Motion Detection and Tracking

CS6350: Computer Vision

Indian Institute of Technology, Madras

Visualization and Perception Lab
Introduction

• **AIM:** To detect and track objects moving independently to the background

• Two situations encountered are
  – Static Camera (fixed viewpoint)
  – Moving Camera (moving viewpoint) (research topic, out of scope, left for exploration)
Definition of Motion Detection

• Action of sensing physical movement in a given area

• Motion can be detected by measuring change in speed or vector of an object
Applications of Motion Detection and Tracking

• Surveillance/Monitoring Applications
  – Security Cameras
  – Traffic Monitoring
  – People Counting

• Control Applications
  – Object Avoidance
  – Automatic Guidance
  – Head Tracking for Video Conferencing

Many intelligent video analysis systems are based on motion detection and tracking
Detecting moving objects in a static scene

• Moving objects can be detected by applying Background Subtraction Algorithms

• Simplest method (frame differencing):
  – Subtract consecutive frames
  – Ideally this will leave only moving objects
  – Following conditions effect the background subtraction
    • Moving background (e.g. swaying of trees)
    • Temporarily stationary objects
    • Object shadows
    • Illumination variation
Background Subtraction

- **Motivation:** Simple difference (frame differencing) of two images shows moving objects
- Uses a reference background image for comparison purposes
- Current image (containing target object) is compared to reference image pixel by pixel
- Places where there are differences are detected and classified as moving objects
<table>
<thead>
<tr>
<th>BGS Algorithm</th>
<th>Reference Paper</th>
<th>Salient Features</th>
</tr>
</thead>
</table>
  • Parametric thus less memory intensive |
  • It first models the person, then the scene and then does analysis |
  • Gaussians modify and adapt with each new incoming frame |
### Overview of Various BGS Algorithms (contd..)

<table>
<thead>
<tr>
<th>BGS Algorithm</th>
<th>Reference Paper</th>
<th>Salient Features</th>
</tr>
</thead>
</table>
• The Gaussian mixture parameters and components of each pixel is updated online |
| Eigenbackgrounds               | Oliver, N.M.; Rosario, B.; Pentland, A.P.; "A Bayesian computer vision system for modeling human interactions", IEEE Transactions on Pattern Analysis and Machine Intelligence , vol.22, no.8, pp.831-843, Aug 2000 | • PCA by way of eigenvector decomposiiion is a way to reduce the dimensionality of a space  
• PCA can be applied to a sequence of n frames to compute the eigenbackgrounds  
• Faster than MoG approach |
• Uses Median function |
Basic BGS Algorithms

• Background as the **average** or the **median** (Velastin, 2000; Cucchiara, 2003) of the previous $n$ frames:
  
  — rather fast, but very memory consuming: the memory requirement is $n \times \text{size(frame)}$

• Background as the Approximate Median Filtering (AMF) (**running average**)
  
  $$B_{i+1} = \alpha \times I_{i+1} + (1 - \alpha) \times B_i$$
  
  — $\alpha$, the learning rate, is typically 0.05
  
  — no more memory requirements
Basic BGS Algorithms – rationale

• The background model at each pixel location is based on the pixel’s recent history

• In many works, such history is:
  – just the previous $n$ frames
  – a weighted average where recent frames have higher weight

• In essence, the background model is computed as a chronological average from the pixel’s history

• No spatial correlation is used between different (neighbouring) pixel locations
Results - Simple frame differencing

INPUT Video

Foreground Mask
Results - Approximate Median Filtering

Background Model

Foreground Mask
Mixture of Gaussians (MoG)

• Mixture of $K$ Gaussians ($\mu_i$, $\sigma_i$, $\omega_i$) (Stauffer and Grimson, 2000)

• In this way, the model copes also with multimodal background distributions; however:
  – the number of modes is arbitrarily pre-defined (usually from 3 to 5)
  – how to initialize the Gaussians?
  – how to update them over time?
Gaussian Mixture Models

- Each pixel modeled with a mixture of Gaussians
- Flexible to handle variations in the background

\[
\hat{p}(\bar{x} | \mathcal{X}_T, BG+FG) = \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}(\bar{x}; \hat{\mu}_m, \hat{\sigma}_m^2 I)
\]
Mixture of Gaussians (MoG) (contd.)

• All weights $\omega_i$ are updated (updated and/or normalized) at every new frame.

• At every new frame, some of the Gaussians “match” the current value (those at a distance $< 2.5\, \sigma_i$): for them, $\mu_i,\, \sigma_i$ are updated by the running average.

• The mixture of Gaussians actually models both the foreground and the background: how to pick only the distributions modeling the background?:
  – all distributions are ranked according to their $\omega_i/\sigma_i$ and the first ones chosen as “background”.
GMM Background Subtraction

• Two tasks performed real-time
  – Learning the background model
  – Classifying pixels as background or foreground

• Learning the background model
  – The parameters of Gaussians
    • Mean
    • Variance and
    • Weight
  – Number of Gaussians per pixel

• Enhanced GMM is 20% faster than the original GMM*

* Improved Adaptive Gaussian Mixture Model for Background Subtraction, Zoran Zivkovic, ICPR 2004
Classifying Pixels

• $\bar{x}^{(t)} = \text{value of a pixel at time } t \text{ in RGB color space.}$

• Bayesian decision $R$ – if pixel is background (BG) or foreground (FG):

$$R = \frac{p(BG|\bar{x}^{(t)})}{p(FG|\bar{x}^{(t)})} = \frac{p(\bar{x}^{(t)}|BG)p(BG)}{p(\bar{x}^{(t)}|FG)p(FG)}$$

• Initially set $p(FG) = p(BG)$, therefore if $p(\bar{x}^{(t)}|BG) > c_{th.r}$ decide background

$$p(\bar{x}^{(t)}|BG) = \text{Background Model}$$

$$\hat{p}(\bar{x} | \mathcal{X}, BG) = \text{Estimated model, based on the training set } X$$
The GMM Model

• Choose a reasonable time period T and at time t we have
  \[ \mathcal{X}_T = \{ x^{(t)}, \ldots, x^{(t-T)} \} \]

• For each new sample update the training data set \( \mathcal{X}_T \)

• Re-estimate \( \hat{p}(x|\mathcal{X}_T, BG) \)

• Full scene model (BG + FG)

\[
\hat{p}(x|\mathcal{X}_T, BG+FG) = \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}(x; \hat{\mu}_m, \hat{\sigma}_m^2 I)
\]

GMM with M Gaussians where

• \( \hat{\mu}_1, \ldots, \hat{\mu}_M \) - estimates of the means

• \( \hat{\sigma}_1, \ldots, \hat{\sigma}_M \) - estimates of the variances

• \( \hat{\pi}_m \) - mixing weights non-negative and add up to one.
The Update Equations

• Given a new data sample \( \vec{x}^{(t)} \) update equations

\[
\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha (o_m^{(t)} - \hat{\pi}_m) \\
\hat{\mu}_m \leftarrow \hat{\mu}_m + o_m^{(t)} (\alpha / \hat{\pi}_m) \delta_m \\
\hat{\sigma}_m^2 \leftarrow \hat{\sigma}_m^2 + o_m^{(t)} (\alpha / \hat{\pi}_m) (\delta_m^T \delta_m - \hat{\sigma}_m^2)
\]

where, \( \delta_m = \vec{x}^{(t)} - \hat{\mu}_m \)

\( o_m^{(t)} \) is set to 1 for the ‘close’ Gaussian and 0 for others

and \( \alpha = 1/T \) is used to limit the influence of old data (learning rate).

• An on-line clustering algorithm.
• Discarding the Gaussians with small weights - approximate the background model:

\[
p(\vec{x}|\mathcal{X}_T, BG) \sim \sum_{m=1}^{B} \hat{\pi}_m N(\vec{x}; \hat{\mu}_m, \hat{\sigma}_m^2 I)
\]

• If the Gaussians are sorted to have descending weights \( \hat{\pi}_m \)

\[
B = \arg\min_b \left( \sum_{m=1}^{b} \hat{\pi}_m > (1 - c_f) \right)
\]

where \( c_f \) is a measure of the maximum portion of data that can belong to FG without influencing the BG model.
Mixture of Gaussians (MoG) (contd.)

Results - Mixture of Gaussians (MoG)

Background Model

Foreground Mask
But if camera moves?

Desired →
Conclusion

• Studied various motion detection and tracking algorithms

• Multiple BGS methods are needed for
  – Indoor (relative stable lighting)
  – Outdoor
    • Relative stable
    • High dynamic
  – Crowded environment
  – Camera Jitter/Shaking
Conclusion (contd.)

• Speed
  – Fast
    • Average, Median, Approximate median filtering
  – Intermediate
    • Eigenbackgrounds
  – Slow
    • MoG

• Memory requirements
  – High
    • Average, Median
  – Intermediate
    • MoG, Eigenbackgrounds
  – Low
    • Approximate median filtering