BUPT-MCPRL@TRECVID 2014: Surveillance Event Detection (SED)

Qi Chen (chen_qi1990@163.com)
Zhicheng Zhao, Wenhui Jiang, Jinlong Zhao, Yuhui Huang,
Xiang Zhao, Lanbo Li, Yanyun Zhao, Fei Su, Anni Cai
BUPT-MCPRL
Beijing University of Posts and Telecommunications
## Our Submission

- **BUPT_MCPRL 2014 Retrospective Result**

<table>
<thead>
<tr>
<th>Event</th>
<th>Rank</th>
<th>ADCR</th>
<th>ADCR of Other Best Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embrace</td>
<td>2</td>
<td>0.8318</td>
<td>0.8113</td>
</tr>
<tr>
<td>PeopleMeet</td>
<td>4</td>
<td>1.0354</td>
<td>0.8587</td>
</tr>
<tr>
<td>PeopleSplitUp</td>
<td>4</td>
<td>0.9476</td>
<td>0.8353</td>
</tr>
<tr>
<td>PersonRuns</td>
<td>4</td>
<td>0.9070</td>
<td>0.8256</td>
</tr>
<tr>
<td>Pointing</td>
<td>1</td>
<td>0.9998</td>
<td>1.0027</td>
</tr>
</tbody>
</table>
Outline

• Retrospective System Overview
• Pedestrian Detection
• Pedestrian Tracking
• Detected by CNN
  – Embrace and Pointing
• Detected by Trajectory Analysis
  – PeopleMeet and PeopleSplitUp
  – PersonRuns
• Performance Evaluation
• Conclusion
Retrospective System Overview

- Pedestrian Detection by CNN
- Pedestrian Tracking
- People Meet, People Split Up, and Person Runs Detection
- Embrace and Pointing Detection
- Trajectory Analysis

Events Fusion

Detections
Pedestrian Detection

- Pedestrian Detection by Head-Shoulder-CNN
  -- suppress the effect of partial occlusion
Pedestrian Detection

• The Architecture of Our CNN
  – much smaller than Krizhevsky’s network
    [Krizhevsky, NIPS 2012]
Pedestrian Detection

• Samples
  – from TrecVid08-Dev_set and TrecVid08-Eval_Set
  – positive
    • 11,538 for training
    • 4,946 for testing
    • randomly horizontal flipping
  – negative:
    • anything of non-positive
    • three times the number of positive

• Details of Training
  – single NVIDIA GTX 780Ti GPU
  – Core i7 desktop CPU
  – 3 hours for training
  – learning rate : 0.01
Pedestrian Tracking

• Multi-Target Tracking [Bo Yang et al. CVPR 2013]
  – online approach to learn non-linear motion patterns and robust appearance models
  – deal with detection result with long gap
  – more robust for tracking with lots of occlusion
Pedestrian Tracking

- We Propose to use Gaussian process regression to smooth the trajectory.

Detection responses $x$

Detection responses $x$ and the true trajectory $t$

The relationship $\Pr(w|x)$ between the response $x$ and point $w$ of $t$

Unsmoothed trajectories

Smoothed trajectories
Outline

• Retrospective System Overview
• Pedestrian Detection
• Pedestrian Tracking
• Detected by CNN
  – Embrace and Pointing
• Detected by Trajectory Analysis
  – PeopleMeet and PeopleSplitUp
  – PersonRuns
• Performance Evaluation
• Conclusion
Embrace and Pointing

• Regard the events detection as the detection of key-poses

• Key-poses for Embrace and Pointing
Embrace and Pointing

• Method
  – adopt CNN to recognize the key-pose
  – use the architecture of pedestrian detection
  – the inputs of models are the pedestrian detection results with 1.5-fold expansion

The architecture of our CNN
Embrace and Pointing

• Samples
  – from TrecVid08-Dev_set and TrecVid08-Eval_Set
  – positive
    • total : 2100
    • randomly cropping
    • randomly horizontal flipping
    • RGB jittering
  – negative
    • any pedestrian detection results of non-Embrace or non-Pointing
    • three times the number of positive

• Details of Training
  – single NVIDIA GTX 780Ti GPU
  – Core i7 Desktop CPU
  – 2 hours for training
  – learning rate : 0.01
# Embrace and Pointing

## retro-Embrace

<table>
<thead>
<tr>
<th>Years</th>
<th>ADCR</th>
<th>MDCR</th>
<th>#CorDet</th>
<th>#FA</th>
<th>#Miss</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014</td>
<td>0.8318</td>
<td>0.8318</td>
<td>26</td>
<td>44</td>
<td>112</td>
</tr>
<tr>
<td>2013</td>
<td>1.0503</td>
<td>0.9850</td>
<td>13</td>
<td>380</td>
<td>162</td>
</tr>
</tbody>
</table>

## retro-Pointing

<table>
<thead>
<tr>
<th>Years</th>
<th>ADCR</th>
<th>MDCR</th>
<th>#CorDet</th>
<th>#FA</th>
<th>#Miss</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014</td>
<td>0.9998</td>
<td>0.9910</td>
<td>21</td>
<td>57</td>
<td>774</td>
</tr>
<tr>
<td>2013</td>
<td>1.6387</td>
<td>1.0064</td>
<td>219</td>
<td>2576</td>
<td>844</td>
</tr>
</tbody>
</table>
Outline

• Retrospective System Overview
• Pedestrian Detection
• Pedestrian Tracking
• Detected by CNN
  — Embrace and Pointing
• Detected by Trajectory Analysis
  — PeopleMeet and PeopleSplitUp
  — PersonRuns
• Performance Evaluation
• Conclusion
PeopleMeet and PeopleSplitUp

- **PeopleMeet**
  - split into 3 subevents: walking closely, slowing down and stay
  - use HMM (Hidden Markov Model) to model the event [Chan et al. ICPR 2004]
  - observe every two persons based on their trajectories
  - the distances between persons and their speed are used as features to construct observation sequence

- **PeopleSplitUp**
  - split into 3 subevents: stay, speeding up, walking away
  - similar to the detection of PeopleMeet
PersonRuns

• Distinguish running trajectories
  – pick the fast-moving pedestrian tracks by Forward-backward Motion History Image (MHI) [Z Yin et al. AVPI 2009]
  – $FB-MHI = F-MHI \& B-MHI$
  – set a threshold of the ratio of non-zero pixels in the region of the pedestrian detection result
## Performance Evaluation

<table>
<thead>
<tr>
<th>Event</th>
<th>Rank</th>
<th>ADCR of Other Best Systems</th>
<th>BUPT_MCPRL 2014 Retrospective Result (Update Version)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>ADCR</td>
</tr>
<tr>
<td>Embrace</td>
<td>2</td>
<td>0.8113</td>
<td>0.8318</td>
</tr>
<tr>
<td>PeopleMeet</td>
<td>4</td>
<td>0.8587</td>
<td>1.0354</td>
</tr>
<tr>
<td>PeopleSplitUp</td>
<td>4</td>
<td>0.8353</td>
<td>0.9476</td>
</tr>
<tr>
<td>PersonRuns</td>
<td>4</td>
<td>0.8256</td>
<td>0.9070</td>
</tr>
<tr>
<td>Pointing</td>
<td>1</td>
<td>1.0027</td>
<td>0.9998</td>
</tr>
</tbody>
</table>

- **Method of CNN**
  - *Embrace and Pointing*
  - works very well
- **Method of Trajectory Analysis**
  - *PeopleMeet, PeopleSplitUp and PersonRuns*
  - not good
Conclusion

• We proposed the methods of CNN and trajectory analysis for event detection

• Method of CNN
  – works very well
  – detects a small number of false alarms and a relatively big number of correct detections
  – much less computations
  – easy to implement

• Method of trajectory analysis
  – not good
  – difficult to get the true information such as velocity
Thanks!

www.bupt-mcprl.net