Qetch: Time Series Querying with Expressive Sketches

Miro Mannino
New York University
Abu Dhabi, UAE
miro.mannino@nyu.edu

Azza Abouzied
New York University
Abu Dhabi, UAE
azza@nyu.edu

ABSTRACT

Query-by-sketch tools allow users to sketch a pattern to search a time series database for matches. Prior work adopts a bottom-up design approach: the sketching interface is built to reflect the inner workings of popular matching algorithms like Dynamic time warping (DTW) or Euclidean distance (ED). We design Qetch, a query-by-sketch tool for time series data, *top-down*. Users freely sketch patterns on a scale-less canvas. By studying how humans sketch time series patterns we develop a matching algorithm that accounts for human sketching errors. Qetch's top-down design and novel matching algorithm enable the easy construction of expressive queries that include regular expressions over sketches and queries over multiple time series. Our demonstration showcases Qetch and summarizes results from our evaluation of Qetch's effectiveness.

ACM Reference Format:
Miro Mannino and Azza Abouzied. 2018. Qetch: Time Series Querying with Expressive Sketches. In *SIGMOD'18: 2018 International Conference on Management of Data, June 10–15, 2018, Houston, TX, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3183713.3193547

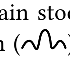
1 INTRODUCTION

Query-by-sketching is a promising paradigm for querying time series databases as it capitalizes on our innate, prelinguistic ability to communicate with sketches [3]. For it to be effective, however, the interface and the sketch-to-time series matching algorithm need to be specifically designed to tolerate human sketching imperfections. Existing tools are often designed bottom-up: they attempt to reconcile imperfect sketches with how time series matching algorithms search for data through (a) *overlays*: users sketch directly above the pattern they are looking for to retrieve similar patterns in the data [10, 11], (b) *shape restrictions*: instead of free-form sketches, users can only sketch sequences of straight lines [6], (c) *pre-sketching constraints*: users specify the temporal range they are interested in and the variance of amplitude they are willing to tolerate [2, 5, 10]. Such restrictions enable these tools to use standard and powerful time-series to time-series matching algorithms like Dynamic time warping (DTW) and Euclidean distance (ED), but they limit overall effectiveness: "good" matching algorithms may fail to produce good similarity rankings when matching a sketch to time series when "goodness" is assessed by humans [2].

In this work, we demonstrate Qetch[1]: a query-by-sketch tool where users can freely sketch patterns on an empty scale-less canvas. Our top-design approach leads us to a novel matching algorithm that handles the absence of any time or amplitude scales and also tolerates human sketching imperfections. Moreover, Qetch preserves the canvas as the primary mode of query specification to maintain user interface consistency. Through *sketch annotations*, users can apply query filters on the time span, amplitude, time or value offset of a query as well as specify *regular expression operations* such as repetition or negation over sketch segments. Through the *relative horizontal positioning* of multiple sketches on the canvas, users can query multiple time-aligned series and specify the order of events across them. Thus, Qetch supports complex time-series querying with expressive sketches.

2 DEMONSTRATION

At SIGMOD 2018, we will demonstrate Qetch's ability to effectively query time series with hand-drawn, scale-free sketches. We will pre-load Qetch with multiple data sets from the domains of finance, economics, technology and medicine. Attendees can query any data set with the help of several pen-enabled tablets. We will guide attendees through the main interface features of Qetch as well as describe Qetch's underlying sketch-to-data matching algorithm. In addition, attendees can flip the underlying algorithm to either DTW or ED to experience first hand the effectiveness of Qetch compared to other algorithms that are not specifically designed for matching scale-free sketches to time series. While attendees can choose any data set to query, we will showcase Qetch with the help of a guided use-case scenario.

Guided Demo Scenario The Amateur Investor: Joe is a database systems researcher and an amateur tech investor. He recently learned that certain stock chart patterns, such as the head and shoulders pattern (), can indicate a reversal in the trend of stock prices and can help him forecast future stock prices. Armed with this knowledge, he loads a collection of closing stock-price time series and sketches a scale-free head and shoulders pattern on Qetch's query canvas (See Figure 1).

Given this scenario, we walk the attendees through (i) the preparation of time series for query-by-sketching, (ii) the design principles behind Qetch's query and result interface, (iii) the top-down design of Qetch's unique sketch-to-time series matching algorithm and (iv) Qetch's support for advanced query features such as regular expressions over sketches and the querying of multiple time series through the relative positioning of sketches on the canvas.

[1]A pre-print of the CHI 2018 paper describing and evaluating Qetch [7], as well as a short video demo can be found here [9].

SIGMOD'18, June 10–15, 2018, Houston, TX, USA

ACM ISBN 978-1-4503-4703-7/18/06…$15.00
https://doi.org/10.1145/3183713.3193547

Figure 1: Qetch's user interface [7].

2.1 Pre-processing

On loading a data set, Qetch immediately stores the data into a standard relational database and starts a background pre-processing thread that iteratively segments and smooths the data.

Motivated by research on human visual perception that suggests that users mentally decompose complex shapes, such as times series data into visually salient parts such as peaks, troughs and do so using curvature [4], Qetch uses changes in curve monotonicity to determine the end of one segment and hence the beginning of a new segment. Qetch also uses an exponential moving average to smooth the time series. Smoothing captures the key patterns of the data, while leaving out noise. Qetch iteratively smooths and segments the time series as long as the number of segments in the times series is reduced by a constant configurable factor. This leads to at most $O(\log(d))$ smoothed and segmented series for each loaded time series, where d is the number of data points in the time series. Qetch searches for matches across all these series.

2.2 Query and Result Interface

As an amateur investor, Joe may find it challenging to answer simple questions such as *how long should the head and shoulders pattern last?* or *how high should each of the three peaks be?* With Qetch, Joe simply sketches the head and shoulders pattern in the query canvas, without having to consider these questions, to immediately visualize results, from which he can further refine his query.

Qetch's canvas has no time or amplitude scales and is independent of the current scales of the visualization. Qetch slightly smooths the sketch to eliminate hand jitter and interpolates it to create a modifiable Bézier curve. As Figure 1 demonstrates, Qetch returns an ordered set of all matches across a single or a collection of loaded time series. Matches are color-coded by their distance from the sketch: green for good (short distance) matches, yellow for fair and red for poor matches. Joe can choose to view matches at different smoothing degrees and order results by length or distance.

Joe can refine his query by adding sketch annotations such as the *query length*, or how long the pattern lasts. For example in Figure 1, the annotation ensures that the query pattern is between 100 and 200 days long. He can also annotate the query with an *amplitude scale*, a *time offset* or a *value offset*. Annotations create filters that are applied after Qetch's matching algorithm finds candidate matches. We chose to use sketch annotations to specify such filters to maintain the user's mental model that the sketch is the primary mode of time series query specification.

2.3 Matching Algorithm

Our design of Qetch's matching algorithm was guided by an extensive crowd study of the sketching behavior of 150 workers. Each worker sketched eight different patterns to query data sets from a wide range of domains [7][2]. We observed the following key sketching behaviors:

(1) *Preservation of visually salient features* such as peaks, troughs and slopes. The more pronounced the feature of interest was — the longer the slope, the deeper or wider the trough or peak — the more likely it was sketched.
(2) *Non-uniform global scaling.* Humans do not respect aspect ratio. Consider a sketch of an upward sloping straight line at a 45°angle. Any upward sloping line segment within the data is an acceptable match to this sketch, provided the time and amplitude scales of the line's visualization are modified accordingly.
(3) *Local distortions.* Certain features within a sketch may be exaggerated such as the width or depth of a peak or trough or the relative difference between the heights of smaller and larger peaks in a pattern.

Thus, to support query-by-sketching we require a matching algorithm that is sensitive to the perceptual features of a sketch and gives equal weight to each feature. Qetch breaks the sketch

[2]We publicly release this data set of crowd sourced sketches [9].

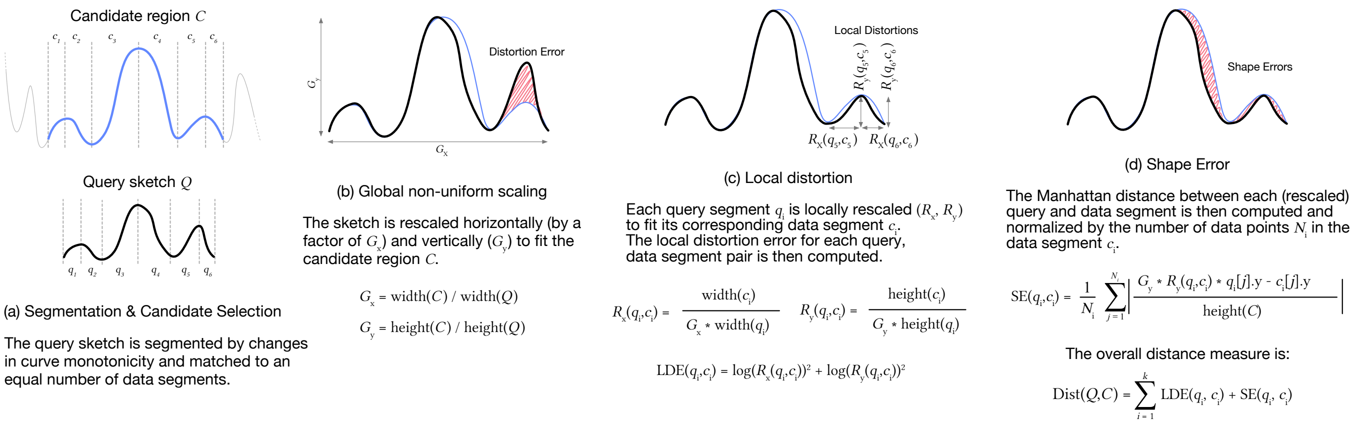


Figure 2: The distance measure computed by Qetch's matching algorithm for a candidate region within the time series.

into segments by changes in curvature and matches query to data segments rather than time slices like ED or DTW (Figure 2(a)). Qetch matches the sketch to all the smoothed series of a time series. This allows Qetch to capture short as well as long patterns.

Taking into consideration that sketches often do not respect aspect ratio, Qetch non-uniformly globally rescales the sketch to fit a sequence of candidate data segments (Figure 2(b)).

After segmentation and global-rescaling, Qetch computes the distance of the sketch from the candidate data region as a linear combination of two errors: *local distortion errors* and *shape errors*. The distortion error is the amount of local rescaling required for each query segment to better fit its corresponding data segment (Figure 2(c)). The distortion error accounts for our tendency to exaggerate features within sketches and minimizes its overall influence on the distance measure compared to standard distance measures such as Euclidean distance (ED). For example, Joe may exaggerate the relative height difference between the head and the shoulders. The shape error is the difference in shape, or Manhattan distance, between a data and query segment after locally distorting the query segment (Figure 2(d)). One can replace the Manhattan distance with other distance measures (e.g. ED).

Qetch returns a ranked list of matches from all smoothed series ordered by their distance from the sketch. Only the closest match from matches that overlap across multiple smoothed series is returned. Further details on the matching algorithm and its run-time complexity can be found in the full paper [7].

2.4 Advanced Query Features

The simple design of Qetch's matching algorithm allows it to be easily extended to support regular expressions as well as multiple sketches for querying different time-aligned series, in addition to tolerating human sketching errors.

2.4.1 Regular Expressions. We provide support for expressing complex queries such as regular expressions over sketches to allow users to easily represent pattern repetition or negations, which can be difficult to sketch. Keeping the canvas as the primary query specification mode, users specify regular expressions as sketch annotations (Figure 3). Joe may be interested in understanding if

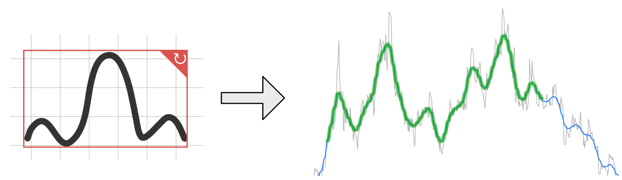


Figure 3: Regular Expressions with Qetch: Joe searches for a repeated head and shoulders pattern. Qetch's top result coincides with two consecutive head and shoulders from 1999 to 2000 in Amazon's closing stock-price time series.

the head and shoulder's pattern can repeat before the predicted fall in stock prices. In Figure 3, he annotates his sketch with the repeat operator and Qetch returns Amazon's stock price from 1999 to 2000 as its first result.

Qetch evaluates regular expressions with the help of a finite-state machine. Each segment in the sketch represents a state in the machine. A sequence of consecutive segments in the sketch is represented by a sequence of connected states in the finite state machine. A repeat operator simply adds more transitions to the machine. Currently Qetch supports only three regular expression operators: repeat, repeat n times and not. We hope to implement more operators in the future.

In the demonstration, we will guide attendees to try out other query scenarios that demonstrate the power of regular expressions over sketches such as: finding patterns of repeated high network traffic followed by repeated low traffic in UK's academic network backbone [1], or finding abnormal heart rhythms in a data set of ECGs [8] by negating a sketch of a normal heart rhythm.

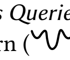
2.4.2 Relative Positioning for Multiple Time Series Queries. Joe also learns that an inverse head and shoulders pattern () predicts a potential rise in stock prices. He starts investigating if Amazon's 2000 predicted fall in stock prices may have triggered a potential rise in another tech company's stock prices. He uses Qetch's relative positioning feature to sketch the head and shoulders patterns and its inverse. By positioning the inverse pattern ahead of the head and shoulders pattern, Joe indicates to Qetch the order of

Figure 4: Querying multiple time series with Qetch: Joe queries multiple time-aligned stock prices to find periods where a head and shoulders pattern observed in one stock was followed by an inverse head and shoulders in a different stock. The relative positioning of the sketches determines whether the matches should overlap or occur in a certain order.

events he is interested in. In Figure 4, Qetch finds a match for the ordered events in Amazon's and Microsoft's time series.

Relative positioning maintains the canvas as the primary query specification interface. Moreover it is more effective than directly specifying simplified order constraints of the form query C before query A, query B after query C, etc [7].

3 EVALUATION SUMMARY

We conclude our demonstration by describing the main takeaways from our evaluation of Qetch's user interface and matching algorithm [7]:

(1) We conducted a within-subjects comparative user study of Qetch's novel querying features: (i) regular expressions for querying repeated patterns and for anomaly detection versus no regular expressions and (ii) relative positioning of sketches for querying across multiple data sets verses specifying order constraints over sketches. Our participants were given timed query tasks to objectively determine if Qetch's features improved a user's query performance. We found that users completed tasks in significantly less time and with fewer errors with Qetch's query features enabled.

(2) We asked participants to indicate the minimum, preferred and maximum degree of smoothing they would like to view a highlighted query result at. We found that Qetch's smoothing choices are within a 95% confidence interval of the mean preferred smoothing degree of most queries, indicating that Qetch's visualization of query results matches user's preferences.

(3) Using our collection of crowd sourced query sketches, we compared the precision of Qetch's matching algorithm to Dynamic time warping (DTW) and Euclidean distance (ED) in targeted search tasks, i.e. finding a specific region of interest within a time series. With a few exceptions, Qetch outperforms both DTW and ED. Qetch, unlike DTW and ED, did not use query length to filter out irrelevant results. Thus, queries with nonspecific patterns such as a single trough (◡) benefited from DTW's and ED's query length.

(4) Finally, we asked participants to perform multiple exploratory search tasks on multiple data sets. We asked them to rank the top ten results of Qetch's matching algorithm and DTW. We found that Qetch outperforms DTW on the popular Normalized Discounted Cumulative Gain measure across users.

Details of the different experiments and our analysis can be found in the main publication [7].

4 CONCLUSION

Our goal in this demonstration is to illustrate how the top-design of a query-by-sketching tool is key to its effective performance. By studying how humans sketch time series patterns, we not only re-designed the query interface but also the matching algorithm, producing one that is better suited for sketches and more tolerant of common sketching errors. In our demonstration, attendees can experience for themselves the power of Qetch's design. Looking forward, we wish to ensure the scalability of Qetch's matching algorithm on massive time series and to expand Qetch's query expressiveness with more support for regular expressions and other features such as motif detection, searching for correlations across series, clustering series by similarity, etc., while preserving Qetch's visual, canvas-driven, querying interface.

REFERENCES

[1] P. Cortez, M. Rio, M. Rocha, and P. Sousa. 2006. Internet Traffic Forecasting using Neural Networks. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. 2635–2642. https://doi.org/10.1109/IJCNN.2006.247142

[2] Philipp Eichmann and Emanuel Zgraggen. 2015. Evaluating Subjective Accuracy in Time Series Pattern-Matching Using Human-Annotated Rankings. In *Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI '15)*. ACM, New York, NY, USA, 28–37. https://doi.org/10.1145/2678025.2701379

[3] Mathias Eitz, James Hays, and Marc Alexa. 2012. How Do Humans Sketch Objects? *ACM Trans. Graph.* 31, 4, Article 44 (July 2012), 10 pages. https://doi.org/10.1145/2185520.2185540

[4] Donald D Hoffman and Manish Singh. 1997. Salience of visual parts. *Cognition* 63, 1 (1997), 29 – 78. https://doi.org/10.1016/S0010-0277(96)00791-3

[5] Christian Holz and Steven Feiner. 2009. Relaxed Selection Techniques for Querying Time-series Graphs. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*. ACM, New York, NY, USA, 213–222. https://doi.org/10.1145/1622176.1622217

[6] Eamonn Keogh and Padhraic Smyth. 1997. A Probabilistic Approach to Fast Pattern Matching in Time Series Databases. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD'97)*. AAAI Press, 24–30.

[7] Miro Mannino and Azza Abouzied. 2018. Expressive Time Series Querying with Hand-Drawn Scale-Free Sketches. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (to appear) (CHI '18)*. ACM, New York, NY, USA, 10. https://doi.org/10.1145/3173574.3173962

[8] George B. Moody and M.D. Ary L. Goldberger. 2017. MIT-BIH Heart Rate Time Series. (July 2017). http://ecg.mit.edu/time-series/

[9] New York University Abu Dhabi, Design Technology Lab. 2018. Qetch source code repository. (January 2018). https://github.com/dtl-nyuad/qetch

[10] Martin Wattenberg. 2001. Sketching a Graph to Query a Time-series Database. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems (CHI EA '01)*. ACM, New York, NY, USA, 381–382. https://doi.org/10.1145/634067.634292

[11] Kostas Zoumpatianos, Stratos Idreos, and Themis Palpanas. 2015. RINSE: Interactive Data Series Exploration with ADS+. *Proc. VLDB Endow.* 8, 12 (Aug. 2015), 1912–1915. https://doi.org/10.14778/2824032.2824099