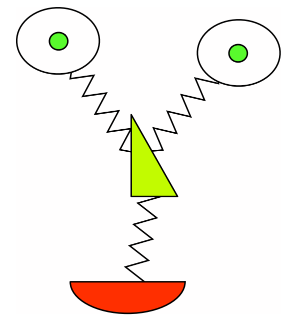
Object Recognition with Pictorial Structures

Pedro F. Felzenszwalb
University of Chicago
pff@cs.uchicago.edu

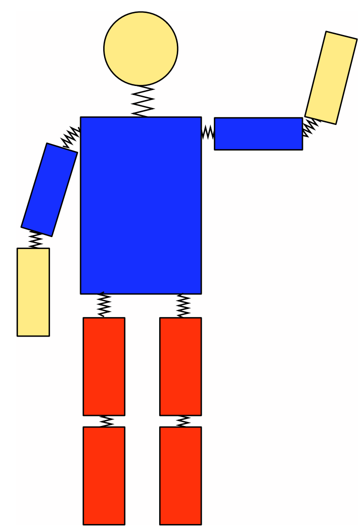
Joint work with Daniel P. Huttenlocher

Pictorial structures



Part-based representation:

- Each part models local visual properties.
- "Springs" model spatial relationships.
- Joint estimation of part locations.
  - No hard detection of parts or features.
  - No initialization parameters.



- Model is represented by a graph $G = (V, E)$.
  - $V = \{v_1, \ldots, v_n\}$ are the parts.
  - $(v_i, v_j) \in E$ indicates a connection between parts.
- $m_i(l_i)$ is the cost of placing part i at location l_i.
- $d_{ij}(l_i, l_j)$ is a deformation cost.
- Optimal location for object is given by $L^* = (l_1^*, \ldots, l_n^*)$,

$$L^* = \underset{L}{\operatorname{argmin}} \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right)$$

Efficient minimization

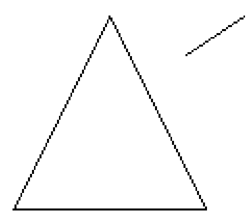
$$L^* = \underset{L}{\text{argmin}} \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right)$$

- n parts and h locations gives h^n configurations.
- If graph is a tree we can use dynamic programming.
  - $O(nh^2)$, much better but still slow.
- If $d_{ij}(l_i, l_j) = ||T_{ij}(l_i) - T_{ji}(l_j)||^2$ can use DT.
  - $O(nh)$, as good as matching each part separately!!

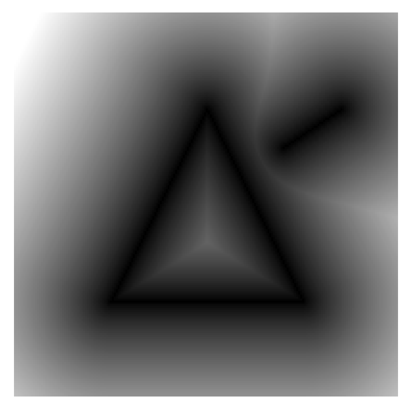
Distance transform

Given a set of points on a grid $P \subseteq \mathcal{G}$,
the quadratic distance transform of P is,

$$\mathcal{D}_P(q) = \min_{p \in P} ||q - p||^2$$



P



$\mathcal{D}_P$

Generalized distance transform

Given a function $f : \mathcal{G} \rightarrow \mathbb{R}$,

$$\mathcal{D}_f(q) = \min_{p \in \mathcal{G}} \left(||q - p||^2 + f(p) \right)$$

- for each location q, find nearby location p with $f(p)$ small.

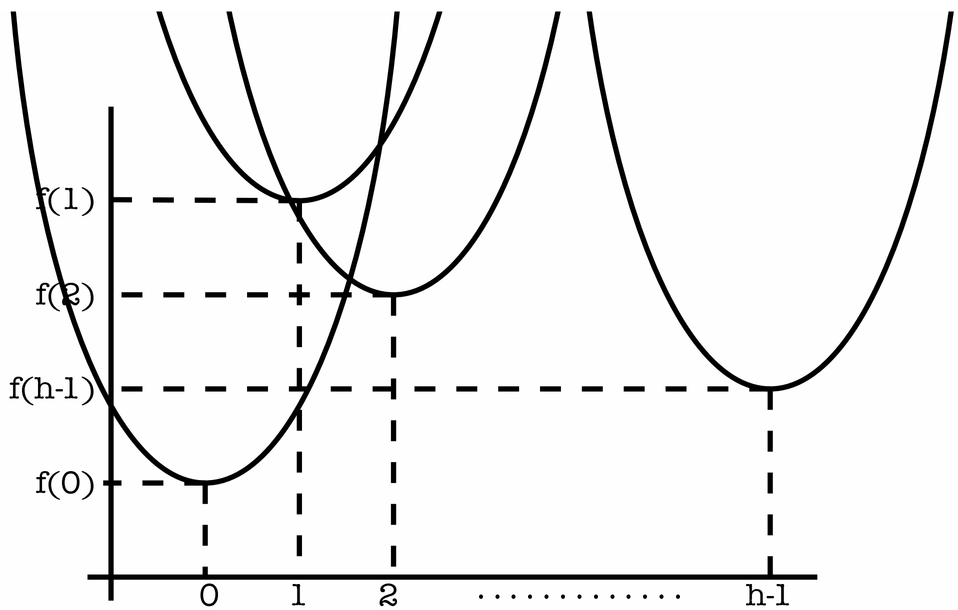
- equals DT of points P if f is an indicator function.

$$f(p) = \begin{cases} 0 & \text{if } p \in P \\ \infty & \text{otherwise} \end{cases}.$$

1D case: $\mathcal{D}_f(q) = \min_{p \in \mathcal{G}} \Big((q-p)^2 + f(p) \Big)$

For each p, $\mathcal{D}_f(q)$ is below the parabola rooted at $(p, f(p))$.

$\mathcal{D}_f(q)$ is defined by the lower envelope of h parabolas.



There is a simple geometric algorithm that computes $\mathcal{D}_f(p)$ in $O(h)$ time for the 1D case.

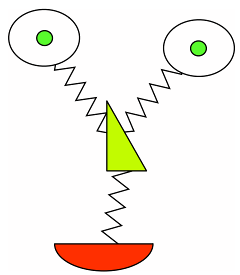
- similar to Graham's scan convex hull algorithm.
- about 20 lines of C code.

The 2D case is "separable", it can be solved by sequential 1D transformations along rows and columns of the grid.

See **Distance Transforms of Sampled Functions**, Felzenszwalb and Huttenlocher.

Simple face model

- Locations are positions in the image grid.
- Match cost $m_i(l_i)$ for placing part i at l_i.
- Central part v_1 - the nose.
- Each part has an ideal position p_i relative to nose.
  - Let $T_{1i}(l_1) = l_1 + p_i$,



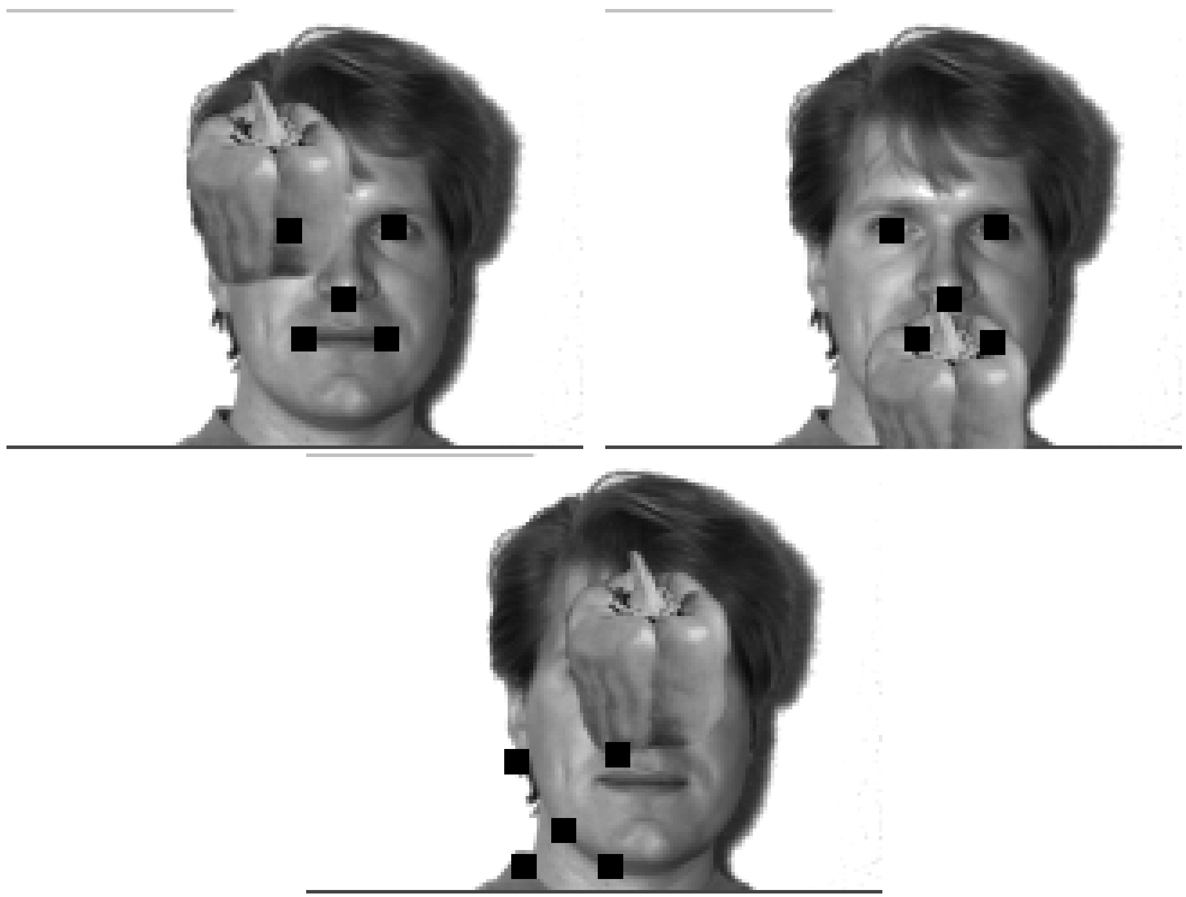
$$E(l_1, \ldots, l_n) = \sum_{i=1}^{n} m_i(l_i) + \sum_{i=2}^{n} ||l_i - T_{1i}(l_1)||^2$$

Efficient minimization

$$
\begin{aligned}
L^* &= \operatorname*{argmin}_{L} \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{i=2}^{n} ||l_i - T_{1i}(l_1)||^2 \right) \\
L^* &= \operatorname*{argmin}_{L} \left(m_1(l_1) + \sum_{i=2}^{n} m_i(l_i) + ||l_i - T_{1i}(l_1)||^2 \right) \\
l_1^* &= \operatorname*{argmin}_{l_1} \left(m_1(l_1) + \sum_{i=2}^{n} \min_{l_i}(m_i(l_i) + ||l_i - T_{1i}(l_1)||^2) \right) \\
l_1^* &= \operatorname*{argmin}_{l_1} \left(m_1(l_1) + \sum_{i=2}^{n} \mathcal{D}_{m_i}(T_{1i}(l_1)) \right)
\end{aligned}
$$

Matching results

Matching results



Summary

- Generic framework for part-based modeling.
- Global minimization for deformable objects can be fast.
- Soft detection avoids unnecessary early decisions.
- Partial occlusion is handled automatically.