

# People Tracking via a Modified CAMSHIFT Algorithm (DCABES 2009)

Fahad Fazal Elahi Guraya, Pierre-Yves Bayle  
and Faouzi Alaya Cheikh

Department of Computer Science and Media Technology,  
Gjovik University College Gjovik , Norway

Email: [fahad.guraya@hig.no](mailto:fahad.guraya@hig.no)

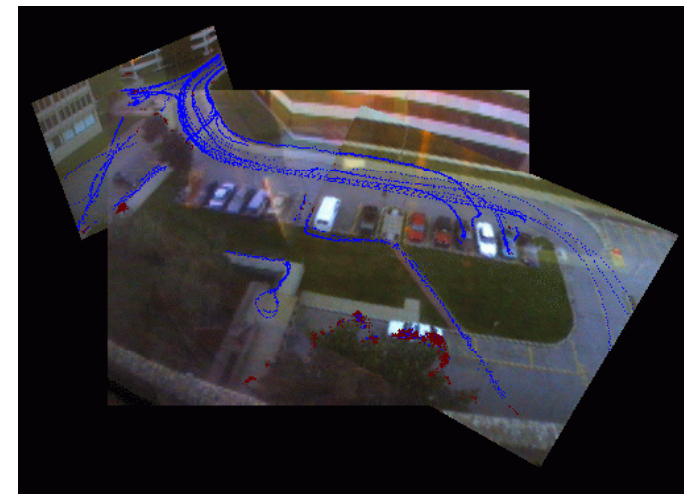
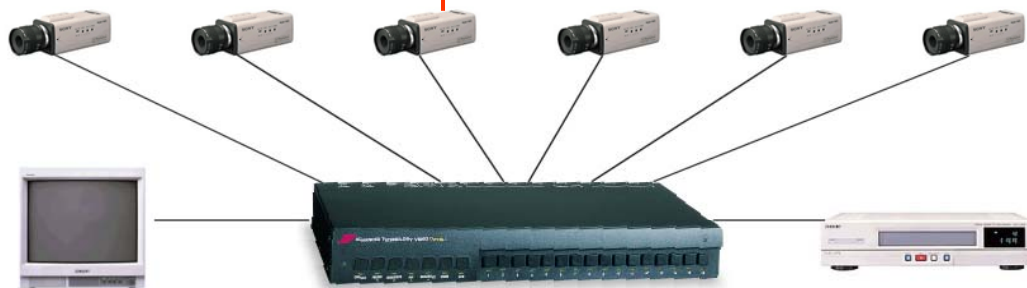




# Outline

- Introduction
- Problem Statement
- Tracking Algorithms
- Mean-shift tracking
- CAMSHIFT tracking
- Extended CAMSHIFT tracking
- Implementation
- Experimental Results
- Conclusion

# Introduction



- Multi-Camera Cooperated Object Detection, Tracking, and Event Analysis

fahad.guraya@hig.no

Gjøvik University College Norway

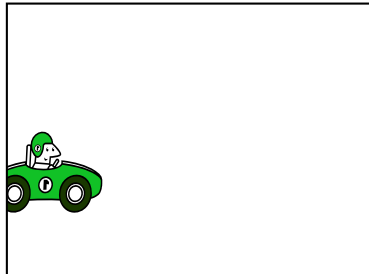


# Introduction (Applications)

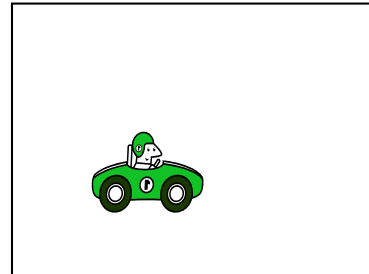
- **Motion-based recognition:**
  - Human identification based on gait, automatic object detection, etc.
- **Automated Video surveillance:**
  - Monitoring a scene to detect suspicious activities or unlikely events
- **Video indexing:**
  - Automatic annotation and retrieval of the videos in multimedia databases
- **Human-computer interaction:**
  - Gesture recognition, eye gaze tracking for data input to computers, etc.
- **Traffic monitoring:**
  - Real-time gathering of traffic statistics to direct traffic flow
- **Vehicle navigation:**
  - Video-based path planning and obstacle avoidance capabilities

# Problem Statement

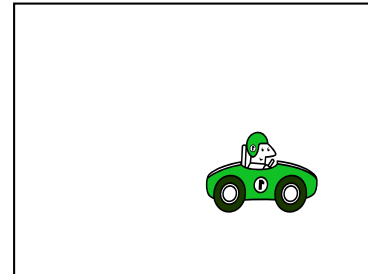
Frame 1



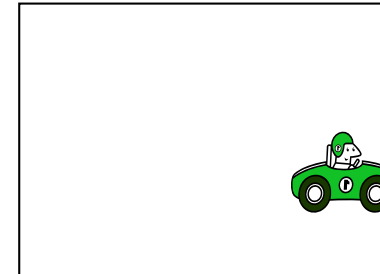
Frame 2



Frame 3



Frame 4



- Can we estimate the position of the object?
- Can we estimate its velocity?
- Can we predict future positions?



# Problem Statement

- Given a Sequence of Images/frames
- Find center of moving object
- Camera might be moving or stationary

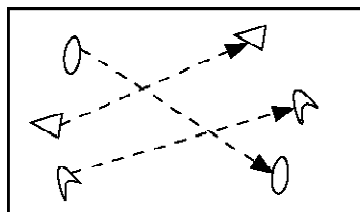
## **We Assume:**

- We can find object in individual frames

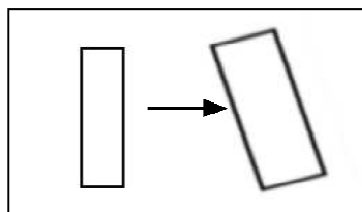
## **The Problem:**

- Track across multiple frames
- A fundamental problem in the field of video analysis

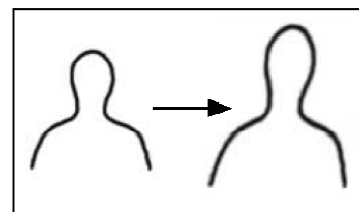
# Tracking Algorithms



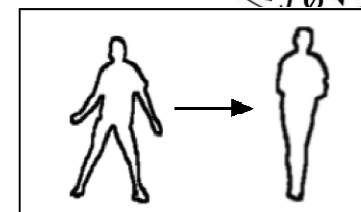
(a)



(b)



(c)



(d)

**a) Point Tracking by Bayesian Filters**  
**(Differ in representing probability densities (pdf))**

- Kalman Filters
- Particle Filters
- Grid-based approach
- Multi-hypothesis(MHT) filter

**b) Kernel Tracking**

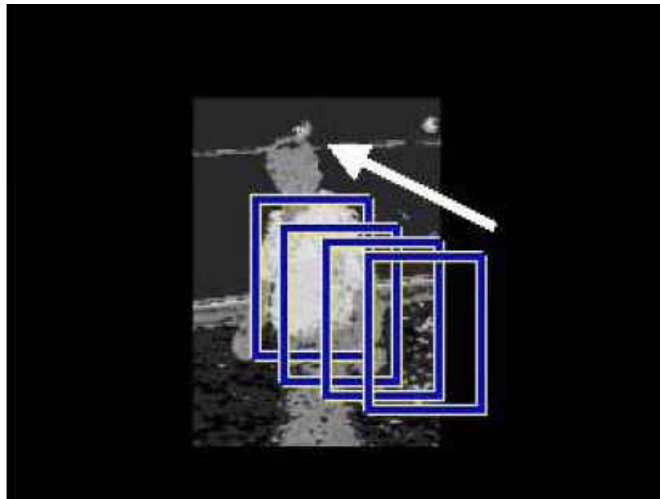
- Mean-shift tracking
- Continuously Adaptive Mean-shift (CAMSHIFT)
- Modified CAMSHIFT with Motion
- Kanade-Lucas-Tomasi Feature Tracker (KLT)
- Support Vector Machines (SVM) Tracker
- Eigen Tracking

**c,d) Silhouette Tracking**

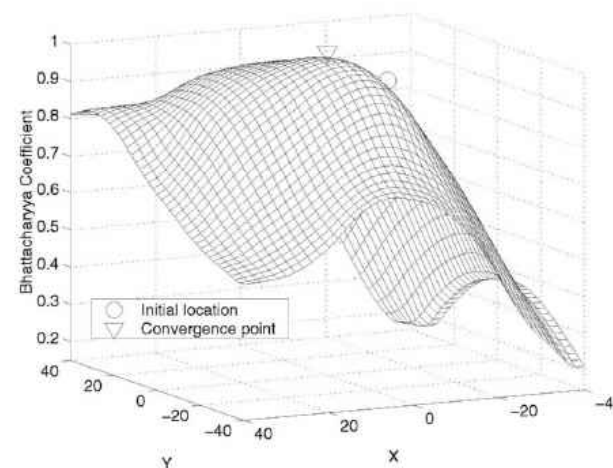
- Contour evolution state space models
- Contour evolution by variational methods
- Shape Matching hough transform

# Mean shift Tracking

- Introduced by Y. Cheng. in “Mean Shift, Mode Seeking, and Clustering” PAMI 1995
- Mean shift algorithm **climbs** the gradient of a probability distribution to find the nearest domain mode (**peak**)



@R. Collins CVPR 2003



@Comaniciu PAMI 2003



# Mean shift Tracking

- Given a **likelihood** image, find the optimal location of the tracked object
- The likelihood image is generated by computing, **at each pixel**, the **probability** that the pixel belongs to the object based on the **distribution of the feature**
- Obtain mean-shift vector  $\mathbf{y}$  by maximizing the Bhattacharyya coefficient, which is equivalent to minimizing the distance

maximize: 
$$\rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{\mathbf{y}}_0) \hat{q}_u} + \frac{C_h}{2} \sum_{i=1}^{n_h} w_i k \left( \left\| \frac{\mathbf{y} - \mathbf{x}_i}{h} \right\|^2 \right)$$

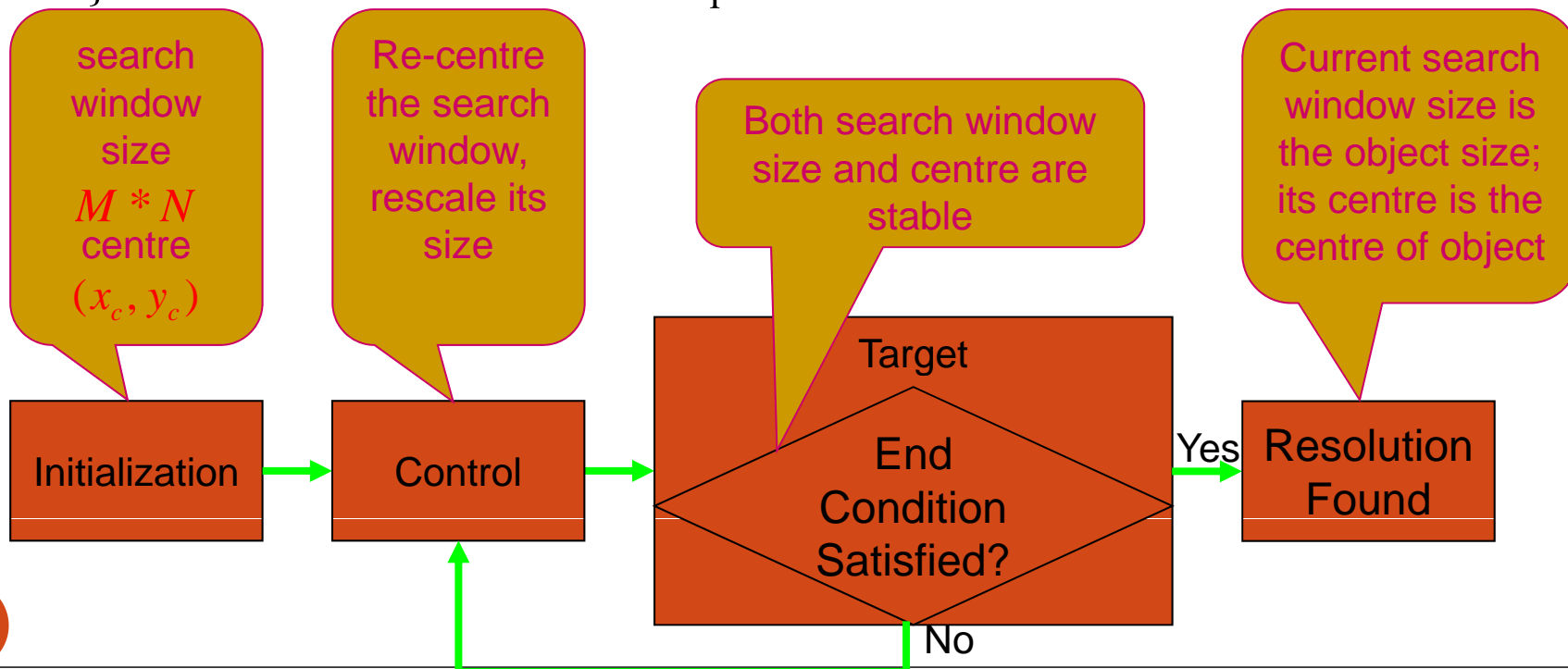
where

$$w_i = \sum_{u=1}^m \delta [b(\mathbf{x}_i) - u] \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{y}}_0)}} \leftarrow \text{Bhattacharyya coefficient for a single bin } \mathbf{u}$$

# Continuously Adaptive Mean-SHIFT (CAMSHIFT) Tracking



- Modified form of Mean-shift tracking algorithm
- Introduced by GR Bradski. in “Computer vision face tracking for use in a perceptual user interface”. Intel Technology Journal 1998
- Differs from Mean-shift: Search window adjusts itself in size.
- If we have well-segmented distributions(face) then CAMSHIFT will automatically adjust itself for the size of face as the person moves closer or further from camera.





# CAMSHIFT Tracking

1. Choose the initial location of the 2D mean shift search window
2. Calculate the color probability distribution in the 2D region centered at the search window location in an ROI slightly larger than mean shift window size
3. Run Mean Shift algorithm to find the search window center. Store the zeroth moment ( $M_{00}$ ) area or size and centroid location
4. For the next video frame, center the search window at the mean location stored in Step-3 and set the window size to a function of the zeroth moment  $M_{00}$  found there. Go to Step-2.

$$M_{00} = \sum_x \sum_y I(x, y)$$

$$M_{10} = \sum_x \sum_y xI(x, y)$$

$$M_{01} = \sum_x \sum_y yI(x, y)$$

$$x_c = \frac{M_{10}}{M_{00}}; y_c = \frac{M_{01}}{M_{00}}$$

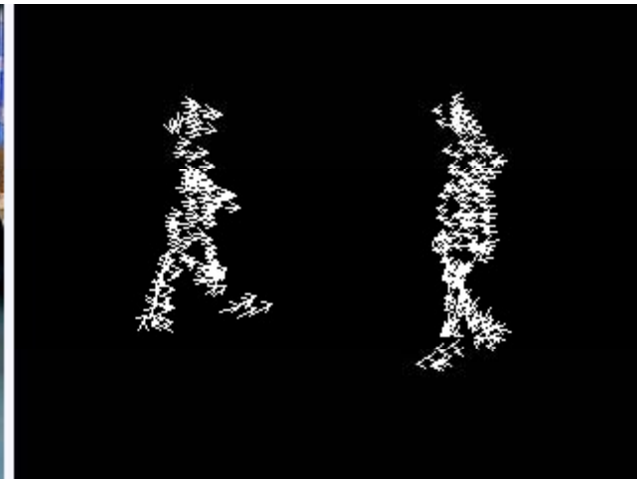
# Extended CAMSHIFT tracking

- Optical Flow computation by Lucas-Kanade Method
- Brightness Constancy, Temporal Persistence, Spatial Coherence

$$\varepsilon(\delta_x, \delta_y) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} (F_1(x, y) - F_2(x + \delta_x, y + \delta_y))$$



Video Frame



Motion Direction



# Extended CAMSHIFT tracking

- **Purposed Method:** Compute motion information of the moving objects and add it linearly to the back-projected color histogram
- **Steps:**
  - Select initial location of the person i.e ROI (Region of interest)
  - Compute equalized histogram of the ROI
  - Compute back projection image using the current histogram for the next frame
  - Compute the motion of the blob using Lucas-Kanade algorithm
  - Update the back-projection image using motion information
    - Combine direction of motion with back projection image linearly, give more weightage to pixels moving in same direction as in the previous frame and less weightage to pixels moving in the other directions
  - Use updated back-projection image to track object/s in the new frame



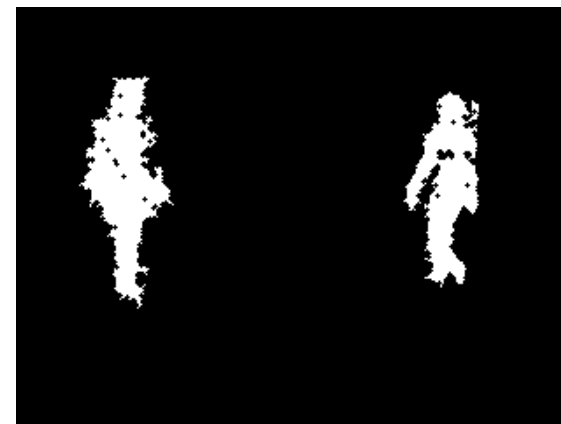
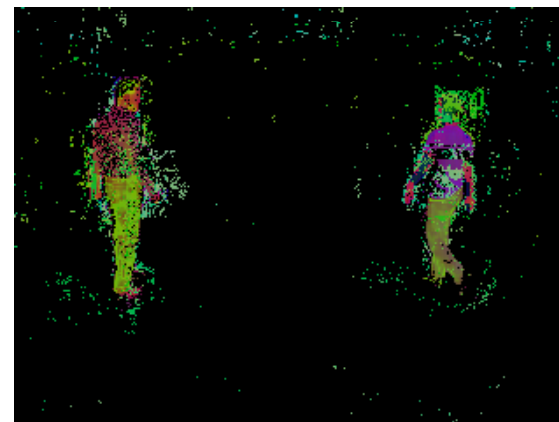
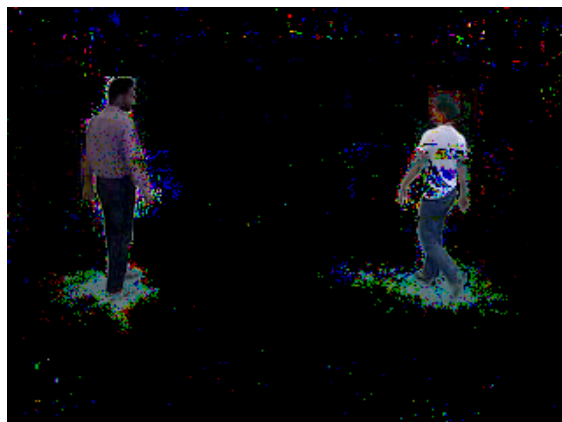
# Extended CAMSHIFT Tracking

- Pros:
  - Computationally efficient (robust statistics and probability distributions) - Working in real-time: **fast** processing
  - Robust to image noise
  - Robust to distractors (e.g. other objects)
  - Irregular object motion (linear / non-linear)
  - Robust to partial / full occlusion
  - Robust to background-foreground color
- Cons:
  - Need manual input to initialize template window

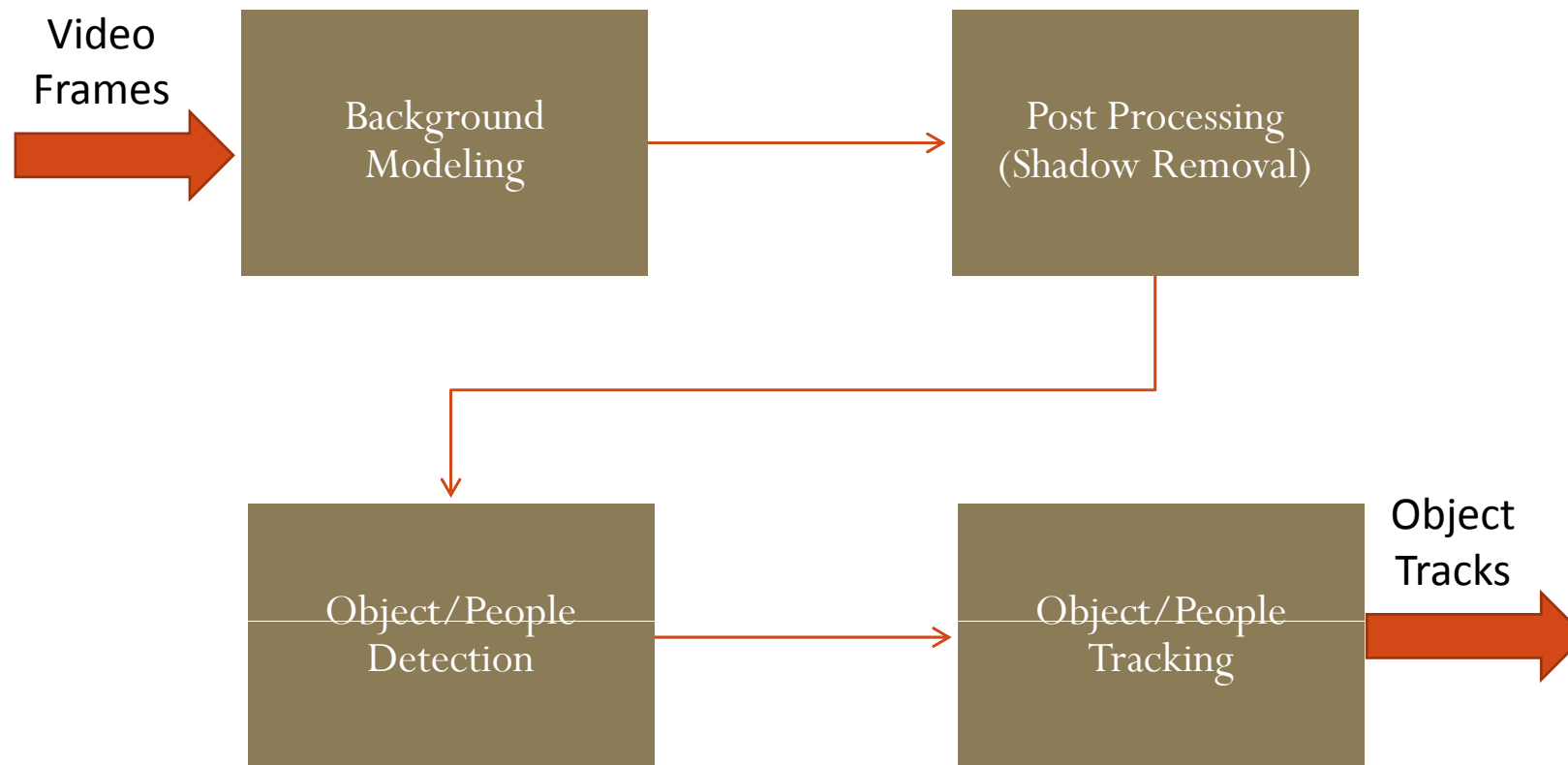
# Implementation



- OpenCV Computer Vision Library/ Visual C++



# Implementation

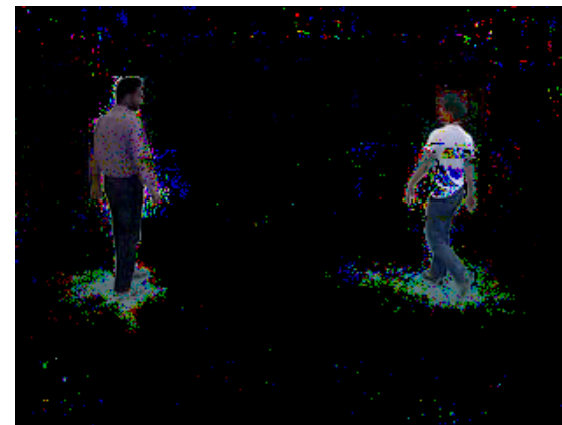




# Implementation(Background Modeling)



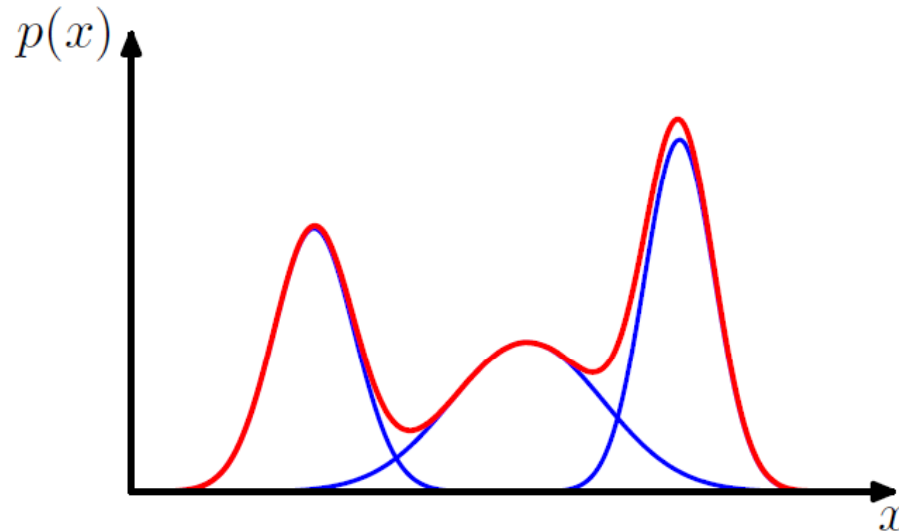
- Recursive techniques
  - Running Gaussian average (RGA)
  - Gaussian mixture model (GMM)
  - GMM with adaptive number of Gaussians (AGMM)
  - Approximated median filtering (AMF)
- Non-recursive techniques
  - Median filtering
  - Mediod filtering
  - Eigenbackgrounds (EigBG)



# Implementation (GMM)

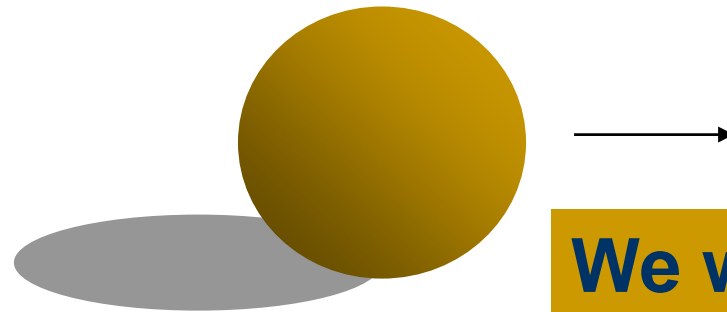
- Data is represented by a mixture of  $N$  Gaussians
- Different Gaussians represent different colors

$$p(x) = \sum_{l=1}^3 \pi_l \cdot \mathcal{N}(x|\mu_l, \sigma_l^2); \quad \sum_{l=1}^3 \pi_l = 1, \quad 0 \leq \pi_l \leq 1$$



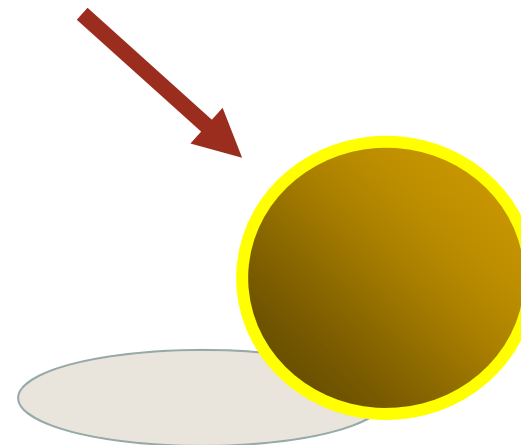
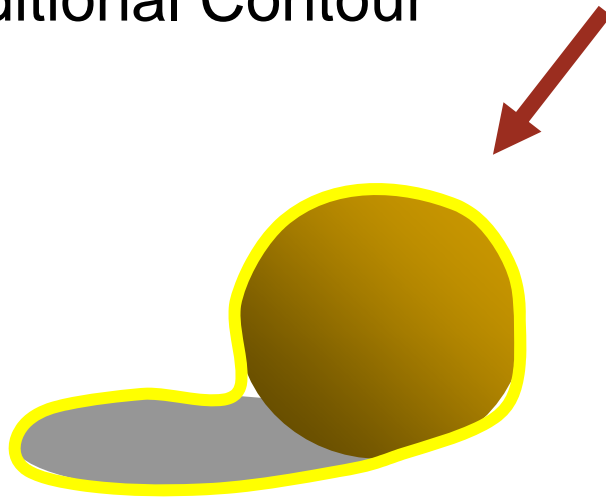
Linear combination of three Gaussians gives rise to very complex densities

# Implementation(Shadows Removal)



**We want to realize this!**

Traditional Contour





# Implementation(Shadows Removal)

- **Original background (B):** Brightness and chromaticity similar to those of the same pixel in the background image.
- **Shaded background (S):** Similar chromaticity but lower brightness.

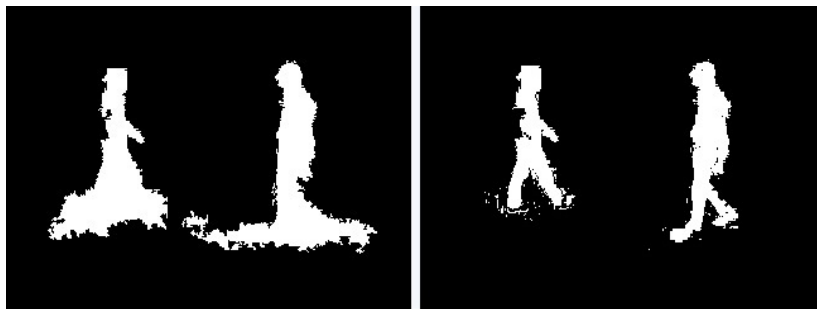
$$Shadow\{x,y\} = \begin{cases} 1 & \text{if } brightness_{img} < brightness_{bg} \\ & \& chromaticity_{img} = chromaticity_{bg} \pm T \\ 0 & \text{otherwise} \end{cases}$$

- **Moving foreground object (F):** Chromaticity different from the expected values in the background image.

# Experimental Results

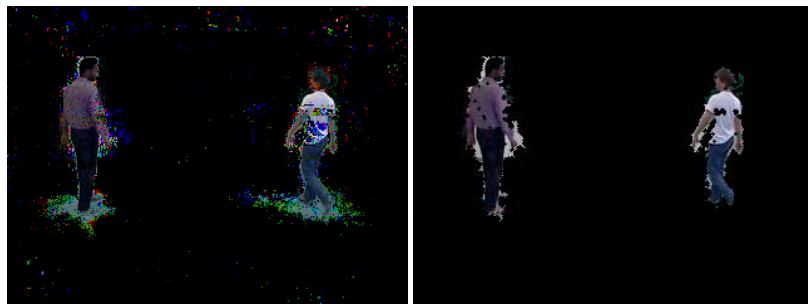
## Shadow Removal

$$Shadow\{x, y\} = \begin{cases} 1 & \text{if } brightness_{img} < brightness_{bg} \\ & \& chromaticity_{img} = chromaticity_{bg} \pm T \\ 0 & \text{otherwise} \end{cases}$$



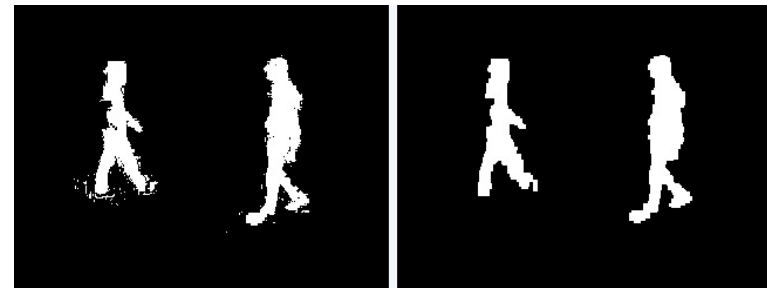
Background subtraction

Shadows removal



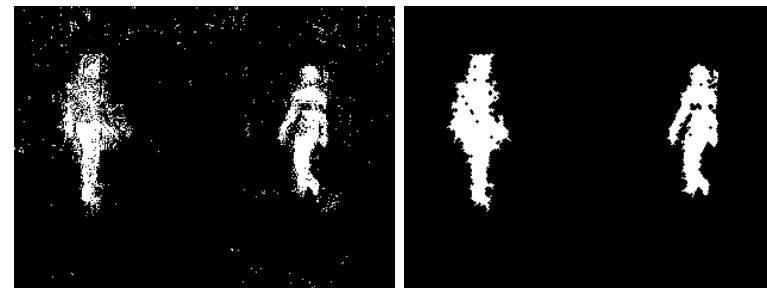
## Morphological filtering

- Closing operation  
Structuring element: 3x3
- Removing small regions



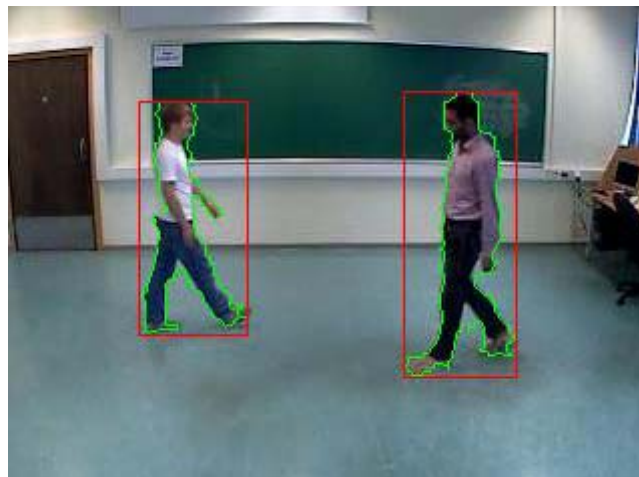
Before

After

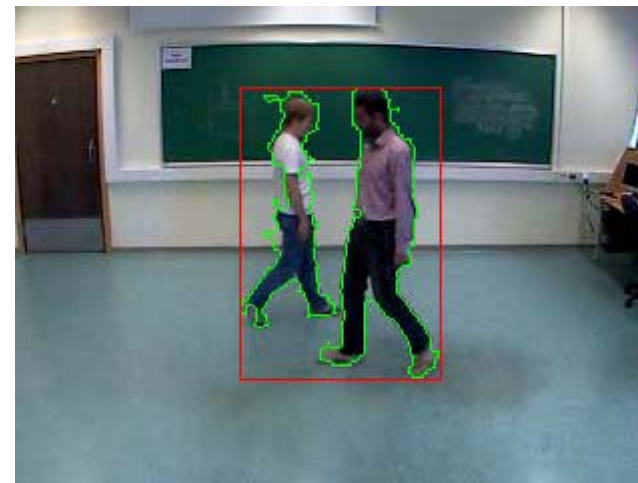


# Experimental Results

- Object/People Detection
  - Contours detection
  - Bounding box



Contours and bounding boxes for 2 separate persons



Contours and bounding boxes for 2 persons starting to overlap

# Experimental Results



Back projection images (with/without) motion information



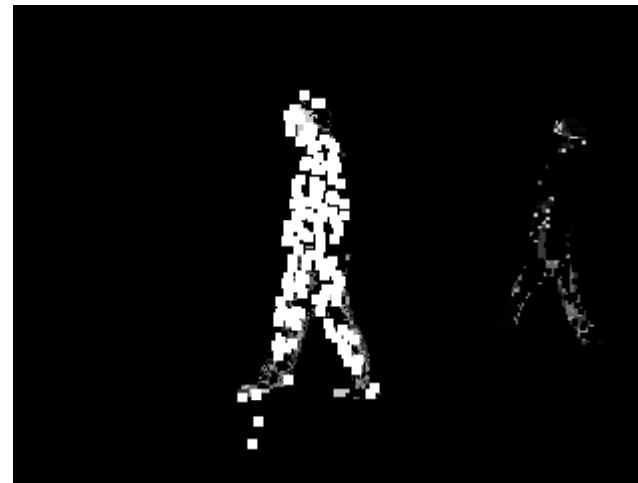
Backprojection person.1 using color feature



Backprojection person.2 using color feature



Backprojection person.1 using color and motion features

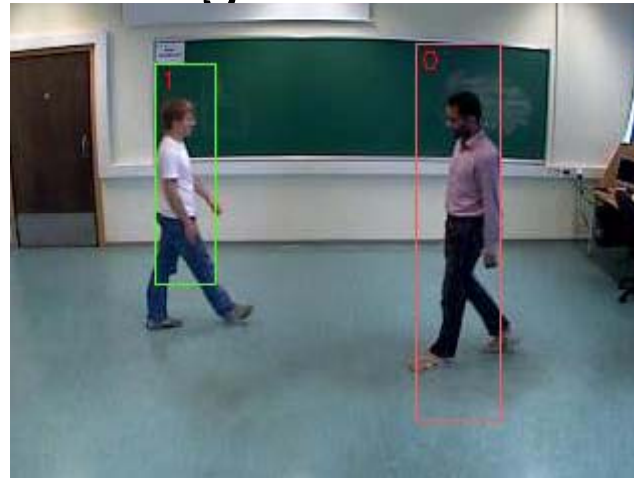


Backprojection person.2 using color and motion features

# Experimental Results



- Tracking Results using CamShift + Optical Flow



Tracking Without Motion Information

Tracking windows around moving persons



Tracking Using Motion Information





# Conclusion

- A modified CAMSHIFT algorithm is presented
- The Algorithm use color and motion features
- The Algorithm is tested and verified for a set of videos
- As future work the algorithm should be tested/verified for indoor/outdoor videos with strong shadows and partial/full occlusion of objects
- The algorithm should also be tested and modified for multiple objects and multiple camera tracking