INF 5300 21.1.15
Introduction and a taste of the course

Lecturers:

- Are Jensen
- Anne Schistad Solberg

Contact information

- Are Jensen
  – On IFI wednesdays, room 4457
    Email: arej@ifi.uio.no
- Anne Schistad Solberg
  – Room 4458
  – anne@ifi.uio.no , 22852435

Course hightlights

- Two main parts:
  – Computer vision
  – Pattern recognition
- One mandatory exercise
- Lab exercise sessions
- Oral exam if less than 10 approx. students

Lecture plan

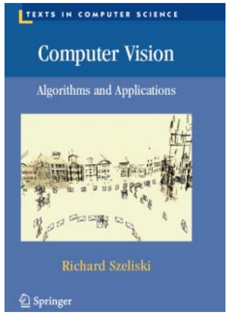
- 21.01: Introduction/Classification basics (Anne/Are)
- 28.01: Feature selection (Are)
- 04.02: Linear feature transforms (Are)
- 11.02: Lab on feature selection and linear transforms (Are)
- 25.02: Regularization and snakes (Anne)
- 04.03: Markov random fields and contextual models (Anne)
- 11.03: Lab on snakes/Markov models (Anne)
- 18.03: Support vector machines (SVM) (Are)
- 25.03: Lab on SVM (Are)
- 08.04: Extracting good features for matching/tracking (Anne)
- 22.04: Image alignment and RANSAC (Anne)
- 29.04: Motion estimation (Anne)
- 06.05: Lab (Anne)
- 13.05: Label propagation/graph-based segmentation (Are)
- 20.05: Lab on graphs and segmentation (Are)
- 27.05?: Repetition

Curriculum

- Lecture notes are most important!
- Some lectures are based on:
  - ”Pattern Recognition” by S. Theodoridis and K. Koutroumbas:
  - (A paper copy of selected sections will be provided)
- Many of the lectures are based on:
  - «Computer vision: Algorithms and applications» by R. Szeliski

Computer Vision book

- See http://szeliski.org/Book/
- See webpage for ordering
- A PDF of book is also available



Every picture tells a story

- Goal of computer vision is to write computer programs that can interpret images

Can computers match (or beat) human vision?

- Yes and no (but mostly no!)
  - humans are much better at “hard” things
  - computers can be better at “easy” things

Introduction to feature selection

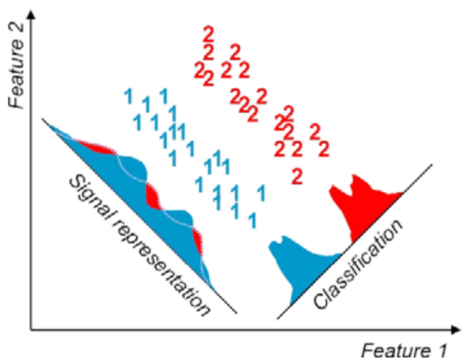
- Goal: use few, but good features.
- From a set of many features, how can we find the best features or feature combinations with only a few features?
  - Find the best subset using feature selection
  - Find a low-dimensional linear combination of the features

Feature selection

- Given a large set of N features, how do we select the best subset of m features?
  - How do we select m?
  - Finding the best combination of m features out a N possible is a large optimization problem.
  - Full search is normally not possible.
  - Suboptimal approaches are often used.
  - How many features are needed?

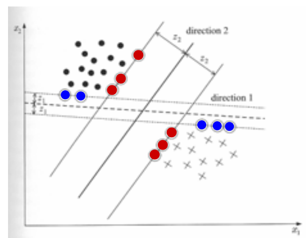
Introduction to feature transforms

- Compute lower-dimensional projections of the N-dimensional space
  - PCA
  - Fisher's linear discriminant
  - Projection pursuit and other non-linear approaches



Classification by Support Vector Machines

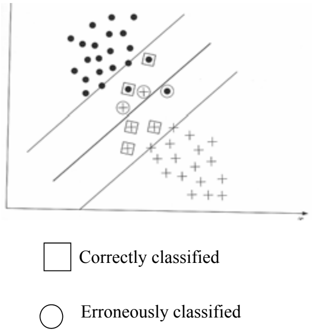
- SVM defines the decision boundary from the samples closest to the other classes.
- If a problem is linearly separable, many hyperplanes can divide the classes.
- In SVM the hyperplane is selected as the one the the largest distance to the other class.



The support vectors for hyperplane 1 are the blue circles.
The support vectors for hyperplane 2 are the red circles.

SVM: The nonseparable case

- If the two classes are nonseparable, a hyperplane satisfying the conditions $w^Tx-w_0=\pm1$ cannot be found.
- A penalty will now be given to the samples closest to the boundary.
- A magic trick solves the classification problem in a higher dimensional space.

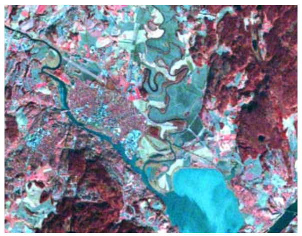


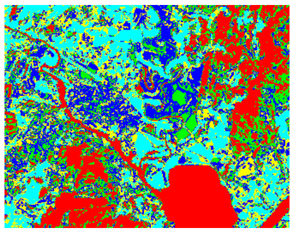
A taste of regularization

- In segmentation, we often want objects with smooth boundaries.
- Regularization is the process of constraining an algorithm to produce smooth objects/boundaries.
- We can regularize simple classification using contextual models
  - The class labels for a pixel will be modelled as a function of its neighbors.
- We can regularize the shape of the boundaries
  - Example: snakes
- We can regularize the estimated motion of objects in e.g. a video.

Introduction – contextual classification

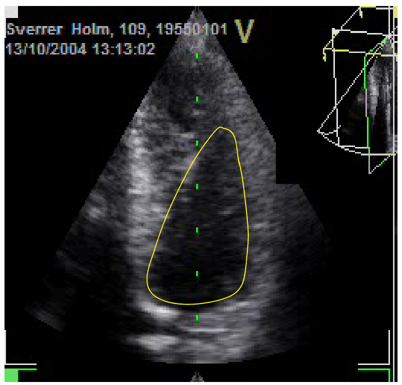
- An image normally contains areas of similar class
  - neighboring pixels tend to be correlated.
- Classified images based on a non-contextual model often contain isolated misclassified pixels (or small regions).
- How can we get rid of this?
  - Majority filtering in a local neighborhood
  - Remove small regions by region area
  - Relaxation (Kittler and Foglein – see INF 3300 Lecture 23.09.03)
  - Bayesian models for the joint distribution of pixel labels in a neighborhood.
- How do we know if the small regions are correct or not?
  - Look at the data, integrate spatial models in the classifier.





Snakes – active contours

Example – segmenting ultrasound images of the hearth

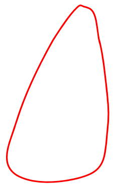


Find the border of the left ventricle

- 2D views have partly discontinuous border
- Noisy image

The initial idea: Snakes

- An active contour (snake) is a set of points which aims to enclose a target feature.
- Snakes are model-based methods for localization and tracking of image structures.
- The snake is defined as an energy minimizing contour (often defined using splines).
- The energy of the snake depends on its shape and location within the image.
- Snakes are attracted to image boundaries through forces.



Which features are good for tracking?

- In many applications, we need to match or track different parts of the image.
- Applications are e.g.
  - Tracking in video
  - Motion estimation
  - Multi-camera views

Image matching

- How do we compute the correspondence between these images?

•by Diva Sian

•by swashford

What type of features are good?

•Point-like features?

•Region-based features?

•Edge-based features?

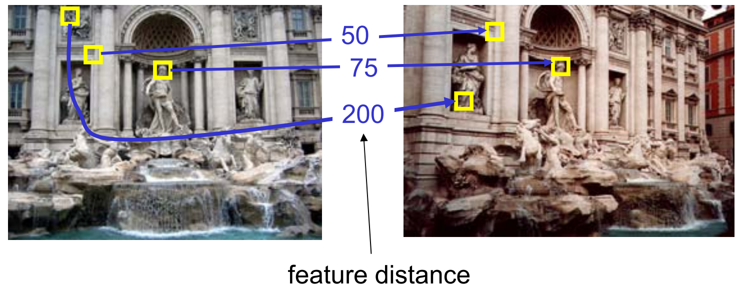
•Line-based features?

Four steps in feature matching

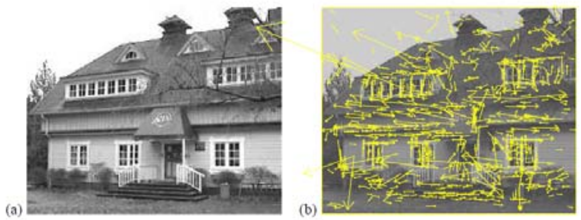
1. Feature extraction
   – Search for characteristic locations
2. Feature description
   – Select a suitable descriptor that is easy to match
3. Feature matching
   – Efficient search for matching candidates in other images
4. Feature tracking
   – Search a small neighborhood around the given location
     • An alternative to step 3.

Feature detection and matching

A set of keypoints are detected and matched in two images



Robust matching of key points

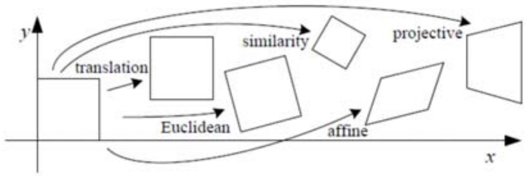


- A set of corresponding feature points in two images.
- Goal: estimate the geometrical transform that we need to align the two images.
- Problem: movements are noisy and establishing ONE geometric transform for the image is difficult.

Goals of image matching

- Consider two images containing partly the the same objects but at different times, from different sensors, or from different views.
- Assume that a set of features has been detected and the matching between corresponding features determined.
- Now we need to:
  – Verify that the mathing is geometrically consistent
  – This is the case if we can compute the motion between the features using a simple 2D or 3D geometric transform
  – How do we do this robustly?

2D and 3D feature-based alignment



- We restrict us to parametric transforms such as the ones illustrated above.

Simple operations:

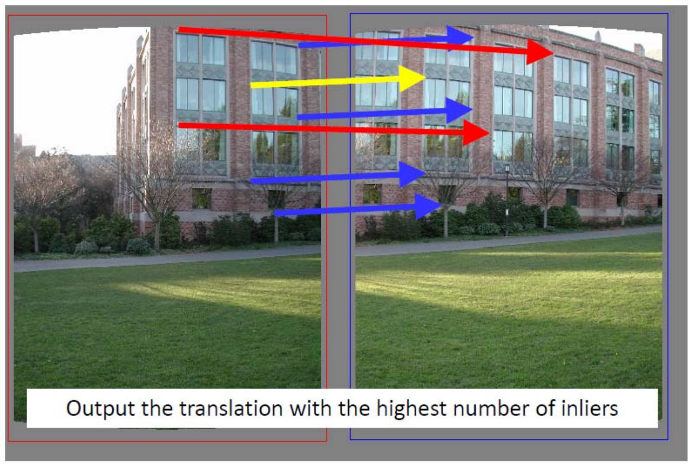
- Translation
- Euclidean = translation + rotation
- Affine transforms
- Similarity = scaled rotation
- Projection

Robustness of matching

Introducing a robust matching algorithm

- The detected features are not perfect, there may be outliers where the match is NOT good.
- If we want to fit a line:
  - Count the number of points that agree with the line.
    - Agree means that the distance between the location of the estimated and the true coordinates is very small.
    - Points which fulfill this criterion are called inliers.
    - Other points are called outliers.
  - For all possible lines, select the one with the larges number of inliers.

<u>RAN</u>dom **<u>S</u>**ample **<u>C</u>**onsensus



Motion estimation

- Motion estimation in e.g. medical ultrasound images gives information about the object (e.g. how the hearth behaves).
- Motion estimation in a video sequence can be due to camera instability and might be removed.
- Visual motion indicates the dynamics in the scene.
- Geometrical motion models are used in parametric motion estimation.

Dense motion estimation

- Motion vectors are now estimated from every point an a image sequence.
- Motion maps are created, and each pixel can have a different motion vector.
- Some regularization of the motion vectors is done to get smooth estimates.
  - No restriction that all pixels move in the same average direction.
- Video normally has high frame rate:
  - Small motion between one fram and the next frame

Patch matching

- How do we determine correspondences?
  - *block matching* or *SSD* (sum squared differences)
- How do we ensure a smooth motion estimate?

Essential steps in motion estimation

- An error metric to compare the two images must be chosen.
- A search technique to compute the best match is needed.
  - Pyramid search is often used to speed up the process.
- Accurate motion estimates might need subpixel accuracy.
- Regularization is often applied since the motion vectors are not reliable in all regions.
  - For compex motion layered motion models might also be needed.

Seeing motion from a static picture?