Computer Vision
Object and People Tracking

Background Subtraction

Prof. Didier Stricker
Stephan Krauß
Kaiserlautern University

http://ags.cs.uni-kl.de/

DFKI – Deutsches Forschungszentrum für Künstliche Intelligenz

http://av.dfki.de
Outline

- Introduction and Examples
- Background Subtraction Concept
  - Background Subtraction Principle
  - Model Initialization
  - Threshold Selection
  - Post-Processing
- Background Subtraction Methods
  - Frame Differencing
  - Single Gaussian Model
  - Gaussian Mixture Model
  - Color and Pixel Based Background Subtraction
  - Patch-Based Background Subtraction
Background Subtraction

- **Goal:** Separation of *Foreground* (object) and *Background* (everything else = noise)
- **Result (foreground mask) could be a**
  - Binary image, representing the foreground
  - Probability image, containing the likelihood of each pixel being foreground
- **Useful for**
  - Further processing, such as using silhouettes, etc.
  - Less processing, less room for errors
Background Subtraction

- Simple techniques can do ok with static cameras
- …but hard to do perfectly

- Widely used:
  - Traffic monitoring (counting vehicles, detecting and tracking vehicles, pedestrians)
  - Human action recognition (running, walking, jumping, squatting)
  - Human-computer interaction
  - Object segmentation
  - Object detection and tracking
Background Subtraction Example
Why is it Hard?

- **Illumination changes**
  - Gradual (evening to night)
  - Sudden (overhead clouds)

- **Changes in the background geometry**
  - Parked cars (should become part of the background)

- **Camera related issues**
  - Camera oscillations (shaking)
  - Grainy noise
  - Motion blur

- **Changes in background objects**
  - Tree branches
  - Sea waves

- **Cast lights and shadows**
  - Lights and shadows cast by foreground objects
Background Subtraction

- A largely unsolved problem…

One video frame  Estimated background  Difference Image  Thresholded foreground on blue
Blue/Green Screen
Blue/Green Screen

- Background is modeled as a range of blue/green colors in an appropriate color space, e.g. HSL

Advantages
- Works very accurately
- Camera may move

Disadvantages
- Restricted applicability (controlled environment required)
- Foreground objects may not contain the background color
An example: The Ultimatte System
The Ultimatte System

- Underlying method: color differencing
- Difference of blue channel to red or green channel: $\alpha = B - \max(G, R)$

- The system additionally uses various patented improvements to this basic approach
Background Subtraction

- **Typical assumptions**
  - Static background
  - Foreground objects move
  - Static cameras

- **Standard approach**
  - Build a model of the background
  - Create a foreground mask by deciding for each pixel for foreground or background based on the distance between the current pixel value in the frame and background model
  - Update the background model

- **Main components of BS algorithms**
  - Background model, distance metric, mask generation method and model update procedure
Background Subtraction Base Algorithm

1. Initialize Model
2. Repeat for each frame:
   1. Compute distance between background model and current frame
   2. Compute foreground mask based on distance metric
   3. Update the model using the current frame
Example: Frame Differencing

- Naive approach for static backgrounds

Main components
- **Model**: RGB values of each pixel
- **Distance metric**: sum of absolute differences of RGB values per pixel
- **Mask generation**: binary mask, where a pixel is foreground iff absolute difference is above a threshold
- **Model update**: replace model with current frame

Disadvantages
- Pixel noise
- Only object outlines may appear in the mask (depending on frame-rate and object self-similarities)
- Manual threshold selection
Frame Differencing
Overview of Different BS Methods

Background subtraction methods vary in …

▪ … what is modeled:
  ▪ Per-pixel: color or color and gradient
  ▪ Per-pixel patch: texture

▪ … how it is modeled:
  ▪ Parametric models: e.g. Gaussian models
  ▪ Non-parametric models: e.g. Kernel density estimation

▪ … how the mask is generated:
  ▪ Binary masks can be obtained by thresholding the distance
  ▪ Probability masks can be obtained either
    ▪ from the distance function directly if it yields probabilities or
    ▪ by mapping the value range of the distance function into a probability range with an appropriate mapping function
**Single Gaussian Model**

- **Main components**
  - Model: mean color value (e.g. RGB) $\mu$ and variance $\sigma^2$
  - Distance metric: probability of pixel’s RGB values $x$ to belong to the model:
    \[
    p(x) = \frac{1}{(2\pi)^{3/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right\}, \text{ where } \Sigma = \sigma^2 I
    \]
  - Model update
    - Mean: $\mu_{t+1} = \alpha \cdot x + (1 - \alpha) \mu_t$
    - Variance: $\sigma^2_{t+1} = \alpha \cdot (x - \mu_{t+1})^2 + (1 - \alpha) \cdot \sigma_t^2$ (per channel)
    - With suitable learning rate $\alpha$

---

**Exponential decay**

\[1 - \alpha = 0.9\]
Single Gaussian Model: Pros and Cons

- **Advantages**
  - Easy to implement and use
  - Very fast method
  - Background model can change over time, e.g., due to gradual illumination changes

- **Disadvantages**
  - Cannot handle non-static (= dynamic) backgrounds
  - Pixel noise
  - Manual, global threshold for mask generation is not optimal
Single Gaussian Model: Threshold

- Threshold should be different for each pixel
Local Adaptive Thresholds

- Threshold $\theta$ can be chosen based on the standard deviation $\sigma$ of a pixel’s value over time → a local threshold

- Sample mean $\mu$
  \[
  \mu = \frac{1}{N} \sum_{i=1}^{N} x_i
  \]

- Sample variance $\sigma^2$
  \[
  \sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2
  \]

- For example:
  - If $\theta_{\text{min}} < d < \theta_{\text{max}}$ → BG pixel
  - $\theta_{\text{min}} = \mu - K \cdot \sigma$
  - $\theta_{\text{max}} = \mu + K \cdot \sigma$
  - $K$ is set by you! (usually $K = 2$)

### Example Table

<table>
<thead>
<tr>
<th>Range</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma$</td>
<td>68.26 %</td>
</tr>
<tr>
<td>$2\sigma$</td>
<td>95.44 %</td>
</tr>
<tr>
<td>$3\sigma$</td>
<td>99.73 %</td>
</tr>
<tr>
<td>$4\sigma$</td>
<td>99.99 %</td>
</tr>
</tbody>
</table>
Model Initialization

- Often foreground objects are present continuously
- But the background model needs to be initialized
- → Accumulate background data over multiple frames, e.g., through
  - Mean filtering
  - Median filtering
Mean Filtering

- Each background pixel is the mean of N frames

Mean of 15 frames
Median Filtering

- Each background pixel is the median of N frames
Recall: Median Filtering

- A median filter operates over a window by selecting the median intensity in the window.

1D median filtering
- **input**
  
  | 10 | 52 | 20 | 23 | 90 | 27 | 33 | 31 | 30 |

- **sort**
  
  | 10 | 20 | 23 | 27 | 30 | 31 | 33 | 52 | 90 |

- **output**

2D median filtering
- serialize input and apply 1D median filter, e.g. 3x3 filter:

  | 10 | 52 | 20 |
  | 23 | 90 | 27 |
  | 33 | 31 | 30 |

  | 10 | 52 | 20 | 23 | 90 | 27 | 33 | 31 | 30 |
Median Image

1D median filtering along the temporal axis
Post Processing

- Pixel noise can be reduced by post processing
  - 2D median filtering
  - Morphological filtering
  - Depending on filter size, bigger spots can be erased

Median filtering used to remove noise
Morphological Filtering

- Basic operations: erosion and dilation
  - Based on structuring element (SE)
  - Structuring element describes neighborhood relation for a pixel (shown in red)
  - The value of the filtered pixel is defined as the minimum (erosion) or maximum (dilation) value of the all the pixels in the neighborhood

- Dilation example:

original image: [image]

dilated image: [image]
Morphological Filtering

- **Opening:** *erosion* followed by *dilation*
  - Foreground structures smaller than the structuring element will be removed
Morphological Filtering

- **Closing**: *dilation* followed by *erosion*
  - Holes in the foreground smaller than the structuring element will be filled

![Morphological Closing Diagram]
Gaussian Mixture Model (GMM)

- Extension of the single Gaussian model
- Goal: handling of non-static backgrounds

Model: K clusters (j=1..K) per pixel with
  - Mean color value (e.g. RGB) $\mu_j$
  - Per color-channel variance $\sigma^2_j = (\sigma^2_R, \sigma^2_G, \sigma^2_B)_j^T$
  - Mixture weight $w_j$

Distance metric
  - Probability of pixel’s RGB values $x = (R, G, B)^T$ to belong to the background model:
    $$p(x) = \sum_{j=1}^K w_j \frac{1}{(2\pi)^{3/2} |\Sigma_j|^{1/2}} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma_j^{-1} (x - \mu) \right)$$
    where
    $$\Sigma_j = \sigma^2_j I$$
    $$p(x) = \sum_{j=1}^K w_j \cdot g_j(x; \mu_j, \sigma^2_j)$$
Gaussian Mixture Model (GMM)

Model update

- Compute matching quality $p$ for each cluster $j$:
  \[ p_j = w_j \cdot g_j(x; \mu_j, \sigma_j) \] where $g$ is the Gaussian function

- Find for each pixel $x$ the best matching cluster $j$ and update:
  - **Mean:** $\mu_{j,t+1} = \alpha \cdot x + (1 - \alpha) \mu_{j,t}$
  - **Covariance:** $\Sigma_{j,t+1} = \alpha \cdot ((x - \mu_{j,t+1})(x - \mu_{j,t+1})^T) + (1 - \alpha)\Sigma_{j,t}$
  - **Weight:** $w_{j,t+1} = (1 - \alpha)w_{j,t} + \alpha$

- For all other clusters of the pixel reduce their weight:
  - $w_{j,t+1} = (1 - \alpha)w_{j,t}$

- **Learning rate $\alpha$:** $0 \leq \alpha \leq 1$
  - Has to be tuned to match the speed of gradual changes in the background, e.g. due to lighting, in your application
Gaussian Mixture Model (GMM)

- **Advantages**
  - Can handle dynamic background

- **Disadvantages**
  - Pixel noise → apply morphological filtering
  - Pixels are considered independently (the context is ignored)
  - Spotlights and shadows cast by foreground objects will appear as foreground

- **Reading**
Pixel and Gradient Based Approach


Model: K clusters per pixel \((i, j)\) with
- Mean color value, variance and mixture weight as in the standard GMM
- Additionally considers the gradient \(\Delta\) in terms of magnitude \(\Delta_m\) and direction \(\Delta_d\) computed on the grayscale values of neighboring pixels in x and y direction
- Gradient is not stored explicitly but can be derived from the probability distribution of the color values

Model update
- Probability distribution of the color values are updated as in the GMM
- Gradient distribution does not need to be updated explicitly
Pixel and Gradient Based Approach

Distance metric
- Based on two probability distributions

- Probability $p_c$ of pixel’s RGB values $x = (R, G, B)^T$ to belong to the background model (same as in GMM)
- Probability $p_g$ of pixel’s gradient (M,D) to belong to the background model
  - Complex function
  - But all parameters can be derived from $p_c$ of involved pixels
- Masks are generated by thresholding on the probability values $p_c$ and $p_g$
Pixel and Gradient Based Approach

- **Fusion of masks on region level**
  - Border pixels of a foreground blob in the pixel mask are verified with gradient mask
  - Blob is then either confirmed as foreground or dismissed as ghost, e.g. caused by a light spot or a moved background object

- **Advantages**
  - Can handle gradual and sudden (local) illumination changes
  - Can detect ghost, e.g. if a background object is moved (the objects original location will not be detected as foreground)

- **Disadvantages**
  - Appearance of background defined on pixel level → no texture information considered
Patch-Based Approach

- **Model**
  - Overlapping image patches of fixed size (8x8)
  - Patch descriptor vector $d$: 4 lowest frequencies of a 2D DCT of each color channel (RGB)

- **Distance Metrics**
  - Composed of 3 individual metrics:
    - Gaussian model of patch descriptor distribution $(\mu, \sigma^2)$
    - Illumination robust metric: cosine distance metric
      
      $$
      cosdist(d, \mu) = 1 - \frac{d \cdot \mu}{\|d\| \|\mu\|}
      $$
      
    - Temporal correlation check: $cosdist(d^t, d^{t-1})$ if patch was classified as background in previous frame
  
  - **Final metric: multi-stage binary classification**
    - Check of thresholds on those 3 metrics:
      - If one votes for background, the patch is considered background
      - If all vote for foreground, the patch is considered foreground
Patch-Based Approach

Probabilistic Mask Generation for each pixel x:
- Foreground probability $p$: number of foreground patches containing pixel x divided by no. of all patches containing x

Model Update
- Standard update with learning rate $\alpha$
- Performed only if Gaussian model votes for background
- Slow motion and gradual illumination changes are learned into the background model

Advantages
- Considers pixel context, robust to dynamic background, gradual and sudden illumination changes, no pixel noise

Disadvantages
- Prone to over-smoothing of blob contours
- Cannot distinguish gray values due to use of illumination robust metric
Patch-Based Approach

Reading

- Vikkas Reddy, Conrad Sanderson, Andreas Sanin, Brian C. Lovell. „Adaptive Patch-Based Background Modelling for Improved Foreground Object Segmentation and Tracking“
Applications: Blob Extraction and Tracking
Blob Extraction

Blob = connected component
- Maximal subset of pixels which are all connected to each other
- Neighborhood relation, e.g. 4-connectivity or 8-connectivity
- Two pixels x, y are connected if
  - x and y are neighbors and have the same value or
  - if there is a pixel z, where x and z are neighbors and have the same value and z and y are connected

Connected component labeling
- Task: label each component, i.e. all connected pixels, with the same label

Challenge
- Efficient implementation
- How would you implement it?

4-connectivity

8-connectivity
An Efficient CC Labeling Algorithm

Reading

Key idea
- assign provisional labels in a first pass over the image and final labels in the second

Definitions
- **run**
  - the maximal sequence of consecutive pixels in a row with the same pixel value
- **representative label**
  - the label (from a label set) with the smallest value
## CC Labeling Example

### Provisional labels:

<table>
<thead>
<tr>
<th>Row 1</th>
<th>Row 2</th>
<th>Row 3</th>
<th>Row 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

### Sets of equivalent labels:
- \(S(1) = \{1, 2, 3, 4, 5\}\)
- \(S(2) = \{1\}\)
- \(S(3) = \{2, 3\}\)
- \(S(4) = \{4\}\)
- \(S(5) = \{5\}\)

Each label is in exactly one set.

### Representative label for each label set:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
The Two Pass Algorithm by He et al

First pass
1. **Iterate through the image rows**
2. **Iterate through the runs in the row**
   1. Get the runs in the previous row which are 8-connected to the current run
   2. If there are no neighbors, uniquely label the current run with a new provisional label and continue
   3. Otherwise, find the leftmost neighbor and assign its label to the current run
   4. Merge the provisional label sets of all runs in the previous row, which are 8-connected with the current run

Second pass
1. **Iterate through each pixel of the image**
   1. If it is not a background pixel then relabel it with the label set’s representative label
Efficient Implementation

Creation of a new label set $S(m) = \{m\}$:

- `rtable[m] = m` // representative of set $m$ is $m$
- `next[m] = -1` // $m$ has no next label in the set
- `tail[m] = m` // the last label in the set is $m$

```
S(1) = \{1, 2\}, S(2) = \{2\}
S(1) = \{1\}, S(2) = \{2\}
```

**merge**($u,v$): // merge two sets $u$, $v$

```
for (i = v; i != -1; i = next[i]) // for all labels in set $v$
    rtable[i] = u; // update representative to $u$
end
next[tail[u]] = v; // connect tail of $u$ with head of $v$
tail[u] = tail[v]; // set tail of set $u$ to tail of set $v$
```
Efficient Implementation

```plaintext
resolve(x, y): // resolve equivalence of two label sets
u = rtable[x]; // get representative label for label x
v = rtable[y]; // get representative label for label y
if (u < v)
    merge(u, v); // u will be the new representative
else if (u > v)
    merge(v, u); // v will be the new representative
end
```

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

S(1) = {1}  S(2) = {2}
S(3) = {3}  S(1) = {1,2}
S(1) = {1, 2, 3}

S(1) = {1,2}  S(3) = {3}  S(1) = {1}
Multi-Target Blob Tracking

Task
- Identity assignment of blobs from different frames containing the same object

Solution
- Assign identity of blob from previous frame which is most similar to the current blob and is above a minimal threshold

How to describe the similarity of blobs?
- Blob location (center)
- Blob size (area)
- Blob shape (e.g. convexity)
- Similarity of the contained object, e.g. color histograms
Color Histogram

**Naive approach**
- Binning per color channel into equidistant bins of width $w$
- Concatenation of color channel bins
- Normalization of the combined histogram
- Problems:
  - Color noise in dark areas
  - Limited descriptiveness
  (color channels are considered independently)

$$N = 3 \cdot \left( \frac{256}{w} \right)$$
Color Histogram

Clever approach
- Binning directly in the 3D color space $\rightarrow$ 3D bins
- Avoids problems due to color noise in dark areas
- Preserves interdependence of color channel values
- Disadvantage: more bins for same binning width $w$

Distance metric
- Use e.g. histogram correlation or histogram intersection

Reading
- M. Swain, D. Ballard.
  “Color Indexing”
Outlook and Further Reading

- Non-standard BS approaches
  - Fatih Porikli. “Multiplicative background-foreground estimation under uncontrolled illumination using intrinsic images.”

- BS approaches with relaxed requirements

- On-the-fly online background model learning
  - Dar-Shyang Lee. “Effective Gaussian Mixture Learning for Video Background Subtraction”

- Non-parametric models
Thank you!