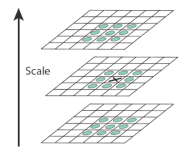
Scale invariance

- SIFT achieves scale stability by focusing attention on structure that is defined in terms of scale.
- If a structure has an inherent scale, then it can be extracted from an image of unknown scale by considering that image at different scales.
- *Image scale space*
  - Consider the images at many scales
  - Each scale leads to a different blurry image
  - Consider sigma as a 3rd coordinate
  - So an image “cube” now is (x,y,sigma)
  - In what follows, we use a *difference* scale space

- For each scale we use the difference between the image at two successive (discrete) scales (σ and $k\sigma$).
- This detects structure in a scale invariant way (think blobs)
- Keep points that are optimal in three directions: (x, y, σ).



From Lowe 2004

Figure 2: Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).

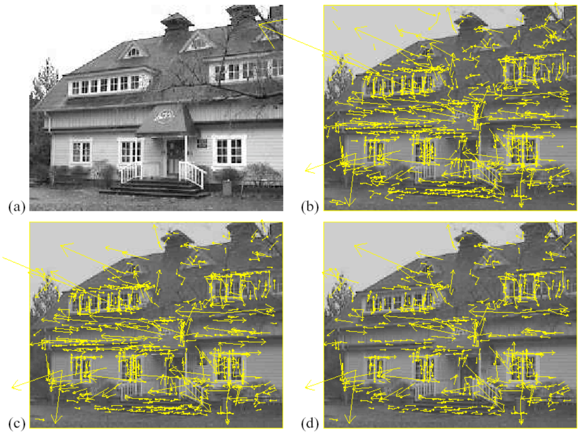
Distinctiveness

- For points that are optimum from previous …
  - These points have an associated scale
- Only points in regions that have significant edges in two directions are considered “distinctive” interest points (reject the rest)
  - Corners localize better than edges
  - The gradient gives only a single direction
  - We need to consider additional information around the point to distinguish corners from edges
  - One method is to collect edge information in a region around the point
  - Lowe (04) instead uses a method based on principle curvature

---

Direction

- For points that have not been rejected …
  - These points have an associated scale σ.
- Consider a disc with radius $3\sigma/2$
- Look at edge direction a locations in this disc, and build a histogram of the angles.
- Values in a dominant peak provide a direction
  - A second direction can create a second keypoint if it is at least 80% as popular.
- From scale and direction, we can establish a coordinate system



From Lowe, IJCV 2004

Descriptor

- Capture the edge structure (essentially texture) in the region (of size of order σ) around the point in a vector (Lowe 04 uses 128 elements)

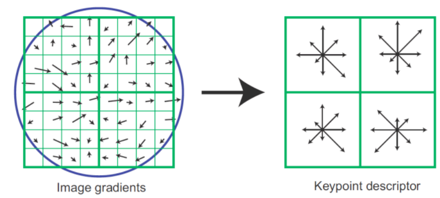


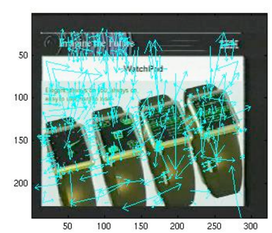
Figure 7: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper use 4x4 descriptors computed from a 16x16 sample array.

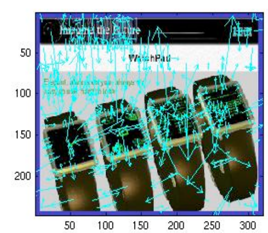
Invariant feature detection

- Descriptor is invariant to scale and in plane rotation
  - Feature had a natural scale
  - We established a direction
- Scaling and rotation can approximate out of plane camera rotation view changes for small patches (locally planar)

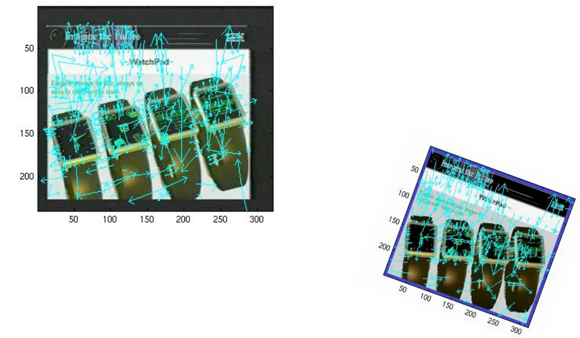
Invariant feature matching

- To “find” the object, match the local features





Invariant feature matching



Should work similarly to non-rotated case.

Initial matching

Constraining to correct part of image based on other information

After pruning outliers (Covered later)

• Keypoints from an object map into an image in an organized way.

• We will study how to improve matching on this bases in the context of *grouping*.

Syllabus Notes

- Next topics segmentation, grouping and fitting.
- We will do perhaps half each of §14, §15, and §16.

Grouping

http://www.youtube.com/watch?v=2VFG5fQHMro&feature=related

Segmentation, Grouping, and Fitting

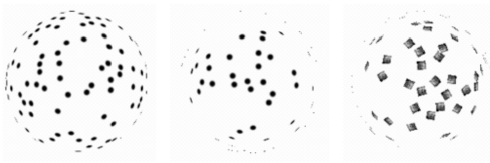
- Collect together tokens that belong together
- Gives a compact representation from an image/motion sequence/set of tokens that can be significantly easier to deal with
- What is the "right" group is often dependent on the application
- Broad theory is not known at present (and may not exist)
- These are general concepts--apply to many things, not just breaking images into regions of the same color.

Segmentation, Grouping, and Fitting

- Terminology varies and the usage and the meaning of segmentation, grouping, and fitting overlap. Somewhat common usage:
  - Grouping (or clustering) is quite general sometimes suggest a relatively high level (group the black and white halves of a penguin together).
  - Segmentation is suggestive of the grouping is done at a low level and is quite spatially (or temporally coherent) given regions in time or space.
  - Fitting when the focus is on a model associated with tokens. Issues:
    - which model?
    - which token goes to which element in the model (correspondence)?
    - how many elements in the model (how complex should it be)?

General ideas

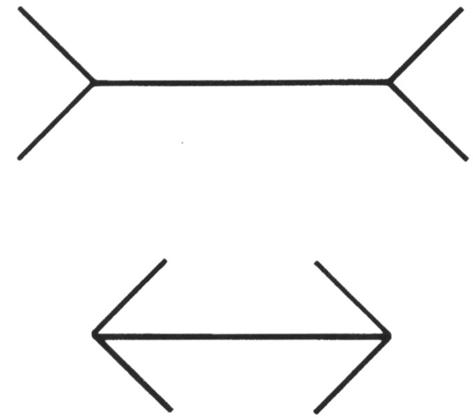
- Tokens
  - whatever we need to group (e.g. pixels, points, surface elements)
- Top down segmentation
  - tokens belong together because they lie on the same object
- Bottom up segmentation
  - tokens belong together because they are locally coherent
- These two are not mutually exclusive



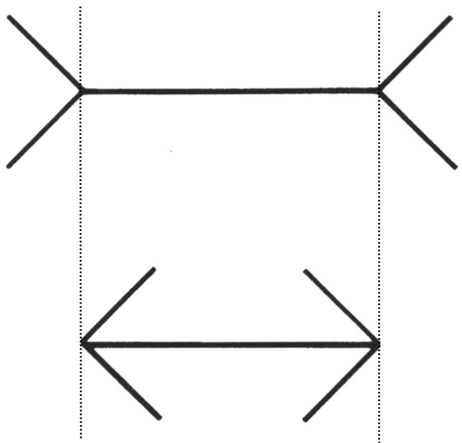
Why do these tokens belong together?

Basic ideas of grouping in humans

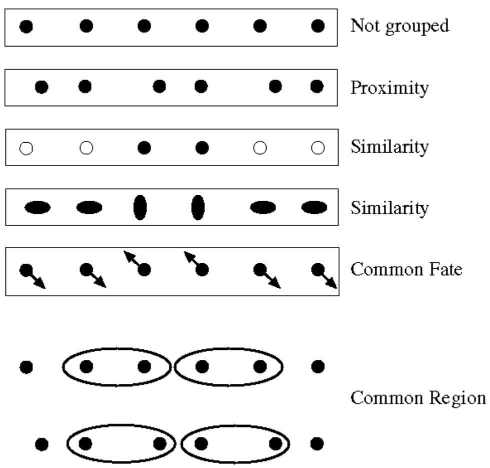
- Figure-ground discrimination
  - grouping can be seen in terms of allocating some elements to a figure, some to ground (impoverished theory)
- Gestalt factors
  - Elements in a collection of elements can have properties that result from relationships (e.g. Muller-Lyer effect)
  - A series of factors affect whether elements should be grouped together

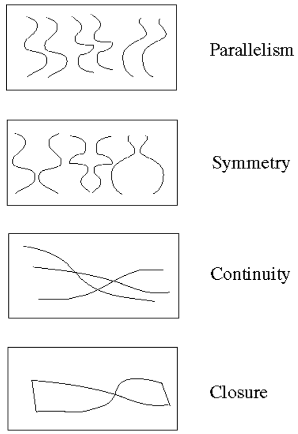


The Muller-Lyer illusion; the horizontal bar has properties that come only from its membership in a group



The Muller-Lyer illusion; the horizontal bar has properties that come only from its membership in a group



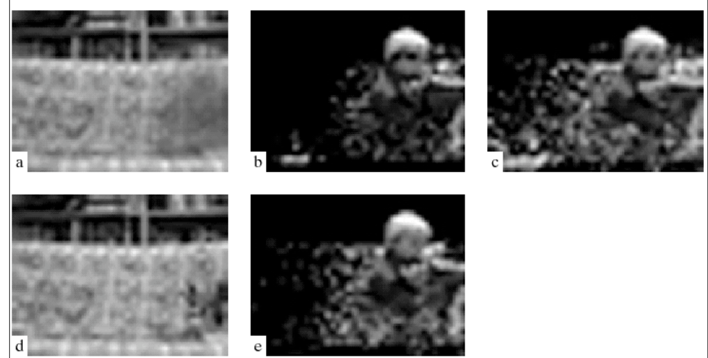


Background Subtraction

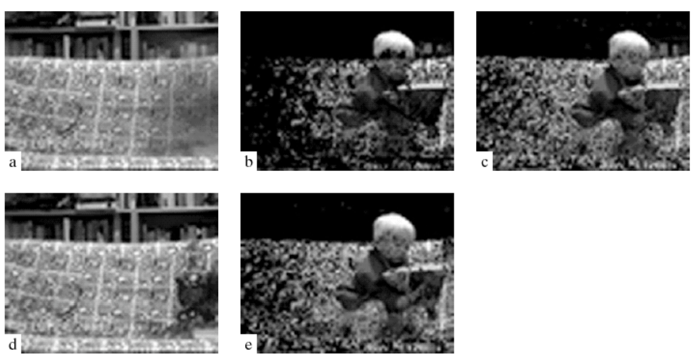
Background Subtraction

- If we know what the background looks like, it is easy to identify “interesting bits”
- Applications
  - Person in an office
  - Tracking cars on a road
  - Surveillance
- Approach:
  - Use a moving average to estimate background image
  - Subtract from current frame
  - Large absolute values are interesting pixels
    - trick: use morphological operations to clean up pixels (remove “holes”)

a) average of sequence b) difference from background > T1 c) difference from background > T2



d) using EM (discussed later) e) Forground using EM (discussed later)



Higher resolution version of previous. Note increased impact of noise.

Segmentation as clustering

- Cluster together (pixels, tokens, etc.) that belong together
- We assume that we can compute how close tokens are, or how close a token is to cluster.

Why is clustering hard?

Main reason

- The number of possible clusterings is exponential in the number of data points

Other important issues

- The number of clusters is usually **not** known
- A good distance function between points may not be known
- A good model explaining the existence of clusters is usually not available.
- High dimensionality

Data Representation

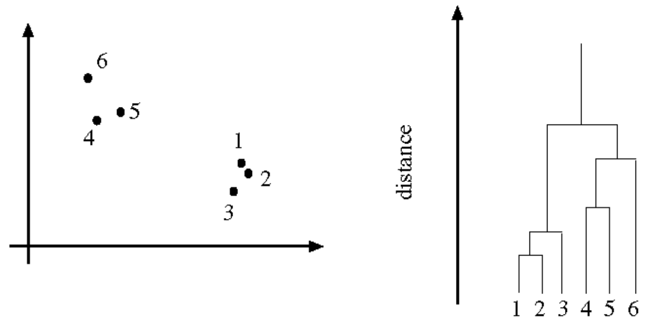
- Most common is an N dimensional “feature” vector.
- Most common distance is Euclidian distance.
- Be careful with scaling and units!
- Probabilistic modals finesse multiple modalities
- Problems with correlated variables can be mitigated using transformations and data reduction methods such as PCA, ICA.

Clustering approaches

- Agglomerative clustering
  - initialize: every item is a cluster
  - attach item that is “closest” to a cluster to that cluster
  - repeat
- Divisive clustering
  - split cluster along best boundary
  - repeat
- Probabilistic clustering
  - Define a probabilistic grouping model

Simple clustering approaches

- Point-Cluster or Cluster-Cluster distance
  - single-link clustering (minimum distance from point to points in clusters or among pairs of points, one from each cluster)
  - complete-link clustering (maximum)
  - group-average clustering (average)
  - (terms are not important, but concepts are worth thinking about)
- Dendrograms
  - classic picture of output as clustering process continues



K-Means

- Choose a fixed number of clusters ("K")
- Choose cluster centers (**means**) and point-cluster allocations (membership) to minimize the error

$$\sum_{i \in \text{clusters}} \left\{ \sum_{j \in \text{elements of i'th cluster}} \left\| x_j - \mu_i \right\|^2 \right\}$$

- **x**'s could be any set of features for which we can compute a distance (careful with scaling)

K-Means

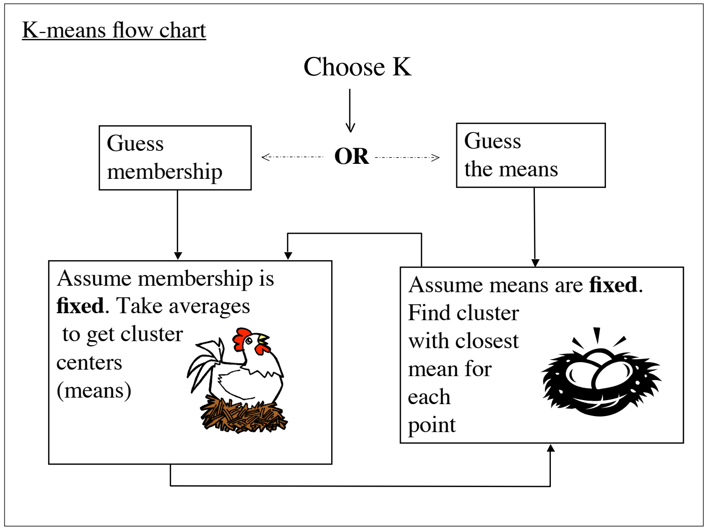
- Want to minimize

$$\sum_{i \in \text{clusters}} \left\{ \sum_{j \in \text{elements of i'th cluster}} \left\| x_j - \mu_i \right\|^2 \right\}$$

- **Cannot** do this optimization by search, because there are too many possible allocations.
- Standard difficulty which we handle with an iterative process (chicken and egg)

K-Means algorithm (intuition)

- If we know the cluster centers, the best cluster for each point is easy to compute
  - Just compute the distance to each to find the closest
- If we know the best cluster for each point, the cluster centers are also easy to compute
  - Just average the points in each cluster
- Algorithm
  - 1) Guess one of the two.
  - 2) Alternatively re-compute the values for each



K-means flow chart
Choose K
Guess membership
OR
Guess the means
Assume membership is **fixed**. Take averages to get cluster centers (means)
Assume means are **fixed**. Find cluster with closest mean for each point