Augmenting Aerial Earth Maps with Dynamic Information

Kihwan Kim, Sangmin Oh, Jeonggyu Lee and Irfan Essa
School of Interactive Computing
Georgia Institute of Technology
Motivation

+ • Nice quality of 3D buildings
• Realistic Aerial Textures
• Fly around anywhere
• Annotated information
• Pictures, Videos

− • Everything is static
• No dynamic information

Virtual Earth Maps (Google Earth)
Motivation

Our Goal is:

Putting Dynamic Information onto Static Earth Map

Then, making it come ALIVE...
Overview

• Resources in current Virtual Earth..

Update interval:
: almost 4~6 months

Hard to interpolate
 temporal information!
Overview

• What if we make use of distributed videos*?

Easier to extract dynamism and spatial correlation.

But methods should vary!

* Public cameras, sport footage, individual videos
Overview

- Four different Scenarios
  - Traffic Visualization
  - Sports Visualization
  - Sky/Clouds Visualization
  - Pedestrian Visualization

- Sparsity of Camera
- Motion Complexity
Scenario 1: Pedestrian

Direct Mapping: Single view, Limited motion

Animating pedestrian using Motion Capture Data*

* Carnegie Mellon MoCap database
Scenario 1: Pedestrian

Direct Mapping: Single view, Limited motion

- Tracking (Velocity, Pos)
- Animating pedestrian using Motion Capture Data*
- Mapping using Homography

*Carnegie Mellon MoCap database
Scenario 2 : Sports Visualization

Dynamic sports scenes :

Complex motion*          Overlapping cameras

→ View Synthesis

*Efros et al 2003
Scenario 2: Sports Visualization

Structure and Views:

\[ \theta \geq \alpha \]

\[ \theta \cong 0 \]

Rectified video
Scenario 2: Sports Visualization

- Five views and their optimal virtual views

View 1
View 2
View 3
View 4
View 5

Plane on the Aerial Earth Map
Scenario 2: Sports Visualization

• In an arbitrary view → Globally blended views

\[ p_v = f(\omega_i)p_i + g(\omega_i)p_{i+1} + \omega_{bkg}p_{bkg} \]
Scenario 3: Traffic Visualization

Traffic cameras
Sparsely distributed cameras,
Simple motion
(Follows the road)
Scenario 3: Traffic Visualization

Extraction:

(1) Tracking Cars*

(2) Registration
Vanishing Pts**
Structure for plane patch

*Veenman01, Shi/Tommasi94, **Kosecka02
Scenario 3: Traffic Visualization

Construct nodes:
Estimate velocity & flow of the unobserved node from observed node

[Diagram showing observed and unobserved nodes with video observations]
Scenario 3 : Traffic Visualization

Simulating cars with the velocity of each node and behavioral models (flock, steering)*

\[
V_{i,k} = \alpha V_{i}^{flock} + \beta V_{i}^{steer} + \gamma V_{k}^{node}, \quad V_{node} = \sum_{j=0}^{n} \alpha_j (V_{j}^{node})
\]

* Reynolds (SIGGRAPH87, GDC99)
Scenario 3: Traffic Visualization - Results
Scenario 4: Sky and Clouds

Sparse cameras

Complex motion

(Particle Model)
Scenario 4: Sky and Clouds

Interpolating motion/texture using Radial Basis Function (RBF)

Data-driven, procedurally rendered clouds
Scenario 4: Sky and Clouds

- Projecting clouds maps onto sky dome
Results

• Putting it all together
  (Q9300 Quad-core 2.53GHz : ~ 10-15 fps)
Contributions

Novel framework to visualize dynamism

- **Global view-blending**: sports scene
- **Parameterized behavioral model**: traffic
- **Vision driven procedural rendering of clouds**: sky and clouds
Possible Applications

• Augmented online map services
• Interactive sports broadcasting
• Virtual environment or game has a metaphor of the real world (Second-life)
• New marketing opportunities
Conclusion

Limitations:

• Algorithmic Trade-offs
• High-level traffic events (e.g., accidents, traffic light)
• Direct Mapping only covers given FOV

Future work:

• Aim to overcome above limitations
• Apply to other situations (scenarios)
Thank you!

More information:

http://www.cc.gatech.edu/cpl/projects/augearth/
Related work

- AVE System (SentinelAVE LLC.)
- Skyline Globe (Skyline Soft)
- Video Flashlight
- GooPS system

And lots of literatures in Vision, Graphics and animation field
(To refer at the end of presentation)
Scenario 2: Sports Visualization

Globally interpolating views

1. Select most closest views
2. Weighted sum of each view with backgrounds
3. Back-projects it

\[ p_v = f(\omega_i)p_i + g(\omega_i)p_{i+1} + \omega_{bkg}p_{bkg} \]
Conclusion

Privacy Issues

• Detection and limited category recognition (Objects are symbolized)
• Sports visualization : Copyright
• Friendly games from individual videos : Publish in public or private (Youtube)