QuiltView: A Crowd-Sourced Video Response System

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

2. QuiltView Design

3. Architecture & Prototype

4. Discussions
Emergence of Smart Glass

Head-up display
Quick information access

Camera
Near effortless capture

Functionality
Roughly a smart phone
Micro-interactions

Not distract users from current context
– Brief stimulus from the system
– Quick and unobtrusive response

Two second rule
“Anything that takes you more than two seconds to accomplish, needs a good reason to be done. Anything that takes significantly less you will do without a second thought.” †

What high value micro-interactions can one have with a Glass user?
2. QuiltView Design

1. Background

3. Architecture & Prototype

4. Discussions
QuiltView Introduction

QuiltView is a near-real-time crowd-sourced system in which users reply to queries with videos through Glass.

– Queries are geo-casted to selected users
Why Video?

How is the night market now (at Taipei)?

Choice one
- Respond with text
- “Um… the weather is good here, but it’s pretty crowded and noisy, as usual. I see people selling pants, bags…”

Choice two
- Respond with video
Simple Queries, Deep Answers

How is the night market now (at Taipei)?

“How... the weather is good here, but it’s pretty crowded and noisy, as usual. I see people selling pants, bags...”

- Videos “bring you to the scene”
  - “If a picture is worth a thousand words, then a video is worth a million”†

- Answer depends on personal taste
  - Requester can judge with his own knowledge and taste
  - Raw data is better than interpreted data

†Olson, T., Loquist, J., October 2010
http://youtube-global.blogspot.com/2010/10/if-picture-is-worth-thousand-words-then.html
3. Architecture & Prototype
QuiltView Service

Global catalog owned by QuiltView

- Query
- Video
- User

QuiltView Service

Internet

1. Is the party exciting?

2. Metropolitan area network

3. People at the party

4. YouTube

5. User
Avoid Too Many Queries

Result caching
- “Similar” queries within a short time

Opt-in preferences
- Max queries/unit time
- Topics

Random sampling
- Choose a subset from eligible users

Credit system
- Similar to that in AMT
- Prevent spam users
Prototype – Server & Client

- Server: Amazon EC2, Django
- Client: GDK, background service
Prototype – Similarity Detection

- Semantic similarity – Gensim
  - Open-sourced python package
  - 9GB compressed corpus of the entire English Wikipedia

<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>
1. Background
2. QuiltView Design
3. Architecture & Prototype
4. Discussions
Queries – What Can QuiltView Do?

**Good Examples** – local, near-real-time, subjective

- Is the situation serious?
- How’s the free food?
- How long is the queue?
- How is the traffic?

**Bad Examples** – large-area, long-term facts, objective

- What is the population in the US?
Privacy

Share only with friends (e.g. Facebook)

Automatic denaturing (removing face, etc.)

Glass screen turned on while recording

Explicit confirmation by the Glass wearer
Missing in the Talk

- Restriction on queries
- SSL
- Mozilla Persona
- Inaccuracy in query similarity detection
- Virtual user for simulation
- Related work
Micro-interactions have great potential

QuiltView offers a valuable use

Interesting questions to be explored

Thank you! Questions?
Contact: zhuoc@cs.cmu.edu