

Perceptual Guidelines for Creating Rectangular Treemaps

Category: Research

Abstract—Treemaps are space-filling visualizations that make efficient use of limited display space to depict large amounts of hierarchical data. Creating perceptually effective treemaps requires carefully managing a number of design parameters including the aspect ratio and luminance of rectangles. Moreover, treemaps encode values using area, which is known to be less accurate than judgments of other visual encodings, such as length. We conduct a series of controlled experiments aimed at producing a set of design guidelines for creating effective rectangular treemaps. We find that while luminance does not affect area judgments, aspect ratio does have an effect. Specifically, we find that the accuracy of area comparisons suffers when the compared rectangles have extreme aspect ratios or when both are squares. Contrary to common assumptions, the optimal distribution of rectangle aspect ratios within a treemap should include non-squares, but should avoid extremes. We then compare treemaps with hierarchical bar chart displays and identify the data densities at which length-encoded bar charts become less effective than area-encoded treemaps. We report the transition points at which treemaps exhibit judgment accuracy on par with bar charts for both leaf and non-leaf tree nodes. We also find that even at relatively low data densities treemaps result in faster comparisons than bar charts. Based on these results, we present a set of guidelines for the effective use of treemaps and suggest alternate approaches for treemap layout.

Index Terms—Graphical Perception, Information Visualization, Treemaps, Visual Encoding, Experiment, Mechanical Turk.

1 INTRODUCTION

Treemaps have become increasingly popular for displaying large hierarchical datasets [27]. Though hand-drawn visualizations using area partitioning have existed for centuries [15], Shneiderman and colleagues [26] were the first to define computational techniques for generating rectangular treemaps that make efficient use of display space by recursively subdividing area. Each rectangle in a treemap represents a node in a tree with area proportional to the value of the node. To encode hierarchy, parent node rectangles enclose child rectangles. This space-filling approach produces a layout in which node values remain comprehensible at much higher data densities than in a node-link representation. Visualization designers have used treemaps to visualize a variety of data types, including presidential election data [1], popular internet sites [2], the stock market [39], and Google News [41].

However, creating perceptually effective treemaps requires carefully managing a number of design parameters including the *aspect ratio* of rectangles (controlled indirectly via choice of layout algorithm), the *luminance* of rectangles (frequently used to encode additional quantitative variables) and the *thickness of borders* between rectangles (used to encode hierarchy). Poor choices of these parameters can obscure the data, making it difficult to extract information from the treemap (see Figure 1).

Even when these parameters are well chosen, treemaps can be difficult to read accurately because they encode values using area. Prior studies have shown that people perceptually underestimate area in contrast to other visual encodings such as length [12, 34]. Thus, a bar chart that uses length encodings may be a more effective display for quantitative data than a treemap. However, bar charts are inherently less space-efficient than treemaps because they must include white space around each bar; in contrast, treemaps maximize the number of data-representative pixels. Treemaps also directly convey the hierarchical structure of the tree using containment.

Despite the popularity of treemaps, there is little guidance on how to choose the design and layout parameters or manage the tradeoff between space-efficiency and readability. Prior work on treemap design has primarily focused on developing new layout algorithms that provide control over aspect ratio [6, 28, 9, 26], shading [37], and border design [23]. Researchers have also extended the recursive area subdivision approach beyond rectangles to Voronoi regions [4], circles [42] and jigsaw shapes [40]. Prior studies of treemap effectiveness have focused on treemap systems that enable interactive analysis of data [3, 21, 31, 36] and did not explicitly examine design parameters. Choosing the most effective design and layout parameters is largely left to the intuition of the visualization designer or constrained by the available layout algorithm(s).

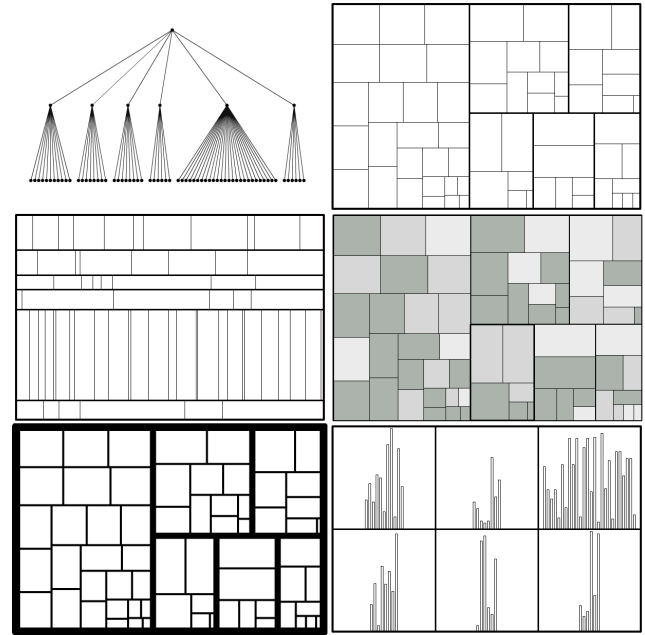


Fig. 1. A node-link tree with corresponding treemap and bar chart representations. Treemap design parameters that can affect perception of rectangle area include the aspect ratios of rectangles (top right, middle left) the luminance of rectangles (middle right) and border thickness (bottom left). Bar charts are an alternative encoding of the leaf node tree data that use length rather than area. At lower data densities such bar charts can be easier to read, but as the amount of data increases bar charts become less effective because they are not as space-efficient as treemaps. In addition, bar charts do not directly encode the hierarchical structure of a tree.

In this paper we conduct a series of controlled experiments aimed at producing a set of design guidelines for creating effective rectangular treemaps. Specifically, we study how luminance and aspect ratio of rectangles affect the perception of area. Our studies suggest that area comparisons are perceptually independent of luminance. We find no evidence that the luminance of rectangles impacts the way people judge area. However, aspect ratio does affect area judgments. We find that the accuracy of area comparisons suffers when both rectangles have extreme aspect ratios or when both rectangles are squares. Contrary to common assumptions, the optimal distribution of rectangle

aspect ratios within a treemap should include non-squares but should avoid extremes. We then compare treemaps with hierarchical bar chart displays and identify the data densities at which length-encoded bar charts become less effective than area-encoded treemaps. We identify the transition points at which treemaps exhibit comparison judgments with accuracy on par with bar charts for both leaf and non-leaf tree nodes. We also find that even at relatively low data densities, treemaps support faster comparisons than bar charts. Based on these results, we conclude with a set of guidelines for the effective use of treemaps and suggest alternate approaches for treemap layout.

2 RELATED WORK

We review related work in three overlapping areas: general studies of graphical perception, experiments assessing the effects of data-density, and evaluations of treemap visualizations.

2.1 Graphical Perception

A wealth of prior research has investigated how visual encodings such as length, area, color, and texture affect graph comprehension. Bertin [8] was among the first to consider this issue, claiming that “*resemblance, order, and proportion are the three signifieds in graphics.*” At a basic level visualizations help us identify like and non-like elements, perceive rank-order relations, and compare quantities. Both theoretical and empirical evaluations assess how different visual encoding techniques facilitate these basic judgments.

Based on his experience as a cartographer, Bertin proposed an ordering of visual encodings for three common types of data: nominal, ordinal, and quantitative. He wrote that spatial encodings are superior to other encodings, and that hue effectively encodes nominal (categorical) data but not quantitative data. Cleveland & McGill [12] extended Bertin’s work by applying results from psychology to provide empirical grounding for the order of visual encoding. Their human-subjects experiments established a significant accuracy advantage for position judgments over both length and angle judgments.

S. S. Stevens [33] modeled the mapping between the physical intensity of a stimulus (e.g., a shape’s length or area) and its perceived intensity as a power law: $P = kI^\alpha$, where P is the perceived intensity, I is the physical intensity, k is an empirically determined scale constant, and α is the power law exponent. If $\alpha > 1$, perception tends toward overestimation: e.g., doubling the weight of an object makes it feel more than twice as heavy. If $\alpha < 1$, perception tends towards underestimation: e.g., an object twice as large may not look so. If $\alpha = 1$, there is no systematic bias to the perceived intensity.

Stevens found that the exponents for length and area judgments were roughly $\alpha \approx 1$ and $\alpha \approx 0.7$, respectively. Thus subjects perceived length with minimal bias, but underestimated differences in area. Based on this finding, Cleveland & McGill [12] argued that length should be preferred to area when encoding quantitative variables. Cleveland et al. [11] replicated the finding that, on average, people underestimate area; however, they also found wide variation in per-subject exponents, suggesting that *perceptual rescaling* of area encodings is unlikely to provide benefits. While often reporting different exponent values, additional research on this subject continues to find that people in general underestimate area [30, 34].

More recently, Heer & Bostock [17] investigated how the shape of an area affects judgment accuracy. They found that area comparisons among circles and rectangles have similar judgment accuracy, and that both are less accurate than length judgments, which in turn are less accurate than position judgments. They also found that when comparing rectangles with aspect ratios drawn from the set $\{\frac{2}{3}, 1, \frac{3}{2}\}$, a comparison of two squares is the least accurate. In this paper, we extend this analysis to a greater diversity of aspect ratios.

2.2 Data Density

Researchers (e.g., [19, 26, 32]) often promote visualization techniques such as treemaps for their “space-filling” properties. Similarly, Tufte [35] advised designers to maximize data density: the number of data marks per chart area. However, only a few studies have characterized the effectiveness of various data-dense displays. Cleveland

et al. [10] investigated scale effects on correlation perception in scatterplots by varying axis ranges while holding display size constant. Woodruff et al. [44] presented methods for promoting constant data density in semantic zooming applications, but did not present an empirical evaluation. Lam et al. [22] studied the effects of low and high resolution time-series displays on visual comparison and search tasks. Their low- and high-res displays used different visual encoding variables (color vs. position), confounding data density and visual encoding type. Finally, Heer et al. [18] evaluated a form of data-dense time-series display known as *horizon graphs* [14, 25]. They steadily decreased chart sizes until they identified the points at which horizon graphs outperformed standard line charts. In this paper, we characterize a similar trade-off between treemap and bar chart displays under conditions of increasing data density.

2.3 Treemap Evaluation

Shneiderman [26] introduced treemaps as a way to visualize large hierarchical data sets (or colloquially, “trees”). Since then, a number of researchers have compared treemaps to other tree visualization techniques [3, 21, 31, 36]. These studies found that when treemaps were used in a hierarchy exploration tool, they performed as well or better than other common tools, such as directory browsers. However, these experiments considered fully interactive systems that included search and filtering capabilities, making it difficult to separate the effect of the interaction widgets from the graphical perception of the charts.

Other studies have investigated static treemap displays. Barlow & Neville [5] compared static representations of decision trees, including treemaps, to one another. They found that treemaps perform poorly when used to compare two values or examine the structure of the tree. One factor that may have contributed to these difficulties is that the treemaps they used included large borders between levels. While users could click the borders to select both leaf and non-leaf nodes, the thickness of the borders further reduced the effectiveness of the area encoding. Their study demonstrated a tradeoff between enabling interaction and improving graphical perception that resulted from changing a design parameter.

Bederson et al. [6] proposed ordered treemaps, introducing a collection of layout algorithms that preserve the ordering of data. They then compared ordered treemaps to other treemap algorithms using metrics such as average aspect ratio. For three different trees drawn from a log-normal distribution, they found squarified treemaps had the lowest average aspect ratio (between 1.19 and 1.75) and that slice-and-dice treemaps had the highest (between 26.10 and 304). For stock market data, squarified treemaps had an aspect ratio of 3.21, whereas slice-and-dice treemaps had an aspect ratio of 369.83. They did not investigate how aspect ratio affects area estimation, a primary goal of the current work.

Ziemkiewicz & Kosara [45] conducted an experiment involving a number of hierarchy visualization techniques and found that the metaphor implied by a task prompt (e.g., using the word “under” vs. “contained in”) affects estimation time for structural tasks. While we do not examine structural questions in this work, we do incorporate value comparisons between nodes at different levels of a hierarchy.

3 RESEARCH GOALS AND METHODS

Our objective is to characterize the effects of treemap design choices on graphical perception. To this end, we run a series of controlled experiments investigating the effects of luminance, aspect ratio, and data density on value estimation tasks. These properties are commonly varied in treemaps, and our experiments are designed to determine optimized settings for each one.

Each of our studies employs a similar experimental design: we highlight two shapes in a display and ask subjects to judge the percentage the smaller is of the larger. Throughout the paper, we use the term *true percentage* to denote the actual (physical) percentage the smaller element is of the larger. To analyze results, we use a measure of log absolute error: $\log_2(|\text{judged percent} - \text{true percent}| + 1)$. We chose this measure to facilitate comparison with prior work in graphical perception [12, 17, 29, 30, 43] which also uses this measure.

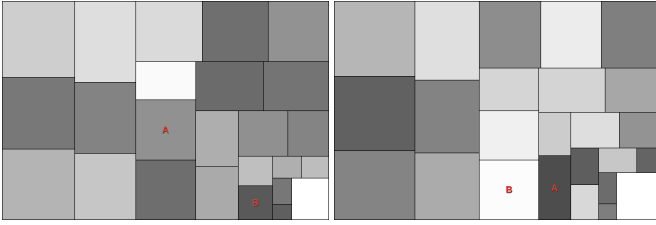


Fig. 2. Two example stimuli from Pilot 1 with randomized area and luminance values. We generated the stimuli using the squarified treemap algorithm [9]. We randomly marked two rectangles A and B in each stimulus image and asked subjects to identify the smaller rectangle and estimate what percentage the smaller was of the larger.

We conduct each experiment using Amazon’s Mechanical Turk (MTurk), a popular crowdsourcing platform. On MTurk, *requesters* post small tasks, called *Human Intelligence Tasks* (or *HITs*), that *workers* perform for a small *reward*, typically a few cents per task. Each HIT has a set number of *assignments*, which is the maximum number of workers who may perform the task. An individual worker can only perform one assignment per HIT. HITs may require one or more *qualifications*, such as a HIT acceptance rate above 95%. Requesters may also create their own qualifications, which may serve as instructions and also screen out workers who do not understand the task.

Though the use of MTurk for experimental studies is relatively new, it has quickly gained popularity as a low-cost means of recruiting subjects and conducting experiments. For example, to evaluate rendering techniques Cole et al. [13] have collected 275,000 judgments of 3D surface orientation from 550 Turkers. Other researchers have explicitly evaluated MTurk as an experimental platform. Kittur et al. [20] used a Wikipedia article rating task to evaluate different facets of crowdsourced experiments. They recommended using verifiable questions and crafting questions such that the effort required for an honest response matches that of a malicious response. Heer & Bostock [17] assessed MTurk specifically as a tool for graphical perception experiments. Because MTurk provides no control over subjects’ display configurations (e.g. display resolution, physical display size, and environmental lighting), it is not immediately clear that crowdsourced experiments can produce reliable results. However, Heer & Bostock showed that MTurk results are reliable by replicating prior laboratory studies, including the work of Cleveland & McGill [12]. They also suggested techniques to improve experimental reliability, such as the use of well-designed qualification tasks.

4 PILOT STUDY: TRUE PERCENTAGE AND LUMINANCE

Before investigating the effects of aspect ratio or data density, we first conducted a pilot study that tested the effects of true percentage (the percentage the smaller is of the larger) and luminance on area judgments. Prior studies [12, 17] have demonstrated an effect of the true percentage difference on the accuracy of proportion judgments: viewers are more accurate when the true percentage is either small (5%) or large (95%), with maximal error around 60%. We sought to replicate this result and assess the effect of true percentage so that we could control it appropriately in our subsequent studies. Moreover, treemap designers commonly use the luminance of a treemap cell to encode an additional quantitative variable. Our pilot also verifies that such luminance differences do not interfere with area judgments.

We showed subjects 600×400 pixel squarified treemap displays visualizing 24 uniform random values. Figure 2 shows two example stimuli. In each case we labeled two rectangles and marked them A and B. We asked subjects to identify which rectangle was smaller and what percentage the smaller was of the larger. In each trial we varied the luminance (L^*) of each cell randomly between 35 and 100 in CIE $L^*a^*b^*$ color space according to a uniform distribution.

We conducted the pilot on MTurk as 100 distinct HITs, each with 24 assignments and a reward of \$0.03. A total of 41 subjects provided 2400 responses; we removed 121 outlier responses (5%) with

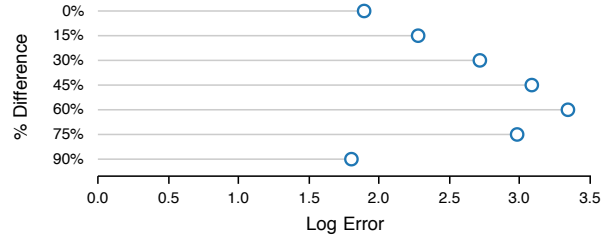


Fig. 3. Average area judgment errors in Pilot 1, binned by true percentage. Errors peak at 60% and diminish at the extremes of the scale.

an absolute error greater than 35%. As true percentages and luminance were assigned randomly across trials, we analyzed responses by applying Analysis of Covariance (ANCOVA) to the log absolute error of subjects’ proportion estimates, including both true percentage and luminance difference as covariates in the model.

As shown in Figure 3, our pilot study results exhibit an error profile by true percentage similar to that of Heer & Bostock [17]. Our analysis finds that the true percentage has a strong, statistically significant effect on judgment accuracy ($F(1,2252) = 253.90, p < 0.001$). These results show that the true percentage must be carefully controlled across experimental conditions. If it is not controlled, the effect of true percentage on judgment accuracy may obscure the effects of other independent variables and thereby confound the experiment.

We also found a high prevalence of responses that were a multiple of 5 (not visible in Fig. 3 due to binning). As a result, trials for which the true difference is also a multiple of 5 correspondingly exhibited less error. This reduced error is probably due to response bias – subjects may prefer specifying numbers that are factors of 5 – and not due to improved perception. As a result, experimenters may wish to select true percentages in a manner that mitigates this response bias.

Finally, we found no significant effect on judgment accuracy due to luminance, ($F(1,2252) = 0.086, p < 0.767$). These results suggest that rectangle luminance does not interfere with area judgments, in congruence with prior work finding that area and luminance are *separable* perceptual dimensions [16, 38]. The implication is that studies of area judgment may ignore interactions with luminance and still produce generalizable results.

5 EXPERIMENT 1: THE EFFECTS OF ASPECT RATIO

Our first experiment assessed the effects of aspect ratio on rectangular area judgments. A common criticism of the original “slice-and-dice” treemap algorithm [26] is that it produces rectangles with a wide distribution of aspect ratios (e.g., Figure 1); in trees with high branching factors it can produce aspect ratios with magnitudes of 4 and higher. Such extreme ratios may complicate area comparisons. In response, researchers developed “squarified” treemap algorithms [9, 39] that attempt to optimize rectangle aspect ratios to squares. Bruls et al. [9] posited multiple benefits for squarified layouts:

- Squares minimize rectangular perimeter, reducing border ink.
- Squares are easier to select with a mouse cursor.
- Rectangles with similar aspect ratios are easier to compare.

While the first assertion is mathematically true and the second assertion is supported by both theory and empirical evidence (e.g., Fitts’ Law [24]), the validity of the third assertion is less clear. The assumption that square aspect ratios are optimal is not rooted in empirical perception data. A recent experiment by Heer & Bostock [17] found that comparing two squares leads to significantly higher error when varying aspect ratios among the set $\{\frac{2}{3}, 1, \frac{3}{2}\}$. The cause of this effect remains unclear.

In this experiment, we further examined the effects of aspect ratio on proportional judgments. We replicated Heer & Bostock’s study design, but incorporated more extreme aspect ratios and also assessed the effects of orientation (rotation) on judgment accuracy. Based on prior results, we hypothesized that both extreme aspect ratios and squares

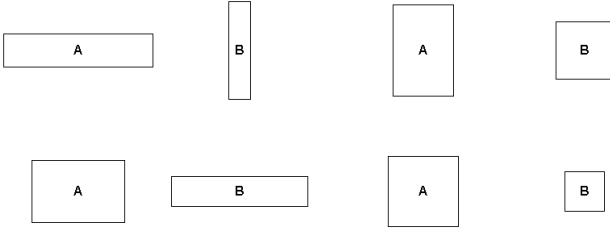


Fig. 4. Example stimuli from the aspect ratio study. Rectangles varied in terms of both percentage difference and aspect ratio.

would hamper judgment accuracy. We also hypothesized that judgments of rectangles with different primary orientations (horizontal or vertical) would result in increased error. This hypothesis is based in part on prior perceptual research [7] finding that mental rotation is more cognitively demanding than either translation or scaling.

5.1 Method

We asked subjects to compare rectangular areas of varying size and aspect ratio. We showed subjects a 600×400 pixel image containing two center-aligned rectangles and instructed them to identify which of the rectangles (A or B) was the smaller and then estimate the percentage the smaller was of the larger by making a “quick visual judgment.” The stimulus images consisted of two rectangles (Fig. 4) – not a full treemap display. Heer & Bostock [17] tested both stand-alone rectangles and rectangles within a treemap while varying aspect ratio. They found no significant accuracy differences between these two stimulus types, so we believe our results generalize to treemap displays.

We controlled both the true percentage between rectangles and their aspect ratios. True percentages varied over 32%, 48%, 58%, 72%. To reduce accuracy differences due to response bias, these values were explicitly placed at equal, symmetric distances from the nearest multiple of 5. We chose the aspect ratios for each pair of rectangles from the set $\{\frac{2}{3}, \frac{2}{3}, 1, \frac{3}{2}, \frac{3}{2}\}$ crossed with itself. This set of aspect ratios extends the set used by Heer & Bostock [17] with more extreme values. Since non-square aspect ratios have a matching rotated variant (e.g., a rectangle with ratio $\frac{2}{3}$ is a 90° rotation of a rectangle with ratio $\frac{3}{2}$), we included an additional replication of the 1 × 1 condition to improve statistical power. Our experiment design thus consisted of 104 unique trials (HITs): 4 (difference) × 26 (aspect ratio pairs with replication).

As a qualification task we used multiple-choice versions of two example trial stimuli. For each trial, subjects first specified which rectangle was smaller and then entered their judgment of the numerical percentage the smaller rectangle was of the larger. We focused on analyzing estimation error in this experiment. We requested a total of 104 HITs with N=25 assignments and paid a reward of \$0.03 per HIT.

5.2 Results

We collected 104×25 = 2,600 responses, from which we removed 18 outliers (0.7%) with absolute errors above 35%. To analyze the data, we used log absolute error: $\log_2(|\text{judged percent} - \text{true percent}| + 1)$. We then conducted an ANOVA with a 4×6×2 factorial design:

- (4) *True percentage*: one of 32%, 48%, 58%, or 72%.
- (6) *aspect ratio pairs*: any rotated variants are treated as the same ratio (e.g., $\frac{2}{3} \approx \frac{2}{3}$, $\frac{2}{3} \approx \frac{3}{2}$), and denoted by the greater value.
- (2) *relative orientation*: indicates if the compared rectangles have identical ($\frac{9}{2} \times \frac{9}{2}$) or different ($\frac{9}{2} \times \frac{3}{2}$) orientations.

5.2.1 True Percentage Dominates Comparison Accuracy

We again found a strong effect due to the true percentage ($F(3,2173) = 94.56$, $p < 0.001$). As shown in Figure 5, the results exhibit a similar profile as in our pilot study. Moreover, true difference produced the strongest effect in our model. This result again argues for the importance of including true difference as a controlled factor in proportional judgment studies. Applying Bonferroni-corrected post-hoc t-tests, we

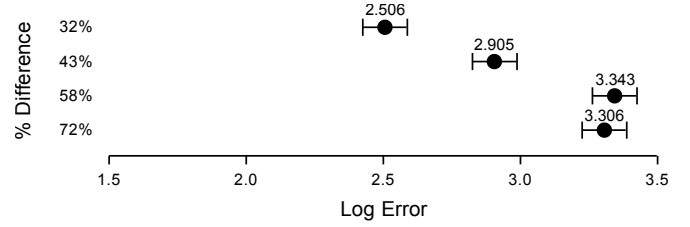


Fig. 5. Experiment 1 area judgment error by true percentage. Error bars indicate 95% confidence intervals.

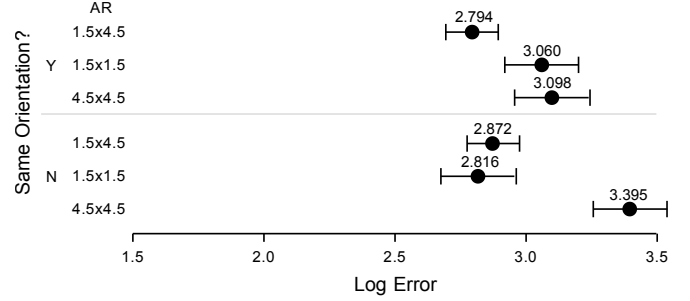


Fig. 6. Judgment error by orientation and aspect ratio. Squares omitted due to rotation invariance. Error bars indicate 95% confidence intervals.

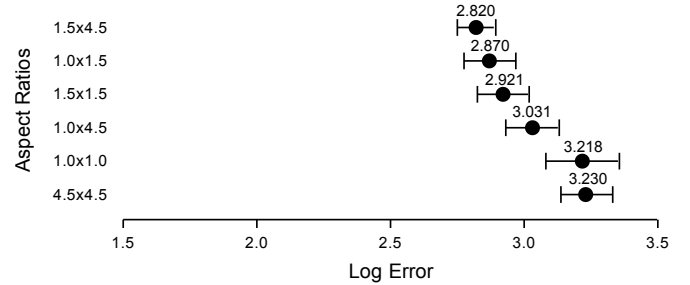


Fig. 7. Area judgment error by aspect ratio. Squares and extreme ratios have the highest error. Error bars indicate 95% confidence intervals.

found that all true percentage conditions were significantly different ($p < 0.05$) except for 58% and 72%. Finally, we found no significant interactions of true percentage with either orientation or aspect ratio.

5.2.2 Orientation Affects Extreme Aspect Ratios

Examining the effects of orientation on judgment accuracy, we found no main effect ($F(1,1490) = 0.669$, $p = 0.414$). This result implies that, on average, 90° rotation of rectangles has little to no effect. However, we did find a significant interaction effect between orientation and aspect ratio ($F(2,1490) = 7.23$, $p < 0.001$). Figure 6 shows error rate by both orientation and aspect ratio. When orientations differ, error appears to increase for comparisons involving the most extreme ratios in our study ($\frac{9}{2} \times \frac{9}{2}$). This result suggests that rotation may contribute to higher judgment errors as aspect ratios deviate further from squares (e.g., as occurs in slice-and-dice treemaps [26]).

5.2.3 Diverse Aspect Ratios Improve Accuracy

Finally, we analyzed the impact of aspect ratio on judgment accuracy, finding a significant effect ($F(5,2173) = 13.85$, $p < 0.001$). Applying post-hoc t-tests with Bonferroni correction, we found that aspect ratio pairs of $\frac{9}{2} \times \frac{9}{2}$ and 1×1 exhibited significantly higher error than the pairs $1 \times \frac{3}{2}$, $\frac{3}{2} \times \frac{3}{2}$ and $\frac{3}{2} \times \frac{9}{2}$. Similarly $1 \times \frac{9}{2}$ was significantly more error prone than $\frac{3}{2} \times \frac{9}{2}$. No other differences were significant. Figure 7 shows the resulting rank ordering by error of aspect ratio pairs and their corresponding confidence intervals. The results indicate that average judgment accuracy improves when comparing rectangles with diverse aspect ratios, even when one of the ratios is large. The highest

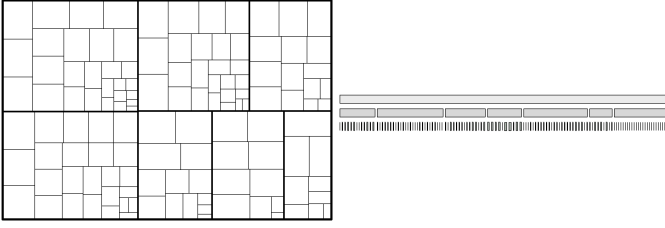


Fig. 8. Data visualized as both a (a) treemap, and (b) icicle plot. Note that it is difficult to distinguish leaf values within the icicle plot.

error occurred when comparing two extreme aspect ratios or comparing squares. The latter result replicates Heer & Bostock’s [17] finding that comparing squares leads to increased error.

5.3 Discussion

Our experiment found that graphical perception suffers when comparing extreme aspect ratios, particularly when the rectangles have different orientations. These results support the general intuition against using treemap layout algorithms that produce rectangles with extreme aspect ratios (e.g., slice-and-dice [26]). On the other hand, subjects exhibited equally poor accuracy when comparing squares. As a result, the perceptual justification for squarified treemap layout algorithms [9, 39]—that squares promote more accurate comparisons—appears to be faulty. It instead seems that squarified algorithms are effective in part because (a) they avoid extreme aspect ratios and (b) in most cases they are unable to perfectly achieve their “squarification” objective, instead producing a distribution of aspect ratios.

6 EXPERIMENT 2: THE EFFECTS OF DATA DENSITY

As the data density of a visualization increases, the marks that encode the data must either overlap or decrease in size. Past a certain point, such overlap and reduction in size makes it difficult to distinguish individual marks and read the values they encode. The point at which such difficulties occur depends on the space efficiency of the visual encoding. For example, an area encoding, as in a treemap, makes more efficient use of space than a length encoding, as in a bar chart. Therefore, we hypothesized that at high densities a treemap would provide a more effective display than a bar chart. Yet, area is less perceptually effective than length for encoding quantitative values, as people typically underestimate area [12, 17]. Thus, we hypothesized that at lower densities the bar chart would be more perceptually effective than a treemap. We designed our second experiment to determine the data density at which treemaps become more effective than bar charts for comparing quantitative values. This experiment did not examine how well people could extract the structure of the tree. Instead we chose to focus on the value comparison tasks, which we believe is the most common perceptual task performed with treemaps.

Nevertheless, one advantage of treemaps is that, unlike bar charts, they directly encode hierarchical structure via containment. Icicle plots are a type of visualization that fall between treemaps and bar charts as they directly encode hierarchy but use a length encoding for value. However, Figure 8 shows that as data density increases and the number of nodes exceeds the horizontal extent of the display, the icicle plot becomes very difficult to read. Therefore, to assess trade-offs between area and length encodings, we designed a hierarchical layout algorithm which uses bar charts to encode the values of leaf nodes.

Figure 9 shows an example of our hierarchical bar chart and a treemap, each encoding the same data. Each bar of the hierarchical bar chart represents a leaf node in the tree and the bars are grouped together into a cell based on their sibling relationships. Thus, the value of a parent node is computed as the sum of all bars within the corresponding parent cell. The hierarchical bar chart makes more efficient use of space than an icicle plot, but at the cost of increasing the cognitive load when comparing non-leaf values. We hypothesized that, regardless of data density, treemaps would be more effective than hierarchical bar charts for value comparison tasks that involve non-leaf

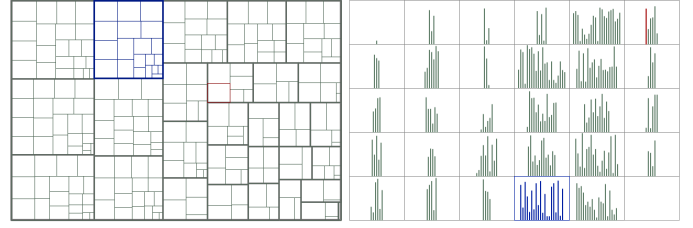


Fig. 9. Example Exp. 2 stimuli with 256 leaves. (a) Squarified treemap. (b) Hierarchical bar chart; each bar represents a leaf node and sibling bars are grouped together in a cell. The charts depict the same data.

nodes (e.g., categories) containing multiple children because of the additional cognitive load required to sum the bars.

Our hierarchical bar charts are designed to depict two-level trees where all leaves are at the second level. Many real-world data sets fit this structure, including stock values organized by sector or products organized by category. In fact, many treemap visualizations of such two-level data may be found on the web [1, 2, 39]. Since the goal of our second experiment was to examine how data density affects value judgments rather than how well people can extract structural properties of the tree, we worked with such two-level bar charts. We leave it to future work to design hierarchical bar charts that can encode trees containing more than two levels.

Our experiment included three types of comparisons: leaf to leaf (LL), leaf to non-leaf (LN), and non-leaf to non-leaf (NN). In all three cases, we presented participants with either a treemap or a hierarchical bar chart and asked them to compare two elements. With a treemap, participants were asked to compare two rectangular areas. With a hierarchical bar chart, participants were asked either to compare two bars (LL), or to compare groups of bars to one another (LN or NN). To extract the value of non-leaf nodes subjects had to aggregate the lengths of all of the bars within a cell. As a result, comparisons involving non-leaf nodes in the bar chart required a more complicated cognitive process than simply judging the length of a single bar.

6.1 Methods

For each trial, we showed participants a chart with two highlighted nodes. We asked participants to indicate which of the two nodes was smaller, and then estimate the percentage the smaller was of the larger. We tested 2 chart types (treemap and hierarchical bar chart), 5 data densities (256, 512, 1024, 2048, 4096 leaves), and 3 comparison types (LL, LN, NN). Based on the results of our first pilot study we also controlled for true percentage difference between rectangles, using 4 levels for each comparison condition. For the LL and NN comparisons we used the same differences as in Experiment 1: 32%, 48%, 58%, and 72%. As we will explain in the next subsection, the value associated with a first-level parent node was the sum of the values associated with its children nodes. As a result, first-level nodes were usually much larger in value than leaf-nodes. Thus, for the LN comparison we used the following differences: 3%, 8%, 13%, 17%.

Analysis of our results revealed bar charts to be as accurate as treemaps at higher densities, so we added three more density conditions (6000, 7000, and 8000 leaves) for the LL case. Our experiment design consisted of 432 unique trials (HITs): for the LN and NN cases, $2 \text{ (chart)} \times 5 \text{ (density)} \times 2 \text{ (LN or NN comparisons)} \times 4 \text{ (true percentages)} \times 3 \text{ (repetitions)} = 240 \text{ HITs}$, and for the LL case, $2 \text{ (chart)} \times 8 \text{ (density)} \times 4 \text{ (true percentages)} \times 3 \text{ (repetitions)} = 192 \text{ HITs}$.

Similar to Experiment 1, our qualification task contained multiple-choice versions of the trials, one for each chart type. Each chart was sized at 600×400 pixels. We implemented the experiment using Mechanical Turk’s “external question” option, allowing us to host the trials on our own server and use JavaScript to track response times (c.f., [17]). We requested a total of 432 HITs with $N=20$ assignments and paid a reward of \$0.03 per HIT.

6.1.1 Data Generation

We generated two-level trees for our experiment using a randomized process. We first added a fixed number of children to an implicit root node, depending on the data density (i.e., total number of leaves). For example, the root node had 12 children in the 256 leaf condition (3 rows x 4 columns), 56 children in the 512 leaf condition (7 x 8), and 30 children in the 1024 leaf condition (10 x 3). Each first-level child of the root defines a cell in our hierarchical bar chart and we fixed the number of first-level nodes in order to control the number of cells in our hierarchical bar charts. To each of these first-level nodes we then randomly added between 2 and 16 children representing leaf nodes. Finally we added additional leaf nodes to randomly selected first-level nodes until we achieved the required data density. Trees created with this method have two levels below the root node and all the leaves occur at the second level. Figure 10 shows an example tree and its equivalent hierarchical bar chart.

We assigned each leaf node a random value between 5 and 100. We set the value of each first-level node to the sum of its children. We then chose two stimulus nodes at random, and adjusted their values to ensure that their difference was set to the true percentage difference for that trial. If one of the stimulus nodes was a first-level node, we distributed the change in its value to all of its children, while ensuring that the value of every child was greater than zero after the adjustment. We then rendered both a treemap and hierarchical bar chart, ensuring that subjects see exactly the same data in both chart conditions.

6.1.2 Stimulus Design

Figure 9 shows example trial stimuli for each chart type. We created treemaps using Bruls et al.'s [9] squarified treemap layout algorithm. We used both the thickness and luminance of cell borders to encode tree depth: the border is light and 2 pixels wide for nodes just below the root, and dark and 1 pixel wide for nodes two levels below the root.

Our design goals for the hierarchical bar chart layout were to: 1) use space as efficiently as possible, and 2) reveal some of the hierarchical structure of the data, but to encode all leaf nodes using length. Figure 10 shows an example tree visualized as a hierarchical bar chart. Each cell of the bar chart represents a first-level node. Each bar inside a cell represents a second level, leaf node. In this example, the root has four children, which produces four cells.

We used a regular grid to layout the cells of the hierarchical bar chart in order to aid visual comparison of bars across cells. In choosing the number of rows and columns in the grid, we sought to minimize the aspect ratio of each cell while still allowing for enough space to layout all the bars. We first computed the maximum number of columns by computing the minimum cell width, which is equal to the width of the widest bar chart (i.e., the first-level node with the largest number of children), assuming bars were 1-pixel wide and spaced by 1-pixel gaps. We then divided the total width of the display area by the minimum cell width to obtain the maximum number of possible columns. Next we calculated the number of rows that minimized the aspect ratio of a cell (making each cell as square as possible) given the maximum number of columns. Although this algorithm could result in unused cells, we believe that minimizing cell aspect ratios in this way better facilitates perception than purely maximizing space efficiency.

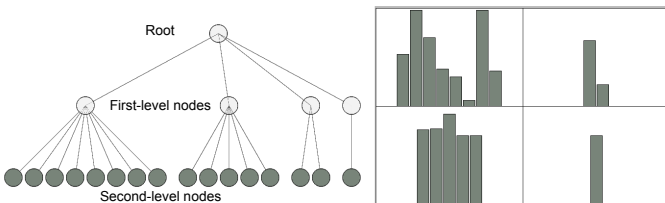


Fig. 10. A sample tree shown as a (a) node-link tree, and (b) hierarchical bar chart. Each group of sibling leaves form a cell of bar charts.

6.2 Results

We collected $432 \times 20 = 8,640$ responses, from which we removed 389 outliers (4.5%) with absolute errors above 70% or estimation times greater than 60 seconds. We were more conservative in eliminating outliers in this experiment than in the first experiment because this task was more difficult and produced greater variability. As in Experiment 1, we analyzed log absolute errors. Analyzing the data we found that the different comparison types (LL, LN and NN) exhibited different distributions, and so we analyzed each separately. For each comparison type (LL, NN, LN), we conducted a MANOVA of both response time and log absolute error using a $2 \times \{5, 8\} \times 4$ factorial design:

- (2) *Chart type*: Treemap or bar chart.
- (5 or 8) *Data density*: one of 256, 512, 1024, 2048, or 4096 leaf nodes (LN, NN), or one of 256, 512, 1024, 2048, 4096, 6000, 7000, or 8000 leaf nodes (LL).
- (4) *True percentage*: one of 32%, 48%, 58%, 72% (LL, NN), or one of 3%, 8%, 13%, or 17% (LN).

Figure 11 summarizes the estimation error and estimation time data we collected across the entire experiment. Note that in the following subsections we omit discussion of the effect of true percentages, as these results match those of our prior studies.

6.2.1 Leaf-Leaf Comparison: Treemaps Excel at High Density

The first row of Figure 11 shows estimation accuracy and time for leaf-leaf (LL) comparisons by chart type. We found a significant effect due to chart type on accuracy ($F(1,3586) = 13.998, p < 0.001$). Bar charts were more accurate than treemaps on average ($\mu_{\text{bar}} = 3.38, \mu_{\text{treemap}} = 3.65$). This result is largely due to performance differences at low data densities, as bar charts were significantly more accurate at the 512, 1024, and 2046 leaf conditions (Fig. 11). However, at higher data densities, errors equalized between bar charts and treemaps. We also found a significant main effect due to density on accuracy ($F(7,3586) = 14.233, p < 0.001$), as responses became less accurate as density increased. Finally, we found a significant interaction between chart type and density ($F(7,3586) = 2.159, p = 0.035$), as bar chart accuracy degraded more rapidly than treemap accuracy as density increased.

We found a main effect of chart type on estimation times ($F(1,3586) = 17.949, p < 0.001$), and an interaction between density and chart type ($F(7,3586) = 7.323, p < 0.001$). Treemaps and bar charts were equally fast up to 2048 leaves, but treemaps became significantly faster at the higher densities. The difference was 5 seconds in the 8000 leaf condition. Moreover, treemap accuracy was equivalent to bar chart accuracy beyond 4096 leaves.

6.2.2 Treemaps More Accurate For Non-Leaf Nodes

The second and third rows of Figure 11 show error rates by chart type and data density for the LN and NN comparisons, respectively. In both cases, we found a strong main effect of chart type on accuracy ($F(1,2229) = 21.189, p < 0.001$ for NN, $F(1,2281) = 68.535, p < 0.001$ for LN). In the NN condition, the mean log error for treemaps was 3.24, compared to 3.62 for the hierarchical bar charts. In the LN condition, the mean log error of treemaps was 2.14 compared to 2.61 for the bar charts. We also found a significant interaction between chart type and density in the LN comparison task ($F(4,2281) = 10.837, p < 0.001$), but not the NN comparison. As shown in the second and third rows of Figure 11, treemaps maintained their accuracy as data density increased, while bar charts trended towards higher error rates. Treemaps were more accurate at all densities in NN comparisons, and outperformed bar charts beyond 1024 leaves in LN comparisons.

We did not find a significant effect of chart type on estimation time for either NN or LN comparisons. We also did not find an interaction effect between chart type and density for response times in either non-leaf comparison condition.

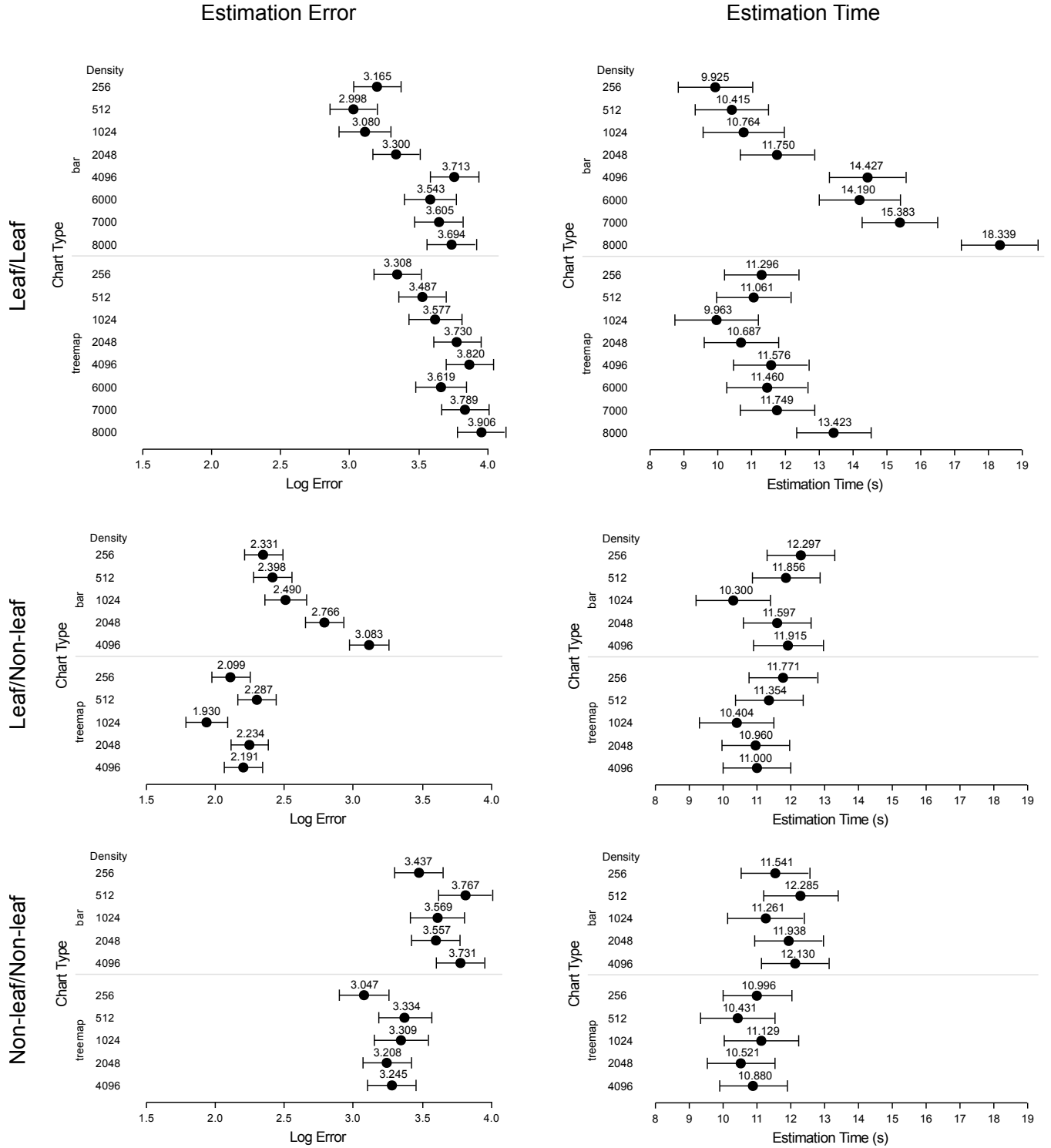


Fig. 11. Estimation time and error for each node comparison type. (a) Leaf/leaf (LL) comparisons (*first row*). Bar charts are more accurate than treemaps at densities up to 2048 leaves, after which treemaps become equally accurate. At 4096 leaves, treemaps become faster than bar charts, up to almost 5 seconds faster in the 8000 leaf condition. (b) Leaf/non-leaf (LN) comparisons (*second row*). Treemaps are more accurate than bar charts at all densities, but no faster than bar charts. (c) Non-leaf/non-leaf (NN) comparisons (*third row*). As in LN comparisons, treemaps are more accurate than bar charts at all densities, but exhibit similar estimation times.

6.3 Discussion

The results support our hypothesis that treemaps are more accurate for comparisons of non-leaf nodes. Of course, this finding is unsurprising, as estimating the value of non-leaf nodes in the bar chart display involves the cognitive overhead of combining bars. More surprisingly, treemaps were not significantly faster than bar charts in either NN or LN comparisons. We expected treemaps to be faster in these cases as

they do not have the cognitive overhead of the bar charts. It is possible that participants may have made a quick guess rather than trying to add up the bars, trading accuracy for speed given the difficult task.

Our results also demonstrate that when comparing leaf nodes the effectiveness of bar charts versus treemaps is modulated by the data density. At low data densities, bar charts result in significantly more accurate estimations without a significant difference in response time.

As data density increases, the accuracy difference equalizes, with treemaps matching bar charts at data densities past 4096 leaves.

Treemaps result in significantly better estimation times, particularly at higher densities: at 8000 leaves treemaps are almost 5 seconds faster. One possible reason for this extreme difference is visual search time for the highlighted nodes; at high densities, individual bars in the bar chart display may be small and difficult to find.

7 DESIGN GUIDELINES

Based on our experimental results, we offer the following guidelines for the design and use of treemaps in a perceptually effective manner.

Use Treemap Layouts that Avoid Extreme Aspect Ratios

Our results show that graphical perception suffers when comparing rectangles with extreme aspect ratios ($\frac{9}{2} \times \frac{9}{2}$) and squares. In addition, we found that diverse orientations adversely affected the perception of rectangles with extreme aspect ratios. Based on these two findings we advise – in keeping with common wisdom – that squarified treemap layouts should be preferred to slice-and-dice layouts. Although we found that comparison of squares also reduces judgment accuracy, the inability of squarified layouts to achieve their optimization objective appears to work in their favor. However, these findings suggest alternative approaches that may improve accuracy. For example, future work might assess judgment accuracy in a treemap layout that optimizes towards a $\frac{3}{2}$ aspect ratio.

Use Bar Charts at Low Density, Treemaps at High Density

Bar charts resulted in significantly lower error when comparing leaf nodes at low densities. If a data set has only a few hundred elements, bar charts are more effective than treemaps. At higher densities treemaps perform as well as, or better than, bar charts in both accuracy and time. As data density increases, treemaps become faster than bar charts while exhibiting equivalent accuracy. The transition point we found was at 4096 leaves, where treemaps were almost 3 seconds faster. At 8000 leaves, treemaps were almost 5 seconds faster. If a data set has thousands of elements, treemaps are more effective.

Use Treemaps When Comparing Non-Leaf Nodes

We found that treemaps were more accurate than bar charts when comparing leaf nodes to non-leaf nodes (Figure 11b) and comparing two non-leaf nodes (Figure 11c). Therefore, we advise using treemaps in cases requiring comparisons among non-leaf nodes.

Use Luminance To Encode Secondary Values In Treemaps

We found that area perception was perceptually separable from luminance. This result suggests that designers can use luminance to encode an additional variable without affecting judgment accuracy or estimation time. We leave testing the converse – whether differing areas bias luminance judgments – to future work.

Our guidelines are based on studies that used fixed-size 600×400 pixel stimulus images. Because we used MTurk as our experimental platform and cannot control the physical screen size of our subjects, we are measuring data density as the *number of marks per pixel area*. Future work might characterize area judgments in terms of physical measures such as number of marks per cm² or per optical steradian.

8 LIMITATIONS AND FUTURE WORK

While we designed our studies to provide generalizable insights into treemap design, inevitably our experiments have limitations.

In Experiment 1 (the aspect ratio experiment), we tested five aspect ratios, ranging from $\frac{2}{9}$ to $\frac{9}{2}$. However, some treemap layout algorithms, such as the slice-and-dice algorithm, can produce rectangles with even more extreme aspect ratios. We leave it to further work to explore comparisons with these ratios. Furthermore, Experiment 1 was performed with stand-alone rectangles, out of the context of treemaps. While Heer & Bostock [17] found that placing rectangles in the context of a treemap did not produce a significant effect on their

results, it may be worth further investigating whether context affects the accuracy of comparisons of rectangles with extreme aspect ratios.

Our results suggest that squarified treemaps work well because they aim to minimize aspect ratio but are unable to squarify all marks. Furthermore, the results suggest that a treemap which optimizes for $\frac{3}{2}$ (or other non-extreme, non-unity) aspect ratios may perform better than squarified treemaps. We would like to implement this treemap layout and compare it to squarified treemaps.

In Experiment 2 (the data-density experiment), our data generation algorithm created fixed-depth trees with leaves occurring only at the second level. Although this structure is common in many real-world datasets, future studies could consider more general tree structures with deeper hierarchies and leaves at different levels of the tree. Designing a hierarchical bar chart for such general tree structures remains an open problem. Moreover, our studies did not ask users questions about the structure of the data. Instead we focused on value comparison tasks because we were primarily interested in how data density affected the perception of area or length. Extending the study to include structural questions is an open direction for future work.

We found treemaps were much faster than bar charts when comparing leaf nodes at data densities beyond 4096 leaves in our 600×400 stimulus images. One possible reason is that it can be very difficult to find the stimuli when they are small, particularly in the bar chart display. Combining this study with eye-tracking is one way to separate visual search time from estimation time.

In addition, replicating these studies in the laboratory will help ensure our crowdsourced results are externally valid [17]. Because we ran the studies using MTurk we could not control for resolution and physical screen size of the displays used by our participants. An open question is whether varying physical display size while keeping the number of elements fixed would produce results similar to those we obtained through MTurk.

We have only scratched the surface of investigating the consequences of design choices of treemaps. Other choices include the border thickness between nodes (sometimes used to place labels), and alternate layout algorithms that enforce ordering of the data [6], ensure visibility of the hierarchy [23], or use non-rectangular shapes [4, 40]. In tandem with crowdsourcing, it is now possible to quickly explore how these choices affect graphical perception.

REFERENCES

- [1] Visualization lab: Treemap for comparisons of county level 2008 presidential election returns with demographics. <http://vizlab.nytimes.com/visualizations/treemap-for-comparisons-of-county-le>.
- [2] Bbc news - the top 100 sites on the internet, March 2010. <http://news.bbc.co.uk/2/hi/8562801.stm>.
- [3] K. Andrews and J. Ksanicka. A comparative study of four hierarchy browsers using the hierarchical visualisation testing environment (hvtc). In *Information Visualization, 2007. IV '07. 11th International Conference*, pages 81–86, July 2007.
- [4] M. Balzer, O. Deussen, and C. Lewerentz. Voronoi treemaps for the visualization of software metrics. In *Proceedings of the 2005 ACM symposium on Software Visualization*, pages 165–172. ACM, 2005.
- [5] T. Barlow and P. Neville. A comparison of 2-d visualizations of hierarchies. In *INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, page 131, Washington, DC, USA, 2001. IEEE Computer Society.
- [6] B. B. Bederson, B. Shneiderman, and M. Wattenberg. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. *ACM Trans. Graph.*, 21(4):833–854, 2002.
- [7] M. Bertamini and D. Proffitt. Hierarchical motion organization in random dot configurations. *Journal of Experimental Psychology: Human Perception and Performance*, 26:1371–1386, 2000.
- [8] J. Bertin. *Sémiologie Graphique*. Gauthier-Villars, 1967.
- [9] D. M. Bruls, C. Huizing, and J. J. van Wijk. Squarified treemaps. In *Data Visualization Eurographics/IEEE Symp.*, pages 33–42, 2000.
- [10] W. S. Cleveland, P. Diaconis, and R. McGill. Variables on scatterplots look more highly correlated when the scales are increased. *Science*, 216:1138–1141, Jun 1982.

- [11] W. S. Cleveland, C. S. Harris, and R. McGill. Judgments of circle sizes on statistical maps. *Journal of the American Statistical Association*, 77(379):541–547, 1982.
- [12] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *J. Am. Statistical Assoc.*, 79:531–554, 1984.
- [13] F. Cole, K. Sanik, D. DeCarlo, A. Finkelstein, T. Funkhouser, S. Rusinkiewicz, and M. Singh. How well do line drawings depict shape? In *ACM SIGGRAPH*, pages 1–9, 2009.
- [14] S. Few. Time on the horizon, Jun/Jul 2008.
- [15] M. Friendly. A brief history of data visualization. In C. Chen, W. Härdle, and A. Unwin, editors, *Handbook of Data Visualization*, pages 15–56. Springer, 2008.
- [16] W. R. Garner. *The processing of information and structure*. Erlbaum, 1974.
- [17] J. Heer and M. Bostock. Crowdsourcing graphical perception: Using mechanical turk to assess visualization design. In *ACM CHI*, 2010.
- [18] J. Heer, N. Kong, and M. Agrawala. Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. In *ACM CHI*, pages 1303–1312, 2009.
- [19] D. A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Trans. on Visualization and Comp. Graphics*, 6:59–78, 2000.
- [20] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *ACM CHI*, pages 453–456, 2008.
- [21] A. Kobsa. User experiments with tree visualization systems. In *Information Visualization*, pages 9–16. IEEE CS Press, 2004.
- [22] H. Lam, T. Munzer, and R. Kincaid. Overview use in multiple visual information resolution interfaces. *IEEE Trans. on Visualization and Comp. Graphics*, 13:1278–1285, Nov/Dec 2007.
- [23] H. Lü and J. Fogarty. Cascaded treemaps: Examining the visibility and stability of structure in treemaps. In *Proceedings of graphics interface 2008*, pages 259–266. Canadian Information Processing Society, 2008.
- [24] I. S. MacKenzie. Fitts’ law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7:91–139, 1992.
- [25] T. Saito, M. H. N., M. Yamamoto, H. Saito, Y. Hoshiya, and T. Kaseda. Two-tone pseudo-coloring: Compact visualization for one-dimensional data. In *IEEE Information Visualization*, pages 173–180, 2005.
- [26] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM TOG*, 11:92–99, 1992.
- [27] B. Shneiderman and C. Plaisant. Treemaps for space-constrained visualization of hierarchies. <http://www.cs.umd.edu/hcil/treemap-history/>.
- [28] B. Shneiderman and M. Wattenberg. Ordered treemap layouts. In *Proceedings of the IEEE Symposium on Information Visualization*, volume 2001, pages 73–8. Citeseer, 2001.
- [29] D. Simkin and R. Hastie. An information-processing analysis of graph perception. *J. Am. Stat. Assoc.*, 82:454–465, Jun 1987.
- [30] I. Spence. The apparent and effective dimensionality of representations of objects. *Human Factors*, 46:738–747, 2004.
- [31] J. Stasko, R. Catrambone, M. Guzdial, and K. McDonald. An evaluation of space-filling information visualizations for depicting hierarchical structures. *International Journal of Human-Computer Studies*, pages 663–694, 2000.
- [32] J. Stasko and E. Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *IEEE Information Visualization*, pages 57–65, 2000.
- [33] S. S. Stevens. The psychophysics of sensory function. *American Scientist*, 48:226–253, 1960.
- [34] M. Teghtsoonian. The judgment of size. *The American Journal of Psychology*, 78(3):392–402, 1965.
- [35] E. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 1983.
- [36] F. van Ham and J. J. van Wijk. Beamtrees: Compact visualization of large hierarchies. In *Information Visualization*, pages 93–100. IEEE CS Press, 2002.
- [37] J. J. van Wijk and H. V. de Wetering. Cushion treemaps: visualization of hierarchical information. In *Information Visualization*, number 147, pages 73–78. IEEE, 1999.
- [38] C. Ware. *Information Visualization: Perception for Design*. Morgan-Kaufmann, 2nd edition, 2004.
- [39] M. Wattenberg. Visualizing the stock market. In *ACM CHI Extended Abstracts*, pages 188–189, 1999.
- [40] M. Wattenberg, I. Res, and N. White Plains. A note on space-filling visualizations and space-filling curves. In *IEEE Symposium on Information Visualization*, 2005. *INFOVIS 2005*, pages 181–186, 2005.
- [41] M. Weskamp and D. Albritton. Newsmap. Retrieved May, 11:2008, 2002.
- [42] K. Wetzel. Circular treemaps. <http://lip.sourceforge.net/ctreemap.html>.
- [43] D. Wigdor, C. Shen, C. Forlines, and R. Balakrishnan. Perception of elementary graphical elements in tabletop and multi-surface environments. In *ACM CHI*, pages 473–482, Apr 2007.
- [44] A. Woodruff, J. A. Landay, and M. Stonebraker. Constant information density visualizations of non-uniform distributions of data. In *ACM CHI*, pages 19–28, 1998.
- [45] C. Ziemkiewicz and R. Kosara. The shaping of information by visual metaphors. *IEEE TVCG*, 14:1269–1276, Nov/Dec 2008.