2D Image Processing
Introduction to Bayesian Tracking

Prof. Didier Stricker
Kaiserlautern University

http://ags.cs.uni-kl.de/

DFKI – Deutsches Forschungszentrum für Künstliche Intelligenz

http://av.dfki.de

Slides from G. Bleser based slides from G. Panin, S. Thrun, K. Smith, ..
Examples of tracking results
Tracking applications

- Biological Research
  - Goal: develop a method to “trap” Salmonella bacteria

P. Horvath, Q. Buhkari, 3D Tracking of point-like objects in 2D image sequences, LMC, ETHZ
Video tracking is the process of locating a moving object (or multiple objects) over time using a camera. The objective of video tracking is to associate target objects in consecutive video frames.

Models of the objects are supposed to be available in advance, from which the relevant image features can be automatically selected.
What is visual tracking?

- Definition: using image measurements and a predictive dynamic model to consistently estimate the state(s) $X_t$ of one or more object(s) over the discrete time steps corresponding to video frames.

Here: object position
What is visual tracking?

- Why not just do detection?
  - Inefficient
  - Data association problem
What is visual tracking?

- Why not just do detection?
  - 😞 Inefficient
  - 😞 Data association problem
Why not just do detection?

Realtime Perception for Catching a Flying Ball with a Mobile Humanoid
Oliver Birbach, Udo Frese, Berthold Bäuml

DFKI, Safe and Secure Cognitive Systems

Rollin Justin catches flying balls

Courtesy of U. Frese
Why not just do detection? (example: human-robot-interaction)

- Complex tracking problem:
  - Distinguish flying balls from other objects that look like balls
  - Predict ball trajectories for catching! Early enough to move robot hand to right position

- Approach:
  - Physical model for ball dynamics (constant acceleration due to gravity)
  - Multiple hypothesis tracking
Summary: What we want to achieve

- Include model assumptions, e.g. motion/environment models
- Overcome temporary tracking loss
- Predict future
- Handle multiple targets
- Handle multiple hypotheses
- Provide confidence values for estimates
- Fuse different sensors (same or different types) and modalities
How can we do all the things indicated before?

**Probabilistic estimation framework:** Bayesian filtering, recursive state estimation

Pre-requisites: probability theory, linear algebra
Introductory example: How Bayesian filtering works

- Mobile robot localization in a known environment based on motion (odometry) and sensing (door detection)
- The robot knows a map of the environment (corridor with 3 indistinguishable doors)

- Wanted: position of the robot in the map

Introductory example

Measurement $z_1$: here is a door.
Introductory example

Odometry input $u_1$: 1m forward.

Measurement $z_2$: here is a door.
Introductory example

Odometry input $u_2$: 1.5m forward.

- **Bayes filter algorithm:**
  - Recursively estimate posterior belief or state distribution based on control data and sensor measurements
  - Information represented as probability density function
  - All entities are random variables
Recursive Bayesian filtering

- **Key idea 1**: Probability distributions represent our belief about the state of a dynamical system

- **Key idea 2**: Recursive cycle
  1. Predict from motion model
  2. Sensor measurement
  3. Correct the prediction
     ...repeat
Outline for next steps

- Basic concepts in probability
- Terminology, notation, probabilistic laws
- Bayes filters

→ Basis for the next three lectures
Basic concepts in probability
Discrete random variables

- denotes a random variable.
- can take on a countable number of values in according to probabilistic laws.
- , or , is the probability that the random variable takes on value .
- is called probability mass function.

Example:
- denotes the outcome of a coin flip
- For a fair coin:
Continuous random variables

- $X$ takes on values in the continuum.
- $p(X = x_i)$, or $p(x_i)$, is called probability density function (PDF) with:
  \[
  \int_x p(x) \, dx = 1
  \]

- Cumulative distribution function (CDF):
  \[
  F(a) = \int_{-\infty}^{a} p(x) \, dx
  \]

- Probability of $X$ taking on a value in an interval:
  \[
  p(a < X \leq b) = F(b) - F(a) = \int_{a}^{b} p(x) \, dx
  \]
Joint and Conditional Probability

- **Joint distribution** of two random variables:
  \[ p(x, y) = p(X = x \text{ and } Y = y) \]

- **Conditional distribution**:
  \[ p(x|y) = p(X = x|Y = y) = \frac{p(x, y)}{p(y)} \]
  \[ \Rightarrow p(x, y) = p(x|y)p(y) \]

*We know already that Y’s value is y. What is the probability that X’s value is x conditioned on that fact?*

- If \( X \) and \( Y \) are **independent** (carry no information about each other) then:
  \[ p(x, y) = p(x)p(y) \]
  \[ p(x|y) = p(x) \]
Joint and Conditional Probability: example

- Ideal cube, dice toss: \( G = \{2, 4, 6\}, \ A = \{4, 5, 6\} \)
- \( p(G) = ? \)
- \( p(A) = ? \)
- \( p(G, A) = ? \)
- \( p(G|A) = ? \)
Joint and Conditional Probability: example

- Ideal cube, dice toss: G={2, 4, 6}, A={4, 5, 6}

- \( p(G) = \frac{1}{2} \)

- \( p(A) = \frac{1}{2} \)

- \( p(G, A) = p(\{4,6\}) = \frac{1}{3} \)

- \( p(G|A) = \frac{p(G,A)}{p(A)} = 2 \cdot p(\{4,6\}) = \frac{2}{3} \)
Theorem of Total Probability

- **Discrete case**
  \[ \sum_x p(x) = 1 \]
  \[ p(x) = \sum_y p(x, y) \]
  \[ p(x) = \sum_y p(x \mid y)p(y) \]

- **Continuous case**
  \[ \int_x p(x) \, dx = 1 \]
  \[ p(x) = \int_y p(x, y) \, dy \]
  \[ p(x) = \int_y p(x \mid y)p(y) \, dy \]
Bayes Theorem

- Relates a conditional to its "inverse":

\[ p(x, y) = p(x|y)p(y) = p(y|x)p(x) \]

\[ p(x|y) = \frac{p(y|x)p(x)}{p(y)} \]
Bayes Theorem: example

- $Y =$ sum of two dice tosses (red and white die)
- $X =$ white die

$p(X = 4 | Y = 10) =$?
Bayes Theorem: example

- $Y = \text{sum of two dice tosses (red and white die)}$
- $X = \text{white die}$

\[ p(X = 4 | Y = 10) = ? \]

\[
= \frac{p(Y = 10 | X = 4)p(X = 4)}{p(Y = 10)} = \frac{\frac{1}{6} \cdot \frac{1}{6}}{\frac{3}{36}} = \frac{1}{3}
\]

- Bayes rule is useful, if $p(Y = 10 | X = 4)$ is easier to compute than $p(X = 4 | Y = 10)$
Bayes Theorem

- **In the context of state estimation:**
  - Assume \( x \) is a quantity that we want to infer from \( y \)
  - Think of \( x \) as state and \( y \) as sensor measurement

Generative model: how state variables cause sensor measurements

\[
p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}}
\]

- Posterior probability
- Independent of \( x \)
- Denoted as normalizer \( \eta \)
Conditional Independence

- All rules presented so far can be conditioned on an arbitrary random variable, e.g. \( Z \)

\[
p(x|y, z) = \frac{p(y|x, z)p(x|z)}{p(y|z)}
\]

\[
p(x, y|z) = p(x|y, z)p(y|z)
\]

- **Conditional Independence:** \( x \) and \( y \) are independent, given that \( z \) is known

\[
p(x, y|z) = p(x|z)p(y|z)
\]

\( \Leftrightarrow p(x|z) = p(x|y, z), p(y|z) = p(y|x, z) \)

\( \Rightarrow \) does **not** necessarily imply absolute independence
Towards the state estimation problem: terminology, notation, probabilistic laws
Terminology: overview

- State

- Measurement model/likelihood

- Motion model (control inputs)

- Inference
State

- The environment or world is considered a (dynamical) system
- The state contains all information that we want to know about the system
- Notation: $x_t$ denotes the state at time $t$
- A state is called complete, if it is the best predictor for the future
  - knowledge of past states, measurements, or controls carries no information about evolution of the state in the future

- Typical examples of states:
  - Position/velocity of an object in global coordinate system
    - continuous, dynamic
  - 3D positions of landmarks
    - continuous, stationary
  - Whether a sensor is broken or not
    - discrete, dynamic
  - Object identity
    - discrete, stationary

This is called a Markov chain
State: examples

- Example for people tracking

\[
\begin{align*}
x_t &= x \\
x_t &= (x, y) \\
x_t &= (x, y, h) \\
x_t &= \{x_t^1, x_t^2\}
\end{align*}
\]
State: example

- Object defined by a point in an image
  - position
  - velocity
  - acceleration

\[
x_t = (x, y) \\
x_t = (x, y, \dot{x}, \dot{y}) \\
x_t = (x, y, \ddot{x}, \ddot{y})
\]

State: example

- Articulated & part-based models
  - set of vertices
  - locations
  - scales
  - constraints

\[ x_t = \{ v^1, v^2, \ldots, v^N \} \]
\[ v^i = (x^i, y^i, s^i) \]

M. Andriluka, S. Roth, B. Schiele,
People Tracking by Detection and People-Detection-by-Tracking, Computer Vision and Pattern Recognition (CVPR'08), Anchorage, USA, June 2008
Measurements

- **Sensor measurements** provide noisy (indirect) information about the state of the dynamical system under consideration
- Increase the knowledge/certainty about the state
- Notation:
  - $z_t$ denotes a measurement at time $t$
  - $z_{t_1:t_2}$ denotes the set of all measurements acquired from time $t_1$ to $t_2$
- Typical examples of sensor measurements:
  - Camera images: pixel-/feature-/object-level
  - Other sensors, e.g.: GPS
Measurement likelihood example (pixel-level)

- Likelihood $p(z|x)$ based on learnt skin color model (Gaussian)
- Assumption: “If the green bounding box contains the face that we are looking for, many pixels within the bounding box should have skin color”

LAB space:
  - designed to approximate human vision
  - $L \rightarrow$ lightness; $A, B \rightarrow$ color

Video of K. Smith
Control inputs (inputs to motion model)

- **Control inputs** carry noisy information about the change of the dynamical system under consideration.
- Reduce the certainty about the state (due to inherent noise).

**Notation:**
- $u_t$ denotes control data at time $t$.
- $u_t$ corresponds to the change of the state in time interval $(t - 1; t]$.
- $u_{t_1:t_2} = u_{t_1}, u_{t_1+1}, \ldots, u_{t_2}$ denotes sequences of control data.

**Typical examples of control inputs:**
- Velocity: setting the velocity of a robot to 10 cm/s for the duration of 5 seconds suggests that the robot is 50 cm ahead of its pose before.
- Odometry: odometers measure the revolution of wheels.
- Gyroscopes: measure angular velocity.
- No input (often the case in visual tracking).
Probabilistic generative laws

- The evolution of state and measurements is governed by probabilistic laws
- State $x_t$ is generated stochastically from state $x_{t-1}$:
  $$p(x_t|x_{0:t-1}, z_{1:t-1}, u_{1:t})$$

- Assuming that the state is complete:
  $$p(x_t|x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t|x_{t-1}, u_t)$$

Markov assumption: example of conditional independence
Probabilistic generative laws

- Measurement $z_t$ is generated stochastically from state $x_t$:

$$p(z_t|x_{0:t}, z_{1:t-1}, u_{1:t})$$

- Assuming that the state is complete:

$$p(z_t|x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t|x_t)$$

Another Markov assumption (conditional independence)
**Probabilistic generative laws**

- \( p(x_t | x_{t-1}, u_t) \) This is what we model! \( p(z_t | x_t) \)

- **Motion model**
  - Specifies, how the state evolves over time as a function of the previous state and the current control data

- **Measurement model/likelihood**
  - Specifies how measurements are generated as function of the state
  - Measurements can be understood as noisy projections of the state

Hidden Markov model
Bayesian inference

- We usually want to estimate the state $x_t$ given sequences of measurements $z_{1:t}$ and control data $u_{1:t}$ and the respective motion $p(x_t|x_{t-1}, u_t)$ and measurement model $p(z_t|x_t)$

- Our estimate of the true state $x_t$ is also called belief:

  $\textbf{Measurement update/correction:}$ calculation of posterior from predicted state and current measurement

  $\text{bel}(x_t) = p(x_t|z_{1:t}, u_{1:t})$

  Posterior distribution of $x_t$ conditioned on all available data
  $\rightarrow$ after including the current measurement $z_t$

  $\textbf{Time update:}$ calculation of predicted state from current state and control input

  $\overline{\text{bel}}(x_t) = p(x_t|z_{1:t-1}, u_{1:t})$

  Prediction of $x_t$ before including the current measurement $z_t$
Introductory example

Odometry input $u_1$: 1m forward.

Measurement $z_2$: here is a door.
How do we get from belief at timestep $t$ to $t+1$ using calculus of probability functions

$\rightarrow$ inference

Bayes filter
Recursive Bayesian filtering

- **Prediction/time update**: calculate prior belief based on dynamic model
- **Correction/measurement update**: calculate posterior belief based on measurement model

\[
bel(x_t) = \eta p(z_t|x_t) \int p(x_t|x_{t-1}, u_t) \, bel(x_{t-1}) \, dx_{t-1}
\]

Basis for all estimation methods that will be presented in this lecture!!!
Recursive Bayes filter algorithm

1. Bayes_filter($\text{bel}(x_{t-1}), u_t, z_t$):
2. for all $x$ do
3. \[ \text{\overline{bel}}(x_t) = \text{Time\_update}(\text{bel}(x_{t-1}), u_t) \]
4. \[ \text{bel}(x_t) = \text{Measurement\_update}(\text{\overline{bel}}(x_t), z_t) \]
5. endfor
6. return \text{bel}(x_t)

Assumes here a finite state-space!
Measurement update step derived

\[ bel(x_t) = \text{Measurement\_update}(\overline{bel}(x_t), z_t) \]

\[ bel(x_t) = p(x_t|z_{1:t}, u_{1:t}) \]

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

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

\[ = \eta p(z_t|x_t) \overline{bel}(x_t) \]
Time update step derived

\[ \overline{bel}(x_t) = \text{Time\_update}(\overline{bel}(x_{t-1}), u_t) \]

\[ \overline{bel}(x_t) = p(x_t | z_{1:t-1}, u_{1:t}) \]
\[ = \int p(x_t | x_{t-1}, z_{1:t-1}, u_{1:t}) p(x_{t-1} | z_{1:t-1}, u_{1:t}) dx_{t-1} \]
\[ = \int p(x_t | x_{t-1}, u_t) p(x_{t-1} | z_{1:t-1}, u_{1:t-1}) dx_{t-1} \]
\[ = \int p(x_t | x_{t-1}, u_t) \overline{bel}(x_{t-1}) dx_{t-1} \]
Bayes update rule

\[
\text{bel}(x_t) = \eta p(z_t|x_t) \int p(x_t|x_{t-1}, u_t) \text{bel}(x_{t-1}) dx_{t-1}
\]

Posterior at time \( t \)

Measurement likelihood

Measurement model

Motion/dynamic model
Summary: Bayes filter framework

- **Given:**
  - Stream of measurements $z_{1:t}$ and control data $u_{1:t}$
  - Measurement model $p(z_t|x_t)$
  - Dynamic model $p(x_t|x_{t-1}, u_t)$
  - Prior/Initial probability of the system state $p(x_0)$

- **Wanted:**
  - Estimate of the state $x_t$ of the considered system
  - Posterior is called **belief**: $\text{bel}(x_t) = p(x_t|u_{1:t}, z_{1:t})$

- **Assumptions:**
  - State is a complete summary of the past
  - The world is Markovian
Outlook

- Bayes filter can only be directly implemented, if integral and product of the PDFs have closed-form solutions, or if we restrict ourselves to a finite state-space!

- Next lectures:
  - (Approximate) representations for belief and concrete PDFs
  - Implementable and tractable filter approximations for continuous estimation problems
  - Hands-on experience

- Readings:
Hands-on example of Bayesian inference

Sensor model
\[ p(\text{image} \mid \text{staircase}) = 0.7 \]
\[ p(\text{image} \mid \text{no staircase}) = 0.2 \]

Prior belief
\[ p(\text{staircase}) = 0.1 \]

Calculate \( p(\text{staircase}) \) and \( p(\text{no}_\text{staircase}) \) after measurement update
Hands-on example of Bayesian inference

\[ p(\text{staircase}) = 0.28 \]

1. for all \( x \) do
2. \[ bel(x_t) = \eta \cdot p(z_t|x_t) \cdot \overline{bel}(x_t) \]
3. endfor

Bayesian inference (measurement update)

\[ p(\text{staircase}|\text{image}) = \eta \cdot p(\text{image}|\text{staircase}) \cdot p(\text{staircase}) = \eta \cdot 0.7 \cdot 0.1 = 0.07\eta \]

\[ p(\text{no staircase}|\text{image}) = \eta \cdot p(\text{image}|\text{no staircase}) \cdot p(\text{no staircase}) = \eta \cdot 0.2 \cdot 0.9 = 0.18\eta \]

Compute normalizer (resulting distribution should integrate to 1):

\[ \eta \cdot (0.07 + 0.18) = 1 \rightarrow \eta = 4 \]

\[ \rightarrow p(\text{no staircase}|\text{image}) = 4 \cdot 0.18 = 0.72 \]
\[ \rightarrow p(\text{staircase}|\text{image}) = 4 \cdot 0.07 = 0.28 \]
State estimation example

- **Dynamical system:** tracking of billiard balls by means of a camera looking from above (without spin, collision, etc.)
- **Pre-requisite:** camera pose known with respect to table
State estimation example

- **Dynamical system**: tracking of billiard balls by means of a camera looking from above (without spin, collision, etc.)
- **Pre-requisite**: camera pose known with respect to table
- Which components are contained in:
  - State $x_t$:
  - Measurement $z_t$:
  - Control input $u_t$:
State estimation example

- **Dynamical system:** tracking of billiard balls by means of a camera looking from above (without spin, collision, etc.)
- **Pre-requisite:** camera pose known with respect to table
- **Which components are contained in (simple model):**
  - State $x_t = (p_x, p_y, \dot{p}_x, \dot{p}_y)$ [m]
    - position and velocity in reference frame of billiard table
  - Measurement $z_t = (i_x, i_y)$ [Pixel]
    - pixel position of ball in camera image
  - Control input $u_t = ()$
    - empty, however, we can assume constant velocity during a time interval
- **Question:** how could the Markov assumption be violated here?
  - E.g. badly calibrated camera
  - Interaction with other balls or table (collisions)
  - Physical aspects: spin, friction, …