

Graphoto: Aesthetically Pleasing Charts for Casual Information Visualization

Ji Hwan Park, Arie Kaufman, *Fellow, IEEE*, and Klaus Mueller, *Senior Member, IEEE*

Abstract—Graphoto is a framework that automatically generates a photo or adjusts an existing one to match a line graph. Since aesthetics is an important element in visualizing personal data, Graphoto provides users with aesthetically pleasing displays for casual line graph information visualization. More specifically, after creating a line graph of the input data, a photo that resembles the input data on the line graph is selected from a photo archive. If a selected photo does not match the line graph, we deform the photo to match the line graph. Once the photo matches the line graph of the data, the line graph is superimposed on the photo using different colors and styles depending on the user's artistic preferences. Additional embellishments are integrated into Graphoto, such as tick marks and text labels. We further present a user study to show the effectiveness of Graphoto in terms of data interpretation and aesthetics.

Index Terms—Information visualization, visual embellishments, aesthetics, casual information visualization



1 INTRODUCTION

The effects of visual embellishments are a controversial topic in the visualization community. Some argue that data has to be shown without any visual embellishment (e.g., [1]). However, recent work in information visualization suggests that a chart with visual embellishments can have a positive effect on memorability and aesthetics without affecting the interpretation of data (e.g., [2], [3]).

Due to the development of sensor technologies, people can easily obtain data from various sources related to their personal lives. There are many factors affecting comprehension in visualizing such data, including aesthetics, which is an important element in visualizing personal data [4]).

In this paper, we propose an automatic visual embellishment approach, called Graphoto, for casual information visualization of the line graph (see Sidebar). In Graphoto, we automatically adjust or generate a photo to match a line graph, where the photo is seamlessly integrated into the line graph to improve aesthetics. Graphoto is designed for the backdrop or lock screen of various types of devices such as computers, mobile devices, and TVs. Our target users are people who want to understand their personal data with an aesthetically pleasing display, rather than to perform focused tasks in data analysis. We are particularly interested in time-varying data from a single source, for example, energy consumption, exercise logs, and bank account information because these data are readily available for most people. In order to visualize the data, we choose a traditional line graph, which is one of the most popular methods to depict data. We deploy the line graph to show, for example, weekly, monthly, and yearly views of the data. We select a photo as a medium to convey

aesthetics in Graphoto because people have access to many photos through the Internet and many of them might be aesthetically pleasing to them.

In order to reduce distraction from putting a photo as a background under the line graph, we select a photo that is perceptually similar to the line graph by searching for it, or manipulating an existing photo. In addition, we offer various visual embellishments to improve the perceived accuracy of the visualization. We evaluate Graphoto in a user study that measures differences between Graphoto and traditional line charts in terms of aesthetics and data interpretation. The results of our study show that Graphoto is as good as traditional line graphs in terms of accuracy and completion time and is more aesthetically pleasing than the line graphs alone.

2 GRAPHOTO DESIGN

The main objective of Graphoto is to improve the visual aesthetics and illustrate a series of data for casual information visualization. In order to achieve this goal, we use visually-pleasing photos associated with positive emotions [5] such as beautiful natural scenes. In Gestalt principles of figure and ground [6], lower regions and objects with a wide bottom and narrow top could be perceived as figures. This perfectly matches with mountains in a photo. Thus, in our current framework, we specifically choose photos with mountains as sources for Graphoto and then superimpose the line graphs of input data onto the mountains to focus on the line graphs. The pipeline of Graphoto is shown in Figure 1. We find or manipulate a photo to combine the photo and the traditional line chart to show the data with an aesthetically pleasing display. In order to visualize the data that is divided into different categories such as weekly, monthly and yearly views, we choose a simple line graph (with a few fluctuations), a complicated line graph (with many fluctuations), or a long

• J. Park, A. Kaufman, and K. Mueller are with the Computer Science Department, Stony Brook University, Stony Brook, NY 11794-2424. E-mail: {jihwpark, ari, mueller}@cs.stonybrook.edu.

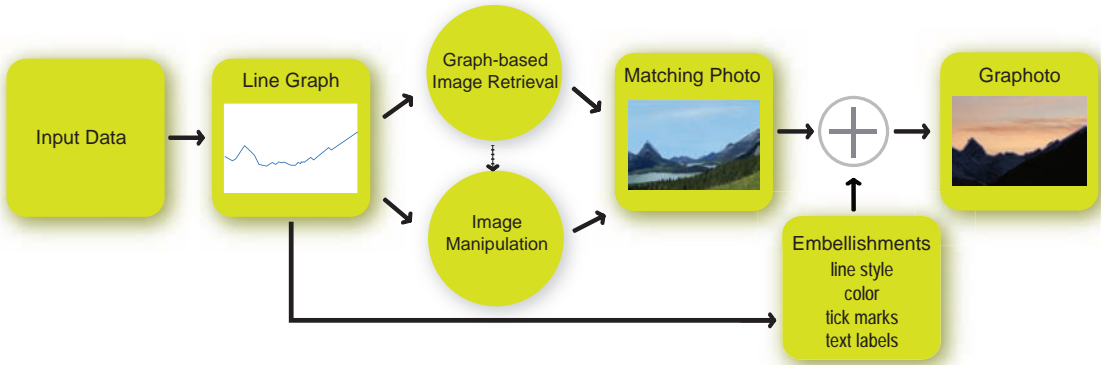


Fig. 1: An overview of Graphoto. After creating a line graph of the input data, a matching photo to the line graph is found via graph-based image retrieval (GBIR) and/or created or modified by image manipulation (IM). Various visual embellishments are generated and applied to the matching photo.

line graph (one that cannot fit in one photo). This allows us to visualize the data with different granularities. Social comparisons are one of the effective ways to encourage users to change their behavior [4]. A stacked line graph is used to enhance the aesthetic aspects of charts for comparing multiple time series. Even though the stacked line graph has a legibility issue, it works for some cases such as detecting large differences in a few time series with similar overall shapes [7]. In Graphoto, we need to compare data from a user with the average of his/her neighbors as social comparisons. Thus, we also use the stacked line graph for such a comparison. Since every user has different artistic taste, we provide various techniques to accommodate their artistic preferences, including six line styles, four colors, and embellishments such as tick marks and text labels. These techniques are applied to the matching photo.

2.1 Graph-based Image Retrieval

In Graphoto, we utilize a photo to enhance the aesthetics of a line graph. However, if the salient parts of a photo in the background of the line graph are dissimilar to the line graph, it makes the line graph difficult to understand. Therefore, in our framework, we select a mountain photo which resembles the line graph of the input data as we mentioned. Since we store a large quantity of photos with mountains obtained from a user’s photo album(s) or downloaded from the web into a photo archive, an algorithm for finding a matching photo should be efficient in searching for a photo in such a photo archive. Our method is based on the sketch-based image retrieval approach by Eitz et al. [8] which is simple, yet still finds a well-matched image. In their system, a user draws a sketch as an input to find the matching image. We instead use the line graph of the data as input. After obtaining the line graph, we choose a photo from our photo archive, and resize the line graph to match the size of the photo. We compute the similarity between the line graph and the photo as follows:

- 1) Remove noise in the photo by applying a Gaussian lowpass filter.
- 2) Extract edges in the photo via Canny edge detection.

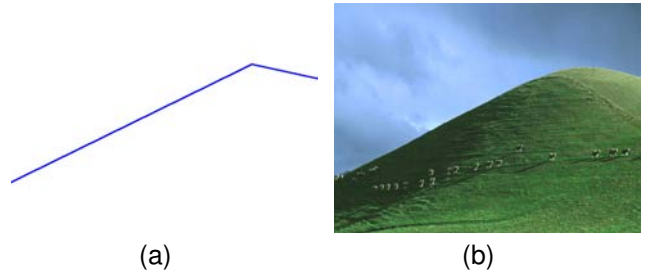


Fig. 2: An example of graph-based image retrieval. (a) A line graph of the input data, and (b) a retrieved photo.

- 3) Divide the photo into cells, where the size of each cell is 0.25 percent of the size of the photo.
- 4) Compute a histogram of oriented gradient (HOG) with six bins in each cell.
- 5) Calculate the distance between HOGs from the line graph and the photo by using L_1 -norm.

Once we procure all distances between the line graph and all photos in the photo archive, a photo is selected, which has the shortest distance and hence is the most similar to the line graph in the photo archive. We noted that HOGs of photos are computed only once when the photos are added to the photo archive.

Our purpose in graph-based image retrieval (GBIR) is to find a matching photo, which is perceptually similar to the line graph. For this purpose, we further refine our results with a different method, which is less efficient, but perceptually better. Beaudot and Mullen [9] found that a long edge was perceptually salient even when it was faint. Based on this observation, Bhat et al. [10] introduced a method to compute the local gradient saliency from edge lengths and orientations. In order to incorporate this method in our pipeline, we first find twenty matching photos from the HOGs. Next, we calculate the edge lengths of the photos as follows:

- 1) Compute the local edge magnitude (e_m) and orientation (e_o) of each pixel (p_i) by applying a steerable filter.
- 2) Normalize the local edge magnitude of each pixel as follows:

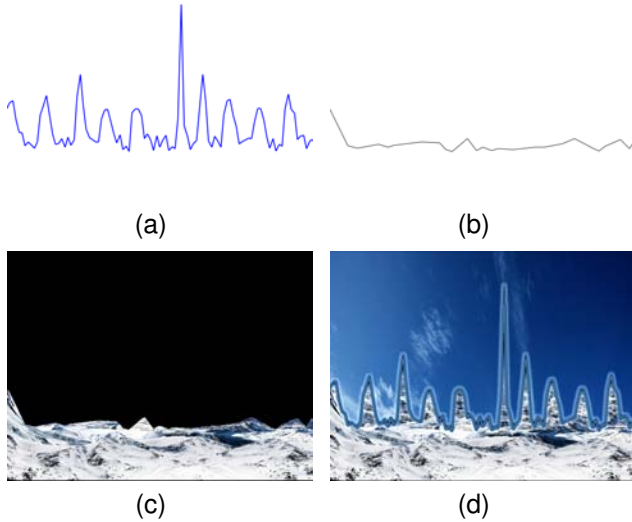


Fig. 3: An example of image manipulation. (a) A line graph of the input data, (b) a simplified line graph, (c) a deformed photo with the simplified line graph, and (d) the final photo.

$$e'_m = \frac{e_m - \mu_n}{\sigma_n - \varepsilon}$$

where μ_n and σ_n are the average and the standard deviation of p_i 's neighboring edge magnitudes.

- 3) Compute the length of two sub-edges of each pixel located in two opposite directions ($e_o, e_o + \pi$). We initially set the length of the sub-edges to 0. If p_i 's neighboring pixel along a direction (e_o or $e_o + \pi$) has a similar local edge orientation, we add its local edge magnitude to the length of a sub-edge and then searching its neighborhood along that direction. We repeat this process several times (60 times for the results in this article).
- 4) Sum the local edge magnitude and the length of two sub-edges of each pixel and then set it as the edge length for each pixel.

Once we have computed the edge length of each pixel, we determine the visibility of each edge; an edge is considered visible if the length of an edge is greater than a certain threshold. In our examples, we set 0.1 as the threshold. Next, we use the edges from the long edge detection instead of the Canny edge detection to compute the distance between the input line graph and a photo. The rest of our process is the same as the process mentioned above. Our graph-based image retrieval with the long edge detection further finds a perceptually matching photo. Figure 2 shows a result of graph-based image retrieval. We note that we could apply long edge detection directly to the initial stage without computing HOGs, but it takes more time. In our experiment, our method produced almost the same results compared to using only the long edge detection, but was faster.

2.2 Image Manipulation

If the line graph is stable such that the values of data do not frequently surge or plummet, our GBIR method may successfully find a matching image due to the simplicity of the line graph. However, values of the input data can have high variability and the resultant line graphs could assume various shapes and be rather complex. Therefore, our GBIR method may fail to find a matching image even after searching from hundreds of thousands of photos. To deal with this issue in Graphoto, we manipulate an image, which enables us to create a matching photo from almost any photo. In order to deform a photo to the desired line graph, we use Thin Plate Spline (TPS) [11] as follows:

- 1) Select a photo and extract an object in the foreground of the photo by using either the GrabCut segmentation method [12] or pre-segmented photos. We use the GrabCut method, included in OpenCV, to compute results for this article.
- 2) Compute the boundary of the extracted object and the line graph. The boundary of the line graph comprises the input data, and x - and y -axes on both sides of the line graph.
- 3) Sample the same number of points on both boundaries and find a point located at the bottom left of each boundary. We connect these two points, and then the rest of the points based on their distances from the two connected points.
- 4) Compute two TPS functions with these samples and then transform each pixel in the photo to a new position through the TPS functions.
- 5) Superimpose only the transformed object on a simple background and apply one of the line styles (Section 2.3).

If the input line graph is too jagged, however, our experiments showed that deforming a photo by using TPS was not very successful because some parts of the photo were too distorted. To solve this problem, we first compute the local minima of the input line graph and then create a simplified line graph by connecting the local minima with the first and last points of the line graph. With the simplified line graph, we deform a photo to the simplified line graph via TPS. Next, we fill the gap between the original input line graph and the simplified line graph by using a region filling method from Criminisi et al. [13]. To use the region filling method, we have to define a target and a source region. In our approach, the target region is the gap between the original input line graph and the simplified line graph, and the source region is a deformed photo matching the simplified line graph. The target region is then filled with patches from the source region. The filled photo is superimposed on the background, as shown in Figure 3. This approach can be easily extended to a stacked line graph by segmenting multiple regions, which correspond to regions between the lines in the line graph.

In some cases, an input line graph is too long to fit into one photo. To solve this problem, we divide the line graph into several parts, where the width of each part is the

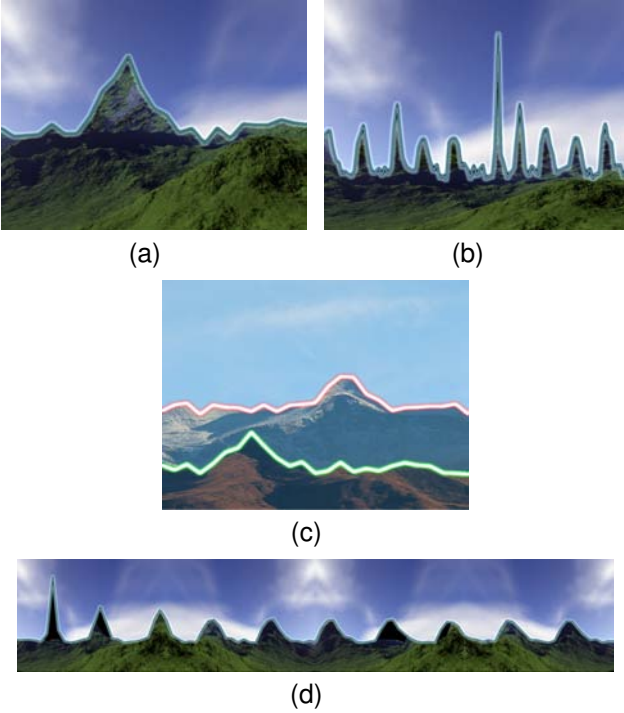


Fig. 4: An example of Graphoto with various line graphs. (a) A simple line graph, (b) a complicated line graph, (c) a stacked line graph, and (d) a long line graph.

same as the width of a photo. We use the same photo in creating matching photos for all parts. Before transforming a photo for each part, a photo corresponding to every other part of the line graph is flipped horizontally to prevent discontinuity in the final Graphoto. We then deform the photo for each part and then stitch these together.

We note that we use a simple scene as a background, e.g., a sky without any attractive or outstanding objects. If we used a scene with any attractive or outstanding objects as a background, it could attract the users' attention to the background of a photo, not to the foreground. Figure 4 shows an example of Graphoto with different line graphs using our image manipulation (IM) technique.

2.3 Line Styles

Once we acquire the matching photo, we superimpose the line graph on the matching photo to show the data more clearly. However, a simple visualization of the line graph may not be appealing to the user. In Graphoto, we decorate the line graph with various line styles for an aesthetically pleasing display. For example, if users prefer a plain line graph to a line graph with a 3D line shape, they can choose a style of the line graph close to the plain line graph. In the current Graphoto, we provide six different line styles to accommodate users' discrepancies on aesthetic preferences, as illustrated in Figure 5. Some styles mimic a 3D line, and the other styles are close to 2D plain line graphs. To emulate a 3D line effect, we first generate three blurred lines by applying Gaussian lowpass filters with three different kernel sizes and standard deviations to the line graph. Once we

obtain the three blurred lines, $Blur_1$, $Blur_2$, $Blur_3$, we compute a gradually faded line, I_{norm} as:

$$I_{norm} = \frac{line}{line_{max}}, \quad line = \sum_{i=1}^3 Blur_i \quad (1)$$

where $line_{max}$ is the maximum value in the $line$. It is deployed as a base component for 3D line effects.

The first style is a simple 2D line graph, which simply visualizes the line graph with a certain color; this is further described in the next subsection (Figure 5(a)). To apply a 2D line graph style to the line graph, we simply replace pixels in the photo corresponding to ones in the line graph with pixels in the colored line graph.

The second style is to highlight parts of the photo corresponding to the line graph (Figure 5(b)). Since we can conserve all contents in the photo and show data at the same time in this style, we offer it to users as a line style. In order to highlight the parts corresponding to the line graph, we compute a new highlighted photo, $I_{highlight}$ as:

$$I_{highlight} = I_{photo} * \beta + I_{photo} * I_{norm} \quad (2)$$

where I_{photo} is the input photo, β is a weight to darken the photo, which is less than 1.0, and I_{norm} is a weight, which is the faded line, as computed in Eq. 1.

A neon style is motivated by a neon lamp (Figure 5(c)). This style emphasizes the data from the line graph as glowing with a white color and surrounding areas around the line graph with a different color. To achieve this effect, we compute a photo with a neon styled line, I_{neon} as,

$$I_{neon} = I_{photo} * (1 - I_{norm}) + (color + line) * I_{norm} \quad (3)$$

where $color$ is a specific color selected by the user and $line$ is an unnormalized faded line in Eq. 1

The next style is a 3D line style, which is actually an opposite effect of the neon style (Figure 5 (d)). Unlike the neon style, the center of the line is colored and the border parts of the lines are smoothly faded. Thus, it can be obtained similarly to the neon style, I_{3D} as,

$$I_{3D} = I_{photo} * (1 - I_{norm}) + (color + line_{max} - line) * I_{norm} \quad (4)$$

For the smoke style (Figure 5 (e)), we need to generate a smoke effect along the line graph. We can create a smoke effect through numerical equations, or synthesize smoke from a small texture [14] and use that. In this work, we choose the latter approach, which is simpler and faster. In a smoke style, we first select a small smoke texture as an input to our texture synthesizer. Next, we synthesize a smoke texture, in which the size of the texture is the same as the size of the photo. We then generate the mask which is a faded line, I_{norm} . The mask is applied to the synthesized texture to obtain the line graph filled with smoke and the masked texture is superimposed the same way as the 2D line graph. Lastly, in order to generate a noise style (Figure 5 (f)), we randomly generate pixels along the line graph and superimpose the generated pixels on the photo.

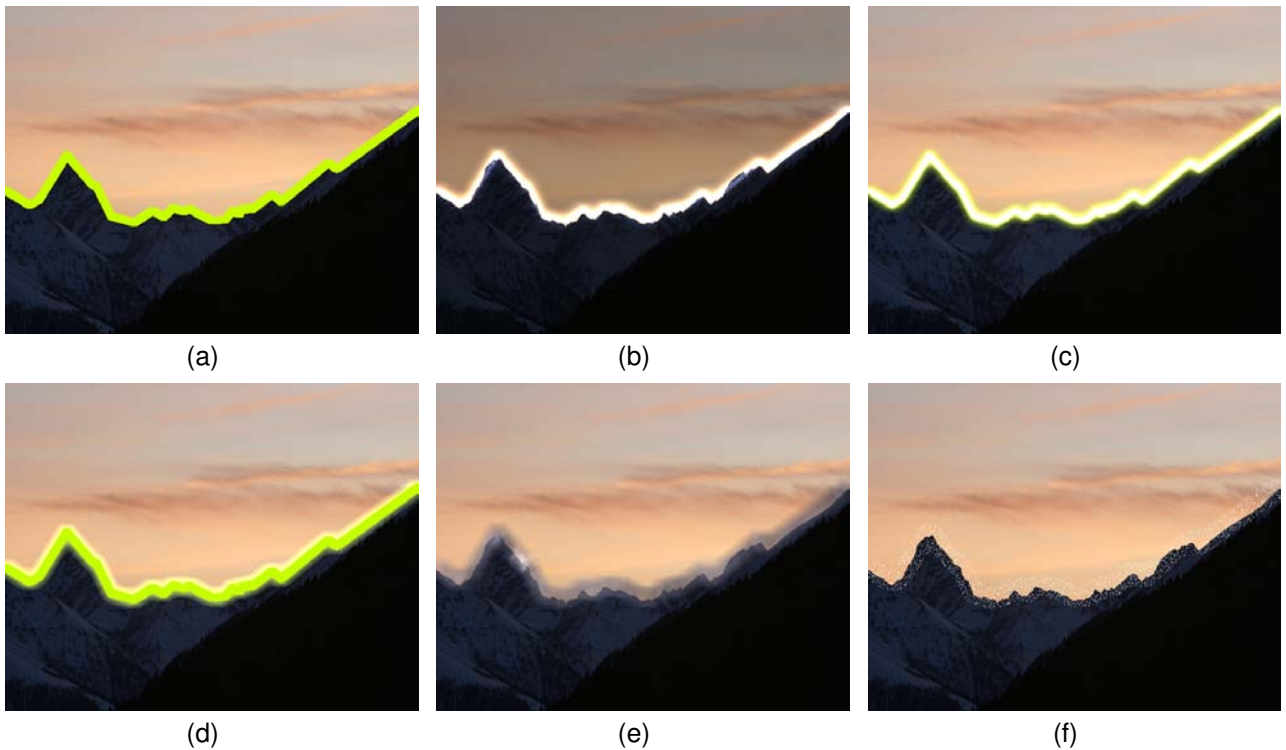


Fig. 5: Various line styles: (a) 2D line, (b) highlighting, (c) neon, (d) 3D line, (e) smoke, and (f) noise.

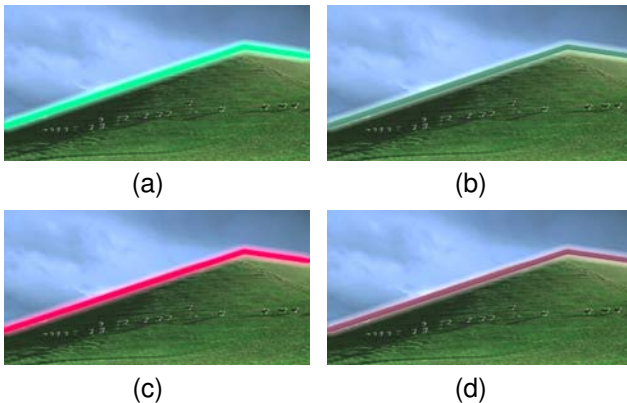


Fig. 6: Examples of line colors: (a) an analogous color with the maximum saturation and lightness, (b) an analogous color with the average saturation and lightness, (c) a complementary color with the maximum saturation and lightness, and (d) a complementary color with the average saturation and lightness.

2.4 Line Colors

Color is one of the essential components to perceive a photo. Harmony among colors in a photo determines whether a photo is aesthetically pleasing or not [15]. Tokumaru et al. [16] proposed a scheme based on hue distribution for color harmony. Thus, in addition to various line styles, different colors can be exploited in Graphoto for the purpose of satisfying people’s different visual preferences. In Tokumaru et al.’s work [16], there are

eight types of hue distribution. For a single line graph (a simple/complicated/long line graph), we chose two types among them, type i and type I, which are similar to complementary colors and analogous colors in the hue space, respectively. We harmonize between the average color of a photo and the color of the line graph in the same way as the neon, 3D line, and 2D line styles as follows:

- 1) Convert the RGB color space to the HSL color space.
- 2) Calculate the average hue, saturation, and lightness of the photo.
- 3) Compute a complementary color that lies opposite the average hue in the HSL color space and an analogous color that is adjacent to the average hue in the HSL color space.

Since colors with the same hue can be distinguished by saturation and lightness [17], we also provide two additional colors by adjusting the saturation and lightness of the complementary and analogous colors. We set the maximum saturation and lightness for both colors to emphasize the line graph. Therefore, we offer in total four colors for the single line graph. An example illustrating this concept is shown in Figure 8. We note that we only consider colors for single line graphs in this study, since we would need to separate colors of lines in a stacked line graph, which is complicated.

2.5 Other Embellishments

Tick marks sometimes play a part as a reference structure to help users to interpret the data. Thus, we offer users ability to display tick marks for accuracy, which are smoothly

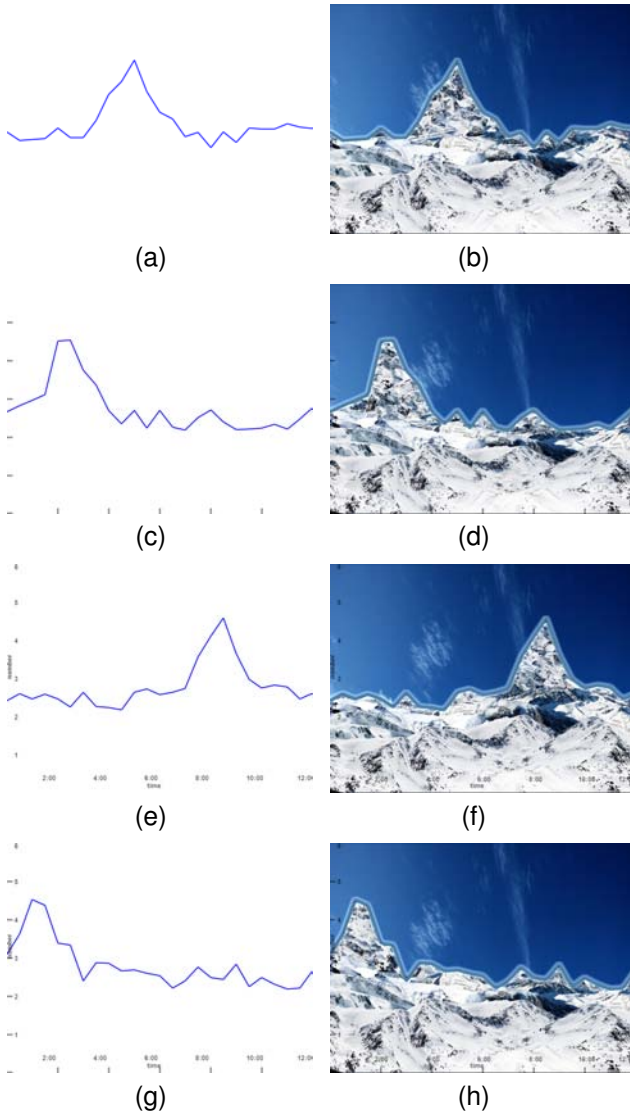


Fig. 7: Examples of other embellishments for traditional line charts (a,c,e,g) and Graphoto (b,d,f,h): (a,b) no additional embellishments, (c,d) tick marks, (e,f) text labels, and (g,h) tick marks and text labels.

integrated into the photo. In order to include tick marks visible enough to comprehend the data, the tick marks should be neither too faint nor too strong. Additionally, text labels are also important in understanding the data in the line graphs, identifying the values represented in the line graph. Therefore, in Graphoto, we smoothly darken the areas to be unobtrusive by combining black tick marks/text labels and the photo through alpha blending where the tick marks and the text labels are supposed to be located, which look like a date stamped on a photo.

All of these features are illustrated in Figure 7. We note that the boundary of a photo acts as the axes of the line graph, so we do not explicitly put those axes into the display. Sometimes, it is difficult to distinguish tick marks and text labels in dark-colored photos. In order to solve this problem, white tick marks and text labels can be used

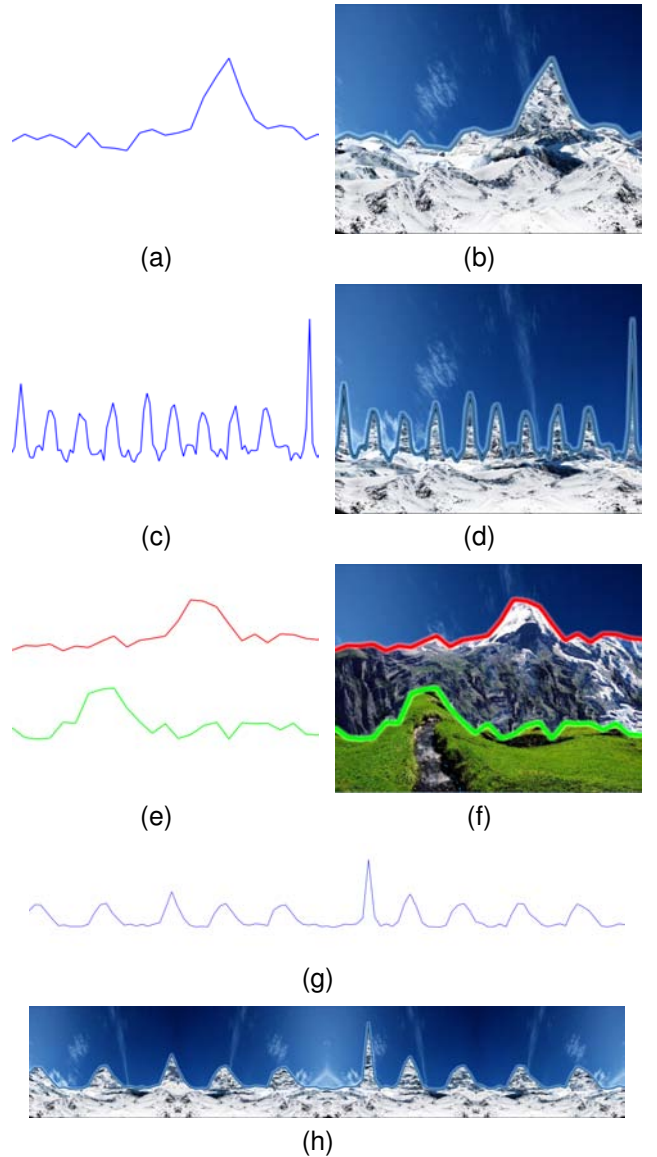


Fig. 8: A set of example line graphs (a,c,e,g) and Graphoto images (b,d,f,h) used for our user study.

instead of black ones.

3 USER STUDY

Our goal in this study was to determine the impact of Graphoto in understanding data, compared to the traditional line graphs. We designed a user study to investigate the difference between Graphoto and traditional line graphs in terms of correctness, completion time, and aesthetics.

3.1 Design & Tasks

We have the following factors in our study:

- Visualization Type (V): Graphoto, traditional line chart
- Chart Type (C): Simple (24 points), Complicated (96 points), Long (96 points), Stacked (24 points for each line)

- Additional Embellishment (A): Plain (without any additional embellishment), Tick marks, Text labels, Tick marks and text labels

In Graphoto, we used a 3D line style and an analogous color with the average saturation and lightness for simple, complicated, and long charts. We used the same colors for stacked charts in Graphoto and traditional line charts. We designed the study for a within-subjects analysis, which included factors, V, C, and A. Each participant completed a total of $2(V) \times 4(C) \times 4(A) = 32$ trials. Figures 7 and 8 illustrate a set of example images used in our study.

A goal of Graphoto is to illustrate the input data for casual information visualization. To test whether we can achieve this goal, we chose tasks that were commonly used for a line graph [18] and modified them for our purpose.

- Max (with six optional answers): find the specific time period with the highest value(s) during the whole time period in a line graph (a simple, complicated, and long line graph) or each line graph (a stacked line graph).
- Slope (with six optional answers): find the specific time period with the highest increase during the whole time period in a simple, complicated, and long line graph.
- Comparison (with three optional answers): compare a value in the upper line graph to one in the lower line graph at a specific time in the stacked line graph.

Additionally, we asked participants about the aesthetics of each image on a 5-point Likert scale from 1 (Not at all) to 5 (Almost totally) (“*Did you think this chart is visually pleasing?*”).

The order of tasks was not counterbalanced; the max task was always shown first to the participants, followed by the slope or the compare task (similar to previous work [18]). The order of chart type and additional embellishments were counterbalanced between participants using a Latin square design. The order of visualization type was randomized.

3.2 Participants and Procedure

We posted our study on Amazon Mechanical Turk with a series of images, containing two types of line graphs such as Graphoto and traditional line graphs. We created four types of synthetic input data for control over the characteristics of line graphs (simple, complicated, long, and stacked line graphs), by following the previous work [18]. The experiment was divided into two sections. In the first section, each participant completed the 32 trials. In the second section, we asked participants about which visualization type would be good for visualizing their personal data (“*Do you think that a chart with a photo is better than a traditional chart (line graph) to show data when it is used as the lock screen of an electronic device (a cell phone, a tablet, a computer, or a TV)?*”).

We recruited participants from Amazon Mechanical Turk, and 80 participants from the USA (49 females, 31 males) successfully completed our study. In order to participate in our experiment, each participant was required to have a greater than 95% approval rate in Amazon’s

TABLE 1: Effects of the visualization type (V), the chart type (C), and additional embellishments (A) on correctness and completion time.

Task	Factors	correctness		time	
		F	p	F	p
Max	V	$F(1, 79) = 0.031$	*	$F(1, 79) = 1.399$	*
	V×C	$F(3, 237) = 0.943$	*	$F(3, 237) = 1.854$	*
	V×A	$F(3, 237) = 0.571$	*	$F(3, 237) = 0.551$	*
Slope	V	$F(1, 79) = 2.777$	*	$F(1, 79) = 0.654$	*
	V×C	$F(3, 237) = 0.798$	*	$F(3, 237) = 0.499$	*
	V×A	$F(3, 237) = 0.954$	*	$F(3, 237) = 0.333$	*
Comparison	V	$F(1, 79) = 3.189$	*	$F(1, 79) = 0.02$	*
	V×A	$F(3, 237) = 0.161$	*	$F(3, 237) = 1.605$	*

*= $p > 0.05$

system. If a participant completed a study too fast (< 5 min), we assumed it was a false result and thus discarded it. They took on average about 21 min to finish our experiment and were paid \$0.40 upon completing our full experiment. Participants had various occupations including bank teller, teacher, IT consultant, homemaker, and designer. Their ages ranged from 20 to over 50 and expertise with visualization was between minimal experience and good understanding and experience ($M=2.36$, $SD=1.08$) on a scale from 1 (novice: no experience) to 5 (expert: authoritative understanding and experience). All participants self-reported that they were not color-blind.

Before starting our experiment, all participants were asked to fill out a demographic questionnaire. We first provided instructions, followed by the tasks.

3.3 Results

We analyzed all the responses from the tasks. 75% (Max), 58% (Slope), and 67% (Comparison) of tasks were successfully completed, and participants took on average of 13 seconds (Max), 7 seconds (Slope), and 11 seconds (Comparison) to complete the tasks. We then ran a repeated measure analysis of variance (RM-ANOVA) and Tukey HSD test on the results. There was no significant effect of (i) the visualization type and (ii) the interaction between the visualization type and the chart type, and (iii) between visualization type and additional embellishments on correctness and completion time (see Table 1). For finding the highest value(s) in each line of a stacked line graph, there was no significant effect of different baselines between Graphoto and traditional line graphs ($F(1, 79) = 0.035$, $p > 0.05$). Unsurprisingly, the visualization type had a significant effect on aesthetics ($F(1, 79) = 55.15$, $p < 0.0001$), where Graphoto had on average a higher aesthetic score than the traditional line graph. Additionally, there was a significant effect of the interaction between the visualization type and the chart type ($F(3, 237) = 21.86$, $p < 0.001$), but not between the visualization type and the additional elements ($F(3, 237) = 0.5$, $p > 0.05$). Figure 9 shows a summary of responses from participants to our aesthetics question.

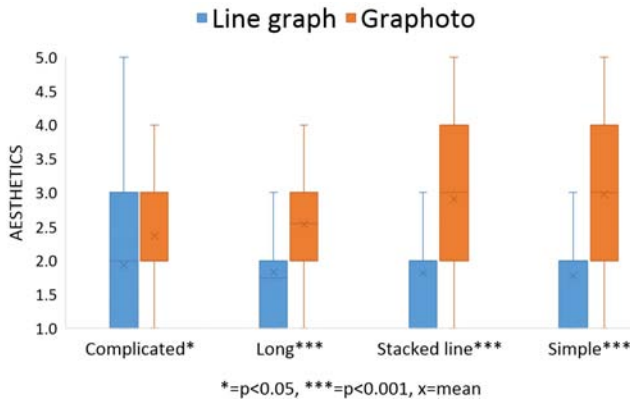


Fig. 9: Comparisons of Graphoto to a traditional line graph in terms of aesthetics.

3.4 Discussion

Our results confirm that Graphoto provides users with more aesthetically pleasing displays than a traditional line graph. More importantly, in Graphoto, users could answer questions as accurately and quickly as using traditional line graphs; this finding is similar that found in a previous study [19]. Based on these results, we believe that Graphoto would be a good technique for users who want to use it for visualizing their personal time-varying data.

Graphoto is targeted at visualizing personal time-varying data as the lock screen of various devices such as computers, mobile devices, and TVs. In our target applications, the input data from a single source is transmitted to a server, and we create a matching photo with embellishments on the server. We then transmit the matching photo to a target device. One of our target applications is household energy consumption data, where a smart meter collects data on consumption of electricity at certain intervals (15–60 minutes). In the current system, we can create a matching photo to the input data within 1–2 minutes by image manipulation (on a computer with Intel Xeon E5-2620 2.0GHz, 8 GB RAM and GeForce GTX 670 graphics with 2 GB memory). Our GBIR takes several hours to find a matching photo in a photo archive with millions of images. Thus, if we can find the matched photo in a photo archive, it would be appropriate for daily data, e.g., daily weather forecast. The retrieval time of our GBIR can be accelerated by computing the distances between the input line graph and the photos in parallel (e.g., a MapReduce framework) using a bag-of-features approach [8].

Our approach can be extended to other types of charts, such as treemaps and pie charts. For example, we can use several aerial images of rice fields and different cake images from the top view for a treemap and a pie chart, respectively. After segmenting the images, we are then able to deform each subfield or cake for each section of the charts. Our various line styles can be applied as area styles such that each section in a chart is an input of our method instead of a line graph.

There are some limitations to our work. First, a participant in our study expressed that our technique is useful

for understanding data as a big picture concept, but it can obscure the specifics of the data. This is because we decorate the input line graph with various effects. Thus, our technique would be appropriate for providing an overall trend rather than analyzing subtle changes over time, e.g., “When does a user consume the most electricity during a day?”, “Does a user consume more energy on the weekend or during the summer?”.

A stacked line graph might not be the best method for social comparisons because previous studies have shown that graph comparison is difficult with different baselines [18]. However, the main reason we used a stacked line graph in social comparison tasks is to enhance the aesthetic aspects of charts, as we mentioned earlier. Additionally, participants had no issue on our stacked line graph in the comparison tasks and stated that our chart is visually appealing, e.g., “It is more interesting and appealing to people rather than a boring graph.”.

Half of the participants (52%) expressed that their preference of traditional charts over Graphoto because they wanted to focus on the data even though Graphoto is visually pleasing and interesting. This result was also observed in other artistic visualizations, where some participants liked traditional ways to visualize their data [20]. Therefore, even though our method is as good as traditional line graphs in our target use cases, Graphoto might only be suitable for some users who like a visually pleasing display.

We conducted a simple user study to evaluate our technique on Amazon Mechanical Turk. A limitation of this evaluation is that workers only spent about 15 minutes to answer quantitative and qualitative questions in a limited space (mostly in their home), so most qualitative results were based on their initial reaction to Graphoto. Additionally, in the user study, we described that Graphoto can be used on a cell phone, a tablet, a computer, or a TV. If a user belongs to a single-person household, all these examples are personal computing devices. However, if a user is a member of a multi-person household, a cell phone and a tablet are personal computing devices, and a computer might be a shared device, while a TV is most probably a shared display device. To evaluate Graphoto on our target devices, we plan to conduct a longitudinal study in different household conditions such as single- and multi-person households to examine several aspects such as the changes of users’ personal data awareness and understanding by using Graphoto and their preferences to understand their personal data (Graphoto vs. traditional charts).

A photo in Graphoto does not contain any message. However, if we integrate a message into Graphoto by adjusting the color of an image defined by a user, it might successfully deliver a message. For example, a user defines a reddish image as a warning for their energy consumption, and then the image of Graphoto is turned to reddish when he/she consumes more energy than the average of his/her neighborhood. We note that we tested embedding a message into a photo by changing the content of the photo provided by us, but it didn’t work well because participants

interpreted the content in a different way. It caused them to spend more time to answer questions because they had to memorize its meaning if it was different from their interpretation.

Lastly, Graphoto contains tick marks and text labels, but there are other techniques to aid in understanding data such as grid lines and highlights. These elements may intrude on data interpretation and the aesthetic aspect if we combine these elements with Graphoto. Additionally, all participants used their desktop to conduct our study. The type of a device to perform tasks might influence users' correctness and/or completion time due to different sizes of charts. Because we are targeting different types of devices, we will investigate in the future the effects of different display sizes and of different elements.

4 CONCLUSION

We have presented Graphoto, a chart or line graph combined with a photo with the aim to show the data with an aesthetically pleasing look for casual information visualization. We provided a diverse set of techniques to obtain a matching photo and embellish the line graph of input data. We compared interpretation accuracy, completion time, and aesthetics for Graphoto and traditional line graphs. The comparison showed that Graphoto is better than the traditional line graph in terms of aesthetics, and users could understand data in Graphoto as accurately and quickly as with the traditional line graph.

ACKNOWLEDGMENTS

This work has been partially supported by the National Science Foundation grants IIP1069147, CNS1302246, CNS0959979, NRT1633299, CNS1650499, and IIS1527200.

References

- [1] E. Tufte, *The Visual Display of Quantitative Information*, Second edition. Graphics Press, 2001.
- [2] R. Borgo, A. Abdul-Rahman, F. Mohamed, P. W. Grant, I. Reppa, L. Floridi, and C. Min, "An empirical study on using visual embellishments in visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2759–2768, 2012.
- [3] M. A. Borkin, A. A. Vo, Z. Bylinskii, P. Isola, S. Sunkavalli, A. Oliva, and H. Pfister, "What makes a visualization memorable?" *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2306–2315, 2013.
- [4] D. Huang, M. Tory, B. Aseniero, L. Bartram, S. Bateman, S. Carpendale, A. Tang, and R. Woodbury, "Personal visualization and personal visual analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 3, pp. 420–433, 2015.

- [5] E. Dan-Glauser and K. Scherer, "The geneva affective picture database (gaped): A new 730-picture database focusing on valence and normative significance," *Behavior Research Methods*, vol. 43, no. 2, pp. 468–477, 2011.
- [6] J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, M. Singh, and R. von der Heydt, "A century of gestalt psychology in visual perception: i. perceptual grouping and figure-ground organization.," *Psychological bulletin*, vol. 138, no. 6, pp. 1172–1217, 2012.
- [7] R. Kosara, "Presentation-oriented visualization techniques," *IEEE Computer Graphics and Applications*, vol. 36, no. 1, pp. 80–85, 2016.
- [8] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa, "Sketch-based image retrieval: Benchmark and bag-of-features descriptors," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 11, pp. 1624–1636, 2011.
- [9] W. Beaudot and K. Mullen, "How long range is contour integration in human color vision?" *Visual Neuroscience*, vol. 15, pp. 51–65, 2003.
- [10] P. Bhat, C. L. Zitnick, M. Cohen, and B. Curless, "Gradientshop: A gradient-domain optimization framework for image and video filtering," *ACM Transactions on Graphics*, vol. 29, no. 2, 10:1–10:14, 2010.
- [11] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp. 567–585, 1989.
- [12] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: Interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.
- [13] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," *Proceedings of the Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 721–728, 2003.
- [14] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pp. 341–346, 2001.
- [15] D. Cohen-Or, O. Sorkine, R. Gal, T. Leyvand, and Y.-Q. Xu, "Color harmonization," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 624–630, 2006.
- [16] M. Tokumaru, N. Muranaka, and S. Imanishi, "Color design support system considering color harmony," *IEEE International Conference on Fuzzy Systems*, vol. 1, pp. 378–383, 2002.
- [17] C. Ware, *Information Visualization: Perception for Design*, second. Morgan Kaufmann Publishers, 2004.
- [18] W. Javed, B. McDonnell, and N. Elmqvist, "Graphical perception of multiple time series," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 927–934, 2010.
- [19] S. Bateman, R. L. Mandryk, C. Gutwin, A. Genest, D. McDine, and C. Brooks, "Useful junk?: The

effects of visual embellishment on comprehension and memorability of charts,” *Proceedings of the Conference on Human Factors in Computing Systems*, pp. 2573–2582, 2010.

- [20] J. Rodgers and L. Bartram, “Exploring ambient and artistic visualization for residential energy use feedback,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2489–2497, 2011.

SIDEBAR: AESTHETICS IN INFORMATION VISUALIZATION

There has been extensive research on casual and traditional information visualization improving aesthetics. Casual information visualization is the use of computer-generated visual shapes, metaphors, or images to represent data in everyday life. It is targeted at diverse users from novices to visual experts and from everyday personal information to work related information (see also [1]). There have been several attempts to improve the aesthetic aspects of the visualizations in casual information visualizations. For example, Skog et al. [2] have created an ambient information visualization system for casual data such as weather and bus departure times. Their system uses the existing Mondrian’s painting style to map data onto colors and onto the size of squares in the image. They evaluated their visualization in a public space at a university, and some participants could get information. Rodgers and Bartram [3] have designed an ambient and artistic visualization for residential energy consumption feedback. They exploited simple patterns, and the population, size, and/or motion indicate current power and/or total energy use. They tested their designs in two scenarios such as kitchen and home theater. The results contained different visualization preferences among subjects (artistic visualization vs. traditional visualization) and the desire for customization. The InfoCanvas [4] has brought many pieces of information such as weather, time, e-mail, etc. into a single artistic scene. A color, shape, size, or position of objects in the scene depict each data value. Viegas et al. [5] have presented Artifacts of the Presence Era, which uses video and audio from a museum and visualizes activities in the museum through a geological metaphor. Their observation of museum visitors was that most visitors enjoyed and understood the visualization.

In traditional information visualization, there are mechanisms to improve aesthetics by adding visual embellishments into traditional charts such as line and bar charts. Pictograms, photos, and/or other visual elements can be used as visual embellishments to decorate charts, referred to as “chartjunk”. However, there is a debate over the usefulness of “chartjunk”. Tufte [6] has suggested that charts should use minimal ink to show data such that the visualization should not include any superfluous graphical elements and be void of any distracting element. Gillan et al. [7] have conducted two experiments to verify a minimal chart based on Tufte’s rules. The result shows that a minimal chart following Tufte’s idea was better

than a traditional 2D bar chart and a 3D bar chart with a background image in terms of response time. Ghani and Elmquist [8] have used a photo as a texture for a node-link diagram, where the photo did not match the diagram. The experimental result showed that there was no effect of a photo on accuracy, compared to color encoding as a texture.

In contrast to these studies, recently there were a number of studies that evaluated the effectiveness of visual embellishments on memorability and comprehension. Bateman et al. [9] have presented the results of their experiment to validate differences between plain charts and charts with visual embellishments by a designer, Nigel Holmes. They concluded that people might interpret an embellished chart as accurately as a plain chart and an embellished chart may be better than a plain chart in terms of memorability. Borkin et al. [10] conducted an experiment on Amazon Mechanical Turk to see the effect of various visualization attributes on memorability. Their result showed that visual embellishments were correlated to memorability such that including pictograms helped people to remember an image of a visualization.

References

- [1] Z. Pousman, J. T. Stasko, and M. Mateas, “Casual information visualization: Depictions of data in everyday life,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1145–1152, 2007.
- [2] T. Skog, S. Ljungblad, and L. E. Holmquist, “Between aesthetics and utility: Designing ambient information visualizations,” *IEEE Symposium on Information Visualization*, pp. 233–240, 2003.
- [3] J. Rodgers and L. Bartram, “Exploring ambient and artistic visualization for residential energy use feedback,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2489–2497, 2011.
- [4] C. Plaue, T. Miller, and J. Stasko, “Is a picture worth a thousand words?: An evaluation of information awareness displays,” *Proceedings of the Conference on Graphics Interface*, pp. 117–126, 2004.
- [5] F. Viegas, E. Perry, E. Howe, and J. Donath, “Artifacts of the presence era: Using information visualization to create an evocative souvenir,” *IEEE Symposium on Information Visualization*, pp. 105–111, 2004.
- [6] E. Tufte, *The Visual Display of Quantitative Information*, Second edition. Graphics Press, 2001.
- [7] D. J. Gillan, E. Richman, and M. Neary, “Minimalism in graphics,” *Posters and Short Talks of the Conference on Human Factors in Computing Systems*, pp. 75–76, 1992.
- [8] S. Ghani and N. Elmquist, “Improving revisitation in graphs through static spatial features,” *Proceedings of the Conference on Graphics Interface*, pp. 175–182, 2011.

- [9] S. Bateman, R. L. Mandryk, C. Gutwin, A. Genest, D. McDine, and C. Brooks, “Useful junk?: The effects of visual embellishment on comprehension and memorability of charts,” *Proceedings of the Conference on Human Factors in Computing Systems*, pp. 2573–2582, 2010.
- [10] M. A. Borkin, A. A. Vo, Z. Bylinskii, P. Isola, S. Sunkavalli, A. Oliva, and H. Pfister, “What makes a visualization memorable?” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2306–2315, 2013.

Ji Hwan Park received his BS degree in computer engineering from Hongik University, Seoul, South Korea, in 2004. Currently, he is a PhD candidate in the Department of Computer Science at Stony Brook University. His research interests include information visualization, visual analytics, scientific visualization, and human computer interaction. Contact him at jihwpark@cs.stonybrook.edu.

Arie Kaufman is a Distinguished Professor and Chair of Computer Science, Director of the Center for Visual Computing (CVC), and Chief Scientist of the Center of Excellence in Wireless and Information Technology (CEWIT) at Stony Brook University. He received his PhD in Computer Science from Ben-Gurion University, Israel (1977). He is internationally recognized for his pioneering and seminal contributions to visualization, graphics, virtual reality, and their applications, especially in biomedicine. He is Fellow of IEEE, Fellow of ACM, member of European Academy of Sciences, recipient of IEEE Visualization Career Award, and was inducted into LI Technology Hall of Fame. He was the founding Editor-in-Chief of IEEE Transaction on Visualization and Computer Graphics (TVCG), 1995-1998. Contact him at ari@cs.stonybrook.edu.

Klaus Mueller received the PhD degree in computer science from The Ohio State University. He is currently a professor of computer science at Stony Brook University and an adjunct senior scientist at Brookhaven National Lab. His current research interests include visualization, visual analytics, data science, and medical imaging. He won the US National Science Foundation (NSF) Early CAREER award in 2001 and the SUNY Chancellors Award in 2011. He has authored more than 160 papers which were cited more than 7,000 times. He currently is Associate Editor-in-Chief at IEEE Transactions on Visualization and Computer Graphics and a senior member of the IEEE. For more information, see <http://www.cs.sunysb.edu/~mueller>. Contact him at mueller@cs.stonybrook.edu.