Graphical Models for Finding People in Images

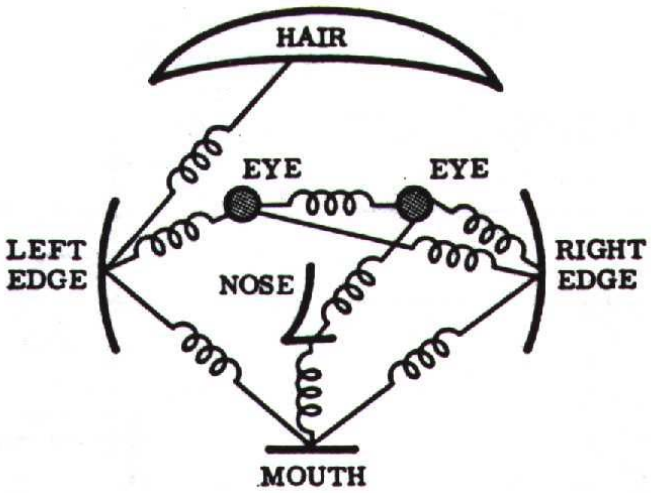
Daniel Huttenlocher
Computer Science Department
January 2005

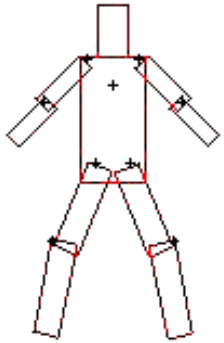
Joint work with Pedro Felzenszwalb and Xiangyang Lan

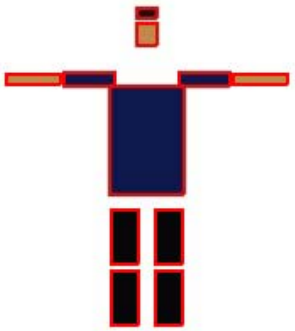
Cornell University
Faculty of Computing and Information Science

Patches in Deformable Configuration

- Dates at least to pictorial structures [FE73]
  - Cost of placing patch at a location
  - Spring-like cost between certain patch pairs
- Can view as undirected graphical model
  - Parts are nodes, connections are edges



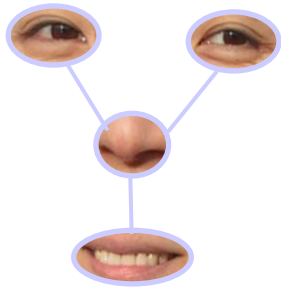




Main Topics

- Tree structured (2D) probabilistic models
  - Capture kinematic relations
- Appearance model (likelihood)
  - Power of soft detection
- MAP estimation vs. sampling
  - Hypothesize and test
- Beyond tree structured models
  - Coordination of limbs
- Hierarchies of models
  - Composing graphical models

Undirected Graphical Model



- Set of parts $V=\{v_1, \ldots, v_n\}$
- Configuration $L=(l_1, \ldots, l_n)$
  - Discrete random variable l_i for each part v_i
- Appearance parameters $A=(a_1, \ldots, a_n)$
  - Model of how each part looks (e.g., templates)
- Spatial relations $S=\{s_{ij} \mid e_{ij} \in E\}$
  - Edge e_{ij}, $(v_i,v_j) \in E$ for neigboring pairs of parts
- Undirected graph $G=(V,E)$
  - First order Markov random field model

Statistical View

- Posterior probability of configuration L given model M=(A,S) and image I

$$p_M(L|I) \propto p_M(I|L)P_M(L)$$

- Likelihood $p_M(I|L)$ of observing image I given configuration L of model M
  - Appearance for fixed configuration
- Prior $p_M(L)$ probability of model M being in configuration L
  - Generally limited to *relative* locations of parts, as absolute location not informative

Estimation Problems

- Detection
  - Likelihood ratio $P_M(I|present)/P_M(I|absent)$
  - Where $P_M(I|present) = \sum_L P_M(L)P_M(I|L)$
- Localization
  - Maximum a posteriori (MAP)

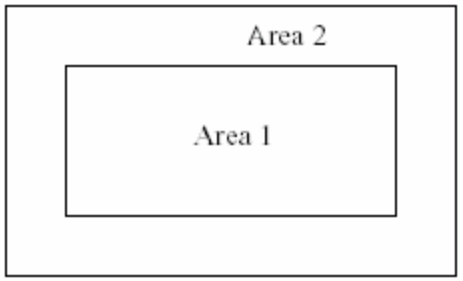
    $\text{argmax}_L P_M(L|I) = \text{argmax}_L P_M(L)P_M(I|L)$
- Supervised learning
  - Maximum likelihood (ML) given pairs (I_j,L_j)

    $S^* = \text{argmax}_S \prod_j P_M(L_j)$

    $A^* = \text{argmax}_A \prod_j P_M(I_j|L_j)$

Form of Likelihood

- Assume $P_M(I|L) = \prod_V g_i(I,l_i)$
  - Appearance factors into product of functions, each dependent on location of a single part
  - Common assumption in part-based approaches
- Simple part appearance model
  - Binary silhouette
- Rectangular region specifying probability of silhouette pixel
  - Interior: probability close to 1
  - Exterior: probability close to 0



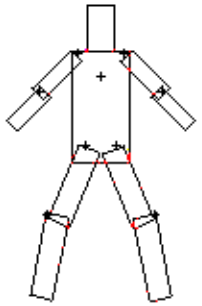
- Likelihood based on pixel counts in image

Soft Detection

- Compute likelihood of image (patch) given model for every location of every part
  - Quality map for each part
  - Make robust to missing parts by truncation or choice of foreground/background probabilities
- Combine maps for all the parts to compute the posterior
  - Efficient for trees and using fast "convolution"
- No detection decisions made about individual parts, only entire configuration

Form of Prior

- Efficient algorithms if graph G acyclic
  - Chains, trees both often used to represent kinematic structure
    - E.g., tree edges correspond to joints of skeleton



- Tree-structured prior $p_M(L)$ factors

$$\frac{\prod_E p_M(l_i, l_j)}{\prod_V p_M(l_i)^{\deg(v_i)-1}}$$

  - Generally no reasonable prior on location of single part, choose to be 1 so drops out

About Acyclic Models

- Tractable algorithms
  - Traditionally $O(m^2n)$ for n parts and m discrete locations per part
  - But O(mn) using distance transform or box sum to compute "convolutions" [FH00,FH05]
- Consistent solutions
  - Chain or tree recursion selects compatible values for neighboring nodes
    - Recursive methods not possible when loops
- Reasonable for modeling people, hands, …
  - Will consider some limitations and extensions

Relation Between a Pair of Parts

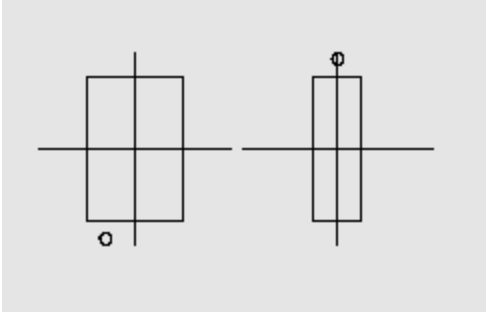
- Spring model of revolute joint
  - Gaussian distribution over transformed part locations

$$p_M(l_i,l_j) \propto \mathcal{N}(T_{ij}(l_i)-T_{ji}(l_j),0,D_{ij})$$

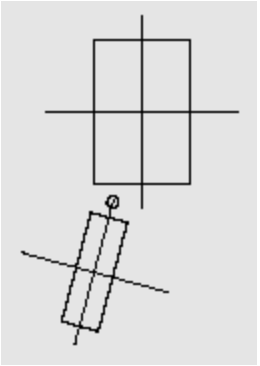
  - T_{ij},T_{ji} capture ideal relative locations of v_i,v_j
    - Fast methods require that $T_{ij}(l_i)$, $T_{ji}(l_j)$ can be discretized on a grid
  - Covariance matrix D_{ij} measures deformation
    - Consider case of diagonal covariance
- Negative log probability is a Mahalanobis distance between part locations (cost)

Relation Between a Pair of Parts

- Part location l_i specifies
  - Location (x_i, y_i)
  - Orientation, o_i
  - Foreshortening along main axis, s_i

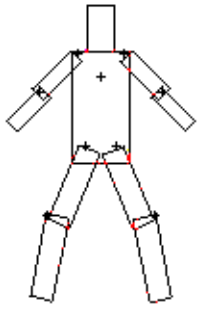


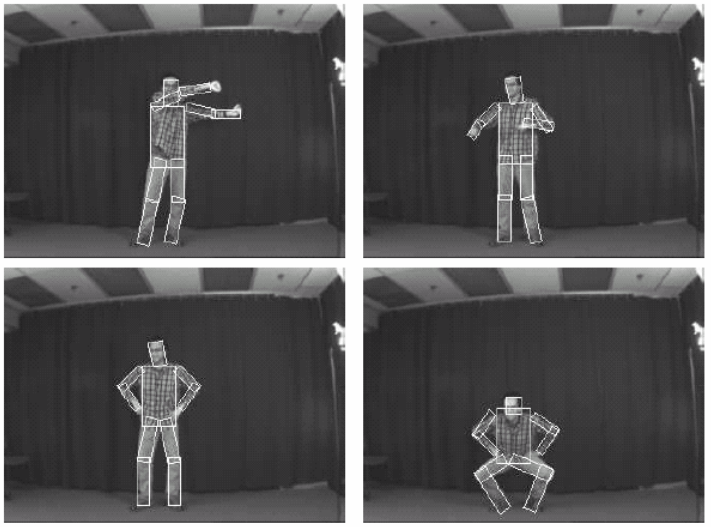
- Distribution $\mathcal{N}(T_{ij}(l_i)-T_{ji}(l_j),0,D_{ij})$
  - This is simple revolute joint model
    - T_{ij} and T_{ji} map each pair of connected parts to common coordinate frame
    - Degree of deviation – range of motion – represented by covariance D_{ij}



Learned Prior from Labeled Examples

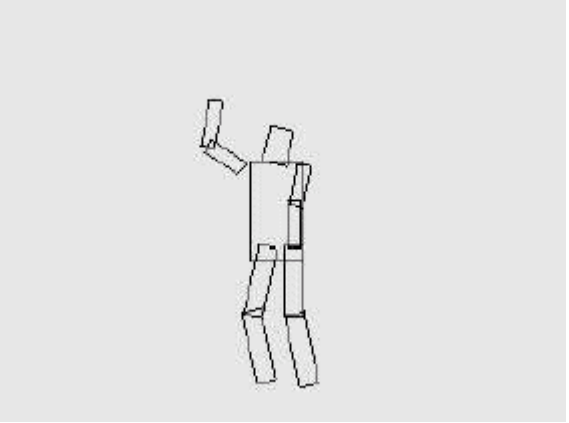
- Training data specifies part locations in image but not connectivity
  - Learn which parts form tree (in ML sense) as well as connection parameters: mean, variance
- Model for relatively wide range of forward facing poses
  - Kinematic structure





MAP Pose Estimation

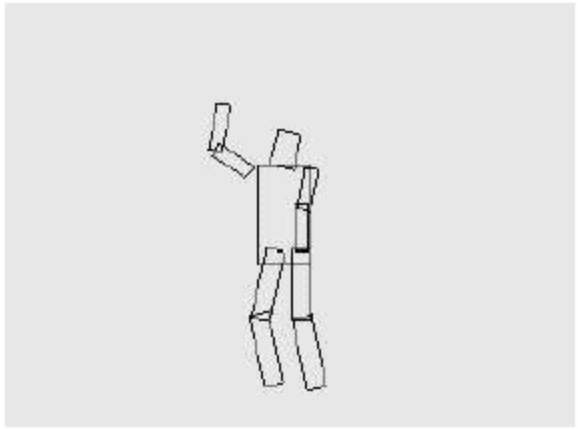
- Fast algorithm using distance transforms to compute $\text{argmax}_L \; P_M(L)P_M(I|L)$
  - Most likely configuration of model in image
- May be multiple good configurations
  - Highest posterior probability not necessarily "best" when using tree prior and silhouettes

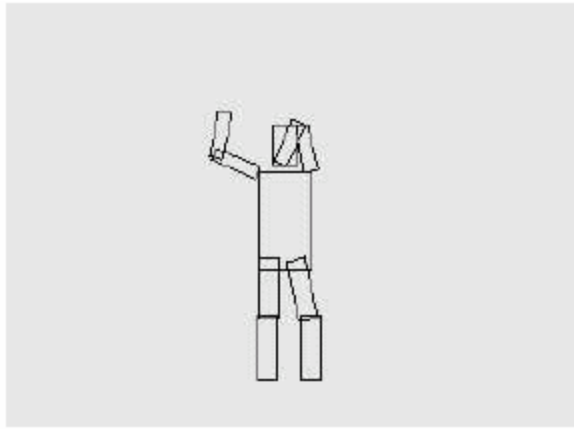


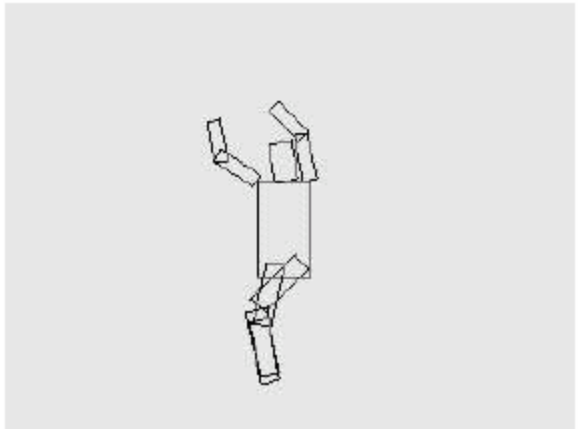
Sampling

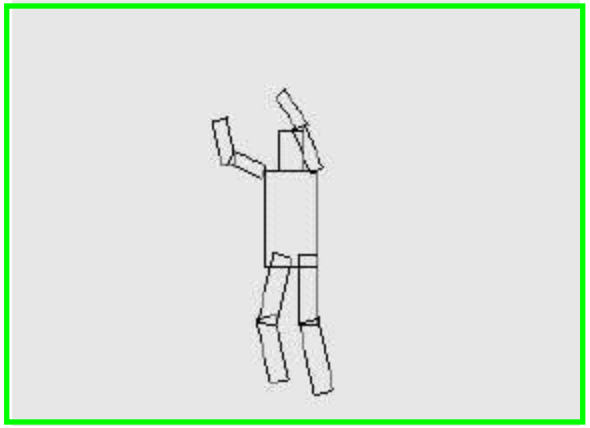
- Hypothesize and test paradigm
  - Postulate configurations with high posterior probability, verify using other means
- Fast algorithm using box sum or FFT to compute factored posterior [FH05]
- Efficiently generate sample configurations
  - For one part sample a location with high posterior probability
  - Given that part location sample a high (conditional) probability location for each child, and so on

Sampling Example



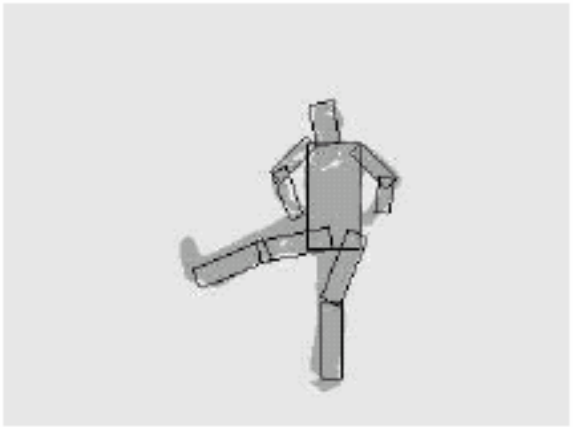


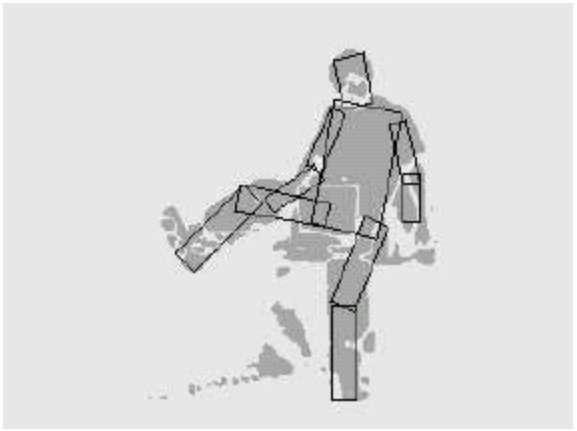




Sampling Results

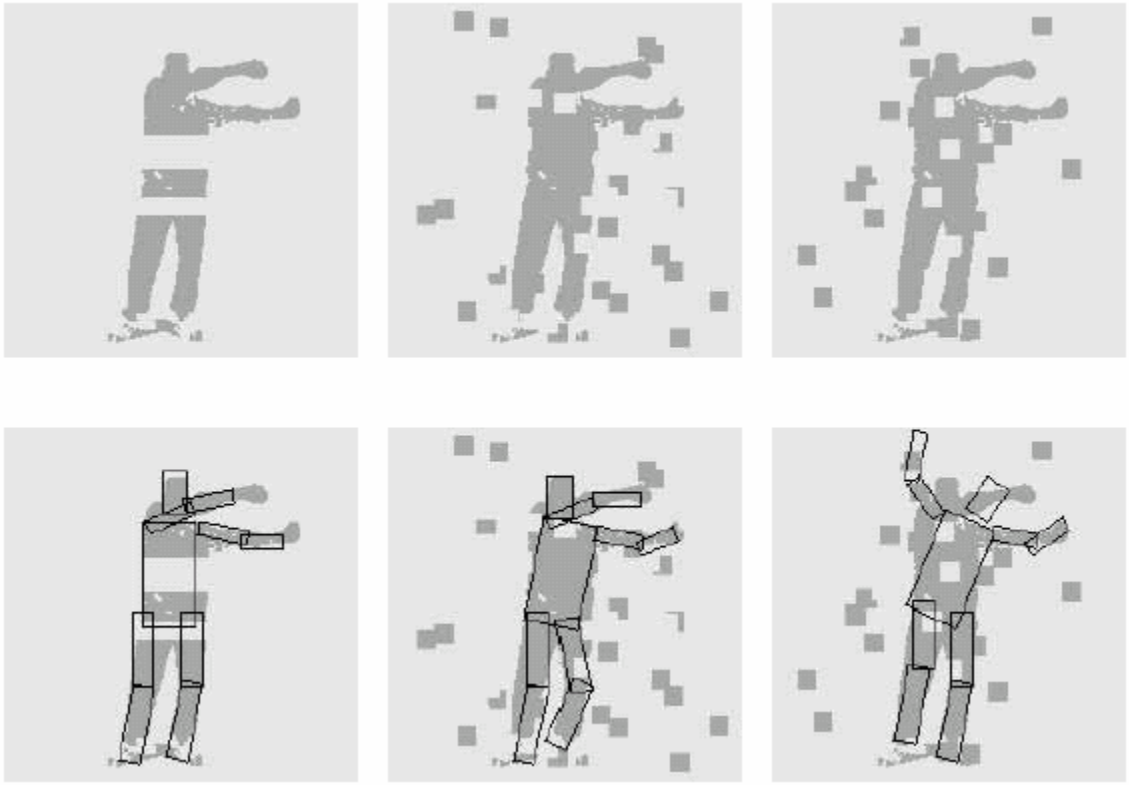
- Pick best match using Chamfer distance





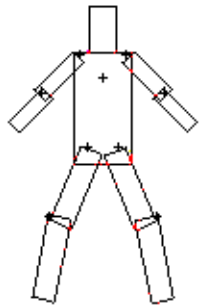
Robustness of Soft Detection

- Parts can be missing or occluded and still be inferred from overall configuration



Beyond Modeling Kinematics

- Coordination of limbs
  - E.g., balance, walking, running, dancing, …
  - Not captured by tree models
- Conditional independence of location of limb given torso does not hold
  - While limbs can move independently for many activities do not
  - Represent such additional spatial dependencies
    - But with computationally tractable models, not simply adding more edges to graph

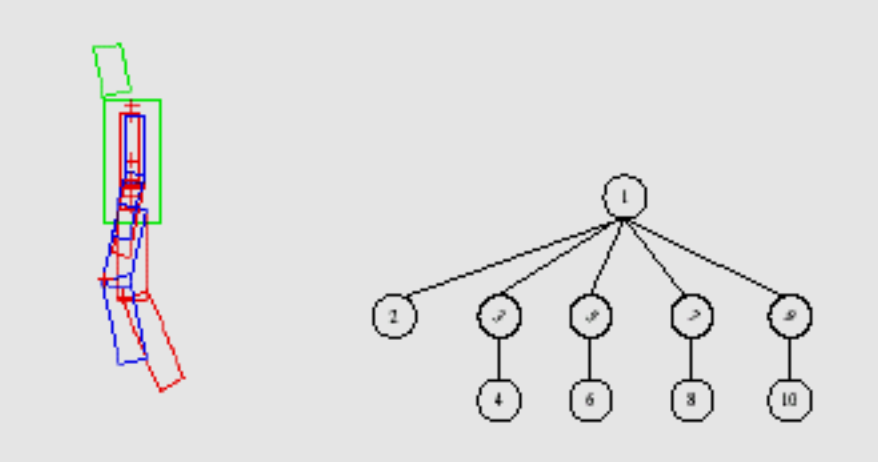


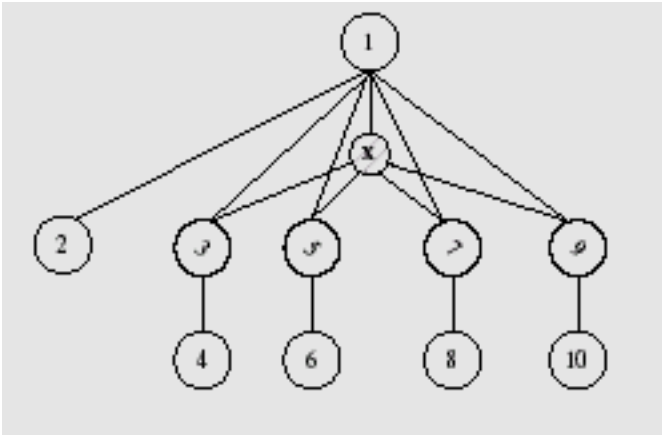
Example: Side View of Walking

- High degree of correlation in orientation of upper arms and legs conditioned on torso
- Rule out approach of simply connecting torso and upper arms/legs into 5 clique
  - Exact inference exponential in clique size
- Analyze dependencies in terms of common factor(s)
  - Factor analysis of siblings yields single underlying orientation parameter
    - Not surprisingly, corresponds to "swing" or extent parameter

A Latent Swing Variable

- Introduce additional variable into model corresponding to common factor
  - Does not correspond to any part
  - Capture dependencies among orientation parameters of torso and four upper limbs





Inference With This Model

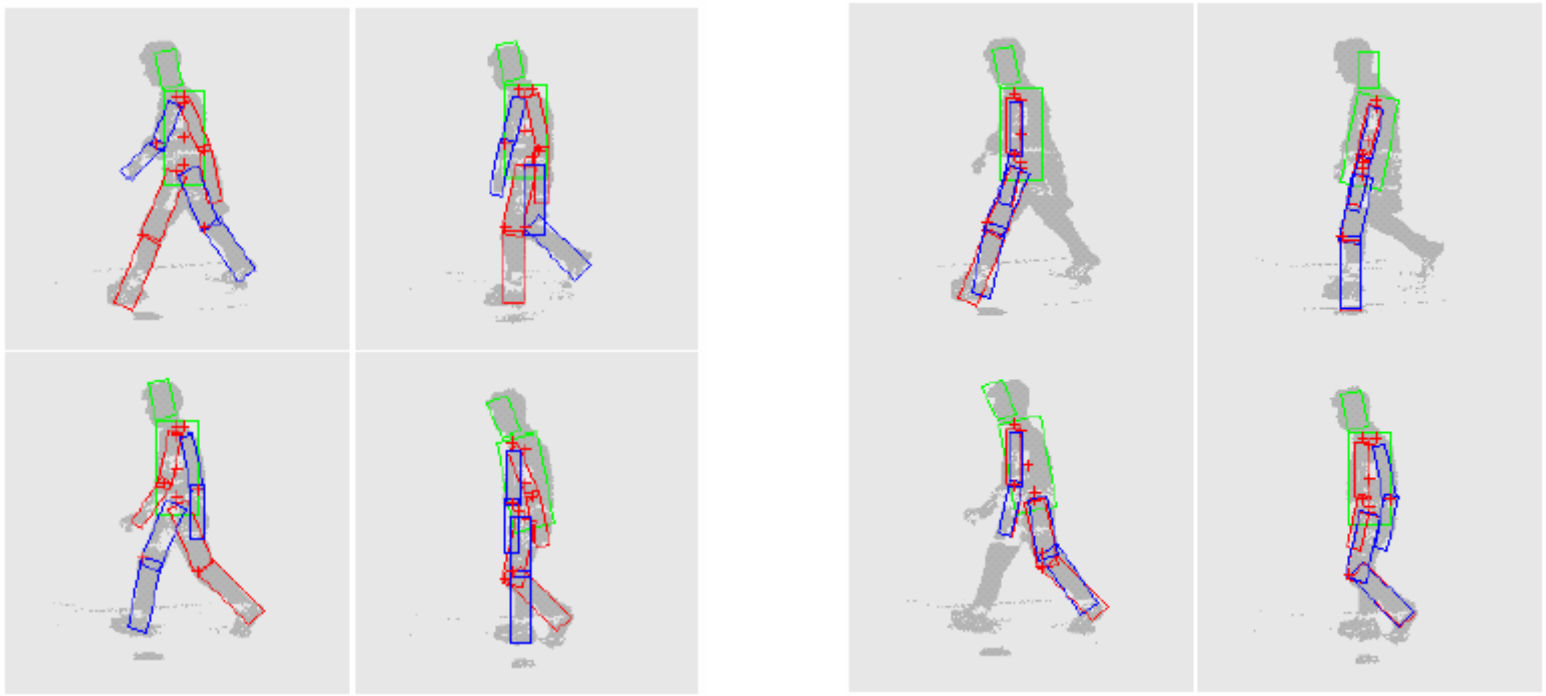
- Each value of latent variable defines a tree
  - Finite set of trees for discrete problem
- Compute MAP estimate by maximizing posterior over these trees
  - Each tree solved efficiently
  - Because sub-problems are all trees and select one, also get consistent solution
- Explicit search over values of swing parameter
  - Can use coarse search to rule out

Learning This Model

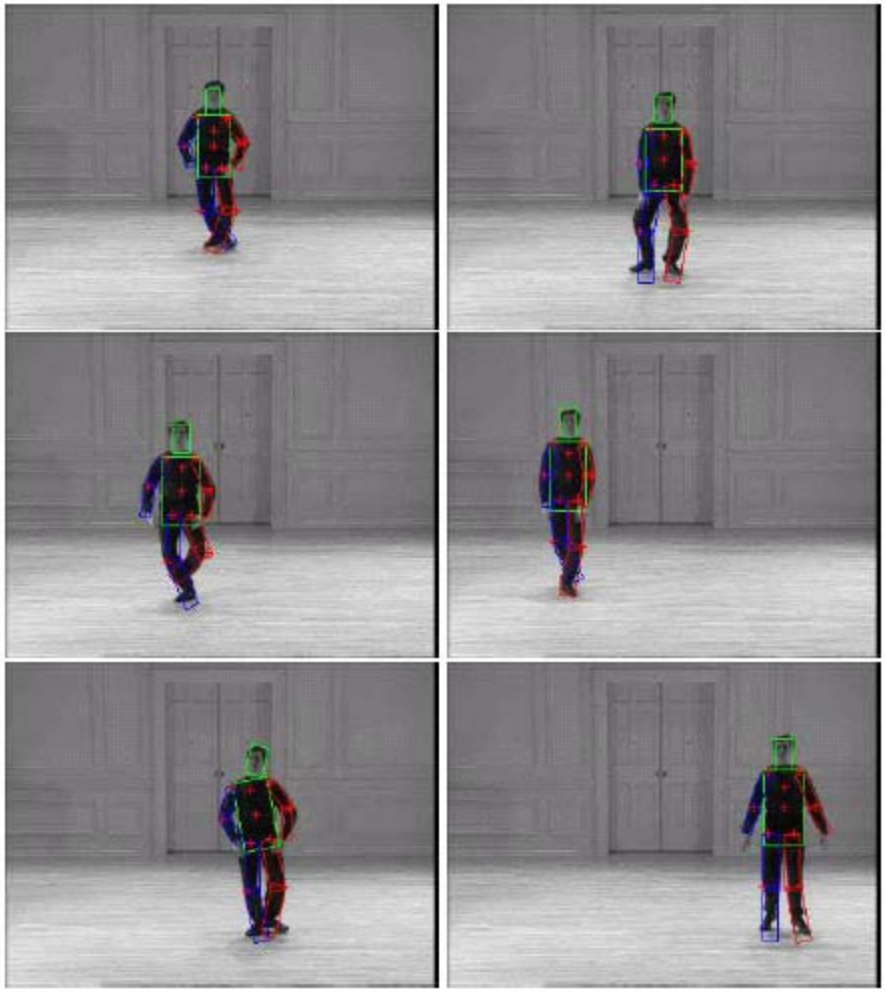
- Limb coordination parameters, such as swing, do not affect kinematic structure
  - Connections between limbs and range of motion – revolute joint – remain unchanged
- Learn tree structured model
  - To capture kinematics
- Use factor analysis to determine which siblings related by common factors
  - Introduce latent coordination variable(s)
  - To capture limb coordination

Simple Experiments

- MAP estimate for images containing side view of person walking
  - Latent variable model vs. tree model

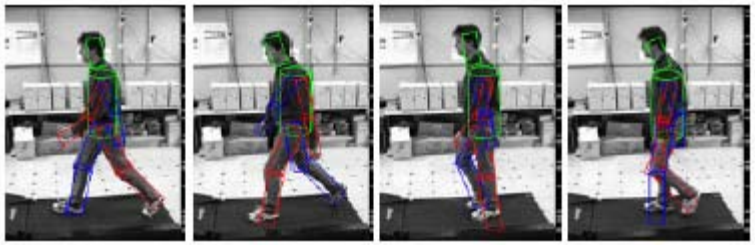


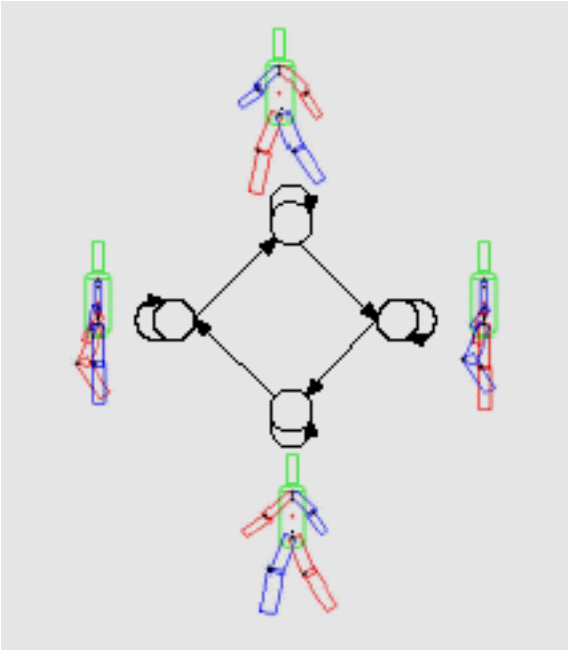
Front View Person Dancing



Composing Graphical Models

- Hierarchies of models
  - For example: HMM for tracking using pictorial structure models as states
  - Observe silhouette at each time
  - Determine which model most likely and parameters aligning that model with image
  - Display aligned model



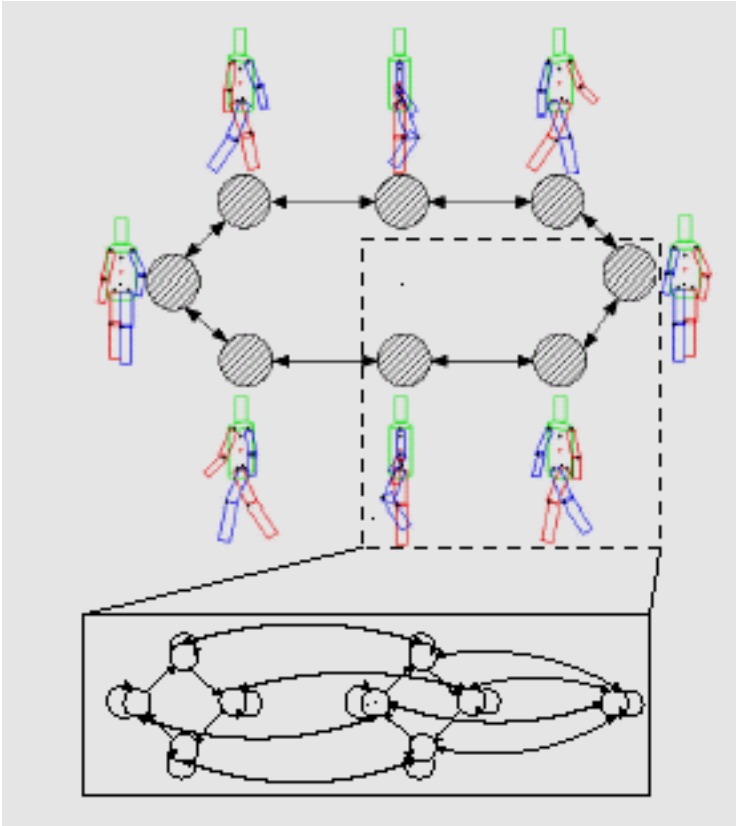


Relatively Efficient Inference

- Generally not easy to combine state model with HMM inference
  - E.g., linear dynamical system (LDS) models, "switching linear state models" [PRM00]
- With tree-like graphical models tractable to compute $P(I|M) = \sum_L P_M(L)P_M(I|L)$
- Or use MAP as an estimate of sum
  - Precisely what needs to be computed to determine which model most likely at each time step

More Complex Temporal Model

- Combine possible viewing directions with 4 state gait model
  - As simplification allow change in viewpoint or gait state, but not both simultaneously
- Approximate inference method to speed up
  - Using "extent" of silhouette to select among models



Example Image Sequence

- Walking with change in viewpoint
  - Automatic selection of view and gait state, alignment of chosen model with image
  - Correctly tracks left vs. right arms and legs through sequence

Summary

- Tree structured (2D) models
  - Capture kinematic structure
- Soft detection
  - Detect entire configuration not individual parts
- Sampling useful for hypothesize and test
- Beyond tree structured models
  - Coordination of limbs using factor analysis and latent variable in graphical model
- Hierarchies of models
  - Graphical models can be composed efficiently