# Learning Objects and Parts in Images

Chris Williams

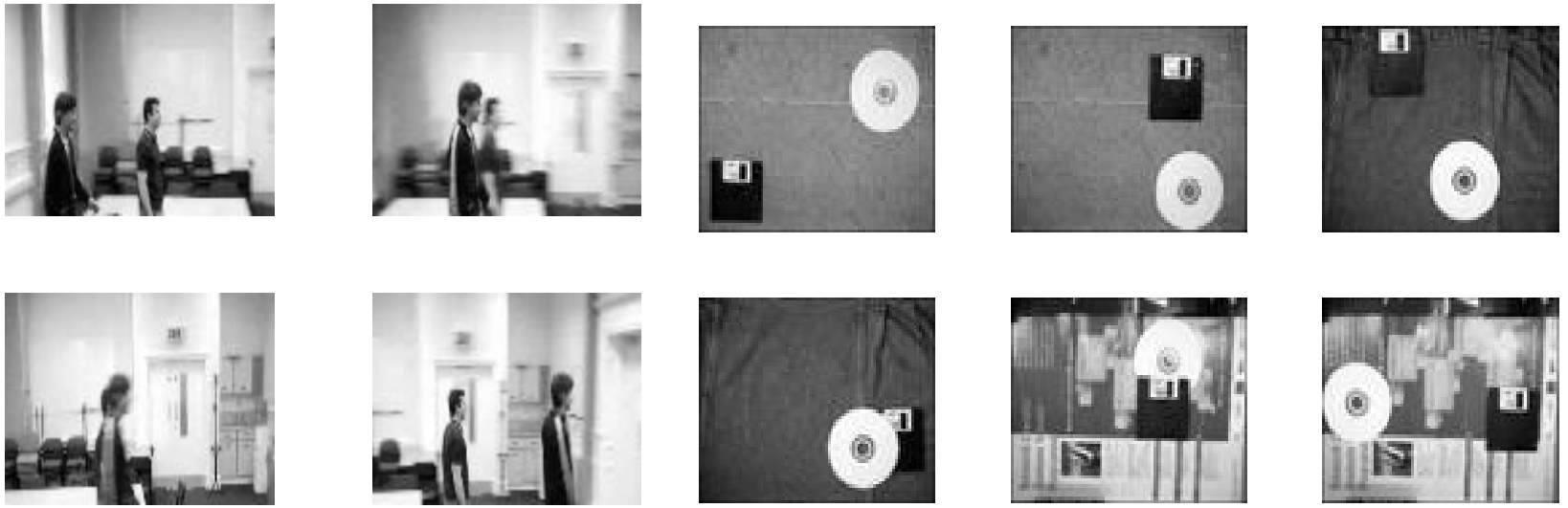

*School of Informatics, University of Edinburgh, UK*

- Learning multiple objects and parts from images
  (joint work with Michalis Titsias)

- Modelling whole/part relationships with Dynamic Trees
  (joint work with Nick Adams, Steve Felderhof, Amos Storkey)

- Other acknowledgements: Geoff Hinton, Rich Zemel

# Learn the Objects



# Motivation

- Our data is images containing multiple objects
- Task is to learn about each of the objects in the images
- With a true generative model each image must be explained by instantiating a model for each of the objects present with the correct instantiation parameters
- This leads to combinatorial explosion: $L$ models with $J$ possible values of the instantiation parameters $\rightarrow O(J^L)$ combinations

- We avoid the combinatorial search by extracting models *sequentially*
- Achieved by using a robust statistical model so that certain parts of the image (e.g. where the other objects are) are modelled by an outlier process; learning by ignoring!
- This method works for images, where the multiple objects combine by *occlusion*
- A simplification of this idea works for fitting mixture models sequentially

# Overview

- Learning One Object
- Coping with Multiple Objects
- Results
- Related work

# Learning One Object

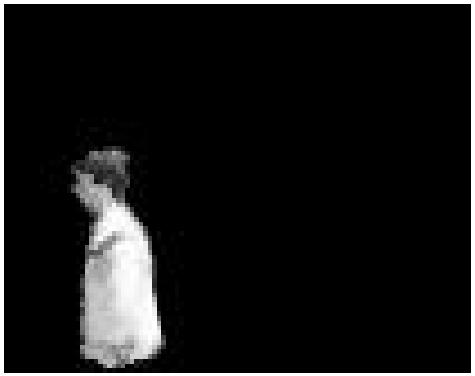
Have to deal with

- foreground/background issue
- transformations of the object

Images are viewed as vectors of length $P$. We learn foreground $\mathbf{f}$, background $\mathbf{b}$ and mask $\pi$; the latter specifies the probability that a pixel is from the foreground or background.

- Foreground/background only

$$p(\mathbf{x}) = \prod_{p=1}^{P} [\pi_p p_f(x_p; f_p) + (1 - \pi_p) p_b(x_p; b_p)]$$



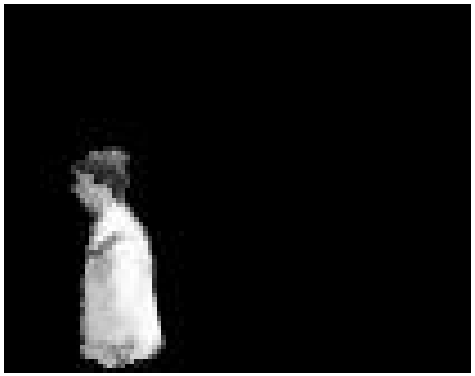
foreground

mask

- Coping with transformations

$$p(\mathbf{x}|T_j) = \prod_{p=1}^{P} [(T_j\boldsymbol{\pi})_p p_f(x_p; (T_j\mathbf{f})_p) + (1 - (T_j\boldsymbol{\pi})_p) p_b(x_p; b_p)]$$

$$p(\mathbf{x}) = \sum_{j=1}^{J} p_j p(\mathbf{x}|T_j)$$



foreground (original)

mask (original)

foreground (transformed)

mask (transformed)

# Fitting the model to data

- $\mathbf{f},\ \mathbf{b},\ \boldsymbol{\pi},\ \sigma_f^2,\ \sigma_b^2$ can be learned by EM
- Model is similar to Jojic and Frey (2001) except that $\boldsymbol{\pi}$ has probabilistic semantics, which means that an exact M-step can be used
- Can also introduce latent variable for moving background

# Coping with multiple objects: previous work

Layered approach:

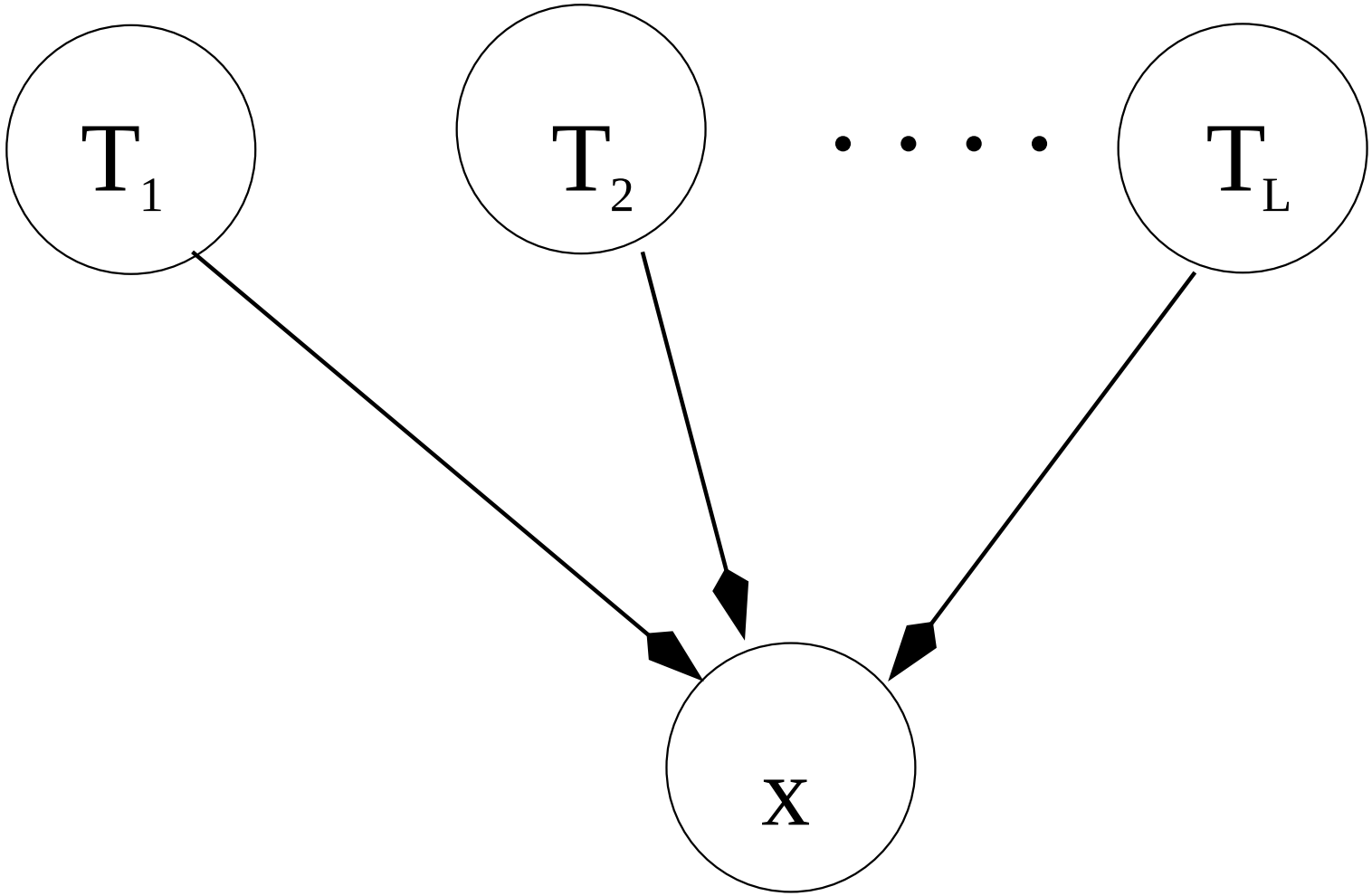
$$p(\mathbf{x}|T_1, \ldots T_L, T_b) = (T_1\boldsymbol{\pi}_1).*N(T_1\mathbf{f}_1, \sigma_1^2)+$$

$$(\mathbf{1} - T_1\boldsymbol{\pi}_1).*(T_2\boldsymbol{\pi}_2).*N(T_2\mathbf{f}_2, \sigma_2^2) + \ldots$$

$$(\mathbf{1} - T_1\boldsymbol{\pi}_1)\ldots.*(\mathbf{1} - T_L\boldsymbol{\pi}_L).*N(T_b\mathbf{b}, \sigma_b^2)$$

where layer 1 is in front of layer 2, ... , layer $L$.

- Each pixel is modelled as a $L+1$ component mixture given $T_1, \ldots, T_L$
- Can't afford to deal with multiple objects exactly due to the combinatorial explosion $O(J^{L+1})$
- Ghahramani (1995) and Jojic & Frey (2001) use variational inference


T1
T2
TL
X

# Coping with multiple objects: our approach

- We take a sequential approach, modelling *one object at a time*
- Need to *robustify* foreground and background models due to occlusion.

$$p_f(x_p; f_p) = \alpha_f N(x_p; f_p, \sigma_f^2) + (1 - \alpha_f) U(x_p)$$

$$p_b(x_p; b_p) = \alpha_b N(x_p; b_p, \sigma_b^2) + (1 - \alpha_b) U(x_p)$$

  - Both foreground and background can be occluded by other objects
  - Ordering now less important
  - Cf work by Black and colleagues (e.g. Black and Jepson, 1996)
- A simple algorithm tries random starting positions in order to try to find multiple objects. However, we have found that this works poorly and a greedy method works much better.

# The Greedy Method

- Once an object has been identified in an image it is removed (cut out) and then we learn the next object by applying the same algorithm

- Assume we have learned one model already to give $\mathbf{f}_1,\ \boldsymbol{\pi}_1$

- For each image $\mathbf{x}$ use the responsibilities $p(T_{i_1}|\mathbf{x})$ to find the most likely transformation $i_1^*$.

- Let $r_{f_1,p}^{i_1^*}$ be the foreground responsibility for pixel $p$ in image $\mathbf{x}$ using transformation $i_1^*$

$$r_{f,p}^{i_1^*} = \frac{\alpha_f N(x_p; (T_{i_1^*}\mathbf{f}_1)_p, \sigma_f^2)}{\alpha_f N(x_p; (T_{i_1^*}\mathbf{f}_1)_p, \sigma_f^2) + (1-\alpha_f)U(x_p)}$$

- Define $\boldsymbol{\rho}_1 = (T_{i_1^*}\boldsymbol{\pi}_1) \,.*\, \mathbf{r}_{f_1}^{i_1^*}$

- A pixel $p$ that is cut out has $(\boldsymbol{\rho}_1)_p \simeq 1$

- This means that an image in which some pixels of the learned object are occluded only has the foreground pixels cut out

- The second stage of the greedy algorithm optimizes a lower bound on the log likelihood, where each pixel $p$ is weighted by $(\mathbf{1} - \boldsymbol{\rho}_1)_p$

# Whole Algorithm

1. Learn the background and infer the most probable transformation $j_b^n$ for each image $\mathbf{x}^n$.
2. Initialize the vectors $\mathbf{z}_0^n = \mathbf{1}$ for $n = 1, \ldots, N$
3. For $\ell = 1$ to $L$
   - Learn the $\ell^{th}$ object parameters $\{\mathbf{f}_\ell, \boldsymbol{\pi}_\ell, \sigma_\ell^2\}$ by maximizing $F_\ell$ using EM algorithm, where

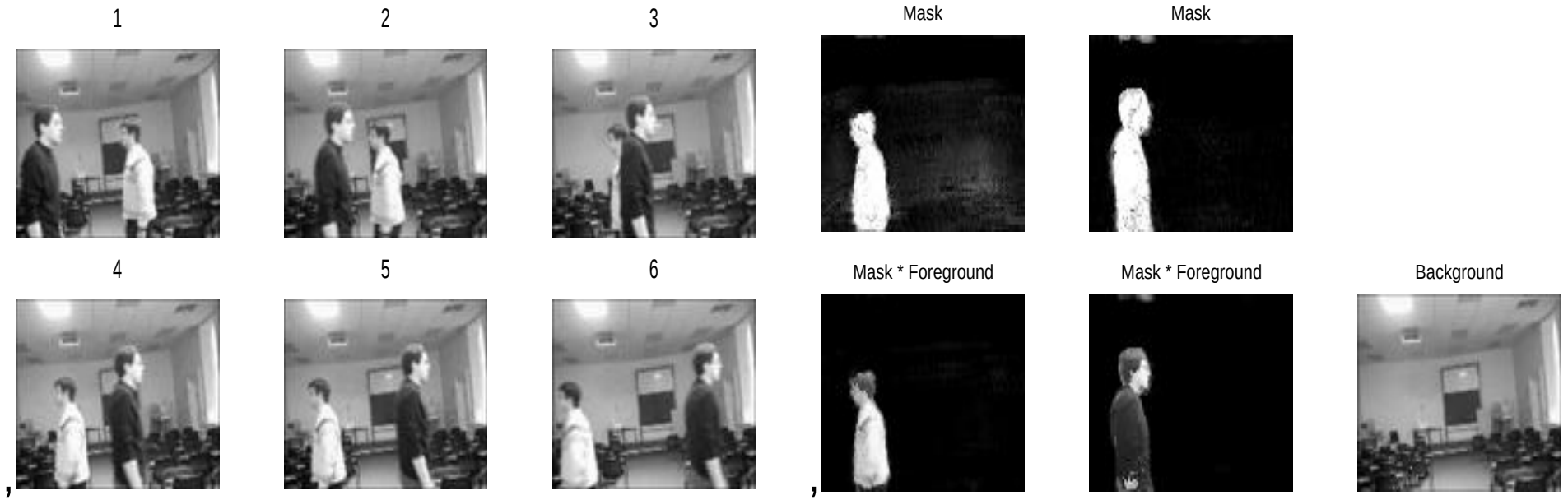
$$F_\ell = \sum_{n=1}^{N} \sum_{j_\ell=1}^{J_f} Q^n(j_\ell) \Big\{ \sum_{p=1}^{P} (\mathbf{z}_{\ell-1}^n)_p \log[(T_{j_\ell}\boldsymbol{\pi}_\ell)_p p(x_p; (T_{j_\ell}\mathbf{f}_\ell)_p) + (1 - T_{j_\ell}\boldsymbol{\pi}_\ell)_p p_b(x_p; (T_{j_b^n}\mathbf{b})_p)] - \log Q^n(j_\ell) \Big\}.$$

   - Infer the most probable transformation $\{j_\ell^n\}$ and update the weights $\mathbf{z}_\ell^n = \mathbf{z}_{\ell-1}^n . * \overline{\boldsymbol{\rho}}_\ell^n$

# Results

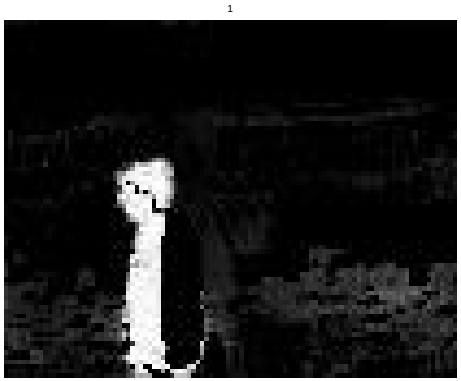
Data

Results



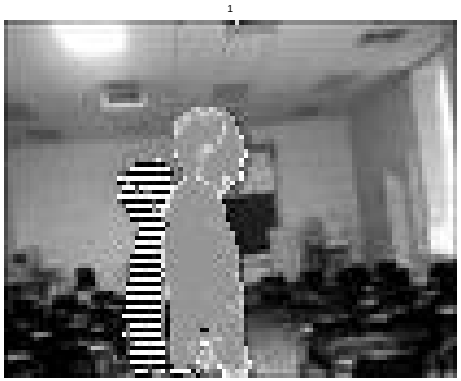

- Consider two people comoving—what happens?


data




mask1 * foreground_resp1


shaded area “removed”




mask2 in position

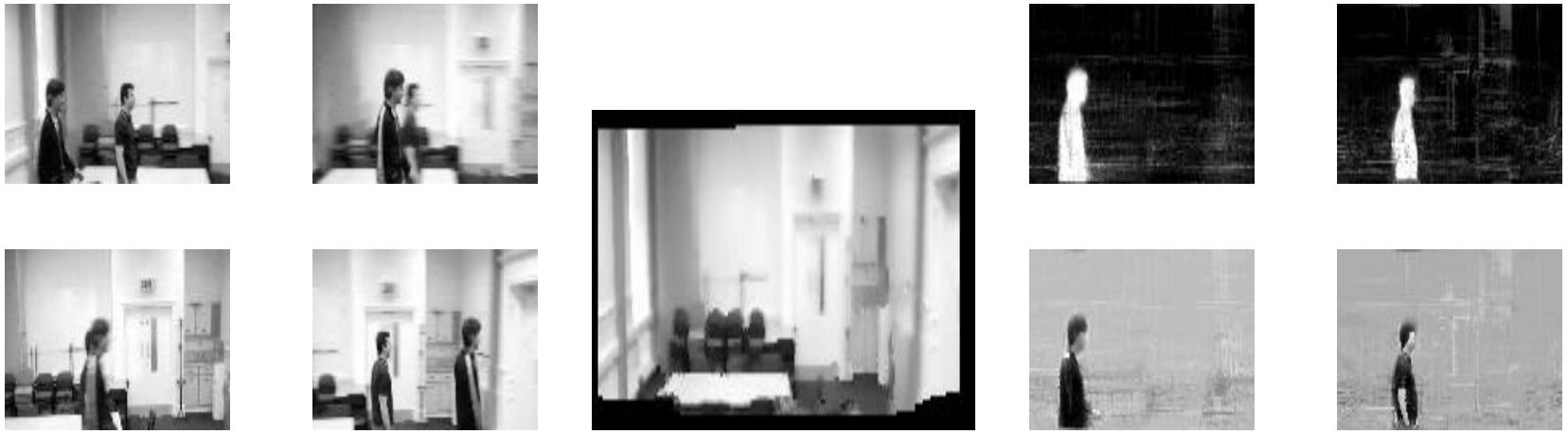
# Frey and Jojic Video Sequences



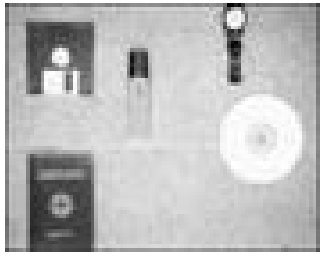
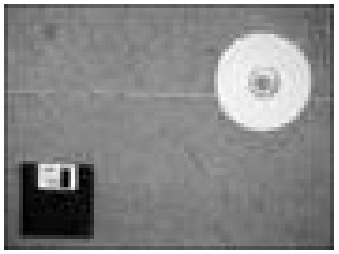
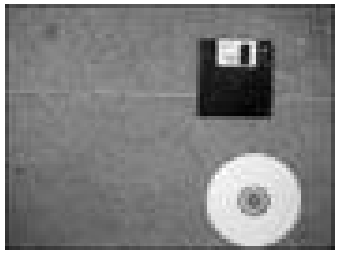
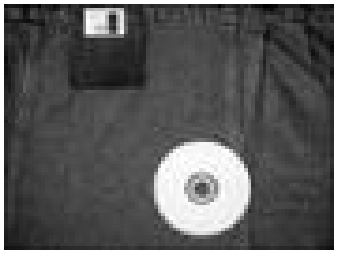
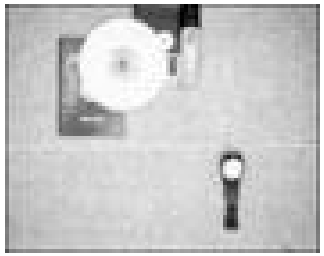
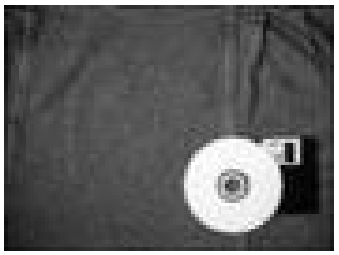
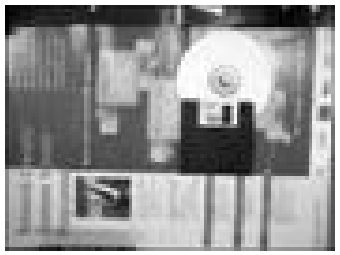
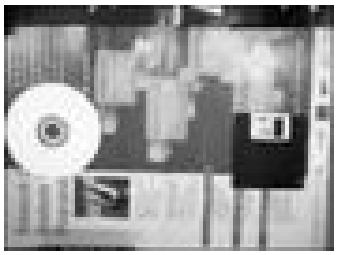
mask1 * foreground_resp1

mask1 * foreground_resp1

# 2 Objects and Moving Background



# Further Examples

# Related work

- Reminiscent of sequential PCA algorithms (deflation) where a PC is identified, and then that component is subtracted out from the input; but here we *mask* out pixels that have already been explained

- Shams and von der Malsburg (1999) obtained candidate parts by matching images in a pairwise fashion, trying to identify corresponding patches in the two images. These candidate patches were then clustered.
  - S/vdM have $O(N^2)$ complexity (pairwise comparison of images)
  - They need to remove background from consideration
  - Their data is synthetic CAD-type models, and is designed to eliminate complicating factors such as background, surface markings etc

- Computer vision approaches e.g. Wang and Adelson (1994), Tao et al (2000) find layers by clustering optical flow vectors. Our method can be applied to unordered collections of images, and is not limited when flow information is sparse

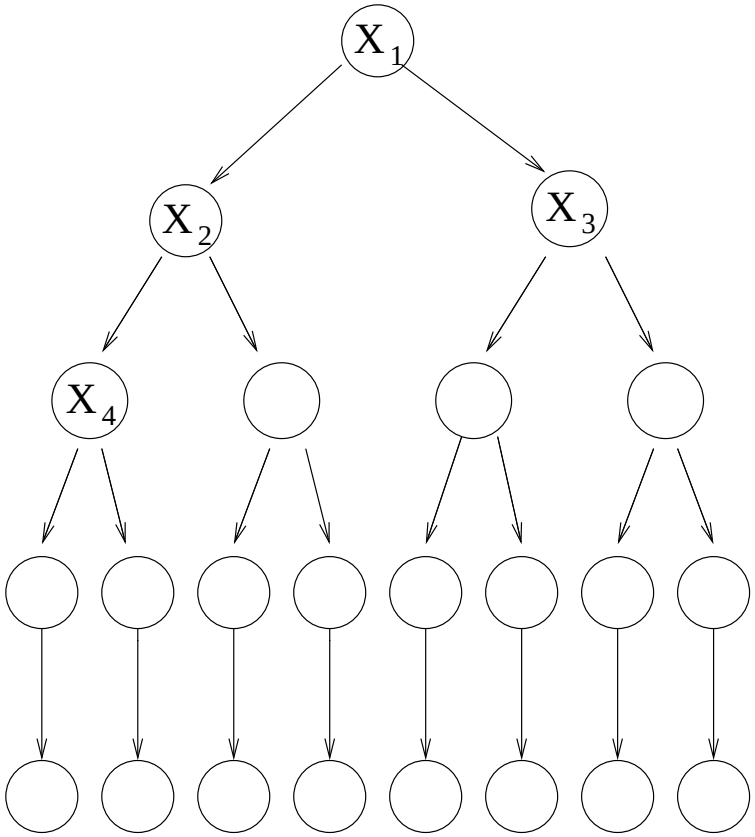
# Summary, Future Work

- The sequential approach works, making use of the combination-by-occlusion regularity
- Can deal with many objects/parts
- Finding parts of articulated objects
- Representing the relationships between parts

# Dynamic Trees

- Need to represent parts and wholes and their relationships
- Parse-tree like structures for images are appealing
- Dynamic belief network structure where children choose their parents

# Tree Structured Belief Networks

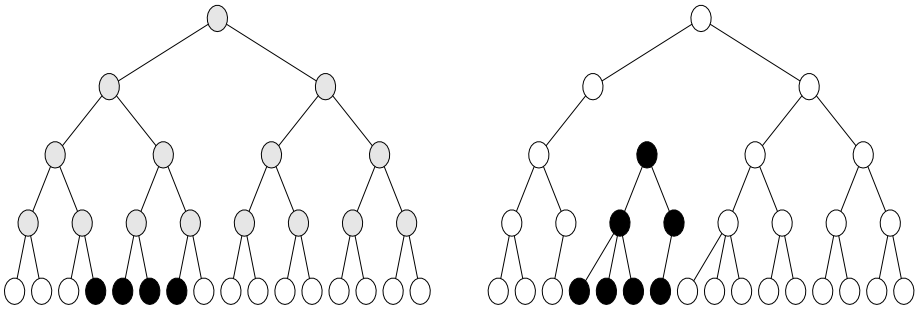



$$P(X_1, \ldots X_m) = P(X_1)P(X_2|X_1)P(X_3|X_1)P(X_4|$$

- Conditional probability tables (CPTs)
- Generating label images from a TSBN
- Important: only leaf nodes are observed (cf multiscale/wavelets)
- Bouman and Shapiro (1994), Laferté et al (1995), Willsky et al (1990s)

# Dynamic Tree Image Models

Use the basic TSBN strategy, but specify a prior over the tree structure $Z$

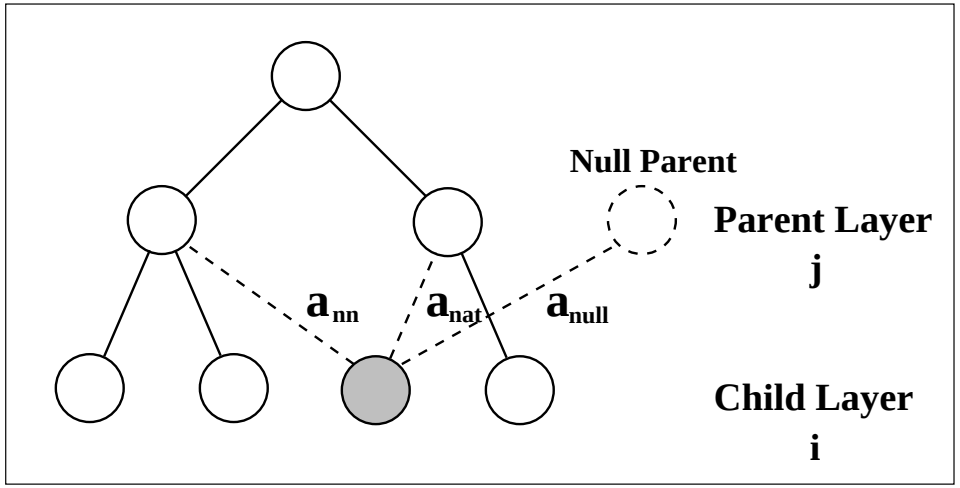


- Unbalanced trees prevent “blockiness”
- Disconnections enable creation of objects at an appropriate scale, parse-tree like
- Visible (leaf) nodes $X_v$, hidden nodes $X_h$

$$P(Z, X_v, X_h) = P(Z)P(X_v, X_h|Z)$$

- Related work: Credibility Nets (Hinton, Ghahramani, Teh)

# Specifying the Prior




$z_{ij}$ is a binary variable denoting the connection between parent $j$ and child $i$ with $P(z_{ij} = 1) = e^{a_{ij}} / \sum_k e^{a_{ik}}$

# Inference and Learning in DTs

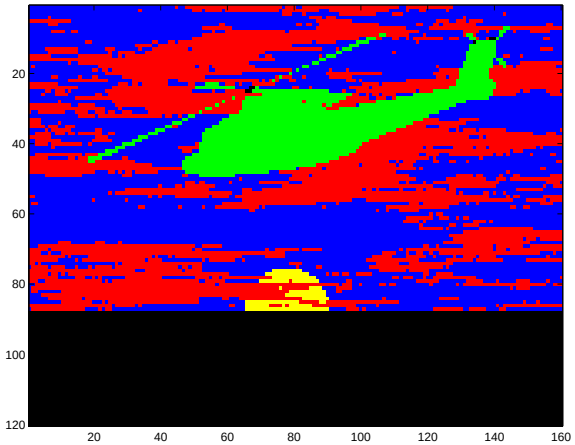
The goal is to obtain the posterior $P(Z, X_h|X_v)$, or the posterior marginal $P(Z|X_v)$, or $P(X_v)$

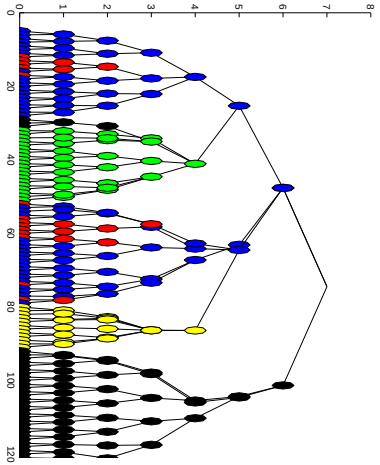
Two main approaches:

- Markov Chain Monte Carlo (MCMC)
- Variational inference

Parameters defining the affinities and CPTs can be learned from training data via mean field EM

# Pixel labelling task




slice from MAP tree

# Where Next?

- DT example uses relatively simple parent-child propagation of labels, position, but could be extended to parents with slots/fillers matching appropriate children
- Combine this with learning of parts from first part of talk to give learning of hierarchical object models