Fast Globally Optimal 2D Human Detection with Loopy Graph Models

Paper by
T.-P. Tian and S. Sclaroff

Slides by A. Vedaldi
Extended tree models

Tree part
\[ U(x_1, \ldots, x_N; I) = \sum_{i \in V} U(x_i; I) + \sum_{ij \in T} U(x_i, x_j; I) \]

Full model
\[ U(x_1, \ldots, x_N; I) = \sum_{i \in V} U(x_i; I) + \sum_{ij \in T} U(x_i, x_j; I) + \sum_{ij \in E} U(x_i, x_j; I) \]
Dynamic programming on trees 1/2

\[
\max_{x_1, x_2, x_3} U(x_1, x_2, x_3) \quad \text{\(O(h^N)\)}
\]

- Decompose maximization

\[
\max_{x_1, x_2, x_3} U(x_1) + U(x_2) + U(x_3) + U(x_1, x_2) + U(x_1, x_3) = \max_{x_1} U(x_1) + \left( \max_{x_2} U(x_2) + U(x_1, x_2) \right) + \left( \max_{x_3} U(x_3) + U(x_1, x_3) \right) = \max_{x_1} U(x_1) + B_2(x_1) + B_3(x_1)
\]

\[
O(N h^2)
\]
Dynamic programming on trees 2/2

\[ B_q(x_k) = \max_{x_q} U_{kq}(x_k, x_q) + U_{T(q)}(X_{T(q)}) \]

\( T(q) \) nodes of the tree rooted at \( q \)

\[ B_q(x_k) = \max_{x_q} U_{kq}(x_k, x_q) + U_q(x_q) + B_t(x_q) + B_l(x_q) + \ldots \]
• Use the tree part to get lower bounds:

\[
LB(\Omega) = \begin{cases} 
U(X), & \Omega = \{X\}, \\
\min_{X \in \Omega} U_{\text{tree}}(X), & \text{otherwise}.
\end{cases}
\]

\[U_{\text{tree}}(X) \leq U(X)\]
Priority queue

\((\Omega_1, \text{LB}(\Omega_1)), (\Omega_2, \text{LB}(\Omega_2)), \ldots, (\Omega_n, \text{LB}(\Omega_n)), \ldots\)

\(\Omega_1 = \Omega_1' \cup \Omega_1''\)

\(\text{LB}(\Omega_1) \leq \text{LB}(\Omega_1')\)

\(\text{LB}(\Omega_1) \leq \text{LB}(\Omega_1'')\)

Unless \(\Omega_1'\) is singleton, because of LB def.:

\(\text{LB}(\Omega_1) = \text{LB}(\Omega_1')\)

\((\Omega_1', \text{LB}(\Omega_1')), (\Omega_2, \text{LB}(\Omega_2)), \ldots, (\Omega_1'', \text{LB}(\Omega_1'')), \ldots, (\Omega_n, \text{LB}(\Omega_n)), \ldots\)

Stopping:

\((\{X\}, U(X)), (\Omega_2, \text{LB}(\Omega_2)), \ldots, (\Omega_n, \text{LB}(\Omega_n)), \ldots\)
Generic partitions

\[ X = (x_1, \ldots, x_N) \in S_1 \times \cdots \times S_N = \Omega \]

Constrain each part to a subset of poses

- Computing bounds

\[
B_q(x_k; S_{T(q)}) = \max_{X_{T(q)} \in S_{T(q)}} U_{kq}(x_k, x_q) + U_{T(q)}(X_{T(q)})
\]

\[ O(TN \Delta^2) \]
$$\Omega = (\{x_1\}, \{x_2\}, \ldots, \{x_k\}, [x'_q, x''_q], [1, h] \ldots [1, h])$$

- always subdivide the \textit{first non-singleton} component
- most of the tree variables are fixed
- tables are computed once
- reduces to repeated range queries

$$B_q(x_k; S_{T(q)})$$
$$= B_q(x_k; [x'_q, x''_q])$$
$$= \arg\max_{x_q \in [x'_q, x''_q]} U(x_k, x_q) + B_t(x_q) + B_l(x_q) + \ldots$$

$$O(TNh^2)$$
$$O(Nh^2 + T)$$
Fast range query

\[ Q(i, j) = \min_{k \in [i, j]} A_k \]

- **Naive implementation**
  - \(O(N)\), \(N\) length of array

- **Constant time implementation**
  - Precompute all range queries of size 2, 4, 8, 16, ...
    - Time: \(O(N \log N)\)
  - Answer a new query
    - Time: \(O(1)\)
• LB quality vs num. iterations T

![Graph showing LB quality vs num. iterations T](image)

• Strength of non-tree edges vs time

![Graph showing Strength of non-tree edges vs time](image)
Qualitative results