PFunk-H: Approximate Query Processing using Perceptual Models

ABSTRACT

Interactive visualization tools (e.g., crossfilter) have become critical to many data analysts. These tools make the discovery and verification of hypotheses quick and seamless. However, the increasing amounts of data available for analyses these days has made the scalability of these tools a necessity. To bridge the gap between data sizes and interactivity, many visualization systems have turned to sampling-based approximate query processing frameworks. However, these systems are currently oblivious to human perceptual visual accuracy. Both for correctness and efficiency, we can use empirical knowledge of human perceptual limitations to bound the error of approximate answers meant for visualization.

In this paper, we explore a preliminary model of sampling-based approximate query processing that uses perceptual models (encoded as functions) to construct approximate answers intended for visualization. We present initial results that show that (with a very high probability) the approximate and non-approximate answers for a given query differ by a *perceptually indiscernible* amount, determined by perceptual functions.

1. INTRODUCTION

Dynamic and interactive visualization tools (e.g., crossfilter [**?**]) make it easier for data analysts to discover and verify hypotheses seamlessly. However, with the increasing amounts of data available for analyses these days is the accompanying need for scalability of these interactive tools. High latency in visualization tools make the use of such tools less immersive and even sometimes frustrating. To bridge the gap between data sizes and interactivity, many modern visualization systems have turned to approximation [**?**]. Approximation can be used to trade off computational cost for query answer accuracy.

Many existing systems employ some form of approximation. Sampling-based approximate query processing (AQP) systems like BlinkDB, Aqua, and IFocus can evaluate a query on a subset of the underlying dataset [6, 2, 1]. The larger the sample size used during query processing, the better — but slower — the approximation. Sometimes, the accuracy of the approximation is determined by certain formal visual guarantees. For example, IFocus [6], an online sampling algorithm, selects samples until — with a very high probability — the visual property of correct ordering is satisfied, whereby groups or bars in a visualization are ordered correctly. On the other hand, the accuracy of the approximation procedure can be determined by resource availability. For example, BlinkDB [2] can use a user-specified time limit to determine the optimal sample size. The difference in the provided guarantees by IFocus and BlinkDB highlights the trade-off between the accuracy and computational cost of an approximation.

Common to many sampling-based AQP systems are global tuning parameters or thresholds that can be used to determine when sampling stops. For instance, IFocus relies on a minimum resolution parameter r below which the visual ordering property does not have to be satisfied [6]. [Another example: BlinkDB] Although the paper on IFocus does not specify how r should be set, this parameter is perceptually motivated as the perceptual difference between visual objects becomes indiscernible below a certain threshold. As a simple example, the difference between bar heights of 90 and 91 pixels is perceptually indiscernible in most situations. As a result, we can set the minimal resolution to 1 pixel. Similarly, in most situations, users cannot distinguish between small shades of red [**?**]. In fact, minimal thresholds for perceptual indiscernibility of visual objects apply in any situation where information can be encoded graphically.

[(i)We believe there's a lot of promise. (ii)Lot of work on this — people know how and have run these studies. Examples: Tasks classified (iii) Humans have limited perceptual abilities.] The field of graphical perception explores how humans decode information on data visualizations. Literature on graphical perception contains extensive studies to measure user perception and judgement accuracies in a multitude of settings [10, 9, 32, 4, 14, 25]. We can rely on the foundational work on graphical perception to construct perceptual models that can be used to produce approximate answers meant for visualization. For instance, Cleveland et al. [10] identified a set of elementary perceptual tasks that are carried out when people extract quantitative information from graphs and ordered the tasks on the basis of how accurately people perform them. In section 2, we present an initial construction of perceptual models that can be used in sampling-based approximate query processing.[Sequence: Perceptual limitations of humans → HCI have done studies

HILDA '16 June 26, 2016, San Francisco, CA, USA

DOI: 10.1145/1235

(for example, Cleveland's) $\rightarrow$ The key finding we draw upon is that there is a significant variability due to a large number of factors $\rightarrow$ We hope we can leverage in this paper for performance related to sampling]

In this paper, we present an online sampling algorithm that can use knowledge of human perceptual error (encoded in perceptual functions) to provide approximate answers that differ from the true answers by a perceptually indiscernible amount. The algorithm, PFunk-H, uses perceptual functions to automatically determine the confidence interval for approximate answers. Since the width of the confidence interval determines both the efficiency and the correctness of query approximation, we would like to be able to automatically determine the width of the interval such that the interval is lax enough so that we do not consume too many samples but strict enough to ensure correctness. For PFunk-H, the accuracy of the resulting approximation depends on the univariate perceptual function used. More conservative functions will produce stricter confidence intervals than less conservative ones.

The introduction of PFunk-H and our preliminary experiments is an initial foray into the use of perceptual models in approximate query processing. Perceptual models can be used in a multitude of settings, some of which we outline in section **??**. The gains of using perceptual models during query approximation is accentuated in highly interactive visualizations. Such visualizations generate bursts of highly correlated queries that produces results that are ultimately perceived by humans [36].

Prior work suggested using this, but we have concrete instantiation of a possible st of pfunks. (STRONGER).

We now begin with a quick primer on graphical perception and our simplified model for perception in data visualizations.

2. GRAPHICAL PERCEPTION AND SIMPLE PERCEPTUAL MODELS

This section gives an overview of graphical perception research and introduces the perceptual models used in this paper.

The broad perception literature has studied and developed theories for the factors that affect human perception. Within this literature is the rich area of graphical perception, which studies how accurately humans can perform visual judgements in data visualizations. For instance, the power law of psychophysics [32, 24] models the perceived magnitude of the encoded value using an exponential relationship with the true value. Beyond general models, research have performed studies specific for visual encodings, types of judgement tasks, as well as type of visualization. Color perception has been extensively studied and there are numerous proposed color spaces [29, 28] designed to be perceptually uniform such that perceived color differences uniformly translate to differences of the encoded values. Similarly, Cleveland et al. [9] measure perceptual error of proportional comparison tasks between pairs of visually encoded values (e.g., bar heights, or slopes of lines). Furthermore, recent work has extended graphical perception studies to animated data visualizations [35].

These studies provide evidence of substantial variability of human perceptual accuracy that is affected by the data values, the visual encoding, the judgement task, and many other factors. The simple perceptual functions introduced in this section are meant as a first step to succinctly capture this variability in a manner than can be used by a visualization system for e.g., optimization.

2.1 Framework for Perceptual Functions

The general form of a perceptual function is as follows: [1]

$$P(v_1^*, \dots, v_n^* | C) = \epsilon \qquad (1)$$

where $v_1^*, \dots, v_n^*$ are the non-approximate values to be encoded visually. C encompasses the possible contexts for which this model is valid; these can include the specific user, the time of day, the user's mental fatigue, and other possible settings. ϵ specifies the maximum allowable deviation of the true values from their approximated counterparts.

The simplest case of a perceptual function is the univariate case where we assume the values to be encoded are independent of one another, a reduction to the $n = 1$ case in equation 1. We make use of univariate perceptual functions $P^{(u)} : \mathbb{R} \rightarrow \mathbb{R}$ in this paper (the context C is not explicitly specified here). $P^{(u)}$ maps a single visually encoded value to the maximal perceived error. Consider a simple linear univariate function $P^{(u)}(v) = 5 \cdot 10^{-5} v$. $P^{(u)}(10^5) = 5$ could be interpreted as a user perceiving an encoded value of 10^5 within a margin of error of ± 5.

We have been running several classes of perceptual experiments for use in fitting perceptual functions. For a specific class of experiments, we assume that ϵ, the perceptual margin of error, is normally distributed with an unknown mean and variance. Measurement errors are often modeled as gaussian noise [25]. Users were asked to estimate the height of a specific bar in a bar chart. Then for each answer, we calculate the absolute difference between the user's estimate and the true bar height. As an example, suppose that a user is asked to estimate the height of a bar of height 100 in some scenarios. After 20 tasks for the user, we aggregate error values and calculate the sample mean and std values as 5 and 2 respectively. Then a 99% lower bound for ϵ_{100} (perceptual error when estimating bar height 100) is 3.96 so we set $P^{(u)}(100) = 3.96$. In this manner, we can aggregate lower bounds for ϵ and then fit these values to a function that gives us the least square error. Figure 1c shows log-fitted univariate perceptual functions with encoded truth values normalized to [0, 1]. $R-90, R-99$, and $R-99.9$ use the 90th, 99th, and 99.9th lower bounds for the perceptual errors obtained for each possible bar height.

2.2 A Library of Perceptual Functions

It is important to recognize that prior studies also suggest that user accuracy is context sensitive. For instance, a model fit to a data from a low contrast setting may not apply when the contrast is high [17]; the accuracy for estimating the heights of tall bars and short bars are different; and fatigued users may not be as accurate as alert users [23]. For this reason, we imagine a growing library of perceptual models collected under different contextual settings (e.g., tasks, users, lighting, etc). With such a library, the system may pick the most applicable model for a given user based on the user's own perceptual information, as well as her environment. Over time, we hope this library can provide more

[1] similar notation is used to specify conditional probability distributions

accurate accuracy estimates and be useful for larger scale perceptual research. Along these lines, methods to scale the collection of context-sensitive perceptual models, and identifying models that are robust to different contexts are promising directions of future work.

3. THE ALGORITHM AND ITS ANALYSIS

In this section, we present PFunk-H. This algorithm supports approximate answers for `SUM` and `COUNT` based aggregation functions and we present the version for the `AVG` aggregate function. For ease of description, we have stripped the algorithm to the simplest form possible — we present very simple and conservative concentration inequalities, and we assume a monotonically increasing perceptual function and that each record is a single numerical value within $[0, 1]$. In practice, our implementation of PFunk-H supports `SELECT-PROJECT-GROUPBY` queries over full records with bounded numerical ranges beyond $[0, 1]$, and we use stronger inequalities. Further extensions not described in this paper extend the algorithm to non-monotonic perceptual functions, as well as bi-variate functions.

3.1 The PFunk-H Algorithm

PFunk-H uses a single perceptual function to return an approximate average of the dataset $\mathbf{X}(= \{x_1, \ldots, x_N\})$. Suppose μ is the population mean. At the end of the algorithm, we want to estimate with high probability a sample mean v such that

$$v \in [\mu - P^{(u)}(\mu), \mu + P^{(u)}(\mu)] \tag{2}$$

Equation 2 is a guarantee that the margin of perceptual error is $\leq P^{(u)}(\mu)$ with probability $\geq (1 - \delta)$. In other words, with a high probability, the final approximation error is not perceptually discernible by humans, as defined by the perceptual function $P^{(u)}$. Table 1 describes the symbols used in Algorithm 1.

Algorithm 1: Basic Univariate PFunk-H Algorithm

Data: $P^{(u)}, \mathbf{X}, \delta$

1 Initialize $s = 0, v = R(\mathbf{X}), t = P^{(u)}(v)$;

2 **while** $t > P^{(u)}(v - t)$ *and* $s < N$ **do**

3 $\quad s = s + 1$;

4 $\quad t = \sqrt{\frac{\log(2/\delta)}{2s}}$;

5 $\quad v = \frac{s-1}{s} \cdot v + \frac{1}{s} \cdot R(\mathbf{X})$;

6 **return** v;

Algorithm 1 depicts the iterative algorithm in detail. s represents the total number of samples required for a margin of error of t. On each iteration, we sample without replacement one more element from the dataset using $R(\mathbf{X})$. Using Hoeffding's inequality [**?**], we can compute the decreased margin of error t. We continue this procedure until $t \leq P^{(u)}(v - t)$, when the number of samples ensures that equation 2 is satisfied. The challenge is that the perceptual function $P^{(u)}$ is conditiioned on the true value μ, and we outline the proof in Subsection 3.1.1.

EXAMPLE 1. *We now provide an example execution sequence of algorithm 1. Suppose the dataset is generated from a normal distribution $x_1, \ldots, x_N \sim \mathcal{N}(0.2, 1)$ where*

Var	Description
$\mathbf{X}$	Dataset $0 \leq X_0, \ldots, X_n \leq 1$
$R(\mathbf{X})$	Proc to return a random without-replacement sample from $\mathbf{X}$
$P^{(u)}$	Univariate perceptual function
δ	Chance of failure of algorithm (default: 0.05)
Δ	Step size for sample set size

Table 1: Table of Notation for the PFunk-H Algorithms

$N = 1$ million points, and the perceptual function is linear $P^{(u)}(x) = \frac{x}{10}$. The maximum allowable perceptual error when computing the `AVG` *is defined as $P^{(u)}(\mu = 0.2) = 0.02$, however we do not know the value of μ and must learn it as part of the algorithm such that, by the time the algorithm terminates, the confidence interval for the sample mean v should be a sub-interval of $[\mu \pm P^{(u)}(\mu)] = [0.2 \pm 0.02]$.*

The conditional in Line 2 compares the empirical margin of error t with the output of the perceptual function over the sample mean minus the margin of error $v - t$. After taking 5867 samples we find that $v = 0.21$, and $t = 0.018$ is less than the true perceptual error 0.02 *and ensures that the approximated result is not perceptually discernible.*

Note that the goal is to only use the minimum number of samples such that the empirical margine of error is, with high probability, less than 0.02.

To extend this algorithm to records with multiple attributes, we simple need to keep track of the margins of error t_{attr} for each attribute *attr*. In addition, the error bound computed in Line 4 can be replaced by using stronger inequalities. For instance, the current (loose) bound is uses Hoeffding's inequality [19] which restricts the values to $[0, 1]$. Alternatives such as Hoeffding-Serfling[5] reduce the margin of error

the current algorithm assumes that all values ar within $[0, 1]$ to simplify the proof, however it is simple to extend to handle values within an fixed numerical range $[a, b]$ where $a < b$[2]. In addition, making distributional assumptions about the dataset would allow us to compute the empirical variance and further reduce the bounds. For example, we have found that assuming that the attribute values come from a gaussian distribution can reduce the number of samples by 10× or more to reach the same error bound.

3.1.1 Proof of Correctness

We now illustrate a proof of correctness for Algorithm 1.

CLAIM 1. *Assuming $P^{(u)}$ is a monotonically non-decreasing function, at the end of the algorithm, $\Pr[P^{(u)}(\mu) > t] > 1 - \delta$ where t is as defined in the algorithm 1.*

PROOF. We prove the contrapositive $\Pr[P^{(u)}(\mu) \leq t] \leq \delta$. We use the one-sided case of Hoeffding's classical inequality. By theorem 1 in [19], $\Pr[v - \mu \geq t] \leq e^{-2st^2}$ where s is the number of samples used in calculating the sample average v, and equivalently, $\Pr[\mu \leq v - t] \leq e^{-2st^2}$. Using

[2]By replacing Line 4 with $t = \sqrt{\frac{\rho_s \log(2/\delta)}{2s}}$ where

$$\rho_s = \left\{ \begin{array}{ll} \left(1 - \frac{s-1}{N}\right), & \text{for } s \leq N/2 \\ \left(1 - \frac{s}{N}\right)\left(1 + \frac{1}{s}\right), & \text{for } s > N/2 \end{array} \right\}$$

the monotonicity assumption for $P^{(u)}$,

$$\Pr[P^{(u)}(\mu) \leq P^{(u)}(v-t)] \leq e^{-2st^2} \quad (3)$$

Solving $e^{-2st^2} \leq \delta$ obtains a lower bound for s, where $s \geq \frac{\log(1/\delta)}{2t^2}$. When the algorithm terminates, $s = \lceil \frac{\log(2/\delta)}{2t^2} \rceil$.

$\Pr[P^{(u)}(\mu) \leq P^{(u)}(v-t)] \leq \delta$ is equivalent to $\Pr[P^{(u)}(\mu) > P^{(u)}(v-t)] > 1-\delta$, while the stopping condition ensures that $P^{(u)}(v-t) \geq t$. Thus, $\Pr[P^{(u)}(\mu) > t] > 1-\delta$.
□

CLAIM 2. *If $s < N$, then $v \in [\mu - P^{(u)}(\mu), \mu + P^{(u)}(\mu)]$*

PROOF. In claim 1, we invoked the one-sided case of Hoeffding's inequality[19]. For the two sided case, $\geq \frac{\log(2/\delta)}{2t^2}$ samples are needed so that $v \in [\mu - t, \mu + t]$. By claim 1, with probability $(1-\delta)$, $P^{(u)}(\mu) > t$. As a result,

$$[\mu - t, \mu + t] \subseteq [\mu - P^{(u)}(\mu), \mu + P^{(u)}(\mu)] \quad (4)$$

Therefore, $v \in [\mu - P^{(u)}(\mu), \mu + P^{(u)}(\mu)]$ with probability $\geq (1-\alpha)$. □

4. EXPERIMENTS

In this section, we present univariate perceptual functions fitted on data collected from perceptual experiments and on synthetic datasets. Then we show that the margin of errors produced by PFunk-H closely matches the perceptual function used. This means that when used with more conservative perceptual functions, PFunk-H is more computationally intensive than when used with less conservative functions. Furthermore, for each perceptual function used, we show results that indicate that the sampling complexity varies by the chance of failures (δ). The sampling complexity is inversely proportional to δ.

4.1 Setup

We ran all experiments on a 8-core Intel(R) Xeon(R) CPU E5-2680 2.80GHz server running Ubuntu 12.04.3 LTS. However, all experiments were single-threaded to avoid speedup from parallelization.

PFunk-H is evaluated on top of NEEDLETAIL, a database system designed to sample records matching a set of adhoc query predicate conditions [26]. NEEDLETAIL uses inmemory bitmap-based indexes to quickly retrieve satisfying tuples.

For our experiments, we generated synthetic datasets with 10 million records, each containing 20 continuous columns and 2 discrete columns. Each continuous field is drawn from a (truncated) normal distribution with mean $\mu \in [0.1, 0.8]$ and std $\sigma \in [0.1, 0.5]$. Our preliminary experiments focused on the continuous fields.

We show results for six perceptual functions, three synthetic and the other three based on perceptual experiments performed on human subjects. The synthetic functions are shown in figure 1a. These functions are non-linear functions. $S-0, S-1$ and $S-2$ are equivalent to quadratic functions $s_0(x) = 0.005x^2 + 0.01$, $s_1(x) = 0.02x^2 + 0.01$, and $s_2(x) = 0.05x^2 + 0.01$.

The functions shown in figure 1c are based on perceptual experiments... true and est values fps = 10 and maxpos = 50% and linear rate of change is ... and then this stuff happens. .... $R-90$, $R-99$, and $R-99.9$ are equivalent to the linear functions $r_{90}(x) = 0.04165x + 0.005687$, $r_{99}(x) = 0.03061x + 0.004389$, and $r_{99.9}(x) = 0.02254x + 0.00344$. R-90, R-99, and R-99.9 use the 90th, 99th, and 99.9th percentiles of the sample perceptual errors aggregated from the experiments. See 2 for information on why and how we model the functions in this manner.

A natural question to ask is: how close is the estimated margin of error (variable t produced in algorithm 1) to $P^{(u)}(\mu)$ (allowable perceptual error for true mean μ)? Recall, that since we do not know μ before PFunk-H runs, we also do not know $P^{(u)}(\mu)$. We used PFunk-H with function $S-2$ on different columns with means μ ranging from 0.1 to 0.7. Figure 2 shows the original function $S-2$ (ground truth) and estimated margin of errors $S-2-est$ for the query answers produced by PFunk-H. As expected (and already proven in section 3), $P^{(u)}(\mu) > t$ which makes $S-2$ strictly greater than but close to $S-2-est$.

4.2 Comparing the Perceptual Functions

There are various experimental results we can show: the empirical sampling complexity of the perceptual function versus δ, the probability of failure of the algorithm; δ versus runtime; runtime versus the sampling complexity, and so on. Because of the space limitations, we show how the sampling complexity varies with δ and the univariate function used by PFunk-H.

Figures **??** and **??** show how the number of samples chosen by PFunk-H varies with the failure probability of the algorithm δ. The dataset is drawn from a truncnorm distribution with mean 0.45 and std 0.48. Recall that the δ parameter in algorithm 1 is used to specify the chance of failure of the sampling algorithm. Since the failure probability δ is directly related to the sampling complexity (see algorithm 1), we can expect δ to correlate with the percentage of the dataset sampled. Indeed, it is. Figures **??** and **??** show that (for all six perceptual functions), as the failure probability increases, the number of samples chosen decreases. This makes sense since $\delta \to 0$ means that the chance of failure of the algorithm would approach 0 and would imply that the entire dataset should be used to calculate a non-approximate answer; $\delta \to 1/2$ would mean that our answer could be an incorrect approximation at least half the time.

Figures **??** and **??** also show the relative sampling complexity of the six perceptual functions. As expected, for a constant δ, R-99.9 samples more than R-99 and R-99 samples more than R-90. For the synthetic squared functions, S-0 samples more than S-1 and S1 samples more than S-2. The ordering of the functions in **??** match functions R-99.9 samples more than both R-90 and R-99 P2, P3 and P3 samples less than P1, P2. This result follows the (reverse) relative ordering of the functions shown in figures 1a and 1c.

5. RELATED WORK

The work related to PFunk-H can be placed in a few categories:

Graphical Perception: One of the first studies to formalize visual perceptual accuracy was done by Cleveland & McGill [10]. The field of graphical perception is mainly focused on the visual decoding of information encoded on graphs. Cleveland & McGill ran experiments to quantify the accuracy of pairwise comparison of static objects in data visualizations. The goals of the experiments were two-fold: identify a set of elementary perceptual tasks implicitly performed by humans when extracting quantitative information from graphical models; and – most related to this paper – to order the tasks on the basis of how accurately people perform them [10, 9]. This ordering is based on experiments and on theory in psycophysics. The power law of theoretical psycophysics, which has been shown to be realistic [4], relates the perceived magnitude of a stimuli to the actual magnitude[32]. Therefore, much ground work has been done toward formalizing the theory and experimentation of human visual perceptual accuracy [32, 4, 14, 25].

A range of experiments have been performed to further validate Cleveland & McGill's foundational work on graphical perception. Talbot et al. [31] extend the original experiment to study the effects of distractor bars and separation between non-adjacent bar charts. Heer et al. [17] approximately replicate the original experiment using crowdsourcing and a larger collection of chart types. Zacks et al. [37] evaluates the same perceptual task but on 3D charts.

A concept similar to perceptual functions has been introduced by Demiralp et al.[12]. They formulate perceptual kernels, a distance metric of aggregate pairwise perceptual distances used to measure how color, shape, and size affect graphical perception. But it is not immediately obvious (based on its formal definition) how to use perceptual kernels in sampling-based AQP systems. On the other hand, our initial definition of perceptual functions aim to provide error metrics that can be used in a sampling-based algorithm like PFunk-H. The definition of perceptual functions makes it especially conducive for online algorithms.

Last, weber models (both linear and log-linear) have been formulated to provide concise means to model the human perception of differences in the correlation of objects in graphical models [16, 24].

Approximate Query Processing: There are two broad categories of algorithms for approximate query processing: online and offline. We consider related work for both categories.

Online AQP is most closely related to our work presented in this paper because PFunk-H is an online algorithm. The main idea behind online aggregation [18] is the interactive and on-the-fly construction and refinement of confidence intervals for estimates of aggregate functions (AVG, SUM, and so on). Online aggregation is particularly appealing for the interactive query processing functionality it provides. Users are provided with an interactive tool that can be used to stop processing aggregate groups when the users deem fit. Thus, the onus rests on the user to the approximate processing. This interface, although very intuitive for users that have knowledge of statistics, could be daunting for lay users with no background in math. On the other hand, PFunk-H stops processing automatically when the query answers intended for visualization are perceptually indiscernible from the true answers. In this manner, our answers are computed approximately, yet satisfy formal visual guarantees. IFocus [6] is another online sampling algorithm which provides a formal *probabilistic* guarantee different from what PFunk-H gives: the visual property of correct ordering, whereby groups or bars in a visualization are ordered correctly.

As per offline AQP, Garofalakis et al. [13] surveys the area. Examples of systems that support offline AQP inlude BlinkDB [2] and Aqua [1]. In an online AQP system, samples are typically chosen a-priori (i.e., via Neyman Allocation [11]) and are tailored to a workload or set of predictable queries [3, 7, 8, 22].

Scalable and Interactive Visualization Tools: A slew of interactive visualization tools and frameworks have been introduced by both the database and visualization communities. Examples include Tableau, SeeDB, and SnapToQuery [15, 33, 21].

Online perceptually-aware approximate query processing algorithms like PFunk-H can, in theory, be incorporated into any of such systems so that the visualizations produced can be approximate to enhance interactivity and scalability yet guarantee certain perceptual properties.

Statistical Estimators: Statistical sampling theory is a well-established field with many formulas and theorems that database researchers have leveraged to make sampling-based AQP systems [27, 34, 30].

The core of PFunk-H is based on the Hoeffding-Serfling concentration inequality which works well in practice when the variance of the population is unknown. The concentration inequality provides confidence intervals that makes no distributional assumptions [20]. The estimated confidence intervals could be tighter if the variance were known or even approximated. The Hoeffding-Bernstein concentration inequality uses the empirical variance of a subset of the dataset and produces tighter confidence intervals than Hoeffding-Serfling for certain probability distributions [5].

In future iterations of PFunk-H, we hope to incorporate information from other estimators (such as empirical variance) during online processing.

6. CONCLUSION & FUTURE DIRECTIONS

In this paper, we advocate for the top-down design of visualization systems based on the use of perceptual functions, or pfunks, to model human perception. We introduced a general API for perceptual functions and have shown a proof-of-concept use of perceptual functions for sampling-based approximate query processing. Specifically, we designed the PFunk-H algorithm that can provide formal guarantees on the confidence interval of approximated query results using univariate perceptual functions. The key challenge is developing an iterative approach that allows the output of the perceptual function to be conditioned on the true result values. We then outlined the performance and accuracy benefits by running PFunk-H on both synthetic perceptual functions, as well as functions fit to existing perceptual data.

We are currently working on extensions to the PFunk-H algorithm to support different distributional assumptions, non-monotonic perceptual functions, bi-variate functions, as well as functions to describe animated and interactive settings. In addition, we are developing GPaaS (Graphical Perception as a Service) as a way to simplify the development, deployment, and analysis of perceptual experiments in order to facilitate the collection of quantitative results for different visualizations, tasks, and contexts.

Looking broader, pfunk is a small part of a larger goal of combining the HCI/perceptual literature with data visualization system design. We believe that there is a large body of literature describing human limitations in many visual analysis contexts that have the potential to be translated into high-level constraints that can be used by the visualization system—either for performance as illustrated in this paper, or for general design recommendations as in the visualization recommendation work [**? ?**]. **Design debugger.** **(SOMETHING ABOUT HOW THIS CAN

BE GOOD)**

References

[1] S. Acharya, P. B. Gibbons, and V. Poosala. Aqua: A Fast Decision Support Systems Using Approximate Query Answers. *International Conference on Very Large Databases (VLDB)*, pages 754–757, 1999.

[2] S. Agarwal, B. Mozafari, A. Panda, H. Milner, S. Madden, and I. Stoica. BlinkDB: queries with bounded errors and bounded response times on very large data. In *EuroSys '13*, 2013.

[3] B. Babcock, S. Chaudhuri, and G. Das. Dynamic sample selection for approximate query processing. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, SIGMOD '03, pages 539–550, New York, NY, USA, 2003. ACM.

[4] J. C. Baird. *Psychophysical analysis of visual space.* Pergamon Press Oxford, New York, [1st ed.] edition, 1970.

[5] R. Bardenet and O.-A. Maillard. Concentration inequalities for sampling without replacement. *Bernoulli*, 21(3):1361–1385, 08 2015.

[6] E. Blais, A. Kim, P. Indyk, and S. Madden. Rapid Sampling for Visualizations with Ordering Guarantees. In *41st International Conference on Very Large Data Bases*, volume 8, pages 521–532, 2015.

[7] S. Chaudhuri, G. Das, M. Datar, R. Motwani, and V. Narasayya. Overcoming limitations of sampling for aggregation queries. In *Data Engineering, 2001. Proceedings. 17th International Conference on*, pages 534–542, 2001.

[8] S. Chaudhuri, G. Das, and V. Narasayya. Optimized stratified sampling for approximate query processing. *ACM Trans. Database Syst.*, 32(2), June 2007.

[9] W. Cleveland and R. McGill. Graphical Perception and Graphical Methods for Analyzing Scientific Data. *Science*, 229(4716):828, 1985.

[10] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):pp. 531–554, 1984.

[11] W. Cochran. *Sampling Techniques.* Wiley, third edition, 1977.

[12] Ç. Demiralp, M. S. Bernstein, and J. Heer. Learning perceptual kernels for visualization design. *IEEE Trans. Vis. Comput. Graph.*, 20(12):1933–1942, 2014.

[13] M. N. Garofalakis and P. B. Gibbon. Approximate query processing: Taming the terabytes. In *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pages 725–, 2001.

[14] M. Gleicher, M. Correll, C. Nothelfer, and S. Franconeri. Perception of average value in multiclass scatterplots. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2316–2325, 2013.

[15] P. Hanrahan. Analytic database technologies for a new kind of user: The data enthusiast. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, pages 577–578, New York, NY, USA, 2012. ACM.

[16] L. Harrison, F. Yang, S. Franconeri, and R. Chang. Ranking visualizations of correlation using weber's law. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1943–1952, Dec 2014.

[17] J. Heer and M. Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 203–212. ACM, 2010.

[18] J. M. Hellerstein, P. J. Haas, and H. J. Wang. Online aggregation. *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, pages 171–182, 1997.

[19] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[20] W.-C. Hou, G. Ozsoyoglu, and B. Taneja. Statistical Estimators for Relational Algebra Expressions. In *Proceedings of the 7th ACM SIGACT-SIGMOD-SIGART Symposium on PODS*, pages 276–287, 1988.

[21] L. Jiang and A. Nandi. SnapToQuery : Providing Interactive Feedback during Exploratory Query Specification. 8(11):1250–1261, 2015.

[22] S. Joshi and C. M. Jermaine. Robust stratified sampling plans for low selectivity queries. In *Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, 2008, Cancún, México*, pages 199–208, 2008.

[23] K. Kahol, M. J. Leyba, M. Deka, V. Deka, S. Mayes, M. Smith, J. J. Ferrara, and S. Panchanathan. Effect of fatigue on psychomotor and cognitive skills. *The American Journal of Surgery*, 2008.

[24] M. Kay and J. Heer. Beyond weber's law: A second look at ranking visualizations of correlation. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2016.

[25] D. Kersten, P. Mamassian, and A. Yuille. Object Perception as Bayesian Inference. *Department of Statistics, UCLA*, pages 1–34, 2011.

[26] A. Kim, E. Blais, A. Parameswaran, P. Indyk, S. Madden, and R. Rubinfeld. Rapid sampling for visualizations with ordering guarantees. *Proc. VLDB Endow.*, 8(5):521–532, Jan. 2015.

[27] A. Kleiner, A. Talwalkar, and S. Agarwal. A general bootstrap performance diagnostic. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 419–427, 2013.

[28] P. Kovesi. Good colour maps: How to design them. *arXiv*, 2015.

[29] C. Li, M. R. Luo, and C. Li. Assessing colour rendering properties of daylight sources part ii: A new colour rendering index: Cri-cam02ucs. 2009.

[30] F. Olken and D. Rotem. Simple random sampling from relational databases. In *Proceedings of the 12th International Conference on Very Large Data Bases*, VLDB '86, pages 160–169, San Francisco, CA, USA, 1986. Morgan Kaufmann Publishers Inc.

[31] J. Talbot, V. Setlur, and A. Anand. Four Experiments on the Perception of Bar Charts. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2152–2160, Nov. 2014.

[32] R. Teghtsoonian. *The American Journal of Psychology*, 88(4):677–684, 1975.

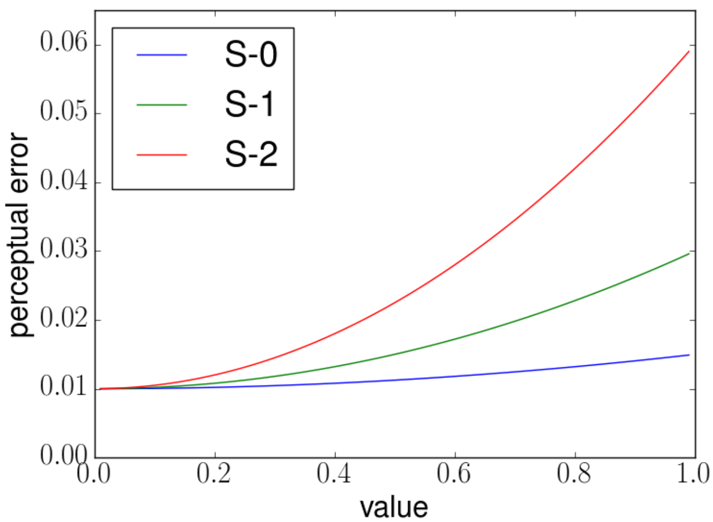
[33] M. Vartak, S. Madden, A. Parameswaran, and N. Polyzotis. Seedb: Automatically generating query visualizations. *Proc. VLDB Endow.*, 7(13):1581–1584, Aug. 2014.

[34] J. S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, Mar. 1985.

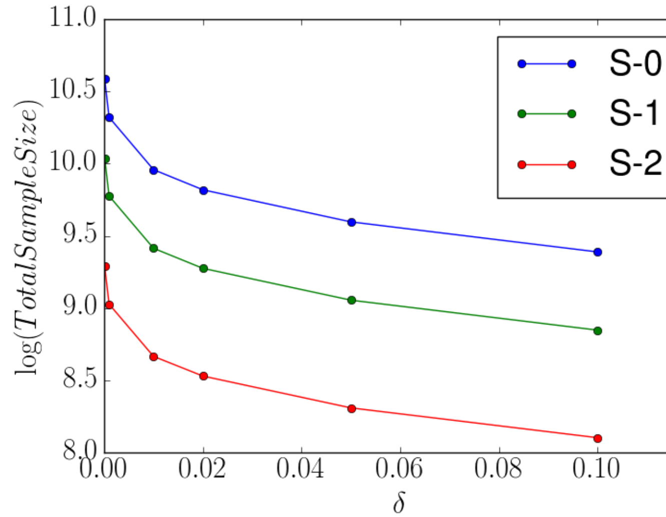
[35] E. Wu, L. Jiang, L. Xu, and A. Nandi. Graphical perception in animated bar charts. *arXiv*, 2016.

[36] E. Wu and A. Nandi. Towards Perception-aware Interactive Data Visualization Systems. 2015.

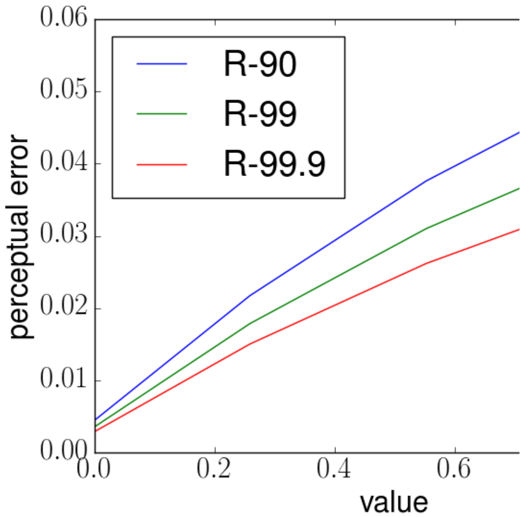
[37] J. Zacks, E. Levy, B. Tversky, and D. J. Schiano. Reading bar graphs: Effects of extraneous depth cues and graphical context. *Journal of Experimental Psychology: Applied*, 4(2):119 – 138, 1998.



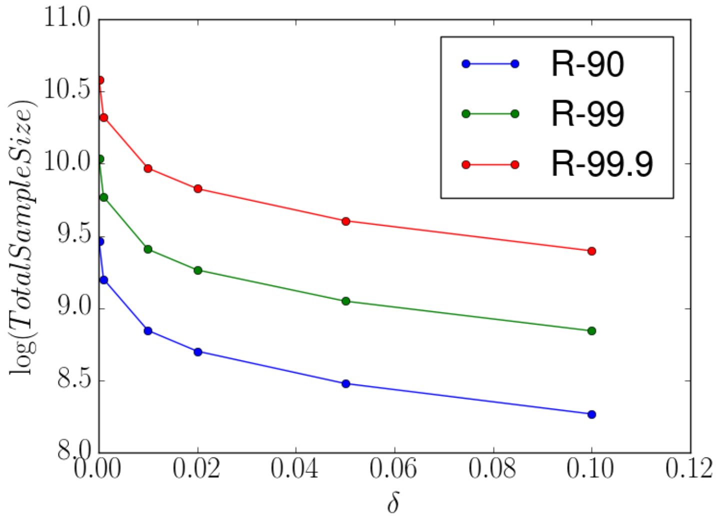
(a) Synthetic quadratic perceptual functions



(b) δ vs. Number of Samples — for $S-0, S-1, S-2$



(c) Logarithmic univariate perceptual functions based on perceptual experiments



(d) δ vs. Number of Samples — for $R-90, R-99, R-99.9$

Figure 1

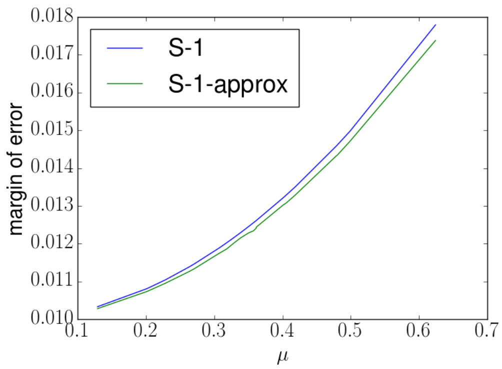


Figure 2: Empirical margin of error produced by PFunk-H follows the perceptual function used