transformation smoothing. The salient nodes marked by the red ellipses are clearly distorted. We formulate the smoothing problem as an optimization problem.

$$\sum_{k\in F}\sum_{q\in N(k)} w_q(s'_k - s'_q)^2 + \sum_{k\in F} w_k(s'_k - s_k) = 0$$

where $N(k)$ denotes the quads surrounding the quad k, w_q is the average significance of all nodes in the grid, and w_k indicates the significance of the quad k. Compared to the smoothing function defined in [35], our function uses w_q rather than $0.5(w_q + w_f)$ to weight $(s'_k - s'_q)^2$. Fig. 5 shows the different resizing results using the original function (Fig. 5(b)) and ours (Fig. 5(c)). Generally, our tailored function can produce better results with less distortion to the salient objects in the red ellipses. The smooth scaling s'_k can be estimated by minimizing the function. This process is repeatedly performed after we have obtained s_k at every iteration of the optimization.

5.3 Multi-Focus+Context Visualization

The scaling factors S in $b(V)$ are automatically determined during the optimization. This allows for creating distortion-free results, as the all visual items are uniformly scaled. The distortion is largely absorbed by the empty space. Fig. 3(g) and Fig. 4(h) present two examples of distortion-free results in which the words are uniformly scaled based on the change of the display size. The spatial relations between the words are mostly preserved and thus the results are consistent with the original ones.

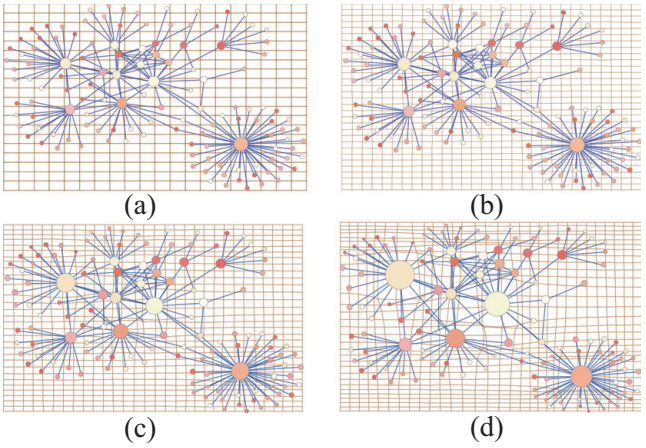


Fig. 6. Multi-focus+context visualization. (a) Result created by uniform resizing. (b)-(d) Results created with the specified scaling factors: $s_k = 1$, $s_k = 2$, and $s_k = 4$ for important graph nodes, respectively.

Many applications usually prefer the distortion-free results. However, focus+context techniques are still needed in some scenarios, especially when the display size is limited [19]. Techniques such as data abstraction might have more or less limitations as we discussed in Section 2. Our framework naturally supports multi-focus+context visualization during the resizing process to address this issue. It allows a user to specify a desired uniform scaling factor δ for any object in the resized visualization. Thus, the quad scaling factors of the quads covering the objects are fixed to the constant value δ specified by the user during the optimization. This can produce a result similar to multi-focus+context visualization (see Fig. 6). Therefore, it can be regarded as a combination of significance-aware focus+context visualization and visualization resizing techniques. ViSizer transforms a visualization through a grid and thus can seamlessly provide the background grid to support the user's comprehension of geometric distortion. It has been reported that the background grid can help improve the accuracy of visualization performance [22], [38].

6 Results and Discussion

In this section, we demonstrate the effectiveness of our framework and show how we can apply it to different visualizations. The techniques described in this work were implemented by Java and Prefuse. All results were generated on an Apple Macbook Pro with an Intel Core i7 2.66GHz CPU and 4 GB Ram.

6.1 Experiments

6.1.1 Word Clouds

In the first experiment, we tested ViSizer for showing its usefulness for resizing word clouds. The resizing technique is important for word cloud visualizations. It is usually difficult to resize a word cloud to fit a new display. Uniformly scaling a word cloud to a smaller display may create a word cloud in which many words are too tiny to be easily recognized. Regenerating a new word cloud might be another option. However, the word cloud, e.g., re-created by [34] or [8], can be totally different from the original one. As they are based on either a random algorithm [34] or a force-directed algorithm [8], they usually fail to create stable word clouds with a specified aspect ratio. Additionally, context-preserving word clouds [8] use the relative positions of the words in the clouds to encode important semantic information in the original text. This requires that the relative positions between words in the original word cloud should be preserved.

The framework views a word as a visual item. Each word is attached to a grid quad via an anchor point. After the gird is deformed, the anchor point positions are adjusted by interpolating the four nodes of the quad. The size of each word is changed based on how the size of the associated grid quad changes.

We generated a context-preserving word cloud by a force-directed algorithm [8] using a real dataset with $13,828$ news articles spanning one year (from 2008 to 2009) that were related to American International Group (AIG). In the word cloud, semantically-similar words get close to each other. As we wanted to shrink the word cloud for a small display, we filtered out tiny words in grey that are almost unrecognizable in the target display (see Fig. 3 (a)). Figs. 3(b)-(d) show the visual clutter, DOI, and significance maps of the word cloud used for guiding the resizing optimization process. Fig. 3(a) (right) presents the color encoding scheme for these maps. We resized the word cloud vertically to reduce it to half its size.

Fig. 3(e) shows a uniformly resized result in which previously large words become unnoticeable, not to mention the smaller ones. Fig. 3(f) is a result created by our method guided only by the DOI map. Compared to Fig. 3(e), it distributed most of the distortion to the empty space, thus reserving more room for important keywords. However, the words inside the red ellipses were overly packed, making it challenging for users to recognize the words quickly. We can remedy this problem with the help of the visual clutter map (see Fig. 3(b)) that can inform the optimization process of the crowding degree in every local region, preventing the words from being excessively packed. Fig. 3(g) presents the result created by using the significance map (Fig. 3(d)). We can clearly observe that the overcrowding problem was fixed using the perception-based clutter measure.

We further tested different types of grids. Fig. 4(a) is a uniformly deformed result in which most of the words become tiny. We can use a uniform grid for resizing the word cloud in most cases. However, it cannot always produce a good result due to the linearization artifacts in the optimization process. Fig. 4(b) shows an example where only the left part of the word cloud is deformed correctly. We then used three types of grids (Figs. 4(c)-(e)) from the original word cloud (Fig. 3(a)) to reduce the linearization artifacts. Fig. 4(f) shows a result based on the grid in Fig. 4(c) generated by the original energy function [35]. We can see that the grid is distorted too much. The words in the red ellipse in Fig. 4(f) are too separated and their relative sizes change a great deal.

Fig. 4(g) shows the result based on the grid in Fig. 4(d) created by our adapted energy function (Eq. (2)). The grid was modestly adjusted without too much distortion. However, some words originally neighboring each other, e.g., in the red ellipse, are still far away. To solve the problem, we used an adaptive grid (Fig. 4(e)) to resize the word cloud. Fig. 4(h) presents the result in which the words are nicely packed and their relative sizes and positions are mostly preserved. We found that the adaptive grid generally worked better than other grids. However, it took more time (2.553 seconds in this experiment) than the significance-aware grids (within 1 second).

6.1.2 Graph Visualization

The second experiment was conducted to show the usefulness of our technique in a graph visualization. Regenerating a new graph layout by, e.g., a force-directed algorithm, for a different display is usually time-consuming and the new layout could be totally different from the original. Furthermore, most existing algorithms do not take into account the different display aspect ratios and cannot make efficient use of the screen space. In resizing a graph, every graph node is regarded as a visual item. Each node is attached to a grid quad via an anchor point. After the grid is deformed, the anchor points are adjusted by interpolating the four nodes of the quad. The size of the graph node is changed based on how the size of the associated grid quad changes.

We tested ViSizer with two real graph datasets. One is a social network dataset from Prefuse with 129 nodes and 161 edges, while the other contains major airline routes of Northwest Airlines in the United States with 235 nodes and $2,101$ edges. We used both the size and color of a graph node to encode its degree. The graph of the social network data was generated by a force-directed algorithm. Fig. 6(a) presents a uniformly resized result where nodes become too small to analyze. Figs. 6(b)-(d) demonstrate the resized results with an increasing s_k (1, 2, and 4) manually specified for the important graph nodes. The quads covering the nodes were

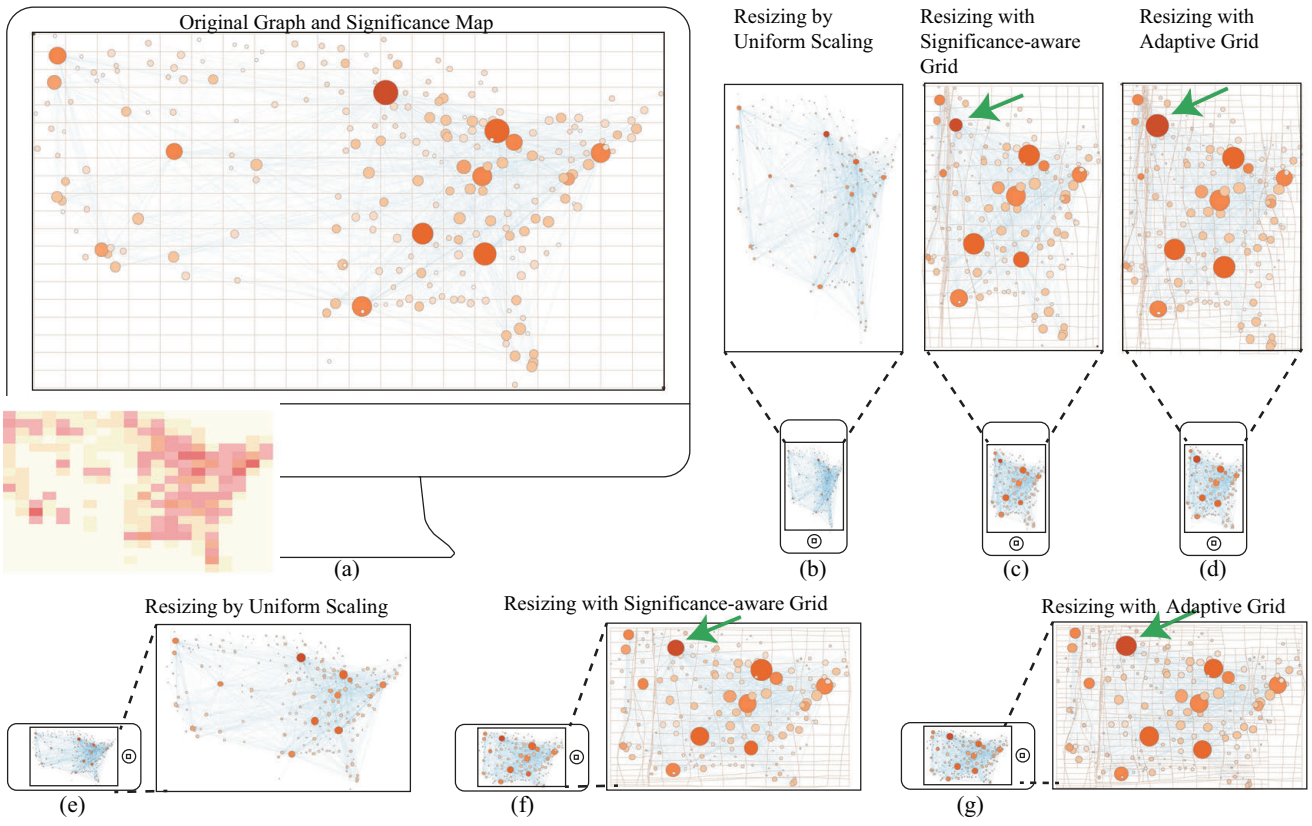


Fig. 7. Results created by resizing a graph visualization originally shown on a 27 inch display with 1920×1200 pixels (top-left) to a 3.5 inch display with 960×640 pixels in different orientations using uniform scaling, ViSizer with a significance-aware grid, and ViSizer with an adaptive grid. ViSizer makes more efficient use of the small displays than uniform scaling. The adaptive and the significance-aware grids both work well in most cases for maintaining original information. However, the significance-aware grid might have a chance to produce artifacts. The node indicated by the green arrow is not well preserved in the results of the significance-aware grid.

uniformly expanded by the specified s_k to produce results similar to focus+context visualization. ViSizer distributed the distortion to the less important nodes and empty space across the entire visualization.

Fig. 7 shows a typical use of ViSizer for resizing a graph originally shown on a 27 inch display with 1920×1200 pixels to a 3.5 inch display with 960×640 pixels in horizontal and vertical orientations. The used airline data contains spatial information for each graph node. Uniformly scaling the graph to the small display with a very different aspect ratio produced a squeezed visualization (see Fig. 7 (b)). In addition, it is difficult for a user to explore and interact with the graph in the much smaller display (see Fig. 7 (b) and (e)) because the graph nodes are barely discernible in such a display. Simply increasing the sizes of the nodes would cause the graph nodes to overlap one another. On the other hand, we can see that there is a great deal of white space in the left part of the graph. Therefore, we could compress the white space to make room for enlarging significant regions.

Fig. 7 (c) and (f) show the results created by ViSizer with the significance-aware grid, while Fig. 7 (d) and (g) present the results created by ViSizer with the adaptive grid. These results assigned more display space to important nodes (the larger the nodes, the more important they are) by compressing the white space in the graph while still preserving the overall graph structure. Comparing these results, we can observe that the two types of grids produced similar results. Nevertheless, the adaptive grid (Figs. 7(d) and (g)) works slightly better than the other (Figs. 7(c)and (f)) in preserving the original information. For instance, the node indicated by the green arrow is distorted in the results of the significance-aware grid. Since the spatial information and the sizes of the nodes are useful and important for analysis, it is therefore desirable to preserve the information.

6.1.3 *Scatterplots*

The third experiment was conducted to demonstrate the use of our technique for scatterplots. The scatterplots used in this experiment were created by projecting a high dimensional data set to two dimensional space with multidimensional scaling. Therefore, the x and y axes do not a concrete meaning. Fig. 8 (left) shows an original scatterplot of the data from INSPIRE [37] on a 21.5 inch display with 1920×1080

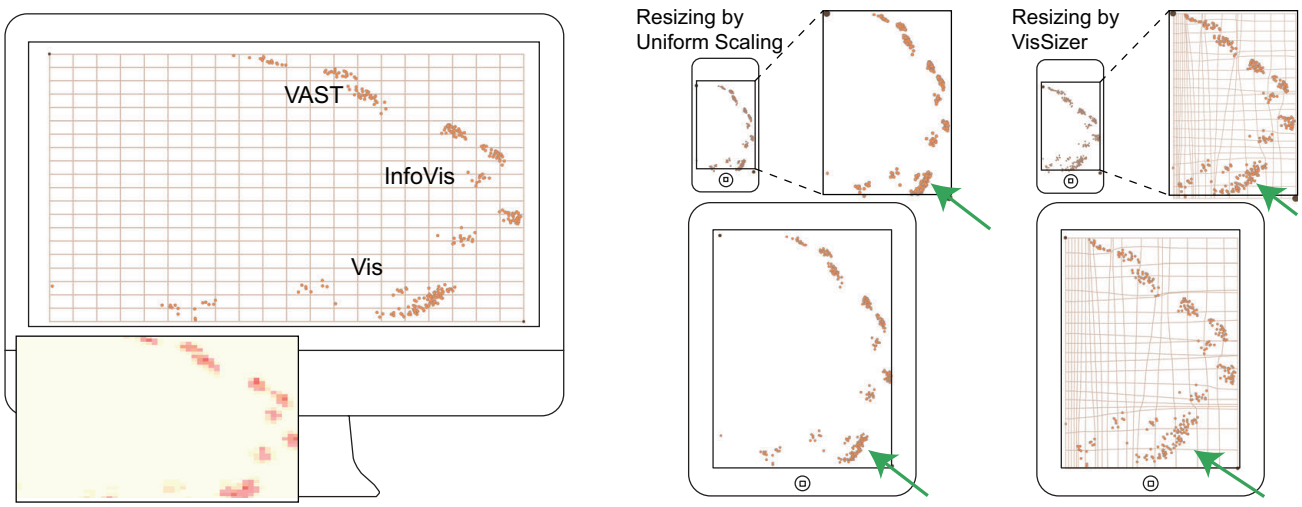


Fig. 8. Results created by resizing a scatterplot originally shown on a 21.5 inch display with 1920×1080 pixels (left) to a 3.5 inch display with 960×640 pixels and a 9.7 inch display with 1024×768 pixels by uniform scaling (middle) and ViSizer (right). Through the judicious removal of white space, ViSizer assigns more space to important areas and maintains the overall pattern, thus showing the relationships among the points in clusters more clearly.

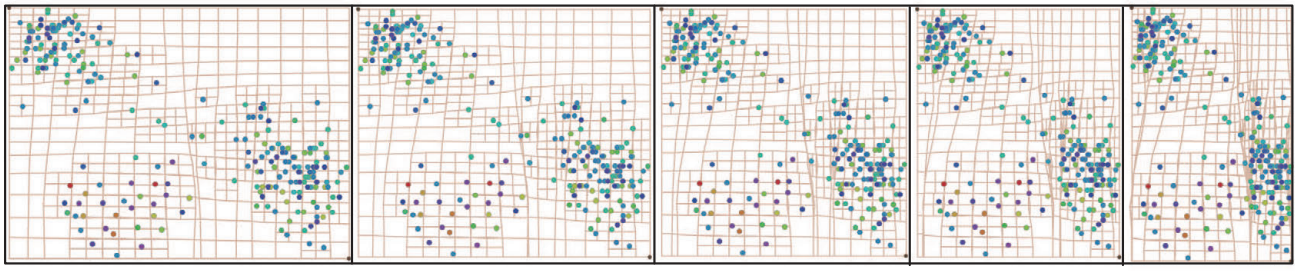


Fig. 9. Smooth transition between a visually faithful representation of a visualization and a representation meant to facilitate selection.

pixels. It reveals the topic distribution of IEEE Vis, IEEE InfoVis and, IEEE VAST proceeding papers published from 2006 to 2008. Each paper is represented by a point with a fixed size in the scatterplot. This scatterplot is very sparse and has a great deal of white space. Fig. 8 (middle) presents two uniformly scaled scatterplots that are shown on a 3.5 inch display with 960×640 pixels and a 9.7 inch display with 1024×768 pixels, respectively. These results (particularly the smallest scatterplot) look squeezed. Additionally, the points in the scatterplots are too small and too close to be visually distinguished from one another. For example, the points inside the cluster indicated by the green arrow look quite cluttered in these results. It would be challenging for a user to interact with these cluttered points to facilitate visual analysis. One solution might be to decrease the point sizes to reveal the relations among the points. However, this would result in very tiny points in the small scatterplots that are hardly discernible. Applying a simple magnifying lens does not work either for distortion-sensitive applications. In addition, these techniques do not take different aspect ratios into account and cannot preserve the overall pattern.

Fig. 8 (right) shows the results of VisSizer for the same smaller displays. Every point in the scatterplot is regarded as a visual item and considered as equally important. It is attached to a grid quad and its position is adjusted by interpolating the four nodes of the quad after the quad deformation. From these results we can see that in contrast to uniform scaling, ViSizer works better in preserving the overall pattern (or shape) of the original scatterplot. It compressed only the white space and allocated more room to more significant regions, thus helping to clearly reveal details of a cluster (see the cluster indicated by the green arrow). This would be especially helpful when a viewer needs to interact with the points for detailed analysis of the data. Moreover, ViSizer employed a significance-aware grid to deform the scatterplot and thus had a strict spatial constraint for minimizing the chance of distorting the overall pattern. The results show the effectiveness of ViSizer for resizing a larger scatterplot to smaller ones with very different aspect ratios.

Fig. 9 presents a smooth transition between a visually faithful representation of a visualization and

a representation meant to facilitate selection. The smooth transition of the change can provide an alternative to background grids for enhancing the understanding of geometric deformation introduced by the resizing process. The clutter map (see Fig. 1) shows that all three clusters in the scatterplot are visually cluttered. Although the two clusters (bottom-right and top-left) exhibit a higher level of density, there is much more color diversity in the bottom-left clusters. According to our clutter estimation measure, both cases suggest a certain level of feature congestion in the feature space, thus leading to a high degree of visual clutter. For illustration, we manually specified the DOI map and assigned a higher importance level to the bottom-left cluster. Combining the DOI map and the clutter map, we generated a significance map which indicates that the bottom-left cluster is more significant than other regions.

Guided by the significance map, the scatterplot can be prudently resized to ensure that most distortion is distributed to less significant regions, such that significant regions as well as the overall layout are maintained. From Fig. 9, we can see that the significant region and the overall layout are well preserved consistently during the transition. In addition, ViSizer mostly compress only the empty space among the clusters at the beginning. Only when no more empty space is available does ViSizer start to deform the two less significant clusters (top-left and bottom-right clusters). This result demonstrates the effectiveness of our clutter estimation measure and shows that ViSizer can successfully generate a smooth transition of change, which is useful for comprehension of the resizing deformation.

6.2 Time Performance

Fig. 10 shows the time of resizing different visualizations using the significance-aware grid and the adaptive grid. For each visualization, we recorded the time needed for resizing it to its original $95\%, 90\%, \cdots, 5\%$. Table 1 presents the average time performance. As shown in our results, the adaptive grid usually creates better results than the significance-aware grid (see Fig. 4 and 7). A possible reason is that the adaptive grid has a regular shape which can better preserve the overall structure of a visualization. However, this advantage is gained at the cost of performance and using the adaptive grid generally needs more time for resizing a visualization (see Fig. 10 and Table 1). We can also observe that it took much less time to resize scatterplots than other visualizations using the adaptive grid. This is because the scatterplots used in the experiments have more empty space than other visualizations, thus resulting in far less grid quads and a smaller linear system to optimize.

The implementation of the resizing framework using an adaptive grid is more complicated and challenging. Based on our experimental results, we suggest that the adaptive grid should be used when users need more accurate results and do not care much about the performance. We recommend the significance-aware grid for most applications although it might slightly deform the items.

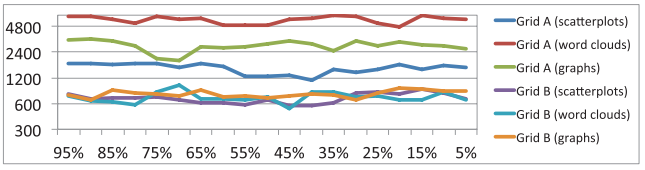


Fig. 10. Time (measured in milliseconds) needed for resizing different visualizations using the adaptive grid (Grid A) and the significance-aware grid (Grid B).

TABLE 1
Average time performance using different grids

Visualization	Significance-Aware Gid	Adaptive Grid
Scatterplots	698ms	1557ms
Graphs	780ms	2864ms
Word clouds	713ms	5748ms

6.3 Discussion

ViSizer is not fully optimized for time performance, but it can be easily adapted to run in real time by precomputing keyframes for different sizes then interpolating these keyframes. We will optimize the current system and employ a GPU-accelerated technique [16] to solve the linear system involved in the optimization. This would enable many interesting interactive applications, including smart support for window resizing, for semantic zooming in ZUIs, and for rendering legible overview insets.

As discussed in the previous experiments, ViSizer can also ease the difficulty of target selection in small displays. Although researchers have developed many target selection techniques by reducing the distance D or increasing the width W of the target, they usually do not scale well to the situation in which multiple targets are crowed together [2], especially for small displays. In contrast, our framework naturally scales well to the situation by the significance-guided resizing method and the embedded focus+context scheme. In particular, ViSizer can facilitate target selection by expanding the target and removing empty space between the cursor and the target. That is it can facilitate target selection by both decreasing D and increasing W. With the technique, one interesting application is that a user facing a large (such as wall-sized) display uses her hand-held device to select visual items, which are otherwise too tiny and indistinguishable to be selected readily. In the future, we want to extend our framework to support a more

sophisticated target selection method by considering the likelihood of selection.

The main advantage of ViSizer is that it can uniformly scale important items through diverting deformation to other regions without increasing clutter magnitudes with the help of the perception-based significance map. Moreover, by distributing deformation to only the white space, ViSizer can produce distortion-free results in which every local non-empty region is well preserved. ViSizer deforms the whole visualization through a gird and thus it can also largely retain the overall pattern of the original visualization. Nevertheless, the deformation does have an impact on the alignment of objects, which could result in misinterpretation of the data and unfair comparisons between visual items. Therefore, ViSizer may not be appropriate for resizing visualizations requiring accurate relative positioning of visual items. In contrast, ViSizer can be used primarily for visualization tasks that do not require accurate and quantitative understanding of the data items. For instance, it can help users better discern data patterns or distribution in a smaller hand-held device.

The framework can produce multi-focus+context results by highlighting important regions while compressing others. The important regions are uniformly scaled to the sizes specified by a user. As other focus+context visualizations, these distorted results might lead to impaired spatial comprehension [7]. ViSizer uses background grids [22], [38] to provide necessary visual cues, which helps improve comprehension of the distortion and task accuracy. Although this problem can be alleviated to some extent by the background grids, the inevitable distortion may have negative impacts on visualizations that require accurate interpretation of visual variables such as position and size of visual items. Distortion and misalignments could also prevent users from comparing data accurately. This is a limitation of our approach. We believe that this problem could be addressed by developing better user interactions, providing animated transition, and/or providing better background visual cues other than simple background grids. We plan to study this problem and improve our framework to better support comparison tasks in the future.

7 Conclusions and Future Work

This work introduces a perception-based resizing framework for automatically resizing a visualization to any display size without introducing additional visual clutter. Prominent objects can either be uniformly scaled or fixed to a size specified by the user during the resizing process. The deformation introduced by the resizing operation is distributed to less important regions globally over the visualization. Our framework targets at non-space-filling visualizations that usually contain some empty space among visual items. As for space-filling visualizations, some of the visual items (e.g., rectangles in the a treemap) can be very large and uniform and they could therefore be treated like empty space. We plan to extend our framework for space-filling visualizations.

Acknowledgments

This research was supported in part by the HP Labs and U.S. National Science Foundation through grants CCF-0808896, CNS-0716691, CCF 0811422, CCF 0938114, and CCF-1025269.

References

[1] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM Transactions on Graphics*, vol. 26, no. 3, 2007.

[2] R. Balakrishnan, ""beating" fitts' law: virtual enhancements for pointing facilitation," *International Journal of Human-Computer Studies*, vol. 61, no. 6, pp. 857–874, 2004.

[3] S. Baldassi, N. Megna, and D. C. Burr, "Visual clutter causes high-magnitude errors," *PLoS Biology*, vol. 4, no. 3, p. e56, 2006.

[4] J. Böttger, U. Brandes, O. Deussen, and H. Ziezold, "Map warping for the annotation of metro maps," *IEEE Computer Graphics and Applications*, vol. 28, no. 5, pp. 56–65, 2008.

[5] S. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia, "3-dimensional pliable surfaces: for the effective presentation of visual information," in *Proceedings of the ACM symposium on User interface and software technology*, 1995, pp. 217–226.

[6] S. Carpendale, J. Ligh, and E. Pattison, "Achieving higher magnification in context," in *Proceedings of the ACM symposium on User interface software and technology*, 2004, pp. 71–80.

[7] A. Cockburn, A. K. Karlson, and B. B. Bederson, "A review of overview + detail, zooming, and focus+context interface," *ACM Computing Surveys*, vol. 41, no. 1, pp. 1–31, 2008.

[8] W. Cui, Y. Wu, S. Liu, F. Wei, M. X. Zhou, and H. Qu:, "Context preserving dynamic word cloud visualization," *IEEE Computer Graphics and Applications*, vol. 30, no. 6, pp. 42–53, 2010.

[9] W. Cui, H. Zhou, H. Qu, P. C. Wong, and X. Lis, "Geometry-based edge clustering for graph visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1277–1284, 2008.

[10] P. Dragicevic, S. Chatty, D. Thevenin, and J.-L. Vinot, "Artistic resizing: A technique for rich scale-sensitive vector graphics," in *Proceedings of the ACM symposium on User interface software and technology*, 2005, pp. 201–210.

[11] T. Dwyer, K. Marriott, and P. J. Stuckey, "Fast node overlap removal," in *Proceedings of the international conference on Graph drawing*, 2005, pp. 153–164.

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

[13] N. Elmqvist and J.-D. Fekete, "Hierarchical aggregation for information visualization: Overview, techniques and design guidelines," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 3, pp. 439–454, 2010.

[14] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner, "Hierarchical parallel coordinates for exploration of large datasets," in *Proceedings of IEEE Visualization*, 1999, pp. 43–50.

[15] G. Furnas, "Generalized fisheye views," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 1986, pp. 16–23.

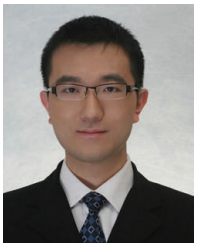
[16] N. Galoppo, N. K. Govindaraju, M. Henson, and D. Manocha, "Lu-gpu: Efficient algorithms for solving dense linear systems on graphics hardware," in *Proceedings of the ACM/IEEE conference on Supercomputing*, 2005.

[17] S. E. Hudson and I. Smith, "Ultra-lightweight constraints," in *Proceedings of the ACM symposium on User interface software and technology*, 1996, pp. 147–155.

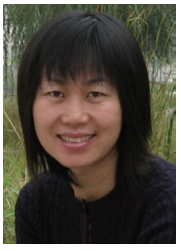
[18] B. Jenny and L. Hurni, "Studying cartographic heritage: Analysis and visualization of geometric distortions," *Computers & Graphics*, vol. 35, no. 2, pp. 402–411, 2011.

[19] B. Karstens, R. Rosenbaum, and H. Schumann, "Information presentation on mobile handhelds," in *Proceedings of IRMA2003*, 2003.
[20] T. A. Keahey and E. L. Robertson, "Techniques for non-linear magnification transformations," in *Proceedings of the IEEE Symposium on Information Visualization*, 1996, pp. 38–45.
[21] C. T. Kelley, *Iterative Methods for Optimization (Frontiers in Applied Mathematics)*, 1st, Ed. Society for Industrial Mathematics, 1987.
[22] H. Lam, R. A. Rensink, and T. Munzner, "Effects of 2D geometric transformations on visual memory," in *Proceedings of the symposium on Applied perception in graphics and visualization*, 2006, pp. 119–126.
[23] J. D. Mackinlay, "Automating the design of graphical presentations of relational information," *ACM Transactions on Graphics*, vol. 5, no. 2, pp. 110–141, 1986.
[24] R. D. Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," *Chemometrics and Intelligent Laboratory Systems*, vol. 50, no. 1, pp. 1–18, 2000.
[25] T. Munzner, F. Guimbretiere, S. Tasiran, L. Zhang, and Y. Zhou., "TreeJuxtaposer: Scalable tree comparison using focus+context with guaranteed visibility," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 453–462, 2003.
[26] W. Peng, M. O. Ward, and E. A. Rundensteiner, "Clutter reduction in multi-dimensional data visualization using dimension reordering," in *Proceedings of IEEE Symposium on Information Visualization*, 2004, pp. 89–96.
[27] R. Rosenholtz, Y. Li, J. Mansfield, and Z. Jin, "Feature congestion: A measure of display clutter," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2005, pp. 761–770.
[28] R. Rosenholtz, Y. Li, and L. Nakano, "Measuring visual clutter," *Journal of Vision*, vol. 7, no. 2, p. Article 17, 2007.
[29] M. Sarkar and M. H. Brown, "Graphical fisheye views," *Communications of the ACM*, vol. 37, no. 12, pp. 73–83, 1994.
[30] M. Sarkar, S. S. Snibbe, O. J. Tversky, and S. P. Reiss, "Stretching the rubber sheet: a metaphor for viewing large layouts on small screens," in *Proceedings of the ACM symposium on User interface software and technology*, 1993, pp. 81–91.
[31] A. Shamir and O. Sorkine, "Visual media retargeting," in *ACM SIGGRAPH ASIA Courses*, 2009.
[32] E. R. Tufte, *The Visual Display of Quantitative Information*. Graphics Press, 2001.
[33] R. van den Berg, F. W. Cornelissen, and J. B. T. M. Roerdink, "A crowding model of visual clutter," *Journal of Vision*, vol. 9, no. 4, p. Article 24, 2009.
[34] F. B. Viegas, M. Wattenberg, and J. Feinberg, "Participatory visualization with Wordle," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1137–1144, 2009.
[35] Y.-S. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee, "Optimized scale-and-stretch for image resizing," *ACM Transactions on Graphics*, vol. 27, no. 5, 2008.
[36] L. Wolf, M. Guttmann, and D. Cohen-or, "Nonhomogeneous content-driven video-retargeting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2007, pp. 1–6.
[37] P. C. Wong and J. Thomas, "Visual analytics: Building a vibrant and resilient national scienc," *Information Visualization*, vol. 8, no. 3, pp. 302–308, 2009.
[38] A. Zanella, S. Carpendale, and M. Rounding, "On the effects of viewing cues in comprehending distortions," in *Proceedings of NordiCHI*, 2002, pp. 119–128.

Yingcai Wu received the BEng degree in computer science and technology from the South China University of Technology in 2004 and the PhD degree in computer science from the Hong Kong University of Science and Technology in 2009. Since June 2010, He has been a postdoctoral researcher at the Visualization and interface Design Innovation (VIDi) research group in the University of California, Davis. Prior to his current position, he was a postdoctoral researcher in the Hong Kong University of Science and Technology. His research interests include information visualization, visual analytics, medical visualization, and user interface design.



Xiaotong Liu is a Ph.D student in Department of Computer Science & Engineering of The Ohio State University, currently working in Prof. Han-Wei Shen's GRAVITY Group. His research interests lie in Information Visualization, Visual Analytics, Computer Graphics and Human Computer Interaction. Xiaotong Liu received his Bachelor degree in Software Engineering at Shanghai Jiao Tong University in 2011, and worked at Microsoft Research Asia in Internet Graphics Group with Dr. Shixia Liu and Dr. Yingcai Wu on information visualization research from 2010 to 2011.



Shixia Liu is a lead researcher in the Internet Graphics Group at Microsoft Research Asia. Her research interest mainly focuses on interactive, visual text analytics and interactive, visual graph analytics. She received a B.S. and M.S. in Computational Mathematics from Harbin Institute of Technology, a Ph.D. in Computer Aided Design and Computer Graphics from Tsinghua University. Before she joined MSRA, she worked as a research staff member and research manager at IBM China Research Lab, where she managed the departments of Smart Visual Analytics and User Experience.

Kwan-Liu Ma is a professor of computer science and the chair of the Graduate Group in Computer Science at the University of California, Davis. He leads the VIDi research group and directs the DOE SciDAC Institute for Ultrascale Visualization. Professor Ma received his PhD degree in computer science from the University of Utah in 1993. He was a recipient of the PECASE award in 2000. His research interests include visualization, high-performance computing, and user interface design. Professor Ma was a paper chair of the IEEE Visualization Conference in 2008 and 2009, and an associate editor of IEEE TVCG in 2007-2011. He co-founded the IEEE Pacific Visualization Symposium in 2008 as well as the IEEE Symposium on Large Data Analysis and Visualization (LDAV) in 2011. Professor Ma presently serves on the editorial boards of the IEEE CG&A, the Journal of Computational Science and Discoveries, and the Journal of Visualization. He is an IEEE Fellow. Contact him via email: ma@cs.ucdavis.edu.