Behavioural Pedestrian Tracking

Javier Cruz, Michel Bierlaire and Jean-Philippe Thiran
{javier.cruz, michel.bierlaire, jean-philippe.thiran}@epfl.ch

École Polytechnique Fédérale de Lausanne
Motivation

[Images of pedestrian tracking software interfaces, showing input and output sequences with highlighted areas for tracking.]
Motivation
Motivation
Outline

- Introduction
- Visual tracking
- Pedestrian Visual Tracking and Detection
- Questions and future work
Introduction

Common pedestrian tracking systems: Detection and Inter-Frame Tracking

Detection:
- Haar/HOG feature + Boosting
- Background substraction
- Model based detection (skeleton models, silhouettes, etc.)

Tracking:
- Kalman filter
- Condensation algorithm
- Mean-shift
- Covariance tracking
Introduction

Usual approach:

DETECTION → TRACKING
Introduction

Usual approach:

DETECTION \rightarrow TRACKING

What about doing a tracking that may end in a detection?

TRACKING \rightarrow DETECTION
Visual Tracking


- On-Line low-dimensional subspace representation (incremental PCA)
- Gaussian variables
- Particle filtering
Visual Tracking

Definitions:

- $I_t$: frame $t$
- $X_t = (x_t, y_t)$: position
- $\varphi_t$: rotation
- $s_t$: scale
- $r_t = \frac{w_t}{h_t}$: aspect ratio
- $W_t = f(I_t, X_t, \varphi_t, s_t, r_t)$: a patch in $I_t$
- $\Psi_t = \{X_t, \theta_t\} = \{X_t, \varphi_t, s_t, r_t\} \in \Theta_t$
Visual Tracking

Everything is supposed to be Gaussian:

- \( X_t \sim \mathcal{N}(X_{t-1}, \sigma_X) \)
- \( \varphi_t \sim \mathcal{N}(\varphi_{t-1}, \sigma_\varphi) \)
- \( s_t \sim \mathcal{N}(s_{t-1}, \sigma_s) \)
- \( r_t \sim \mathcal{N}(r_{t-1}, \sigma_r) \)

\[ \Psi_t = \{X_t, \varphi_t, s_t, r_t\} \in \Theta_t \]
Visual Tracking

\[
\Psi^*_t = \arg \max_{\Psi_t \in \Theta_t} p(I_t | \Psi_t) p(\Psi_t | \Psi_{t-1})
\]

Observation:

\[
p(I_t | \Psi_t) = \mathcal{N}(W_t; \mu, UU^\top + \varepsilon I) \mathcal{N}(W_t; \mu, U\Sigma_o^{-2}U^\top)
\]

Dynamics:

\[
p(\Psi_t | \Psi_{t-1}) = \mathcal{N}(\Psi_t; \Psi_{t-1}, \Sigma_{\Psi})
\]
Visual Tracking


Takes into account:

- Direction
- Destination
- Speed
- “Leader-follower”
- “Collision avoidance”
Visual Tracking

This forces some assumptions:

- Camera is calibrated
- Camera is fixed
- Pedestrians walking in normal conditions
- Destination known!!
Visual Tracking

Pedestrian model + Gaussian:

- $X_t \sim \text{pedestrian walking behaviour model (PWBM)}$
- $\varphi_t \sim \mathcal{N}(\varphi_{t-1}, \sigma_{\varphi})$
- $s_t \sim \mathcal{N}(s_{t-1}, \sigma_s)$
- $r_t \sim \mathcal{N}(r_{t-1}, \sigma_r)$

$$\Psi_t = \{X_t, \theta_t\} = \{X_t, \varphi_t, s_t, r_t\} \in \Theta_t$$
Visual Tracking

\[ \Psi_t^* = \arg \max_{\Psi_t \in \Theta_t} p(I_t | \Psi_t) p(\Psi_t | \Psi_{t-1}) \]

Observation:

\[ p(I_t | \Psi_t) = \mathcal{N}(W_t; \mu, U U^\top + \varepsilon I) \mathcal{N}(W_t; \mu, U \Sigma_o^{-2} U^\top) \]

Dynamics:

\[ p(\Psi_t | \Psi_{t-1}) = \text{PWBM}(X_t; X_{t-1}) \mathcal{N}(\theta_t; \theta_{t-1}, \Sigma_{\theta}) \]
Considered models:

- $M_G$ a Gaussian model
- $M_P$ the Pedestrian model

Bayesian Model Averaging over $\mathcal{M} = \{M_G, M_P\}$
Visual Tracking

Modified probabilities of the Visual Tracking Algorithm:

\[
p(\Psi_t | \Psi_{t-1}, I_t) = \sum_{M_j \in \mathcal{M}} p(\Psi_t | M_j, \Psi_{t-1}, I_t) p(M_j | \Psi_{t-1}, I_t)
\]

\[
p(M_i | \Psi_{t-1}, I_t) = \frac{p(\Psi_{t-1}, I_t | M_i) p(M_i)}{\sum_{M_j \in \mathcal{M}} p(\Psi_{t-1}, I_t | M_j) p(M_j)}
\]

\[
p(\Psi_{t-1}, I_t | M_i) = \int_{\Theta_t} p(\Psi_{t-1}, I_t | \Psi_t, M_i) p(\Psi_t | M_i) d\Psi_t
\]
Visual Tracking

Interpretation of the ratio of posterior model probabilities (The Occam’s Window):

\[ \frac{p(M_p|Data)}{p(M_d|Data)} \]

Inconclusive evidence

Strong evidence for \( M_G \)  
Strong evidence for \( M_P \)

Common values are \( O_R = 20 \) and \( O_L = O_R^{-1} \)
Proposed algorithm:

1: Initialization: set priors, $\mathcal{M} = \{M_P, M_G\}$
2: Foreground detection in $I_0$
3: for $t = 1$ to $N$ do
4:   for each obtained or tracked blob do
5:     Apply “Visual Tracking Algorithm” with modified probabilities
6:     Classify blob according to model posteriors
7:     if $M_P$ is chosen then
8:       $\mathcal{M} = \{M_P\}$
9:     end if
10:   end for
11: end for
12: Foreground detection in $I_t$
Questions and Future work

- What kind of priors?
- “Occam’s window” per frame or with memory?
- Particle filters, Gibbs sampling, Metropolis-Hastings algorithms?
- Implement and test
- Generalization to $N$ models
- Action detection
- Other “objects”