PFunk-H: Approximate Query Processing using Perceptual Models

Daniel Alabi
Columbia University
daniel.alabi@columbia.edu

Eugene Wu
Columbia University
ewu@cs.columbia.edu

ABSTRACT

Interactive visualization tools (e.g., crossfilter) are critical to many data analysts by making the discovery and verification of hypotheses quick and seamless. Increasing data sizes 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. This could either lead to overly aggressive sampling when the approximation accuracy is higher than needed or an incorrect visual rendering when the accuracy is too lax. Thus, for both correctness and efficiency, we propose to use empirical knowledge of human perceptual limitations to automatically bound the error of approximate answers meant for visualization.

This paper explores 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 the approximate and non-approximate answers for a given query differ by a *perceptually indiscernible* amount, as defined by perceptual functions.

1. INTRODUCTION

Dynamic and interactive visualization tools (e.g., crossfilter [37]) make it easier for data analysts to discover and verify hypotheses seamlessly. However, the increasing amounts of data available for analyses is accompanied by the need for scalability of these interactive tools. Increased latency in visualization tools has been shown to negatively affect interactive exploratory visual analytics [31]. To bridge the gap between data sizes and interactivity, many modern visualization systems have turned to approximation [6] as a trade-off between computational cost and query answer accuracy.

Many existing systems employ some form of approximation. Sampling-based approximate query processing (AQP) systems like BlinkDB [2], Aqua [1], and IFocus [7] can evaluate a query on a subset of the underlying dataset. The larger the sample size used during query processing, the better – but slower – the approximation. Recent work has argued for approximation that preserves visually motivated guarantees [24]. For example, IFocus [7], an online sampling algorithm, selects samples until, with a very high probability, the pairwise ordering of bar heights in a bar chart is preserved. In other systems, the accuracy of the approximation procedure is determined either by the user or by resource availability. For example, BlinkDB [2] can use a user-specified time limit or error threshold 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 [7]. Although the paper does not specify how r should be set, this parameter could potentially be automatically set based on human perceptual inaccuracies. For instance, the difference between bar heights of 90 and 91 pixels is perceptually indiscernible in most situations. As a result, we might set the minimal resolution to 1 pixel. Similarly, many humans have trouble distinguishing between small shades of red [8]. Such minimal thresholds for perceptual indiscernibility of visual objects apply in any situation where information can be encoded graphically.

We can use knowledge of human visual perceptual limitations to design better visualization systems. The field of graphical perception explores how humans decode information on data visualizations and contains extensive studies on user perception and judgement accuracies in a multitude of settings [12, 11, 35, 4, 16, 27]. Some of these studies quantitatively assess human perceptual accuracy. For instance, Cleveland et al. [12] 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. Motivated by such studies, in Section 2, we present an initial construction of perceptual models that can be used in sampling-based approximate query processing.

In this paper, we present an online sampling algorithm that can use knowledge of human perceptual error, encoded in perceptual models, to provide approximate answers that differ from the true answers by a perceptually indiscernible

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

DOI: http://dx.doi.org/10.1145/2939502.2939512

amount. The algorithm, PFunk-H, uses simple perceptual functions (linear or non-linear) 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 perceptual correctness.

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 6. We are particularly excited about the potential performance benefits of applying the PFunk ideas in interactive visualizations, where recent studies [39] have found cases where user accuracy is highly robust to many forms of approximation.

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 [35, 26] 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 [30, 29] designed to be perceptually uniform such that perceived color differences uniformly translate to differences of the encoded values. Similarly, Cleveland et al. [11] 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 [38].

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]

$$P(v_1^*, \ldots, v_n^* | C) = \epsilon \tag{1}$$

where $v_1^*, \ldots, v_n^*$ 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. When $\epsilon \to 0$, this corresponds to a constraint that the visualization must render non-approximate values.

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 $P^{(u)} : \mathbb{R} \to \mathbb{R}$ 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 $P^{(u)}(v) = 10^{-2}(5 \cdot v + 1)$. $P^{(u)}(10^2) = 5.01$ represents a margin of error of ± 5.01 for an encoded value of 10^2. If the encoded value is 10, then the margin of error would be ± 0.51.

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 [27]. 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 2c shows logarithmic univariate perceptual functions with encoded truth values normalized to range [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 data from a low contrast setting may not apply when the contrast is high [19]; 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 [25]. 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. Alternatively, users may carry and use personalized libraries of perceptual models. It's important to note that some of these models depend on the user or the visualization environment, while others depend on the computed results—the former case requires quickly detecting and picking the most appropriate model, while the latter case necessitates more advanced approximation techinques than those described here. Over time, we hope this library can provide more 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

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

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 [21], 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 conditioned 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 $N = 1$ 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.

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

To extend this algorithm to records with multiple attributes, we simply 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 stronger inequalities. For instance, the current (loose) bound uses Hoeffding's classical inequality [21] which restricts the values to $[0, 1]$. The "-H" in "Pfunk-H" is a reference to our use of Hoeffding's classical inequality. Alternatives such as Hoeffding-Serfling [5] can reduce the margin of error [2]. Also, the current algorithm assumes that all values are within $[0, 1]$ to simplify the proof. However, it is simple to extend to handle values within a fixed numerical range $[a, b]$ where $a < b$ [21]. 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 [21], $\Pr[v - \mu \geq t] \leq e^{-2st^2}$ 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 the monotonicity assumption for $P^{(u)}$,

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

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, $s = \lceil \frac{\log(2/\delta)}{2t^2} \rceil$.

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

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

$\Pr[P^{(u)}(\mu) \leq P^{(u)}(v-t)] \leq \delta$ 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 [21]. 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,

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

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

4. EXPERIMENTS

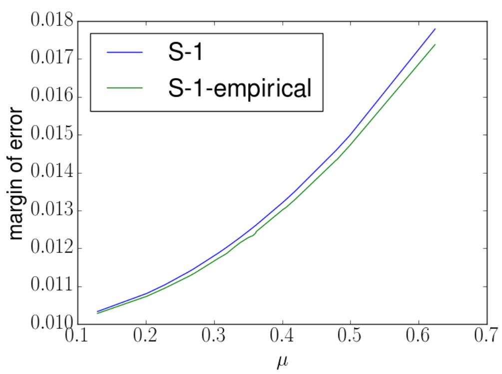


Figure 1: Idealized (S-1) and PFunk-H's empirical margin of error (S-1-empirical) for a quadratic perceptual function

We ran a range of experiments on PFunk-H using different perceptual functions. However, because of space limitations, we show two classes of perceptual functions: logarithmic univariate perceptual functions based on perceptual experiments; and synthetic quadratic functions. We show that the empirical margin of error produced by PFunk-H is very close to the true margin of error based on the perceptual function used. Further, we show how the sampling complexity varies (inverse relation) with δ, the chance of failure.

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 ad-hoc query predicate conditions [28]. 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 and std in ranges $\mu \in [0.1, 0.8]$ and $\sigma \in [0.1, 0.5]$ respectively. Our preliminary experiments focused on the continuous fields.

Figure 2a shows three synthetic quadratic functions S-0, S-1, and S-2. We include these synthetic functions to show the generality of PFunk-H to handle *any* monotonically increasing function, both linear and non-linear. Figure 2c shows three logarithmic functions based on data from perceptual studies on bar charts [38]. The functions R-90, R-99, and R-99.9 use the 90th, 99th, and 99.9th lower bounds, respectively, for the perceptual errors obtained for each true bar height. As a result, the relative order of the log functions is R-90 $>$ R-99 $>$ R-99.9. See Section 2 for more detail on modeling.

4.2 Comparing the Perceptual Functions

Figures 2b and 2d show how the number of samples chosen by PFunk-H varies with the failure probability of the Algorithm δ. 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 2b and 2d show that – for all six perceptual functions) – as the failure probability increases, the number of samples chosen decreases. This makes sense since $\delta \to 0$ implies that the entire dataset should be used to calculate a non-approximate answer.

Figures 2b and 2d 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 which samples more than R-90. For the synthetic squared functions, S-0 samples more than S-1 which samples more than S-2. This result follows the (reverse) relative ordering of the functions shown in figures 2a and 2c.

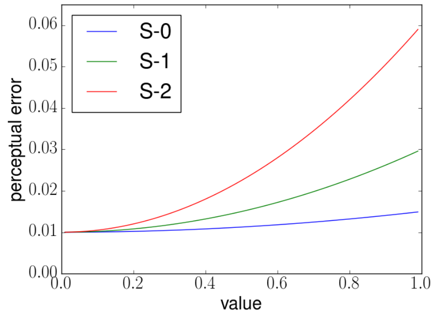
A natural question to ask is: how close is the estimated margin of error (variable t produced in Algorithm 1) to $P^{(u)}(\mu)$ (allowable perceptual error for true answer μ)? Recall, that since we do not know μ before PFunk-H runs, we also do not know $P^{(u)}(\mu)$. We used PFunk-H with function S-1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

5. RELATED WORK

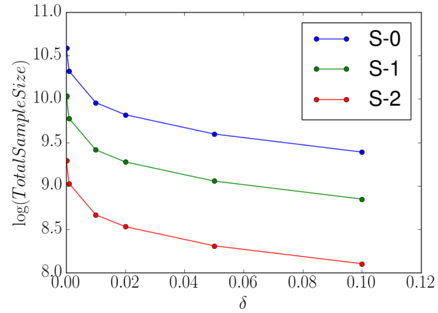
The work related to PFunk-H can be placed in a few categories:

Graphical Perception: One of the first studies to formalize visual perceptual accuracy was done by Cleveland & McGill [12]. A range of experiments have been performed to further validate Cleveland & McGill's foundational work on graphical perception. Talbot et al. [34] extend the original experiment to study the effects of distractor bars and separation between non-adjacent bar charts. Heer et al. [19] approximately replicate the original experiment using crowdsourcing and a larger collection of chart types. Zacks et al. [40] evaluates the same perceptual task but on 3D charts.

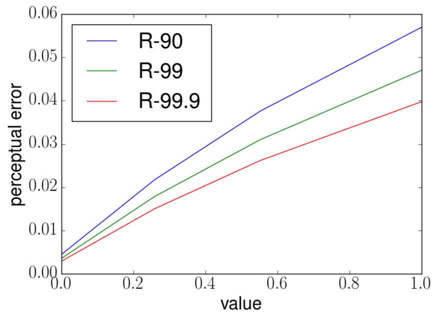
A concept similar to perceptual functions has been introduced by Demiralp et al. [14]. They formulate perceptual kernels, a distance metric of aggregate pairwise perceptual distances used to measure how color, shape, and size affect graphical perception. But it is not immediately obvious (based on its 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



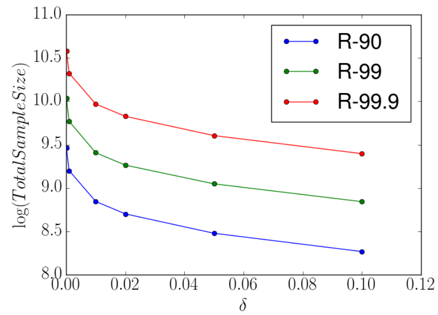
(a) Synthetic quadratic perceptual functions



(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 2

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 [18, 26].

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 [20] 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 [7] 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. [15] 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 [13]) and are tailored to a workload or set of predictable queries [3, 9, 10, 23].

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 [17, 36, 22].

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.

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, our work on the use of perceptual functions in sampling-based approximate query processing 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, for general visualization recommendations [33, 32], or even visualization debuggers that alerts the user or designer when the visual rendering violates visual constraints.

This work is supported in part by NSF 1527765.

References

[1] S. Acharya, P. B. Gibbons, and V. Poosala. Aqua: A Fast Decision Support Systems Using Approximate Query Answers. 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. 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] L. Battle, M. Stonebraker, and R. Chang. Dynamic reduction of query result sets for interactive visualizaton. In *Proceedings of the 2013 IEEE International Conference on Big Data, 6-9 October 2013, Santa Clara, CA, USA*, pages 1–8, 2013.

[7] 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.

[8] B. Brewer. Perception and its objects. *Philosophical Studies*, 132:87–97, 2007.

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

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

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

[12] 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.

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

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

[15] M. N. Garofalakis and P. B. Gibbon. Approximate query processing: Taming the terabytes. VLDB, pages 725–, 2001.

[16] 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.

[17] P. Hanrahan. Analytic database technologies for a new kind of user: The data enthusiast. SIGMOD '12, pages 577–578, New York, NY, USA, 2012. ACM.

[18] 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.

[19] 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.

[20] J. M. Hellerstein, P. J. Haas, and H. J. Wang. Online aggregation. *SIGMOD*, pages 171–182, 1997.

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

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

[23] 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.

[24] U. Jugel, Z. Jerzak, G. Hackenbroich, and V. Markl. M4: A visualization-oriented time series data aggregation. *Proc. VLDB Endow.*, 7(10):797–808, June 2014.

[25] 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.

[26] 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.

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

[28] 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.

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

[30] 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.

[31] Z. Liu and J. Heer. The effects of interactive latency on exploratory visual analysis. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2014.

[32] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141, Apr. 1986.

[33] J. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1137–1144, Nov. 2007.

[34] 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.

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

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

[37] C. Weaver. Multidimensional visual analysis using cross-filtered views. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology, IEEE VAST 2008, Columbus, Ohio, USA, 19-24 October 2008*, pages 163–170, 2008.

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

[39] E. Wu and A. Nandi. Towards Perception-aware Interactive Data Visualization Systems. *DSIA Workshop, IEEE VIS*, 2015.

[40] 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.