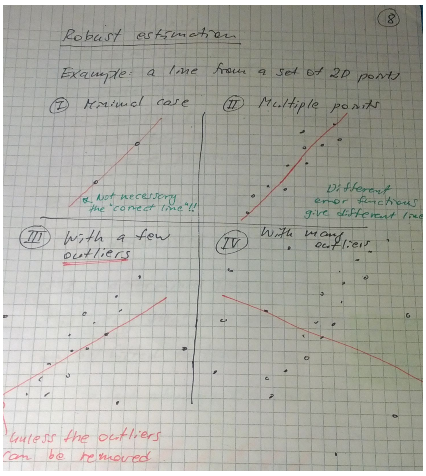
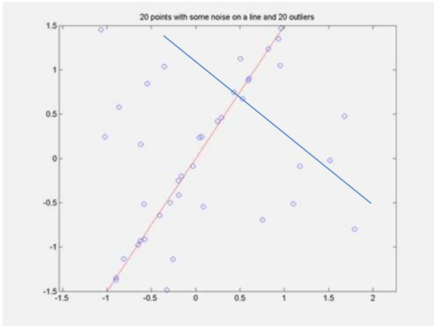


Line estimated from 2 inliers



Observations

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

Basic iteration

1. Pick 2 random points
2. Fit a line l to the points
3. Determine how many other points in the dataset that can be fitted to 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 K 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

- An undeterministic algorithm
- Finds a line estimated from only inliers with a probability p after K iterations

$1 - p$ = Pr(pick at least one outlier every time)
$= (1 - w^2)^K$
$p = 1 - (1 - w^2)^K$

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

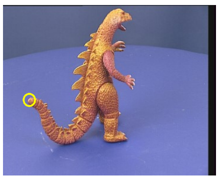
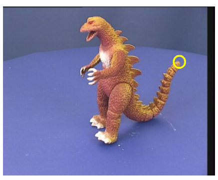
- If w is known, we can choose the number of iterations, K, to make p reasonably high
- Example
  $w = 0.5$
  $p \approx 0.94$ for $K = 10$
  $p \approx 0.99$ for $K = 20$

The correspondence problem

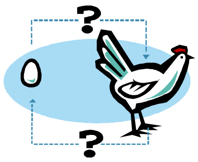
- Given a set of interest points in two images, we want to determine correspondences, i.e., pairs of points that correspond to the same 3D point
- If there is a small relative baseline:
  - Use tracking (Lucas-Kanade, etc)
    - Track POIs in image 1 to their corresponding positions in image 2
    - 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

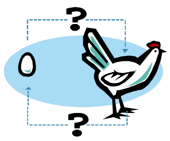


We need corresponding points to estimate **F**

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

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

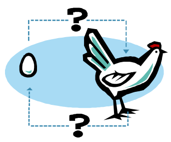
Chicken and egg revisited



- 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 S 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 S
- 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 S
  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 K times

Probabilities

- Let w be the fraction of inliers in S
- 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 K iterations is $(1 - w^N)^K$
- The probability that in K iteration, at least once, all N points are inliers: $p = 1 - (1 - w^N)^K$
- Solve for K: $K = \frac{\log(1-p)}{\log(1-w^N)}$

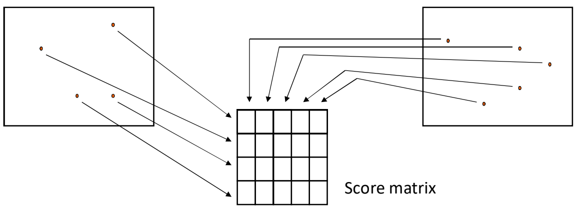
The odds are against us

- 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 K must be VERY^8 large in order to make p_{SUCCESS} close to 1
- Possible strategies for dealing with this problem?

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
  based on **visual appearance**
- 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



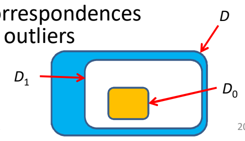
Each entry in the matching matrix describes how well a certain point in image 1 matches another point in image 2.
For example: high score = good match

Matching matrix

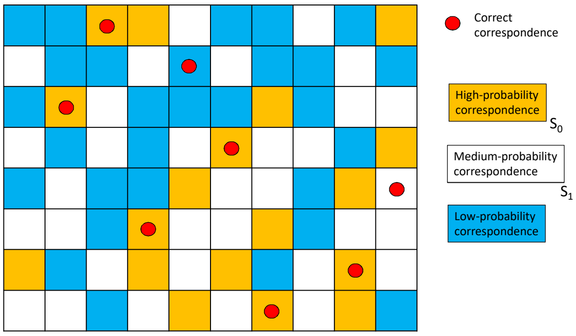
- The matching score can be based on *similarity of visual appearance* or other a priori knowledge about the scene (rather than geometric properties)
- For example
  – SIFT features [see previous lecture!]
  – MSER [see previous lecture!]
  – Color description
  – Camera motions in relation to scene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



Other ways to reduce *K*

- Try work with models for correspondences that require less than 8 pairs as a minimal case.
- Essential matrix **E** (N = 5)
- P3P (N=3)

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**

BREAK

Project 2

Reconstruction of 3D object / scene from multiple views

Initial assumptions

- A single camera is moving around in 3D space, taking pictures at multiple **distinct** positions of one and the same object/scene.
- These positions are **not known** with sufficient accuracy
- The camera has **known internal calibration** parameters that are constant
- Lens distortion effects are neglected
  - Or has been compensated for
  - **The pin-hole camera model valid**

Example: multiple views of a dinosaur

Initial assumptions (II)

- The images are ordered, for example, over a temporal parameter
  - Two consecutive images in the sequence have a smaller baseline than images that are far apart in the sequence
  - Adjacent images in the sequence can be expected to have a significant overlap. This means that many points are visible in both images.
- The camera path may or may not be closed

Project goal

- Based only on these images and the camera calibration:

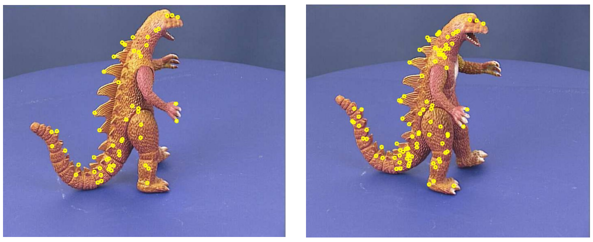
Generate a 3D representation of the object/scene

- The 3D representation can then be rendered from any viewpoint, even one not included in the data set

The 3D representation

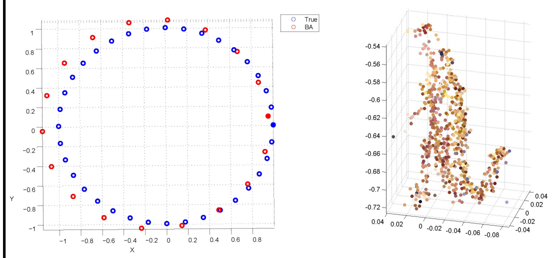
- The object(s) in the scene is represented in terms of a set of 3D points
  - Initially unordered (a *point cloud*)
  - Using spatial relations in 3D space and in the 2D images the 3D points can be connected into one or more 2D surfaces in 3D spaces
  - These surfaces can be texture mapped using the 2D images

Example: initial data



Two examples of images from the *dinosaur* sequence, with corresponding interest points

Example: result



Results from 2011 project by Bertil Grelsson and Freddie Åström