# Towards Perceptual Optimization of the Visual Design of Scatterplots

Luana Micallef, Gregorio Palmas, Antti Oulasvirta, and Tino Weinkauf

**Abstract**—Designing a good scatterplot can be difficult for non-experts in visualization, because they need to decide on many parameters, such as marker size and opacity, aspect ratio, color, and rendering order. This paper contributes to research exploring the use of perceptual models and quality metrics to set such parameters automatically for enhanced visual quality of a scatterplot. A key consideration in this paper is the construction of a cost function to capture several relevant aspects of the human visual system, examining a scatterplot design for some data analysis task. We show how the cost function can be used in an optimizer to search for the optimal visual design for a user's dataset and task objectives (e.g., "reliable linear correlation estimation is more important than class separation"). The approach is extensible to different analysis tasks. To test its performance in a realistic setting, we pre-calibrated it for correlation estimation, class separation, and outlier detection. The optimizer was able to produce designs that achieved a level of speed and success comparable to that of those using human-designed presets (e.g., in R or MATLAB). Case studies demonstrate that the approach can adapt a design to the data, to reveal patterns without user intervention.

**Index Terms**—Scatterplot, optimization, perception, crowdsourcing.

## 1 Introduction

In this paper, we investigate the design of scatterplots, one of the most important methods for visualization of bivariate data and widely used beyond the sciences, even in newspapers and other non-academic contexts. Scatterplots are utilized by numerous people who are unfamiliar with the method's technical details.

Our goal is to support non-experts in the rapid design of effective scatterplots. For a non-expert, designing a scatterplot can be a complex task. Consider a meteorologist, a domain expert but a novice in visualization, studying the correlation between pressure and temperature for Hurricane Isabel. Using a readily available statistical package, the scatterplots in Figure 1(a) are quickly generated, yet the default designs often represent the data poorly. This is mainly because they cannot adapt to the data. Figure 1(a) shows the same data with two sampling resolutions and yet, due to the fixed design, the scatterplots do not convey the strong similarity in the data. Improved scatterplot designs can be obtained by manually adjusting the marker size, marker opacity, the colors, the aspect ratio of the plot, and other technical aspects in line with the data and the required analysis task, but such adjustments are difficult for non-experts. Furthermore, they depend on the task: spotting outliers will be difficult when one uses a low marker opacity, but estimating the density of a large number of data points is a task that can benefit from a low opacity.

This paper presents results from an investigation of algorithmic approaches to automatic scatterplot design – in particular, by exploiting models and measures of human perception. With our approach, the plots in Figure 1(b) are automatically generated for the meteorologist once she provides the dataset and specifies her principal data analysis task (see Figure 2 for a system overview).

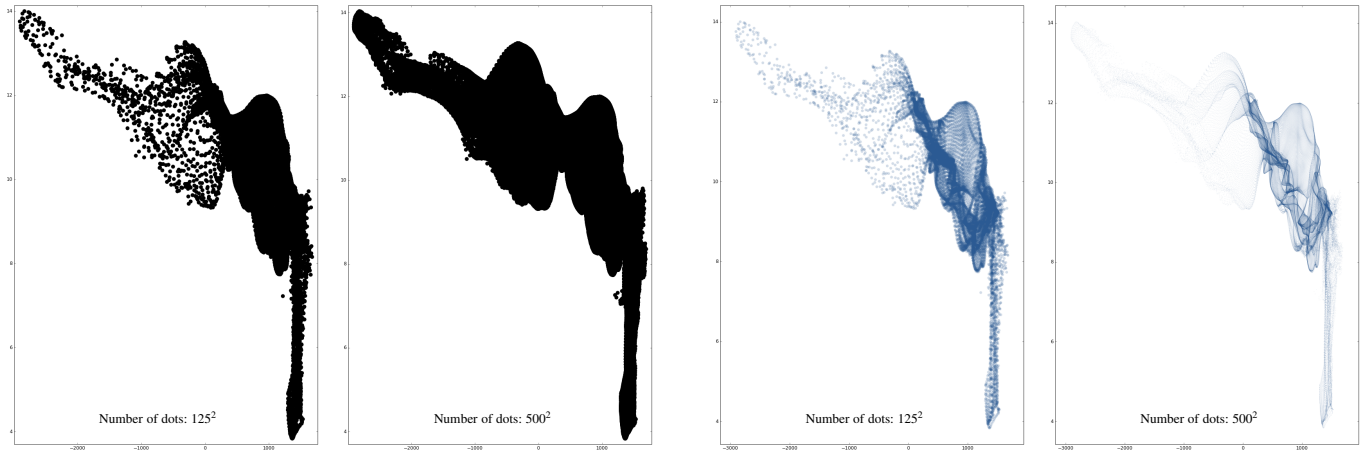
Forming a key part of our research is a set of quality metrics and models that aid in predicting how a user might examine a scatterplot when performing a task with it. We build a cost function for an optimizer that can address aspects such as contrast, unit formation (how users see groups in scatterplots), and structural similarity. The cost function is generic and can be adapted to various data analysis tasks. For doing so, the components of the model are tuned by means of a set of weights. After calibration, the optimizer can be used in different instances of the task. In this paper, we consider three tasks often carried out through scatterplots: correlation estimation, class separation, and outlier detection. We consider *class separation* and not *cluster detection* because often data analysts color-code the points of the dimensionally-reduced data with respect to pre-existing class labels [1], [2]. Clusters are different from classes: *clusters* are spatially separated groups of non-labeled data. In contrast, *classes* are groups of data points with the same label that are not necessarily spatially separated, leading to various class-perceivability-related design challenges in the creation of scatterplots. To set the weights for these three tasks, we performed a series of crowdsourced user studies comparing user performance across different weights.

We use the model in a combinatorial optimization with a discretized search space to find a design for the scatterplot automatically that is well suited to a given task and dataset. Technically, this is a multi-objective optimization approach with each weighted term describing a perceptual aspect of a scatterplot. The idea of using predictive models in the design of displays and interfaces is applied also in computer graphics and human-computer interaction. The benefit of this approach is that the optimizer can predict how users will respond to a given design.

A key component of this paper describes the construction of a cost function, building on the long history of research into quality metrics in visualization (Section 2). To the best of our knowledge,


- *L. Micallef is with the Helsinki Institute for Information Technology HIIT, Department of Computer Science, Aalto University, Finland. E-mail: luana.micallef@hiit.fi*
- *G. Palmas and T. Weinkauf are with Department of Computational Science and Technology, School of Computer Science and Communication, KTH Royal Institute of Technology, Sweden. E-mails: {gpalmas,weinkauf}@kth.se*
- *A. Oulasvirta is with the Department of Communications and Networking, Aalto University, Finland. E-mail: antti.oulasvirta@aalto.fi*


*Manuscript received X, Y; revised Z, V.*




(a) The standard design loses fidelity and requires manual adjustment.

(b) Optimized designs automatically adapt to increasing dataset resolution.

Figure 1. Our method uses models of human perception for automatically determining design parameters of a scatterplot for given objectives. It can be used for one-shot design as well as in cases wherein input data or requirements vary. Here, we plot pressure and temperature variables from the Hurricane Isabel dataset with increasing dataset resolutions.

our approach represents the most extensive implementation of perceptual models and measures for the algorithmic design of scatterplots and we provide the first optimization-based formulation of the visual design of scatterplots.

The work presented in this paper may be beneficial to users in two distinct ways: Firstly, the proposed optimization approach may offer a useful tool that complements the default settings over a wide spectrum of settings. A scatterplot design can be generated simply by supplying a dataset and specifying priorities for the data analysis goals. Assuming that the cost function has been calibrated (see below), the end user does not need to set visual parameters manually. At the same time, the series of crowdsourced user studies we carried out yielded insight into user performance with scatterplots generated with different modeling assumptions, which may also be useful for other approaches for automating the design of scatterplots.

In summary, our main contribution is a model-based optimization approach that synthesizes knowledge in our field in a manner that can be readily operationalized. Instead of pooling more empirical results, we attempt to formulate design as a mathematical cost function that is fully scrutinizable and builds on existing work. The main contributions are that we:

- Present a novel method for perceptual optimization of the visual design of scatterplots.
- Adapt image quality measures and models of the human visual system to drive the selection of optimal design parameters.
- Show how to support three common data analysis tasks by using this framework and how it can be extended to other tasks through a calibration procedure.

The paper is organized as follows: After providing an overview of the related work (Section 2), we briefly review the definition, data requirements, design parameters, and data analysis tasks for scatterplots (Section 3). An overview of the method (Section 4) precedes the technical details, including the perceptual model, which are presented in Section 5. We describe a general empirical method to calibrate the weight factors of the perceptual model to a specific data analysis task (Section 6). After this, we report on an evaluation study (Section 7) and discuss characteristics of our approach with reference to real data (Section 8). Finally, we present conclusions (Section 9), along with limitations and avenues for future work (Section 10).

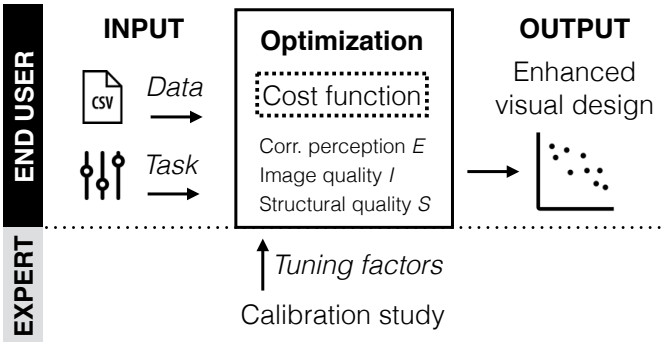



Figure 2. An overview of the approach, showing the end user's view and the visualization expert's involvement.

## 2 Related Work

In this brief review, we focus on three subjects: (i) empirical studies examining the visual design of a scatterplot in relation to user performance, (ii) visual quality metrics for scatterplots, and (iii) optimization approaches for scatterplot design. For more details, refer to excellent overviews by other authors such as [3], [4], [5], [6].

### 2.1 Empirical Studies of Scatterplot Perception

Munzner [7] proposed a taxonomy for goals related to scatterplot inspection, distinguishing correlation, outliers, classes, trends, and distribution. We use this as the organizational principle for reviewing empirical studies.

Correlation estimation was studied by Meyer et al. [8], who concluded that correlation estimation is subject to biases and relatively difficult for humans. Intuitive estimates are typically lower than the statistical coefficient $r$ and affected by visual

characteristics. Cleveland et al. [9] found that the point cloud in the scatterplot tends to be perceived more correlated as the scales on the axes are increased such that the cloud's size is decreased. Li et al. [10] concluded that scatterplots double the number of reliably distinguished levels compared to parallel coordinate plots. Harrison et al. [11] compared scatterplots to other visualization methods for correlation perception tasks, and concluded that the *just noticeable differences* in correlation in all of these visualizations can be modeled by Weber's law. Rensink and Gideon [12] had previously reported a similar result from their laboratory study. Kay and Heer [13] further analyzed Harrison et al.'s collected data and found that, of all the visualizations evaluated, scatterplots exhibit the least inter-individual variance and the highest precision. However, in all of this work, only one scatterplot design (with a small marker size of ≈2 pixels, full opacity, and an aspect ratio of 1) and one type of dataset (≈100 normally distributed points) was assessed.

Outlier detection was studied by Last and Kandel [14], who present an automated approach to detecting and isolating outliers. Rahman et al. [15] propose a multiple linear regression analysis for the same purpose. Several outlier detection algorithms have been proposed (see survey [16]), but typically properties of human perception are not taken into account and the effectiveness of the methods is not evaluated empirically.

Cluster separation was studied by Sedlmair et al. [2], who list within-class factors (e.g., density and outlier) and between-class factors (e.g., variance of density and split) affecting how easily clusters can be found, verified, and matched to classes on a scatterplot. They identify class separation as a between-class factor that is dependent on and strongly influenced by all other within- and between-class factors. These design factors were consistent with the scagnostics (aspects of scattered points, such as outlier, shape, trend, density, and coherence) defined by Wilkinson et al. [5]. A follow-up study [17] indicated that a 2D scatterplot is as good as an interactive 3D scatterplot or a scatterplot matrix in showing classes. Gleigher et al. [18] explored the visual design of multi-class scatterplots for the task of identifying the mean value of a set of points. Lewis et al. [19] compared grouping quality metrics with human evaluations by using a diverse set of measures based on a theoretical taxonomy. The authors concluded that grouping evaluation skill is present in the general population.

### 2.2 Visual Quality Metrics

Visual quality metrics are typically used for automatically determining how much information a visualization provides to the analyst [6].

Visual class separation was studied by Sedlmair and Aupetit [20], who proposed a data-driven framework to evaluate metrics designed to identify how separable classes in a projection of a high-dimensional dataset would be perceived. Using this framework, 2002 new visual separation measures were defined and evaluated [21]. User judgment was exploited in selection of the best quality metric for use to automatize a specific task. In contrast, we exploit user judgment of scatterplots to calibrate the weights used for selection of the most effective design for a specific task.

Scheidewind et al. [22] proposed automatic analysis of pixel images produced by different parameter mappings to rank them according to their potential value to the user. This allows reducing the parameter space to obtain insights more quickly.

### 2.3 Automatic and Semi-Automatic Design

Etemadpour et al. [23] proposed perception-based evaluation of projection methods based on empirical data but not perceptual models and quality metrics as in our approach. Color optimization in scatterplots was adopted by Chen et al. [24] to improve the discernibility of multiple classes when the scatterplot is overplotted. Mayorga et al. [25] used a similar technique to blend colors and show color-filled regions when classes overlap greatly. Yet these techniques optimize only one design parameter (i.e., class color). Johansson and Johansson [26] demonstrated an interactive technique to reduce dimensionality via quality metrics. They used weight functions to preserve as many important structures as possible for exploration, along with quality metrics for class separation, outliers, and correlation in the data space – but not in the image space as in our approach. Tatu et al. [27] used quality metrics in the image space for scatterplots and parallel coordinates to rank 2D projections of a multivariate dataset with the aim of speeding up data exploration, rather than optimizing the plot design.

Matejka et al. [28] defined a model of opacity scaling for scatterplots that is based on the data distribution and crowdsourced responses to opacity scaling tasks. Yet they did not assess the effect of scatterplot design or marker opacity on user performance in accomplishing data analysis tasks. Heer and Bostock [29] proposed a crowdsourcing-based approach for assessing visual designs. In particular, they studied separation and layering via luminance contrast in a scatterplot, but they did not define any models incorporating their findings.

## 3 Scatterplots: Data, Design and Tasks

A scatterplot shows the relationship between two variables $x, y$ by plotting a marker for each data point $(x_i, y_i)$ in a 2D Cartesian coordinate system spanned by $x$ and $y$. It is a well-established tool for inspecting potential correlations and other patterns between variables. This is done by considering various aspects of the shown markers, including how spread out they are, their slope, the distribution of mass, and the existence of outliers.

Scatterplots are typically used with numeric variables. The data points can be assigned a class label and color-coded on the basis of their class. Outliers represent another interesting characteristic of the data. The definition of outliers is application-dependent and often difficult to formalize, but the general idea is that outliers deviate from the main distribution. The two variables shown in a scatterplot can be part of a higher-dimensional multivariate dataset, in which case, the scatterplot represents a 2D projection. If grouping occurred in the higher-dimensional space, it is likely that there is overlap in the 2D projection also. Similarly for outliers, they may not appear isolated in the 2D scatterplot.

The design of a scatterplot consists of several interrelated choices of: (i) marker size, (ii) marker opacity, (iii) marker color, (iv) marker shape, (v) aspect ratio of the plot, (vi) width of the plot, (vii) drawing order of points or classes, and many more aspects. These items not only are interrelated but depend on the data and the analysis task. Consider as an example the size of the markers: it depends on the overall size of the plot, the varying density of the data points in the plot, and whether or not the analysis task requires the user to identify individual points. Furthermore, the visual impact of a marker is affected by its opacity. Overlapping markers and overplotted images may be beneficial for some tasks but not others. In short, the design possibilities can be overwhelming,

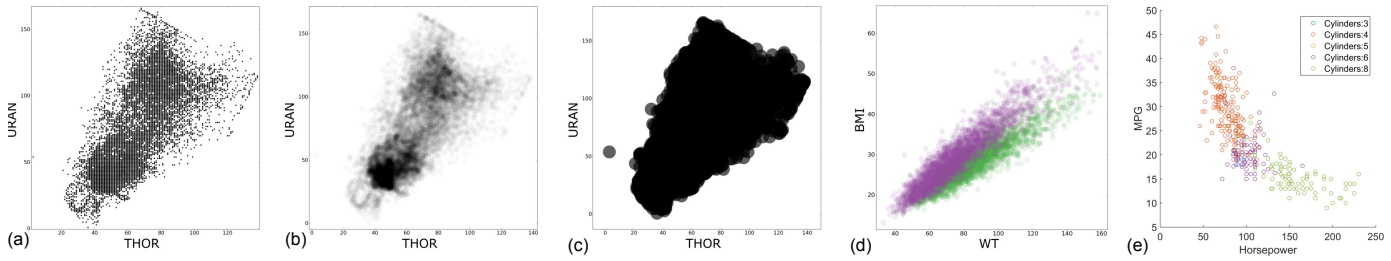



Figure 3. The design possibilities for scatterplots vary greatly. Different designs support different tasks for the same data: (a)-(c) show two variables from the *out5d* dataset in radically different designs, thereby differing in the information they reveal: the discretization of the variables, the density of the data points, and the outlier. Plots (e)-(f) show variations in color, marker size, and shape.

particularly for novices. Figure 3 illustrates the wide variety of design possibilities for scatterplots.

Scatterplots are used for many purposes. These are the most common user tasks reported in the literature (see e.g. [7]):

- **Correlation estimation:** Assessing the linear correlation between two variables by considering the spread of the points and the slope of the point cloud [9], [10], [11], [12]. The design influences the perception of correlation – in particular the aspect ratio of the plot – as demonstrated in Section 6 for sub-study *CorrAR*.
- **Class separation:** Finding all classes [1], [2], [21]. These might be visually separated or overlapping; it depends on how the grouping was performed. We assume that different classes are shown with different colors. Many design parameters can influence this task. If nearby or overlapping classes are shown with very similar colors, it may be difficult to distinguish between them. High opacity may make the borders of a class pronounced but also hide other classes underneath.
- **Outlier detection:** Finding all outliers [14]. Since outliers are singular points, designs with low opacity and small marker sizes render it difficult to find them.
- **Distribution detection:** Assessing the varying density of the points [28].
- **Point value reading:** Being able to read the $x,y$-values for specific points [29].

We note that there are a myriad tasks, imposing different requirements on the design of a scatterplot. In the following sections, we present our approach to automatic determination of a design suitable for a specific task and dataset on the basis of perceptual models and measures.

## 4 Overview of the Approach

Figure 2 provides an overview of how the approach can be deployed. After initialization (calibration to a task), the system can be used for any instance of that task. In our case, we calibrate it to three data analysis tasks, thereby allowing optimization of any dataset for any of the three tasks. Figure 1(b) was generated with an optimizer that was pre-calibrated for correlation estimation, class separation, and outlier detection, as discussed in Section 6. To then use a calibrated system, the user inputs a dataset, specifies task objectives, and a scatterplot with a design optimized for that data and task is returned. For instance, Figure 1(b) is returned for the Isabel data and for the user's correlation task objective.

The system implements several models and measures of human visual perception to predict user performance of common tasks with scatterplots. This allows the optimizer to search numerous candidate designs (or even all of them) in order to find one that best meets the task objectives. The approach is easy and especially meaningful for non-experts who may not have the time or skills to explore large design spaces manually. This approach builds on model-based optimization such as that adopted for view optimization (e.g., [6]), user interface optimization (e.g., [30], [31]), layout optimization using metrics and heuristics (e.g., [32]), and perceptually optimized graphics (e.g., [33]).

The following sections describe how this approach can be made to work with three common design parameters and three common tasks. We address the design parameters marker size, marker opacity, and aspect ratio. This design space is challenging, since it can produce results that vary greatly; compare between figures 3 and 4. Other design parameters, such as image width, colors, marker shape, and the drawing order of classes, are fixed. In further specification, we select the following three tasks: (T1) correlation estimation, (T2) class separation, and (T3) outlier detection. As discussed above, these are among the most common data analysis tasks with scatterplots. We assume that classes and outliers have been precomputed, since data points can be grouped in various ways and the definition of outliers is application-dependent (see Section 3).

A key part of our work is the design of a novel cost function (see Section 5.1). It implements several perceptual models and measures. To model how users perceive correlation from a scatterplot, we predict how sparsely scattered points constitute a perceived entity. To do so, we adapt a model of unit formation [34] previously used to predict perception of objects in dot lattices. We use the Canny edge detector [35] and ellipse fitting to identify the perceived entities in a scatterplot. To model the perception of classes and outliers, we propose using the highly reliable image quality assessment model *structural similarity* [36], which assesses perceived differences between two images. In addition, we compute several generic perceptual measures such as contrast, the average amount of opacity, the amount of overplotting, and the overlap between markers. To our knowledge, this is the most extensive implementation of perceptual models and measures for the algorithmic design of scatterplots. Subsection 5.2 provides details.

These models and measures are combined in our cost function through weight factors. Different weight factors support different tasks. To determine how to weigh these factors for a given task, we carried out a series of crowdsourcing-based studies wherein we compared pre-defined weight sets to ascertain the best settings for each task. We chose the best weight factor settings using statistical testing.

The resulting cost function can be solved with any black box method [37] for discrete optimization – e.g., ant colonies [38]. After

discretization of the design parameters, the size of our design space is 4,851, but this can be easily changed in the code by adding or removing levels. Since our interest is not in real-time performance but in assessing the result quality that can be obtained, we used exhaustive search to find the global optimum.

The approach we describe here can be extended to cover additional design parameters and tasks. To add a design parameter, we do not necessarily need to add new terms to the cost function, since it already captures a wide range of perceptual aspects. For example, the impact of new design parameters such as *image width* or the *drawing order of classes* is already captured by the cost function. On the other hand, adding a new task requires a weight calibration following the procedure described in Section 6.

## 5 Perceptual Scatterplot Optimization

This section first describes the cost function and then introduces the constituent cost measures. In the latter part of the section, we describe the implementation of the optimizer.

### 5.1 Cost Function

Let us take two variables $x, y$ for a dataset to be visualized by means of a scatterplot. The set of all possible designs is

$$\mathcal{D} = \mathcal{S} \times \mathcal{O} \times \mathcal{A} \tag{1}$$

where $\mathcal{S}$ denotes the set of marker sizes considered, $\mathcal{O}$ is the set of marker opacities considered, and $\mathcal{A}$ is the set of aspect ratios considered for the plot. Certain designs $d \in \mathcal{D}$ will be better suited than others to quickly and accurately perform a particular data analysis task $t \in \{\mathrm{T1}, \mathrm{T2}, \mathrm{T3}\}$. We quantify this by using a cost function, $C(x,y,d,t)$, which has a low value when the design is deemed well suited to the data and task, and a high value otherwise. The cost function is a weighted sum of cost measures $\sum w(t)M(x,y,d)$, where weights $w$ depend on the task and the cost measures $M$ are determined by the data and the design. Each of the three terms in the cost function addresses a distinct perceptual aspects:

$$C(x,y,d,t) = E(x,y,d,t) + I(x,y,d,t) + S(x,y,d,t). \tag{2}$$

With these terms, we assess the perception of linear correlation $E(x,y,d,t)$, image quality measures $I(x,y,d,t)$, and the perception of individual classes and outliers in terms of structural similarity $S(x,y,d,t)$. Each of these terms is a weighted sum of cost measures:

$$E(x,y,d,t) = w_\alpha E_\alpha + w_r E_r \tag{3}$$

$$I(x,y,d,t) = w_\mu I_\mu + w_\sigma I_\sigma + w_{\bar{\mu}} I_{\bar{\mu}} + w_{\bar{\sigma}} I_{\bar{\sigma}} + w_\ell I_\ell + w_p I_p \tag{4}$$

$$S(x,y,d,t) = w_c S_c + w_o S_o \tag{5}$$

An overview is presented here, with details further on and a discussion of limitations saved for the end of the paper. All cost measures are functions of the form $(x,y,d) \to [0,1]$; i.e., they are normalized and do *not* depend on the task. They have the following meanings:

- $E_\alpha$: angle difference between covariance and perceived ellipse
- $E_r$: axes length ratio diff. between cov. and perceived ellipse
- $I_\mu$: average amount of opacity
- $I_\sigma$: image contrast
- $I_{\bar{\mu}}$: distance from a desired average amount of opacity
- $I_{\bar{\sigma}}$: distance from a desired image contrast
- $I_\ell$: amount of overlapping of the markers
- $I_p$: amount of overplotting
- $S_c$: perceivability of classes
- $S_o$: perceivability of outliers

We balance the weight of these measures in Equations (3)-(5) with the set of factors

$$\mathcal{W}(t) = \{w_\alpha, w_r, w_\mu, w_\sigma, w_{\bar{\mu}}, w_{\bar{\sigma}}, w_\ell, w_p, w_c, w_o\}. \tag{6}$$

We support different tasks by means of different weight factors. For a particular task $t$, the weight factors are constant values in the range $[-1,1]$. Negative weights allow us to reverse the meaning of a cost measure. For example, $w_p < 0$ denotes that we want to have a low amount of overplotting, whereas $w_p > 0$ indicates that we allow for a large amount of overplotting. By setting a weight factor to 0, we omit it from consideration. The non-zero weights are set empirically in the manner described in Section 6.

In summary, with a given dataset and task, our goal is to find the design for which the value of the cost function $C$ becomes minimal; i.e., we want to solve the following optimization problem:

$$\arg\min_d \; C(x,y,d,t). \tag{7}$$

### 5.2 Perceptual Metrics

In the following three subsections, we discuss the three terms in our cost function from Equations (3), (4), and (5).

#### 5.2.1 Perception of Linear Correlation

Pearson's coefficient and covariance: Linear correlation between two variables $x, y$ can be described via Pearson's coefficient as

$$r(x,y) = \frac{\mathrm{cov}(x,y)}{\sigma_x \sigma_y} = \frac{\mathrm{cov}(x,y)}{\sqrt{\mathrm{cov}(x,x)\,\mathrm{cov}(y,y)}}, \tag{8}$$

which is closely related to the covariance matrix

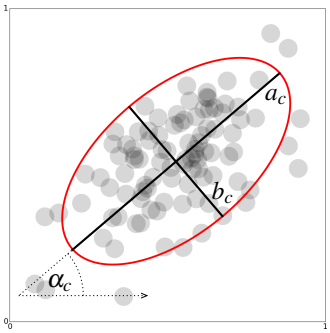
$$C(x,y) = \begin{bmatrix} \mathrm{cov}(x,x) & \mathrm{cov}(x,y) \\ \mathrm{cov}(y,x) & \mathrm{cov}(y,y) \end{bmatrix}. \tag{9}$$

$C$ is a real, symmetric matrix. Hence, it has two real eigenvalues and two orthogonal eigenvectors, which uniquely describe the shape of the *covariance ellipse*, $e_c$. The length of the major axis of the ellipse is two times the larger eigenvalue, and its direction is given by the corresponding eigenvector; the equivalent is true for the minor axis. Figure 4a shows an example. We describe the covariance ellipse as a tuple $e_c = (\alpha_c, a_c, b_c)$, where $\alpha_c$ denotes the angle of the major axis to the $x$-axis and $a_c, b_c$ the lengths of the major and minor axis, respectively.

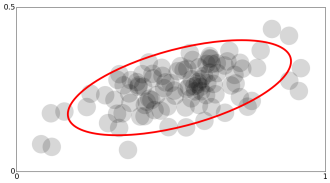
The covariance matrix and ellipse depend on the scaling of the variables. Consider scaling variable $x$ with a constant factor $s$; in this case, the covariance matrix changes as follows:

$$\begin{aligned} C(sx,y) &= \begin{bmatrix} \mathrm{cov}(sx,sx) & \mathrm{cov}(sx,y) \\ \mathrm{cov}(y,sx) & \mathrm{cov}(y,y) \end{bmatrix} \\ &= \begin{bmatrix} s^2\,\mathrm{cov}(x,x) & s\,\mathrm{cov}(x,y) \\ s\,\mathrm{cov}(y,x) & \mathrm{cov}(y,y) \end{bmatrix}. \end{aligned} \tag{10}$$

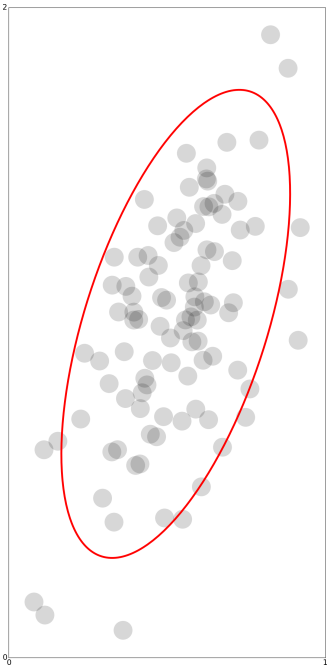
For example, changing the aspect ratio of a scatterplot essentially constitutes such a scaling operation and can dramatically alter the shape of the covariance ellipse. Figures 4b–c demonstrate this.




(a) Covariance ellipse $e_c$ (in red) can be computed from the covariance matrix. We describe it by using the angle to the $x$-axis $\alpha_c$, and the lengths of the major and minor axes: $a_c, b_c$.



(b) Data from subfigure (a) scaled in the $y$ direction with a factor of $1/2$.



(c) Data from subfigure (a) scaled in the $y$ direction with a factor of 2.

Figure 4. Covariance ellipses for a dataset that has been scaled with different factors in the $y$ direction. The shape of the ellipse depends on the scaling of the axes.

Yet Pearson's coefficient *does not* depend on the scaling of the variables.

$$r(sx,y) = \frac{\operatorname{cov}(sx,y)}{\sqrt{\operatorname{cov}(sx,sx)\operatorname{cov}(y,y)}} = \frac{s\operatorname{cov}(x,y)}{\sqrt{s^2\operatorname{cov}(x,x)\operatorname{cov}(y,y)}} = r(x,y) \tag{11}$$

We will return to this shortly.

Unit formation: To model how users perceive objects in a scatterplot, we need to predict how sparsely scattered points constitute a perceived entity. To do so, we adapt an existing model of unit formation [34]. that has been used to explain the perception of objects in dot lattices. The basic premise is that points are perceived as a single unit if they are sufficiently close and might form a Gestalt, or a shape with a contour. Objects can be identified as a unit even if they are partially occluded.

In our adaptation, we use the Canny edge detector [35] with $\sigma = 4$ *px* to detect edges in the scatterplot image, which we consider to be part of perceived object borders. The output is a set of perceived border points. If there are several classes in the data, we have a different set for each class. See the supplemental material for a justification of our $\sigma$ and several examples.

Correlation cost measures: If two variables $x, y$ are linearly correlated, then we fit an ellipse to the set of perceived border points in the scatterplot. We call this the *perceived ellipse* $e_p(\alpha_p, a_p, b_p)$. If the variables are faithfully represented by the visual design of the scatterplot, the perceived ellipse coincides with the covariance ellipse computed from the actual data. We assess the difference between ellipses $e_c$ and $e_p$ with an *angle difference* $E_\alpha$ and an *axes length ratio difference* $E_r$:

$$E_\alpha = |\alpha_c - \alpha_p| \qquad E_r = \left| \frac{b_c}{a_c} - \frac{b_p}{a_p} \right|. \tag{12}$$

As seen before in (10), the scaling of the variables is a degree of freedom and can dramatically change the shape of the covariance ellipse. To eliminate this degree of freedom and give users a fixed frame of reference, we scale the variables such that $x_{\max} - x_{\min} / y_{\max} - y_{\min} = 1$. Therefore, our error measures are minimal when $x$ and $y$ have the same amount of visual space. This scaling disregards outliers, which can be identified through, for example, the Mahalanobis distance [39]. The supplemental material presents an example for which $E_\alpha, E_r$ have been maximized and minimized.

We are aware of the recommendation by Cleveland et al. [9] and Li et al. [10] to scale the variables such that $\sigma_y/\sigma_x = 1$. This fixes the major axis of the covariance ellipse to be close to parallel to the line $y = \pm x$ and leaves the ratio $b/a$ as the sole varying aspect for judgment of the correlation. In our setting, however, we want to be able to compare the *computed-from-the-data* covariance ellipse to the *perceived-from-the-image* ellipse. The latter is not always parallel to the line $y = \pm x$, especially when the optimizer tries to find a compromise between several classes. Hence, considering the angle difference $E_\alpha$ is necessary in our application in any case. As we will show with our sub-study *CorrAR* in Section 6, an aspect ratio of $1:1$ is beneficial for correlation estimation when compared to other aspect ratios, and this is what we strive for in our method.

Note that if two variables are *not* linearly correlated, as with $|r(x,y)| < \varepsilon$, we do not compute border points by using the Canny edge filter, nor do we attempt to fit an ellipse. The cost measures $E_\alpha, E_r$ are ignored in those cases.

### 5.2.2 Image Quality Measures

We define six measures for assessing the image quality of the scatterplot. They are computed from the pixel image. We consider only those pixels that have actually received marker ink. This allows a meaningful definition of concepts such as contrast for different numbers of data points. In our implementation, we use the alpha buffer of the scatterplot for computing the image quality measures; i.e., we assess them in terms of the density of the plot. This coincides with a grayscale scatterplot. All values are in the range $[0,1]$.

Let $\mathcal{P}$ denote the set of pixels with non-zero opacity. The *average amount of opacity* per pixel and the *contrast* of the image are obtained as arithmetic mean and standard deviation, respectively:

$$I_\mu = \frac{1}{\|\mathcal{P}\|} \sum_{p \in \mathcal{P}} p \qquad I_\sigma = \sqrt{\frac{1}{\|\mathcal{P}\|} \sum_{p \in \mathcal{P}} (p - I_\mu)^2}\,. \tag{13}$$

The image is fully black (or fully opaque) if $I_\mu = 1$. The image is brighter[1] (or less opaque) as $I_\mu$ approaches 0. Similarly for the contrast, $I_\sigma$, a low value indicates low contrast and a high value indicates high contrast.

Just using $I_\mu$ and $I_\sigma$ in an optimization leads to images with extreme opacity and contrast. To obtain well-balanced images, we prescribe a desired average opacity $\bar{\mu}$ and contrast $\bar{\sigma}$ (in our implementation, $\bar{\mu} = 0.5$ and $\bar{\sigma} = 0.1$), and define

$$I_{\bar{\mu}} = |\bar{\mu} - I_\mu| \qquad I_{\bar{\sigma}} = |\bar{\sigma} - I_\sigma| \tag{14}$$

1. Note that $(1 - I_\mu)$ can be interpreted as *image brightness*.

as the distances from the desired average opacity and contrast, respectively. In essence, using positive weights for $I_{\bar{\mu}}$ and $I_{\bar{\sigma}}$ yields a well-balanced image, which can be nudged into versions with less/more opacity and less/more contrast by decreasing/increasing the weights for $I_\mu$ and $I_\sigma$.

To assess the amount by which markers overlap with each other, we denote the set of pixels covered by a *single* marker with $\mathcal{M}$. The *amount of overlap* is then computed as

$$I_\ell = 1 - \frac{\|\mathcal{P}\|}{n\,\|\mathcal{M}\|}, \tag{15}$$

where $n$ is the number of data points plotted. If no markers overlap, then $\|\mathcal{P}\| = n\,\|\mathcal{M}\|$ and $I_\ell = 0$. Increased overlap causes $I_\ell$ to approach 1. Maximal overlap would be reached when all markers are to be plotted to the same pixel location; then, $\|\mathcal{P}\| = \|\mathcal{M}\|$ and, consequently, $I_\ell = 1 - 1/n$, which is approximately 1 for the practically relevant values of $n$.

Overplotting occurs in a scatterplot at pixels where markers overlap and their opacities accumulate to a value higher than the maximum opacity (255 in our implementation). Hence, the opacity has to be clipped for these pixels – fundamentally, this means that we loose information. The *amount of overplotting* is estimated using

$$I_p = 1 - \frac{\sum_{p\in\mathcal{P}} p}{n \sum_{p\in\mathcal{M}} p} \tag{16}$$

where the nominator adds up the actual opacities in the image, and the numerator computes the sum total of all marker opacities. The supplemental material provides examples for overlap and overplotting.

#### 5.2.3 Perception of Classes and Outliers

We want to quantify how well certain groups of data points in the scatterplot are perceived. This could be relevant for any group of points, or even for individual points. Here, we concentrate on classes and outliers. As already mentioned, we know the classes and outliers before starting the optimization.

We employ *structural similarity* [36] to measure the perceivability of a group of points. Structural similarity is a highly reliable image quality assessment model often applied to measure the similarity between two images. Please consult Wang et al. [36] for details of the algorithm. Our code uses the mean structural similarity as implemented in scikit-image [40], applied to the parts of the scatterplot with non-zero opacity. In our case, for mean structural similarity $\mathrm{SSIM}(a,b)$ the results returned are between 0 and 1, where 1 denotes that images $a$ and $b$ are identical.

Consider a group of points (class or outliers) and two scatterplots $p_0, p_1$, one *without* the group of points and one *with* the group of points. We use structural similarity to measure the perceived similarity between these two scatterplots. If the scatterplots are rather similar, the group of points is difficult to perceive. If, on the other hand, the two scatterplots are rather different, then the group of points is easy to perceive.

Hence, we define the perceivability of a group of points as

$$S = 1 - \mathrm{SSIM}(p_0, p_1) \tag{17}$$

When dealing with several classes $c_i$, we compute $S$ for each class and define the total *class perceivability* on the basis of the lowest result:

$$S_c = \min(S(c_i)) \tag{18}$$

We consider outliers as one group $o$ and compute the *outlier perceivability* straightforwardly as

$$S_o = S(o). \tag{19}$$

The supplemental material provides examples with minimized and maximized perceivability of classes and outliers.

### 5.3 Combinatorial Optimization Approach

Since our primary interest lies not in efficient optimization but in assessing a proof-of-concept, we opted for *exhaustive search* as the optimization method. It is slow but guarantees finding the global optimum. A practitioner using this approach might want to choose a black box method by considering the desired trade-off in terms of speed and probability of finding the global optimum.

For exhaustive search, we needed a design space that is not so small as to be trivial, but not too large and hence practically unsolvable. Our goal was to obtain a solution in a matter of minutes with a regular laptop. Moreover, we aimed at covering a wide range of permissible values for the design parameters in order to ensure that the optimizer can consider also "radical" options.

Given these considerations, we chose the following discretization of the design variables:

- Marker size: $\mathcal{S} = [3, 5.5, 8, \ldots, 48, 50.5, 53]$,
- Marker opacities: $\mathcal{O} = [5, 17.5, 30, \ldots, 230, 242.5, 255]$,
- Aspect ratio: $\mathcal{A} = [0.5, 0.6, \ldots, 1.4, 1.5]$.

This yields a design space with $\|\mathcal{D}\| = 4,851$ possible designs. Note that we used a fixed image width. For example, most scatterplots shown in this paper have been optimized with an image width of 1,000 pixels, for obtaining images that have a reasonable resolution for publication. Addressing image width as part of the optimization would necessitate an acuity model. This would inform us about the permissible marker sizes given a set viewing distance. We leave this for future work.

Our optimization was implemented in Python with matplotlib, numpy, scipy, and parallel python. While these libraries offer significant speedups over classic Python code, the runtime is still far from that of a dedicated C++ CPU/GPU implementation. Our code evaluates around 220 designs per minute for a dataset with 10,000 data points and the selected weight set for correlation (see Table 1) on a Dell XPS 15 laptop with an Intel Core i7-4712HQ processor (4 cores).

With exhaustive search, the effectiveness of each design in the space for the given dataset and task objective is quantified by means of our cost function with specific weight factors. The visualization expert implementing the optimization framework must tune these weight factors for various tasks. To do this, we devised a visualization tool that allows the expert to change these weight factors interactively and create candidate weight sets that produce designs with different characteristics. These candidate weight sets can then be evaluated using the calibration procedure discussed in the next section.

## 6 Calibration of Weight Factors

Weights $\mathcal{W}(t)$ are specific to each data analysis task and must be calibrated accordingly. However, once a valid weight set has been identified, it can be used in varied instances of the respective task, so re-calibration is needed only upon a change in task.

We approach the calibration of weights by using both expert judgment (to set priors) and empirical data (to select best weight

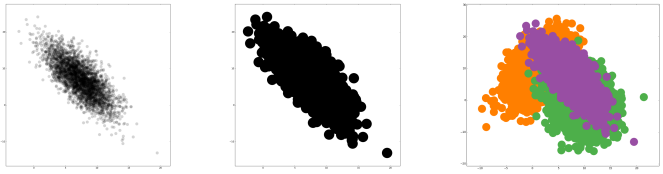


Figure 5. Plots for the same data (except that the left and middle plots show only one of the three classes in the data) optimized using the weight sets in Table 1 for different tasks: *CorrAR* and *Corr* (left), *Outl* (middle), and *Class* (right).

| Task / Sub-study | Superior Weight set | Weight Variables | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $w_\alpha$ | $w_r$ | $w_\mu$ | $w_\sigma$ | $w_{\tilde{\mu}}$ | $w_{\tilde{\sigma}}$ | $w_\ell$ | $w_p$ | $w_c$ | $w_o$ |
| CorrAR | AR 1.0 | n/a | n/a | -½ | 0 | ½ | ½ | -½ | ½ | 0 | 0 |
| Corr | RSMV2 | ½ | 1 | -½ | 0 | ½ | ½ | -½ | ½ | 0 | 0 |
| Class | CB | ½ | 1 | -½ | 0 | 0 | -½ | -½ | 0 | ½ | 0 |
| Outl | OSMV1_OP | 0 | 0 | 0 | 0 | ½ | ½ | -½ | ½ | 0 | 1 |

Table 1
The superior weight sets found in the calibration study for each task (variable names for weights are given in the text).

sets). Because the terms in our cost function are normalized, it would be tempting to use equal weights with no data. However, this disregards the fact that improvements in objectives are incommensurable and task-specific. For example, are improvements of 1% in overplotting and contrast equally valuable for, say, outlier detection? Alas, it is not known whether visualization experts can reliably predict how subtle changes in scatterplot design affect user performance. A weight must be set relative to other weights and to the cost function [41]. In the field of optimization, expert ratings, rankings, or paired comparisons are commonly used for weight selection [41]. For these reasons, we opted for an empirical weight acquisition procedure.

We designed a protocol that relies on user performance in realistic tasks and used it to calibrate weight sets for the three common data analysis tasks chosen. However, experts (co-authors of the paper), however, decided the priors: the candidate weight sets for the studies. Candidates that produce designs with different characteristics were chosen through an extensive visualization of the weight space. To ascertain the best weight set, we carried out statistical testing on objective measures of performance (judgment accuracy and task completion time).

Because weight calibration is task-specific, the calibration experiment was run separately for each task. To avoid overfitting the weights, we explored a wide range of dataset types and of numbers of data points, classes, outliers, aspect ratios, and correlation coefficients. We deemed crowdsourcing preferable to laboratory studies, because it allows a larger, more diverse and representative participant pool. However, crowdsourcing-based experiments have to be short and participant pools large if one is to achieve sufficient statistical power. Hence, only a few candidate weight sets per task could be evaluated.

### 6.1 General Method

In all, four sub-studies were conducted:

- **CorrAR**: Correlation estimation with aspect ratio only
- **Corr**: Correlation estimation without aspect ratio
- **Class**: Class separation
- **Outl**: Outlier detection

Sub-study *CorrAR* justifies the definition of our measures $E_\alpha, E_r$, which are targeted at optimization of the aspect ratio, whereas sub-study *Corr* applies these measures.

All the sub-studies used *judgment tasks* wherein the ground truth is known. A question appeared on a page, with a scatterplot on the left and text and response buttons on the right. A "no clue" button was offered also. Every sub-study contained 36 or 54 questions in total, with each plot generated by means of one of the weight set candidates.

Each participant was introduced to the task with a textual explanation and example plots. Four easy screening questions, similar to the questions in the study proper, were shown, one by one. After the subject's commitment to a response, the correct answer was shown. If more than two responses were incorrect, the participant was disallowed from continuing; this ensured that all participants understood the task.

In the experiment, questions were shown one at a time in random order. The next page was loaded after one of the buttons was clicked, or automatically after 15 seconds. Previous pages could not be visited again.

To identify inattentive participants, an easy problem that participants who succeeded in the training phase could be expected to solve (a "catch question") was assigned after every 8–9 questions. Participants were informed that an award would be given only if certain questions were answered correctly and a certain percentage of correct answers was reached. The percentage of correct answers and the number of remaining questions were shown during a break halfway through and at the end of each problem page. Data of participants who answered under 30% of the questions correctly or got any of the catch questions wrong were discarded, to ensure the validity of the data collected.

We used CrowdFlower as the crowdsourcing platform.[2] Each experiment took about 20 min, for which a worker was paid 1 USD.[3] Only English-speaking participants with the highest level of performance and reliability on CrowdFlower, who had not previously participated in our experiments, and who reported not to be color-blind were permitted to the study. All workers reported using scatterplots at least occasionally and being at least somewhat familiar with them. They had a high-school education at minimum. All of these measures were taken to ensure the validity of the crowdsourced results. At the end of a study, the participants filled in a demographics' questionnaire.

We used pseudo-random synthetic data to (1) obtain ground truth and (2) avoid effects due to semantics. Points were pseudo-randomly generated using a multivariate normal distribution with a bivariate correlation in [-1,1]. In the class separation and outlier detection sub-studies, points were evenly distributed among $k$ classes with $k \in [2,5]$. The Euclidean distance between each pair of colors in a qualitative five-data-classes color scheme from colorbrewer2.org was computed in CIELAB space, and pairs of classes with greater overlap were assigned colors with a greater distance. Other color optimization techniques could also be used (e.g., [24], [25]).

Our key dependent variables were *Success* (proportion of correct responses) and *Time* (completion time). For the correlation task, we could also compute *Error* (distance from ground truth). In all sub-studies, fewer than 5% of responses were discarded for reason of "no clue" answers and timeouts.

2. http://www.crowdflower.com; alternative platforms can be used [42]
3. consistent with similar crowdsourcing-based experiments [29]

Details about the calibration sub-studies, such as the candidate weight sets tested, the number of participants and all results, are given in the supplemental material.

Table 1 presents the weight sets selected for the optimizer, all of which were superior to the other evaluated weight sets for at least two of the dependent variables for the task in question. Different weight sets lead to different designs for the same data. Hence, some weight sets support a particular data analysis task better than others (Figure 5).

| Study | DV | mean | sig. | Post hoc comp. |
|---|---|---|---|---|
| 1. Correlation | Error | 0.19 | | T = M = R = S |
| 1. Correlation | Time | 4.57 | | T = M = R = S |
| 2. Classes | Success | 0.71 | | T = M = R = S |
| 2. Classes | Time | 6.92 | ** | S < M, S < T |
| 3. Outliers | Success | 0.78 | ** | M = R > T > S |
| 3. Outliers | Time | 3.81 | ** | T < M = R < S |

Table 2
Post hoc test results from three evaluation studies, where DV = dependent variable; **T** = the present work, **M** = MATLAB, **R** = Project R, **S** = previous study [12], and "sig." = statistical significance: ** $p < .01$

## 7 Evaluation Study

Three controlled evaluations were carried out on a crowdsourcing platform, one per data analysis task. The goal was to assess whether the optimizer can attain design quality approaching that of two plotters offered in widely used statistical packages and one presented in a previous study. We obtained several real-world datasets and compared user performance against the three baselines. Examples of the optimized plots and baseline designs are given in the supplemental material.

### 7.1 General Method

We used the same tasks (involving correlation, classes, and outliers) as in the calibration study, completing one study per task. We chose the following three baselines for comparison: two plotters in widely used statistical software – MathWorks *MATLAB* and *R: The R Project for Statistical Computing* – and plots from a study of scatterplot perception [12]. We denote these conditions as **T** (presented in this paper), **M** (MATLAB's), **R** (the R Project's), and **S** (for the earlier study's). The baselines use fixed, human-defined design parameters for all tasks. We chose these baselines to cover a wide range of such parameters.[4] Note that **S** is slightly disadvantaged, because its original width (300 px) is significantly less than that of the others. We included it in the set of baselines because it has been used in previous studies of correlation perception [11], [12].

The three studies followed a one-way within-participants design with design method (T, M, R, S) as the factor.

We obtained 3–4 realistic datasets for each task. For the correlation task, we used *Cars* [43], *Out5D* [43], *Poverty* [43], and *Shock* [44]. For the class task, we used *actg320ncc* [44], *Cars* [43], *iris* [43], and *PBC* [44], and *SenseYourCity* [45]. For the outlier task, we used *Bank* [46], *NHanes* [44], *Shock* [44]. Sizes ranged from 150 to 45,211 data points. We included real labels in the rendered plots to make the task as realistic as possible. To generate the optimized plots, we used the superior weights from the calibration study (see Table 1).

In other key respects – i.e., in the task instructions, crowdsourcing platform, measures to ensure validity of the crowdsourced results, dependent variables, and class coloring – the method followed that of the calibration sub-studies. Two breaks were given per participant. Data from participants who got any of the three catch questions incorrect was discarded.

The participants all reported not being color-blind, and the distribution of educational backgrounds was similar to that in the calibration study. Familiarity with scatterplots ranged from 2.8 to 3.0 out of 4. The samples are characterized as follows:

**Correlation**: N=127 (95 males, mean age 32.6 years), 39 countries.
**Class Separation**: N=107 (82 males, mean age 31.7 years), 44 countries.
**Outlier Detection**: N=119 (89 males, mean age 33.7 years), 35 countries.

### 7.2 Results

Table 2 provides an overview of the data and tests for the *T* condition across all three studies. The proportion of responses discarded because of "no clue" answers and timeouts was 6.2% or below in all studies.

We report the results of omnibus tests below and refer to Table 2 for post hoc comparisons. Post hoc comparisons were carried out with the Wilcoxon signed-rank test and Holm's correction.

*Correlation:* There was no statistically significant difference among the four design methods for either of the dependent variables, with all $\chi^2(3) < 4.86$ and all $p > 0.182$.

*Class Separation:* There was a statistically significant effect for time, with $\chi^2(3) = 18.40, p < 0.01$. There was no effect for success rate: $\chi^2(3) = 2.98$.

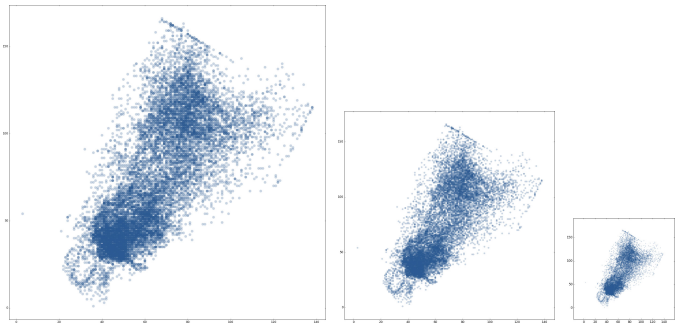
*Outlier Detection:* A statistically significant effect was seen for success rate, with $\chi^2(3) = 255.15, p < 0.01$, and for time, $\chi^2(3) = 154.75, p < 0.01$. Following widely accepted scientific practice [47], we report non-parametric statistics. However, since recent work [48] points to issues with practices in inferential statistics, we provide box plots also, in the supplemental material.

### 7.3 Summary

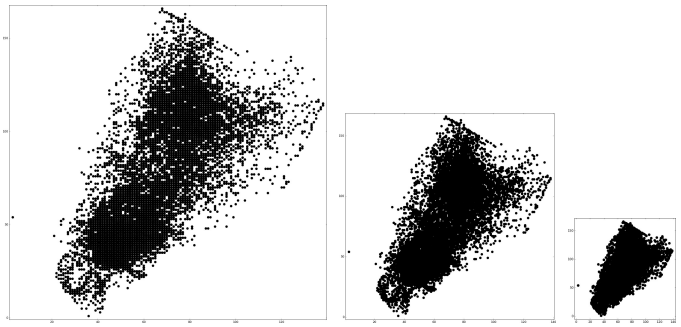
In summary, the optimizer produced plots that achieved a level of end user performance comparable to the baseline levels. For the correlation estimation task, no differences were observed among the methods. With the class separation task, plots in the S condition were linked to the fastest performance, but the methods were otherwise equal. The reverse was seen for the outlier detection task: MATLAB and R produced the best plots in terms of success rate, while our method was the best in terms of task completion time. In that task, the S method produced the worst plots.

Given that (i) some of the human-designed presets produce worse plots than our optimizer for certain tasks (e.g., S for outlier detection), (ii) the performance of our optimizer is comparable to that of human-designed presets (e.g., M, R, and S) across the three tasks considered, (iii) there is a large number of poor scatterplot designs in the design space and yet our optimizer picked up reasonably good designs solely on the basis of perceptual models, (iv) the performance of our optimizer can be improved further (see the discussion in Section 10), non-experts can already

4. Design parameters: **M**: marker size 6 px, opacity 255, aspect ratio 0.75, width 1120 px; **R**: marker size 6 px, opacity 255, aspect ratio 1, width: 480 px; **S**: marker size 2 px, opacity 255, aspect ratio 1, width: 300 px.



(a) The optimized design maintains the general appearance.



(b) The fixed design changes the general appearance.

Figure 6. The Out5d dataset with Thor-Uran scatterplots in decreasing image widths (from left to right: 1,500 pixel, 1,000 pixel, and 500 pixel).

use our optimizer with acceptable performance for any data and tested tasks. Furthermore, this is indicative of such a method's potential for creating effective and efficient designs of any other visual idiom, and it should add impetus to future work to improve on our current method and address the limitations noted through our study as discussed in Section 10.

## 8 Case Studies in Adaptability

This section examines the *adaptability* of our approach to *varying conditions*. The evaluation study focused on "one-shot design" and comparison with default settings of existing software. In real-world activities, however, input data and conditions for presentation of the results may change over time, and some plots might be needed for multiple purposes. The two cases here entail varying (1) the number of data points and (2) the image width.

Figure 1, at the start of the paper, shows scatterplots from a 2D slice of the pressure and temperature fields of the Hurricane Isabel dataset from the IEEE Visualization 2004 contest, produced by the Weather Research and Forecast (WRF) model [49]. The original data are supplied on a $500 \times 500$ grid (250,000 data points), and we lowered the resolution to $125 \times 125$ to obtain a smaller dataset that still contains comparable information. We optimized the scatterplots in Figure 1(b) by using the superior weight set from the *Corr* calibration sub-study (in Table 1). The results show that our method adjusts the marker size and opacity to compensate for the different dataset sizes without obscuring the interesting structure in the data. This stands in contrast to Figure 1(a), for which a fixed design is used, similar to what MathWorks *MATLAB* produces, which requires manual adjustment by the user. This example shows that our optimization approach is useful for obtaining insightful, structure-revealing visualizations right from the start.

Our second case, illustrated in Figure 6, involved depicting scatterplots from the *out5d* dataset with decreasing image width. The optimized plots (with the *Corr* weight set in Table 1) are able to adapt via decreased the marker size and opacity. Note how the overall appearance of the plot stays the same, while the fixed design is not able to maintain its initial appearance.

In summary, the optimizer adapted to these varying conditions in a sensible manner, whereas the default-driven approach was limited to a much narrower range of operation.

## 9 Conclusion

In this paper, we have investigated a novel optimization-based approach to scatterplot design. The contribution is to implement several models and measures of human perception in a cost function and calibrate its weight factors empirically for a given data analysis task of interest. The result is an optimizer that can decide design parameters automatically for the input task and dataset. We see great potential, especially for users who are not trained in visualization. This simplifies design activity. Instead of manual exploration of design parameters, one need only specify the objectives for an optimizer, which does the rest.

Nevertheless, perhaps the most exciting aspect of this approach is that it allows synthesizing research findings that have been scattered across the literature and make them more readily operationalized. Our present implementation includes perception models and measures to address three common data analysis tasks and three common design parameters: marker size, marker opacity, and aspect ratio. This version can already attain the level offered by widely used software for these cases. Given that there is a very large number of poor scatterplots for any given design problem, it is encouraging to observe that the algorithm was able to pick reasonable and good designs with no other instruction than perceptual models. Furthermore, we showed that this approach can intelligently adapt design parameters to changes in input data, automatically revealing patterns for which non-adaptive approaches require user intervention and technical knowledge.

## 10 Limitations and Avenues for Future Work

Future work will need to resolve the issue of *how* to balance existing and new terms in the cost function. The terms employed in the cost function should be well-balanced, such that it is easy to weigh them against each other. For example, our current definitions for $S_c, S_o$ tend to force the design towards large and opaque markers, which has proven beneficial for tasks of outlier detection and class separation, respectively, but makes it difficult to find compromises with other tasks. Similar issues have been noted previously in heuristic optimization of layouts [32].

We should also try to expand the set of supported data analysis tasks. *Density estimation* is especially interesting, since it may benefit from semi-transparent, medium-sized markers; this is a state that our system balances especially well against conflicting constraints. Furthermore, to support differences in user abilities, we could incorporate color models addressing dichromacy or monochromacy. Displays with a low dynamic range could be supported by adapting the design space. For expansion of the design space, we highlight two design parameters: the color of the classes and their rendering order. Both influence perceivability. However, the inclusion of color as a new design parameter should be accompanied by new terms for assessing, for instance, color contrast, in future work. We also plan to extend this work to visual idioms other than scatterplots.

## ACKNOWLEDGMENTS

Luana Micallef was funded by the Academy of Finland, Centre of Excellence in Computational Inference Research (COIN), and grant agreement 305780. Gregorio Palmas and Tino Weinkauf received partial funding through the SkAT-VG project, funded by the EC under FP7-ICT-2013-C, Future Emerging Technologies, grant agreement 618067. Antti Oulasvirta was funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement 637991).

## REFERENCES

[1] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner, "Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences," in *Proc. BELIV*. ACM, 2014, pp. 1–8.

[2] M. Sedlmair, A. Tatu, T. Munzner, and M. Tory, "A taxonomy of visual cluster separation factors," *CGF*, vol. 31, no. 3pt4, pp. 1335–1344, 2012.

[3] M. Tory and T. Möller, "Human factors in visualization research," *IEEE TVCG*, vol. 10, no. 1, pp. 72–84, 2004.

[4] H. Wickham, "A layered grammar of graphics," *Journal of Computational and Graphical Statistics*, vol. 19, no. 1, pp. 3–28, 2010.

[5] L. Wilkinson, A. Anand, and R. L. Grossman, "Graph-theoretic scagnostics," in *Proc. InfoVis*, 2005, pp. 157–164.

[6] E. Bertini, A. Tatu, and D. Keim, "Quality metrics in high-dimensional data visualization: An overview and systematization," *IEEE TVCG*, vol. 17, no. 12, pp. 2203–2212, 2011.

[7] T. Munzner, *Visualization Analysis and Design*. Boca Raton, FL, USA: CRC Press, 2014.

[8] J. Meyer, M. Taieb, and I. Flascher, "Correlation estimates as perceptual judgments." *J. Exp. Psychol.: Applied*, vol. 3, no. 1, p. 3, 1997.

[9] W. S. Cleveland, P. Diaconis, and R. McGill, "Variables on scatterplots look more highly correlated when the scales are increased," *Science*, vol. 216, no. 4550, pp. 1138–1141, 1982.

[10] J. Li, J.-B. Martens, and J. J. van Wijk, "Judging correlation from scatterplots and parallel coordinate plots," *Information Visualization*, vol. 9, no. 1, pp. 13–30, 2010.

[11] L. Harrison, F. Yang, S. Franconeri, and R. Chang, "Ranking visualizations of correlation using weber's law," *IEEE TVCG*, vol. 20, no. 12, pp. 1943–1952, 2014.

[12] R. A. Rensink and G. Baldridge, "The perception of correlation in scatterplots," *CGF*, vol. 29, no. 3, pp. 1203–1210, 2010.

[13] M. Kay and J. Heer, "Beyond Weber's law: A second look at ranking visualizations of correlation," *IEEE TVCG*, vol. 22, no. 1, pp. 469–478, 2016.

[14] M. Last and A. Kandel, "Automated detection of outliers in real-world data," in *Proc. Intelligent Technologies*, 2001, pp. 292–301.

[15] S. K. Rahman, M. M. Sathik, and K. S. Kannan, "Multiple linear regression models in outlier detection," *International Journal of Research in Computer Science*, vol. 2, no. 2, p. 23, 2012.

[16] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[17] M. Sedlmair, T. Munzner, and M. Tory, "Empirical guidance on scatterplot and dimension reduction technique choices," *IEEE TVCG*, vol. 19, no. 12, pp. 2634–2643, 2013.

[18] M. Gleicher, M. Correll, C. Nothelfer, and S. Franconeri, "Perception of average value in multiclass scatterplots," *IEEE TVCG*, vol. 19, no. 12, pp. 2316–2325, 2013.

[19] J. M. Lewis, M. Ackerman, and V. De Sa, "Human cluster evaluation and formal quality measures: A comparative study," in *Proc. Cognitive Science Society (CogSci)*, 2012, pp. 1870–1875.

[20] M. Sedlmair and M. Aupetit, "Data-driven evaluation of visual quality measures," *CGF*, vol. 34, no. 3, pp. 201–210, 2015.

[21] M. Aupetit and M. Sedlmair, "Sepme: 2002 new visual separation measures," in *IEEE PacificVis*, 2016, pp. 1–8.

[22] J. Schneidewind, M. Sips, and D. A. Keim, "Pixnostics: Towards measuring the value of visualization," in *IEEE VAST 2006*. IEEE, 2006, pp. 199–206.

[23] R. Etemadpour, R. Motta, J. G. de Souza Paiva, R. Minghim, F. de Oliveira, M. Cristina, and L. Linsen, "Perception-based evaluation of projection methods for multidimensional data visualization," *IEEE TVCG*, vol. 21, no. 1, pp. 81–94, 2015.

[24] H. Chen, W. Chen, H. Mei, Z. Liu, K. Zhou, W. Chen, W. Gu, and K.-L. Ma, "Visual abstraction and exploration of multi-class scatterplots," *IEEE TVCG*, vol. 20, no. 12, pp. 1683–1692, 2014.

[25] A. Mayorga and M. Gleicher, "Splatterplots: Overcoming overdraw in scatter plots," *IEEE TVCG*, vol. 19, no. 9, pp. 1526–1538, 2013.

[26] S. Johansson and J. Johansson, "Interactive dimensionality reduction through user-defined combinations of quality metrics," *IEEE TVCG*, vol. 15, no. 6, pp. 993–1000, 2009.

[27] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. Magnor, and D. Keim, "Combining automated analysis and visualization techniques for effective exploration of high-dimensional data," in *IEEE VAST 2009*. IEEE, 2009, pp. 59–66.

[28] J. Matejka, F. Anderson, and G. Fitzmaurice, "Dynamic opacity optimization for scatter plots," in *Proc. ACM Human Factors in Computing Systems*. ACM, 2015, pp. 2707–2710.

[29] J. Heer and M. Bostock, "Crowdsourcing graphical perception: using mechanical turk to assess visualization design," in *Proc. ACM CHI*, 2010, pp. 203–212.

[30] A. Oulasvirta, "User interface design with combinatorial optimization," *Computer*, vol. 50, no. 1, pp. 40–47, 2017.

[31] G. Bailly, A. Oulasvirta, T. Kötzing, and S. Hoppe, "Menuoptimizer: Interactive optimization of menu systems," in *Proc. UIST*. ACM, 2013, pp. 331–342.

[32] P. O'Donovan, A. Agarwala, and A. Hertzmann, "Learning layouts for single-pagegraphic designs," *IEEE TVCG*, vol. 20, no. 8, pp. 1200–1213, 2014.

[33] M. Reddy, "Perceptually optimized 3D graphics," *IEEE CGA*, vol. 21, no. 5, pp. 68–75, 2001.

[34] P. J. Kellman and T. F. Shipley, "A theory of visual interpolation in object perception," *Cognitive psychology*, vol. 23, no. 2, pp. 141–221, 1991.

[35] J. Canny, "A computational approach to edge detection," *IEEE TPAMI*, no. 6, pp. 679–698, 1986.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[37] G. Zapfel, M. Bogl, and R. Braune, *Metaheuristic search concepts*. Springer, 2010.

[38] M. Dorigo, G. Di Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," *Artificial life*, vol. 5, no. 2, pp. 137–172, 1999.

[39] P. C. Mahalanobis, "On the generalized distance in statistics," *Proc. of the National Institute of Sciences (Calcutta)*, vol. 2, pp. 49–55, 1936.

[40] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, 6 2014.

[41] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: new insights," *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, pp. 853–862, 2010.

[42] E. Peer, S. Samat, L. Brandimarte, and A. Acquisti, "Beyond the turk: An empirical comparison of alternative platforms for online behavioral research," *Available at SSRN 2594183*, pp. 1–30, 2015.

[43] XmdvTool homepage, http://davis.wpi.edu/xmdv/datasets.html.

[44] Statistical Software Information, University of Massachusetts Amherst, http://www.umass.edu/statdata/statdata/data/.

[45] Data Canvas homepage, http://map.datacanvas.org/#!/data.

[46] UC Irvine Machine Learning Repository, http://mlr.cs.umass.edu/ml/index.html.

[47] C. Jacob, "The earth is round (p<. 05)," *American Psychologist*, pp. 997–1003, 1994.

[48] P. Dragicevic, "Fair statistical communication in HCI," in *Modern Statistical Methods for HCI*. Springer, 2016, pp. 291–330.

[49] IEEE Visualization Contest 2004, http://vis.computer.org/vis2004contest/.

**Luana Micallef** is a Postdoctoral Researcher at the Helsinki Institute for Information Technology, Aalto University and an Honorary Research Fellow at the University of Kent. She completed her Ph.D. in computer science in 2013 from the University of Kent, and her thesis a 2016 CPHC/BCS Academy of Computing Distinguished Dissertation runner-up. Her algorithm eulerAPE has acclaimed international recognition with over 100 citations in two years. Her research focuses on information visualization, visual analytics and human-computer interaction, particularly set visualization and design/layout optimization. Currently she uses techniques in these domains and also cognitive science and artificial intelligence to solve big data problems in medicine and other application areas.

**Gregorio Palmas** received his master's degree in computer science in 2012 from the University of Pisa in collaboration with the Visual Computing Group of the ISTI - CNR research institute. From 2012 he started working at the Max Planck Institute for Informatics on the visual analysis of multidimensional data for biomechanics and human-computer interaction. He received his Ph.D. in computer science in 2016 from KTH Royal Institute of Technology, Stockholm.

**Antti Oulasvirta** is an Associate Professor at Aalto University where he leads the User Interfaces research group. He was previously a Senior Researcher at the Max Planck Institute for Informatics and the Cluster of Excellence on Multimodal Computing and Interaction at Saarland university. He received his doctorate in Cognitive Science from the University of Helsinki in 2006, after which he was a Fulbright Scholar at the School of Information in University of California-Berkeley in 2007-2008 and a Senior Researcher at Helsinki Institute for Information Technology HIIT in 2008-2011. He was awarded the ERC Starting Grant (2015-2020) for research on computational design of user interfaces.

**Tino Weinkauf** received his diploma in computer science from the University of Rostock in 2000. From 2001, he worked on feature-based flow visualization and topological data analysis at Zuse Institute Berlin. He received his Ph.D. in computer science from the University of Magdeburg in 2008. In 2009 and 2010, he worked as a postdoc and adjunct assistant professor at the Courant Institute of Mathematical Sciences at New York University. He started his own group in 2011 on Feature-Based Data Analysis in the Max Planck Center for Visual Computing and Communication, Saarbrücken. Since 2015, he holds the Chair of Visualization at KTH Royal Institute of Technology, Stockholm. His current research interests focus on flow analysis, discrete topological methods, and information visualization.