Computer Vision
Object and People Tracking

Video Tracking

Prof. Didier Stricker
Dr. Alain Pagani
Kaiserlautern University

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

DFKI – Deutsches Forschungszentrum für Künstliche Intelligenz

http://av.dfki.de
Outline

Video Tracking

- Definition
- Applications
- Bayesian filters for video tracking
- Motion models
- Appearance models
- Dynamic Appearance Models
  - Template Update Problem
  - Machine learning based models
- Occlusion handling
- Fusion
- Evaluation
- Redetection
Video Tracking: definitions

- Known as
  - Visual Tracking
  - Visual Object Tracking
  - Vision-based Tracking
- Defined as
  “the process of estimating over time the location of one or more objects using a camera” ¹
- „Objects“: Person, head, hands, animal, ball, car, plane, semi-finished products, commercial products, cell…

Video Tracking: details

- **Sensor**: a video camera
- **Input**:
  - One (continuous) video sequence
  - Images: unique source of data for measurement
  - Visible variable (in the Hidden Markov Model)
- Camera might be moving (non-static background)
- State is known in the first frame only
- Online (i.e. sequential) Tracking
Approaches to tracking

- **Sequential**
  (recursive, online)
  + Inexpensive → real-time
  - no future information
  - cannot revisit past errors

- **Batch Processing**
  (offline)
  - Expensive → not real-time
  + considers all information
  + can correct past errors

\[ t=1, \ldots, T \]
Tracking applications
Video tracking vs. Camera tracking

- Same input: video sequence
- Different aims:
  - Video Tracking: track an object inside the video sequence over time
  - Camera Tacking: track the pose (rotation + translation) of the camera over time
- Typical application of Camera Tracking: Augmented Reality
Video Tracking applications

- Tracking is an essential step in many computer vision based applications

Detection + Feature Extraction → Tracking → Activity Recognition + Event Recognition → Behavior Analysis + Social Models

Video tracking: applications

- **Scene understanding**
  - indexing events
  - clustering actions (e.g., retail intelligence)
  - semantic scene interpretation
  - mining video collections

Video tracking: applications

- Camera control and natural interfaces
  - Prioritization/selection of multiple cameras
  - Smart rooms
  - Optimized scene sampling (PTZ cameras)
  - Multi-party immersive gaming (control)

Sport analysis

Video Tracking applications

- Medical application and biological research
- Cell tracking and mitosis detection

Ryoma Bise, Zhaozheng Yin, and Takeo Kanade, Reliable Cell Tracking by Global Data Association, ISBI 2011
Video Tracking applications

- Robotics, Industrial production and unmanned vehicles

Adept Quattro s650
https://www.youtube.com/watch?v=8unkXTtfEBQ
Video Tracking applications

- Interactive video, product placement
- Clickable video
Bayesian Filters for Video Tracking

- Components of a Bayesian Filter:
  - State $x_t$
  - Measurement $z_t$
  - Control input $u_t$
  - Motion model $p(x_t|x_{t-1},u_t)$
  - Measurement model $p(z_t|x_t)$

- State is hidden, only measurement is observed
- In visual tracking: no control input
Bayesian Filter: generic equation, components

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

State \( x_t \) (hidden state)
Measurement \( z_t \) (observation)
Motion model (State Transition probability, dynamic model)
Measurement model (Appearance model, likelihood)
Bayesian Filters: robot example

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

Appearance model (Measurement model)  
Motion model

State \( x_t \): 1D (position on the floor)  
Measurement \( z_t \): “I see a door/ I see a wall” (binary)  
Motion model: \( p(x_t | x_{t-1}) = \delta(x_{t-1} + u_t) \) (linear 1D + zero noise)  
Appearance model: \( p(z_t | x_t) \) certain (defined by the map) \( \Rightarrow \) zero noise
Bayesian Filters for Video Tracking: state

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

**State** \( x_t \):
- Point in image (2D)
- Bounding box
- Ellipse
- Shape
- Articulated body
- …
Bayesian Filters for Video Tracking: state

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

State \( x_t \):

- **Appearance model** (Measurement model)
- **Motion model**
Exemplary state: bounding box

\[ x_t = \begin{pmatrix} x_c \\ y_c \\ w \\ h \end{pmatrix} \]

\((x_c, y_c)\): center of the box
\((w, h)\): size of the box (optional)
Exemplary state: 3D bounding box

\[ x_t = \begin{pmatrix} x_r \\ y_r \\ \theta \end{pmatrix} \]

\((x_r, y_r)\): position on the map (road)

\(\theta\): angular rotation (yaw angle)

(fixed bounding box size)
Video Tracking: usual states

2D scene interpretation
- position of centroid (fixed bounding box size): 2D
- Centroid + scale (fixed bounding box aspect ratio): 3D
- Free Bounding Box: 4D \((x, y, w, h)\)
- Free Bounding Box + rotation: 5D

3D scene interpretation
- Position in the „real world“ on a map + orientation: 3D
- 3D Bounding Box: up to 6D
- 3D BBox + rotations: up to 9D
Bayesian Filters for Video Tracking: motion model

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

- **Motion model**: \( p(x_t | x_{t-1} ) \)

  Depends on how the scene is interpreted:
  - In 3D, it is possible to define constraints like
    - Constant acceleration
    - Constant velocity…
  - In 2D tracking: the camera projection breaks the linearity
    - \( p(x_t | x_{t-1} ) \) very difficult to express
    - \( \Rightarrow \) often extremely simplified:
      - Zero velocity + large Gaussian noise (very common)
      - Constant velocity + large Gaussian noise
Bayesian Filters for Video Tracking: motion model

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

Motion model: \( p(x_t | x_{t-1} ) \)

Appearance model (Measurement model)
Bayesian Filters for Video Tracking: measurement

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

**Appearance model** (Measurement model)  **Motion model**

**Measurement** \( z_t \):
- The entire image at time \( t \)
- Full HD resolution, 3 color channels: 6.3 Mo values
  - One measurement \( z_t \) is a point in a 6.3 Mo-dimensional space…
  - Compared with a range-finder or a radar \( \rightarrow \) intractable
  - Fortunately, we do not need to consider measurement, but only the *measurement model*
Video Tracking: measurement model

**Measurement model** \( p(z_t | x_t) \):

- Interpretation of the measurement model:
  "Suppose the state is \( x_t \), what is the probability that the entire image looks like the one I have?"

- Without statistical description of the background, we can reduce this probability to a local neighborhood of the object position defined by \( x_t \).

- Often interpreted as "how similar are the image part covered by state \( x_t \) and the object I expect to see?"

- \( p(z_t | x_t) = similarity(subimage(I, x_t), object_model) \)

- \( \Rightarrow \) this explains the term "appearance model"
Simple Example: PacMan tracking

Track the pink ghost – Frame 0

Object model: reference template

Simple object model based on
- Expected color (hue and saturation)
- Approximate shape
Simple Example: PacMan tracking

Track the pink ghost – Frame 1

Object model: reference template

Motion prediction centered on last position
Measurement step based on object model
New belief
Simple Example: PacMan tracking

Track the pink ghost – Frame 2

Object model: reference template

Motion prediction centered on last position
Measurement step based on object model
New belief
Importance of precise modelling

- PacMan tracking failure example:
  - For a human it is still easy to follow
    - Sudden appearance changes are more likely than teleportation (at least in that game)
  - In the tracking algorithm, this was not modelled
  - Computed belief is correct

- What about real scenes?
Appearance changes in a real video

- In a real sequence, the appearance of tracked objects changes permanently:
  - Object’s out-of-plane rotations
  - Non-rigid objects
  - Face expression
  - Changing illumination
  - Camera motion

- Robust appearance models are required!
Appearance Modeling

- Two components:
  - Visual representation: constructs robust objects descriptors using different types of visual features
  - Statistical modeling: constructs mathematical models for object identification using statistical learning techniques

- Choice for the components depends on the specific context/environment of the tracking task

Visual representation

- Visual features for object description
- Uses low-level computer vision techniques (see first lectures): edges, blobs, corners...

- Can be divided into two major classes:
  - **Global Visual representation:**
    the full image is used for representing the object
  - **Local visual representation:**
    subparts of the image are used (keypoints, patches...)

Global visual representation

- Raw pixel representation

- Distance: SAD, SSD, NCC (lecture 4)
Non-parametric representation

- Distribution of pixel-based features in the region
- Possible features:
  - Color information (RGB or Hue/Sat.)
  - Gradient (magnitude + orientation)
  - Laplacian
  - ...
- \( f = (r, g, b, \| \nabla \|, \theta) \)
- Space \( S \) of dimension \( d \)
- \( \text{distribution} \) of the pixels inside the bounding box in the space \( S \)
Non-parametric representation

- Which model for the representation of the distribution?
  - d-dimensional Gaussian
  - Mixture of Gaussians
  - Histograms
Color Histogram

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

Distance: KL-divergence (lecture 10)

Measurements are obtained by converting pixel values within bounding boxes to LAB color space, and concatenating to form an AB channel histogram.
Global visual representation

- Covariance representation

\[
\begin{bmatrix}
  x & y & I & I_x & I_y & I_{xx} & I_{yy}
\end{bmatrix}
\]

Pixel-wise Features

Covariance Matrix

- Reduced descriptor size
- Distance: Förstner Distance

Fatih Porikli *Using covariance improves computer detection and tracking of humans*, SPIE Newsroom, 2006


W. Förstner and B. Moonen *A Metric for Covariance*, Quo vadis geodesia, 1999
Global visual representation

- Other representations:
  - Wavelet filtering based representation

- Active Contour representation

Local visual representation

- Local Template based
  - Set of smaller local templates for object's parts

- Segmentation based
  - Based on meaningful segments (using segmentation algorithms)

Wang et. al., *Superpixel tracking*, ICCV 2011
Local visual representation

- SIFT-, SURF- or MSER- Based (see first lectures)

Local visual representation

- Local Feature-Pool based
  - Use a huge pool of simple features
    - HOGs
    - Gabor features
    - Haar-like features
  - The most discriminative features are selected from the pool in a learning phase

Viola and Jones, *Rapid object detection using a boosted cascade of simple features*, CVPR 2001
Statistical modeling

- Build a mathematical model for object identification using the selected visual features
- Provides a similarity measure between a proposed object position (image part) and a model of the expected object
- Different approaches:
  - Generative models
  - Discriminative models
  - Hybrid approaches

Generative appearance model

- Attempts to fit the provided data to the model
- Uses parameter estimation (e.g. Expectation maximization)
- Can use online update mechanisms to incrementally learn visual representations for the foreground object

**Discriminative appearance model**

- Object tracking as binary classification
- Maximize the separability between object and non-object regions
- Focus on discovering informative features for visual tracking
- Can incrementally learn discriminative classification functions

- Depends on training sample selection (drawback)

Classifiers For Tracking

Algorithm Idea:
- Learn to distinguish the object from its direct (background) neighborhood

Feature space
- Pixel-based features (color, gradient...)
- Higher level information (wavelets, gradient histograms...)

From: S. Avidan. *Ensemble Tracking* CVPR 2005
Types of discriminative appearance models

- SVM-based DAM
- Boosting-based DAM
- Randomized learning-based DAM
- Discriminant analysis-based DAM
- Codebook learning-based DAM

Dynamic appearance models and drift

- Modern models are based on learning from examples

- Typical (online) tracking task: **only the appearance in the first frame is provided** (single instance model)

- For learning-based methods, examples are taken **while tracking** from the **new frames**

- Consequences:
  - model of expected appearance evolves with time
  - $p(z_t | x_t)$ is not constant
  - depending on the tracking quality, this can lead to **drift**
Drift example

Classical example of drift: Lukas-Kanade

- Simple template-based tracker using optical flow (Lukas Kanade - see lecture 6)

- Given: one (warped) bounding box geometry $q_0$ (quadrangle) and its unwarped content $T_0$ (the template) in the first frame

- Computes: the new position $q_i$ in the following frames

- From $i$ to $i + 1$: Geometry optimization using Taylor expansion of the local image intensities
Choice of the new template

- Question: what should be the template in the frame \( i > 0 \)?
  - **Strategy 1**: no updates
    Keep first template \( T_0 \)
    and take geometry \( q_i \)
  - **Strategy 2**: naive update
    Take new unwarped content \( T_i \)
    with geometry \( q_i \)
The template update problem

Strategy 1: No update
Keep the same reference template ➞ Template depreciates

Strategy 2: Naive Update
Update the template at each frame ➞ drift

Strategy 3:
Update the template at each frame, then correct for drift with first template

The Template Update Problem  Iain Matthews, Takahiro Ishikawa, and Simon Baker
The template update problem

- First assumption: object appearance does not change
  - Does generally not hold
  - Template not fully representative of the object
  - \rightarrow tracking loss

- Update the appearance at every frame
  - Error accumulates
  - \rightarrow generates drift

- Solution: 2-step tracking
  - First compute a new position $\tilde{q}_i$ using the last template $T_i$
  - Then correct for drift: compute $q_i$ starting from $\tilde{q}_i$ with template $T_0$
The template update problem: limitations

- Works well for gradient trackers / template trackers
- Not adapted for more complicated objects
- Other strategies are necessary
  - Use advance machine learning techniques
  - Create a pool of object appearances
  - Combine Tracking and Detection
Example: Tracking-Learning-Detection

*Tracking-Learning-Detection*, Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, PAMI 2010
Handling Occlusion in Visual Tracking

Even if the appearance of the tracked object is well modelled, the object can be occluded in the vision field of the camera.
Handling Occlusion in Visual Tracking

Two strategies for coping with occlusions

Taking partial occlusion into account:
- Divide the object into parts
- Compute similarity of parts
- Discard the worst part

Detecting loss of tracks
- When the object is not visible: missing measurement
- Kalman Filter propagates uncertainty while meas. is missing
Fusion of multiple source for tracking

- Although the image sequence is the unique source of information, different visual features can provide different types of measurement

- A visual tracker can fuse different features
- Different strategies:
  - Fusion at tracker level
  - Fusion at measurement level
Tracker-level fusion

- Parallel fusion of trackers
  - Multiplication of the posterior probabilities (indep. Trackers)
  - Trackers used as observables of a Markov Network

- Sequential fusion of trackers
  - Features are considered available at subsequent time-instant
  - Uses an EKF approach

Measurement-level fusion

- Measurements are combined internally by the tracking algorithm

- Mechanisms for fusion:
  - Voting procedure
  - Saliency maps
  - Bayesian networks
  - Mutual Information
  - Multi-feature particle filter: linear combination

\[
p(z_t|x_t) = \sum_{j=1}^{N} \alpha_j p_j (z_t|x_t)
\]

Performance evaluation

- Requires an evaluation sequence with **ground truth**
- **Precision**: distance in pixels between centers of bounding boxes
- **Overlap**: relative overlap between bounding boxes

\[
p = \|c_A - c_B\|\]

\[
o = \frac{C}{A + B - C}\]
Performance evaluation plots

- Visualization for a full sequence through plots:
  - Precision plots: % of frames below a given precision threshold
  - Success plots: % of frames above a given overlap threshold
How good are current trackers?

- Test with 50 sequences and 29 state-of-the-art trackers
- Results are sequence-specific
- Still no generic object tracker
Redetection – real tracker

A complete tracking framework needs to know when it fails

Detection and loss detection: (re-)initialize the system
Tracking: measurement processing and state update
Models: all useful prior information about objects, sensors, and environment
Conclusion

- Visual Tracking is a very specific kind of Bayesian tracking
- Two major difficulties
  - Difficult to establish a correct motion model
  - Measurement model needs an appearance model
- One possible solution: track and detect object at the same time
Literature and online sources

„Video Tracking“ by E. Maggio and A. Cavallaro

Online resources:

http://www.videotracking.org/
Thank You!