Dynamic Opacity Optimization for Scatter Plots

Justin Matejka, Fraser Anderson, and George Fitzmaurice
Autodesk Research, Toronto Ontario Canada
{*first.last@autodesk.com*}

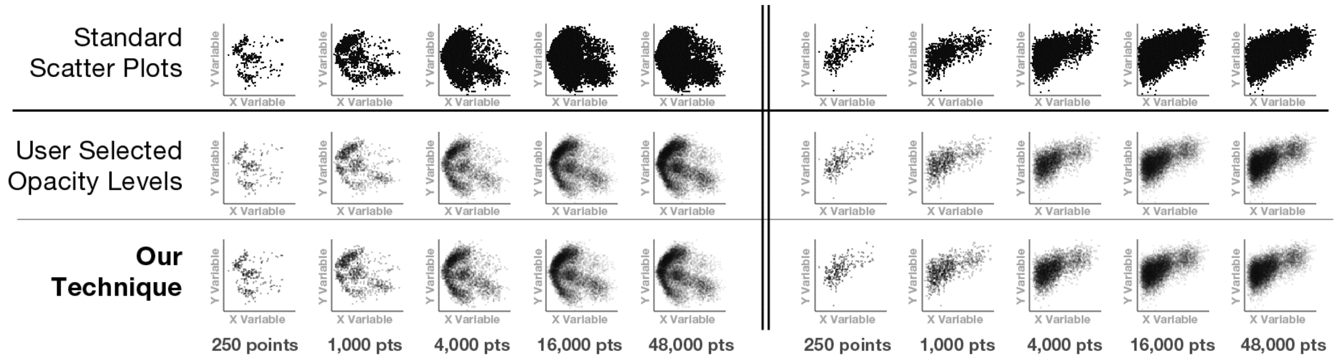

Figure 1. Scatter plots for two data sets (left side and right side) with varying numbers of data points rendered. The top row shows the appearance with an individual point opacity of 100%, while the second and third rows show the crowd-sourced results for the opacity scaling task and the results of our technique respectively.

ABSTRACT
Scatterplots are an effective and commonly used technique to show the relationship between two variables. However, as the number of data points increases, the chart suffers from “over-plotting” which obscures data points and makes the underlying distribution of the data difficult to discern. Reducing the opacity of the data points is an effective way to address over-plotting, however, setting the individual point opacity is a manual task performed by the chart designer. We present a user-driven model of opacity scaling for scatter plots. We built our model based on crowd-sourced responses to opacity scaling tasks using several synthetic data distributions, and then test our model on a collection of real-world data sets.

INTRODUCTION
Scatterplots are a common and effective method to visualize relationships between two-dimensional data and to quickly identify trends and outliers within a dataset [6]. However, scatter plots suffer from over-plotting as the ratio of data points to chart area increases. When over-plotting occurs, data points can be occluded and information may be lost. This can make it difficult or impossible to see the individual data points and lead to misinterpretation of the data, or the inability to perceive the data’s underlying distribution.

The primary strategies to mitigate over-plotting are to: reduce the size of the data points, remove the color fill from the data points, change the shape of the data points, jitter the data position, make the glyphs semi-transparent, and reduce the amount of data [2]. The first four techniques can work in situations of moderate over-plotting. However, they are not applicable in situations with extreme over-plotting or when the data points are already represented by very small marks. It is often undesirable to reduce the amount of data, either through sampling or aggregation, which leaves the modification of the data point opacity as the most desirable option in many situations. Most graphing packages support manual modifications of the point opacity, however they default to 100% opacity, and do not provide any mechanism for automatically selecting a more appropriate opacity level.

Making the individual data points semi-transparent does not modify or distort the underlying data and is applicable to a number of graphing scenarios. However, it is not obvious how to programmatically set the opacity value for a given graph, or in the more complicated case of a dynamic charting environment, how to modify the opacity level as the quantity and distribution of points changes.

In this paper we present a user-driven model of opacity scaling for scatter plots. We built our model based on crowd-sourced responses to opacity scaling tasks using several synthetic data distributions, and then test our model on a set of real-world data with several graph area/point area combinations.

RELATED WORK
Alternative methods of visualizing two-dimensional data include 2D histograms, hexbin plots, and contour plots. While each is useful for situations of high over-plotting, they aggregate data into groups, making them potentially less useful in situations with low over-plotting. We focus on scatter plots since they can provide a uniform and useful experience in scenarios with both low and high over-plotting.

Submitted to CHI 2015

Luboschik, Radloff and Schumann apply a weaving technique which allows observers to easily distinguish plotted groups via hue [5]. This technique is very promising for scatter plots with multiple subgroups, but does not have an effect for single-group data sets. Mayorga and Gleicher addressed the problem of over-plotting in multi-dimensional scenarios by creating a new chart type called a 'splatterplot' by finding contours for each of the data sets and using color blending [6].

In contrast to previous approaches, we present a technique to automatically and dynamically select an appropriate data point opacity level for a given scatter plot. As modification of the opacity level is already supported in most charting packages, our algorithm can be directly and easily incorporated into existing plotting workflows.

STUDY 1: USER DERIVED OPACITY CURVES

We wanted to create a model for opacity scaling which aligns with the aesthetic and functional choices chart designers use when manually setting the opacity level of a scatter plot. So to begin, we collected users' perceptual preferences by asking them to manually set the opacity level over a controlled set of charts.

Participants

Amazon's Mechanical Turk service has been shown to be a useful method for conducting graphical perception studies [3]. We recruited 46 crowd workers from Mechanical Turk. As is common in crowd-sourced studies [4], 15 of the workers attempted to 'game the system' by clicking through the tasks quickly and randomly, leaving 31 valid participants with data for analysis. The workers were paid $2USD for the study which took an average of 10 minutes to complete, for an equivalent wage of $12/hour.

Design

Each trial in the study consisted of showing the participant a scatter plot, and having them move a slider left and right to adjust the opacity level. The participants were instructed to set the opacity to what they thought provided the best overall legibility in both the light and dark areas of the chart. After making their opacity selection, the next trial would begin.

To capture data for a range of values from charts with little or no over-plotting, to charts with a great deal of over-plotting, we generated graphs with 27 unique *number of points*, ranging from 1 (1^3) to 19,683 (27^3) (Figure 2). The *number of points* is a component of the *over-plotting factor* which we define as:

$$\text{over-plotting factor} = \frac{\#\text{ of pts} \times \text{area of each point}}{\text{area of the chart}}$$

For example, if a chart has an over-plotting factor of 4x, there are 4 times as many pixels needed to represent the data than are available in the chart, so over-plotting will be necessary. All plots in the first study were 80x80 pixels in size, and the data points were 2x2 pixel squares, resulting in *over-plotting factors* ranging from 0.0006x for the *1 point* condition, and 12.3x for the *19,683 points* condition.

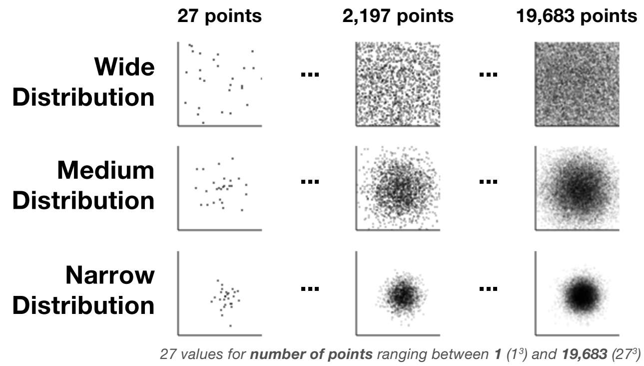

Figure 2. The three *distribution* types used in the first study, with representative samples from the *number of point* range.

We also believed that the distribution of the data points within the chart would affect the ideal opacity setting, so we used three *distribution* types: *wide, medium,* and *narrow* (Figure 2). All three data sets were Gaussian distributions centered at 0.5 and bound between 0 and 1 on each axis, with standard deviations of 0.7, 0.2, and 0.1 respectively.

The study was divided into 3 blocks, with each user selecting an opacity value for each of the 81 *distribution* × *number of points* combinations presented in a random order within each block for a total of 243 trials per participant.

Results

The individual measurements as well as the averaged user-medians from the 81 *distribution/number of points* conditions are shown in Figure 3. As expected, the *distribution* was a significant factor ($F_{2,25} = 31.9$, $p < .0001$) in the resulting opacity values, with the *wide* distribution having the highest opacity settings and the *narrow* distribution the lowest.

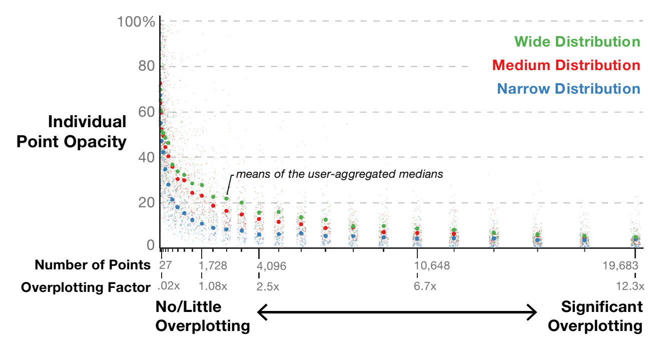

Figure 3. Point opacity values from the first study.

To model these user-generated opacity curves, we wanted to find a property of the resulting scatter plot which stayed relatively constant over a wide range of data point counts and was independent of the *distribution* type.

We discovered a promising metric meeting these criteria by looking at the mean opacity of only the *utilized* pixels of the resulting graph. That is, the sum the final opacities of each pixel in the entire plot, divided by the number of pixels which had an opacity greater than zero (Figure 4).

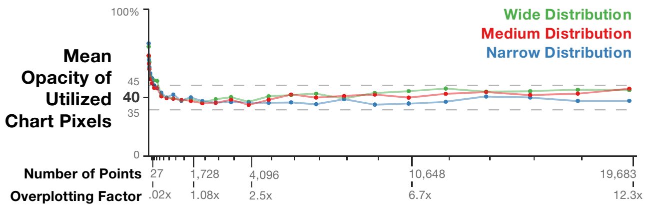

Figure 4. Mean opacity of the *utilized* chart pixels from the charts produced by the users in Study One.

We can see that, except where the *over-plotting factor* is very low, the Mean Opacity of Utilized Pixels (*MOUP*) stays relatively constant (around 40%) and does not seem to be impacted by the distribution type. It is this property, that users create scatter plots with a fixed MOUP of 40% in over-plotting scenarios, which we use to derive our model.

ALGORITHMIC MODEL

To model the user-preference for scatter plot opacities found in the first study, we developed an algorithm which will produce a scatter plot with a MOUP of 40%.

We do this by arranging all data points on the graph surface and for each pixel in the chart area, counting the number of overlapping data points which cover this pixel. Based on the standard color blending model [7] the final opacity (O_f) of a pixel given a number of overlapping layers (l) at a given individual opacity level (α) can be calculated as follows:

$$O_f(l,\alpha) = \begin{cases} \alpha + O_f(l-1,\alpha)\cdot\alpha, & l > 1 \\ \alpha, & l = 1 \end{cases}$$

Then, the MOUP of a given plot with a set of p pixels (P), with a specified number of layers (l_p) and a particular individual point opacity level (α) can be calculated thusly:

$$MOUP(P,\alpha) = \frac{\sum_P^p O_f(l_p,\alpha)}{\sum_P^p 1, \text{where } l_p > 0}$$

The problem then becomes finding an optimal individual point opacity ($\alpha_{MOUP\_0.4}$) which produces a MOUP of 40%:

$$MOUP(P,\alpha) = 0.4, \text{where } \alpha = \alpha_{MOUP\_0.4}$$

Looking at Figure 4, it appears that targeting an individual point opacity level which produces a MOUP of 0.4 should work in cases where the over-plotting factor is greater than around 0.5x, but for low over-plotting factors $\alpha_{MOUP\_0.4}$ will produce a chart with a lower overall opacity than desired. The error between $\alpha_{MOUP\_0.4}$ and α_{user} in these cases is fairly consistent across the distribution types and follows a logarithmic distribution. A *Low Density Multiplier* (*LDM*) term is defined for a given over-plotting factor (*opf*) to account for this gap in the low over-plotting scenarios:

$$LDM_{opf} = \min\left\{1, 1 - 0.15 \times \log\left(\frac{opf}{0.75}\right)\right\}$$

The LDM term only affects the calculation for charts with an over-plotting factor < 0.8x, otherwise the LDM term evaluates to 1. Incorporating the LDM term onto $\alpha_{MOUP\_0.4}$ gives us an optimal individual point opacity ($\alpha_{optimal}$) of:

$$\alpha_{optimal} = LDM_{opf} \times \alpha_{MOUP\_0.4}$$

Our un-optimized implementation solves for $\alpha_{optimal}$ using a binary search over the opacity space to a precision of three decimal places, and can compute at 30fps for graphs up to 250×250 pixels in size.

The individual point opacities generated by our algorithm as compared to the user-generated levels are shown in Figure 5. A Pearson's r test shows the overall correlation of R^2=.9904.

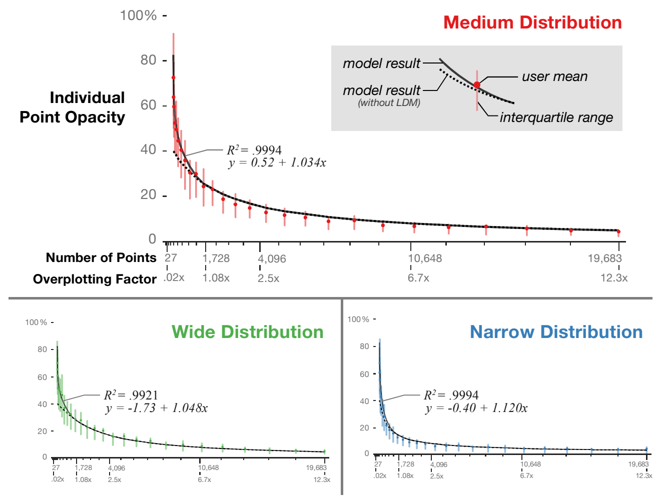

Figure 5. Graph of the algorithmic model results overlaid on the user-generated results.

The correlations for each of the *distributions* are all very high, as shown in Figure 5, and the effect of the LDM can be seen by observing the difference between the solid and dashed model result lines.

STUDY 2: REAL-WORLD DATA

Knowing that our model fits well for a set of procedurally generated scatter plots with very smooth distribution curves, we conducted a second study to see if those results would extend to real-world data sets.

Participants and Design

As in the first study, participants were recruited from Amazon's Mechanical Turk and paid similarly. 55 workers signed up to the task, and after removing 12 for attempting to game the system, we were left with 43 valid participants.

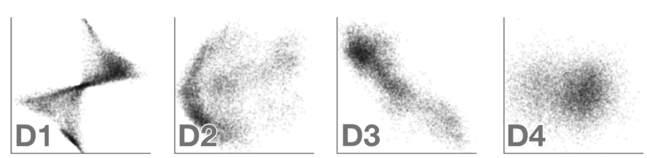

Figure 6. Distributions of data used for the validation study.

The study task was similar to that in Study 1, participants were presented a scatter plot and were instructed to set the individual point opacity using a slider.

Rather than the Gaussian distributions used in the first study, the second study used four *distributions* (*D1-D4*) derived from the physical properties of baseball pitching data [1] (Figure 6). We also wanted to verify that our method applies to various graph and dot sizes, so three *configurations* were used: *C1* is the same configuration as was used in the first study, while *C2* and *C3* are considerably larger at 250x250 pixels, and *C3* uses a larger dot size (Figure 7).

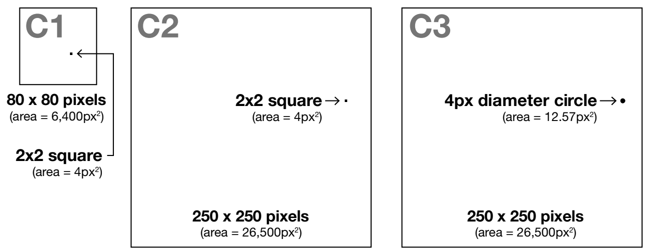

Figure 7. Grid/Dot configurations used for Study Two.

Finally, a set of five *number of point* levels were used: *250*, *1000*, *4000*, *16000*, and *48000*. These point counts resulted in different *over-plotting factors* for each of the grid/dot size configurations, ranging from a minimum of 0.038x in the *250 dot/C2* condition, up to 30x in the *48000 dot/C1* condition. Overall, there are 4 blocks of 5x4x3=60 conditions, resulting in each participant completing 240 trials.

Results

Comparing to the mean of the user-aggregated medians to the opacity level predicted by our model for each of the 60 conditions through a Pearson's r test shows our model has a high correlation to the user data ($R^2=0.9899$), and in all cases the predicted value falls within the interquartile range of the user data. The results for each of the conditions is shown in Figure **8** along with the R^2 correlation value for each *configuration/distribution* combination. Examples of the user-generated scatter plots and the output of our algorithm can be seen in the bottom two rows of Figure 1.

DISCUSSION AND FUTURE WORK

As a simplifying measure, we looked exclusively at scatter plots with a white background and square or circular points. Initial tests indicate that the opacity values produced by our algorithm work well for plots with different colored points, but it would be interesting to validate the model with different backgrounds and data-point shapes.

Our approach was to calculate an optimal opacity value for each specific scatter plot. In cases where those calculations are not feasible to do in real time (for extremely large graphs, or on a mobile device for instance), a 'generic' opacity curve could be pre-calculated, perhaps using the *medium* distribution from Study One. While not tuned precisely to the underlying data, using this generic opacity curve would still be better than the default 100% opacity setting.

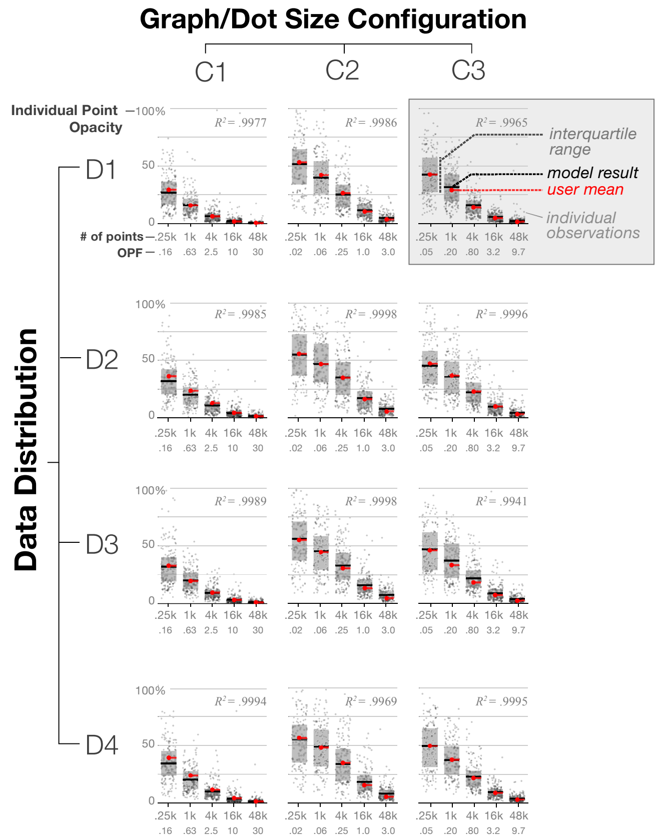

Figure 8. The results of the second study, by *configuration* and *distribution* type.

CONCLUSIONS

We have presented a model of opacity-scaling for over-plotted scatter plots derived from the aesthetic and functional choices made by users when asked to manually choose an opacity value. The output from our model can be easily integrated into existing scatter plot implementations, making them more useful under a variety of over-plotting scenarios.

REFERENCES

1. Fast, M. What the heck is PITCHf/x. *The Hardball Times Annual*, (2010), 153–8.
2. Few, S. Solutions to the Problem of Over-plotting in Graphs. *Visual Business Intelligence Newsletter*, (2008).
3. Heer, J. and Bostock, M. Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design. *ACM CHI '10*, (2010), 203–212.
4. Kittur, A., Chi, E.H., and Suh, B. Crowdsourcing User Studies with Mechanical Turk. *ACM CHI '08*, (2008), 453–456.
5. Luboschik, M., Radloff, A., and Schumann, H. A New Weaving Technique for Handling Overlapping Regions. *ACM AVI '10*, (2010), 25–32.
6. Mayorga, A. and Gleicher, M. Splatterplots: Overcoming Overdraw in Scatter Plots. *IEEE Transactions on Visualization and Computer Graphics 19*, 9 (2013), 1526–1538.
7. Porter, T. and Duff, T. Compositing Digital Images. *ACM SIGGRAPH '84*, (1984), 253–259.