Tracking

actual object position

Time t

Time t+1

LOCALIZE *IT* IN THE NEXT FRAMES
What to track?
What to track?

center point
What to track?

multiple points
What to track?

region
What to track?

(body) parts
What to track?

structure
What to track?

outline
Applications

• Structure-from-Motion
• Gesture/Action Recognition
• Video editing
• Augmented Reality
• Navigation
• Safety and Security
• Motion Compensation
• ….
Applications: SfM

- Tracked Points gives correspondences
Applications: SfM
Motion as a Cue
Motion as a Cue
Approaches

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

(ii) Model-based tracking
    application-specific
    face, human body, …
Trajectory
(Temporal Filtering)
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
Temporal Filtering/Predictions

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

Kalman Filtering

Prior knowledge of state

\[ P_{k-1|k-1} \]
\[ \hat{x}_{k-1|k-1} \]

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

\[ P_k|k \]
\[ \hat{x}_k|k \]

Measurements

\[ y_k \]
Kalman Filtering

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

Parametrization matters!

with Kalman Filter
acceleration model

with Kalman Filter
only velocity model
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

\[ p(z_t \mid p_t) \]

\[ \downarrow \text{update} \]

\[ p(p_t, \dot{p}_t \mid z_t) \]

---

**Computer Vision**

**CONDENSATION**
General Tracking Loop

- **time t**: Predict to $t+1$ and measure at $t+1$
- **update model**: Update model
- **update location**: Update location
Outline

- Region Tracking
- Point Tracking
- Template Tracking

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

- Model-based Body Articulation
- On-line Learning

- Misc (preventing drift, context, issues)
Region Tracking
(and Mean Shift Algorithm)
Computer Vision

Background Modeling

Input

→

Background Model

Moving Foreground Blobs (Objects)
Mean Shift Tracking

- The mean shift tracker tracks a region, with a prescribed (color) distribution

- The similarity between the tracked region and the target region is maximized, through evolution towards higher density in a parameter space

- Typically this search only takes a few iterations

[Comaniciu and Meer, ICCV'99]
Meanshift Tracking

- Region of interest (Kernel)
- Center of mass
- Mean Shift vector
- Measurements
Intuitive Description
Intuitive Description
Intuitive Description
Intuitive Description
Intuitive Description
Intuitive Description
Example: Safety Monitoring
Point Tracking
(and Aperture Problem)
Estimate Optimal Transformation
Sum of Squared Differences

\[ I_1(x) = I_0(x+h) \]

\[ E(h) = [I_0(x+h) - I_1(x)]^2 \]
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 A: Local Minima
Problem A: Local Minima
Problem B: Zero Gradient

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

No gradient along one direction:
Problem B: Aperture problem

No gradient along one direction:
Recall: Optical Flow

\[ l_0 \text{ (time } t) \rightarrow \text{ motion } \mathbf{v} \rightarrow l_1 \text{ (time } t+1) \]
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
“Solving” the Aperture Problem

• How to get more equations for a pixel?
• Spatial coherence constraint: Pixel’s neighbors have the same movement
Least Squares Problem

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

Over determined System of Equations

\[
A \cdot d = b
\]

25x2  2x1  25x1

Pseudo Inverse

\[
(A^T A) \cdot d = A^T b
\]

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 $A^T A$

$$A^T A = \left[ \begin{array}{cc} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{array} \right] = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y] = \sum \nabla I (\nabla I)^T$$

- Haven’t we seen an equation like this before?
- Recall the Harris corner detector!
- “Good Features to Track”
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
Template Tracking
Lucas-Kanade Template Tracker

- Lucas-Kanade is typically 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”

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

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

- From Points to templates
- Estimate „optimal“ warp \( W \)

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

\[
\sum_x [I(W(x; p + \Delta p)) - T(x)]^2
\]
Computer Vision

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 $P \leftarrow P + \Delta P$
Computer Vision

Baker & Matthews, IJCV'04, Lucas-Kanade 20 Years On: A Unifying Framework
Example
Example: Tracking Liver in Ultrasound

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

Our tracking
Manual annotation
Outline

- Region Tracking
- Point Tracking
- Template Tracking
- Tracking-by-Detection
  - of a specific target
  - of the object class
- Model-based Body Articulation
- On-line Learning
- Misc (preventing drift, context, issues)
Model-Based Tracker
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
3. Match Keypoint Descriptors

- Search in the Database
3. Search in the Database
4. Outlier Elimination
Summary

Keypoint Detection

Keypoint Recognition

Search in the Database

Robust 3D Pose Calculation (RANSAC)

Geometric verification

Computer Vision

[Image of staple remover box and keypoint detection]
Tracking by Detection
(of the object class)

“Multiple Object Tracking”
Tracking-by-Detection

- detect object(s) independently in each frame
- associate detections over time into tracks
Multiple Objects
Examples: Multiple Object Tracking
How to get the detections?

Persons

Background

Supervised Learning
Using the classifier
Space-Time Analysis

- Collect detections in space-time volume
Trajectory Estimation

- Trajectory growing and selection
Trajectory Estimation

- Trajectory growing and selection

Space Time Volume
Computer Vision

Result

Input (Object Detections)

“Tracking” Result
Towards Scene Interpretation
Model based Tracking
Use Case: Body Articulation

• Goal
  – Recover a person’s body articulation
  – Detailed parameterization in terms of joint locations or joint angles

• Two classes of approaches:
  – Part-based models
  – Articulated tracking as high-dimensional inference
Articulated Tracking with Part-Based Model

- part appearance + relative geometry.
Pictorial Structures

- 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
Part-based analysis

Localize multi-part objects at arbitrary locations in an image

- Generic object models such as person or car
- Allow for articulated objects

To fit model to image: minimize an energy (or cost) function that reflects both

- Appearance: how well each part matches at given location
- Configuration: degree to which parts match 2D spatial layout
Part-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: $f : \mathcal{V} \to \{1, \ldots, h\}$, i.e. part $a$ takes pose $p_{f(a)}$. 
Part-based analysis

Pictorial structure model – CRF

- Each labelling has an energy (cost):
  \[
  E(f) = \sum_{a \in V} \theta_a; f(a) + \sum_{(a,b) \in E} \theta_{ab}; f(a)f(b)
  \]
  - Unary terms (appearance)
  - Pairwise terms (configuration)

- Fit model (inference) as labelling with lowest energy

Features for unary:
- colour
- HOG
  for limbs/torso
Use Case: Body Articulation

• Goal
  – Recover a person’s body articulation
  – Detailed parameterization in terms of joint locations or joint angles

• Two classes of approaches:
  – Part-based models
  – Articulated tracking as high-dimensional inference
Articulation as a manifold

- Low-dimensional representation of high-dimensional body pose representations, e.g. using PCA
- Tends to be less generic, e.g. for specific, known actions like walking
- Added robustness, but may hallucinate
- Can better infer 3D motion from monocular data

[Sidenbladh ECCV’00] [Jaegli ACCV’07] [Andriluka CVPR’08]
Articulation as a manifold

How can we get from here to here?
Articulation as a manifold

Overview of a tracker for walking

Pedestrian Detection
[Leibe CVPR’05]

Articulated Tracking
based on
[Jaeggli ACCV’07]

Pedestrian Tracking
[Ess CVPR’08]
Training data

- Multiple sequences of „Pose + Silhouette“ pairs
- Pose recorded with MoCap
- Silhouette via 3D rendering
Articulated motion in latent space

- Gaussian Process regression from latent space to
  - Pose \( \rightarrow = P(\text{Pose} | k) \) to recover original pose from latent space
  - Silhouette \( \rightarrow = P(\text{Silhouette} | k) \) to do inference on silhouettes
- Regressors learn from training data

walking cycles have one main (periodic) DOF
Articulation as a manifold
Results
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** from time \( t \) to \( t+1 \)
- Create confidence map
- Evaluate classifier on sub-patches
- Analyze map and set new object position
- Update classifier (tracker)
“Simple tracking”
“Tracking the Invisible”
When does it fail...
When does it fail…

WHY
Computer Vision

update classifier (tracker)

from time t to t+1

evaluate classifier on sub-patches

analyze map and set new object position

create confidence map

Self-learning!
Drift

Tracked Patches

Confidence

![Graph showing tracked patches confidence over frame number]
Drift
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 that
• This can limit drift
Recover from Drift using a fixed/anchor model (e.g. first frame)

[Grabner et al. ECCV'08]
Context in Tracking
SUPPORTERS help Tracking of...

- ... objects which change their appearance very quickly.
- ... occluded objects or object outside the image.
- ... small and/or low textured objects or even “virtual points”.
ETH-Cup Video
ETH-Cup: Humans
Of the Web Tracker
ETH-Cup: With Supporters
Beyond the Image

With Supporters
There are also failure cases…

With Supporters
Problems in Tracking
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
Tracking Issues

- Initialization

Time $t = 0$

object position
Tracking Issues

• Prediction vs. Correction
  – If the \textit{dynamics} model is \textit{too strong}, will end up ignoring the data.
  – If the \textit{observation} model is \textit{too strong}, tracking is reduced to repeated detection.

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

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

- Data Association – Fast Motion
Tracking Issues

- Data Association – Background / Appearance Change
  - Cluttered Background
  - Changes in shape, orientation, color,...
Tracking Issues

- Data Association – Occlusions / Self Occlusions
Tracking Issues

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

- Drift
  - Errors caused by dynamical model, observation model, and data association tend to accumulate over time