Effective detection of moving objects, shadows and ghosts in surveillance videos

R. Cucchiara, C. Grana, A. Prati
Università di Modena e Reggio Emilia, Italy
e-mail: rita.cucchiara@unimo.it, andrea.prati@unimo.it, grana@dsi.unimo.it

M. Piccardi (speaker)
Computer Vision Research Group (CVRG),
University of Technology, Sydney (UTS)
e-mail: massimo@it.uts.edu

Australia-Japan Advanced Workshop on Computer Vision
Adelaide, 9-11 September 2003
Moving object detection

- Detection of moving objects is the basic step of any video surveillance task
- In the literature, the different methods fall into three main classes:
  - optical flow segmentation
  - frame differencing
  - background subtraction
- For static cameras, *background subtraction* is probably the most popular
Background subtraction

- Background subtraction methods detect moving objects by:
  - computing an estimate of the scene’s static background (*background model*)
  - “subtracting” the background model from the current frame

- The background model must be automatically initialised and updated
Background subtraction: problems

- Illumination changes
  - gradual
  - sudden (such as clouds)
  - shadows

- Motion changes
  - camera oscillations
  - high-frequencies background objects (such as tree branches)

- Changes in the background geometry
Background subtraction methods: a quick review

- **Pfinder** (Wren, Azarbayejani, Darrell, Pentland, 1997):
  - background is estimated independently for each image pixel \((x,y)\) (no spatial correlation used)
  - background initialisation:
    - for each pixel, Gaussian \((\mu, \sigma)\) fitting on the previous \(N\) values of the pixel
  - background update:
    \[
    \mu_t = \alpha I + (1 - \alpha) \mu_{t-1}
    \]
Background subtraction methods: a quick review (2)

- Such an adaptive model copes with gradual illumination changes.
- Sudden illumination changes can be detected and model reinitialised.
- However:
  - A single-Gaussian model cannot cope with high-frequency background objects.
  - Background update should exclude moving objects’ pixels (*selective* update).
  - Shadows an open problem.
Background subtraction methods:
a quick review (3)

- Stauffer and Grimson (1999):
  - *mixture of multiple Gaussians* ($\mu_i, \sigma_i, \omega_i$) (typically from 3 to 5) to store pixel's history
  - in this way, it copes also with high frequency objects
  - *however*, update is heavier:
    - all weights $\omega_i$ are updated (updated and/or normalised)
    - some $\mu_i, \sigma_i$ are updated (those of “matched” distributions)
    - NB: $\mu_i$ is a 3-component vector (for the three colours)
    - all distributions are sorted according to a simple metric and the first ones chosen as “background”
Background subtraction methods: a quick review (4)

- Elgammal, Harwood, Davis (2000):
  - the background model is given by the histogram of $N$ previous values smoothed as a Kernel Density Estimator
  - selective update
  - KDE requires only a single value, $\sigma$, to be estimated
  - however:
    - subtraction requires computing $N$ Gaussian values, with $N$ typically 50÷100
      - much storage space
      - subtraction is time-consuming (lookup tables needed)
    - updating the KDE smoothing factor is a heavy process
Sakbot: the goal

Define a moving object detection method:

- **precise**
  - accurate (space)
    (few FP’s and FN’s at each frame)
  - reactive (time)
    (short transient errors)
- **flexible** on different applications
- **efficient** to meet real-time computation requirements

Why?
- Fast and precise tracking
- Correct classification
Main features

- a simple statistical model
- adaptive and selective
- uses global object knowledge

Moving visual object detection

- Camera motion correction
  - Corrected Frame $I_t$
  - Background Suppression
    - Background Distance $DB_t$
    - Shadows Detection
      - Labeling
        - Foreground Blobs $FB_t$
      - Blob Analysis
        - Moving Visual Objects $MVO_t$
      - Tracking
        - Scene understanding
          - Objects History
Statistical approach: median + adaptivity

- \( B_t = \text{median} (\{I_{t-1}, \ldots, I_{t-n}, \omega[B_{t-1}]\}) \)
- \( \omega[B_{t-1}] \): the previous background is accounted \( \omega \) times
- With \( \omega[B_{t-1}] \), FN (a) stable and FP (b) decrease significantly:
Use of colours

- Subtraction:
  \[ DB_t(x,y) = Distance(l_t(x,y), B_t(x,y)) = \]
  \[ = \max \left( |l_t(x,y).c - B_t(x,y).c|, \ c=R,G,B \right) \]

- Use of colours increases sensitivity: at a rough parity of FP (c), FN (d) are much diminished
Statistical and selective approach: knowledge of objects

- Global properties of objects such as the average optical flow are used to exclude non-background points from background update
- FP decrease again
Background subtraction: “ghosts”

- A ghost is an apparent object due to changes in the geometry of the background.
- With selective methods, there is the risk of a “deadlock”.
Dealing with ghosts

- Sakbot computes the average magnitude of the optical flow (AOF) on each foreground blob.
- If $AOF < Th$, the object is classified as a ghost.
- When a ghost “detaches” from its real object, it is immediately classified as a ghost.
- Risk: if the ghost area is immediately occupied by another real object, the ghost will not be detected.
Ghosts: reactivity

- Example: a car reverses in a carpark

1. A ghost appears at frame i

2. With a “normal” adaptive method, a ghost appears and stays

3. Even after many frames, the ghost is not yet fully eliminated

4. With Sakbot, the ghost is almost immediately eliminated
Ghosts: reactivity (2)

- Comparison of false positives and false negatives
  (*) adaptive method □ adaptive + selective ▲ Sakbot

The adaptive method converges slowly, the selective method falls into a “deadlock”, the proposed method eliminates the ghost rapidly

FN are approximately the same (reactivity is not achieved at the price of accuracy)
Shadows

- Shadows of moving objects cause problems
Shadow suppression

Detecting moving points belonging to *shadows*

A moving point is classified as SHADOW iff:

1) it is darker than the “shadowed” bkg

\[ \alpha < \frac{Y_p}{Y_{bkg}} < \beta \]

2) ratios between each color channel is approximately constant (Trivedi 00)

\[ \frac{R_p}{R_{bkg}} \equiv k_R \]
\[ \frac{G_p}{G_{bkg}} \equiv k_G \]
\[ \frac{B_p}{B_{bkg}} \equiv k_B \]
Shadow suppression in the HSV color space

\[
SP_t(x,y) = \begin{cases} 
1 & \text{if } \alpha \leq \frac{I_t(x,y).V}{B_t(x,y).V} \leq \beta \quad \wedge \quad |I_t(x,y).H - B_t(x,y).H| \leq \tau_H \\
\wedge \quad I_t(x,y).S - B_t(x,y).S \leq \tau_s \\
0 & \text{otherwise}
\end{cases}
\]
Improvement on trajectory detection with shadow suppression
### Application examples

<table>
<thead>
<tr>
<th></th>
<th>Shopping center</th>
<th>US highway</th>
<th>Parking zone</th>
<th>Laboratory</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of frames in the sequence</strong></td>
<td>2300</td>
<td>440</td>
<td>500</td>
<td>980</td>
</tr>
<tr>
<td><strong>Total objects detected</strong></td>
<td>140</td>
<td>70</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td><strong>Average number of pixels per MVO</strong></td>
<td>1950</td>
<td>7793</td>
<td>978</td>
<td>3240</td>
</tr>
<tr>
<td><strong>Average number of foreground pixels</strong></td>
<td>3082</td>
<td>13140</td>
<td>1422</td>
<td>4128</td>
</tr>
<tr>
<td><strong>Average MVO size in pixels</strong></td>
<td>1484</td>
<td>3241</td>
<td>2456</td>
<td>7228</td>
</tr>
<tr>
<td><strong>Number of frames of MVO presence</strong></td>
<td>92</td>
<td>19</td>
<td>490</td>
<td>48</td>
</tr>
<tr>
<td><strong>Frame size</strong></td>
<td>352x288</td>
<td>320x240</td>
<td>345x135</td>
<td>320x240</td>
</tr>
</tbody>
</table>
Examples of results

Conclusions

- An efficient algorithm for background subtraction based on a simple statistic, adaptivity, use of object knowledge, shadow suppression
- Used as a first step in surveillance applications
- Nearly frame rate for common video sizes

Open problems:
- Automatic adaption of shadow parameters
- Ghosts in crowded areas
Example video

- **Lab.avi**

  length: 1’28”, 10 fps, 320 x 240

  Moving object detection is performed at 27.7 fps on this PC (P4 2.0 GHz)