# Efficient Belief Propagation for Early Vision

Pedro Felzenszwalb
University of Chicago
Department of Computer Science

Joint work with Daniel Huttenlocher

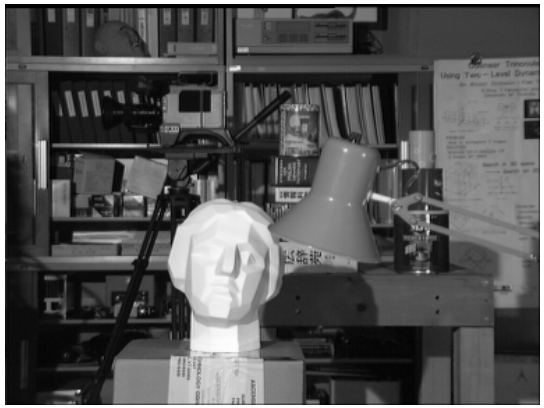
# Overview

- Markov Random Field (MRF) models are broadly useful for low-level vision
  - Framework for expressing tradeoff between spatial coherence and fidelity to data
- Substantial recent advances in algorithms
  - Two main approaches:
    graph cuts, loopy belief propagation (LBP)
- Describe three techniques to speedup LBP
  - Resulting algorithm is hundred of times faster than conventional one

# Low Level Vision Problems

- Estimate label at each pixel
  - Restoration: intensity
  - Stereo: disparity
  - Optical flow: motion vector

left camera



right camera

disparities

# Pixel Labeling Problem

- Pixels $P$, Neighborhood $N$, Labels $L$
- Labeling $f$ assigns label $f_p \in L$ to each pixel $p \in P$
  - Labels should be compatible local measurements
  - Labeling should be smooth almost everywhere
- Quality defined by energy function

$$E(f) = \sum_{p} D_p(f_p) + \sum_{(p,q) \in N} V(f_p - f_q)$$

data costs + discontinuity costs

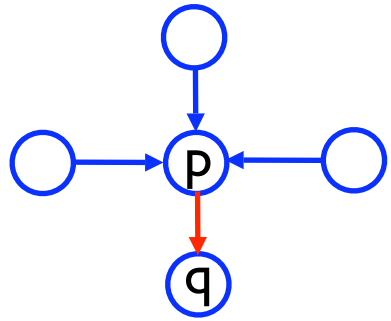
# Types of energy function

- $V(f_p - f_q)$ often truncated quadratic
  - Piecewise smooth assumption
- $D_p(f_p)$ may also be truncated quadratic
  - Image restoration with Gaussian + salt-and-pepper noise
- $D_p(f_p)$ often arbitrary
  - In stereo $D_p(f_p) = (I_{left}(x, y) - I_{right}(x + f_p, y))^2$
- In general minimizing the energy is NP-hard
  - approximate solution through α-expansions, variational methods, loopy belief propagation, ...

# Belief Propagation

- Iterative local update method
- Pixels collect local evidence by sending messages to their neighbors
  - Messages are vectors in $R^k$, $k = |L|$
  - $m^t_{p \to q}$ is message sent from pixel $p$ to pixel $q$ at time $t$
  - $p$'s view of $q$'s label
- $m^t_{p \to q}(f_q)$ is low if $p$ believes $f_q$ is good label for $q$
- Finds optimal labeling if there are no loops in (P, N)

- Messages are updated in parallel




$$m^0_{p \to q}(f_q) = 0$$

$$m^t_{p \to q}(f_q) = \min_{f_p} \left( D_p(f_p) + V(f_p - f_q) + \sum_{s \in N(p) \setminus q} m^{t-1}_{s \to p}(f_p) \right)$$

- After T iterations minimize belief of each pixel

$$b_p(f_p) = D_p(f_p) + \sum_{s \in N(p) \setminus q} m^T_{s \to p}(f_p)$$

- Intuition: If graph (P, N) was a tree this is equivalent to dynamic programming

- Running time: $O(nk^2T)$

# Speedups

- Basic running time: O($nk^2T$)
- Fast message update reduces message computation from O($k^2$) to O($k$)
- Bipartite graph: 2 times faster
- Multigrid reduces $T$ to constant
- Final method is O($nk$)

# Fast Message Updates

Define $h(f_p) = D_p(f_p) + \sum_{s \in N(p) \setminus q} m^{t-1}{}_{s \to p}(f_p)$

Write message as $m^t{}_{p \to q}(f_q) = \min_{f_p} \left( h(f_p) + V(f_p - f_q) \right)$

- Convolution of $h$ and $V$ in the (min,+) semiring
  - No general algorithm like FFT
  - Best known algorithm $O(k^2 / \log k)$
  - Certain important cases can be computed in linear time

# Min-convolution

$$(h \otimes g)(x) = \min_y \left( h(y) + g(x-y) \right)$$

- Comutative, Associative
- Distributes over min ("linearity")

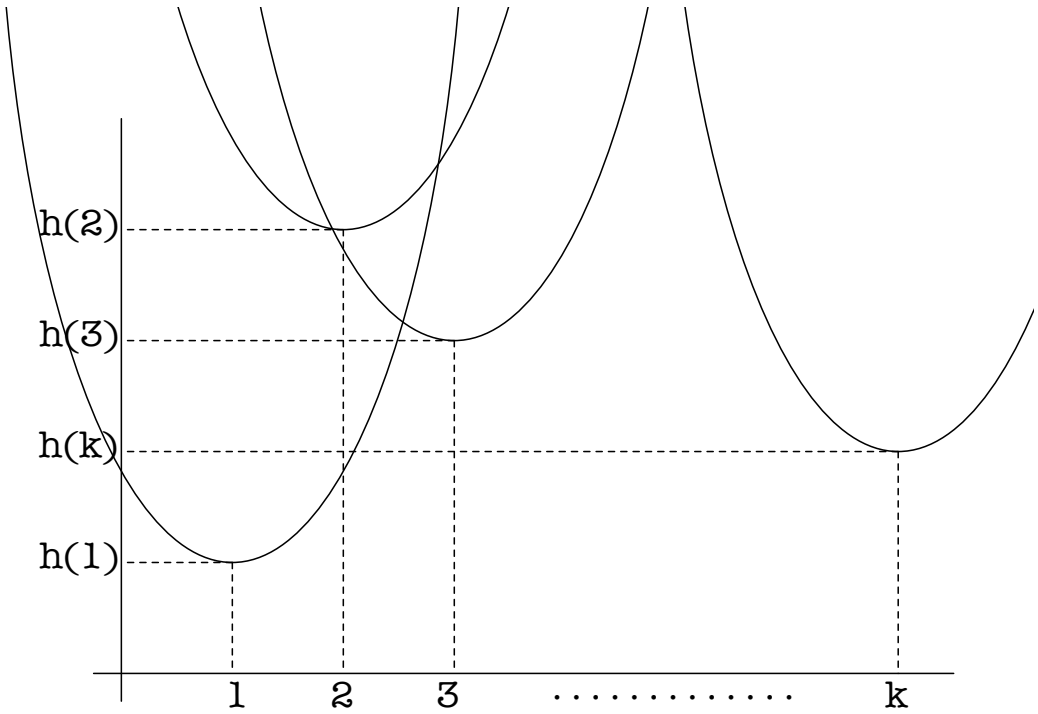
$$h \otimes \min(g_1, g_2) = \min(h \otimes g_1, h \otimes g_2)$$

- Can compute in linear time if $g$ is convex
- Truncated quadratic is min of two convex functions
  - quadratic and constant

# Min-convolution with parabola

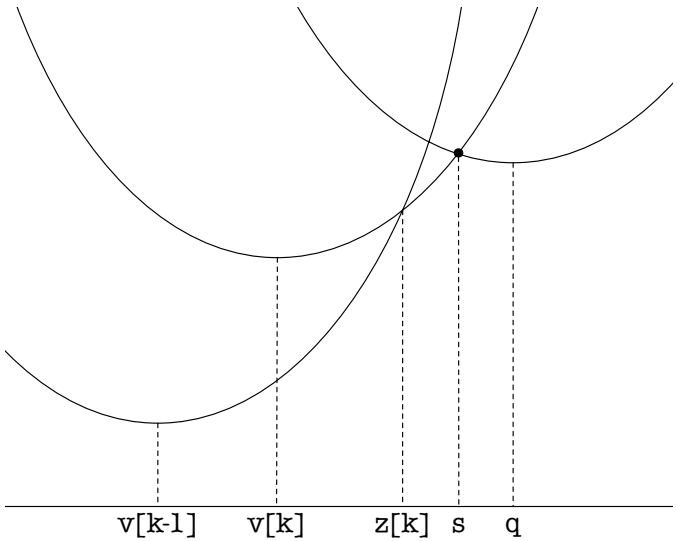
$$(h \otimes g)(x) = \min_y \left( h(y) + (x-y)^2 \right)$$

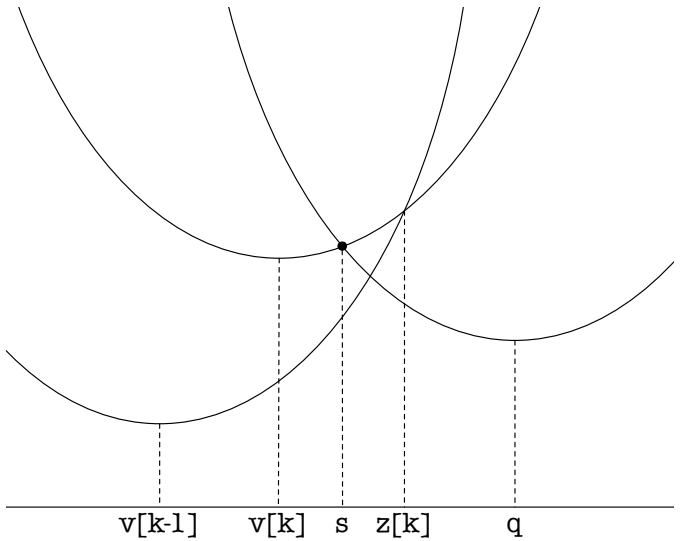
- Each value y constrains $h \otimes g$ to be below a parabola rooted at $(y, h(y))$.
  - result is defined by lower envelope of k parabolas



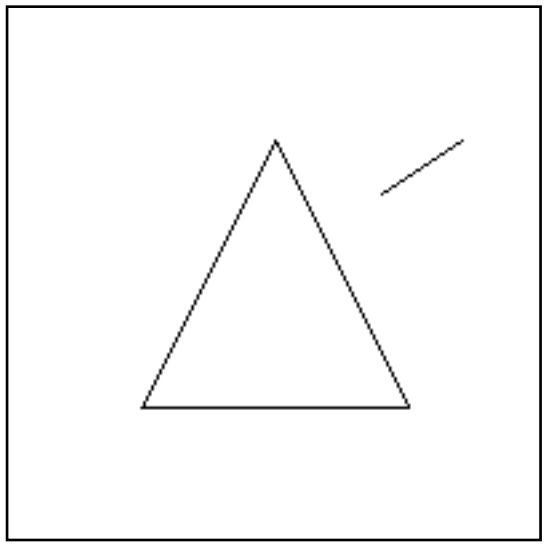
# Lower Envelope of Parabolas

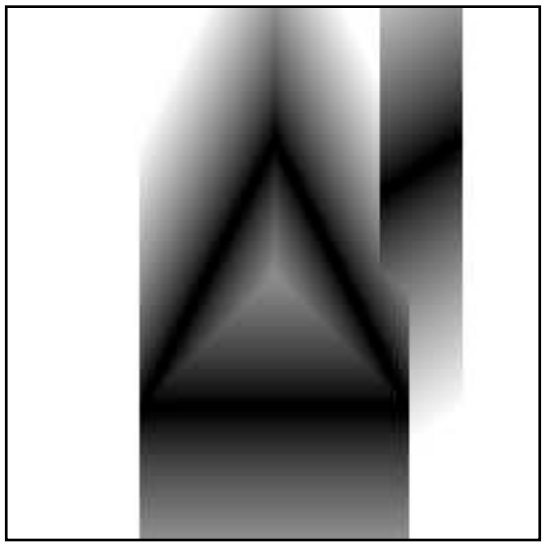
- Parabolas ordered from left to right $y_1 < y_2 < \ldots < y_k$
- At step $j$ add $j$-th parabola to LE of previous ones
  - Compute intersection with rightmost parabola on LE
  - If right of rightmost intersection add parabola to LE
  - If not, remove rightmost parabola and try again

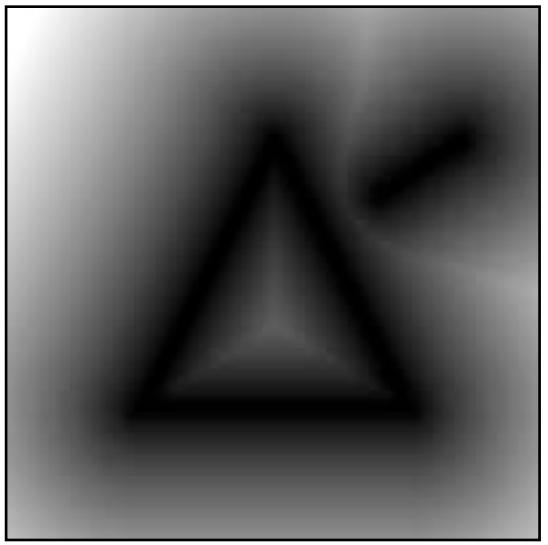


# Application to distance transform







- Sequential min-convolution along dimensions
- Exact computation of distance transform
  - linear time, very fast in practice

# LBP in Bipartite Graphs

- 4-connected grid graph is bipartite
  - Use checkerboard pattern to get partition of $P$ into $(A,B)$
- Alternate updating messages from $A$ and $B$
  - Can update messages "in place"
- Converges to same fixed point with half as many updates, using half as much memory
  - When t is odd (even):

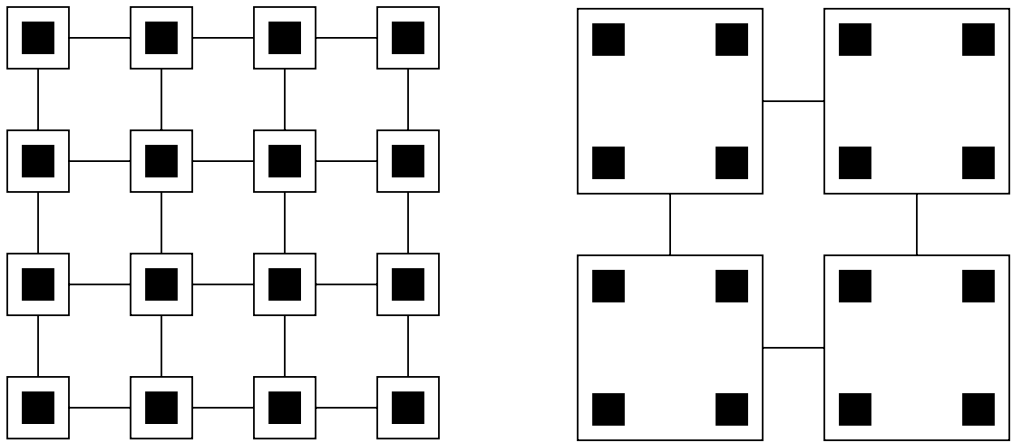
$$\overline{m}^{t}_{p\to q}(f_q) = \begin{cases} m^{t}_{p\to q}(f_q) & \text{if } p \in A \ (p \in B) \\ m^{t-1}_{p\to q}(f_q) & \text{if } p \in B \ (p \in A) \end{cases}$$

# A Multigrid Technique

- Number of message passing iterations T generally proportional to diameter of grid graph
  - Need to propagate information across the whole image
- Use hierarchical approach
  - Previous work with BP does this by changing the graph to a quad-tree with no loops
- We define a hierarchy of problems
  - Each problem has the original graph structure
  - Use messages from one level to initialize the one below

# Problem Hierarchy

- At level $i$ we have a grid graph on blocks of pixels
  - Each block has $\varepsilon \times \varepsilon$ pixels
  - $\varepsilon = 2^i$
- Labeling assigns same label to every pixel in a block



level 0



level 1

# Hierarchical Data Cost

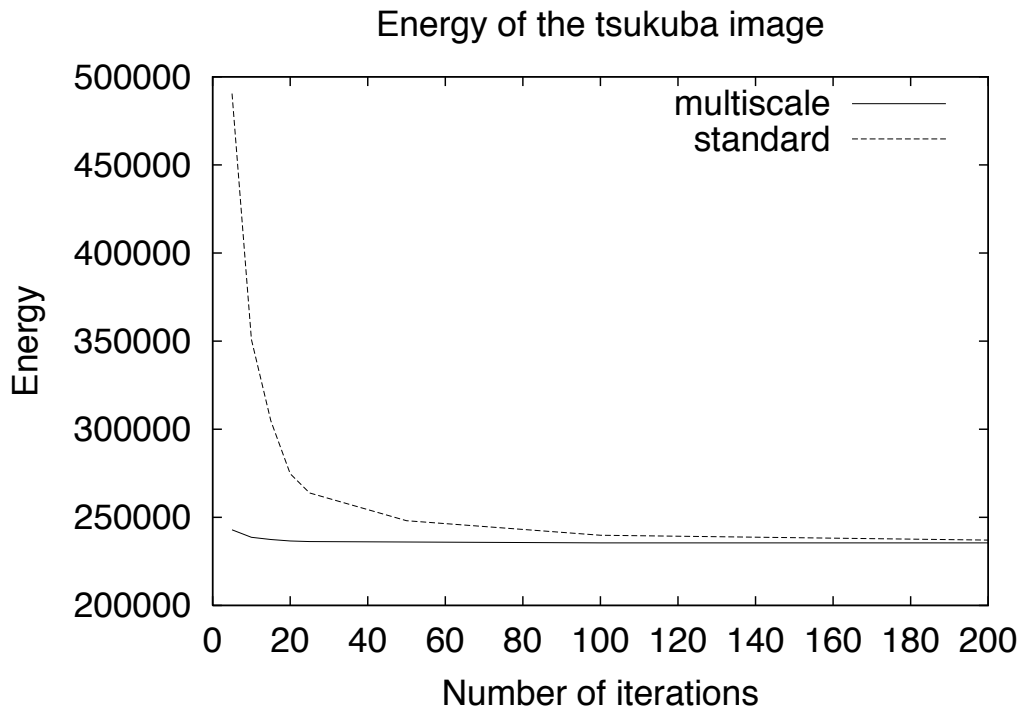
- Finite element approach
- Assigning label $f_b$ to block corresponds to assigning that label to each pixel in the block
  - $D_b(f_b) = \sum D_p(f_b)$
  - Sum data costs in block
  - Recursively compute using 4 data costs from level below
- Captures preference for several labels within a block

# Hierarchical Discontinuity Cost

- Boundary between blocks has length ε
  - Sum discontinuity costs along boundary
- Separation between blocks is ε
  - When measuring derivative of labeling we should divide by separation between blocks
- $V'(f_a - f_b) = \varepsilon V((f_a - f_b)/\varepsilon)$
- Effect of ε depends on form of $V$
  - If $V(x) = |x|$ then $V'(x) = |x|$
  - If $V(x) = x^2$ then $V'(x) = x^2/\varepsilon$

# Hierarchical Algorithm

- Number of levels proportional to log of image size
- Same label set at each level
  - In contrast to classical pyramid methods
- In practice converges after few iterations at each level

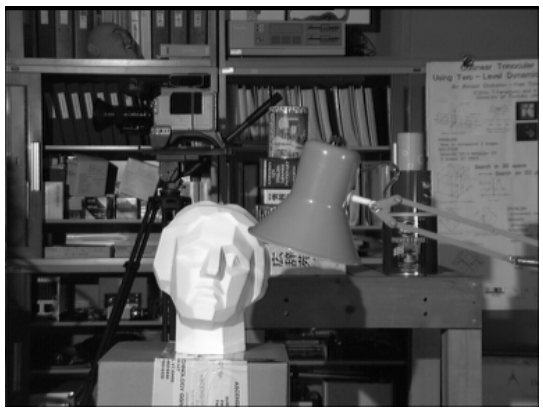


# Some Results

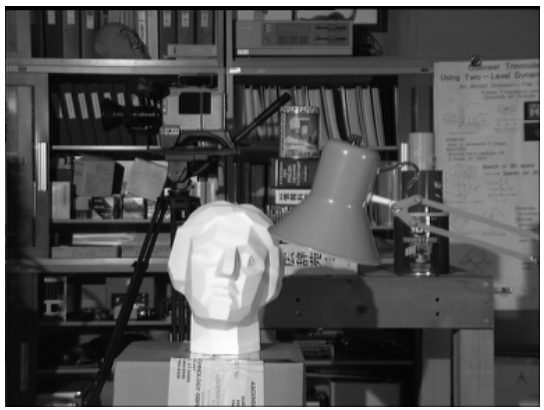
- Image restoration
  - Truncated quadratic discontinuity costs
  - Quadratic data cost with no penalty for masked pixels
- 256 discrete labels, propagating over large areas

# More Results

- Stereo
  - Truncated data and discontinuity costs
  - Runs in under one second for tens of labels (real time with GPU?)
  - Same accuracy as much slower methods



left camera



right camera

disparities