Module 4

ACM Multimedia 13, Barcelona, Catalunya, Spain
October 21-25, 2013

Social Interactions over Geographic-Aware Multimedia Systems

Roger Zimmermann & Yi Yu

Media Management Research Lab
School of Computing
National University of Singapore
Outline: Module Four

I. GeoVid Resources
II. MPEG-DASH Overview and Resources
III. Google Street View Hyperlapse
IV. Synthetic Video Meta-Data Generation
V. Conclusions
GeoVid Architecture (1)
GeoVid Architecture (2)

• Sensor-rich video management design
  – Upload interface
  – Fast content availability
  – Spatio-temporal query support
  – RESTful GeoVid APIs
  – Support for HTML5 video streaming standard
  – Ready for MPEG-DASH video streaming standard
  – Mobile apps (iOS and Android)
GeoVid API (1)

- URL: http://api.geovid.org

GeoVid RESTful APIs

This page describes a variety of API specifications we have designed and implemented so far.

Version 1.0

Official version.

1. API descriptions are here

2. Sample application:
   - GeoVid Viewer on Google Maps
   - GeoVid Grid Viewer
   
   NOTE: please note that http://geovid.org/exp/viewer is outdated.
GeoVid API (2)

• GeoVid API Descriptions

GeoVid APIs version 1.0

Maintained by Beamjoo Seo
Last Modified on Dec., 19, 2012

HISTORY

• Oct., 18, 2012
  • Support GeoVid Metadata Representation Format 1.1
    • aware of different device orientation during video recording
    • time fix by satellite GPS signals
GeoVid API (3)

- Ex.: Search Service

SEARCH SERVICE

You can search videos, using the following geospatio-temporal query APIs.

- RESPONSE FORMAT (Common to every query API)

The response of all search queries has the same format. It consists of the array of playable video unit and is returned as a JSON (JavaScript Object Notation).

```json
RESPONSE: {
  "results": [
    {
      "video_id",
      "start_time",
      "end_time",
      "total_duration",
      "latitude",
      "longitude",
      "description"
    }
  ]
}
```
Outline: Module Four

I. GeoVid Resources
II. MPEG-DASH Overview and Resources
III. Google Street View Hyperlapse
IV. Synthetic Video Meta-Data Generation
V. Conclusions
Dynamic Adaptive Streaming over HTTP (DASH)

• Traditional media streaming (e.g., based on RTP/RTSP/RTCP) streaming faces several challenges in large-scale systems:
  – Special-purpose servers/proxies/peers for media (complex)
  – Difficulties in firewall traversal: protocols use both TCP and UDP transmissions
  – Difficulties in caching data (no “web caching”)

• Advantage of RTP/RTSP/RTCP
  – Short end-to-end latency
DASH Standard (Dec. 2011)

• ISO/IEC Standard:
  – “Information technology — MPEG systems technologies
    — Part 6: Dynamic adaptive streaming over HTTP
      (DASH)”
  – JTC 1/SC 29; FCD 23001-6

• Uses HTTP protocol to “stream” media

• Divides media into small chunks, i.e., streamlets

• Makes playback adaptive
  – Encode media into multiple different streamlet files, e.g., a
    low, medium, and high quality version (different
    bandwidth).

• Supported by: Move Networks, Apple, Microsoft, Netflix, …
DASH Limitations

- DASH standard only describes server-to-client dissemination:
  - Upload mechanism is undefined!

- We are working on a low latency DASH upload protocol for mobile devices.
DASH Upload Design

Mobile device

Segment i

Segment i+1

Segmentation

Video

HTTP POST

Server cluster

Segment i

Transcoding

Media descriptor

High
Medium
Low

Distribute

Edge server

DASH player

DASH

High
Medium
Low
Source Video Stream

- 720p
- 480p

Characteristics of the source video stream

<table>
<thead>
<tr>
<th></th>
<th>Video Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>1280 × 720</td>
</tr>
<tr>
<td>Frame rate</td>
<td>29.97</td>
</tr>
<tr>
<td>Overall bitrate</td>
<td>12.1 Mbps</td>
</tr>
<tr>
<td>Format</td>
<td>H.264/AVC, AAC</td>
</tr>
<tr>
<td>Duration</td>
<td>2 min 49 s</td>
</tr>
<tr>
<td>File size</td>
<td>244 MB</td>
</tr>
<tr>
<td>Sync interval</td>
<td>1 sec</td>
</tr>
<tr>
<td></td>
<td>720 × 480</td>
</tr>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>2 Mbps</td>
</tr>
<tr>
<td></td>
<td>H.263, NB-AMR</td>
</tr>
<tr>
<td></td>
<td>2 min 21 s</td>
</tr>
<tr>
<td></td>
<td>34 MB</td>
</tr>
<tr>
<td></td>
<td>1 sec</td>
</tr>
</tbody>
</table>
**Evaluation Metrics** $T_{\text{startup}}$

\[ T_{\text{startup}} \approx T_{\text{seg}} + T_{\text{upload}} + T_{\text{server}} + T_{\text{player}} \]

**Different delay components and their relationships.**
Segmentation Delay $T_{seg}$

- 720p
- 480p

Segment duration: 3s  
Segment duration: 10s

Normalized median segmentation time (processing time/segment duration).

<table>
<thead>
<tr>
<th>Video Quality</th>
<th>Android</th>
<th>iOS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Motorola Droid</td>
<td>Samsung Galaxy S</td>
</tr>
<tr>
<td>480p</td>
<td>0.89</td>
<td>0.42</td>
</tr>
<tr>
<td>720p</td>
<td>5.74</td>
<td>2.14</td>
</tr>
</tbody>
</table>

Values less than 1 indicate that a device is capable of creating a segment within playback duration, which means continuous processing of long videos in a pipelined manner is possible.
Segmentation Overhead

- 720p

Normalized segmentation delay of 720p video on iPhone 4.

Real-time segmentation of high-quality video will be feasible in any mobile platform in the future.
Network delivery of 480p is quite possible in a stable wireless environment, while 720p video, even with small segment duration, experience network bottleneck.
MPEG-DASH Resources

• Good introduction and overview:

• Dataset:

• ACM MMSys web site: [http://www.mmsys.org/](http://www.mmsys.org/)

• DASH-JS
  – An open source reference DASH player written in JavaScript from the DASH Industry Forum.
Outline: Module Four

I. GeoVid Resources
II. MPEG-DASH Overview and Resources
III. Google Street View Hyperlapse
IV. Synthetic Video Meta-Data Generation
V. Conclusions
Google Street View Hyperlapse

• Google Street View consists of geo-referenced, panoramic images.

• Project: Creating a time-lapse video from Google Street View images.
  – JavaScript hyper-lapse utility for Google Street View.
  – http://hyperlapse.tllabs.io/
Hyperlapse Demo Video

- [http://vimeo.com/63653873](http://vimeo.com/63653873)
Outline: Module Four

I. GeoVid Resources

II. MPEG-DASH Overview and Resources

III. Google Street View Hyperlapse

IV. Synthetic Video Meta-Data Generation

V. Conclusions
Real-World Video Collection

“Capture the sensor inputs and fuse them with the video streams”

- Recorded 134 video clips using the recording prototype system in Moscow, ID (total 170 mins video).
- Videos covered a 6km by 5km region quite uniformly.
- Average camera movement speed was 27km/h, and average camera rotation was around 12 degrees/s.
- Collected meta-data included 10,652 FOV scenes in total.
The collected real-world video data has not been large enough to evaluate realistic applications on the large scale.

- Collecting real-world data requires considerable time and effort.

A complementary solution is to synthetically generate georeferenced video meta-data.
Synthetic Video Meta-data Generation

**Input:** Camera Template Specification

```
<table>
<thead>
<tr>
<th>TIGER/Line Files</th>
<th>The Brinkhoff Algorithm</th>
<th>The GSTD Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Merge Trajectories</td>
<td></td>
</tr>
<tr>
<td></td>
<td>network-based movement</td>
<td>mixed movement</td>
</tr>
<tr>
<td></td>
<td>free movement</td>
<td></td>
</tr>
</tbody>
</table>
```

**Camera direction computation**

```
<table>
<thead>
<tr>
<th>Calculate Moving Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjust Directions on Turns</td>
</tr>
</tbody>
</table>
```

**Output:** Georeferenced Video Meta-data
Camera Movement Computation (1)

Network-based Movement

 Cameras move on a road-network
 Adopted the Brinkhoff algorithm for camera trajectory generation
 Introduced stops and acceleration/ deceleration events at some road crossings and transitions
 Camera accelerates with a constant rate (user defines the acceleration rate)
 In a deceleration event reduction in camera speed is simulated based on the Binomial distribution

\[ v_{\text{next}} = v_{\text{prev}} \times \frac{B(n, p)}{n} \]

When \( n=20 \) and \( p=0.5 \) speed is reduced to half at every time instant
Camera Movement Computation (2)

Free Camera Movement

- Cameras move freely.
- Improved the GSTD algorithm to generate the camera trajectories with unconstrained movement:
  - Added speed control mechanism
  - Camera movement data is generated in geographic coordinate system (i.e., as latitude/longitude coordinates)
Camera Movement Computation (3)

Mixed Camera Movement

Cameras sometimes follow the network and sometimes move randomly on an unconstrained path.

i. Generate a network based trajectory ($T_{init}$)

ii. Randomly select $n$ sub-segments ($S_1$ through $S_n$) on the trajectory

$$0 \leq |S_i| \leq (T_{init}/4) \text{ and } N_{rand} = (\sum S_i) / |T_{init}|$$

(user defines $N_{rand}$)

i. Replace $S_i$ with $T_{rand}(i)$

ii. Update timestamps
Assigning meaningful camera direction angles is one of the novel features of the proposed data generator.

**Fixed camera:**
1) Calculate moving direction
2) Adjust directions on turns

**Random rotation camera:**
1) Calculate moving direction
2) Adjust directions on turns
3) Randomize direction angles

**Output:** Georeferenced Video Meta-data
Camera Rotation Computation (2)

Fixed Camera

1) Calculate moving direction

\[ \overrightarrow{m} = T_k(t-1) \overrightarrow{T_k(t)} \]

\( \overrightarrow{m} \) is the moving direction vector at time \( t \)

\[ T_k(t).dir = \cos^{-1}\left( \frac{\overrightarrow{m}.y}{|\overrightarrow{m}|} \right) \]

2) Adjust directions on turns

Rotation angle from \( t_1 \) to \( t_2 \) is larger than \( \omega_{max} \) (i.e., rotation threshold)

Smooth down the rotation by distributing the rotation amount forwards and backwards
Camera Rotation Computation (3)

Fixed Camera

Real-world data

Synthetic data before direction adjustment

Synthetic data after direction adjustment

Illustration of camera direction adjustment for vehicle cameras
Camera Rotation Computation (3)

Randomly Rotating Camera

1) Calculate moving direction
2) Adjust directions on turns
3) Randomize direction angles
   - Randomly rotate the directions at each sample point towards left or right
   - Rotation amount is inversely proportional to the current camera speed level
   - The rotation amount is guaranteed to be less than rotation threshold $\omega_{\text{max}}$
Experimental Evaluation (1)

Goal: Evaluate the effectiveness of the synthetic data generation approach through a high level comparison between the real-world and synthetic data.

Datasets:
- Generated two groups of synthetic data:
  1. Using vehicle camera template
  2. Using passenger camera template
- Both synthetic data groups were created based on the road network of Moscow, ID.

Methodology:
- Analyze and compare the movements and rotations of real-world and synthetic datasets.
- Report:
  1. The average and maximum values for speed and rotation
  2. Frequency distribution of different speed and rotation levels
Experimental Evaluation (2)

**Comparison of Camera Movement Speed**

<table>
<thead>
<tr>
<th></th>
<th>Maximum speed (km/h)</th>
<th>Average speed (km/h)</th>
<th>StdDev of speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic data with fixed camera</td>
<td>87.91</td>
<td>27.14</td>
<td>12.82</td>
</tr>
<tr>
<td>Synthetic data with free camera rotation</td>
<td>87.28</td>
<td>27.32</td>
<td>13.01</td>
</tr>
<tr>
<td>Real-world data</td>
<td>0.564</td>
<td>27.03</td>
<td>13.68</td>
</tr>
</tbody>
</table>

Characteristics of the camera speed

Illustration of camera movement speed on map

Comparison of camera speed distributions for real-world data and synthetic data with fixed camera.
Experimental Evaluation (3)

Comparison of Camera Rotation

<table>
<thead>
<tr>
<th></th>
<th>Maximum rotation(degrees/s)</th>
<th>Average rotation(degrees/s)</th>
<th>StdDev of rotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic data with fixed camera</td>
<td>32.33</td>
<td>4.64</td>
<td>7.24</td>
</tr>
<tr>
<td>Synthetic data with free camera rotation</td>
<td>55.27</td>
<td>12.59</td>
<td>9.35</td>
</tr>
<tr>
<td>Real-world data</td>
<td>107.30</td>
<td>11.53</td>
<td>14.02</td>
</tr>
</tbody>
</table>

Characteristics of the camera rotation ($\omega_{\text{max}} = 60$ degrees)

Comparison of camera rotation distributions for real-world data and synthetic data with fixed camera

Illustration of camera rotation on map

Comparison of camera rotation distributions for real-world data and synthetic data with random camera rotation

Illustration of camera rotation on map
Experimental Evaluation (4)

**Performance Issues**

- The measured data generation times for different types of datasets and parameter settings.

<table>
<thead>
<tr>
<th>Camera Template</th>
<th>Trajectory Pattern</th>
<th>Rotation Pattern</th>
<th>Number of Videos</th>
<th>Time to Generate camera trajectory (s)</th>
<th>Time to Assign Directions</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle camera</td>
<td>$T_{\text{network}}$</td>
<td>Fixed camera</td>
<td>2,980</td>
<td>124</td>
<td>39</td>
<td>163</td>
</tr>
<tr>
<td>Passenger camera</td>
<td>$T_{\text{network}}$</td>
<td>Random rotation</td>
<td>2,980</td>
<td>115</td>
<td>201</td>
<td>316</td>
</tr>
<tr>
<td>Pedestrian camera</td>
<td>$T_{\text{free}}$</td>
<td>Random rotation</td>
<td>2,970</td>
<td>32</td>
<td>263</td>
<td>255</td>
</tr>
<tr>
<td>Pedestrian camera</td>
<td>$T_{\text{mixed}}$</td>
<td>Random rotation</td>
<td>2,970</td>
<td>271</td>
<td>215</td>
<td>486</td>
</tr>
</tbody>
</table>

- The generator can create synthetic datasets in a reasonable amount of time with off-the-shelf computational resources.
Summary

Proposed a two step synthetic data generation
1. Computation of the camera movements
2. Computation of the camera movements

Compared the high-level properties of the synthetically generated data and those of real-world geo-referenced video data.

The synthetic meta-data exhibit equivalent characteristics to the real data, and hence can be used in a variety of mobile video management research.
Conclusions

• Annotation using sensors can provide *automatic* and *objective* meta-data for indexing and searching.
• Georeferenced video search has a great potential, especially in searching user generated videos.
• Many open questions:
  – Standard format of meta-data
  – Standard way of embedding meta-data
  – Index structures of meta-data for fast searching
  – Supporting new query types
  – Combining with content based features
  – Relevance ranking for result presentation
Future Work

• Sensor values are sometimes noisy and contain errors.
• Better alignment of video screen position with mirror world.
• Support for multiple videos.
• When to rotate the video plane and when to move the mirror world viewpoint.
• Addition of 3D query facility.
Acknowledgment

© Yi Yu and Roger Zimmermann

This research has been supported in part by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

Some of the work has been performed under the Centre of Social Media Innovations for Communities (COSMIC).
End of Module Four

Thank You – Q&A

Further information at:

http://geovid.org

rogerz@comp.nus.edu.sg
yuy@comp.nus.edu.sg