Dictionary:

- [noun] “The pursuit (of a person or animal) by following tracks or marks they left behind”
- [verb] “Observe or plot the moving path of something (e.g., to track a missile)”

What does it mean in Computer Vision?

Many thanks to: H. Grabner, L. van Gool, and V. Ferrari for some of the slides & videos.
What is Tracking

 actual object position

LOCALIZE “IT” IN THE NEXT FRAMES
Why do we need it

What is tracking for you? Why do you think it is relevant and may be important?

Where could it be useful, in real-life applications and engineering scenarios?

Task: “List applications you can think of on a piece of paper”

Discuss in groups of 3-4
Autonomous Driving
Surveillance, Safety, Security

Orcun Goksel, ETH Zurich
Sports
Video Editing

Adobe After Effects
“Plexus Hands”
by Mazyar Sharifian

Before

After

www.cubichead.com/tutorials
Applications: VR/AR glasses

Microsoft HoloLens
SfM: Structure from Motion

- Tracked Points gives correspondences
Defense

“Top Gun”
Of course, “very importantly”
The Cow Tracker
Applications

• Structure-from-Motion
• Autonomous Driving
• Gesture/Action Recognition
• Augmented Reality
• Navigation
• Safety and Security
• Medical Targeting / Guidance
• Motion Compensation
• …
You will be able to:

1. Determine applications of tracking and identify problems solvable by tracking
2. Analyze what methods could work in a practical scenario / situation
3. Assess potential limitations / pitfalls of particular approaches and scenarios
4. Propose an optimal tracking solution

How will we get there:

• (some) common tracking methods
• Few particular keywords & implementation
• What not: details of all individual implementations; cf. “how to google”
Think about

Q. What tracking method would you use in each following application scenario? What limitations you may expect?

Task: “Discuss each in groups”

App1. Safety: In a lumbar mill, you wish to use CV to stop the blade if a hand reaches nearby.

App2. Medical: You wish to track the ultrasound probe, to relate images in 3D space.

App3. Autonomous driving: Tracking other nearby vehicles to adjust speed and course.

(AppX. Your favourite tracking app)
What to track?
What to track?

center point
What to track?

multiple points
What to track?

structure
What to track?

(body) parts
What to track?

region
What to track?

outline
Approaches

(i) Feature tracking
   generic
   corners, blob/contours, regions, …

(ii) Model-based tracking
    application-specific
    face, human body, …
Tracking Requirements

• Strongly depends on the application!

  Robust, Accurate, Fast,…

• Constrain the tracking task!

Information about the object, dynamics, …
Tracking Cues

Object

Saliency

Scene

Model/
Tracking History

Orcun Goksel, ETH Zurich
Motion as a Cue
Motion as a Cue

- Eye perceptive to temporal changes (gradients)
- “Event based camera”
General Tracking Loop

- predict to $t+1$
- measure at $t+1$
- update location
- update model
- time $t$
Trajectory
(Temporal Filtering)
Temporal Filtering/Predictions

- To predict location
- To reduce noise
- To disambiguate multiple objects

Kalman Filtering

Prior knowledge of state $\rightarrow$ $P_{k-1|k-1}$, $\hat{x}_{k-1|k-1}$

Prediction step
Based on e.g. physical model

Next timestep $k \leftarrow k + 1$

$P_{k|k-1}$, $\hat{x}_{k|k-1}$

Update step
Compare prediction to measurements

Output estimate of state

Measurements $y_k$
08.10.2009

<< Rudolf Kalman, ETH-Zurich emeritus professor of mathematics, is awarded the National Medal of Science by Barack Obama – one of the highest accolades for researchers in the USA.

In January 2008, Hungarian-born Kalman received the Charles Draper Prize, which is regarded as the “Nobel Prize” of the engineering world. >>

http://www.ethlife.ethz.ch/archive_articles/091008_kalman_per
Steps of Tracking

• Recap: Particle filtering
  – Tracking can be seen as the process of propagating the posterior distribution of state given measurements across time.
Particle Filter

\[ p(p_{t-1}, \dot{p}_{t-1} \mid z_{t-1}) \]

\[ \downarrow \text{prediction} \]

\[ p(p_t, \dot{p}_t \mid z_{t-1}) \]

weighing with \( p(z_t \mid p_t) \)

\[ \downarrow \text{update} \]

\[ p(p_t, \dot{p}_t \mid z_t) \]
Traditional/Simple Tracking

$t=1$
initialization

$t=2$
position in prev. frame
candidate new positions (e.g., dynamics)
best new position (e.g., max color similarity)
Tracking-by-Detection

detect object(s) independently in each frame

associate detections over time into tracks
Outline

- Region Tracking
- Point Tracking
- Template Tracking
- Tracking-by-Detection
  - a specific target
  - object class
- Model-based Body Articulation
- On-line Learning
- Misc (preventing drift, context, issues)
Region Tracking
(and Mean Shift Algorithm)
Background Modeling

For known (fixed) background, simply save it and subtract from each frame.

Sources of errors, e.g.:
* same color as background
* lighting changes
* camera noise/motion
* occlusion

Large moving blobs are the objects (foreground)

Noise must be filtered, to extract the object.
Deformable models

• One option: Fit deformable curves
Mean Shift Method

• Mean Shift Tracking (general description) Maximize similarity between tracked and target regions through evolution towards higher density in a parameter space

• Can be used to find the object from background modeling, by assuming that the object is formed of a large group of densely located pixels (in contrast to noise as fewer scattered foreground pixels)

• A mean (center) location is iteratively updated by moving it to the centroid of pixels within a chosen radius

[Comaniciu and Meer, ICCV’99]
Intuitive Description
Intuitive Description
Intuitive Description
Intuitive Description
Intuitive Description
Typically this search only takes a few iterations.
Intuitive Description

Initialize multiple means and pick the location where many converges.
Example: Safety Monitoring
Outline

- Region Tracking (and Mean Shift Algorithm)
- Point Tracking
- Template Tracking

- Tracking-by-Detection
  - a specific target
  - object class

- Model-based Body Articulation
- On-line Learning

- Misc (preventing drift, context, issues)
Point Tracking
(and Aperture Problem)
Estimate Optimal Transformation
When can we (not) estimate motion?

Q1. Which direction is the pattern behind the circular hole moving in physical space?

- a) d) c) 2 mins 1 min 2 mins 1 min 2 min 1 min

Q2. Motion in 1D: What mathematical property of curves make it impossible to determine the direction of motion from red to green line in the last case?

Q3. What is common between Q1 & Q2?
Sum of Squared Differences

\[ E(h) = \left[ I_0(x+h) - I_1(x) \right]^2 \]
Displacement

\[ E(h) = \left[ I_0(x+h) - I_1(x) \right]^2 \]

\[ E(h) \approx \left[ I_0(x) + hI_0'(x) - I_1(x) \right]^2 \]

\[ \frac{\partial E}{\partial h} \approx 2I_0'(x) \left[ I_0(x) + hI_0'(x) - I_1(x) \right] = 0 \]

\[ h \approx \frac{I_1(x) - I_0(x)}{I_0'(x)} \]
Intuition

\[ h \approx \frac{I_1(x) - I_0(x)}{I_0'(x)} \]
Problem 1: Zero Gradient

\[ h \approx \frac{I_1(x) - I_0(x)}{I_0'(x)} \]
Problem 1: “Aperture problem”

- For tracking to be well defined, nonzero gradients in all possible directions are needed.
- If no gradient along one direction, we cannot determine relative motion in that axis.
Problem 2: Local Minima

- Motion to closest minimum has to be assumed
- Indirect result: Frame-rate should be faster than motion of half-wavelength (Nyquist rate)
- Nonconvex regions may indicate multiple solns
Problem 2: Local Minima
Recall: Optical Flow in Motion Estimation

- OF recovers (smooth) motion everywhere
- Least-squares regularization: Horn-Schunk makes smooth spatial change assumption
- In contrast, tracking seeks a single motion!
Recall: Optical Flow

\[ I_x u + I_y v + I_t = 0 \]

\[ I_x = \frac{\partial I}{\partial x}, \quad I_y = \frac{\partial I}{\partial y}, \quad I_t = \frac{\partial I}{\partial t} \]

\[ u = \frac{dx}{dt}, \quad v = \frac{dy}{dt} \]

1 equation in 2 unknowns
Treating Aperture Problem in Tracking

• Get additional info to constrain motion:
  – OF: Smoothly regularize in space
  – Tracking: Assume single motion for a region

• Spatial coherence constraint:
  “A pixel’s neighbours all move together”
Least Squares Problem: Single motion with multiple equations

\[
\begin{bmatrix}
I_x(p_1) & I_y(p_1) \\
I_x(p_2) & I_y(p_2) \\
\vdots & \vdots \\
I_x(p_{25}) & I_y(p_{25})
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= -
\begin{bmatrix}
I_t(p_1) \\
I_t(p_2) \\
\vdots \\
I_t(p_{25})
\end{bmatrix}
\]

Over determined System of Equations

\[
A \ d = b \\
25x2 \ 2x1 \ 25x1
\]

Pseudo Inverse

\[
(A^T A) \ d = A^T b \\
2x2 \ 2x1 \ 2x1
\]

\[
\begin{bmatrix}
\sum I_x I_x & \sum I_x I_y \\
\sum I_x I_y & \sum I_y I_y
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= -
\begin{bmatrix}
\sum I_x I_t \\
\sum I_y I_t
\end{bmatrix}
\]
Eigenvectors of $A^T A$

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

- $(u,v)$ can only be found, if this is solvable, i.e. 2x2 image structure matrix is invertible with no small eigenvalue

- This matrix and the requirement sound familiar – have we seen these before?

- Recall Harris corner detector!

- Thus, “good image features (with large structural eigenvalues) are also good for tracking (with which we can find motion)”
Interpreting the Eigenvalues

- "Corner": \( \lambda_2 \approx \lambda_1 \) and large
- "Edge": \( \lambda_2 \gg \lambda_1 \)
- "Flat" region: \( \lambda_2 \ll \lambda_1 \)
Samples: Edge / Low Texture / High Texture
Example
Outline

- Region Tracking (and Mean Shift Algorithm)
- Point Tracking (and Aperture Problem)
- Template Tracking
- Tracking-by-Detection
  - a specific target
  - object class
- Model-based Body Articulation
- On-line Learning
- Misc (preventing drift, context, issues)
Template Tracking
Template Tracking

• Keep a template image to compare with each frame
• This is typically applied for small patches, e.g. 5x5
• Why not run it for the entire object (for a larger window)

• Locally, translation is sufficient to explain motion; but…
Lucas-Kanade Template Tracker

- Motion is more complex in a larger window

- Nonetheless, we can easily generalize the motion model to other parametric models!
  e.g., translation, affine, projective, “warp”

\[
E(u, v) = \sum_{x,y} [I(x + u, y + v) - T(x, y)]^2
\]

\[
E(p) = \sum_{x,y} [I(W(x; p)) - T(x, y)]^2
\]
Lucas-Kanade Template Tracker

- From Points to templates
- Estimate “optimal” warp $W$

\[ \sum_x [I(W(x; p)) - T(x)]^2 \]

\[ \sum_x [I(W(x; p + \Delta p)) - T(x)]^2 \]
Lucas-Kanade Template Tracker

Step 1. Warp $I$ to obtain $I(W([x \ y]; P))$

Step 2. Compute the error image $T(x) - I(W([x \ y]; P))$

Step 3. Warp the gradient $\nabla I$ with $W([x \ y]; P)$

Step 4. Evaluate $\frac{\partial W}{\partial P}$ at $([x \ y]; P)$ (Jacobian)

Step 5. Compute steepest descent images $\nabla I \frac{\partial W}{\partial P}$

Step 6. Compute Hessian matrix $\sum (\nabla I \frac{\partial W}{\partial P})^T (\nabla I \frac{\partial W}{\partial P})$

Step 7. Compute $\sum (\nabla I \frac{\partial W}{\partial P})^T (T(x, y) - I(W([x, y]; P)))$

Step 8. Compute $\Delta P$

Step 9. Update $P \leftarrow P + \Delta P$
Example
Example: Tracking Liver in Ultrasound

[ Makhinya and Goksel: “Motion Tracking in 2D Ultrasound Using Vessel Models and Robust Optic-Flow”, MICCAI CLUST, 2015 ]

- Red circle: Our tracking
- Green plus: Manual annotation

Orcun Goksel, ETH Zurich
Outline

- Region Tracking (and Mean Shift Algorithm)
- Point Tracking (and Aperture Problem)
- Template Tracking (Lucas-Kanade)
- Tracking-by-Detection
  - a specific target
  - object class
- Model-based Body Articulation
- On-line Learning
- Misc (preventing drift, context, issues)
Tracking by Detection
(of a specific target)
3D Object Detection

Reference image(s) of the object to detect

Test image
3D Object Detection

Reference image(s) of the object to detect

Test image
1. Detect Keypoints

– invariant to scale, rotation, or perspective

100 strongest feature points in the reference image

300 strongest feature points in the test image
2. Build Feature Descriptors
Histogram of Oriented Gradients

Example: HOG is a (rotation invariant) feature descriptor

Bin magnitudes of gradients as a histogram

Also, object shapes defined by edges, thus HOG over entire objects can be descriptive

Useful to track specific points

See also SIFT, SURF, ...
3. Match Keypoint Descriptors

- Search in the Database

Query (from image)

Database (of object)
3. Search in the Database
4. Outlier Elimination
Summary

Keypoint Detection

Keypoint Recognition

Search in the Database

Robust 3D Pose Calculation (RANSAC)

Geometric verification
Overall: 3.42 ms
Find Pts: 1.25 ms
Track Pts: 0.32 ms
Features: 1.16 ms
Outliers: 0.50 ms
Pose: 0.19 ms
Corners: 166
Matched Features: 29
Wrong Rotation: 0
Bad Linetest: 0
Bad Homographytest: 0
Correct: 29
From Cache: 0
From ActiveSearch: 20
Levels: 0 0 0 0 0 0 0 0 0 0
Rotation: 6
Avg. Reproj. Err: 1.31
Computer Vision

[Wagner et al. '09]

Orcun Goksel, ETH Zurich
Outline

- Region Tracking (and Mean Shift Algorithm)
- Point Tracking (and Aperture Problem)
- Template Tracking (Lucas-Kanade)

- Tracking-by-Detection
  - a specific target (e.g., keypoints + Ransac)
  - object class

- Model-based Body Articulation

- On-line Learning

- Misc (preventing drift, context, issues)
Tracking by Detection
(of the object class)
also for “Multiple Object Tracking”
Tracking-by-Detection

detect object(s) independently in each frame

associate detections over time into tracks
Multiple Objects

Frame 1

Frame 5

Frame 9
Examples:
Multiple Object Tracking
How to get the detections?

Supervised Learning
(Support Vector Machines, Random Forests, Neural Networks, ...)

Persons
Background

Orcun Goksel, ETH Zurich
Using the classifier
Space-Time Analysis

• Collect detections in space-time volume

[Leibe et al. CVPR’07]

Orcun Goksel, ETH Zurich
Trajectory Estimation

- Trajectory growing and selection
Trajectory Estimation

- Trajectory growing and selection

Space Time Volume
Driving

Input (Object Detections)  “Tracking” Result
Outline

- Region Tracking (and Mean Shift Algorithm)
- Point Tracking (and Aperture Problem)
- Template Tracking (Lucas-Kanade)
- Tracking-by-Detection
  - a specific target (e.g., keypoints + Ransac)
  - object class (multiple object tracking)
- Model-based Body Articulation
- On-line Learning
- Misc (preventing drift, context, issues)
Model based Tracking
Articulated Tracking: Part-Based Models

- Intuitive model of an object
- Model has two components
  1. parts (2D image fragments)
  2. structure (configuration of parts)
- Dates back to Fischler & Elschlager 1973
Parts-based analysis

Objective: detect human and determine upper body pose (layout)

Model as a graph labelling problem
- Vertices $\mathcal{V}$ are parts, $a_i, i = 1, \ldots, n$
- Edges $\mathcal{E}$ are pairwise linkages between parts
- For each part there are $h$ possible poses $p_j = (x_j, y_j, \phi_j, s_j)$
- Label each part by its pose: $f : \mathcal{V} \rightarrow \{1, \ldots, h\}$, i.e. part $a$ takes pose $p_{f(a)}$. 

Orcun Goksel, ETH Zurich
Parts-based analysis

Pictorial structure model – CRF

- Each labelling has an energy (cost):

\[
E(f) = \sum_{a \in V} \theta_a; f(a) + \sum_{(a,b) \in \mathcal{E}} \theta_{ab}; f(a)f(b)
\]

- Fit model (inference) as labelling with lowest energy

Features for unary:
- colour
- HOG
  for limbs/torso

Orcun Goksel, ETH Zurich
Walking

• What temporal info can we use for tracking?

• What variation would we expect in population?
Articulation Space

Tracking Articulated Motion as High-Dimensional Inference

- Walking cycles have one main (periodic) DOF
- Regressors to learn this (latent) space, and its variation (Gaussian Process regression, **PCA**, etc)
- (Pose,Silhouette) training data can be obtained by 3D rendering
Articulation Space

Tracking Articulated Motion as High-Dimensional Inference

- Walking cycles have one main (periodic) DOF
- Regressors to learn this (latent) space, and its variation (Gaussian Process regression, PCA, etc)
- \((\text{Pose, Silhouette})\) training data can be obtained by 3D rendering

\[ P(\text{Silhouette} | k) \] perform inference on silhouettes

\[ P(\text{Pose} | k) \] recover pose from latent space
Articulation Space Tracking
# Outline

<table>
<thead>
<tr>
<th>Feature</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Region Tracking</td>
<td>(and Mean Shift Algorithm)</td>
</tr>
<tr>
<td>Point Tracking</td>
<td>(and Aperture Problem)</td>
</tr>
<tr>
<td>Template Tracking</td>
<td>(Lucas-Kanade)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tracking-by-Detection</td>
<td></td>
</tr>
<tr>
<td>– a specific target (e.g., keypoints + Ransac)</td>
<td></td>
</tr>
<tr>
<td>– object class (multiple object tracking)</td>
<td></td>
</tr>
<tr>
<td>Model-based Body Articulation</td>
<td></td>
</tr>
<tr>
<td>On-line Learning</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Misc</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Misc (preventing drift, context, issues)</td>
<td></td>
</tr>
</tbody>
</table>
Tracking as On-line learning (updating tracking models)
Tracking as Classification

- Learning current object appearance vs. local background.
Tracking as Classification

object vs. background
Tracking as Classification

object vs. background
Tracking Loop

- **actual object position**
- **search Region**
- **create confidence map**
- **evaluate classifier on sub-patches**
- **update classifier (tracker)**
- **analyze map and set new object position**

- **from time t to t+1**
For tracking “the invisible”
When does it fail...
When does it fail…

WHY

WHY
Computer Vision

- evaluate classifier on sub-patches
- search Region
- from time $t$ to $t+1$

- create confidence map
- Set new object position
- update classifier (tracker)

Self-learning!
Drift

Tracked Patches

Confidence
Drift
Outline

- Region Tracking (and Mean Shift Algorithm)
- Point Tracking (and Aperture Problem)
- Template Tracking (Lucas-Kanade)

- Tracking-by-Detection
  - a specific target (e.g., keypoints + Ransac)
  - object class (multiple object tracking)

- Model-based Body Articulation
- On-line Learning

- Misc (preventing drift, context, issues)
Combining Tracking and Detection (to avoid drift)
Refining an object model

- Only thing we are sure about the object is its initial model (e.g. appearance in first frame)
- We can “anchor” / correct our model with this information, in order to help avoid drift
Recover from Drift using a fixed/anchor model (e.g. first frame)
Context in Tracking
Humans use context to track

- … objects which change their appearance very quickly.

- … occluded objects or object outside the image.

- … small and/or low textured objects or even “virtual points”.

Orcun Goksel, ETH Zurich
Using Supporters
Assumptions should hold

With Supporters
In Practice
Which strategy to use?
Depends... No single solution

Some rule-of-thumb suggestions:

• If you can alter the “object” to be tracked, ➔ modify/add tracking info
e.g. optical IR markers, mark with patterns, etc

• If object is fixed/known, but modification not possible/desired ➔ Utilize known info
e.g. use a template image and/or known object features

• If object unknown/variable object, but resides in a known (static) environment ➔ bg modeling!

• If none above, simply follow from initial image/location, or use sophisticated learning techniques for detection

Tracking v.s. segmentation/localization:
Key difference is TEMPORAL consistency

Orcun Goksel, ETH Zurich
Let’s apply

Q. What tracking method would you use in each following application scenario? What limitations you may expect?

Task: “Discuss one (or more) in groups”

App1. Safety: In a lumbar mill, you wish to use CV to stop the blade if a hand reaches nearby.

App2. Medical: You wish to track the motion of an ultrasound probe, to relate images in space.

App3. Autonomous driving: Tracking other nearby vehicles to adjust speed and course.

AppX. Your favourite tracking app
Problems in Tracking
Tracking Issues

• Initialization

Time $t = 0$
Tracking Issues

• Obtaining observation…
  – **Generative**: “render” the state on top of the image and compare
  – **Discriminative**: classifier or detector score

• …and dynamics model
  – specify using domain knowledge
  – learn (very difficult)
Tracking Issues

• Model- vs. Model-free-Tracking
Tracking Issues

• Nonlinear dynamics
  – Sometimes needed to keep multiple trackers in parallel
  – E.g., for abrupt direction changes („Persons“)
Tracking Issues

- Prediction vs. Correction (cf. Kalman Filtering)
  - If the **dynamics** model is **too strong**, tracking will end up **ignoring the data**.
  - If the **observation** model is **too strong**, tracking is **reduced to repeated detection**.
Tracking Issues

- Data Association – Multiple Object Tracking
  - What if we don’t know which measurements to associate with which tracks?
Tracking Issues

- Data Association – Occlusions / Self Occlusions
Tracking Issues

• Data Association – Fast Motion
Tracking Issues

• Data Association –
  Background / Appearance Change
  – Cluttered Background
  – Changes in shape, orientation, color,…
• Drift
  – Errors caused by dynamical model, observation model, and data association tend to accumulate over time
Summary

- Region Tracking (and Mean Shift Algorithm)
- Point Tracking (and Aperture Problem)
- Template Tracking (Lucas-Kanade)

- Tracking-by-Detection
  - a specific target (e.g., keypoints + Ransac)
  - object class (multiple object tracking)

- Model-based Body Articulation
- On-line Learning

- Misc (preventing drift, context, issues)