Object Recognition

Chapter 9: Video Tracking

Prof. Dr. Johannes Maucher

HdM CSM

Version 1.0
22. May 2013

Document History

Version Nr.	**Date**	**Changes**
1.0		Initial Version

Chapter 9: Tracking

1. Introduction
   - Goal
2. Simple Tracking Strategies
   - Tracking by Detection
   - Tracking by Matching
3. Tracking by Background Subtraction
   - The simple case
   - Gaussian Mixture Models based Background Subtraction
4. Kalman Filter
   - State Estimation in General
   - Kalman Filter
5. References

Goal

Goal of tracking

Build space time paths, which are followed by tokens. Tokens can be interest points, regions, image windows or objects.

- More generally one likes to describe the state of a token over time
- State contains everything of the token, what is of interest in the given application, e.g. position, velocity, direction, appearance,...
- Some state variables are observed, others can be modelled.

Applications

- Recognition from Motion: Some objects may have a characteristic motion pattern.
- Surveillance
- Targeting: Track objects and predict their future positions in order to hit them.
- Motion Capturing
- Controlling, e.g. Kinect

Simple Tracking Strategies

Tracking by detection

- If a strong model of the object is available (e.g. face)
- Detect the model in each frame and link the detected positions to a track

Tracking by matching

- Requires a model of how the object moves
- Given a domain (position of the object) in the *nth* frame, apply the movement model to search for the domanin in the $n+1$ *th* frame.

Tracking by Detection

Easy if:

- the object can easily be detected, e.g.
  - Red ball, before green background
  - frontal faces
- there exists only one object in the video

In this case it is enough to report the location of the detector response in each frame

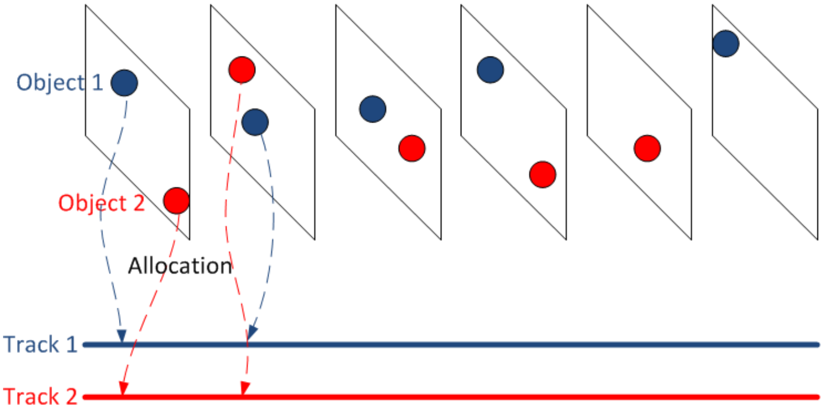
More challenging if:

- More than one object in the video
- Objects can enter or leave a frame

In this case a track is kept for each object in the video. A track represents a timeline.

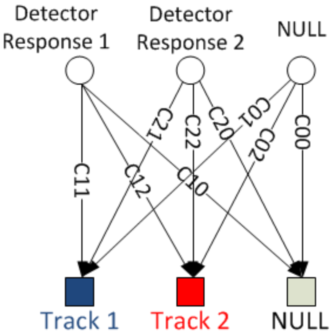
- How to allocate detector responses to tracks?
- What if some detector responses can not be allocated (new objects)?
- What if some tracks do not receive a detector response (vanished objects)?

Tracking by Detection

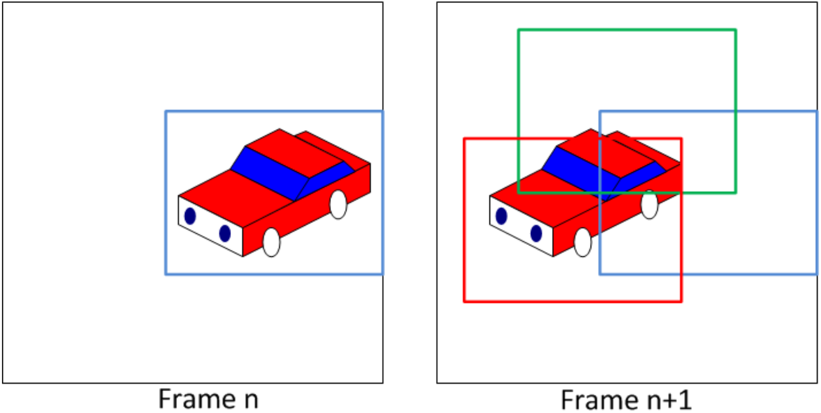


Allocation of Detector Responses to Tracks

- Allocation of detector responses to tracks is realized by applying a cost function.
- Allocate each detector response *i* to the track *j* to which the costs $c_{i,j}$ are minimal.
- Possible cost functions:
  - Distance between location of detector response *i* to the last location of the object belonging to track *j* (suitable for slow-moving objects)
  - Distance between the descriptor of detector response *i* and the last descriptor of the object belonging to track *j*. E.g. the descriptors can be color-histograms and the distance between is e.g. χ^2 distance.



Tracking by Matching



Tracking by Matching

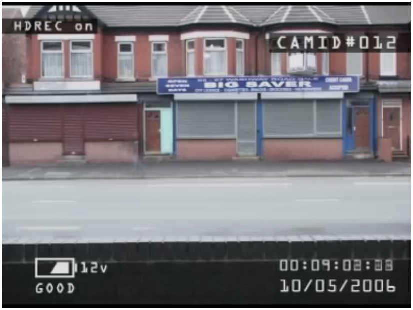
- For the rectangle $R^{(n)}$ in frame n find the rectangle $R^{(n+1)}$ in frame $n+1$ that best matches $R^{(n)}$.
- Let P_n be the set of all pixel positions (i,j) that belong to rectangle $R^{(n)}$.
- Let a and b be the vertical and horizontal offsets between $R^{(n)}$ and $R^{(n+1)}$
- Then the task is to find the offset (a,b) such that the Sum of Squared Error (SSE)

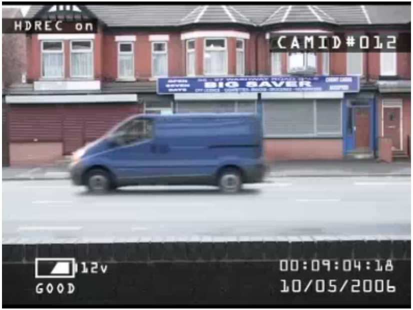
$$\sum_{(i,j)\in P^n} \left(R_{i,j}^{(n)} - R_{i+a,j+b}^{(n+1)} \right)^2 \tag{1}$$

  is minimized.
- If a maximum speed for moving objects can be assumed, the search region in frame $n+1$ can be restricted.
- Instead of applying pixel intensities other features, e.g. color histograms, can be applied. Then instead of minimizing SSE other similarity metrics (see metrics for comparing histograms) are applied.

Simple Background Subtraction

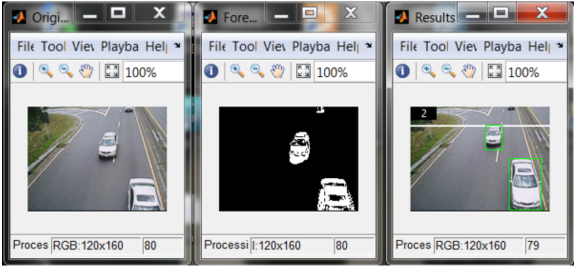
- Belongs to category tracking by detection
- In the case of a constant background (uni-modal background), just determine average background.
- Subtraction of background from current frame yields some blobs, which are the detected objects.





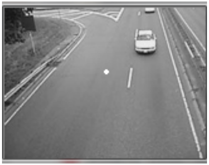
Detection by Background Subtraction

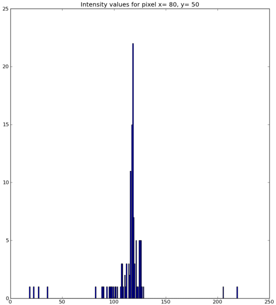
- Modelling background by Gaussian Mixture Models (GMM)[1] according to [Stauffer and Grimson, 1999].
- Intensities or colors of each pixel over entire time are modelled as GMM.
- Implementation e.g. in car tracking demo of Matlab Computer Vision Toolbox
  `http://www.mathworks.de/de/help/vision/gs/object-detection-and-tracking.html#btd13lq`

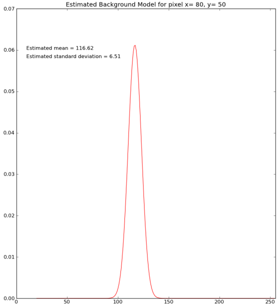


[1] See e.g. Machine Learning Lecture J. Maucher

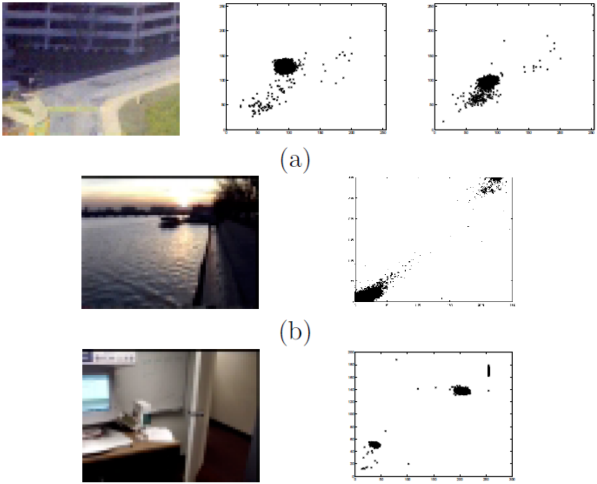
GMM Background Subtraction







Multimodal Backgroundmodels are required



(a)

(b)

(c)

Multimodal Backgroundmodels are required

Three examples in previous slide depict scatter plots of red- and green-values of a single pixel :

- a.) Due to illumination change within only short time distribution of pixel values change
- b.) Again two different distributions for single pixel due to specularities
- c.) Again two different distributions due to monitor flicker

⇒ Background model of single pixel shall be a mixture of Gaussians (GMM for Gaussian Mixture Model)

Recap: Multivariate Gaussian Normaldistribution

- Random Variable X containing d components X_i:

$$X = [X_1, X_2, \ldots X_d]$$

- d-dimensional random variable X is Gaussian, if it's probability density function is:

$$p(\mathbf{x}, \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right], \quad -\infty < x < \infty \tag{2}$$

- with:
  - vector of mean values

$$\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_d]$$

  - covariance matrix

$$\Sigma = \begin{pmatrix} \sigma_{11}^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_{22}^2 & \cdots & \sigma_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_{dd}^2 \end{pmatrix}$$

GMM background model

- Let

  $$\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t\}$$

  be the history of values of a particular pixel in frames 1 to t
- In the case of a RGB video the vectors $\mathbf{x}_i$ are 3-dimensional, in the case of a greyscale video they are 1-dimensional.
- The recent history $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t\}$ of each pixel is modelled as a mixture of K Gaussians. The probability of observing the current pixel $\mathbf{x}_t$ is then:

$$P(\mathbf{x}_t) = \sum_{i=1}^{K} \omega_{i,t} p(\mathbf{x}_t, \boldsymbol{\mu}_{i,t}, \Sigma_{i,t}) \tag{3}$$

- $\omega_{i,t}$ is an estimate for the probability that at time t the pixel is in mode i.
- In [Stauffer and Grimson, 1999] it is assumed that the red, green and blue pixel values are independent and have the same variance. Then the covariance matrix is:

$$\Sigma_{i,t} = \sigma_i^2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Matching Pixel Value to Distribution

- For each pixel a mixture of K dynamically adapted Gaussian distributions exists (typical values of K in the range 3 to 7).
- Every new pixel value $\mathbf{x}_t$ is checked against the existing K distributions
- A pixel value matches distribution i, if

$$\mathbf{x}_t \in \left[\boldsymbol{\mu}_{i,t} - 2.5\sigma_{i,t}, \ldots, \boldsymbol{\mu}_{i,t} + 2.5\sigma_{i,t})\right]$$

  i.e. if it lies within 2.5 times the standard distribution around the mean of the distribution
- If a pixel value matches more than one distribution the best matching distribution is selected, which is the one with the maximum $\omega_{i,t}/\sigma_{i,t}$
- If none of the K distributions matches the current pixel value, the least probable distribution is replaced by a new distribution, whose
  - mean $\boldsymbol{\mu}_{i,t}$ is the current pixel value
  - variance $\sigma_{i,t}$ is selected to be high
  - prior weight $\omega_{i,t}$ is low.

Updating Parameters

- Update of prior weights at time *t*

$$\omega_{i,t} = (1-\alpha)\omega_{i,t-1} + \alpha M_{i,t}, \tag{4}$$

  where α is a learning rate which typically decreases with t and $M_{i,t}$ is 1 for the currently matching model and 0 for all other modells.
- After each weight update they are renormalized, such that the sum over all weights of a GMM is 1.
- The $\boldsymbol{\mu}$ and σ parameters of unmatched distributions are not updated.
- For the matching distribution i, the update is:

$$\begin{aligned}
\boldsymbol{\mu}_{i,t} &= (1-\rho_{i,t})\boldsymbol{\mu}_{i,t-1} + \rho_{i,t}\mathbf{x}_t && (5)\\
\sigma^2_{i,t} &= (1-\rho_{i,t})\sigma^2_{i,t-1} + \rho_{i,t}(\mathbf{x}_t - \boldsymbol{\mu}_{i,t})^T(\mathbf{x}_t - \boldsymbol{\mu}_{i,t}) && (6)\\
\rho_{i,t} &= \alpha p(\mathbf{x}_t, \boldsymbol{\mu}_{i,t}, \Sigma_{i,t}) && (7)
\end{aligned}$$

Which of the K Distributions model Background?

- Some of the K Gaussian Mixtures describe background, some foreground modes of the pixel.
- Assumption: Background Models have
  - higher prior weights $\omega_{i,t}$
  - lower standard deviation $\sigma_{i,t}$
- Determination of background distributions:
  - Order the set of K distributions according to decreasing $\omega_{i,t}/\sigma_{i,t}$
  - Select the first B distributions to be background modes, with

$$B = argmin_b \left(\sum_{i=1}^{b} \omega_{i,t} > T \right)$$

  - Low threshold T yields unimodal, large T yields multimodal background models.

Kalman Filter: Integrating dynamics- and measurement models

- Tracking is based on the measurement of the current position of the relevant object.
- This measurement is erroneous
- On the other hand often the dynamics of the moving object are partly known.
- If the dynamics of the moving object and the erroneous measurement process can be modelled, then Kalman Filter provide a much more accurate position estimate than the measurement alone.

State and Measurements

- $\mathbf{x}_t$: State at discrete time t e.g.

$$\mathbf{x}_t = \begin{pmatrix} x_t \\ y_t \\ v_{x,t} \\ v_{y,t} \end{pmatrix}$$

- $\mathbf{y}_t$: Measurement at discrete time t e.g.

$$\mathbf{y}_t = \begin{pmatrix} x'_t \\ y'_t \end{pmatrix}$$

- In general not all state variables are measured
- Measurements are noisy
- Thus the true state is hidden
- $\mathbf{Y}_{1:t} = \mathbf{y}_0, \mathbf{y}_1, \ldots \mathbf{y}_t$ denotes the sequence of all measurements up to time t.
- $\mathbf{X}_{1:t} = \mathbf{x}_0, \mathbf{x}_1, \ldots \mathbf{x}_t$ denotes the sequence of all states up to time t.

Problem to Solve

Task

Estimate the current state $\mathbf{x}_t$ if all measurements $\mathbf{Y}_{1:t}$ and all previous states $\mathbf{X}_{1:t-1}$ are given.

Assumptions:

- Measurements depend only on the current hidden state:

$$P(\mathbf{y}_k|\mathbf{Y}_{1:N}, \mathbf{X}_{1:N}) = P(\mathbf{y}_k|\mathbf{x}_k)$$

- Probability density of current hidden state is a function only of the previous state:

$$P(\mathbf{x}_k|\mathbf{X}_{1:k-1}) = P(\mathbf{x}_k|\mathbf{x}_{k-1})$$

Belief about current state before and after measurement

- Prediction: Calculate $\overline{\mathbf{x}}_t^-$, which is the belief about $\mathbf{x}_t$ immediately before measurement $\mathbf{y}_t$. This belief is determined from the predictive density

$$P(\mathbf{x}_t|\mathbf{Y}_{1:t-1}) = \int P(\mathbf{x}_t|\mathbf{x}_{t-1})P(\mathbf{x}_{t-1}|\mathbf{Y}_{1:t-1})d\mathbf{x}_{t-1} \qquad (8)$$

- Correction: Calculate $\overline{\mathbf{x}}_t^+$, which is the belief about $\mathbf{x}_t$ immediately after measurement $\mathbf{y}_t$. This belief is determined from

$$P(\mathbf{x}_t|\mathbf{Y}_{1:t}) = \frac{P(\mathbf{y}_t|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{Y}_{1:t-1})}{\int P(\mathbf{y}_t|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{Y}_{1:t-1})d\mathbf{x}_{t-1}} \qquad (9)$$

Process Model and Measurement Model

In order to calculate equations (8) and (9) one requires

- Process Model

$$P(\mathbf{x}_t|\mathbf{x}_{t-1})$$

- Measurement Model

$$P(\mathbf{y}_t|\mathbf{x}_t)$$

- In general the probability distributions in equations (8) and (9) are difficult to calculate and represent.
- However, if the Process Model and the Measurement Model are both linear, all probability distributions are Gaussian.
- In this case of a linear Gaussian system, state estimation (tracking in our case) reduces to updating the mean and variance of the distributions, as defined by the Kalman Filter.

Linear Gaussian System

- General Notation:

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$$

  indicates that **x** is a Gaussian distributed multivariate random variable, with mean $\boldsymbol{\mu}$ and covariance matrix Σ.

- Linear Process Model

$$\mathbf{x}_t = D_t \mathbf{x}_{t-1} + \epsilon_t \quad \epsilon_t \sim \mathcal{N}(\mathbf{0}, \Sigma_{d,t}) \tag{10}$$

$$P(\mathbf{x}_t | \mathbf{x}_{t-1}) \sim \mathcal{N}(D_t \mathbf{x}_{t-1}, \Sigma_{d,t}) \tag{11}$$

- Linear Measurement Model

$$\mathbf{y}_t = M_t \mathbf{x}_t, + \delta_t \quad \delta_t \sim \mathcal{N}(\mathbf{0}, \Sigma_{m,t}) \tag{12}$$

$$P(\mathbf{y}_t | \mathbf{x}_t) \sim \mathcal{N}(M_t \mathbf{x}_t, \Sigma_{m,t}) \tag{13}$$

Example Linear Process and Measurement Models

Constant Velocity

- Let **p** be the position and **v** be the velocity of a point moving with constant velocity: $\mathbf{p}_t = \mathbf{p}_{t-1} + \Delta t \cdot \mathbf{v}_{t-1}$ and $\mathbf{v}_t = \mathbf{v}_{t-1}$.
- The state is modelled as

$$\mathbf{x} = \left(\begin{array}{c} \mathbf{p} \\ \mathbf{v} \end{array} \right)$$

- Matrix D_t of the linear process model is then

$$D_t = \left(\begin{array}{cc} I & \Delta t \cdot I \\ 0 & I \end{array} \right)$$

  where I is the Identity matrix and Δt is the time between two successive discrete steps.
- If only the position is measured, then matrix M_t of the linear measurement model is

$$M_t = \left(\begin{array}{cc} I & 0 \end{array} \right)$$

Example Linear Process and Measurement Models

Constant Acceleration

- The state is modelled as

$$\mathbf{x} = \begin{pmatrix} \mathbf{p} \\ \mathbf{v} \\ \mathbf{a} \end{pmatrix}$$

- Postion Update: $\mathbf{p}_t = \mathbf{p}_{t-1} + \Delta t \cdot \mathbf{v}_{t-1}$
- Velocity Update: $\mathbf{v}_t = \mathbf{v}_{t-1} + \Delta t \cdot \mathbf{a}_{t-1}$
- Acceleration Update: $\mathbf{a}_t = \mathbf{a}_{t-1}$
- Matrix D_t of the linear process model is then

$$D_t = \begin{pmatrix} I & \Delta t \cdot I & 0 \\ 0 & I & \Delta t \cdot I \\ 0 & 0 & I \end{pmatrix}$$

- If only the position is measured, then matrix M_t of the linear measurement model is

$$M_t = \begin{pmatrix} I & 0 & 0 \end{pmatrix}$$

Example Linear Process and Measurement Models

- Matrices D_t and M_t may vary from frame to frame
- Covariances $\Sigma_{d,t}$ and $\Sigma_{m,t}$ may vary from frame to frame.
- Example
  - 3D position is modelled, i.e.

$$\mathbf{p} = \left(\begin{array}{c} x \\ y \\ z \end{array} \right)$$

  - State is defined only by position $\mathbf{x} = \mathbf{p}$
  - At each time instance only one of the 3 coordinates is measured
  - Then the time varying measurement matrices may be:

$$M_{3t} = (1\ 0\ 0) \quad M_{3t+1} = (0\ 1\ 0) \quad M_{3t+2} = (0\ 0\ 1)$$

Kalman Filter Notation and Concept

Notation:

- In the case of linear process and measurement models the probability distributions $P(\mathbf{x}_t|\mathbf{Y}_{1:t-1})$ (equation (8)) and $P(\mathbf{x}_t|\mathbf{Y}_{1:t})$ (equation (9)) are Gaussian.
- $\overline{\mathbf{x}}_t^-$, the belief about $\mathbf{x}_t$ immediately before measurement $\mathbf{y}_t$ is then the mean of $P(\mathbf{x}_t|\mathbf{Y}_{1:t-1})$. The covariance of this distribution is denoted by Σ_t^-.
- $\overline{\mathbf{x}}_t^+$, the belief about $\mathbf{x}_t$ immediately after measurement $\mathbf{y}_t$ is then the mean of $P(\mathbf{x}_t|\mathbf{Y}_{1:t})$. The covariance of this distribution is denoted by Σ_t^+.

Concept of Kalman Filtering:

- For each time t:
  1. Prediciton: Calculate $\overline{\mathbf{x}}_t^-$ from previous knowledge
  2. Perform measurement of $\mathbf{y}_t$
  3. Correction: Integrate the measured value $\mathbf{y}_t$ in order to correct the state estimate. The corrected state is $\overline{\mathbf{x}}_t^+$.

Kalman Filter

Kalman Filter

Prerequisities:

- Process Model: $P(\mathbf{x}_t|\mathbf{x}_{t-1}) \sim \mathcal{N}(D_t\mathbf{x}_{t-1}, \Sigma_{d,t})$
- Measurement Model: $P(\mathbf{y}_t|\mathbf{x}_t) \sim \mathcal{N}(M_t\mathbf{x}_t, \Sigma_{m,t})$
- Estimates for $\mathbf{x}_0^+$ and Σ_0^+

For all $t \in [1, \ldots, N]$:

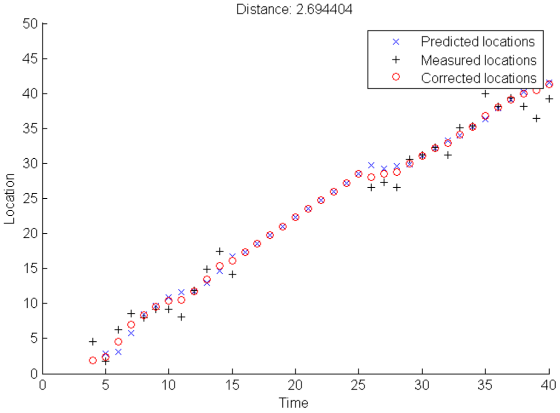
1. Prediction:

$$\begin{aligned} \overline{\mathbf{x}}_t^- &= D_t\overline{\mathbf{x}}_{t-1}^+ \\ \Sigma_t^- &= \Sigma_{d,t} + D_t\Sigma_{t-1}^+ D_t^T \end{aligned}$$

2. Correction:

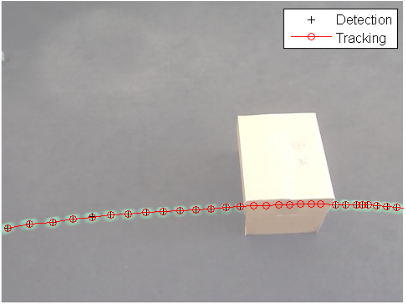
$$\begin{aligned} K_t &= \Sigma_t^- M_t^T \left(M_t\Sigma_t^- M_t^T + \Sigma_{m,t} \right)^{-1} \\ \overline{\mathbf{x}}_t^+ &= \overline{\mathbf{x}}_t^- + K_t \left(\mathbf{y}_t - M_t\overline{\mathbf{x}}_t^- \right) \\ \Sigma_t^+ &= (I - K_tM_t)\,\Sigma_t^- \end{aligned}$$

Example: Tracking of 1-dimensional movement with constant velocity and missing measurements



Matlab Kalman Filter Tracking Demo

- Location of the ball is not perfectly measured.
- Location can not be measured at all as long as ball is under desk.



http://www.mathworks.de/de/help/vision/examples/using-kalman-filter-for-object-tracking.html

Further Remarks

- Field of Kalman Filter applications is extremely wide.
- Fast and often applied for real time tracking
- Many extensions, e.g. toward non-linear models exist. See e.g. [Bradski and Kaehler, 2008], [Forsyth and Ponce, 2003].

References I

Bradski, G. and Kaehler, A. (2008).
Learning OpenCV: Computer Vision with the OpenCV Library.
O'Reilly.

Forsyth, D. A. and Ponce, J. (2003).
Computer Vision: A Modern Approach.
Prentice Hall.

Stauffer, C. and Grimson, W. E. L. (1999).
Adaptive background mixture models for real-time tracking.
In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages –252 Vol. 2.