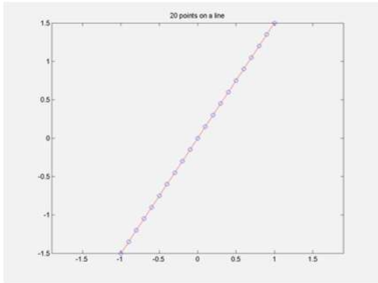
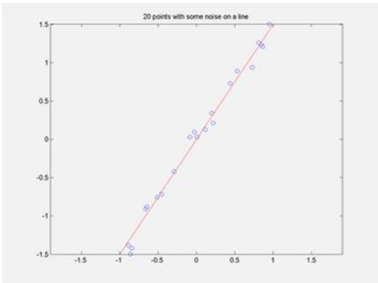
Solving the Correspondence Problem with RANSAC

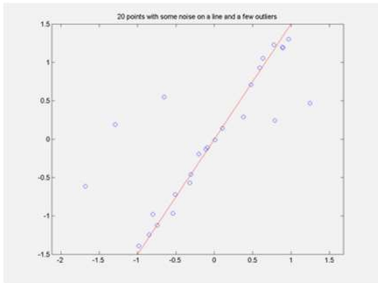
Example: estimation of a line from points



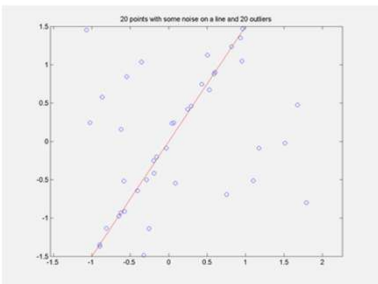
Example: estimation of a line from points



Example: estimation of a line from points



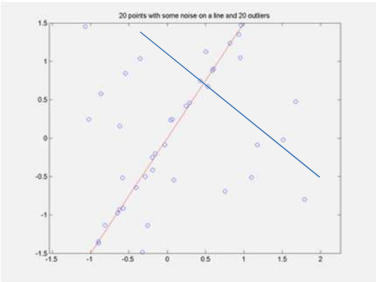
Example: estimation of a line from points



Observations

- We need (in this case!) a minimum of 2 points to determine a line
- Given such a line $\mathbf{l}$, we can determine how well any other point $\mathbf{y}$ fits the line $\mathbf{l}$
  - For example: distance between $\mathbf{y}$ and $\mathbf{l}$
- If we pick 2 *arbitrary* points from the dataset:
  - We can easily determine a line $\mathbf{l}$
  - $\mathbf{l}$ is the correct line with some probability p_{LINE}
  - p_{LINE} is related to the chance of picking only inliers
  - p_{LINE} is larger the fewer points that are used to determine $\mathbf{l}$
  - In general: if $\mathbf{l}$ is the correct line there are more additional points that can be fitted to the line than if $\mathbf{l}$ is an incorrect line

Line estimated from 2 inliers



Basic iteration

1. Pick 2 random points
2. Fit a line $\mathbf{l}$ to the points
3. Determine how many other points in the dataset that can be fitted to $\mathbf{l}$ with some minimal error ϵ.
   - This forms the *consensus set C*
4. If *C* is sufficiently large, then the fitted line is probably OK. Keep it

Basic algorithm

- Iterate *r* times
  1. Pick 2 random points
  2. Fit a line $\mathbf{l}$ to the points
  3. Form the consensus set *C*, together with
     - Number of points in *C*
     - Matching error ϵ_C of the set *C* relative to the line
  4. If the consensus set is sufficiently large, then the fitted line is OK. In particular if *N* and/or ϵ_C is better than the last line that was OK. Then keep it.
- For each iteration, we increase $p_{SUCCESS}$ = the probability that the correct line has been determined
  – We need to iterate sufficiently many time to raise $p_{SUCCESS}$ to a useful level

RANSAC

- This algorithm is called RANSAC
  – RANdom SAmple Consensus
- Published by Fischler & Bolles in 1981
  – "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". *Comm. of the ACM* **24**: 381–395.
- Several extensions / variations in the literature
  – Preemptive RANSAC
  – PROSAC
  – …

RANSAC

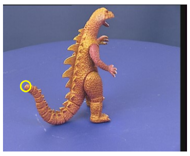
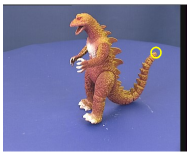
- After r iterations, RANSAC finds a reasonable estimate of the line (i.e. from only inliers) with a probability of *p*
- $1 - p =$
  P(pick at least one outlier in each iteration) = $(1-w^2)^r$
- $p = 1 - (1-w^2)^r$
- If *w* is known, we can choose *r* to make *p* as large as we want (but not = 1!)
- Example: $w = 0.5$
  - $p = 0.94$ when $r = 10$, $p = 0.99$ when $r = 20$

The correspondence problem

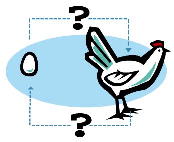
- Given a set of interest points in two images, we want to determine correspondences, i.e., pairs of points that correspond to the same 3D point
- If there is a small relative baseline:
  – Use tracking (Lucas-Kanade, etc)
    - Track POIs in image 1 to their corresponding positions in image 2
    - Can be applied to parts an image sequence
    - A POI typically disappears after a while in a longer sequence
  – Track-retrack
    - Remove all POIs that cannot be tracked forward and backward in time over several images

The correspondence problem

- If there are large baseline between the two images, tracking performance degrades
  - Another approach is needed

A chicken and egg problem

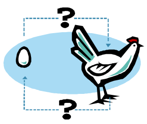


Point correspondences can be determined if we know **F**

We need corresponding points to estimate **F**

Can we determine **F** and correspondences at the same time?

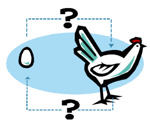
Solving chicken and egg problem?



- Let there be two views with P_1 points in one view and P_2 points in the other view
- We don't know which points in the first view that correspond to which points in the other view
- There is a set *D* of $P_1 \times P_2$ possible correspondences, or *tentative* correspondences

Chicken and egg revisited

- The correct correspondences can be fitted to **F**, i.e., they satisfy the epipolar constraint for some **F** that only depends on which two views are used
- They are the *inliers*
- The incorrect correspondences are *outliers*



Use RANSAC

- Pick 8 random points from *D*
- We don't know if they really correspond, but this can be tested:
  1. Use the 8-point algorithm to estimate **F**
  2. Check how well **F** matches each pair in *D*
  3. Collect those that fit well into the consensus set *C*
  4. If *C* is sufficiently large: **F** is OK: keep **F** and *C*
- Iterate *r* times

Probabilities

- Let w be the fraction of inliers in *D*
- In each iteration we pick N points that are all inliers with probability w^N (approximately)
- The probability of not all N points are inliers is then given by $1 - w^N$
- The probability of not all N points are inliers in r iterations is $(1 - w^N)^r$
- The probability that in K iteration, at least once, all N points are inliers: $p = 1 - (1 - w^N)^r$
- Solve for r: $r = \frac{\log(1-p)}{\log(1-w^N)}$

General observation

- The expected number of iterations, r, to reach a certain probability p is

$$r = \frac{\log(1-p)}{\log(1-w^N)}$$

- For fixed p, r is reduced if w is made larger
- For fixed p, r is reduced if N is made smaller

The odds are against us

- From the outset, the set of all tentative correspondences between two images can be VERY large (= $P_1 \times P_2$)
- VERY few of these are inliers: w is VERY small
- Here $N = 8$
- This means that r must be VERY^8 large in order to make p close to 1

The correspondence problem

- The correspondence problem is often addressed by finding two sets of points that we want to bring into correspondence
  - Typically: interest points in images (POI)
  - Typically: different number of points in the sets
- Without any outer information:
  - Any point in set 1 can correspond to any point in set 2
  - In practice, often not a feasible approach!
  - Too many outliers (w too small)

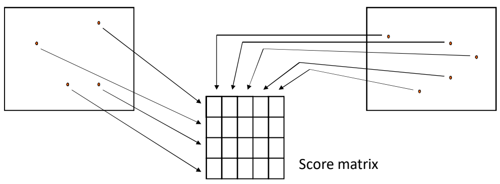
Visual appearance and RANSAC

- The set of correspondences in D has m possible correspondences and only m_0 of them are correct ($m - m_0$ are incorrect)
- Probability of picking a correct correspondence $w = m_0/m$
- If we can reduce the number incorrect correspondences, without removing correct ones, m will decrease while m_0 is constant
  $\Rightarrow$ w increases $\Rightarrow$ r decreases for fixed p

Matching matrix

- Given P_1 points in image 1 and P_2 points in image 2
  - Form a $P_1 \times P_2$ *matching matrix*
  - Each entry (i,j) is a hypothetical correspondence between point i in image 1 and point j in image 2
- Set entry (i,j) =
  a matching score between point i and point j
- For each column or row: keep only the largest entry
  - Reduces m while keeping m_0 constant
  - w increases $\Rightarrow$ r decreases for fixed p
- Run RANSAC on remaining tentative pairs

Matching matrix



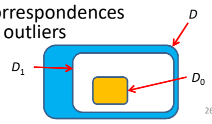
Each entry in the matching matrix describes how well a certain point in image 1 matches another point in image 2. For example: high score = good match

Matching matrix

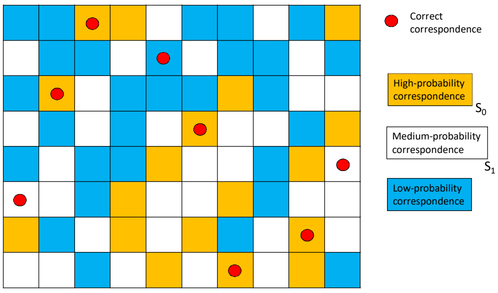
- The matching score can be based on *similarity of visual appearance* or other a priori knowledge about the scene (rather than geometric properties)
- For example
  - SIFT features [see previous lecture!]
  - MSER [see previous lecture!]
  - Color description
  - Camera motions in relation to scene depth
  - Tracking quality
- The resulting correspondences are referred to as
  - *Tentative* correspondences
  - *Putative* correspondences

Matching matrix

- Threshold the matching scores to remove high-probability outliers and to identify high-probability inliers (two thresholds!)
  - Remove high-probability outliers
  - High probability inliers means > 50% probability
- From the original set D of possible correspondences, we have form two sets D_1 and D_0 such that
  - D_0 contains the high-probability inliers
    - A.k.a. *putative correspondences*
  - D_1 contains the remaining correspondences that are not high-probability outliers
  - $D_0 \subset D_1 \subset D$



Matching matrix



Visual appearance and RANSAC

- Remove the low-probability correspondences before RANSAC
- Use the RANSAC algorithm for finding corresponding points based on the tentative correspondences
  - Use only high-probability inliers (D_0) in the initial selection of n points: $w > 0.5$
    $\Rightarrow$ **fewer iterations are needed**
  - Use medium and high-probability correspondences (D_1) to form the consensus step
    $\Rightarrow$ **increases the probability of including correct correspondences in the consensus set**

E vs. **F**

- If we estimate **F** in each RANSAC iteration, then we need $N = 8$ correspondences to determine **F**
- If instead **E** is determined, it is sufficient with $N = 5$ correspondences
  - In practice 6, since we get multiple solutions for **E**
- If the internal calibration **K** is known, we can reduce r = number of RANSAC iterations, by using **E** instead of **F**