BACKGROUND SUBTRACTION USING SPATIO-TEMPORAL CONTINUITIES

Srenivas Varadarajan\textsuperscript{1}, Lina J. Karam\textsuperscript{1} and Dinei Florencio\textsuperscript{2}

\textsuperscript{1}Image, Video, and Usability (IVU) Lab
School of Electrical, Computer & Energy Engineering
Arizona State University, Tempe, Arizona, USA

\textsuperscript{2}Microsoft Research
One Microsoft Way, Redmond, Washington, USA

svarada2@asu.edu, karam@asu.edu, dinei@microsoft.com
ABSTRACT

We present a novel scheme for dynamically recovering a background image from consecutive frames of a video sequence based on spatial and temporal continuities. The proposed algorithm applies a boundary-level spatial continuity constraint in order to detect and correct ghosting, which corresponds to incorrectly classified foreground regions due to fast moving objects. The proposed method can be applied successfully to sequences with deformable foreground objects and non-uniform motion. Simulation results show that the extracted background, when used for foreground detection, results in a higher performance in terms of recall and precision as compared to existing popular schemes.
Outline

1. Motivation / Applications
2. Introduction
   - Existing Approaches for Foreground Detection
   - Automatic Occlusion detection
3. Proposed Algorithm
4. Simulation Results and Analysis
5. Conclusion
Motivation 1: Background Replacement
Motivation 2: 3D effects
Motivation 3: Privacy
Google Street View Privacy Probe Joined by Spain, Italy, France

May 19, 2010, 6:10 PM EDT

By Stephanie Bodoni

May 19 (Bloomberg) -- Google Inc., under investigation in Germany for the data-gathering practices of its Street View mapping service, now faces probes in Spain, France, and Italy for possible violation of privacy laws.

Spain's Data Protection Authority today ordered an
2. Introduction
Can you tell foreground vs. background?
# Existing Approaches for Foreground Detection

<table>
<thead>
<tr>
<th>Approach</th>
<th>Description</th>
<th>Drawbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model-based</td>
<td>Parametric or non-parametric models are fitted to the background and/or to the foreground pixels. Depending on the deviation from these models, a pixel is classified as foreground or background</td>
<td>Optimal number of Gaussians and the learning rate cannot be set a priori for all situations</td>
</tr>
<tr>
<td>Stauffer et al. [1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elgammal et al. [2]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature-based</td>
<td>A feature like color, edges, motion and texture is used for classification</td>
<td>Edge and texture-based approaches fail when the background is smooth</td>
</tr>
<tr>
<td>Z. Wu et al. [6]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pixel-differencing-based</td>
<td>The difference of co-located pixels in adjacent frames is compared with a threshold in order to detect the foreground objects</td>
<td>Ghosting and Foreground-Aperture problems due to fast moving and large uniform foreground objects, respectively</td>
</tr>
<tr>
<td>S.Varadarajan et al.[12]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other Approaches</td>
<td>Loopy belief propagation [4] and those based on fuzzy integrals which combine a set of features</td>
<td>1. Smoothness assumption fails at object transitions in the background</td>
</tr>
<tr>
<td>Xun Xu et al. [4]</td>
<td></td>
<td>2. High complexity multi-pass algorithms</td>
</tr>
<tr>
<td>F. El Baf et al. [5]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Can you tell foreground versus bkgnd?
Can you tell foreground versus bkgnrd?
Algorithm for automatically detecting which object is in front

The inputs
Occluded areas
Unoccluded
Frame 1

Thresholded Frame Difference

Inner and Outer Contours of each region
3. Proposed Algorithm
Assumptions

• A **static background** and a **moving foreground** is assumed. Both the background and foreground may consist of several objects.

• Since the background is static, it exhibits temporal continuity, i.e., co-located background pixels in adjacent frames have similar values.

• We do NOT assume background is unoccluded most of the time.
Proposed Spatio-Temporal Continuity-based Background Subtraction algorithm

- Occlusion detection by hysteresis classification
- Outlier removal
- Occluding Components formation
- Ghosting removal based on spatial continuity
- Temporal continuity based recovery
- Partial Ghosting removal based on spatial continuity

For each Occluding Component
Foreground-Background Hysteresis Classification [12]

- Initial foreground guess based on initial $N$ frames.
- First $N$ consecutive frames are low pass filtered. ($N = 5$ in our implementation). Then, each color pixel is classified as
  - Strong Foreground ($SF$),
  - Weak Foreground ($WF$), or
  - Background ($B$)

$$C_{x,y,2} = \begin{cases} 
  SF, & \text{if } \|P_{x,y,2} - P_{x,y,n}\|_{L_2} > t_2 \text{ for any } n = 2, ..., N \\
  WF, & \text{if } t_1 < \|P_{x,y,2} - P_{x,y,n}\|_{L_2} < t_2 \text{ for any } n = 2, ..., N \\
  B, & \text{else} 
\end{cases}$$

Where $t_1$ and $t_2$ ($t_1 < t_2$) correspond to a low threshold and a high threshold ($t_1 = 3$ and $t_2 = 20$ in our implementation).
- Followed by outlier removal and incorporating WF into neighboring SF.

Occluding Components Formation

- Each moving foreground object corresponds to an Occluding Component (OC) and will be treated independently of others.
Ghosting Detection and Removal

Assume a fast moving object:
- \( X \) be the position of the box in the previous frame and \( Y \) be its position in the current frame. Both these regions are detected as background on pixel-differencing.

**Ghosting Detection:**
Compute a spatial discontinuity metric:

\[
D = \frac{1}{\# B^\text{in}_k} \sum_{(x, y) \in B^\text{in}_k} \| p_{x, y, n} - P_{\text{ClosestOut}} \|_1
\]

where
\( \# B^\text{in}_k \) the number of pixels in the inner boundary of the OC

**Ghosting Detection:**
If \( D < 5 \), the region is recognized as a background blob and replaced with the pixels of the current frame.
Partial-Ghosting Detection and Removal

Assume:
• There is a partial overlap of a foreground object across two successive frames.
• A is the portion of the background uncovered by the object but detected as foreground (ghost), while Y is the actual position of the object.

Partial-Ghosting Detection:
• The spatial continuity criterion is applied at a pixel level instead of the entire object’s boundary.
• The vertical and horizontal boundaries of each OC in the current frame are located by horizontal and vertical scans of the foreground mask, respectively.
• For a horizontal boundary pixel located at \((x,y)\), a horizontal discontinuity metric is computed as follows:

\[
D_H(x,y,n) = \| p_{x-1,y,n} - p_{x+1,y,n} \|_{L_1}
\]

• For a vertical boundary pixel located at \((x,y)\), a vertical discontinuity metric is computed as follows:

\[
D_V(x,y,n) = \| p_{x,y-1,n} - p_{x,y+1,n} \|_{L_1}
\]

Partial-Ghosting Removal:
If the computed \(D_H\) or \(D_V\) is below a threshold (equals 3 in our implementation), then the pixel is considered as a background pixel and recovered.
Temporal Continuity Based Recovery

Issues not solved by a system based only on spatial continuity:

• The assumption of a smooth background fails when the background contains sharp transitions
• These sharp background edges may coincide with edges of the foreground object, in which case the spatial continuity constraint fails.

Temporal Continuity Based Recovery:

Before the Partial Ghosting Removal step, the boundaries of the occluding foreground regions are updated by exploiting a temporal continuity constraint as follows:

\[ c_{x,y,n} = \begin{cases} B, & \text{if } \| P_{x,y,n} - P_{x,y,n-m} \|_2 < t_3 \text{ for all } m = 1, ..., M \\ SF, & \text{else} \end{cases} \]

where \((x,y)\) denotes the location of a boundary pixel of an OC in the current frame \(n\). In our simulation, the threshold \(t_3 = 6\) and \(M = 4\)
4. Simulation Results and Analysis
Progressive background extraction using spatio-temporal continuity for the 640x480 “Office” sequence

(a) Original Frame 25  
(b) Original Frame 88  
(c) Initial background after 6 frames  
(d) Background after 27 frames  
(e) Background after 28 frames (blob removed)  
(f) Extracted background after 63 frames
Performance Metrics for **Foreground** Detection

Recall \( = \frac{\text{Number of pixels correctly detected in the foreground}}{\text{Total number of pixels in the foreground given by Ground Truth}} \)

Precision \( = \frac{\text{Number of pixels correctly detected in the foreground}}{\text{Total number of pixels detected in the image as foreground}} \)
## Performance Evaluation: 640x480 Office sequence

<table>
<thead>
<tr>
<th>Method</th>
<th>Frame 30</th>
<th></th>
<th>Frame 50</th>
<th></th>
<th>Frame 70</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MoG[2]</td>
<td>0.226</td>
<td>0.756</td>
<td>0.530</td>
<td>0.761</td>
<td>0.619</td>
<td>0.735</td>
</tr>
<tr>
<td>Global Motion Comp. [9]</td>
<td>0.910</td>
<td>0.297</td>
<td>0.925</td>
<td>0.289</td>
<td>0.688</td>
<td>0.238</td>
</tr>
<tr>
<td>Block Motion Parameters [12]</td>
<td>0.778</td>
<td>0.912</td>
<td>0.957</td>
<td>0.619</td>
<td>0.975</td>
<td>0.423</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>0.986</td>
<td>0.927</td>
<td>0.993</td>
<td>0.772</td>
<td>0.956</td>
<td>0.692</td>
</tr>
</tbody>
</table>
Performance evaluation:
176 x 144 Hall Monitor sequence

<table>
<thead>
<tr>
<th>Method</th>
<th>Frame 40</th>
<th></th>
<th>Frame 50</th>
<th></th>
<th>Frame 60</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MoG[2]</td>
<td>0.531</td>
<td>0.558</td>
<td>0.599</td>
<td>0.549</td>
<td>0.586</td>
<td>0.554</td>
</tr>
<tr>
<td>Global Motion Comp. [9]</td>
<td>0.710</td>
<td>0.339</td>
<td>0.590</td>
<td>0.342</td>
<td>0.483</td>
<td>0.363</td>
</tr>
<tr>
<td>Block Motion Parameters [12]</td>
<td>0.726</td>
<td>0.692</td>
<td>0.736</td>
<td>0.693</td>
<td>0.770</td>
<td>0.691</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>0.700</td>
<td>0.711</td>
<td>0.719</td>
<td>0.685</td>
<td>0.768</td>
<td>0.686</td>
</tr>
</tbody>
</table>
The proposed algorithm consistently yields higher recall and precision rates.

- Mixture of Gaussians [2], absorbs foreground pixels into the background model.

- The loss of precision in the method based on global motion parameters [9] is due to background recovery at a block level instead of a pixel level.

- The method of background recovery based on motion parameters [12], performs well on the Hall Monitor sequence, but not on the Office sequence due to non-uniform motion.
5. Conclusions
Conclusion

• A new approach for background estimation/subtraction allowing:
  • Complete ghosting removal based on boundary-level spatial continuity constraints
  • Partial ghosting removal based on pixel-level spatio-temporal continuity constraints.

• Robust to non-uniform as well as uniform motion.

• Resilient to pauses in motion.

• Handles well deformable foreground objects and background clutter.
Questions ?
References