Robotic 3D Vision

Lecture 17: 3D Object Tracking

WS 2020/21

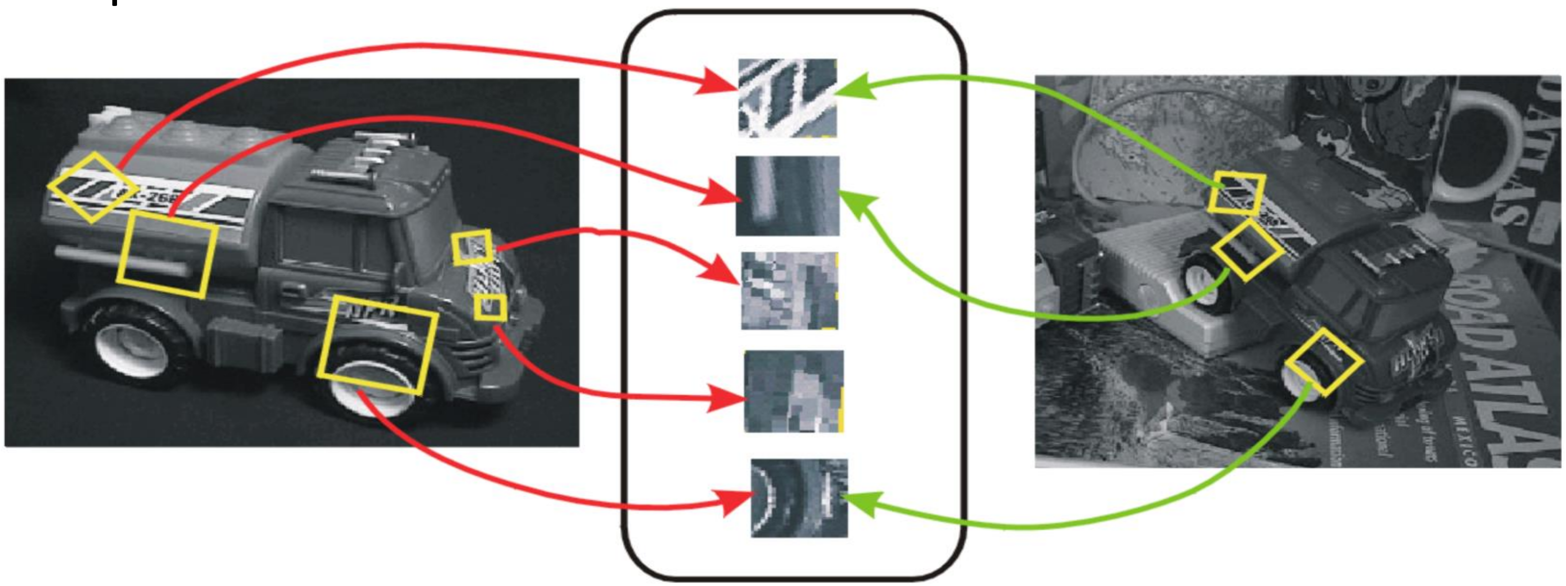
Dr. Niclas Zeller
Artisense GmbH

What We Will Cover Today

- Iterative closest points algorithm (leftover from last lecture)
- Introduction to object tracking
- Tracking-by-registration
- Multi-object tracking based on filtering

Recap: Object Detection with Local Features

- Can we make use of local features to detect a certain object in the scene?
  - Detect and match a set of local keypoints between model and scene (image)
  - Object detection is supposed to be invariant to different view points

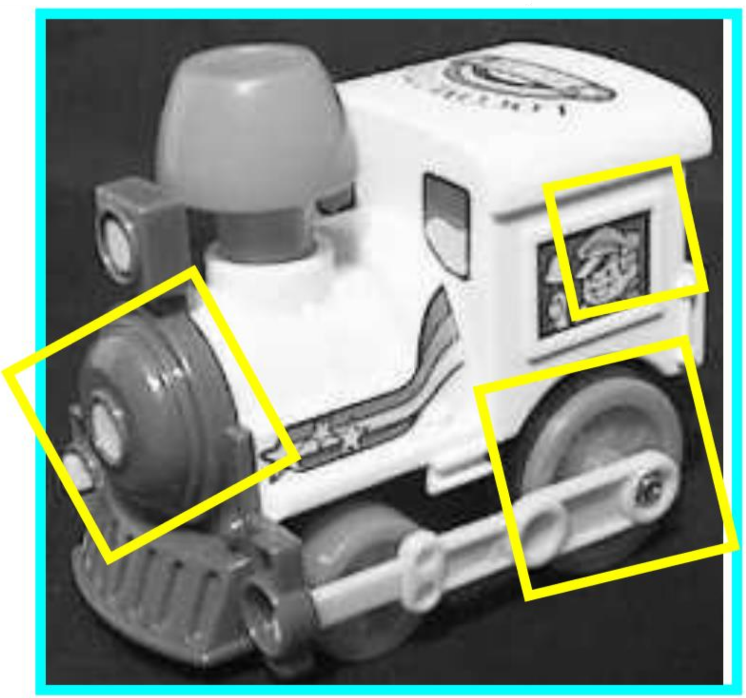


Keypoints e.g. SIFT

Image from D. Lowe

Recap: Hough Voting: 2D-to-2D Matching

- Oriented local 2D keypoint matches cast votes for affine transformations (f.e. 2D translation, scale & 2D rotation)

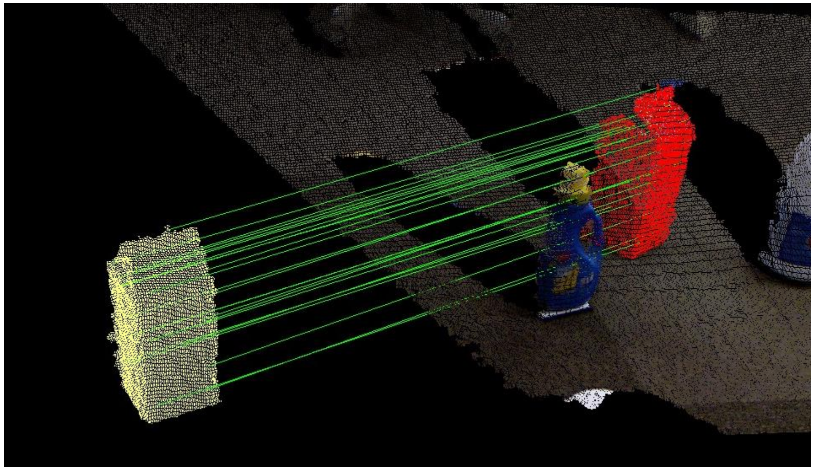


Slide adapted from S. Lazebnik

Recap: 3D Object Detection with Local Keypoints

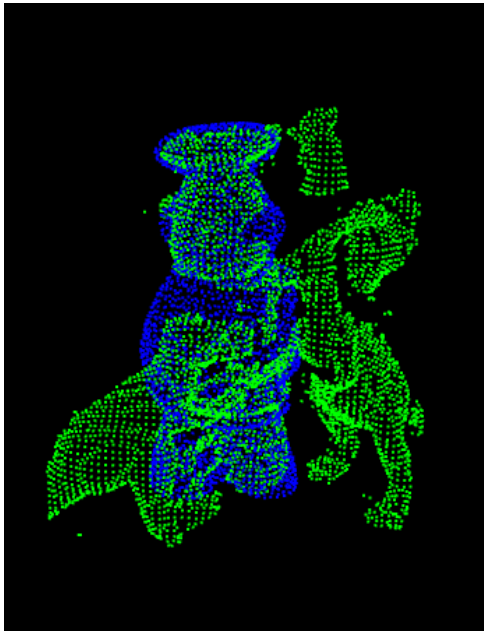
- Render views of 3D CAD models and extract keypoints for rendered views

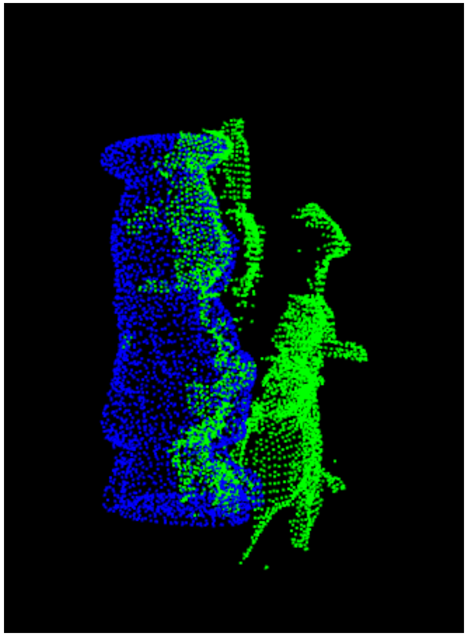
- Or Extract keypoints directly from 3D object models (f.e. CAD or scanned)
  - Rely only on geometry
  - Not on visual appearance



Pose Refinement

- So far, detection strategies provide only a coarse pose estimate
  - Based on keypoint associations (only subset of points)
- Popular strategy for pose refinement
  - Iterative Closest Points (ICP)
- Align scene measurements with model point cloud
  - Using all available points





Scene

Model

Images from pointclouds.org

Iterative Closest Points (ICP)

- Key Idea
  - If we knew the correspondences of points between scene and model, we could directly solve for the 3D-to-3D motion (rotation/translation) estimate

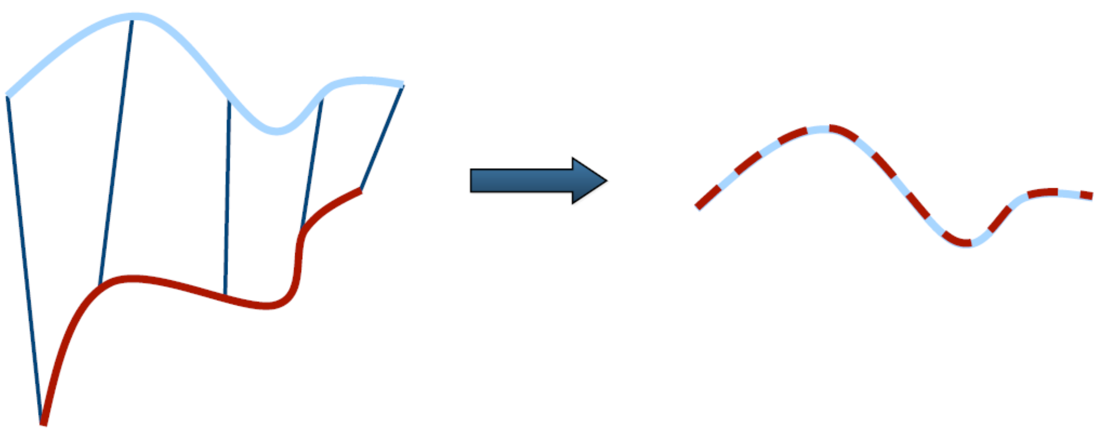
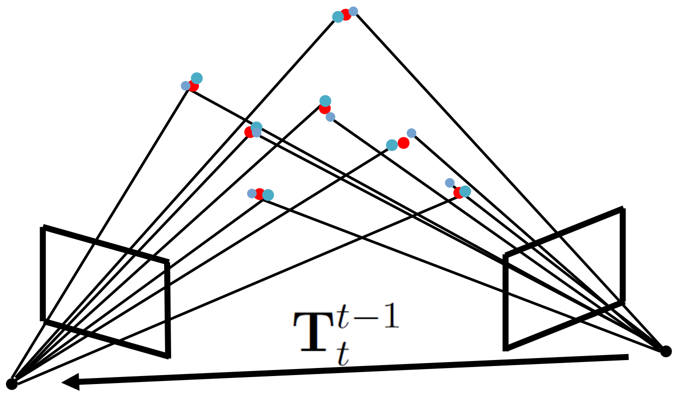


Image from Cyrill Stachniss

Recap: 3D-to-3D Motion Estimation

- Given corresponding 3D points in two camera frames
  $\mathcal{X}_{t-1} = \{\mathbf{x}_{t-1,1}, \ldots, \mathbf{x}_{t-1,N}\}$
  $\mathcal{X}_t = \{\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,N}\}$
  determine relative camera pose $\mathbf{T}_t^{t-1}$



- Idea: determine rigid transformation that aligns the 3D points
- Geometric least squares error: $E\left(\mathbf{T}_t^{t-1}\right) = \sum_{i=1}^{N} \left\|\overline{\mathbf{x}}_{t-1,i} - \mathbf{T}_t^{t-1}\overline{\mathbf{x}}_{t,i}\right\|_2^2$
- Closed-form solutions available, f.e. Arun et al., 1987
- Applicable e.g. to RGB-D cameras or also Lidar
  - Should only be used if we have very accurate depth

Recap: 3D Rigid-Body Motion from 3D-to-3D Matches

- Arun et al., Least-squares fitting of two 3-d point sets, IEEE PAMI, 1987
- Corresponding 3D points, $N \geq 3$

$$\mathcal{X}_{t-1} = \{\mathbf{x}_{t-1,1}, \ldots, \mathbf{x}_{t-1,N}\} \qquad \mathcal{X}_t = \{\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,N}\}$$

- Determine means of 3D point sets

$$\boldsymbol{\mu}_{t-1} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_{t-1,i} \qquad \boldsymbol{\mu}_t = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_{t,i}$$

- Determine rotation from

$$\mathbf{A} = \sum_{i=1}^{N} \left(\mathbf{x}_{t-1} - \boldsymbol{\mu}_{t-1}\right) \left(\mathbf{x}_t - \boldsymbol{\mu}_t\right)^\top \qquad \mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^\top \qquad \mathbf{R}_{t-1}^t = \mathbf{V}\mathbf{U}^\top$$

- Determine translation as $\mathbf{t}_{t-1}^t = \boldsymbol{\mu}_t - \mathbf{R}_{t-1}^t \boldsymbol{\mu}_{t-1}$

Iterative Closest Points (ICP)

- If the correct correspondences are not known, it is generally impossible to determine the optimal relative motion (rotation/translation) in one step

- Idea: Iteratively and alternatingly estimate correspondences and pose alignment between point sets $P = \{\mathbf{p}_i\}_{i=1}^{N}$ and $Q = \{\mathbf{q}_j\}_{j=1}^{M}$

$\text{argmax}_c \; p(P \mid Q, \boldsymbol{\xi}, c)$ $\qquad$ $\text{argmax}_{\boldsymbol{\xi}} \; p(P \mid Q, \boldsymbol{\xi}, c)$

Iterative Closest Points (ICP)

- Idea: Iteratively and alternatingly estimate correspondences and pose alignment between point sets $P = \{\mathbf{p}_i\}_{i=1}^{N}$ and $Q = \{\mathbf{q}_j\}_{j=1}^{M}$

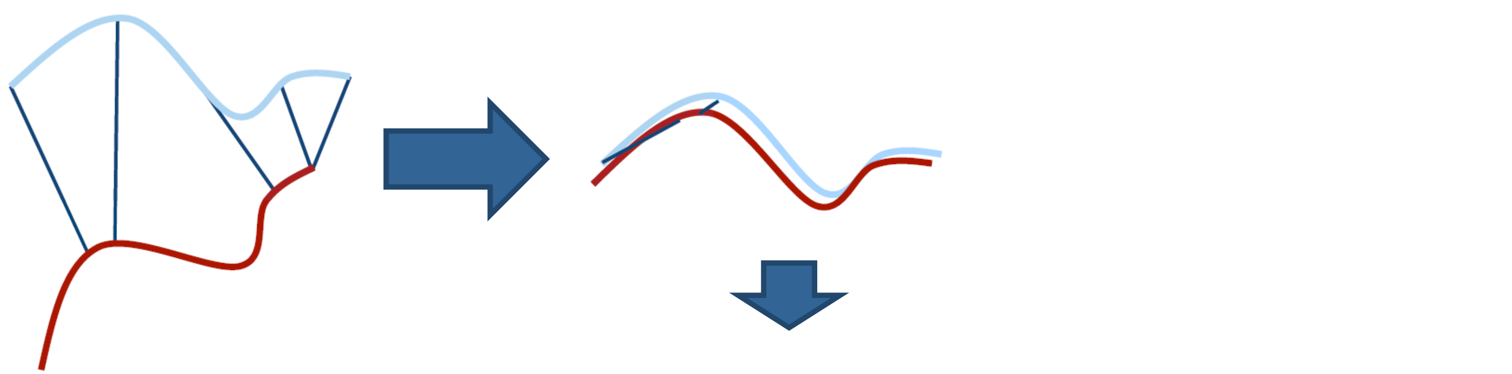
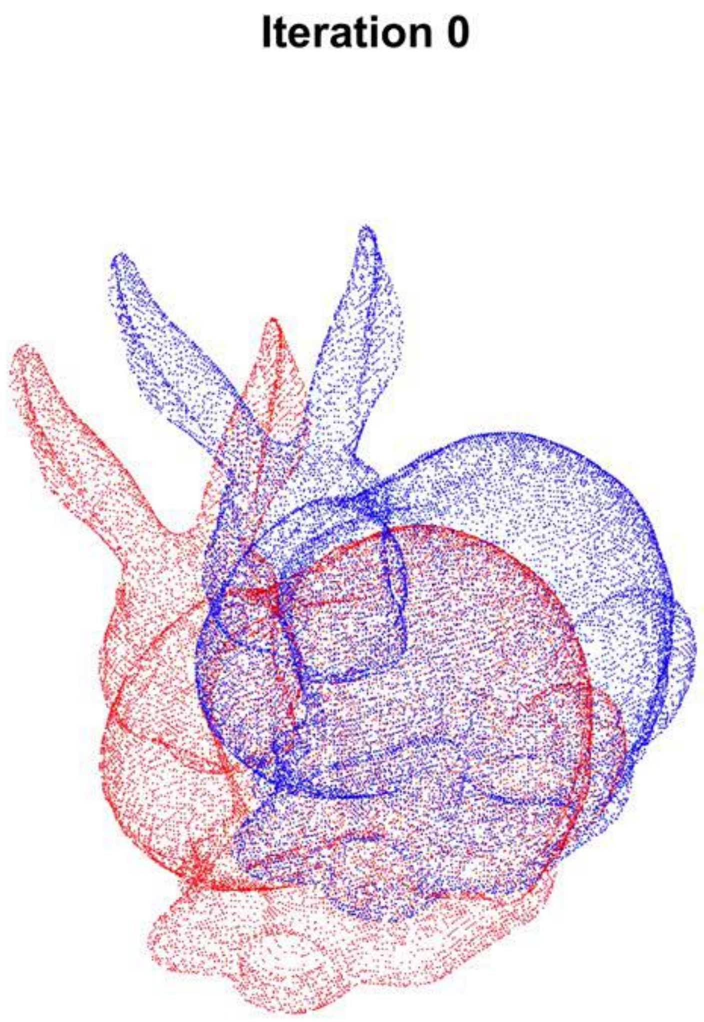


Image adapted from Cyrill Stachniss

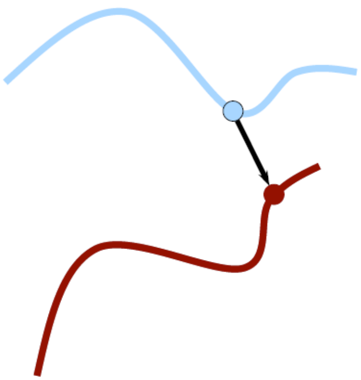
Keypoint Alignment and ICP Example



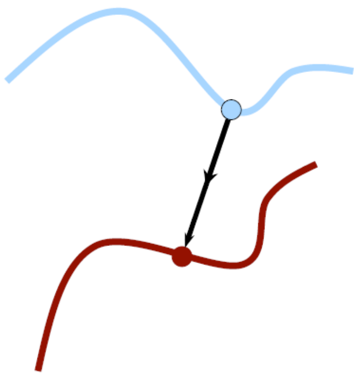
https://www.youtube.com/watch?v=uzOCS_gdZuM

Data Association for ICP

- Closest-points matching



- Normal shooting
  - Requires normal calculation
  - Better convergence than closest-point for smooth structures



Images from Cyrill Stachniss

Projective Data Association

- For aligning depth or point measurements from a sensor, we can use projective data association
- Warping of measured 3D point
- Analogous association as in direct image alignment!

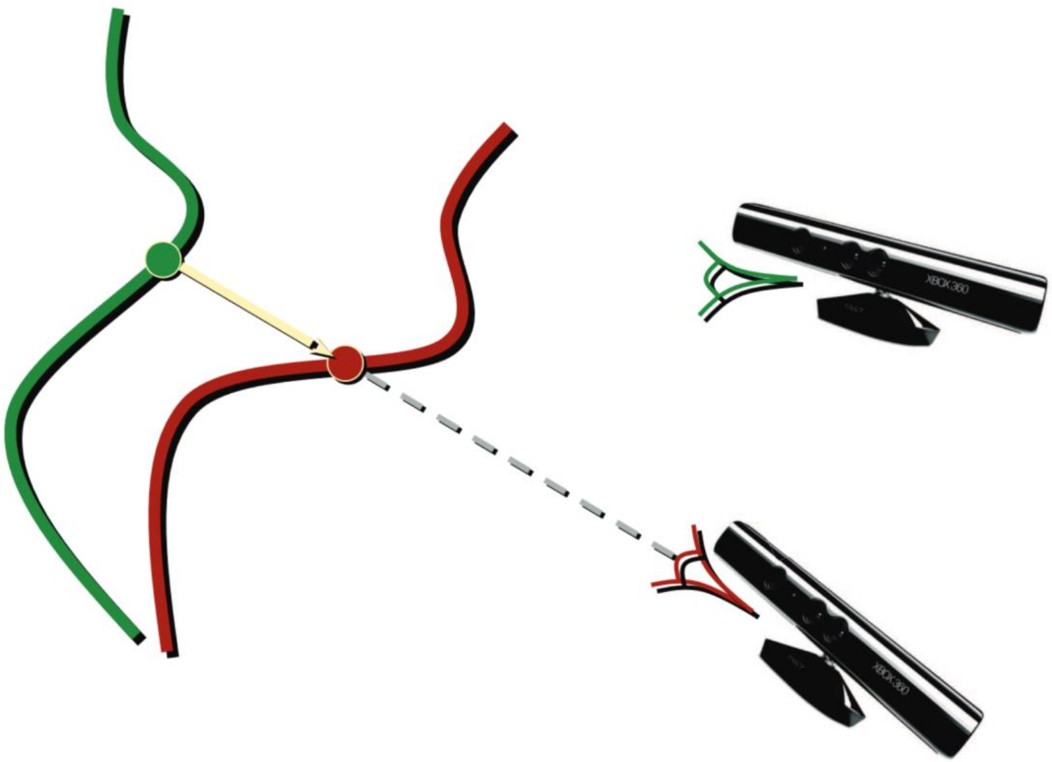
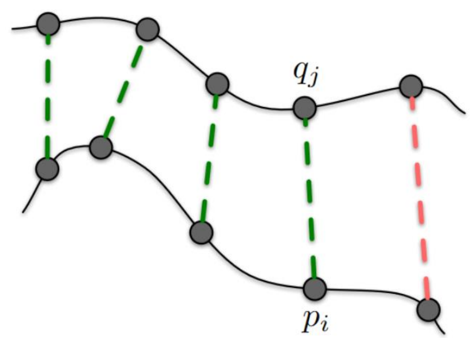


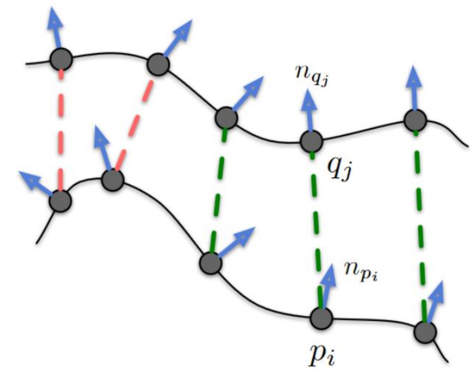
Image from R. Newcombe 2013

Outlier Rejection for ICP

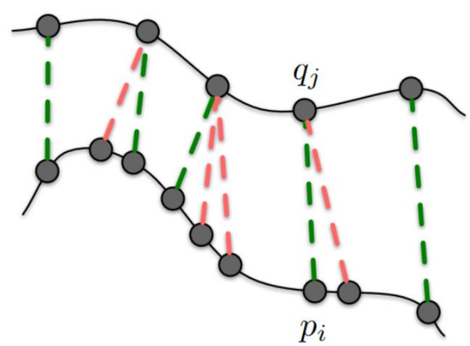
- Optionally perform outlier rejection



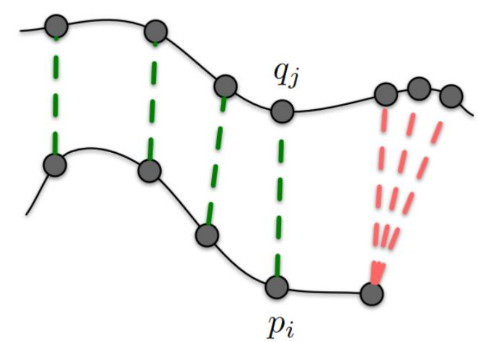
(a) Rejection based on the distance between the points.



(b) Rejection based on normal compatibility.



(c) Rejection of pairs with duplicate target matches.



(d) Rejection of pairs that contain boundary points.

Images from Holz et al., 2015

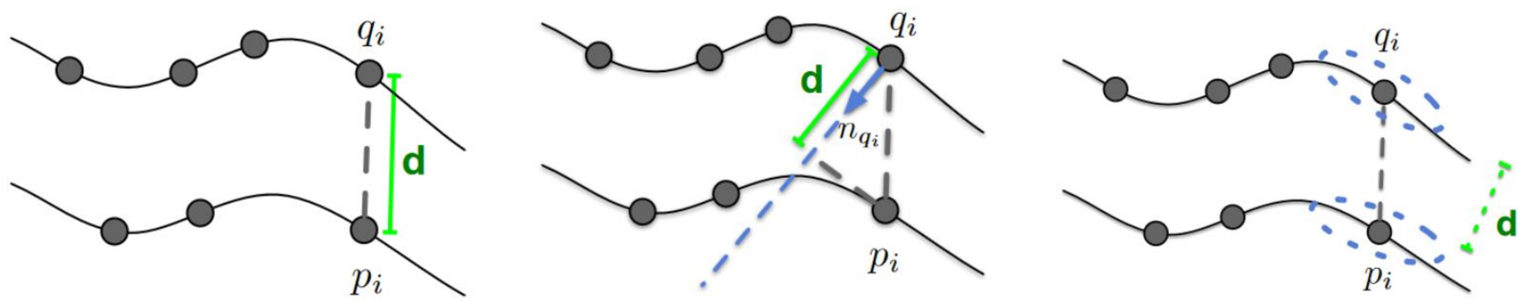
ICP Alignment Objectives

- Alignment objectives: point-point, point-plane, GICP

$$E_{\text{point-to-point}}(\boldsymbol{T}) = \sum_{k=1}^{N} w_k ||\boldsymbol{T}\, \boldsymbol{p}_k - \boldsymbol{q}_k||^2, \text{ and}$$

$$E_{\text{point-to-plane}}(\boldsymbol{T}) = \sum_{k=1}^{N} w_k \left((\boldsymbol{T}\, \boldsymbol{p}_k - \boldsymbol{q}_k) \cdot \boldsymbol{n}_{\boldsymbol{q}_k}\right)^2$$

$$E_{\text{Generalized-ICP}}(\boldsymbol{T}) = \sum_{k=1}^{N} \boldsymbol{d}_k^{(\boldsymbol{T})^T} \left(\boldsymbol{\Sigma}_k^Q + \boldsymbol{T}\boldsymbol{\Sigma}_k^P \boldsymbol{T}^T\right)^{-1} \boldsymbol{d}_k^{(\boldsymbol{T})}$$

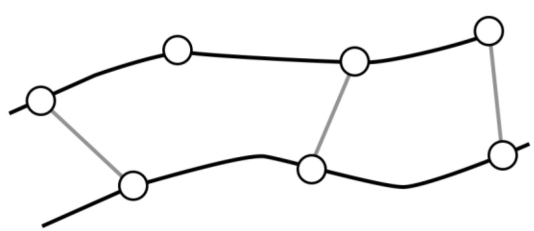


(a) Point to point error (b) Point to plane error (c) Generalized-ICP

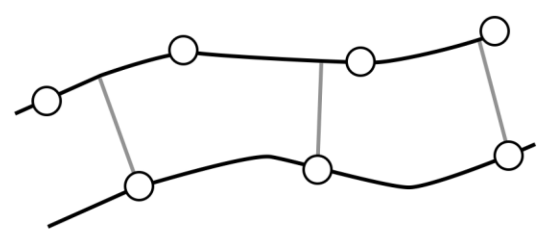
Images from Holz et al., 2015

ICP Alignment Objectives

- Point-to-Point vs. Point-to-plane
  - Requires normal calculation for one of the point clouds
  - Each iteration is generally slower than point-to-point version
  - However, often significantly better convergence rate
  - Using point-to-plane distance instead of point-to-point lets flat regions slide along each other



point-to-point



point-to-plane

Images from Cyrill Stachniss

ICP Alignment Objectives

- Generalized ICP
  - Probabilistic modelling of point clouds
  - Where to get covariance matrices from
    - directly available from sensor measurements
    - Can be estimated from point distribution
    - Covariance matrices need to be calculated for both point clouds

What is Object Tracking?

- Goal
  - *Estimate the number and state of objects in a region of interest*
- State
  - We are using the term state to describe a vector of quantities that characterize the object being tracked.

E.g.	[x, y]	(location)
	[x, y, dx, dy]	(location + velocity)

  - Because observations are typically noisy, estimating the state vector is a statistical estimation problem.

Slide adapted from Robert Collins

What is Object Tracking?

- Goal
  - *Estimate the number and state of objects in a region of interest*

- Variety of objects to track (e.g. persons, cars)
- 3D tracking: Tracking 3D location of an object
  - W.r.t. camera frame or world frame (requires ego-motion compensation)
- Articulated tracking: e.g. tracking body pose

Image sources: Kristen Grauman, Michael Breitenstein, Ahmed Elgammal

Types of Tracking

- Single-object tracking
  - Focuses on tracking a single target in isolation.

[Z. Kalal, K. Mikolajczyk, J. Matas, PAMI'10]

Types of Tracking

- Multi-object tracking

ECCV 2018
European Conference on Computer Vision
8 – 14 September 2018 | Munich, Germany

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Stereo Vision–based Semantic 3D Object and Ego–motion Tracking for Autonomous Driving

Peiliang Li, Tong Qin and Shaojie Shen | HKUST UAV Group

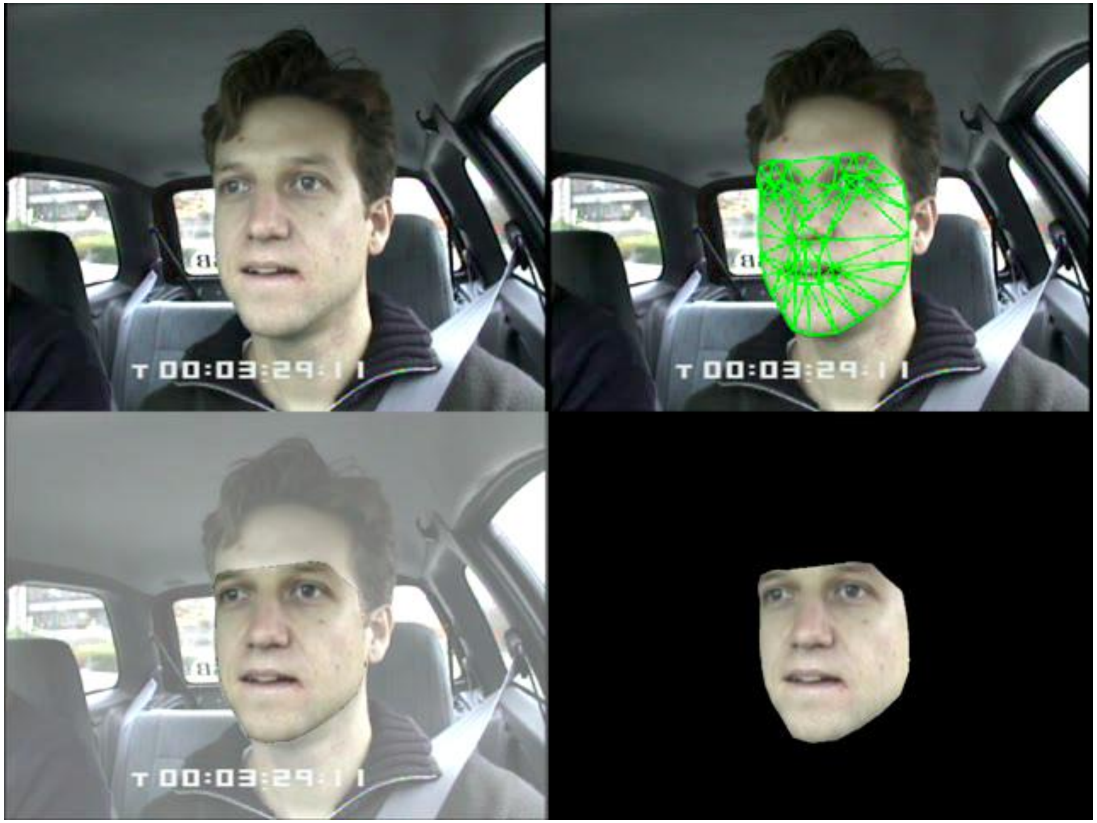
http://uav.ust.hk/

(Li, Qin, Shen, ECCV 2018)

https://www.youtube.com/watch?v=nE2XtCvPEDk

Types of Tracking

- Articulated tracking
- Tries to estimate the motion of objects with multiple, coordinated parts



[I. Matthews, S. Baker, IJCV'04]

Slide credit: Robert Collins

Types of Tracking

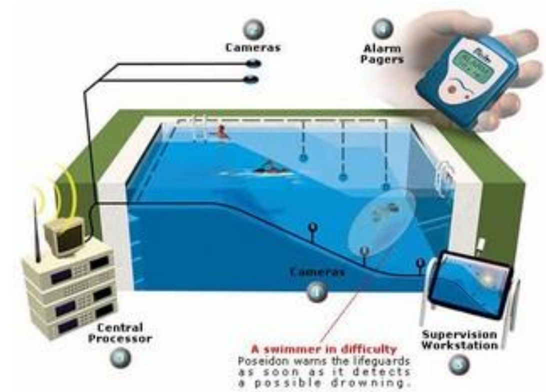
- Active tracking
  - Involves moving the sensor in response to motion of the target. Needs to be real-time!
    - Due to control feed-back, latency is quite important

Slide credit: Robert Collins

Applications: Safety & Security

Autonomous robots

Driver assistance



Monitoring pools
(Poseidon)

Pedestrian detection
[MERL, Viola et al.]

Surveillance

Slide credit: Kristen Grauman

Applications: Human-Computer Interaction

Games
(Microsoft Kinect)

Assistive technology systems
Camera Mouse
(Boston College)

Slide adapted from Kristen Grauman

Applications: Visual Effects

MoCap for *Pirates of the Carribean*, Industrial Light and Magic

Slide adapted from Steve Seitz, Svetlana Lazebnik, Kristen Grauman

Factors: Distinguishability

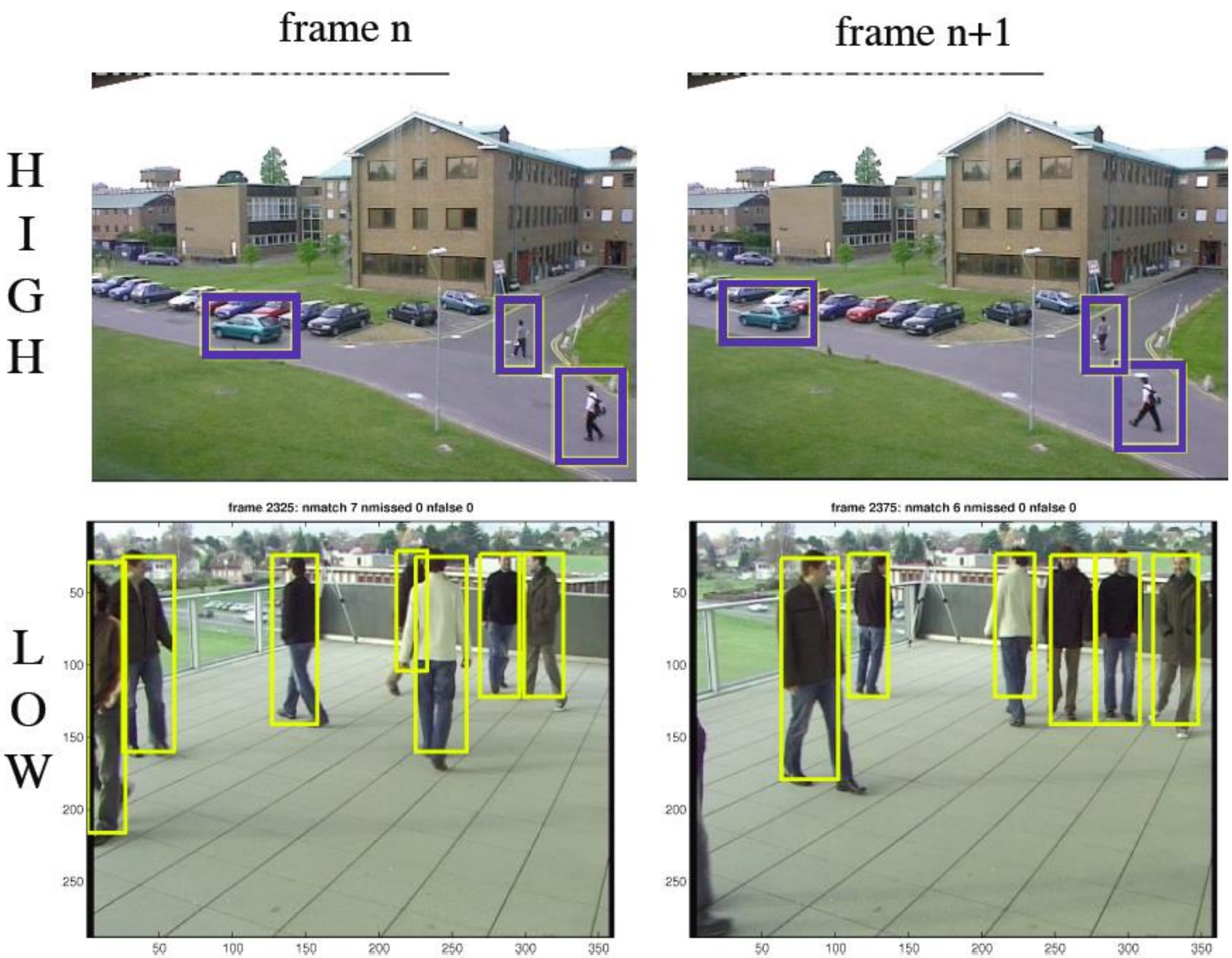
- How easy is it to distinguish one object from another?

Appearance models can
do all the work

Constraints on geometry
and motion become crucial

Slide credit: Robert Collins

Factors: Frame Rate



Using state prediction for data association might be sufficient

Much harder search problem. Good data association becomes crucial.

Slide credit: Robert Collins

Other Factors

- Single target *vs.* multiple targets
- Single camera *vs.* multiple cameras
- On-line *vs.* batch mode
- Do we have a good generic detector? (e.g., faces, pedestrians)
- Does the object have multiple parts?
- ...

Slide credit: Robert Collins

Elements of Tracking

t=1 t=2 … t=20 t=21

- Detection
  - Find the object(s) of interest in the image.

Image credit: Kristen Grauman

Elements of Tracking

t=1 t=2 t=20 t=21

- Detection
  - Find the object(s) of interest in the image.
- Association
  - Determine which observations come from the same object.

Image credit: Kristen Grauman

Elements of Tracking

t=1 t=2 ... t=20 t=21

- Detection
  - Find the object(s) of interest in the image.
- Association
  - Determine which observations come from the same object.
- Prediction
  - Predict future motion based on the observed motion pattern.
  - Use this prediction to improve detection and data association in later frames.

Image credit: Kristen Grauman

3D Object Tracking Approaches

- Strategy 1: Tracking-by-detection
  - Detect object in each frame individually
- Strategy 3: Tracking-by-registration
  - From an initial guess (detection) perform incremental registration
- Strategy 2: Tracking-by-filtering
  - Detect object as measurement within probabilistic filter

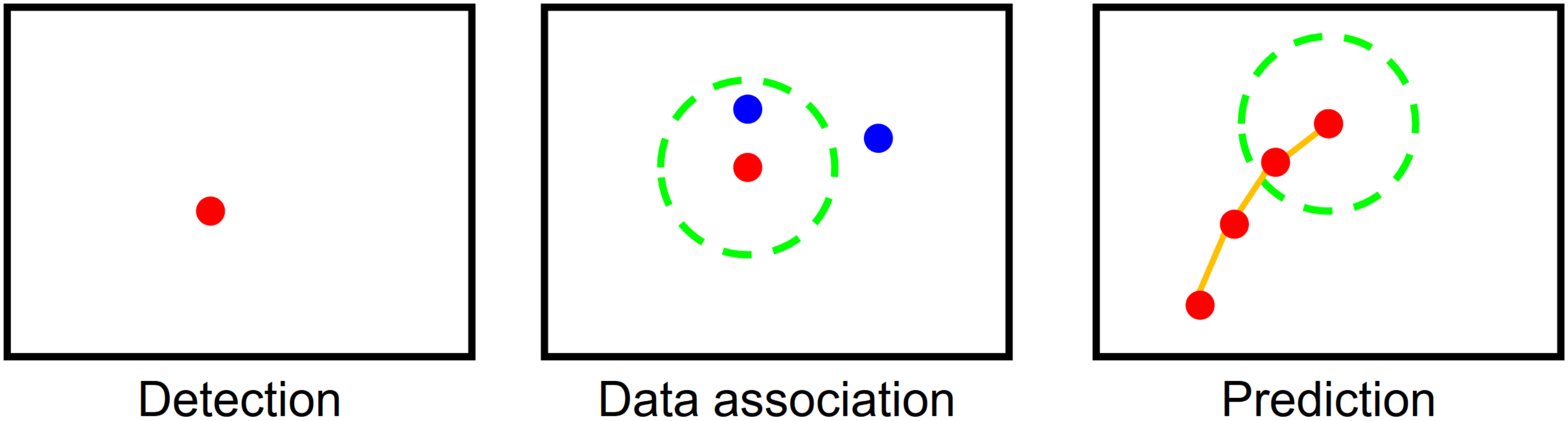
Tracking-by-Registration

- Consider the following approach:
  - Describe object as a set of points $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ in its reference frame
  - Optimize for the pose $\boldsymbol{\xi} \in se(3)$ that aligns object points with measurements $\mathcal{Y} = \{\mathbf{y}_j\}_{j=1}^{M}$ at each time step

$$E(\boldsymbol{\xi}) = \frac{1}{2} \sum_{(i,j)\in\mathcal{C}} \|\mathbf{x}_i - \mathbf{y}_j\|_2^2$$

  - Non-linear least squares…
- However this requires to decide
  - which scene points belong to the object (segmentation)
  - which object and scene points correspond to each other
- Could be solved using an ICP-like approach

Elements of Tracking

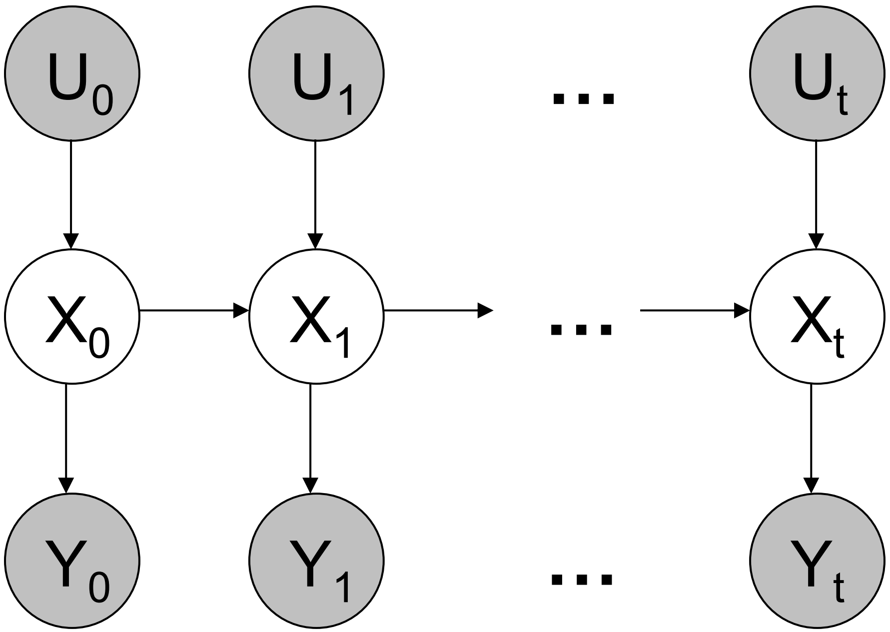


- Detection: Where are candidate objects?
- Data association: Which detections belong to the same object?
- Prediction: Where will a tracked object be in the next time step?

Slide credit: Bastian Leibe

Recap: Probabilistic Model of Time-Sequential Processes

- Hidden state X gives rise to noisy observations Y
- At each time t,
  - the state changes stochastically from X_{t-1} to X_t
  - state change depends on action U_t
  - we get a new observation Y_t



Recap: Markov Assumptions

- Only the immediate past matters for a state transition

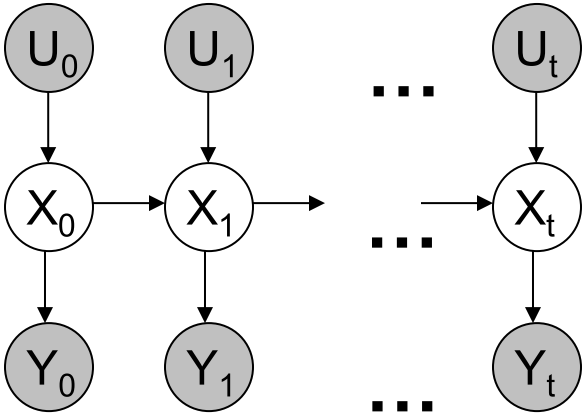
$$p\left(X_t \middle| X_{0:t-1}, U_{0:t}\right) = \boxed{p\left(X_t \middle| X_{t-1}, U_t\right)}$$

state transition model

- Observations depend only on the current state

$$p\left(Y_t \middle| X_{0:t}, U_{0:t}, Y_{0:t-1}\right) = \boxed{p\left(Y_t \middle| X_t\right)}$$

observation model



Recap: Predict-Correct Cycle

- Prediction:

$$p\left(X_t \mid y_{0:t-1}, u_{0:t}\right) = \int p\left(X_t \mid X_{t-1}, u_t\right) p\left(X_{t-1} \mid y_{0:t-1}, u_{0:t-1}\right) dX_{t-1}$$

observation

y_t

action

u_t

- Correction:

$$p\left(X_t \mid y_0, \ldots, y_t\right) = \frac{p\left(y_t \mid X_t\right) p\left(X_t \mid y_{0:t-1}, u_{0:t}\right)}{\int p\left(y_t \mid X_t\right) p\left(X_t \mid y_{0:t-1}, u_{0:t}\right) dX_t}$$

Multi-Object Tracking by Filtering

- Approach: probabilistic filtering of position, velocity, etc. of each object track (state) $\mathbf{x}$ based on measurements

$$\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_M\}$$

  - One filter per object

- Data association before correction step

- Unassociated measurements create new tracks

- Discard tracks that cannot be associated to measurements

Recap: Extended Kalman Filter (EKF)

- Non-linear state-transition model with Gaussian noise:

$$\mathbf{x}_t = g(\mathbf{x}_{t-1}, \mathbf{u}_t) + \boldsymbol{\epsilon}_t \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{d_t})$$

- Non-linear observation model with Gaussian noise:

$$\mathbf{y}_t = h(\mathbf{x}_t) + \boldsymbol{\delta}_t \qquad \boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{m_t})$$

- How to cope with non-linear system?

- Idea: linearize the models in each time step

$$\Rightarrow \quad \mathbf{x}_t \approx g(\mathbf{x}_{t-1}^0, \mathbf{u}_t) + \nabla g(\mathbf{x}, \mathbf{u}_t)|_{\mathbf{x}=\mathbf{x}_{t-1}^0} \left(\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^0\right) + \boldsymbol{\epsilon}_t$$

$$\Rightarrow \quad \mathbf{y}_t \approx h(\mathbf{x}_t^0) + \nabla h(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_t^0} \left(\mathbf{x}_t - \mathbf{x}_t^0\right) + \boldsymbol{\delta}_t$$

Recap: EKF Prediction & Correction

- Efficient approximate correction and prediction steps which involve manipulation of Gaussians and linearization
- The state estimate can be represented as a Gaussian distribution

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

- Prediction: $\boldsymbol{\mu}_t^- = g(\boldsymbol{\mu}_{t-1}^+, \mathbf{u}_t)$

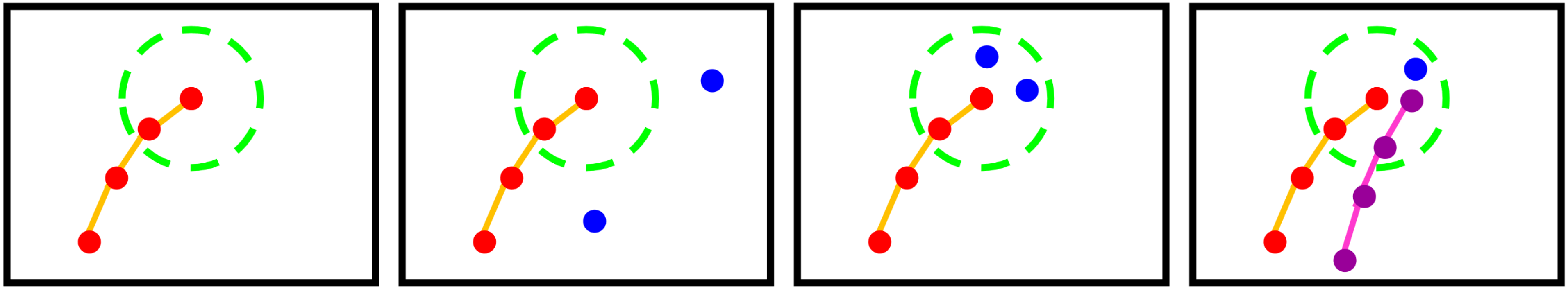
$$\boldsymbol{\Sigma}_t^- = \mathbf{G}_t \boldsymbol{\Sigma}_{t-1}^+ \mathbf{G}_t^\top + \boldsymbol{\Sigma}_{d_t} \qquad \mathbf{G}_t := \nabla g(\mathbf{x}, \mathbf{u}_t)\big|_{\mathbf{x}=\boldsymbol{\mu}_{t-1}^+}$$

- Correction: $\mathbf{K}_t = \boldsymbol{\Sigma}_t^- \mathbf{H}_t^\top \left(\mathbf{H}_t \boldsymbol{\Sigma}_t^- \mathbf{H}_t^\top + \boldsymbol{\Sigma}_{m_t}\right)^{-1}$

$$\boldsymbol{\mu}_t^+ = \boldsymbol{\mu}_t^- + \mathbf{K}_t \left(\mathbf{y}_t - h(\boldsymbol{\mu}_t^-)\right) \qquad \mathbf{H}_t := \nabla h(\mathbf{x})\big|_{\mathbf{x}=\boldsymbol{\mu}_t^-}$$

$$\boldsymbol{\Sigma}_t^+ = \left(\mathbf{I} - \mathbf{K}_t \mathbf{H}_t\right) \boldsymbol{\Sigma}_t^-$$

What Makes Multi-Object Tracking Difficult?



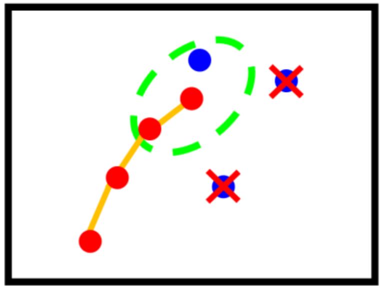
- Predictions may not be supported by detections
  - Occlusion or end of track?
- Unexpected measurements
  - New objects or outliers?
- Correspondence ambiguity for a prediction
  - Which measurement is the correct one?
- Correspondence ambiguity for a measurement
  - Which object track shall the measurement belong to?

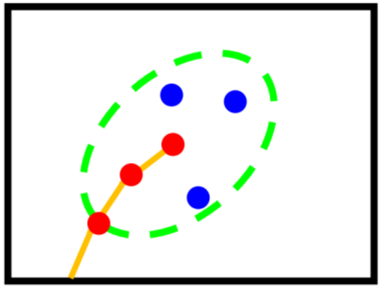
Slide credit: Bastian Leibe

Gating Nearest Neighbor Data Association

- Gating
  - Only consider measurements within a certain area around the predicted location

  ⇒ Large gain in efficiency, since only a small region needs to be searched





- Nearest Neighbor Association
  - Among the candidates in the gating region, only take the one closest to the prediction

Slide adapted from Bastian Leibe

Gating with Mahalanobis Distance

- Recall: Kalman Filter
  - Maintains a Gaussian state estimate $\boldsymbol{\mu}_t,\ \boldsymbol{\Sigma}_t$

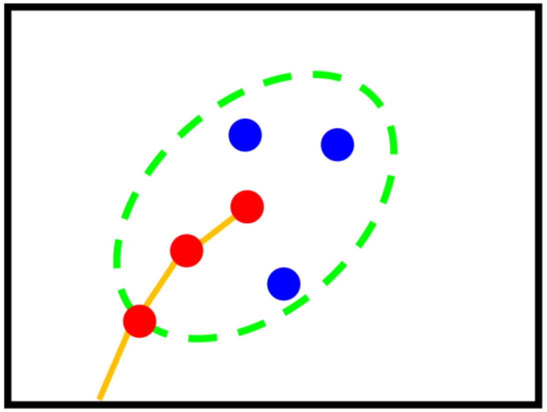
- Perform gating based on the distribution of prediction and measurement

$$\mathcal{N}(h(\boldsymbol{\mu}_t^-), \boldsymbol{\Sigma}_{mt} + \mathbf{H}_t \boldsymbol{\Sigma}_t^- \mathbf{H}_t^T)$$

- Mahalanobis Distance

$$d^2 = (\boldsymbol{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{\mu})$$

$$\boldsymbol{\mu} = h(\boldsymbol{\mu}_t^-) \qquad \boldsymbol{\Sigma} = \boldsymbol{\Sigma}_{mt} + \mathbf{H}_t \boldsymbol{\Sigma}_t^- \mathbf{H}_t^T$$



- Gating volume is ellipsoidal
- E.g. choose volume that corresponds to 95% of probability mass
  - d^2 is χ^2-distributed → look up threshold from table

Slide adapted from Bastian Leibe

Problems with NN Assignment

- Limitations
  - For NN assignments, there is always a finite chance that the association is incorrect, which can lead to serious effects

  $\Rightarrow$ If a Kalman filter is used, a falsely assigned measurement may lead the filter to lose track of its target

  - The NN filter makes assignment decisions only based on the current frame
  - More information is available by examining subsequent images

  $\Rightarrow$ Data association decisions could be postponed until a future frame will resolve the ambiguity

- More powerful approaches
  - Multi-Hypothesis Tracking (MHT)
    - Well-suited for KF, EKF approaches
  - Particle filter based approaches

Slide adapted from Bastian Leibe

Lessons Learned Today

- Object tracking involves detection, motion estimation (prediction) and data association over time
- 3D object tracking of an object model through registration
  - ICP-based tracking-by-registration
- Multi-object tracking involves a harder data association problem
  - Gated Nearest Neighbor filter

Thanks for your attention!

Slides Information

- These slides have been initially created by Jörg Stückler as part of the lecture “Robotic 3D Vision” in winter term 2017/18 at Technical University of Munich.
- The slides have been revised by myself (Niclas Zeller) for the same lecture held in winter term 2020/21
- Acknowledgement of all people that contributed images or video material has been tried (please kindly inform me if such an acknowledgement is missing so it can be added).