Visual Tracking
Part I: Foundation

CSE P576 Autumn 2021
Vitaly Ablavsky
What Is Tracking?
Tracking Implies Prediction

- Tracking a weather system: not just telling you it’s raining outside your house, but whether it will rain tomorrow (easy to guess if you’re in Seattle)

- Tracking an airplane: an estimate where it is currently and where it will be delta-T seconds from now

https://www.wunderground.com/forecast/us/wa/seattle/KWASEATT2713

https://en.wikipedia.org/wiki/Floatplane
Tracking Applications: Parameters

- Number of objects
- Types of objects
- Number of sensors
- Types of sensors
- Distance to sensor
Tracking Applications: Examples

1 object, articulated, 300 pixels tall

10 objects, articulated, 150 pixels tall

100 objects, 15 – 30 pixels tall
Factors that Make Tracking Hard

• (unknown) target dynamics: how fast does the state change?

• target observations/measurements: how noisy and (in)frequent?
Tracking in a Cluttered Environment With Probabilistic Data Association*

Dépistage dans une Ambiance Encombrée, avec Association Probabilistique des Données

Verfolgung in einer örtlich gestörten Umgebung unter Verwendung von Wahrscheinlichkeitsdaten

YAAKOV BAR-SHALOM† and EDISON TSE†

*Tracking a target with uncertainty in the origin of the measurements is accomplished with an algorithm, suitable for real-time implementation, which utilizes the a posteriori probabilities of the measurements having originated from the target.
Probabilistic Formulation
Probability Space

triplet \((\Omega, \mathcal{F}, P)\)

- sample space, all possible outcomes e.g., \{1, 2, 3, 4, 5, 6\}
- event space, e.g., set of all subsets of \(\Omega\) including “die lands even” \{2, 4, 6\}
- probability function
  \(P(A) \geq 0, A \in \mathcal{F}\)
  \(P(\Omega) = 1, P(\emptyset) = 0\)

credit: https://en.wikipedia.org/wiki/Probability_space
Random Variables

Random variable $X$: a function
$X: \Omega \rightarrow \mathbb{R}$
For every Borel subset $B$ of the real line $X^{-1}(B)$ in $\mathcal{F}$

Discrete
Continuous
Stochastic Processes

Stochastic process is an indexed collection of random variables

- **Discrete time**: $X = \{X_n, \ n = 0, 1, 2, \ldots\}$
- **Continuous time**: $X = \{X_t, \ 0 \leq t < \infty\}$, $X : T \times \Omega \to \mathbb{R}$

**Discrete time Markov process**

- $t \to X(t, \omega)$ called *sample path*

**Examples**:

1. $X_t = Z \cdot t^2$, $Z \sim N(0,1)$
2. Wiener process, a Gaussian process; limit of *random walk*

https://en.wikipedia.org/wiki/Markov_chain  
https://en.wikipedia.org/wiki/Wiener_process
Probabilistic Formulation

Multi-Object Configuration - the unknown (or hidden) state of the objects, i.e. position, size, etc.

Observations - information taken from the image such as color, motion, texture, etc.

Probabilistic Formulation

Graphical model for the multi-object tracking problem with $T$ time steps

Probabilistic Formulation

\[ p(x_t | z_{1:t}) = \frac{p(z_t | x_t) p(x_t | z_{1:t-1})}{p(z_t | z_{1:t-1})} \]

\[ p(x_t | z_{1:t-1}) = \int_{x_{t-1}} p(x_t | x_{t-1}) p(x_{t-1} | z_{1:t-1}) dx_{t-1} \]

\[ p(x_t | z_{1:t}) = \mathcal{C}^{-1} p(z_t | x_t) \times \int_{x_{t-1}} p(x_t | x_{t-1}) p(x_{t-1} | z_{1:t-1}) dx_{t-1} \]

Target Dynamics
Discrete Time LDS

- Continuous model are difficult to realize
  - Algorithms work at discrete time steps
  - Measurements are acquired with certain rates

- In practice, **discrete models** are employed

- **Discrete-time LDS** are governed by

\[ x(k + 1) = F(k)x(k) + G(k)u(k) + \xi(k) \]

- \( F \in \mathbb{R}^{nx \times nx} \) is the state transition matrix
- \( G \in \mathbb{R}^{nx \times nu} \) is the discrete-time input gain

- Same observation function of continuous models

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
Discrete Time LDS

- Continuous model are difficult to realize
  - Algorithms work at discrete time steps
  - Measurements are acquired with certain rates

- In practice, discrete models are employed

- Discrete-time LDS are governed by

\[ x(k + 1) = F(k)x(k) + G(k)u(k) + \xi(k) \]

- \( F \in \mathbb{R}^{nx \times nx} \) is the state transition matrix
- \( G \in \mathbb{R}^{nx \times nu} \) is the discrete-time input gain

- Same observation function of continuous models

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

- We want to throw a ball and **compute its trajectory**
- This can be **easily done with an LDS**
- The ball's **state** shall be represented as
  \[
  x = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}^T
  \]
- We ignore winds but consider the **gravity force** \( g \)
  \[
  u = -g
  \]
- No floor constraints
- We **observe** the ball with a noise-free position sensor
  \[
  z = \begin{bmatrix} x & y \end{bmatrix}^T
  \]

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

- Throwing a ball from \( s \) with initial velocity \( v \)
- Consider only the gravity force, \( g \), of the ball

- State vector
  \[ x = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}^T \]

- Initial state
  \[ x_0 = \begin{bmatrix} s_x & s_y & v_x & v_y \end{bmatrix}^T \]

- Input vector (scalar)
  \[ u = -g \]

- Measurement vector
  \[ z = \begin{bmatrix} x & y \end{bmatrix}^T \]

- Process matrices
  \[
  F = \begin{bmatrix}
  1 & 0 & T & 0 \\
  0 & 1 & 0 & T \\
  0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 1 
  \end{bmatrix}
  \]
  \[
  G = \begin{bmatrix}
  0 & \frac{T^2}{2} & 0 & T 
  \end{bmatrix}^T
  \]

- Measurement matrix
  \[
  H = \begin{bmatrix}
  1 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 
  \end{bmatrix}
  \]

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

- Initial State
  \[ x_0 = \begin{bmatrix} 0 & 0 & 9 & 30 \end{bmatrix}^T \]
- No noise
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

- **Initial State**
  \[ x_0 = \begin{bmatrix} 0 & 0 & 9 & 30 \end{bmatrix}^T \]

- It’s **windy** and our sensor is **imperfect**: let’s add Gaussian process and observation noise

\[ Q = \begin{bmatrix} 5.0 & 0 & 0 & 0 \\ 0 & 5.0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}, \quad R = \begin{bmatrix} 10.0 & 0 \\ 0 & 10.0 \end{bmatrix} \]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

System evolution       Observations

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
LDS Example – Throwing ball

[ G. Grisetti, C. Stachniss, K. Arras, and W. Burgard, Univ. of Freiburg Course on Robotics & Target Tracking ]
Kalman Filter

Prior knowledge of state → $P_{k-1|k-1}$, $\hat{x}_{k-1|k-1}$ → Prediction step Based on e.g. physical model

Next timestep $k \leftarrow k + 1$

$P_{k|k}$, $\hat{x}_{k|k}$ → Update step Compare prediction to measurements

Measurements $y_k$

Output estimate of state

source: https://en.wikipedia.org/wiki/Kalman_filter
Figure 1. Kalman filter as density propagation: in the case of Gaussian prior, process and observation densities, and assuming linear dynamics, the propagation process of Fig. 2 reduces to a diffusing Gaussian state density, represented completely by its evolving (multivariate) mean and variance—precisely what a Kalman filter computes.

Figure 2. Probability density propagation: propagation is depicted here as it occurs over a discrete time-step. There are three phases: drift due to the deterministic component of object dynamics; diffusion due to the random component; reactive reinforcement due to observations.

Figure 5. One time-step in the ConDensation algorithm: Each of the three steps—drift-diffuse-measure—of the probabilistic propagation process of Fig. 2 is represented by steps in the ConDensation algorithm.

CONDENSATION

Multiple-Hypothesis Tracking (MHT)


Multiple-Hypothesis Tracking (MHT)

Problem:
• data association (partitioning of observations)
• estimation of target tracks

Assumptions
• A1: one observation comes from one target or clutter
• A2: one target yields zero or one observation
• A3: one target yields zero or several observations

\[ Z^k = \{ Z(1), \ldots, Z(k) \} \]

frame 1
frame 2
frame 3

partition \( \omega \triangleq \{ t_0, \ldots, t_M \} \)
seek most probable partition

\[ Z^k \triangleq \text{all obs.} \quad Z(k) \triangleq \text{obs. current frame} \quad t_0 \triangleq \text{false alarms} \quad t_m \triangleq \text{track } m \]
Multiple-Hypothesis Tracking (MHT)

Key idea: hypothesis tree

\[
\Theta_k = \Theta_{k-1}^{\varphi(l)} \quad \Theta_{l}^{k} = \left\{ \Theta_{k-1}^{\varphi(l)}, \theta_{l}(k) \right\}
\]

MHT integrates

- Track initiation and termination
- Track update with (or without) observation
- Accounting for false alarms
- Enforcement of assumptions (A1, A2)

\[\theta_{l}(k) \triangleq \text{assignment } l \text{ at frame } k \quad \Theta_{\varphi(l)}^{k-1} \triangleq \text{arent hypothesis}\]
Hypothesis Probability

Given $N$ frames, enumerate finite set of hypotheses

Compute probability for each element of this set

$$P\left(\Theta^k_l \mid Z^k\right) \propto P\left\{ Z(k) \mid \theta_l(k), \Theta^{k-1}_{\varphi(l)}, Z^{k-1}\right\}$$

$P(\text{current obs.} \mid \text{current assignment})$

$$P\left\{ \theta_l(k) \mid \Theta^{k-1}_{\varphi(l)}, Z^{k-1}\right\}$$

$\times P(\text{current assignment} \mid \text{parent hypothesis})$

$$P\left\{ \Theta^{k-1}_{\varphi(l)} \mid Z^{k-1}\right\}$$

$\times P(\text{parent hypothesis} \mid \text{prior obs.})$
Probability of Observations given Assignment

- **track**
- **frame 1**
  - 1
  - 2
- **frame 2**
  - 1
  - 2
- **frame 3**
  - 1
  - 2

- $m_3 = 2$
- $\tau_1 = 1$
- $\tau_2 = 0$

- Observations from track $t_i$ are independent

- Normally distributed

- False alarms uniformly distributed in obs. area $V$

- $m_k \triangleq$ num obs. frame $k$

\[
p \{ Z(k) \mid \theta_l(k), \Theta^{k-1}_{\emptyset(l)}, Z^{k-1} \} = \\
\prod_{i=1}^{m_k} \left( \mathcal{N}_{t_i} \left( z_i(k) \right) \right)^{\tau_i} \left( \frac{1}{V} \right)^{1-\tau_i}
\]
Probability of Assignment (1)

Hypothesis 1

Hypothesis 2

similarity:
no assignment for $t_2$
2 false alarms

We can sum probabilities for these hypotheses!
\[ P\{\theta_i(k) | \Theta^{k-1}_{\varphi(l)}, Z^{k-1}\} = \]
\[ \frac{\phi! \nu!}{m_k!} \times \mu_F(\phi) \mu_N(\nu) \]

sum over similar hypotheses

\[ \times \prod_j (P_D)^{\delta_j} (1-P_D)^{1-\delta_j} (P_X)^{\chi_j} (1-P_X)^{1-\chi_j} \]

product over all existing tracks

priors on number of false alarms & new tracks

\[ \phi \triangleq \text{num. false alarms} \quad \nu \triangleq \text{num. new trks.} \quad \chi_i \triangleq 1 \text{ if trk.} \text{i} \text{ ends} \]

\[ m_k = 3 \quad \phi = 2 \]
\[ \nu = 0 \]
\[ \chi_1 = 0 \quad \chi_2 = 0 \]
Multi-Target Tracking as Bayesian Clustering
Bayesian Clustering

\[ \pi_3 \theta_3 \quad \pi_1 \theta_1 \quad \pi_2 \theta_2 \]

\[ \theta \sim H(\lambda) \]
\[ \pi \sim \text{GEM}(\alpha) \]

Goal: sample from \( p(\pi, \theta \mid x_1, \ldots, x_N, \alpha, \lambda) \)

\( \theta \triangleq \) cluster params. \( \pi \triangleq \) cluster weights \( \alpha, \lambda : \text{known, fixed} \)
Modeling Infinitely Many Clusters

How to model infinite mixture?

Via Dirichlet process mixture: \( p(x \mid \theta_1, \theta_2, \ldots) = \sum_{k=1}^{\infty} \pi_k f(x \mid \theta_k) \)

\[ \pi \sim \text{GEM}(\alpha) \]
\[ \beta_k \sim \text{Beta}(1, \alpha) \]
\[ \pi_k = \beta_k \prod_{\ell=1}^{k-1} (1 - \beta_\ell) \]
Key Densities

\[ p( \mathbf{z}_i | \mathbf{z}_{\setminus i}, \alpha ) = \]

\[ \frac{1}{\alpha + N - 1} \left( \alpha \delta(z_i, K+1) + \sum_{k=1}^{K} N_k^{-i} \delta(z_i, k) \right) \]

new cluster
num. obs. for cluster \( k \)

\[ p(x_i | z_i = k, \mathbf{z}_{\setminus i}, x_{\setminus i}, \lambda) = p(x_i | \{ x_j | z_j = k, j \neq i \}, \lambda) \]
obs. all assignments, other obs.
obs. \{ obs. in cluster \( k \) \}
obs.
Sampling the Posterior Distribution

\[ p \left( \text{assignment} \mid \text{all other assignments, obs.}, \alpha, \lambda \right) \propto \]
\[ p \left( \text{assignment} \mid \text{all other assignments, } \alpha \right) \times \]
\[ p \left( \text{obs.} \mid \text{all obs. assignments, } \lambda \right) \]

Algorithm: Rao-Blackwellized Gibbs Sampler

Given prior cluster assignments and cluster statistics

1. sample random permutation \( \{1, \ldots, N\} \)
   (a) For each obs. sample its cluster assignment
   (b) Update that cluster’s statistics
2. delete empty clusters
Data Association as Clustering

\[ x_k(t) = Ax_k(t-1) + Bu_k(t-1) \]
\[ y_k(t) = Cx_k(t) + w_k(t) \]

Cluster parameters

\[ \theta_k \triangleq [x_k(0), u_k(1), \ldots, u_k(T)] \]

\[ p(\text{obs} | \{\text{obs in cluster } k\}) = \]

\[ p(\text{obs} | \{\text{smoothed track } k\}) \]

\[ k \triangleq \text{track id} \]
\[ u \sim \mathcal{N}(0, \Lambda_u) \]
\[ w \sim \mathcal{N}(0, \Lambda_w) \]
\[ A, B, C : \text{fixed} \]
Assessment

true tracks

20,000 iterations

7th most frequent assignment
Pfinder
Pfinder: Real-Time Tracking of the Human Body

INTRODUCTION

Pfinder is a real-time system for tracking people and interpreting human behavior. The ability to find and follow people's actions in real-time has been of interest to visual scientists since the Gestalt psychologist Max Wertheimer 

Abstract

Pfinder adopts a Maximum A Posteriori Probability (MAP) approach to detection and tracking of the human body using simple 2D models. It incorporates a priori knowledge about people primarily to bootstrap itself and to recover from errors. The central problem for arbitrarily complex but single-person, fixed-camera situations is, in fact, a special case of recent Minimum Description Length (MDL) algorithms.

Fig. 1. (a) Video input (n.b. color image, shown here in grayscale). (b) Segmentation. (c) A 2D representation of the blob statistics.
we can reliably determine the location of the head, one hand, and the feet. These locations are then integrated into blob-model building process by using them as prior probabilities for blob creation and tracking. For instance, when the face and hand image positions are identified we can set up a strong prior probability for skin-colored blobs.

The following subsections describe the blob-model building process in greater detail.

4.1 Learning the Scene

Before the system attempts to locate people in a scene, it must learn the scene. To accomplish this Pfinder begins by acquiring a sequence of video frames that do not contain a person. Typically this sequence is relatively long, a second or more, in order to obtain a good estimate of the color covariance associated with each image pixel. For computational efficiency, color models are built in both the standard ($Y, U, V$) and brightness-normalized ($U^*, V^*$) color spaces.

4.2 Detect Person

After the scene has been modeled, Pfinder watches for large deviations from this model. New pixel values are compared to the known scene by measuring their Mahalanobis distance in color space from the class at the appropriate location in the scene model, as per (5).

If a changed region of the image is found that is of sufficient size to rule out unusual camera noise, then Pfinder proceeds to analyze the region in more detail, and begins to build up a blob model of the person.

4.3 Building the Person Model

To initialize blob models, Pfinder uses a 2D contour shape analysis that attempts to identify the head, hands, and feet locations. When this contour analysis does identify one of these locations, then a new blob is created and placed at that location. For hand and face locations, the blobs have strong flesh-colored color priors. Other blobs are initialized to cover clothing regions. The blobs introduced by the contour analysis compete with all the other blobs to describe the data.

When a blob can find no data to describe (as when a hand or foot is occluded), it is deleted from the person model. When the hand or foot later reappears, a new blob will be created by either the contour process (the normal case) or the color splitting process. This deletion/addition process makes Pfinder very robust to occlusions and dark shadows. When a hand reappears after being occluded or shadowed, normally only a few frames of video will go by before the person model is again accurate and complete.

4.3.1 Integrating Blobs and Contours

The blob models and the contour analyzer produce many of the same features (head, hands, feet), but with very different failure modes. The contour analysis can find the features in a single frame if they exist, but the results tend to be noisy. The class analysis produces accurate results, and can track the features where the contour can not, but it depends on the stability of the underlying models and the continuity of the underlying features (i.e., no occlusion).

The last stage of model building involves the reconciliation of these two modes. For each feature, Pfinder heuristically rates the validity of the signal from each mode. The signals are then blended with prior probabilities derived from these ratings. This allows the color trackers to track the hands in front of the body—

Fig. 3. (a) Chris Wren playing with Bruce Blumberg’s virtual dog in the ALIVE space. (b) Playing SURVIVE. (c) Real-time reading of American Sign Language (with Thad Starner doing the signing). (d) Trevor Darrell demonstrating vision-driven avatars.
Background Modeling
Challenges of Finding Moving Regions

A homogeneous disk moves to the right. Change is visible in the black regions only ($J_{t-1}$). The same thing happens one frame later ($J_t$). Only the intersection ($J_{t-1} \cap J_t$) is certain to be foreground in the middle image.

Pixel-Level Model

Wiener filter

\[ s_t = - \sum_{k=1}^{p} a_k s_{t-k} \]

\[ \mathbb{E}[e_t^2] = \mathbb{E}[s_t^2] + \sum_{k=1}^{p} a_k \mathbb{E}[s_t s_{t-k}] \]

Region Level

As each new pair of raw and foreground-marked images, \( I_r \) and \( F_r \), arrives,

1. Compute image differences (Figures 1a and b):

\[
J_r(x) = \begin{cases} 
1, & \text{if } |I_r(x) - I_{r-1}(x)| > k_{\text{motion}}, \\
0, & \text{otherwise}. 
\end{cases}
\]

2. Compute the subset of pixels which occur at the intersection of adjacent pairs of differenced images [1] and the previous foreground image (Figure 1c):

\[
K_r(x) = J_r(x) \land J_{r-1}(x) \land F_{r-1}(x).
\]

3. Find 4-connected regions, \( R_i \), in \( K_r \), discarding regions consisting of less than \( k_{\text{min}} \) pixels [2].

4. Compute \( H_i \), the normalized histogram of each \( R_i \), as projected onto the image \( I_{r-1}(s) \) (\( s \) is a pixel value):

\[
H_i(s) = \frac{\# \{ x : x \in R_i \text{ and } I_{r-1}(x) = s \}}{|R_i|}.
\]

5. Backproject histograms in \( I_r \): For each \( R_i \), compute \( F_{r-1} \land R_i \), and from each point in the intersection, grow \( L_i \), the 4-connected regions in the image,

\[
L_r(x) = \begin{cases} 
1, & \text{if } H_i(I_r(x)) > \varepsilon, \\
0, & \text{otherwise}. 
\end{cases}
\]

where we use \( k_{\text{motion}} = 16, k_{\text{min}} = 8, \varepsilon = 0.1 \).

Challenges of Finding Moving Regions

(a) A homogeneous disk moves to the right. Change is visible in the black regions only ($J_{t-1}$ in text).

(b) The same thing happens one frame later ($J_t$).

(c) Only the intersection ($J_{t-1} \cap J_t$) is certain to be foreground in the middle image.

Comparison to Prior Methods

<table>
<thead>
<tr>
<th>Test Image</th>
<th>Moving Object</th>
<th>Time of Day</th>
<th>Light Switch</th>
<th>Waving Trees</th>
<th>Camouflage</th>
<th>Backgrounding</th>
<th>Foreground Aperative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chair moved</td>
<td>Light gradually brightened</td>
<td>Light just switched on</td>
<td>Tree Waving</td>
<td>Foreground covers monitor pattern</td>
<td>No clean background training</td>
<td>Interior motion undetectable</td>
<td></td>
</tr>
<tr>
<td>Ideal Foreground</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adjacent Frame Difference</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean &amp; Threshold</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean &amp; Covariance [10]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mixture of Gaussians [3]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Normalized Block Correlation [7]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temporal Derivative [4]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bayesian Decision [8]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eigenbackground [9]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear Prediction [this paper]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wallflower [this paper]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$W^4$ System

$W^4$ System

Bayesian Methods for Multi-Target Tracking
BraMBLe

BraMBLe

Using Particles to Track Varying Numbers of Interacting People

K. Smith, D. Gatica-Perez, and J-M. Odobez, “Using Particles to Track Varying Numbers of Interacting People,” CVPR 2005
K. Smith, D. Gatica-Perez, and J-M. Odobez, “Using Particles to Track Varying Numbers of Interacting People,” CVPR 2005
Global Binary Observation Model

Wrong:
\[ p(Z_t | X_t) = \prod_{k=1}^{K} p(Z_t | X_{k,t}) \]

Reasonable:
\[ p(Z_t | X_t) \Delta= p(Z_t^F | X_t) p(Z_t^B | X_t) \]

K. Smith, D. Gatica-Perez, and J-M. Odobez, “Using Particles to Track Varying Numbers of Interacting People,” CVPR 2005
Approximation via RJ MCMC

\[ p(X_t | Z_{1:t}) \]

Samples within Markov chain for a given frame

frame \( t - 1 \)

\[ X_{t-1}^{(1)} \rightarrow X_{t-1}^{(2)} \rightarrow X_{t-1}^{(n)} \rightarrow X_{t-1}^{(N)} \]

frame \( t \)

\[ X_t^{(1)} \rightarrow X_t^{(2)} \rightarrow X_t^{(n)} \rightarrow X_t^{(N)} \]

RJ MCMC moves: update birth swap death

K. Smith, D. Gatica-Perez, and J-M. Odobez, “Using Particles to Track Varying Numbers of Interacting People,” CVPR 2005
Assessment

ranked 2nd compared with KLT, Active Shapes, face detector

K. Smith, D. Gatica-Perez, and J-M. Odobez, “Using Particles to Track Varying Numbers of Interacting People,” CVPR 2005
RJ MCMC Model Selection (1)

\[ \{1, \ldots, M\} \times \prod_{m=1}^{M} \mathcal{X}_m \quad \text{vs.} \quad \bigcup_{m=1}^{M} \{m\} \times \mathcal{X}_m \]

product space \quad RJ MCMC

\[ p(m, x_m \mid Z), \ x_m \in \mathcal{X}_m \]

\[ \mathcal{M}_m, \ m = 1, \ldots, M \]

RJ MCMC Model Selection (II)

$$p_m(x_m | Z)$$

$$\mathcal{X}_{n,m} \triangleq \mathcal{X}_n \times \mathcal{U}_{n,m}$$

$$\mathcal{X}_{m,n} \triangleq \mathcal{X}_m \times \mathcal{U}_{m,n}$$

$$f_{n\rightarrow m} \left( f_{m\rightarrow n} \left( x_m, u_{m,n} \right) \right) = (x_m, u_{m,n})$$

$$u_{n,m} \sim q_{n\rightarrow m} \left( \cdot | n, x_n \right)$$

$$u_{m,n} \sim q_{m\rightarrow n} \left( \cdot | m, x_m \right)$$

$$\mathcal{A}_{n\rightarrow m} = \min \left\{ 1, \frac{p(m, x_m^*)}{p(n, x_n)} \times \frac{q(n | m)}{q(m | n)} \times \frac{q_{m\rightarrow n}(u_{m,n} | m, x_m^*)}{q_{n\rightarrow m}(u_{n,m} | n, x_n)} \times \left| J_{f_{n\rightarrow m}} \right| \right\}$$

$$u_{m,n} \triangleq \text{auxiliary var. with proposal density } q_{m\rightarrow n}(\cdot | \cdot), J_f \triangleq \text{det. of Jacobian}$$
RJ MCMC Example

RBF regression

how many kernels?

RJ MCMC moves: merge split birth death

$$\mu = \frac{\mu_1 + \mu_2}{2}$$

$$q(k \mid k+1) = \frac{1}{k+1}$$

$$\mu_1 = \mu - \beta u_{n,m} \quad \mu_2 = \mu + \beta u_{n,m} \quad u_{n,m} \sim \mathcal{U}_{[0,1]}$$

$$q(k+1 \mid k) = \frac{1}{k}$$

$$J_{\text{split}} = \begin{bmatrix} 1 & 1 \\ -\beta & \beta \end{bmatrix}$$