QuiltView: a Crowd-Sourced Video Response System

Zhuo Chen, Wenlu Hu, Kiryong Ha, Jan Harkes, Benjamin Gilbert, Jason Hong, Asim Smailagic, Dan Siewiorek, and Mahadev Satyanarayanan

Presenter: Chen Cheng
Outline

• Motivation
• Introduction
• Key Design Guideline
• System Architecture
• Prototype Implementation
Is there crowded on the road?
What if you have a Google Glass?
Motivation

• More and more Glass-like devices have been released
• However, their functionality today is roughly that of a smartphone
• Beyond novelty and coolness, however, their true value proposition is not clear.
• To achieve payoff from widespread deployment of Glass-like devices, this paper come up with a crowd-sourced system called QuiltView
What is QuiltView?

• QuiltView is a crowd-sourced system that leverages the ability of Glass-like devices to provide near-effortless capture of first-person viewpoint video.
Key Design Guideline

• Micro-interaction
  – These are very brief episodes of display or audio stimulus from the system, followed by a quick and unobtrusive user response.

• High-value outputs

• At low cost in terms of user distraction
What are examples?

• One possible candidate is video capture in response to a brief query.
• A brief query = 50 characters or less (fits within the small Glass display)
• Video response = 10-second video segment
How could QuiltView make a difference in the real world?

- Traffic Emergency
- Missing Child (Amber Alert)
- Real-time Queue View
- Free Food Finder
- Scavenger Hunt
- Time Machine
System Architecture

Figure 1: The Cloud-based QuiltView Architecture with YouTube and Google Maps Integration
Prototype Implementation

- Glass Client
- QuiltView Service
- Query Workflow
- Query Similarity
- User Load Balancing
- Synthetic Glass Users
Glass Client

- Glass Development Kit (GDK)
  - provides much richer functionality than the Mirror API
  - supports creation of native Glass clients
- Poll the QuiltView server periodically to check for new queries
QuiltView Service

- Web-based service in a single virtual machine at Amazon EC2 East
- Standard load balancing and scaling mechanisms
- Standard distributed systems techniques
- SSL (Secure Sockets Layer)
  - Unique ID for each Glass device
  - Registration authenticated by Mozilla Persona
  - No new password creation involved
- Django application and MySQL database
Query Workflow

Figure 3: Composing a QuiltView Query
# Query Similarity

## (a) Examples of True Positives

<table>
<thead>
<tr>
<th>Queries</th>
<th>Matched Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is there a thunderstorm?</td>
<td>What is the weather?</td>
</tr>
<tr>
<td></td>
<td>Is it raining?</td>
</tr>
<tr>
<td></td>
<td>Is it cold outside?</td>
</tr>
<tr>
<td>How is the thesis defense?</td>
<td>Are the professor’s questions hard?</td>
</tr>
</tbody>
</table>

## (b) Examples of False Positives

<table>
<thead>
<tr>
<th>Queries</th>
<th>Wrongly Matched Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is it sunny?</td>
<td>What is happening in India?</td>
</tr>
<tr>
<td>Is the new year celebration fun?</td>
<td>Is the Mardi Gras exciting?</td>
</tr>
</tbody>
</table>

## (c) Examples of False Negatives

<table>
<thead>
<tr>
<th>Queries</th>
<th>Missed Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>How is the party?</td>
<td>Is anyone drunk?</td>
</tr>
</tbody>
</table>
User Load Balancing

- Users’ locations and the geographic scope of the query
- Users’ preferences – time, location, query topic, and reward offered
- Randomly choose the desired number of users
- Deliver the query to them
Synthetic Glass Users

• Considering hundreds or thousands of Glass users in a city-size area for future
• Limited numbers of Glass devices at present
• Emulate Glass users using Python software
  – Randomly return a .mp4 file in response
  – Only response, never pose any query
Synthetic Glass Users
Why or why not smartphone?

Questions?
Thank you!