Exploring Visualization of Data Transforms

[Extended Abstract]

Larry Xu
UC Berkeley
larry.xu@berkeley.edu

ABSTRACT

In the context of data exploration, users often interact with relational database systems in an interactive query session to form useful insights. Each query a user executes can potentially transform a resultset in complex ways. We explore some of the challenges in understanding these transformations, and how these challenges can be solved through more informative visual representations of data transforms. We present the concept of "tweening" of resultsets as a method of incrementally visualizing data transformations, and explore approaches towards generating these resultset tweens. Through a series of user studies, we evaluate tweening as an effective method of understanding the changes that result from data transformations.

1. INTRODUCTION

There has been a great deal of research on visual interfaces to databases, resulting in the creation of many tools for specifying queries and visualizing results. These visualization tools can greatly improve the usability of database systems, but despite all of the benefits these tools provide, users can still have difficulty understanding how a query specifically affects its results. The visual difference between the input and output of a query transformation can be significant and difficult to comprehend. In our work we focus on the design and evaluation of techniques to bridge the gap between these input and output visualizations, so that users can understand the effects of their queries in a more detailed and intuitive manner.

In a typical relational database setting, a user will write a sequence of queries which explore the structure and content of the data being analyzed. The user enters a feedback loop where each query result reveals more information to help the user write their next query until the desired results are found. Each query iteratively transforms a single resultset, but it can be difficult to understand how the data is being manipulated and filtered as these transformations occur. This interface lacks data lineage information that describes how the data changes throughout the transformation[1]. These problems are not only prevalent when writing a sequence of queries; even individual queries that cause large complex transformations, such as aggregations or pivots, can be difficult to comprehend.

To help users understand data transformations we introduce the concept of "tweening" of resultsets. The concept of tweening stems from the world of animation. It is the process of generating intermediate frames between two or more key frames to give the appearance of a smooth transition. This technique has been widely used in animation, and we believe that it can be adapted towards creating animated visualizations of data transformations. Previous research has shown that animated data visualizations can help in decision making[2] and reducing a user's cognitive load[3]. Furthermore, animated visualizations provide additional data lineage information that traditional interfaces lack.

2. RESULTSET TWEENING

We discuss the concept of resultset tweening by first defining a query session for the purposes of interactive data exploration, followed by different approaches to generating a tweening sequence.

2.1 Query Session

We define a query session as a series of queries on the same relation. A query Q transforms resultset R_{old} into a new resultset R_{new} where Q is a standard SQL query and each resultset is tabular data visually represented as a standard two-dimensional grid. Hence, a resultset tween for query Q visualizes an incremental transformation from R_{old} to R_{new}.

2.2 Approach

A resultset tween can be represented as an ordered sequence of visual operations. Each operation modifies the visual representation of a resultset through structural transforms (e.g. insertions and deletions) or visual cues (e.g. highlights and annotations). A comprehensive set of visual operations can be defined in a formal grammar which expresses the entire space of data transforms on resultsets. Figure 1 showcases a simple tween example where a resultset is modified through a series of such visual operations.

There are multiple approaches that can be used to generate a tweening sequence for a data transform. First we consider a result-based approach which analyzes only the differences between two resultsets (a table diff) without considering the underlying queries which created those resultsets. By mapping a table diff to a sequence of cell/row/column inser-

SIGMOD'16 June 26 - July 01, 2016, San Francisco, CA, USA

ACM ISBN 978-1-4503-3531-7/16/06.

DOI: http://dx.doi.org/10.1145/2882903.2914837

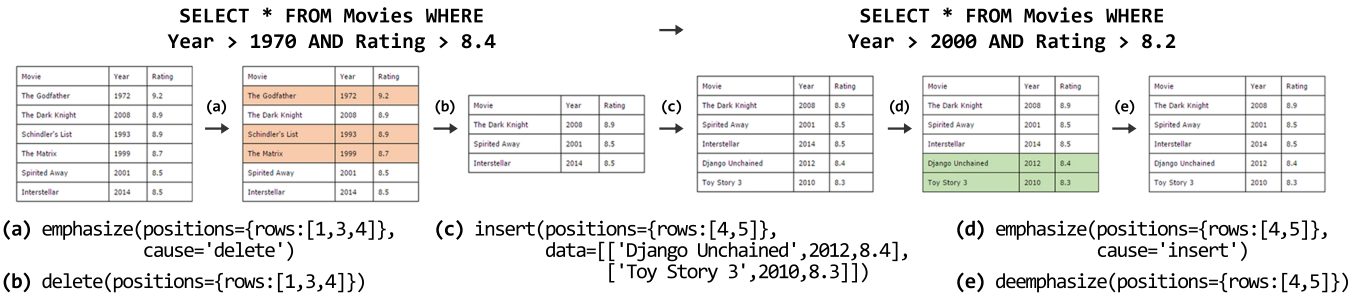

Figure 1: Frames of a sample tweening sequence for a query change in the WHERE clause highlighting the addition and deletion of rows from the original table. The visual interface should animate the tween to produce a smooth transition.

tion, deletion, and reorder operations, we can automatically generate a tween between any two resultsets. This approach works well for very simple resultset transforms, however can create inaccurate visual tweens for more complex types of transformations such as aggregations or pivot operations.

Instead, we consider a query-based approach to tweening that analyzes differences in the underlying query structure combined with a rulebase to accurately generate a tweening sequence. This approach breaks down the two queries Q_{old} and Q_{new} into a query diff Δ_Q that specifies which SQL clauses changed, and uses rules to determine which visual operations to use for the tween. Each rule maps a type of SQL clause change to an ordered set of visual operations.

3. EVALUATION

To evaluate the effectiveness of tweening as a method of visualizing data transformations, we conducted a set of user studies that quantify how well users understand certain properties of a data transformation after being exposed to a resultset tween. For each study, we employed a between-subjects design with two groups, a "tween" group and a "non-tween" group, where the users of each group are shown a tweened and non-tweened visualization, respectively. We deployed each study as a separate task on Amazon Mechanical Turk, ensuring that each user participated in at most one study to prevent any biases from completing multiple studies.

3.1 User Studies

We designed 3 studies that show the user an example transformation that is the result of a query change:

- Study 1 - Change in SELECT clause
- Study 2 - Change in WHERE clause
- Study 3 - Change in SELECT and WHERE clauses

We then ask the user to identify how many rows and/or columns were added or removed in the process. We scored the user responses by summing the absolute differences of the response and expected answer for each question. After collecting a sample distribution of scores for the "tween" group and the "non-tween" group, we ran a Mann-Whitney U test to detect whether the two distributions differed significantly. Table 1 shows the results of a one-tailed test for the 3 different studies we ran. The results suggest that the users in the "tween" group were able to more accurately judge the number of rows and columns that were modified.

Table 1: Mann-Whitney U Test Statistics

Study	Tween Users	Non-tween Users	U	p-value
1	20	20	118	0.02391
2	17	20	103	0.00917
3	18	20	121	0.08528

4. CONCLUSIONS AND FUTURE WORK

The analysis reveals that resultset tweening can be an effective means of portraying a visual representation of the transformations that occur in an interactive query session. The query-based tweening approach described in this paper presents a good first step towards automatically generating tweening sequences. With a proper visual grammar and rulebase it can readily be adapted into existing data visualization tools like Tableau or d3.js to animate SQL transformations. Further exploration into the factors that comprise a "useful" tween and ways to optimize the generation of tweening sequences via visual cost models would be appropriate next steps to furthering our understanding of resultset tweens. More research experimenting with tweening for large resultsets that cannot fit well on screen could further the practicality of tweening as well.

5. ACKNOWLEDGMENTS

I would like to thank Joseph Hellerstein, Arnab Nandi, and Meraj Khan for their guidance and fundamental contributions to this research project. This report serves as a summary of our collective efforts in working on this project, as well as my own personal contributions as an undergraduate researcher working with them.

6. REFERENCES

[1] J. Cheney, L. Chiticariu, and W.-C. Tan. Provenance in databases: Why, how, and where. *Foundations and Trends in Databases*, 1(4):379–474, 2009.

[2] C. Gonzalez. Does animation in user interfaces improve decision making? In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 27–34, 1996.

[3] G. G. Robertson, S. K. Card, and J. D. Mackinlay. Information visualization using 3d interactive animation. *Communications of the ACM*, 36(4):57–71, 1993.