中国科学技术大学
University of Science and Technology of China

Object Tracking

张举勇
中国科学技术大学

Overview

Track the location of target objects in each frame of a video sequence

Topics:

(1) Change Detection

(2) Gaussian Mixture Model

(3) Object Tracking using Templates

(4) Tracking by Feature Detection

Change Detection

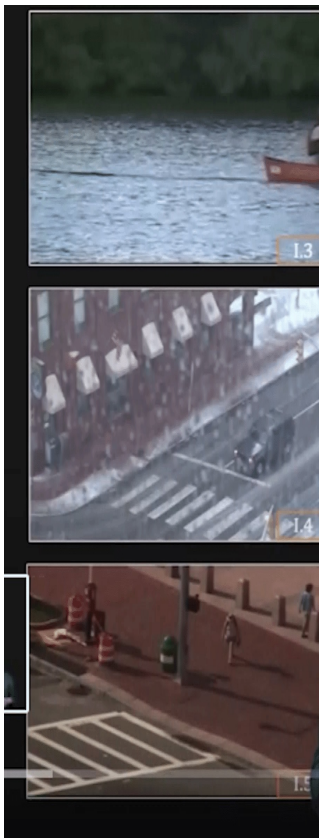
Given: Static cameras observing scene (room, street, etc.)
Find: Meaningful changes (moving objects, people, etc.)

Robust and real-time classification of each pixel as "foreground" (motion/change) or "background" (static).

Change Detection: Challenges

Ignore uninteresting changes:

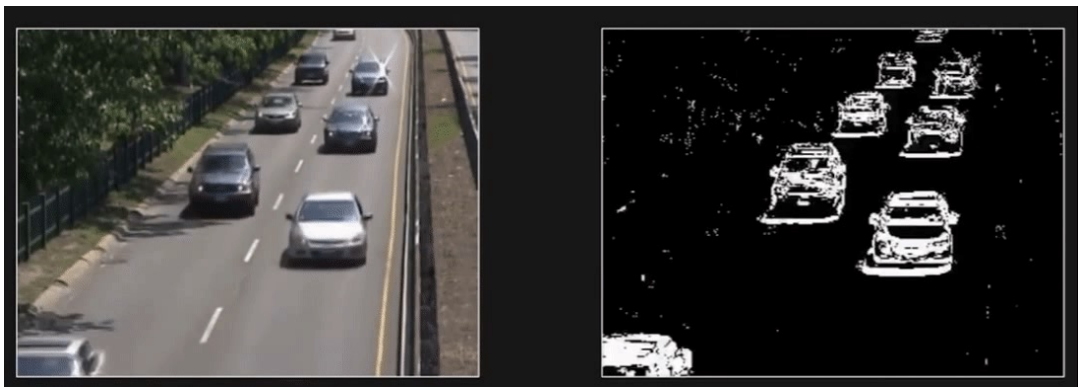
- Background fluctuations
- Image noise
- Rain, snow, turbulence
- Illumination changes & shadows
- Camera shake



Simple Frame Difference

Label significant difference between current and previous frames as background.

$$F_t = |I_t - I_{t-1}| > T \qquad T: \text{threshold}$$



Input video sequence Frame difference

Not Robust!

Background Modeling: Average

Build simple model of background before classification.

Background B
median{I_1, I_2,…, I_K}
(First K frames)

Input Frame I_t

Foreground F_t
$F_t = |I_t - B| > T$

Cannot handle change in lighting, background, etc.

Background Modeling: Median

Build simple model of background before classification.

Background $\mathrm{B_t}$
median{$\mathrm{I_{t-1}}$, $\mathrm{I_{t-2}}$,…, $\mathrm{I_{t-K}}$}
(Last K frames)

Input Frame $\mathrm{I_t}$

Foreground F_t
$F_t = |I_t - \mathrm{B}| > T$

Cannot handle change in lighting, background, etc.

Background Modeling: Moving Median

Build simple adaptive model of background over time.

Background B_t
median{I_{t-1}, I_{t-2},…, I_{t-K}}
(Last K frames)

Input Frame I_t

Foreground F_t
$F_t = |I_t - B| > T$

Requires keeping the last K frames in memory.
Finding median for each pixel is expensive.

Background Modeling: Moving Median

Build simple adaptive model of background over time.

Background B_t
median{I_{t-1}, I_{t-2},…, I_{t-K}}
(Last K frames)

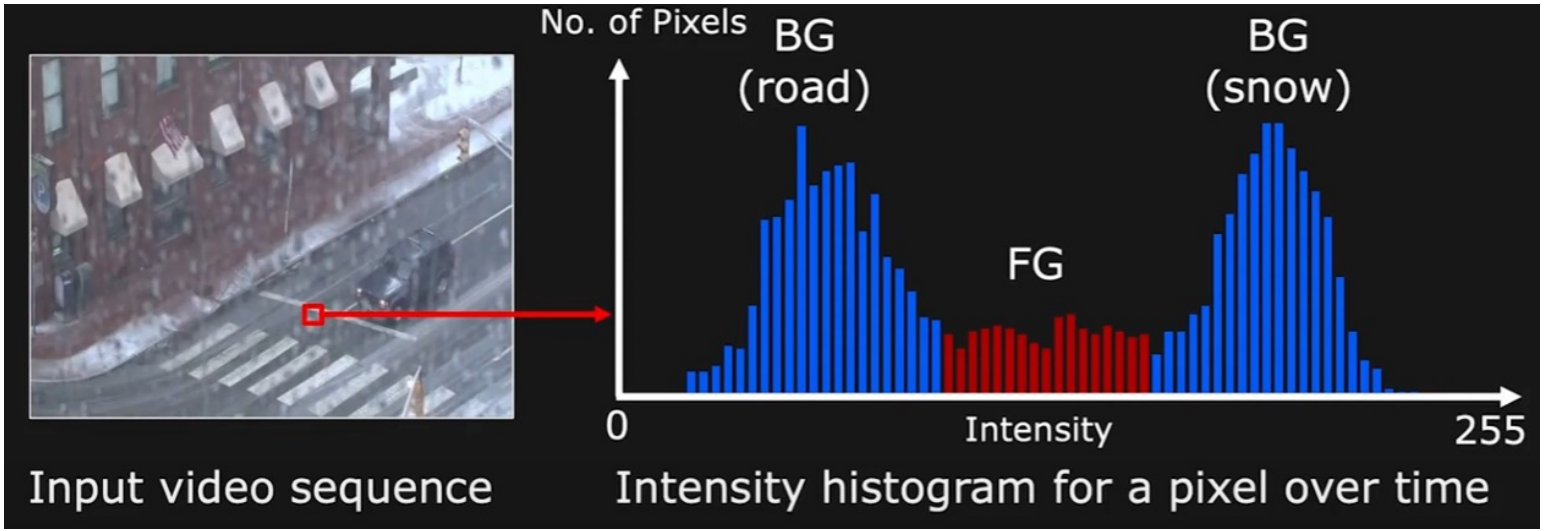
Input Frame I_t

Foreground F_t
$F_t = |I_t - B| > T$

Cannot handle significant pixel fluctuations
(weather shadow, shake, etc.)

Mixture Model

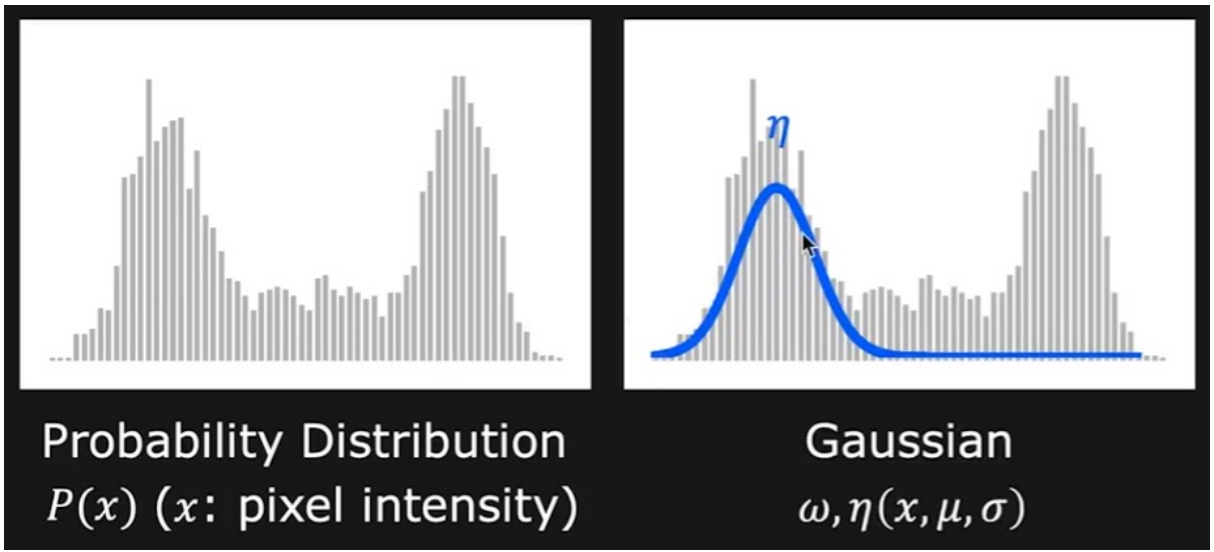
Intensity distribution at each pixel over time:



Intensity variations due to static scene (road), noise (snow), and occasional moving objects(vehicles)

Intuition: Pixels are background most of time.

Gaussian Model



1-Dimensional Gaussian:

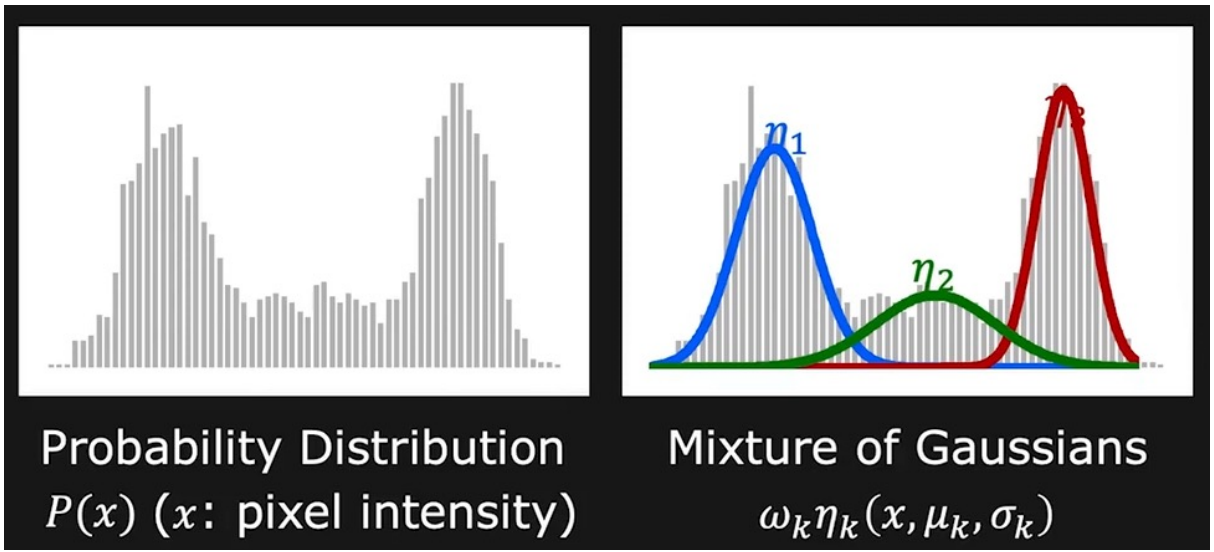
$$\omega\,\eta(x,\mu,\sigma) = \omega\frac{1}{\sqrt{2\pi}\sigma}\,e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

μ: Mean

σ: Std. Deviation

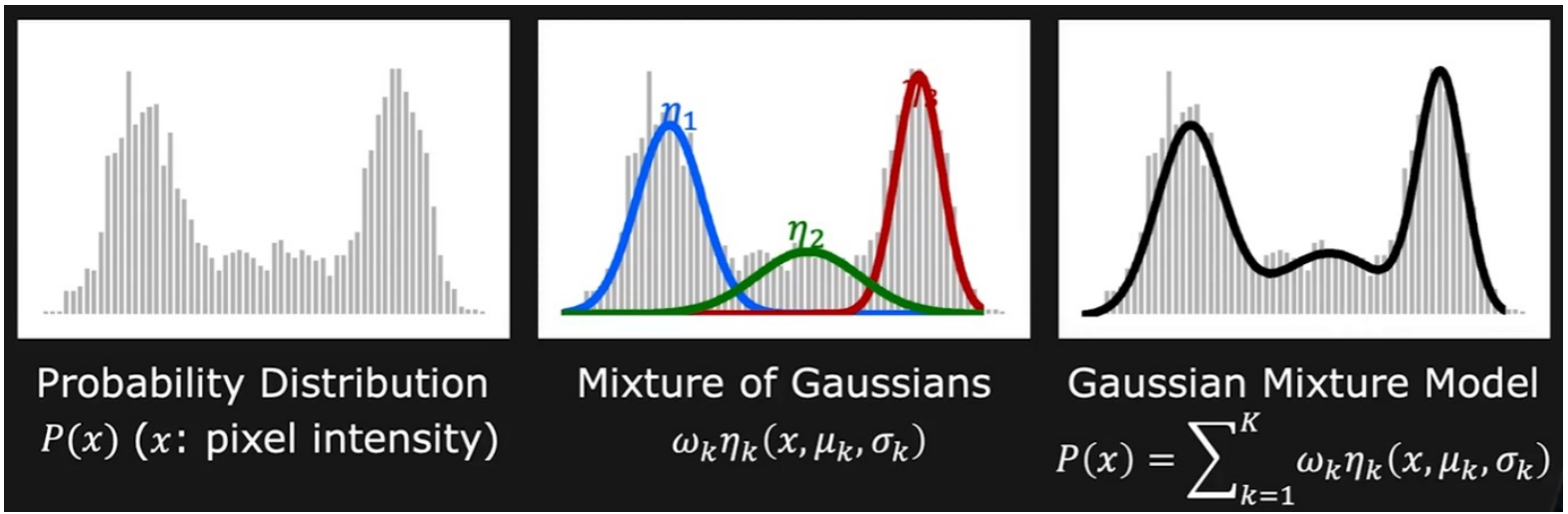
ω: Scale

Mixture of Gaussians



Assume P(x) is made of K different Gaussians.

Gaussian Mixture Model (GMM)



GMM Distribution: Weighted sum of K Gaussians

$$P(x) \cong \sum_{k=1}^{K} \omega_k \eta_k(x, \mu_k, \sigma_k) \quad \text{such that} \quad \sum_{k=1}^{K} \omega_k = 1$$

High Dimensional Model

Let $P(\mathbf{X})$ be a probability distribution of a D-dimensional random variable $\mathbf{X} \in \mathcal{R}^D$.For example: $\mathbf{X} = [r, g, b]^T$

GMM of $P(\mathbf{X})$: Sum of K D-dimensional Gaussians

$$P(\mathbf{X}) \cong \sum_{k=1}^{K} \omega_k \eta_k(\mathbf{X}, \boldsymbol{\mu}_k, \Sigma_k) \text{ such that } \sum_{k=1}^{K} \omega_k = 1$$

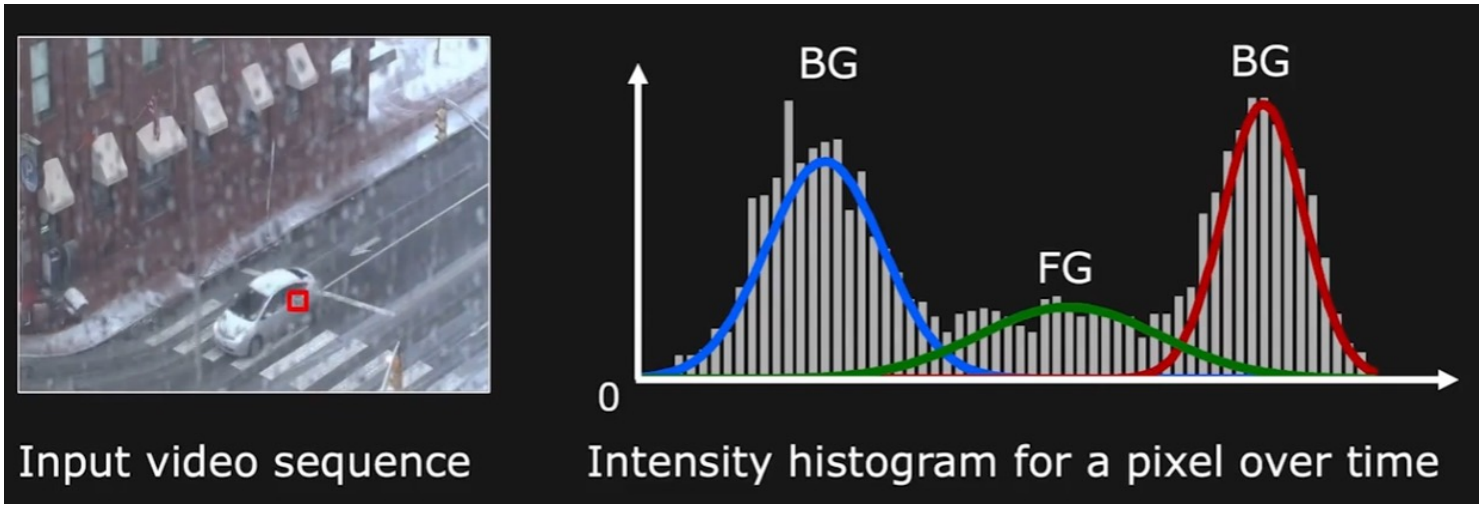
where: $\eta(\mathbf{X}, \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{X}-\boldsymbol{\mu})^T(\Sigma)^{-1}(\mathbf{X}-\boldsymbol{\mu})}$

Mean $\boldsymbol{\mu} = \begin{bmatrix} \mu_r \\ \mu_g \\ \mu_b \end{bmatrix}$ Covariance matrix $\Sigma = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}$ (can be a full matrix)

GMM can be estimated from $P(\mathbf{X})$. (MATLAB: gmdistribution.fit)

Background Modeling with GMM

Given: A GMM for intensity/color variation at a pixel over time
Classify: Individual Gaussians as foreground/background



Intuition: Pixels are background most of time.That is,Gaussians with large supporting evidence ω and small σ.

Large $\frac{\omega}{\sigma}$: Background

Small $\frac{\omega}{\sigma}$: Foreground

Change Detection using GMM

For each pixel:

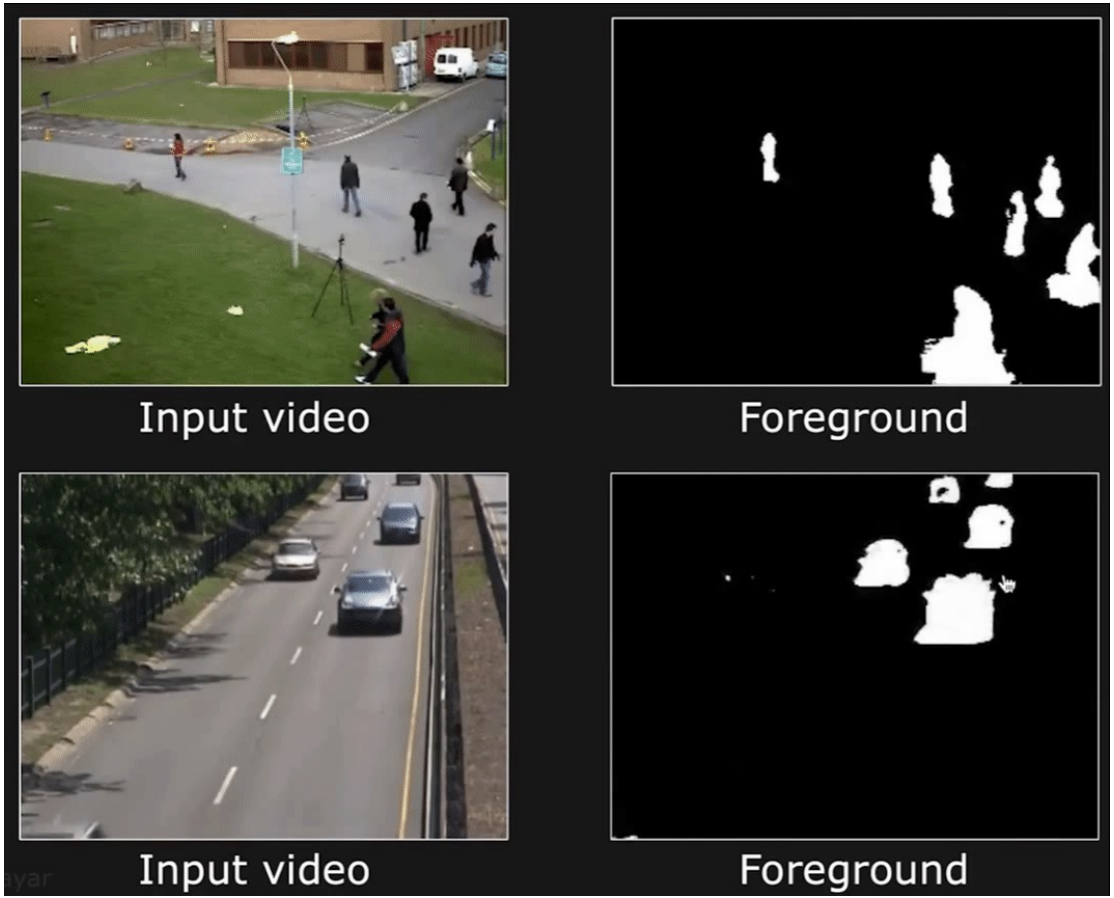
1. Compute pixel color histogram H using first N frames.
2. Normalize histogram: $\widehat{H} \leftarrow H/\| H \|$.
3. Model $\widehat{H}$ as mixture of K (3 to 5) Gaussians.
4. For each subsequent frame:
   a. The pixel value $\mathbf{X}$ belongs to Gaussian k in GMM for which $\|\mathbf{X} - \boldsymbol{\mu}_k\|$ is minimum and $\|\mathbf{X} - \boldsymbol{\mu}_k\| < 2.5\sigma_k$

   b. If ω_k/σ_k is large then classify pixel as background.
   Else classify as foreground.

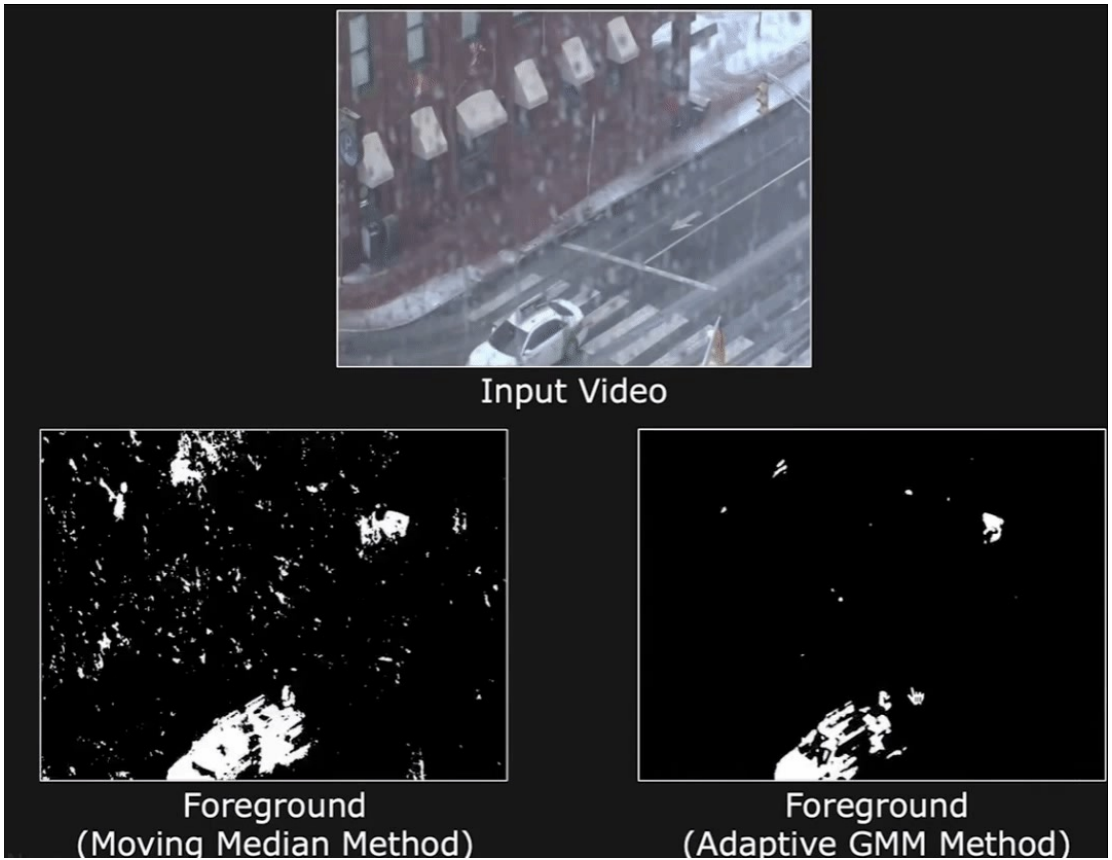
   c. Update histogram H using new pixel intensity.

   d. If $\widehat{H}$ and $H/\| H \|$ differ a lot ($\|\widehat{H} - {}^{H}/\|H \|\|$ is large), $\widehat{H} \leftarrow H/\| H \|$ and refit GMM.

Adaptive GMM based change detection



Adaptive GMM based change detection



Object Tracking

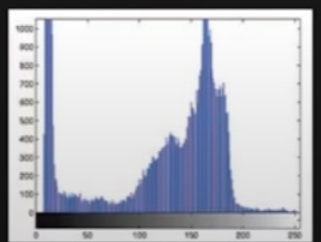
Given: Location of target in initial or previous frame.
Find: Location of target in current frame.

Target templates for Tracking

Appearance based Tracking:

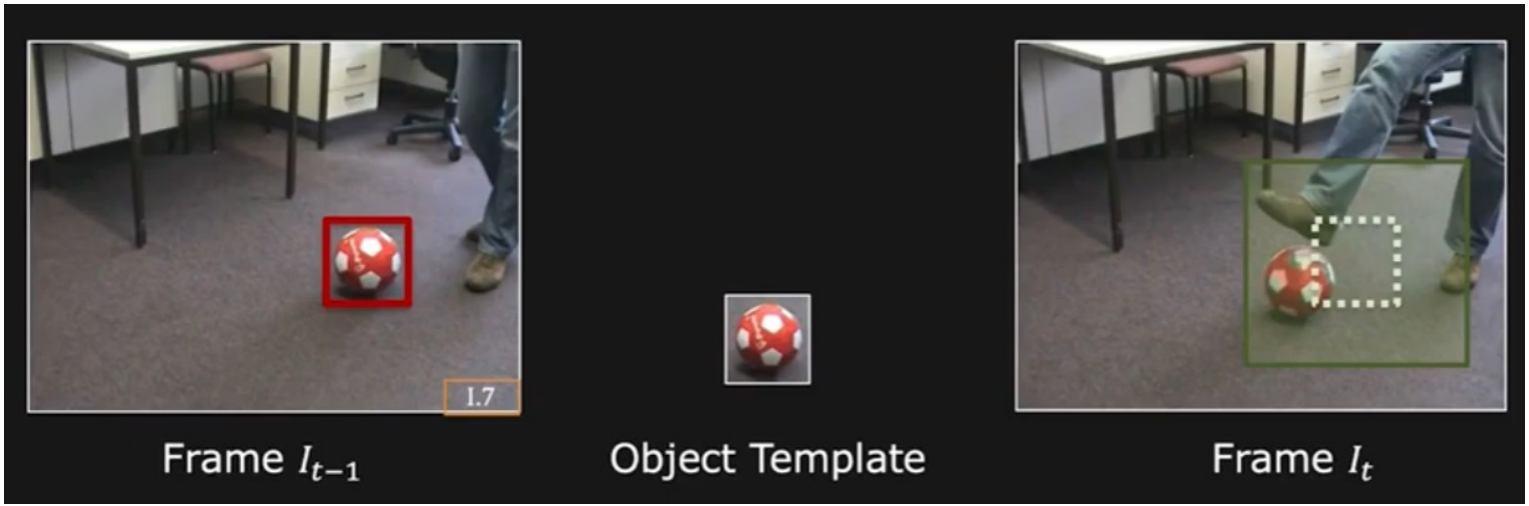
Image Template

Histogram based Tracking:



Histogram Template

Tracking using Appearance Matching



Given template window S in frame I_{t-1}, search neighborhood to find match in image I_t.

Simple implementation.Not robust to change in scale,viewpoint, Occlusion,etc.

Similarity Metrics for Template Matching

Find pixel $(k, l) \in S$ with Minimum Sum of Absolute Differences:

$$SAD(k, l) = \sum_{(i,j) \in T} |I_1(i, j) - I_2(i, +kj + l)|$$

Find pixel $(k, l) \in S$ with Minimum Sum of Squared Differences:

$$\mathrm{SSD}(k, l) = \sum_{(i,j) \in T} |I_1(i, j) - I_2(i + k, j + l)|^2$$

Find pixel $(k, l) \in S$ with Minimum Normalized Cross-Correlation:

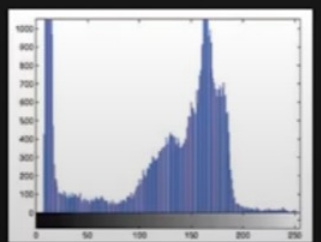
$$\mathrm{NCC}(k, l) = \frac{\sum_{(i,j) \in T} I_1(i, j) I_2(i + k, j + l)}{\sqrt{\sum_{(i,j) \in T} I_1(i, j)^2 \sum_{(i,j) \in T} I_2(i + k, j + l)^2}}$$

Target templates for Tracking

Appearance based Tracking:

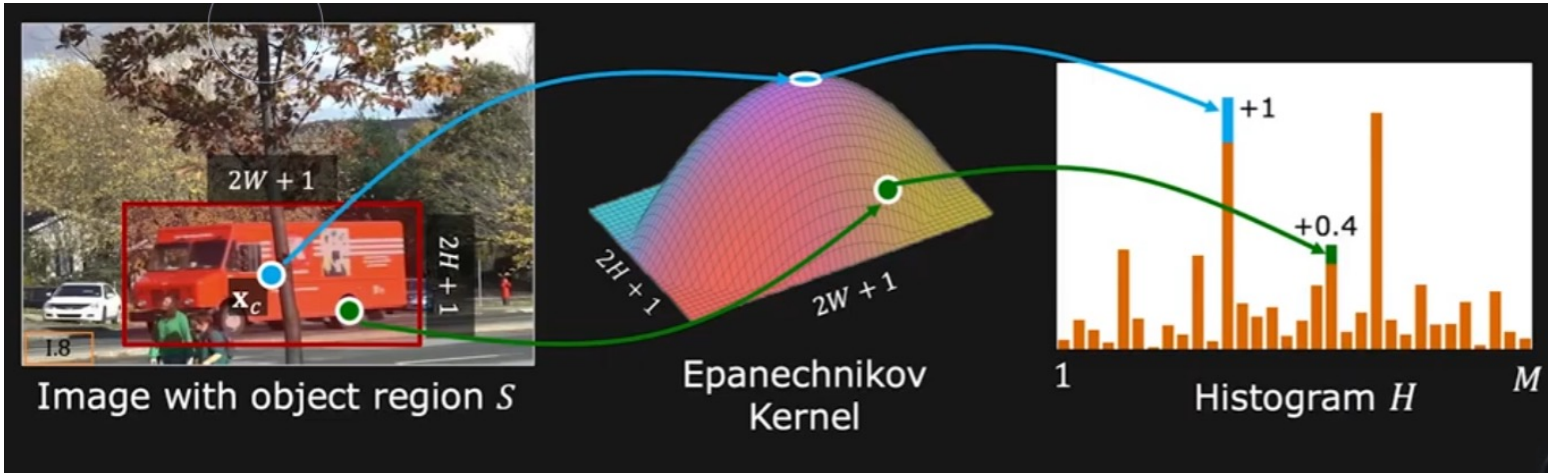
Image Template

Histogram based Tracking:



Histogram Template

Computing Weighted Histogram



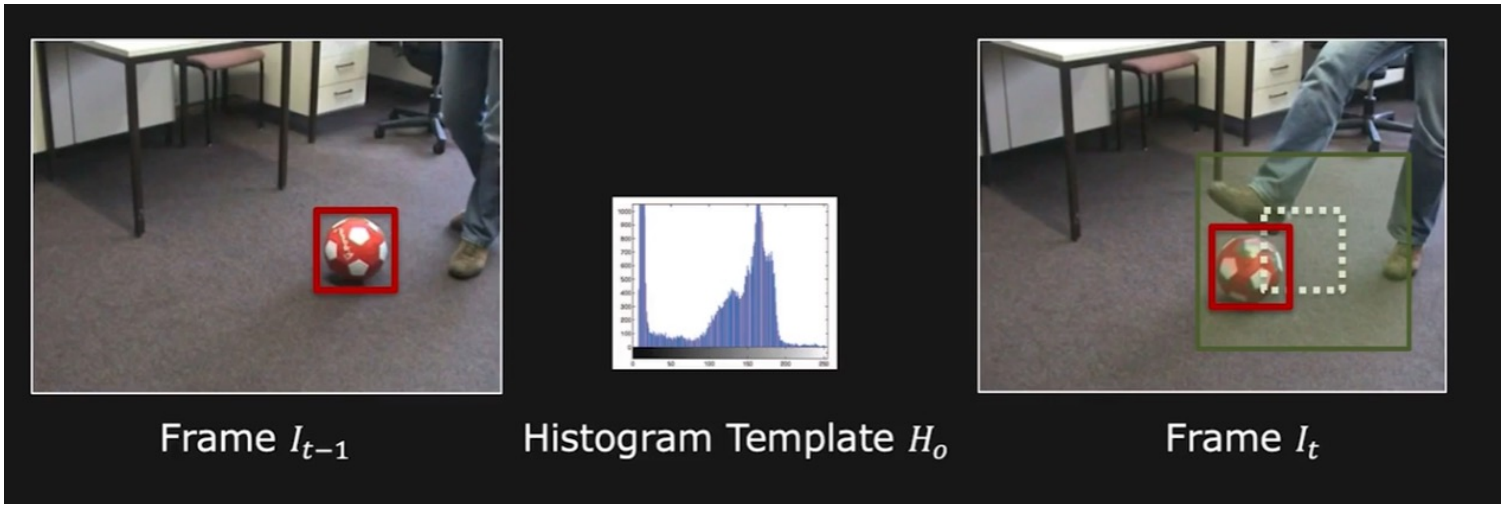
Weighted histogram gives more importance to pixels at center.

Epanechnikov Kernel:

$$k(\tilde{\mathbf{x}}) = \begin{cases} 1 - \| \tilde{\mathbf{x}} \|^2, & \| \tilde{\mathbf{X}} \| < 1 \\ 0, & \text{otherwise} \end{cases} \quad \tilde{\mathbf{x}} = \begin{bmatrix} (x - x_c)/W \\ (y - y_c)/H \end{bmatrix}$$

Comparing Histograms: Correlation, Intersection, etc.

Tracking using Histogram Matching



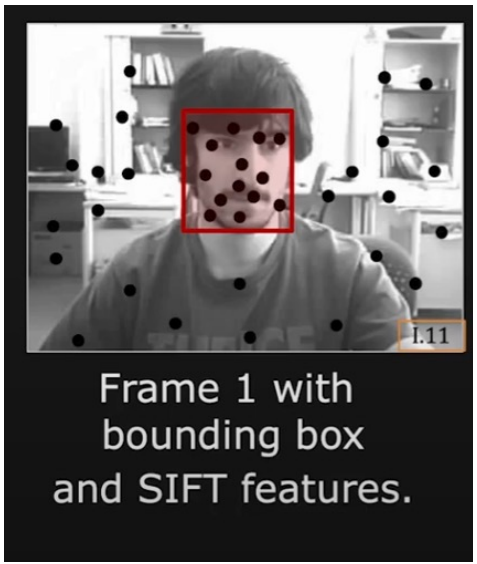
Given a histogram template H_0 and location x_{t-1} in I_{t-1}, search neighborhood in I_t to find window in matching histogram.

More resilient to changes in object pose and/or scale

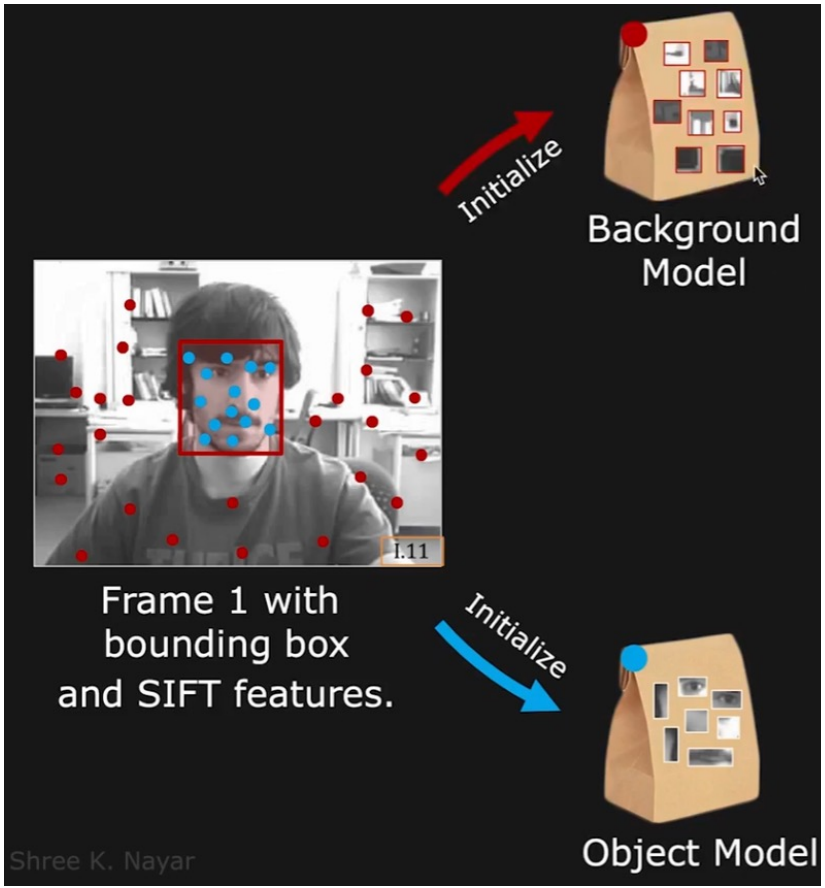
Histogram Based Tracking: Results

Robust when object appearance is unique in the environment and its size remains more or less the same.

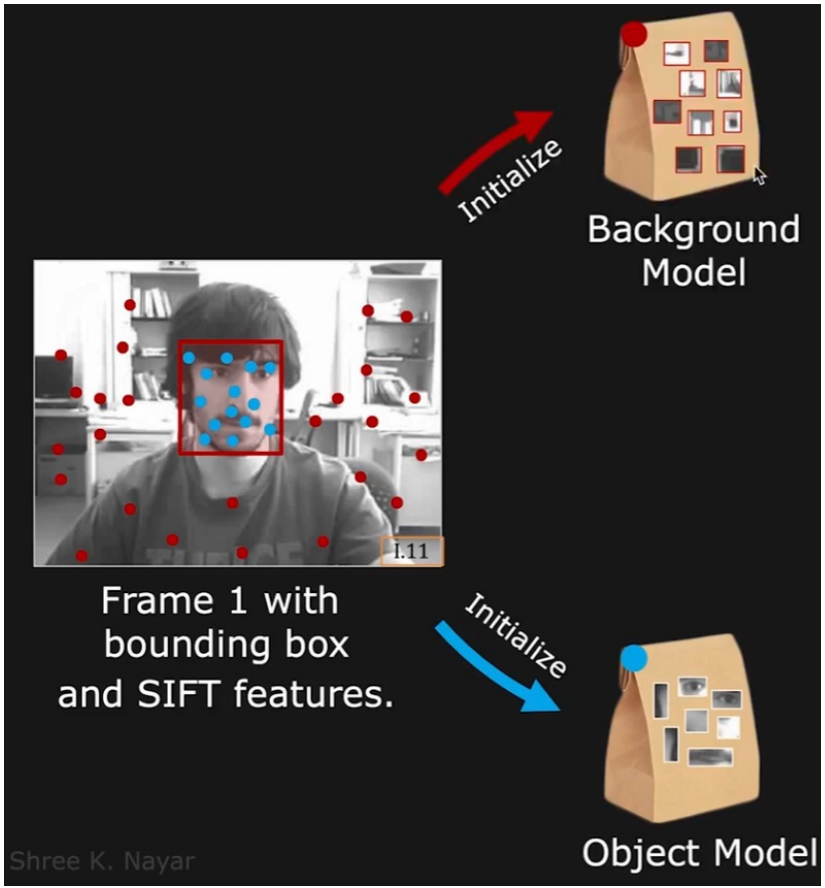
Tracking by Feature Detection

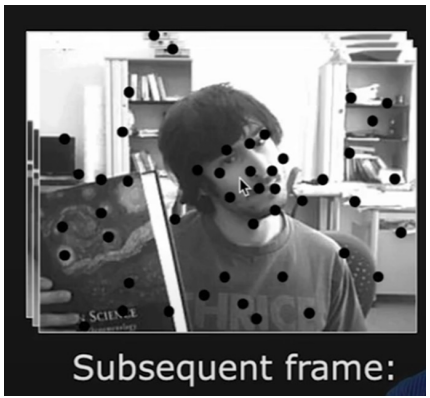


Tracking by Feature Detection

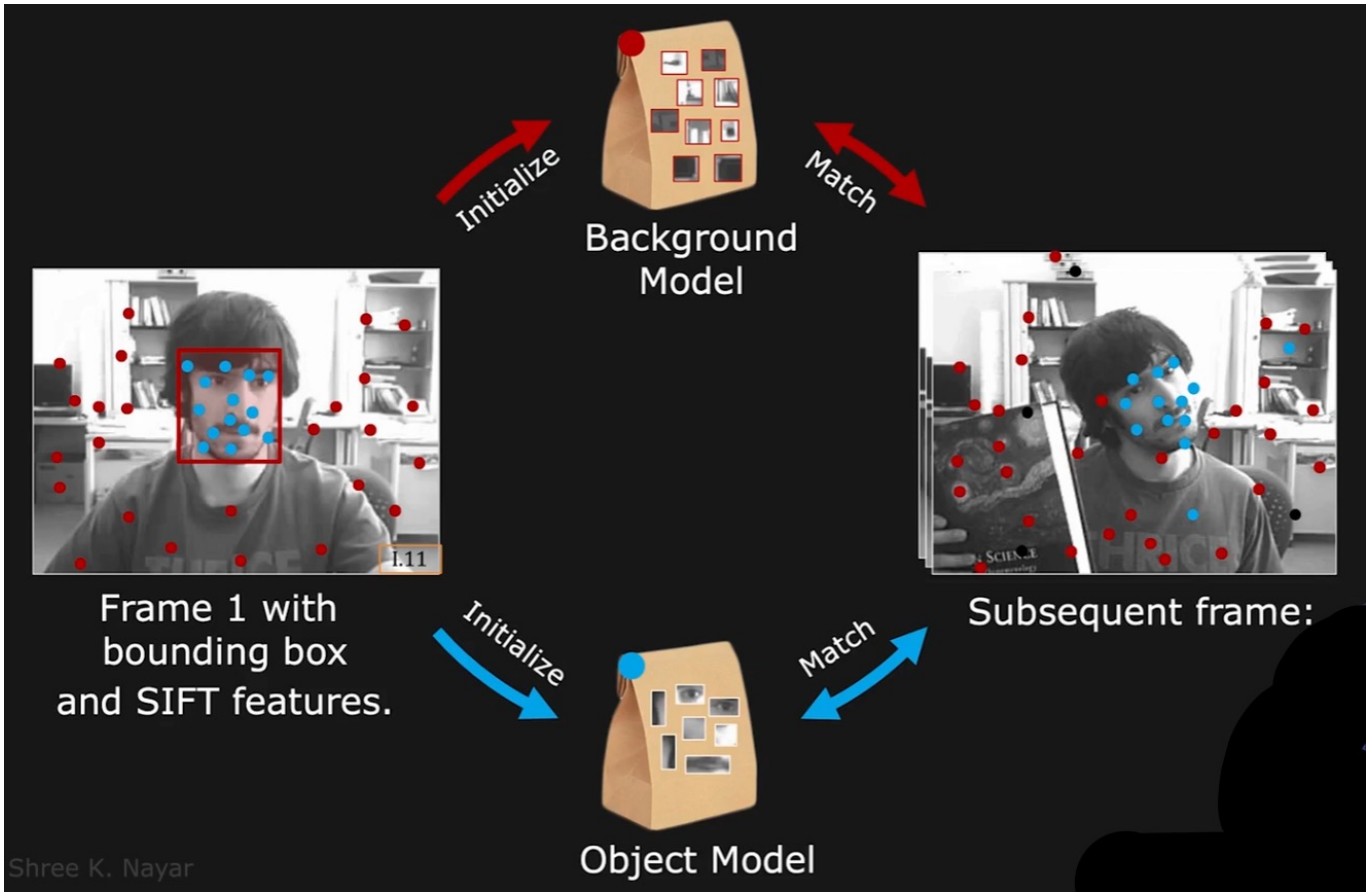


Tracking by Feature Detection

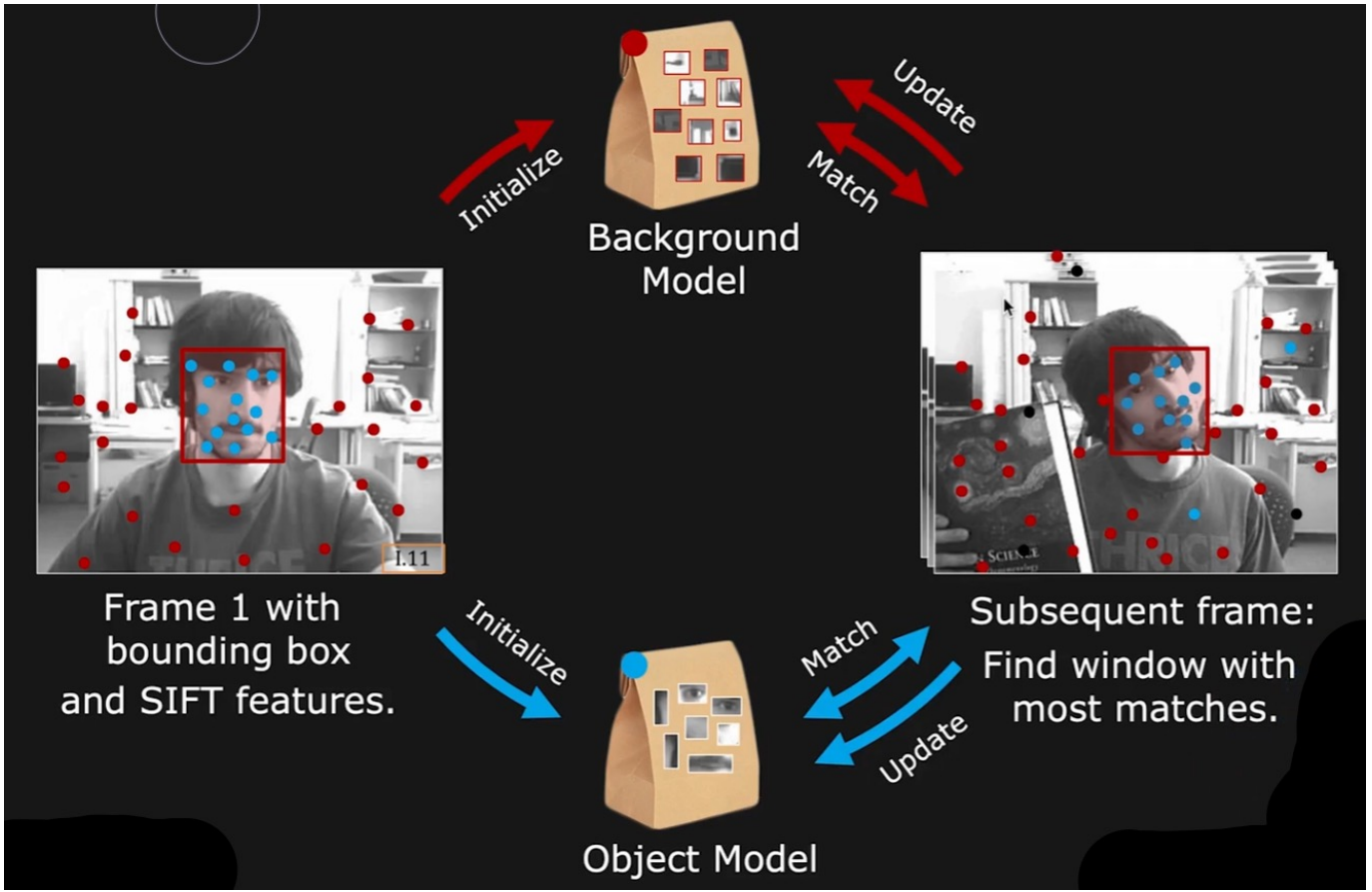




Tracking by Feature Detection



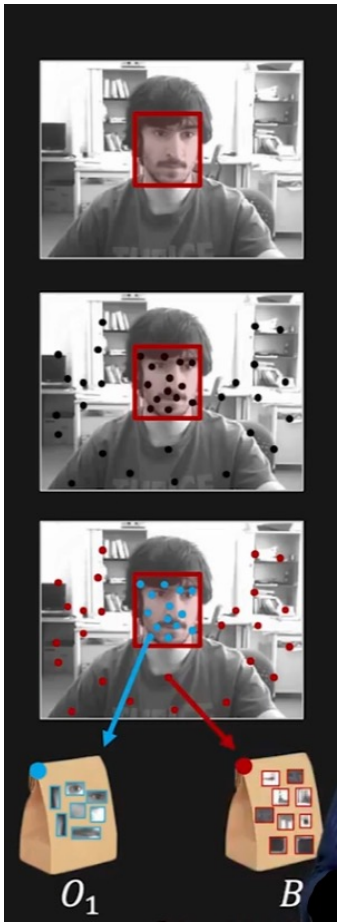
Tracking by Feature Detection



Tracking Intialization

At frame 1:

1. User selects a bounding box W_1 as object/target.
2. Compute SIFT (or similar) features for the frame.
3. Classify features within the box as object and assign them to set O_1.
4. Classify remaining features as background and assign them to set B.



Object Tracking

At frame t :

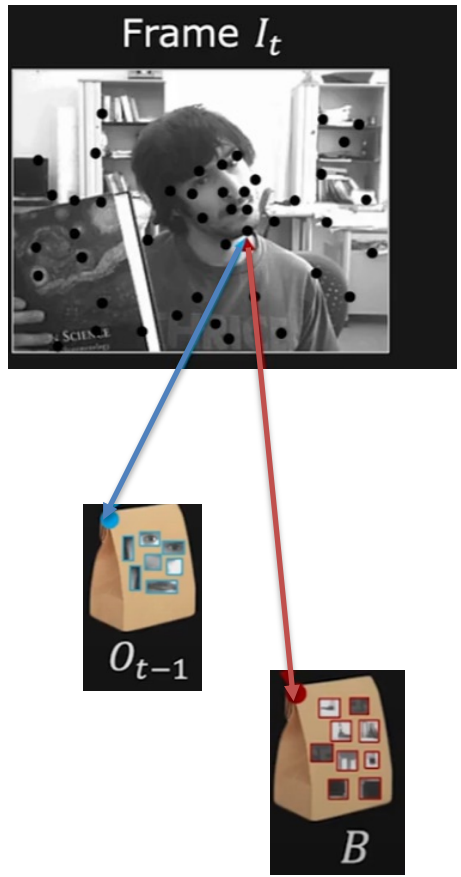
1. Compute SIFT features and SIFT descriptors $\{\mathbf{v}_1, \ldots, \mathbf{v}_K\}$ for frame I_t.
2. For each feature and corresponding descriptor $\mathbf{v}_i$:

   a. Compute distance d_o between $\mathbf{v}_i$ and the best match in object set O_{t-1}

   b. Compute distance d_B between $\mathbf{v}_i$ and the best match in background set B.

   c. $C(\mathbf{v}_i) =$

$$\begin{cases} +1 & \text{if } d_O/d_B < 0.5 (\mathbf{v}_i \text{ may belong to object }) \\ -1 & \text{otherwise } (\mathbf{v}_i \text{ does not belong to object }) \end{cases}$$

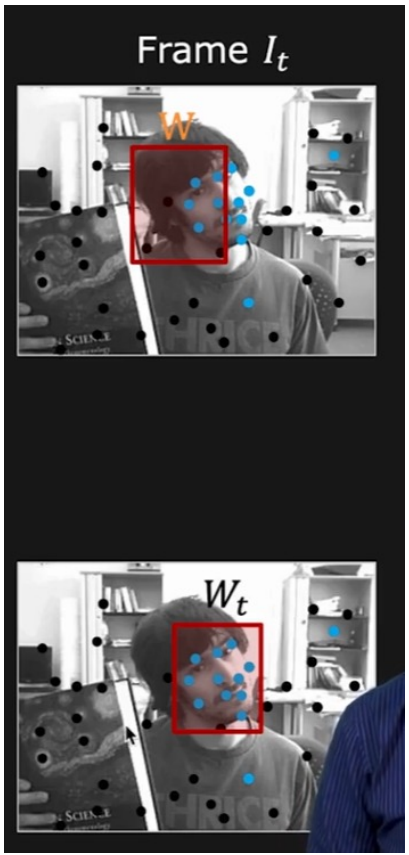


Object Tracking

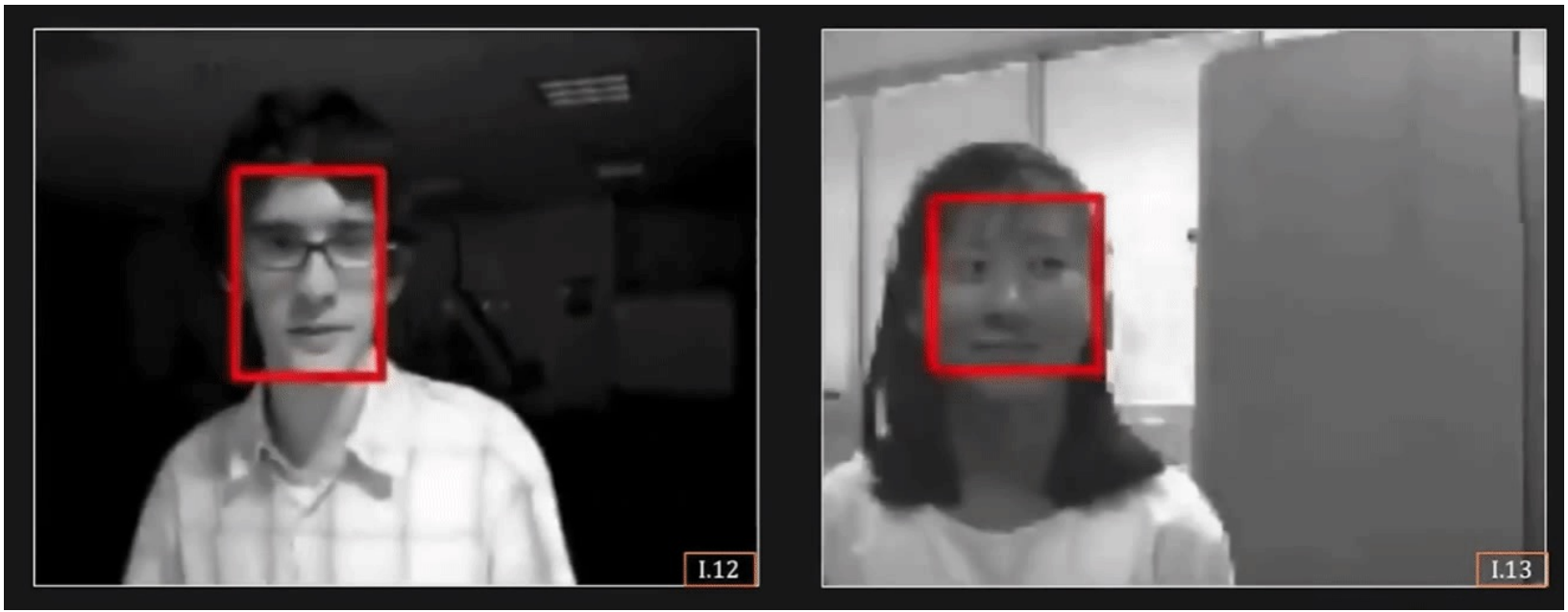
3. For each Search Window W :
   a. Compute $\varphi(W) = \sum C(\mathbf{v}_i)$ for all features $\mathbf{v}_i$ inside W.
   b. Compute a heuristic $\tau(W, W_{t-1})$ that penalizes large deviations from previous location, size and shape W_{t-1}.
   C. Compute Match Score
   $$\mu(W) = \varphi(W) - \tau(W, W_{t-1})$$

4. Select window W_t with the best match score as new object location.

5. Update object appearance model: $O_t = O_{t-1} \cup \{\mathbf{v}_i\} \forall \mathbf{v}_i$ inside W_t such that $C(\mathbf{v}_i) = +1$.



Tracking Results: Scale and Orientation

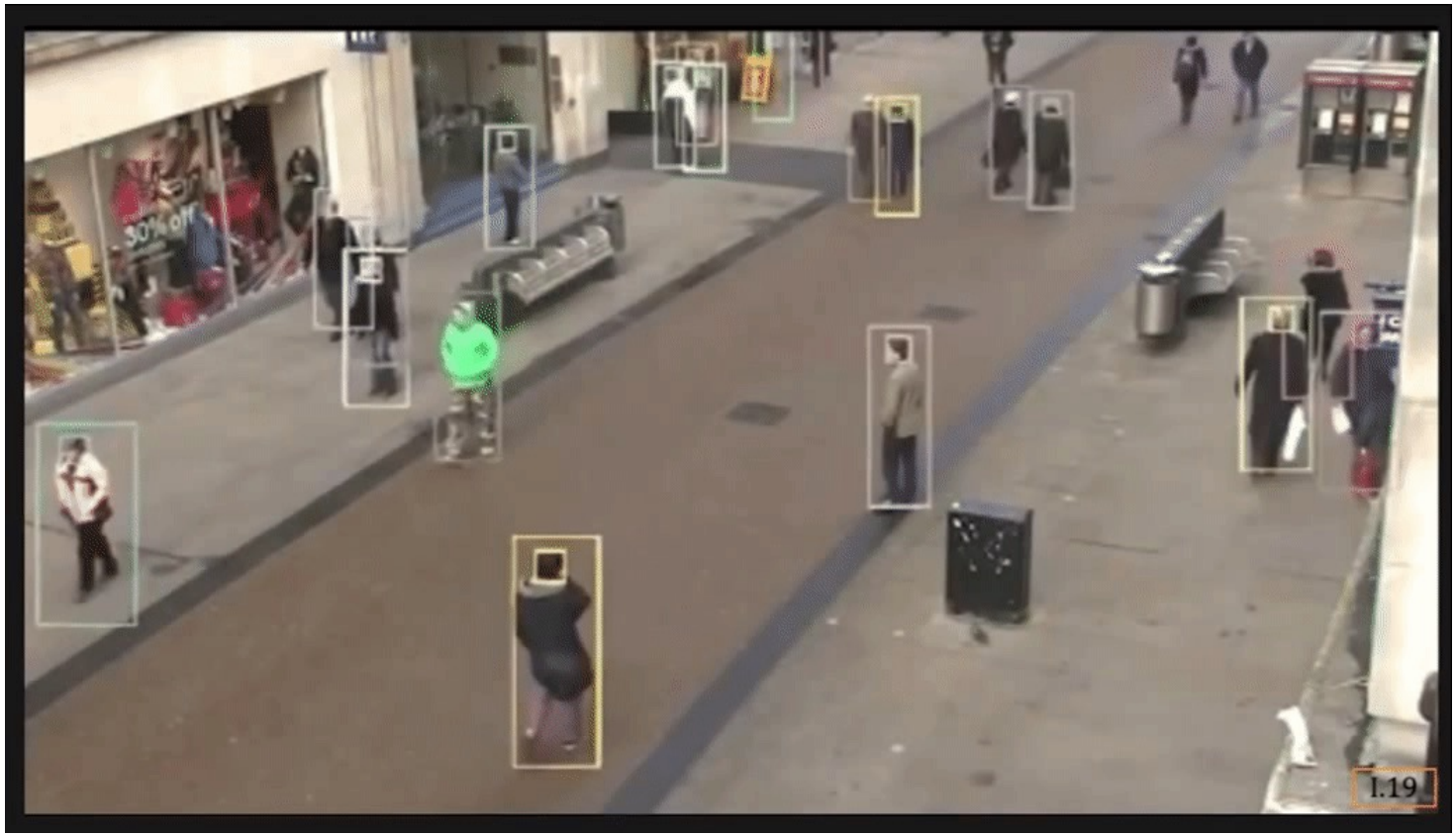


Resilient to changes in scale and orientation.

Tracking Results: Occlusion

Resilient to occlusion.

Tracking Applications



Tracking people in the wild.

Tracking Applications

Tracking people in the wild.

Tracking Applications

Traffic Monitoring.

Tracking Applications

Customer Behavior for In-Store Analytics.