CS664 Lecture #22: Distance transforms, Hausdorff matching, flexible models

Some material taken from:

- **Yuri Boykov, Western Ontario**

Announcements

- The SIFT demo toolkit is available from http://www.evolution.com/product/oem/download/?ch=Vision
  - Usage instructions are at http://www.evolution.com/product/oem/download/vision_usage.masn
    - Including a list of supported cameras
  - Implementation w/o GUI is at http://www.cs.ubc.ca/~lowe/keypoints/
- Next quiz on Tuesday 11/15

Chamfer matching

- Consider features as point sets
- Suppose we assign each model feature the distance to the nearest image feature $y \in P$

$$D_P(x) = \min_{y \in P} \| x - y \|$$

  - Intuition: from each image feature (point in the image) we grow out a cone
- Chamfer distance: $\sum_{x \in X} D_P(x)$
- Works, but must efficiently minimize it over many placements of the model

DP to the rescue!

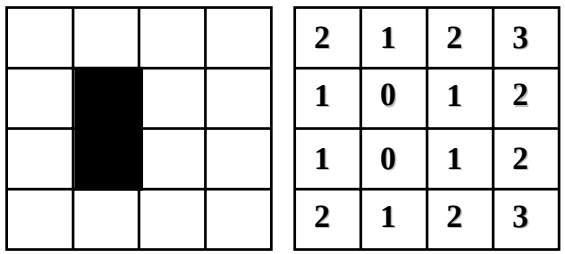
- Lots of computation is re-used, both within and across model placements
  - Places in the image near to image features are “good”, places far away are “bad”
  - This can be computed once only, from the image, and widely re-used
    - Within and across model placements
- You can think of this as “blurring” the binary features

Distance transform

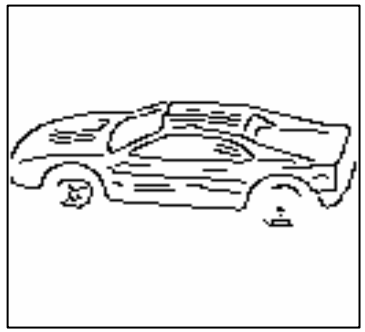
- Depends upon distance measure $\|\bullet\|$
  - We will assume the L_1 distance for simplicity
- Commonly computed on a grid Γ using

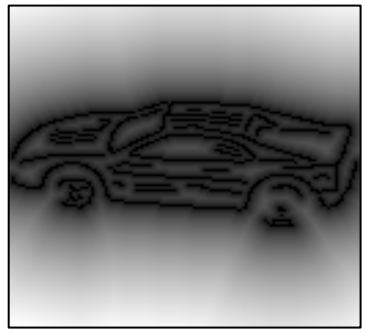
$$D_P(x) = \min_{y \in \Gamma} (\|x - y\| + 1_P(y))$$

  - Where $1_P(y) = 0$ when $y \in P$, ∞ otherwise



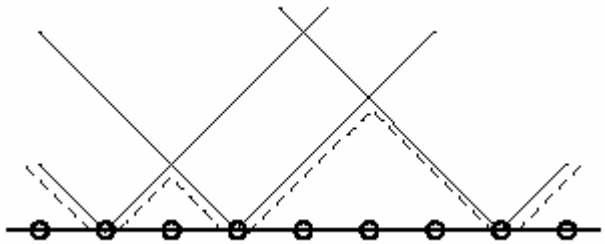
2	1	2	3
1	0	1	2
1	0	1	2
2	1	2	3





L_1 DT Algorithm

- Two pass O(n) algorithm for 1D L_1 norm (for simplicity just distance)
  1. Initialize: For all j
     $D[j] \leftarrow 1_P[j]$
  2. Forward: For j from 1 up to n-1
     $D[j] \leftarrow \min(D[j], D[j-1]+1)$
  3. Backward: For j from n-2 down to 0
     $D[j] \leftarrow \min(D[j], D[j+1]+1)$



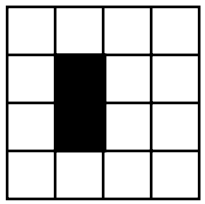
∞	0	∞	0	∞	∞	∞	0	∞
∞	0	1	0	1	2	3	0	1
1	0	1	0	1	2	1	0	1

Morphological operations

- DT related to “mathematical morphology”
  - A field with a very high ratio of terminology to content
- Local operations on binary images
  - Example: count the number of nearby 1’s
    - Make a binary decision based on this
  - Erosion, dilation, etc.
  - Most interesting variant is Conway’s Life

DT and Morphological Dilation

- Dilation operation replaces each point of P with some fixed point set Q
- Dilation by a "disc" C^d of radius d replaces each point with a disc
  - A point is in the dilation of P by C^d ($P \oplus C^d$) exactly when the distance transform value is no more than d (for appropriate disc)
  - $x \in P \oplus C^d \Leftrightarrow D_P(x) \leq d$



2	1	2	3
1	0	1	2
1	0	1	2
2	1	2	3

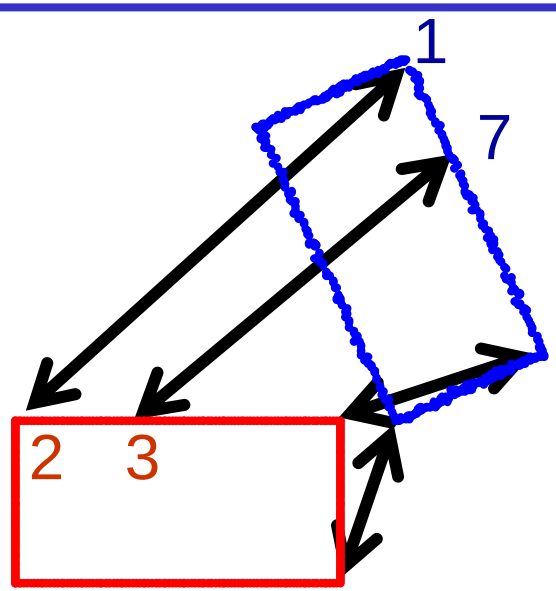
0	1	0	0
1	1	1	0
1	1	1	0
0	1	0	0

1	1	1	0
1	1	1	1
1	1	1	1
1	1	1	0

DT in Matching

- Chamfer measure
  - Sum of distance transform values
    - "Probe" DT at locations specified by model and sum resulting values
- Hausdorff distance (and generalizations)
  - Max-min distance which can be computed efficiently using distance transform
    - Generalization to quantile of distance transform values more useful in practice
- Iterated closest point (ICP) like methods
  - Assume closest point corresponds

Easy if Correspondence Known



1 – 2

7 – 3

...

Hard:

$$\epsilon(T) = \sum_j \min_i d(T * \mathbf{M}_i, \mathbf{D}_j)$$

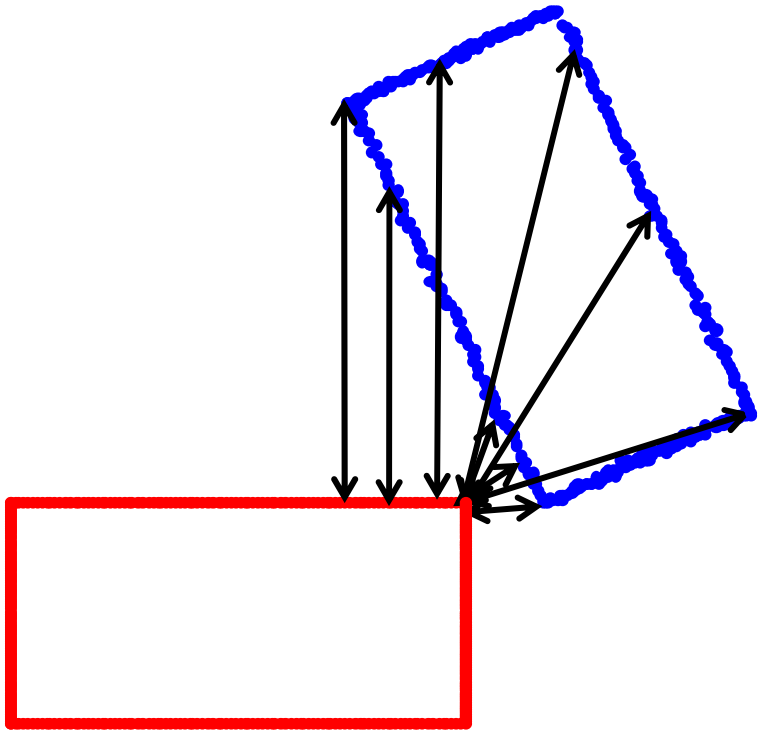
Easy:

Given *correspondences* $j \leftrightarrow \phi(j)$

Can minimize

$$\sum_j d(T * \mathbf{M}_{\phi(j)}, \mathbf{D}_j)$$

Don't Know Correspondence Guess and Try to Improve



- That's OK, just choose the closest point...
- *Of course* it's wrong, but it will get us closer

Problems with ICP

- Slow
  - Can take many iterations
  - Each iteration slow due to search for correspondences
    - Fitzgibbons: improve this by distance transform
- No convergence guarantees
  - Can get stuck in local minima
    - Not much to do about this
    - Can be improved by using robust distance measures (e.g., truncated quadratic)

Hausdorff Distance

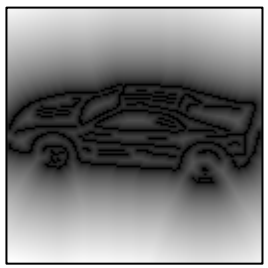
- Classical definition
  - Directed distance (not symmetric)
    - $h(A,B) = \max_{a \in A} \min_{b \in B} \| a-b \|$
  - Distance (symmetry)
    - $H(A,B) = \max(h(A,B), h(B,A))$
- Minimization term is simply a distance transform of B
  - $h(A,B) = \max_{a \in A} D_B(a)$
  - Maximize over selected values of DT
- Classical distance not robust, single “bad match” dominates value

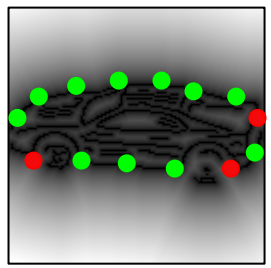
Hausdorff Matching

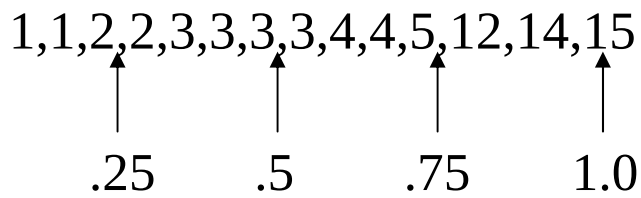
- Best match
  - Minimum fractional Hausdorff distance over given space of transformations
- Good matches
  - Above some fraction (rank) and/or below some distance
- Each point in (quantized) transformation space defines a distance
  - Search over transformation space
    - Efficient branch-and-bound “pruning” to skip transformations that cannot be good

Hausdorff Matching

- Partial (or fractional) Hausdorff distance to address robustness to outliers
  - Rank rather than maximum
    - $h_k(A,B) = kth_{a \in A} \min_{b \in B} \| a-b \| = kth_{a \in A} D_B(a)$
  - K-th largest value of D_B at locations given by A
  - Often specify as fraction f rather than rank
    - 0.5, median of distances; 0.75, 75th percentile

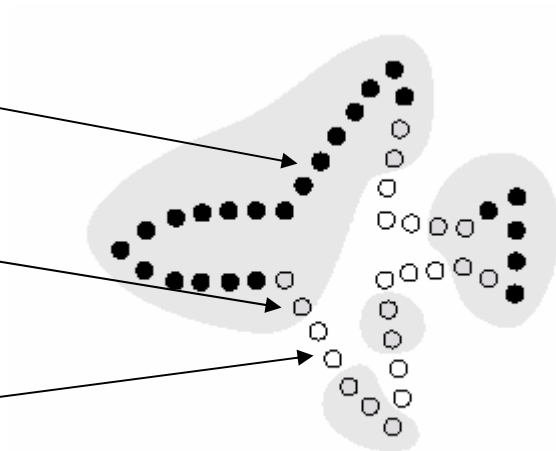






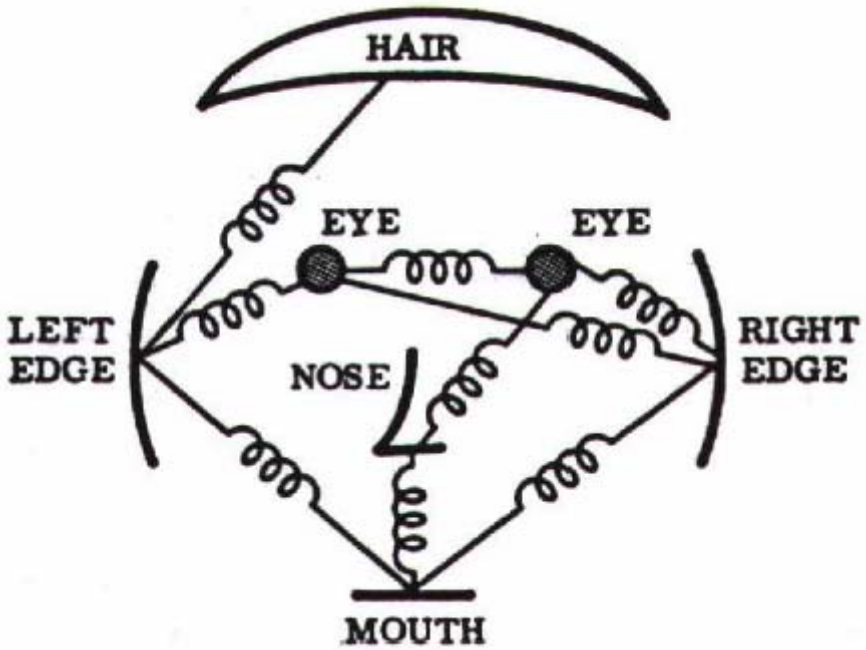
Spatially Coherent Matching

- Hausdorff and Chamfer matching do not measure degree of connectivity
  - E.g., edge chains versus isolated points
- Spatially coherent matching approach
  - Separate features into three subsets
    - Matchable
      - Near image features
    - Boundary
      - Matchable but near un-matchable
    - Un-matchable
      - Far from image features



Flexible models

- Classical recognition doesn't handle non-rigid motions at all well
- Pictorial structures
  - Parts + springs
  - Appearance models

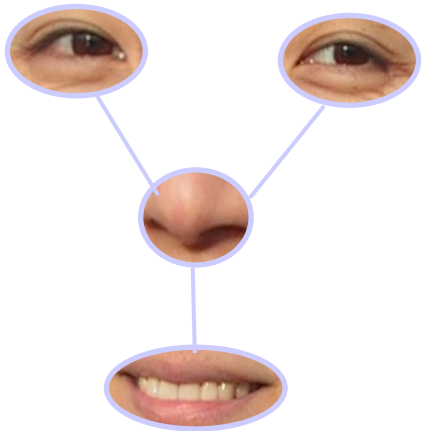


Model definition

- Parts are $V=\{v_1,\ldots,v_n\}$, graph vertices
- Configuration $L=(l_1, \ldots, l_n)$
  - Specifying locations of the parts
- Appearance parameters $A=(a_1, \ldots, a_n)$
  - Model for each part
- Edge e_{ij}, $(v_i,v_j) \in E$ for connected parts
  - Explicit dependency between part locations l_i, l_j
- Connection parameters $C=\{c_{ij} \mid e_{ij} \in E\}$
  - Spring parameters for each pair of connected parts

Flexible Template Algorithms

- Difficulty depends on structure of graph
  - General case exponential time
  - Consider special case in which parts translate with respect to common origin
    - E.g., useful for faces



- Parts $V = \{v_1, \ldots v_n\}$
- Distinguished central part v_1
- Spring c_{i1} connecting v_i to v_1
- Quadratic cost for spring

Central Part Energy Function

- Location $L = (l_1, \ldots, l_n)$ specifies where each part positioned in image
- Best location $\min_L (\Sigma_i m_i(l_i) + d_i(l_i, l_1))$
  - Part cost $m_i(l_i)$
    - Measures degree of mismatch of appearance a_i when part v_i placed at location l_i
  - Deformation cost $d_i(l_i, l_1)$
    - Spring cost c_{i1} of part v_i measured with respect to central part v_1
    - Note deformation cost zero for part v_1 (wrt self)

Consider Case of 2 Parts

- $\min_{l1,l2} (m_1(l_1) + m_2(l_2) + \| l_2 - T_2(l_1) \|^2)$
  - Here, $T_2(l_1)$ transforms l_1 to ideal location wrt l_2 (offset)
- $\min_{l1} (m_1(l_1) + \min_{l2} (m_2(l_2) + \| l_2 - T_2(l_1) \|^2))$
  - But $\min_x (f(x) + \| x - y \|^2)$ is a distance transform
- $\min_{l1} (m_1(l_1) + D_{m2}(T_2(l_1)))$
- Sequential rather than simultaneous min
  - Don't need to consider each pair of positions for the two parts because a distance