SIGCHI Conference Proceedings Format

Leave Authors Anonymous
UC Berkeley, USA
e-mail address

Leave Authors Anonymous
UC Berkeley, USA
e-mail address

Leave Authors Anonymous
UC Berkeley, USA
e-mail address

Leave Authors Anonymous
UC Berkeley, USA
e-mail address

ABSTRACT
[Mick: 100 words?]

Author Keywords
Authors' choice; of terms; separated; by semicolons; include commas, within terms only; this section is required.

CCS Concepts
•**Human-centered computing** → **Human computer interaction (HCI);** *Haptic devices;* User studies; Please use the 2012 Classifiers and see this link to embed them in the text: https://dl.acm.org/ccs/ccs_flat.cfm

INTRODUCTION
[Mick: todo]

RELATED WORK
[Mick: todo]

INTERVIEWS
[Mick: todo] [Todo: andrew]

DATA MODEL
[Mick: todo] [Mick: to focus: the added idea of exploring video data by frames instead if by instances like in apperception] [Todo: shadaj]

DATASET
[Mick: to focus:
- explain briefly about the dataset, what for, shape of the data.
- how we transform the dataset to fit our data model. -] [Todo: Amy]

LANGUAGE
[Mick: to focus:
- by-frames operations: as a response to Lisa's use case where we need to scope the output video to the only interesting parts of each videos.
- by-frames <–> by-instances: as a response to Yousef's comments to allow the language to express more queries]

To address the pain points we noticed in our user interviews, we chose to design a new language for describing queries over geospatial video data as a domain-specific language embedded in Python. Instead of building a new language from scratch, which would require designing a new syntax and editor integrations that users would have to learn about, we are able to inherit the large ecosystem of existing tooling around Python. Furthermore, some potential users of our system are already familiar with Python, which makes it further easier to integrate into existing pipelines.

There are three key factors constraining the design of our language: incremental query creation, support for using the language through a graphical user interface, and performant query execution.

EXTENSIONS
[Todo: Andrew]

USER INTERFACE
[Mick: to focus:
- how we map the programming language to the user interface. For example, how do we present our data model visually in the graphical user tool.
- how do the queries that we can construct in the graphical user tool reflect the queries that we can construct with our query language.
- We do not have to try to cover all the cases that the query language can produce, but we should try to capture the one that are important. For example, how to query by frames, how to query by instances, how to represent the lambda function, how to represent the custom function.
- It might be a good idea to create block-based programming language that mirror our query language (would this be easier than trying to design a new UI from ground up???)] [Todo: Andrew] [Todo: Mick]

CONCLUSION
[Mick: todo]

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

ACKNOWLEDGMENTS

[Mick: todo]

REFERENCES