Adaptive Foreground Detection in Video

Agenda

- Motivation and Examples
- Offline Foreground Detection
- Adaptive Foreground Detection
  - Introduction to adaptive foreground detection
  - Simple techniques
  - State-of-the-art algorithms
- Demo
- Conclusion

Agenda

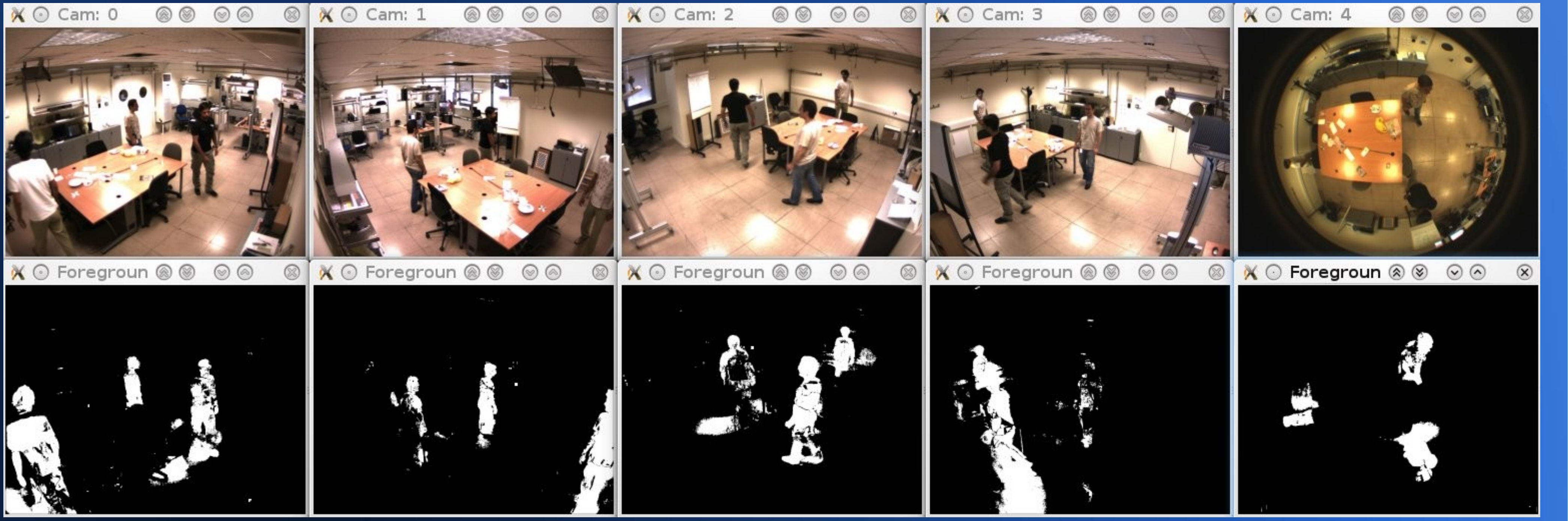
- **Motivation and Examples**
- Offline Foreground Detection
- Adaptive Foreground Detection
    - Introduction to adaptive foreground detection
    - Simple techniques
    - State-of-the-art algorithms
- Demo
- Conclusion

Motivation and Applications for Foreground Detection

- Detect people in autonomous surveillance systems
  - Activate alarms when suspicious activity is detected
- Detect people in human-computer interaction systems, such as games
- Automatic traffic control systems
- Assist in production processes

Motivation and Applications for Foreground Detection

- Video surveillance example:
  - (from current project)



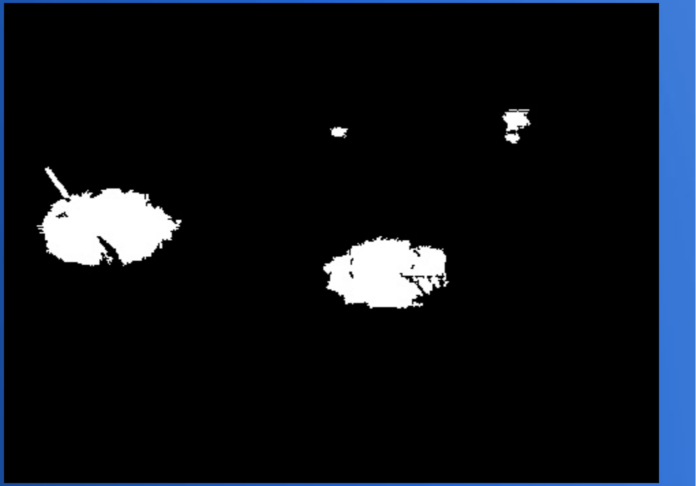
Motivation and Applications for Foreground Detection

- Traffic surveillance example:

Background

Image

Detected Foreground



Motivation and Applications for Foreground Detection

- Alarm systems:

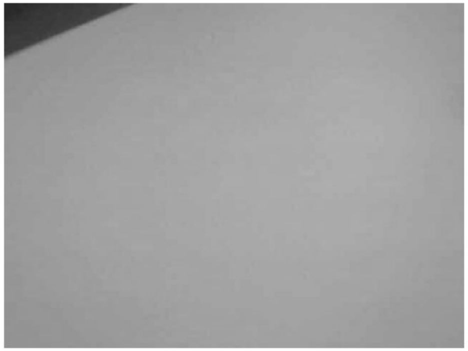
Background

Image

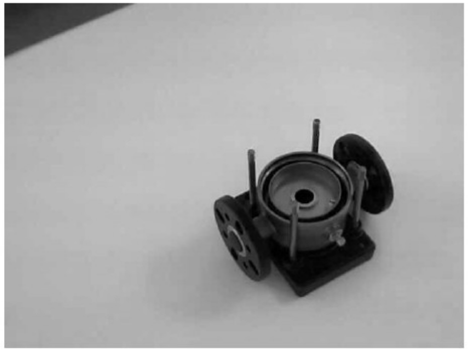
Detected Foreground

Motivation and Applications for Foreground Detection

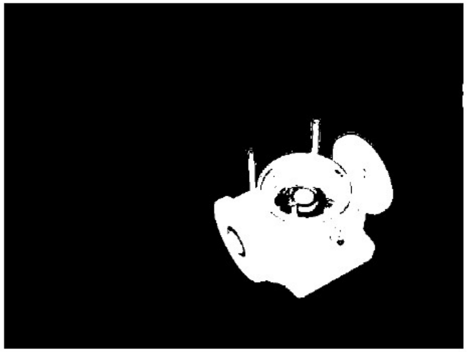
- Assist in production processes



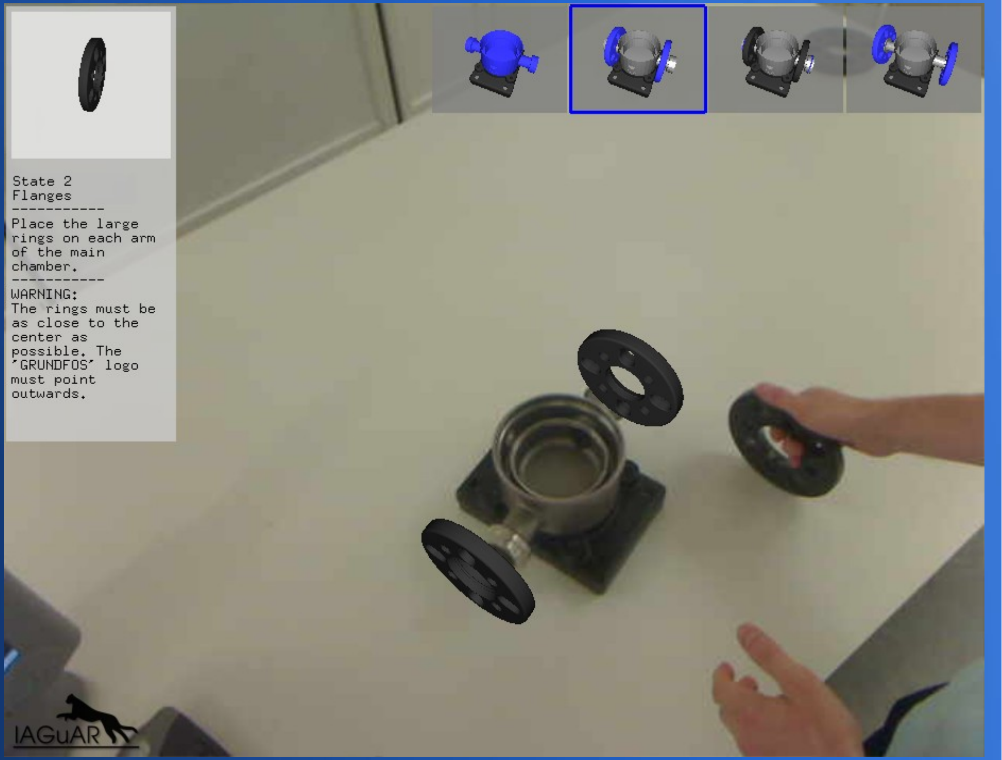
Static background



Current frame



Thresholded Mask



State 2
Flanges

Place the large rings on each arm of the main chamber.

WARNING:
The rings must be as close to the center as possible. The 'GRUNDFOS' logo must point outwards.

Agenda

- Motivation and Examples
- **Offline Foreground Detection**
- Adaptive Foreground Detection
    - Introduction to adaptive foreground detection
    - Simple techniques
    - State-of-the-art algorithms
- Demo
- Conclusion

Offline Foreground Detection

- A background is determined first
- A new image with possible foreground objects is recorded
- Foreground can then be detected by comparison

Offline Foreground Detection using One Background Image

- Simple approach to foreground detection by using one background image

Offline Foreground Detection using One Background Image

- Simple approach to foreground detection by using one background image

1) Take one image of the background

Offline Foreground Detection using One Background Image

- Simple approach to foreground detection by using one background image

  1) Take one image of the background

  2) Take image with possible foreground

Offline Foreground Detection using One Background Image

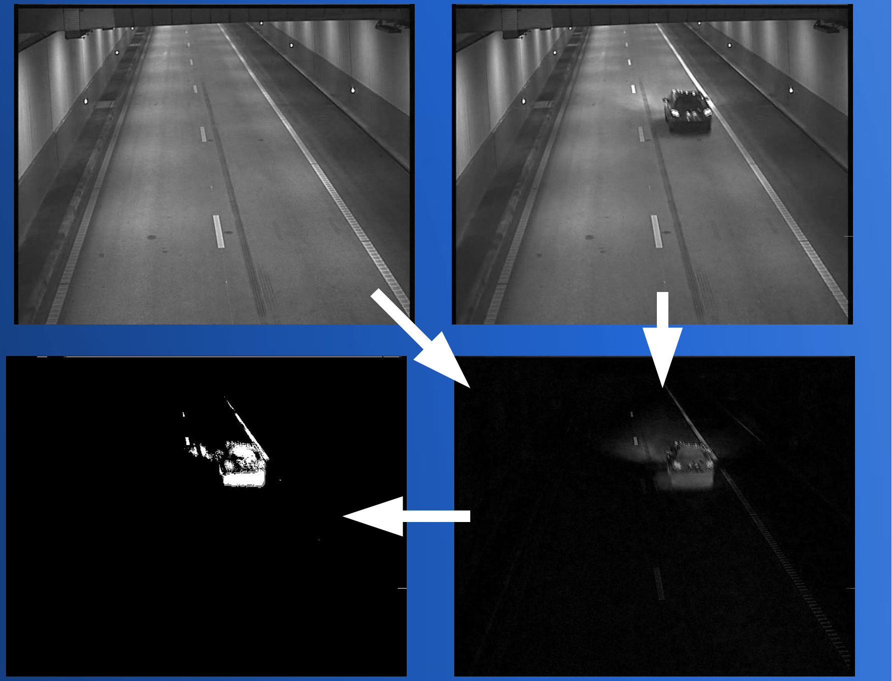
- Simple approach to foreground detection by using one background image

1) Take one image of the background
2) Take image with possible foreground
3) Convert to grayscale and subtract images

Offline Foreground Detection using One Background Image

- Simple approach to foreground detection by using one background image

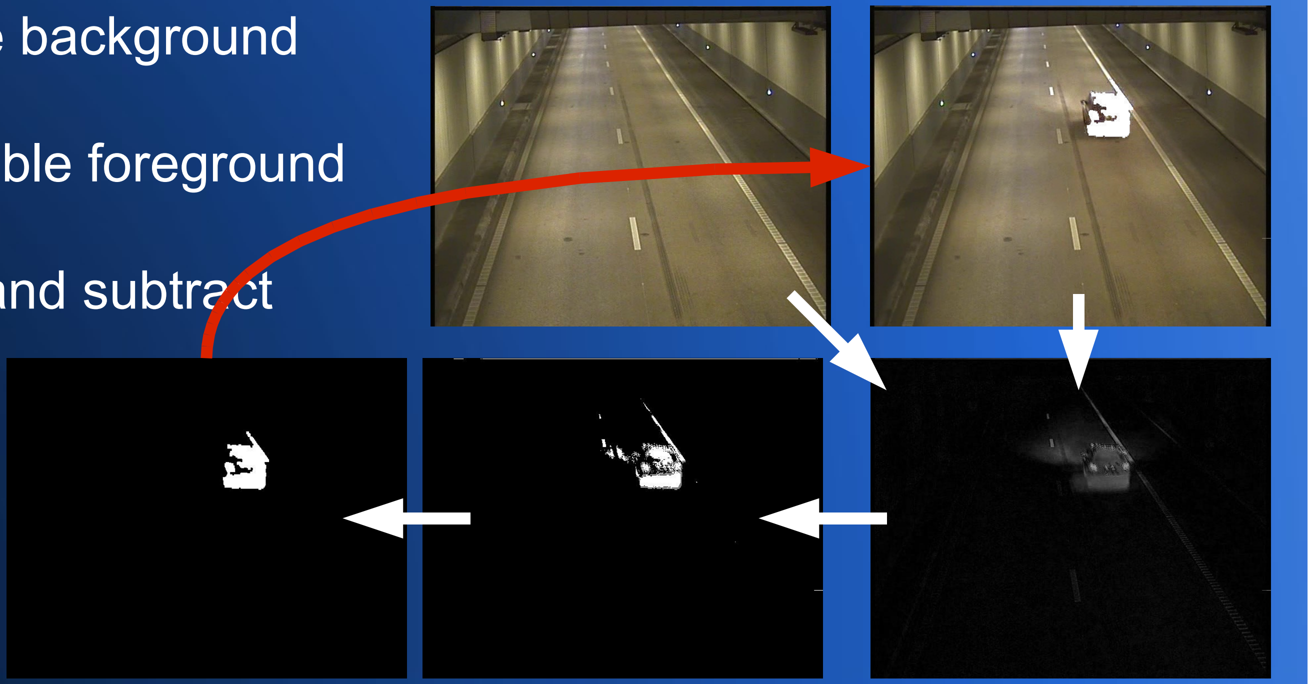
1) Take one image of the background
2) Take image with possible foreground
3) Convert to grayscale and subtract images
4) Threshold image



Offline Foreground Detection using One Background Image

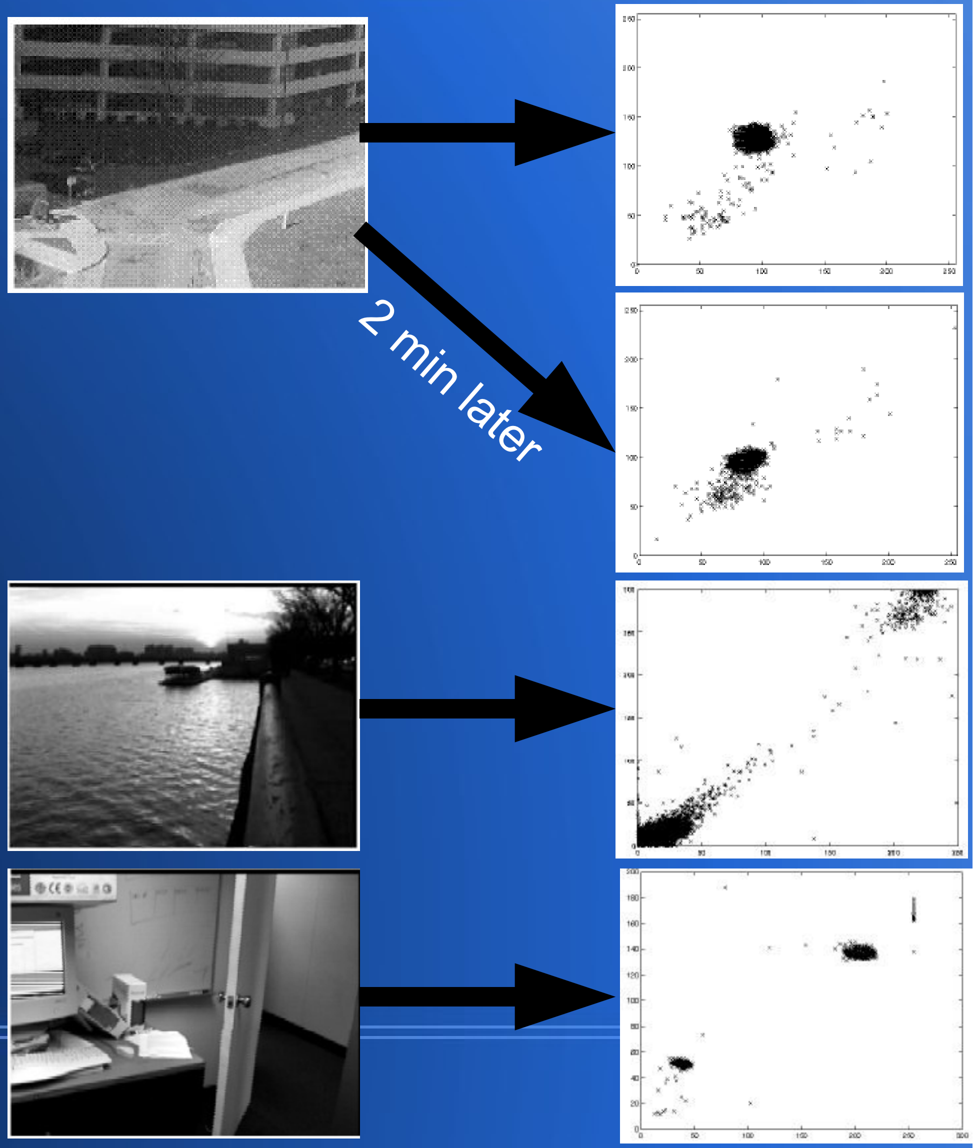
- Simple approach to foreground detection by using one background image

1) Take one image of the background
2) Take image with possible foreground
3) Convert to grayscale and subtract images
4) Threshold image
5) Remove noise (if necessary)



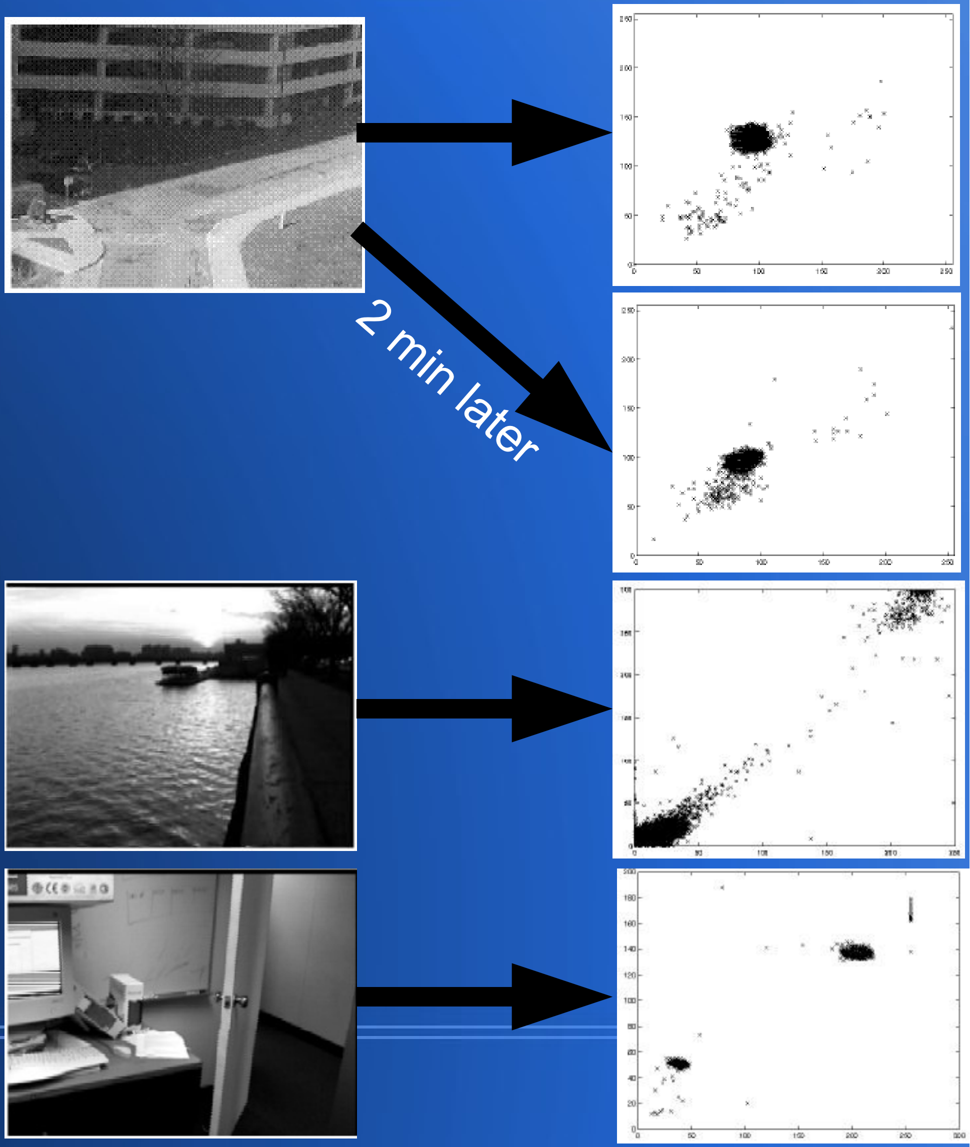
Problems in Offline Foreground Detection

- Random variations
- Light intensity changes
  - Day/twilight
  - Shadows
- Changing background
  - Swaying tree
  - CRT screen



Problems in Offline Foreground Detection

- **Random variations**
- Light intensity changes
  - Day/twilight
  - Shadows
- Changing background
  - Swaying tree
  - CRT screen

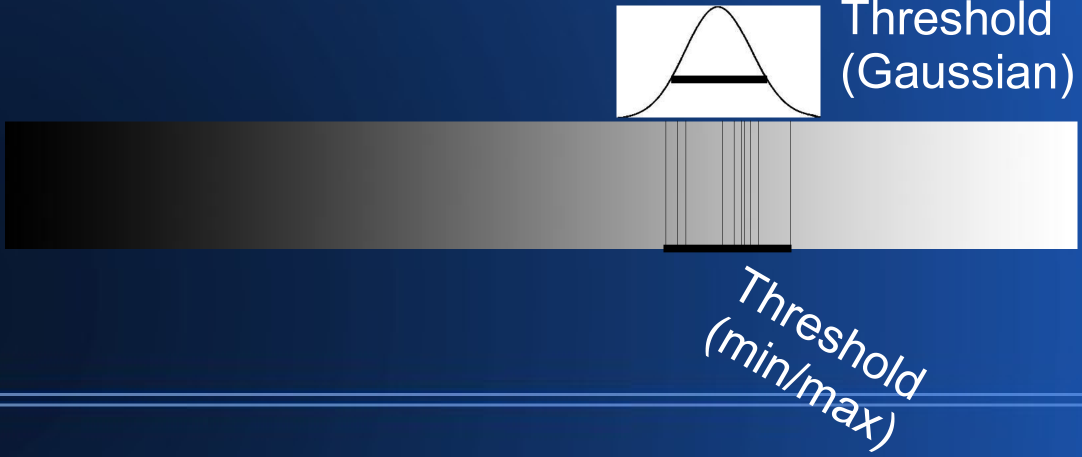


Compensation for Random Variations

- Standard thresholding:

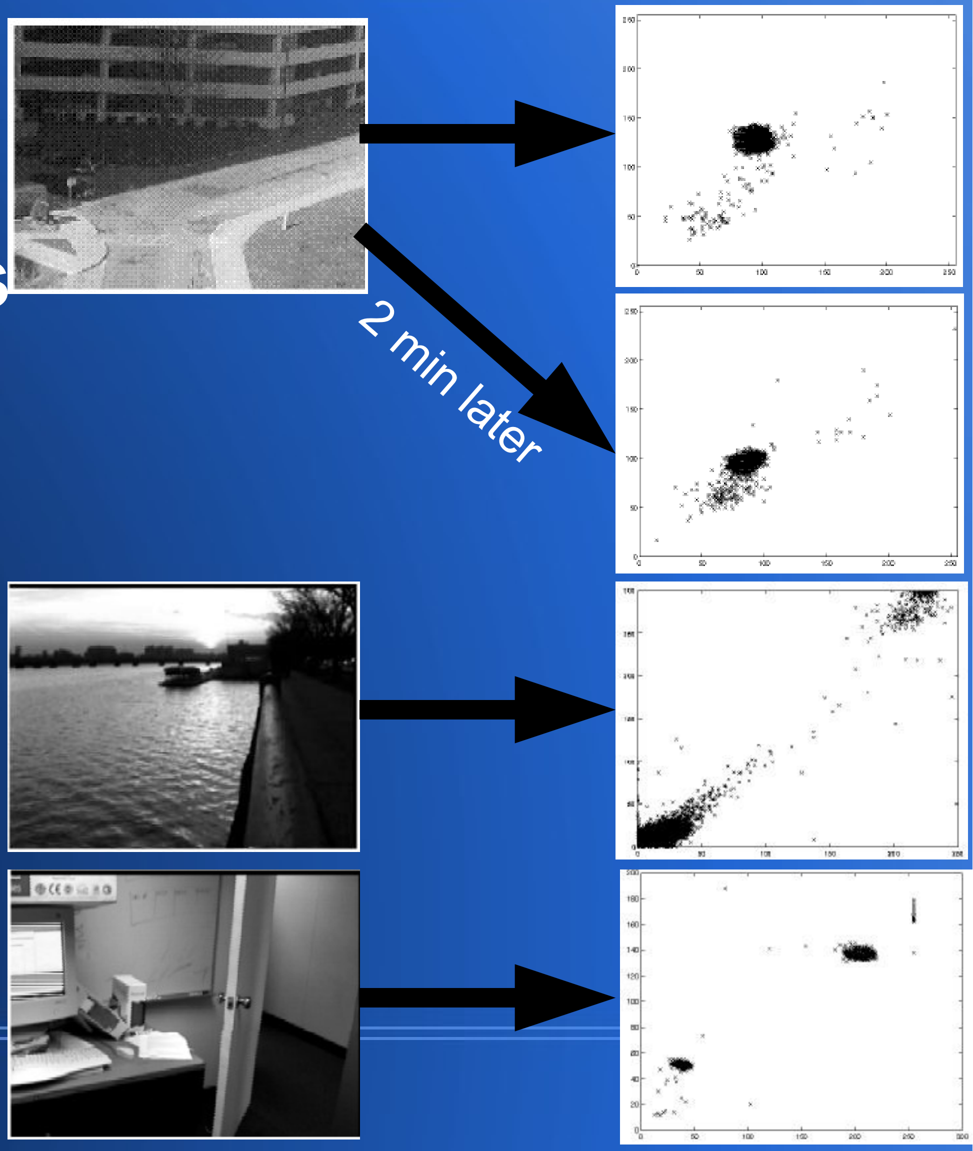
Compensation for Random Variations

- Possible solution:
  - Use information from multiple background images
    - Set intensity threshold to min/max, or
    - Use a Gaussian distribution



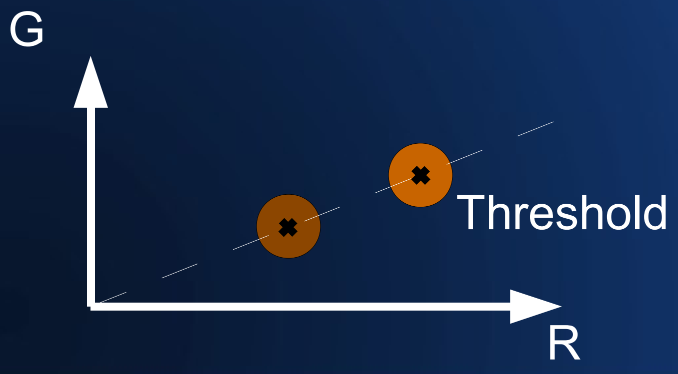
Problems in Offline Foreground Detection

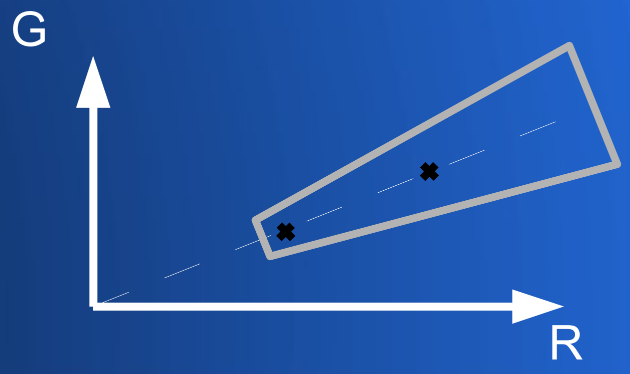
- Random variations
- **Light intensity changes**
  - **Day/twilight**
  - **Shadows**
- Changing background
  - Swaying tree
  - CRT screen



Compensation for Light Intensity Changes

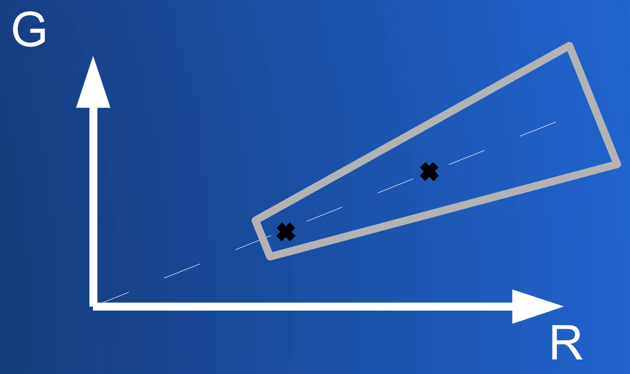
- Potential causes:
  - Changing intensity of the sun light
    - As the sun moves
    - Clouds covering the sun
  - Shadows





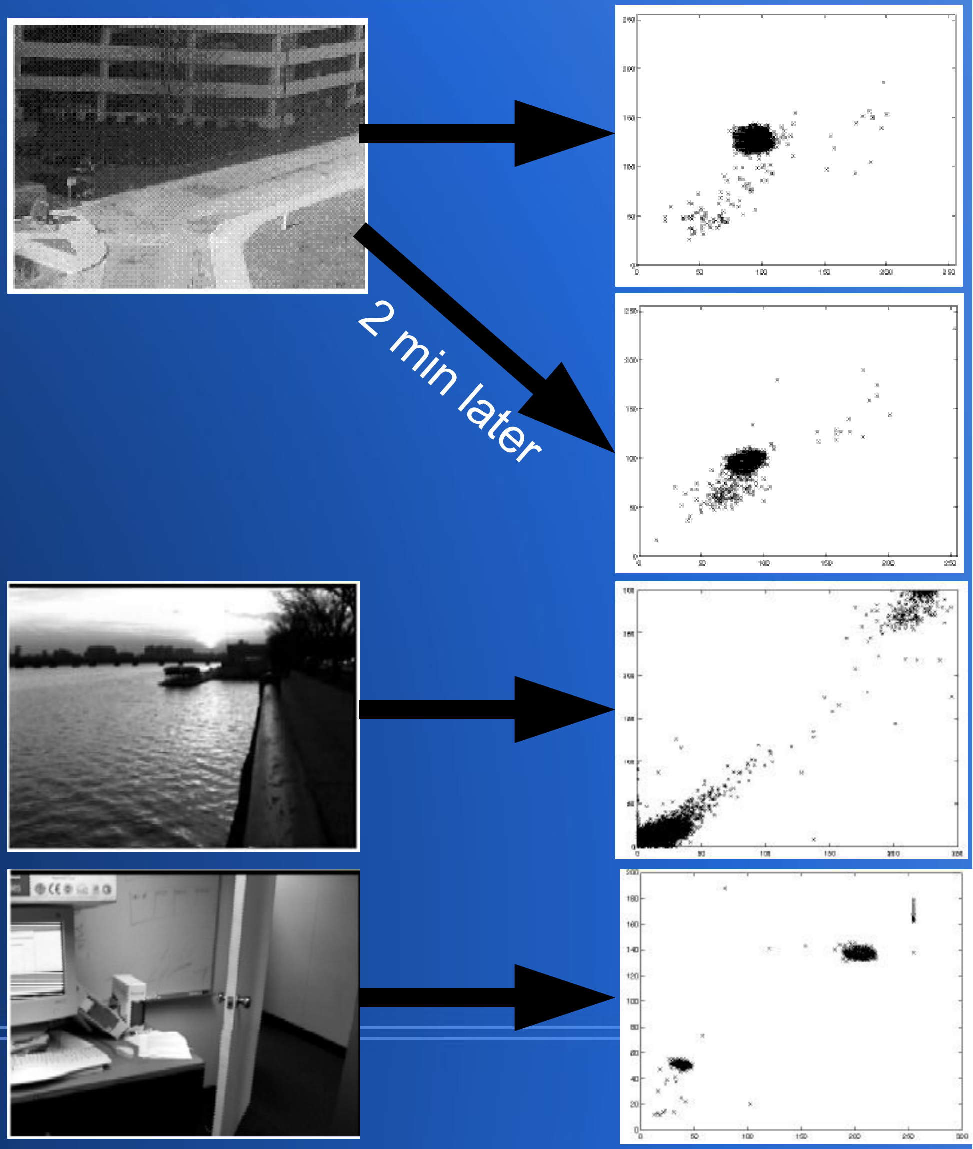
Compensation for Light Intensity Changes

- Solution:
  - Convert from RGB-space to a color space with decoupled luminance, eg. YCbCr
  - Set a wide threshold of the Y-component in the YCbCr color space



Problems in Offline Foreground Detection

- Random variations
- Light intensity changes
  - Day/twilight
  - Shadows
- **Changing background**
  - **Swaying tree**
  - **CRT screen**

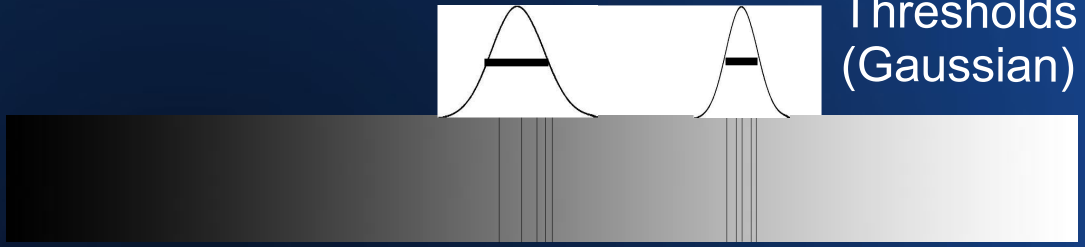


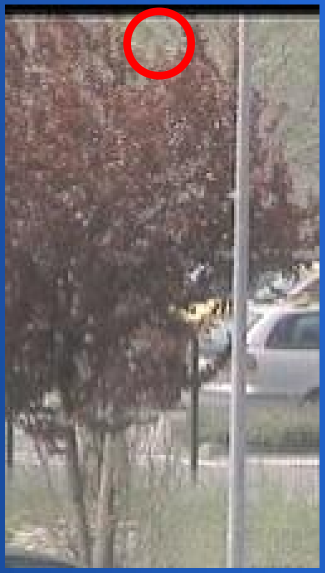
Compensation for Changing Background

Compensation for Changing Background

Compensation for Changing Background

- Derive multiple hypotheses from multiple background images
  - Set multiple thresholds using multiple Gaussian distributions





Compensation for Changing Background

- Derive multiple hypotheses from multiple background images
    - Set multiple thresholds using multiple Gaussian distributions
- It is only possible to compensate for changes that can be predicted, e.g. repetitive changes
- A better approach is to use an adaptive technique, that is able to update the background image on-the-fly

Agenda

- Motivation and Examples
- Offline Foreground Detection
- **Adaptive Foreground Detection**
  - **Introduction to adaptive foreground detection**
  - Simple techniques
  - State-of-the-art algorithms
- Demo
- Conclusion

Motivation for Adaptive Foreground Detection

- No background image may be available
- Tighter thresholds possible with light intensity changes
- An adaptive technique can provide:
  - Adaptation to scene changes
  - Learning and updating of the background continuously

Agenda

- Motivation and Examples
- Offline Foreground Detection
- **Adaptive Foreground Detection**
  - Introduction to adaptive foreground detection
  - **Simple techniques**
  - State-of-the-art algorithms
- Demo
- Conclusion

Adaptive Foreground Detection: Simplest Possible Technique?

- Pixel based IIR-filter with thresholds
- For each pixel:
  - For each frame:
    - Update the background: $\mu_{new} = \alpha \cdot v + (1 - \alpha) \cdot \mu_{old}$

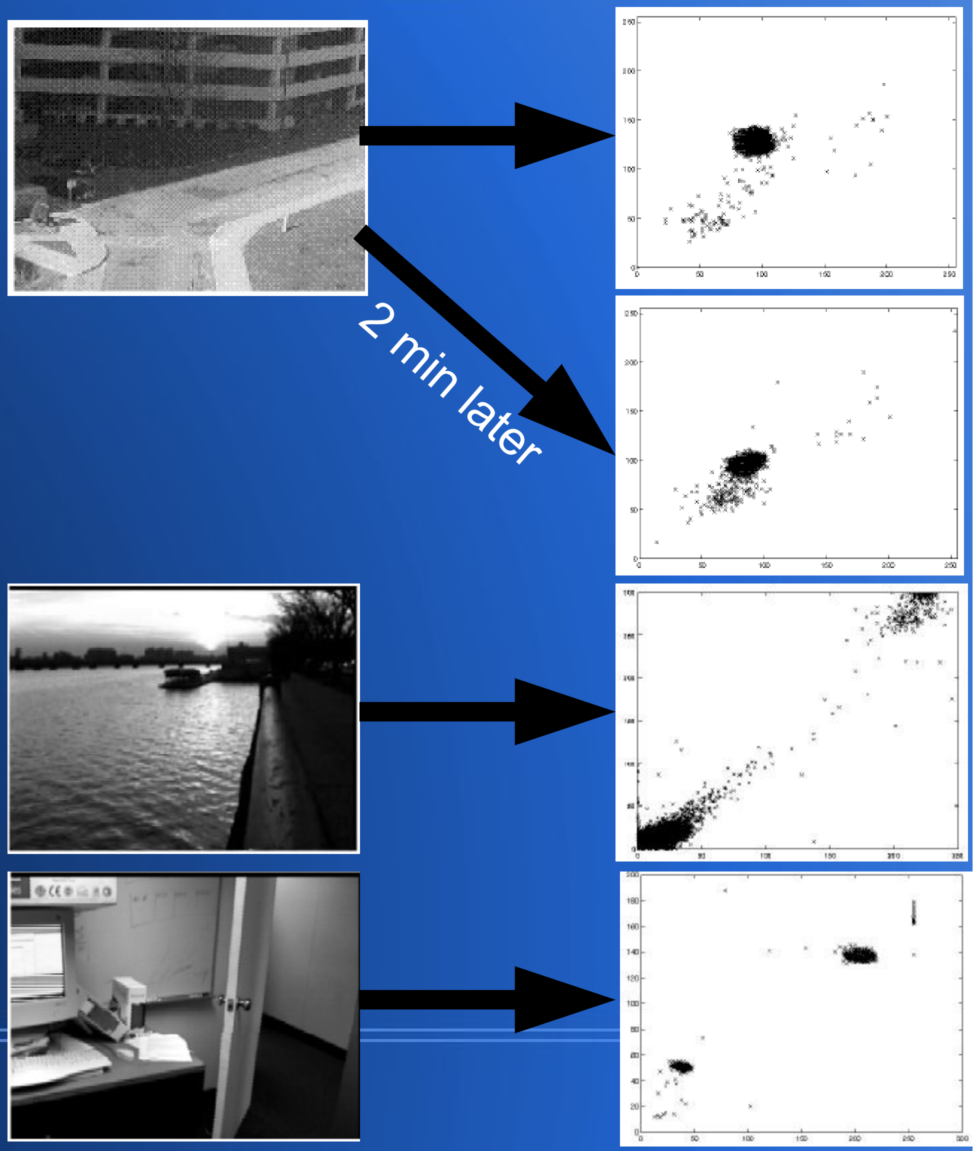
      where ***v*** is the color in the new frame
  - If ***v*** is within the threshold, it is considered background
- Large α gives fast adaptation, but also incorporates new items into the background quickly

Agenda

- Motivation and Examples
- Offline Foreground Detection
- **Adaptive Foreground Detection**
    - Introduction to adaptive foreground detection
    - Simple techniques
    - **State-of-the-art algorithms**
- Demo
- Conclusion

State-of-the-Art Algorithms

- Random variations
- Light intensity changes
  - Day/twilight
  - Shadows
- Changing background
  - Swaying tree
  - CRT screen

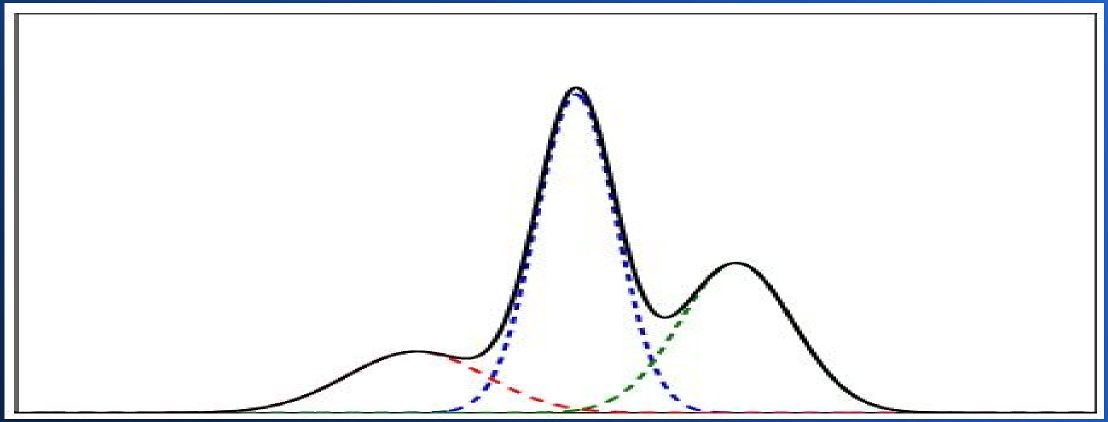


State-of-the-Art Algorithms

- Mixture of Gaussians
  - General algorithm by Stauffer et. al., [2000]
  - Improvements used in OpenCV by KaewTraKulPong et. al., [2001]
- Other Approaches
  - Histogram based techniques by [Liyuan Li et. al. 2004]

Adaptive Foreground Detection: Mixture of Gaussians

- Developed within the last two decades

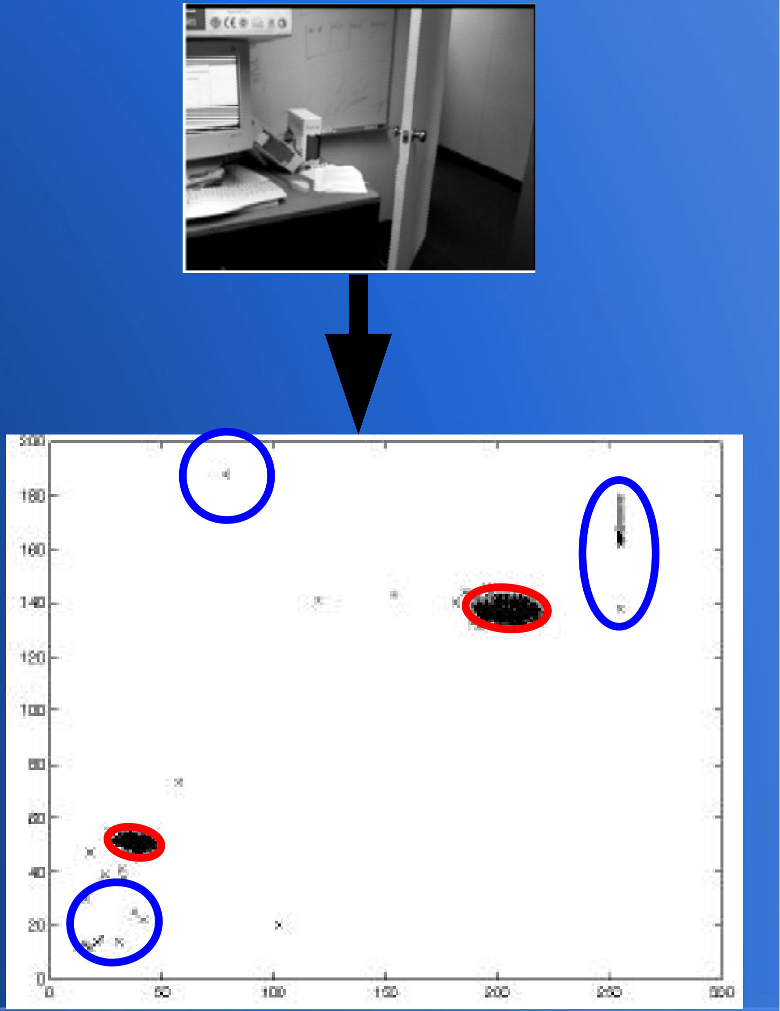


  - Over 150 different papers investigating different kinds of this method [Bouwmans et. al. 2008]
  - Walkthrough an algorithm by Stauffer et. al. [2000] (the authors of the original MoG)

Mixture of Gaussians

- Model each pixel as a mixture of K Gaussians (K = 3 to 5)
- A Gaussian model for a RGB pixel (12 parameters)
  - Means μ_R, μ_G, μ_B
  - co-variance matrix

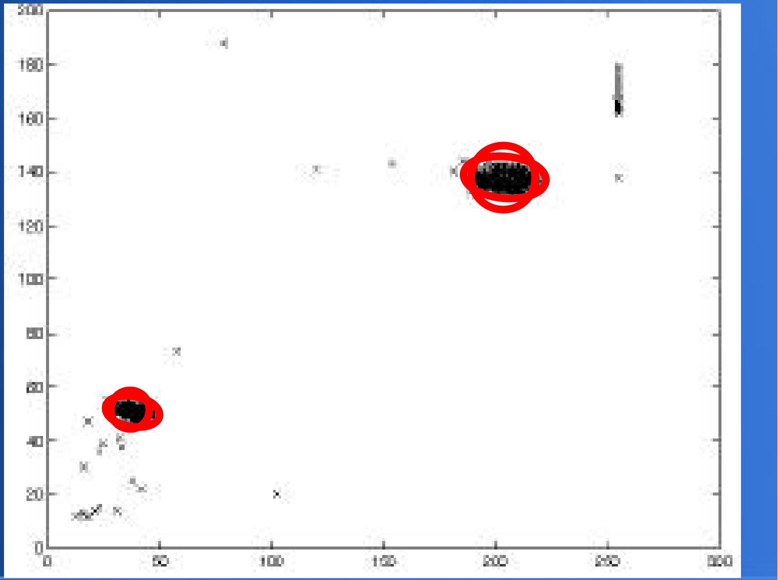
$$\Sigma = \begin{bmatrix} \sigma^2_{R,R} & \sigma^2_{R,G} & \sigma^2_{R,B} \\ \sigma^2_{G,R} & \sigma^2_{G,G} & \sigma^2_{G,B} \\ \sigma^2_{B,R} & \sigma^2_{B,G} & \sigma^2_{B,B} \end{bmatrix}$$



Mixture of Gaussians

- Co-variance matrix to 1 parameter:
- A total of 5*K parameters per pixel (when using K Gaussians)
  - 3 means
  - 1 variance
  - 1 weight

$$\Sigma = \sigma^2 \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Mixture of Gaussians: Updating the Background Model

- Compare the pixel value against the existing Gaussians
    - Match if within 2.5 standard deviations (Mahalanobis distance:)

$$\sqrt{(X_t-\mu_{t-1})\cdot\Sigma_{t-1}^{-1}\cdot(X_t-\mu_{t-1})}<2.5\cdot\sigma_{t-1}$$

- No match:
    - The Gaussian with lowest weight is replaced

Mixture of Gaussians: Updating the Background Model

- Update all weights ω (IIR-filter):
  - $\omega_{k,t}=(1-\alpha)\omega_{k,t-1}+\alpha\cdot M_{k,t}$
  - Where α is the learning rate and *M* is 1 for the model that match (0 otherwise)
- Update the matched Gaussian with X_t
  - $\mu_{k,t}=(1-\rho_t)\omega_{k,t-1}+\rho_t\cdot X_t$
  - $\sigma_t^2=(1-\rho_t)\sigma_{t-1}^2+\rho_t\cdot(X_t-\mu_t)^T(X_t-\mu_t)$
  - Where $\rho_t=\alpha\eta(X_t,\mu_{t-1},\sigma_{t-1})$ is the learning factor

Mixture of Gaussians: Background Model Estimation

- The background model contains K Gaussians at each pixel
- It must be determined which pixels that belong to the background
    - The following criteria suggest that a pixel belongs to the background:
        - Large weight ω
            - The background is present more often than foreground objects
        - Low variance σ
            - A pixel in a moving foreground object will typically have a larger variance than a pixel in the static background

Mixture of Gaussians: Background Model Estimation

- The weight ω and the variance σ can be combined as ω/σ to give the *fitness* value for each Gaussian
- Gaussians with larger fitness values are more likely to be background - thus the Gaussians are sorted according to this

Mixture of Gaussians: Background Model Estimation

- The amount of background is controlled by the parameter ***T***
  - ***T***=0.6 indicates that the background must be present *at least* 60% of the time
- The Gaussians belonging to the background are chosen according to the parameter ***T***: $B = argmin_b[(\Sigma_{k=1}^{b} \omega) > T]$

  (where Gaussians with larger fitness values are chosen first)

Mixture of Gaussians

- Problems and improvements

  (by KaewTraKulPong et. al., [2001], used in OpenCV)

  - Slow initial convergence
    - e.g. $\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha \cdot M$
    - Solution: Larger α in the beginning

      $$\alpha = max\left(\frac{1}{N+1}, \alpha_{specified}\right)$$

    - Similar solutions for the model parameters
  - Shadows
    - Detected by conversion to chromatic color space

      (Not implemented in OpenCV)

Mixture of Gaussians

- Problems and improvements

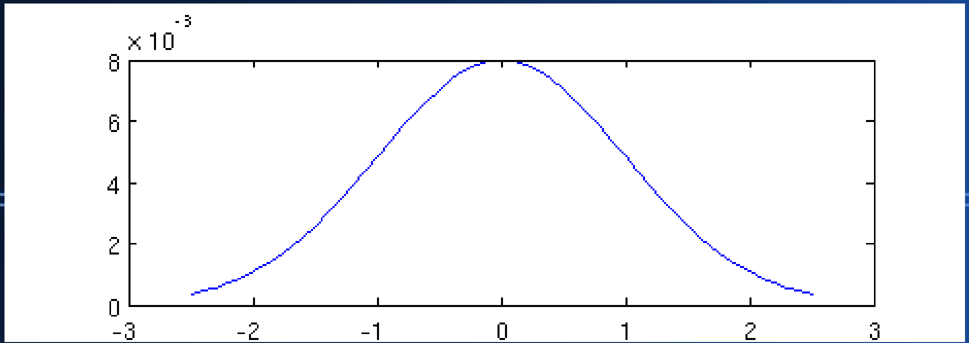
  (by KaewTraKulPong et. al., [2001], used in OpenCV)

  - Slow convergence in the model:

$$\mu_{k,t} = (1-\rho_t)\,\omega_{k,t-1} + \rho_t \cdot X_t$$

$$\sigma_t^2 = (1-\rho_t)\,\sigma_{t-1}^2 + \rho_t \cdot (X_t - \mu_t)^T (X_t - \mu_t)$$

  - Because of small $\rho_t = \alpha\, \cancel{\eta(X_t, \mu_{t-1}, \sigma_{t-1})}$
  - Example initial variance 50



Mixture of Gaussians

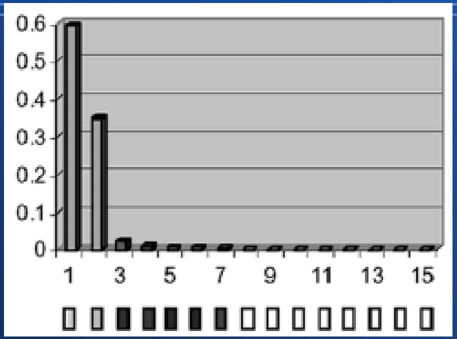
- Nice features:
  - Only stores variance, mean and weight for each Gaussian. Thus it doesn't use much memory.
  - Only two important parameters:
    - Learning rate, α
    - Background threshold, ***T***
- Limitations:
  - Practical implementations assume Gaussian distributions

Adaptive Foreground Detection: Histogram Based Techniques

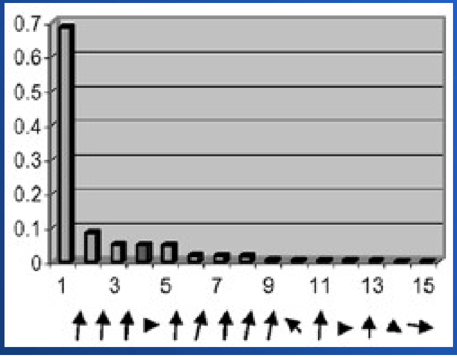
- Developed within the last decade
  - From [Liyuan Li et. al. 2004]
  - Best method implemented in OpenVC
- Uses different kinds of characteristics:
  - Spectral
  - Spatial
  - Temporal

Adaptive Foreground Detection: Histogram Based Techniques

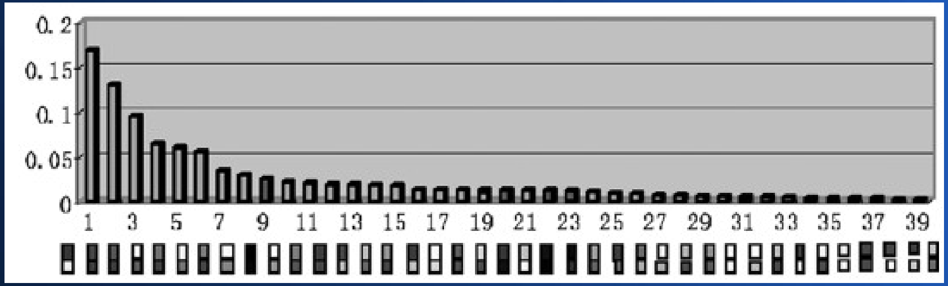
- Spectral:



- Spatial:

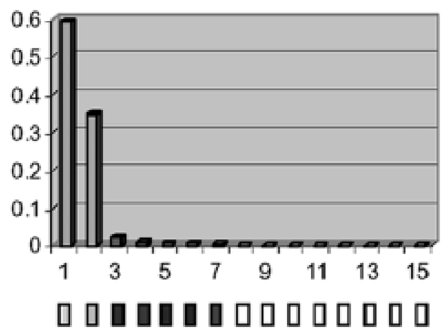


- Temporal:

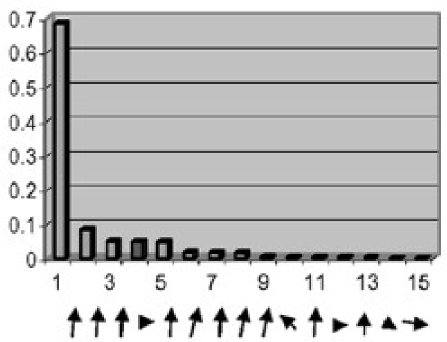


Adaptive Foreground Detection: Histogram Based Techniques

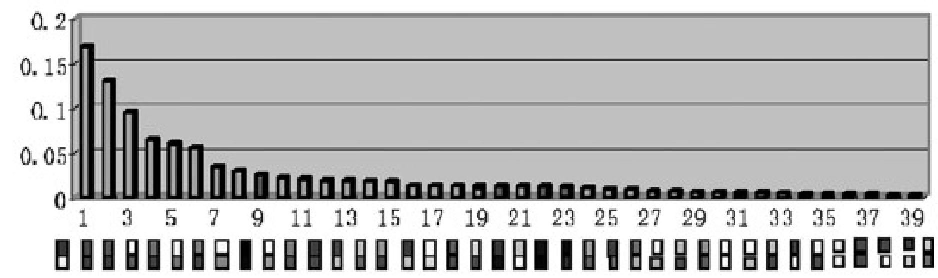
- Spectral:



- Spatial:



- Temporal:



- Note: Mixture of Gaussians only uses spectral

Adaptive Foreground Detection: Histogram Based Techniques

- Principle of the algorithm:
  - For each pixel and frame:
    - Determine if it is static or dynamic
      - Static: Use spectral and spatial characteristics
      - Spectral: Use spatial-temporal characteristics
    - Compare pixel-data from the new frame with existing statistics (histograms) to determine if it is background
    - Update statistics

Agenda

- Motivation and Examples
- Offline Foreground Detection
- Adaptive Foreground Detection
  - Introduction to adaptive foreground detection
  - Simple techniques
  - State-of-the-art algorithms
- Demo
- **Conclusion**

Conclusion

- Very efficient techniques exists
- In general, better algorithms require more memory
  - Histogram based techniques require separate histograms for each pixel
  - Can require many GB of memory
- A lot of research have been done, especially within Gaussian approaches
  - Specialized solutions for most scenarios exist

Agenda

- Motivation and Examples
- Offline Foreground Detection
- Adaptive Foreground Detection
    - Introduction to adaptive foreground detection
    - Simple techniques
    - State-of-the-art algorithms
- **Demo**
- Conclusion

References

[2000]: Article by the original authors of MoG
Stauffer, C.; Grimson, W.E.L.
Learning patterns of activity using real-time tracking
Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.22, no.8, pp.747-757, Aug 2000
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=868677&isnumber=18808

[2001]: Improved MoG partly used in OpenCV
P. KaewTraKulPong and R. Bowden
An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.12.3705&rep=rep1&type=pdf

[2004]: An alternative method, not assuming Gausian distribution (memory demanding)
Liyuan Li; Weimin Huang; Irene Yu-Hua Gu; Qi Tian,
Statistical modeling of complex backgrounds for foreground object detection
Image Processing, IEEE Transactions on , vol.13, no.11, pp.1459-1472, Nov. 2004
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1344037&isnumber=29595

[2008]: A Survay on MoG
T. Bouwmans, F. El Baf, B. Vachon
Background Modeling using Mixture of Gaussians for ForegroundDetection - A Survey
http://hal.archives-ouvertes.fr/docs/00/33/82/06/PDF/RPCS_2008.pdf