

Greedy Learning of Multiple Objects in Images using Robust Statistics and Factorial Learning

Chris Williams, Michalis Titsias and Moray Allan

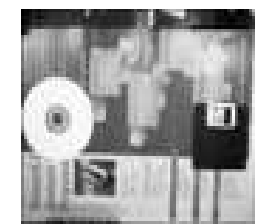
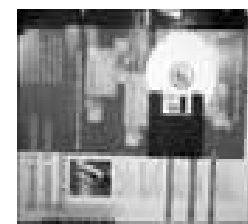
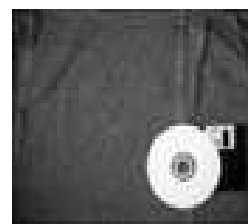
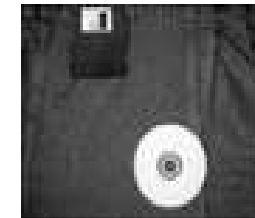
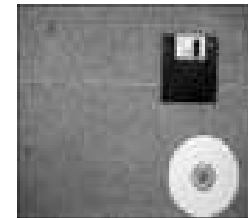
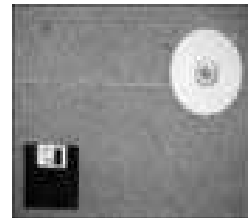


School of Informatics, University of Edinburgh, UK

PASCAL 101 Visual Classes Challenge

- Organizers: Luc van Gool (Leuven/Zurich), Chris Williams (Edinburgh), Andrew Zisserman (Oxford)
- Recognition of objects from a large number of visual object classes in realistic scenes (i.e. not pre-segmented objects)
- Where as well as what
- Issues: intra-class variability, variation in size and pose, partial occlusion
- By end of 2004: database prepared
- Challenge workshop: April 2005?

Learn the Objects



Overview

- Learning One Object
- Coping with Multiple Objects
- Results
- Aside: Sequential fitting of mixtures
- Speeding it up using tracking
- Learning parts
- Fast learning using affine-invariant features
- Related work

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*
- The method can be speeded up for image *sequences* using (approximate) tracking

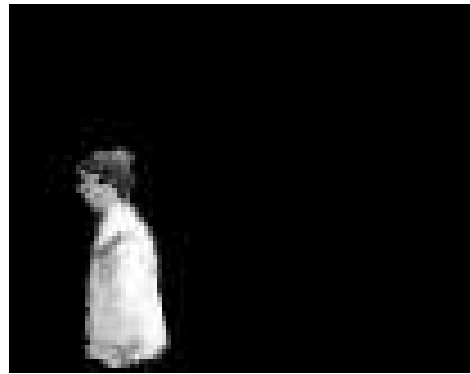
Learning One Object

Have to deal with

- foreground/background issue
- transformations of the object

Images are viewed as vectors of length P . We learn foreground f , background b and mask π . Each element of π is in $[0, 1]$ and specifies the fraction of the pixel's intensity that comes from foreground

Sprite: a graphics term for a “cardboard cutout” model having a shape and appearance



foreground



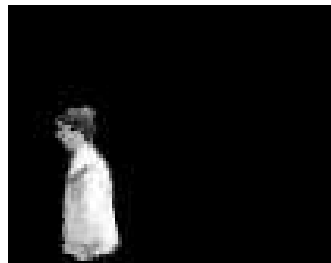
mask

- Foreground/background only

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

- Coping with transformations

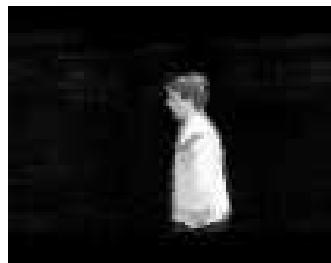
$$p(\mathbf{x}|T_j) = \prod_{p=1}^P [(T_j\boldsymbol{\pi})_p p_f(x_p; (T_j\mathbf{f})_p) + (1 - (T_j\boldsymbol{\pi})_p) p_b(x_p; b_p)]$$



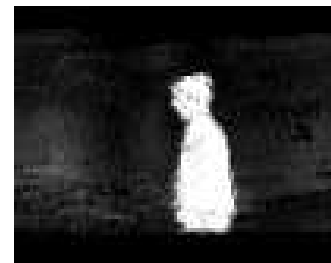
foreground (original)



mask (original)



foreground (transformed)



mask (transformed)

Overall model

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

J can be very large. For translations FFT tricks can be used to speed computation.

Fitting the model to data

- \mathbf{f} , \mathbf{b} , $\boldsymbol{\pi}$, σ_f^2 , σ_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
- Cf GTM, but use *knowledge* to constrain mapping from latent variables (transformations) to image appearances

Coping with multiple objects

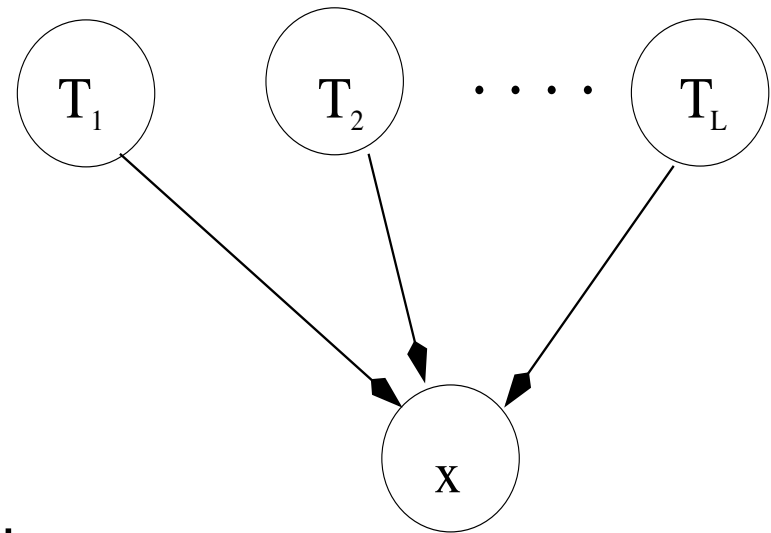
$$p(\mathbf{x}|T_1, \dots, T_L) = (T_1 \pi_1) \cdot * N(T_1 \mathbf{f}_1, \sigma_1^2) +$$

$$(1 - T_1 \pi_1) \cdot * (T_2 \pi_2) \cdot * N(T_2 \mathbf{f}_2, \sigma_2^2) + \dots$$

$$(1 - T_1 \pi_1) \dots * (1 - T_L \pi_L) \cdot * N(\mathbf{b}, \sigma_b^2)$$

Note that layer 2 is in front of layer 1, etc.

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



,

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
 - 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, π_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^*

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

$$\boldsymbol{\rho}_1 = (T_{i_1^*} \boldsymbol{\pi}_1) \cdot * \mathbf{r}_{f_1}^{i_1^*}$$

- 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 the following expression over \mathbf{f}_2 , $\boldsymbol{\pi}_2$, $\sigma_{f_2}^2$, \mathbf{b} and σ_b^2

$$p(\mathbf{x}|T_{i_1^*}, T_j) = \prod_{p=1}^P [(\boldsymbol{\rho}_1)_p N(x_p; (T_{i_1^*} \mathbf{f}_1)_p, \sigma_{f_1}^2) + (1 - \boldsymbol{\rho}_1)_p (T_j \boldsymbol{\pi}_2)_p p_f(x_p; (T_j \mathbf{f}_2)_p)$$

$$+ (1 - \boldsymbol{\rho}_1)_p (1 - T_j \boldsymbol{\pi}_2)_p p_b(x_p; b_p)]$$



data



mask1 * foreground_resp1



shaded area "removed"



mask2 in position

Results

Data

1



2



3



4



5



6



Results

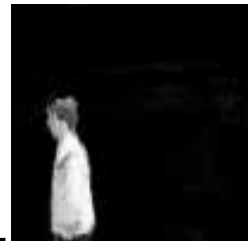
Mask



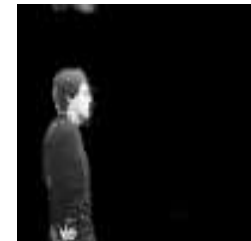
Mask



Mask * Foreground



Mask * Foreground



Background



- Consider two people comoving—what happens?

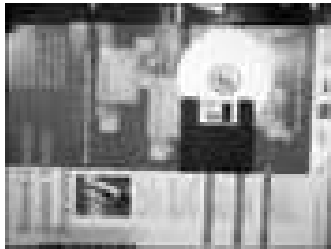
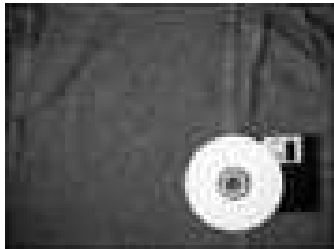
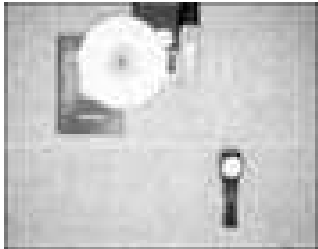
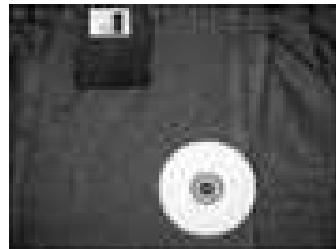
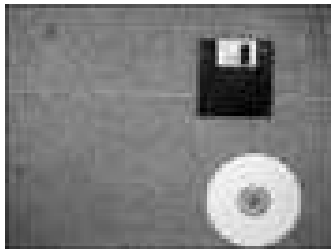
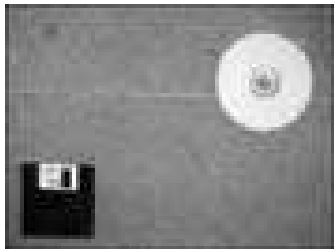
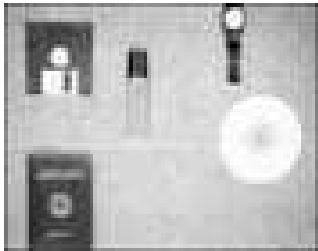
Frey and Jojic Video Sequences

- Relative depth of the layers can be obtained by considering different depth orderings in F&J model, having learned masks, appearances and transformations greedily

2 Objects and Moving Background



Further Examples



Sequential Fitting of Gaussian Mixtures

- Learn one cluster by ignoring the others. Then remove this cluster from consideration and learn another one, until you wish to stop
- Remaining data is explained by an “crud catcher”

$$p(\mathbf{x}) = \alpha N(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) + (1 - \alpha)U(\mathbf{x})$$

Learning a Second Gaussian

- Say we learned a Gaussian $N(\mathbf{x}; \mu_1^*, \Sigma_1^*)$

$$r_1^n = \frac{\alpha^* N(\mathbf{x}^n; \mu_1^*, \Sigma_1^*)}{\alpha^* N(\mathbf{x}^n; \mu_1^*, \Sigma_1^*) + (1 - \alpha^*) U(\mathbf{x}^n)}$$

- Maximize a *constrained* log likelihood $L_2 = \sum_{n=1}^N z_1^n \log P(\mathbf{x}^n)$ with $z_1^n = 1 - r_1^n$
- Boosting interpretation

Whole algorithm

Initialize: $z_0^n = 1$ for all n , $\pi_i = 0$ for all $i = 1 \dots J$

for $j = 1$ to J

Initialize μ_j, Σ_j and set $\pi_j = \alpha(1 - \sum_{i=1}^{j-1} \pi_i)$

Get $\{\pi_j^*, \mu_j^*, \Sigma_j^*\}$ by applying EM algorithm to

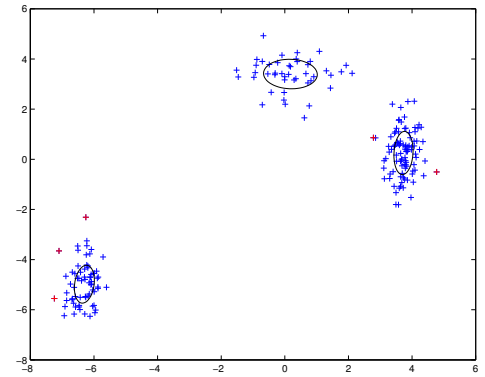
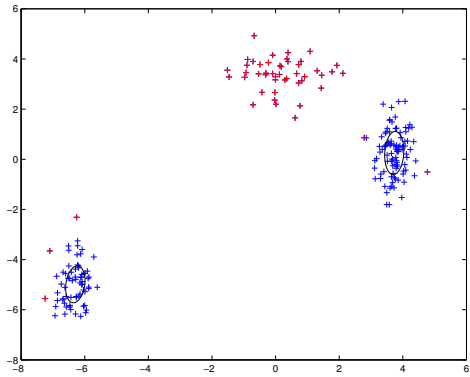
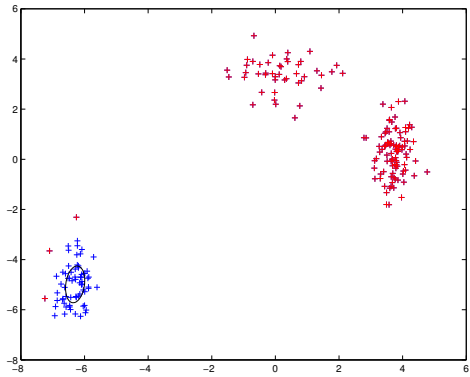
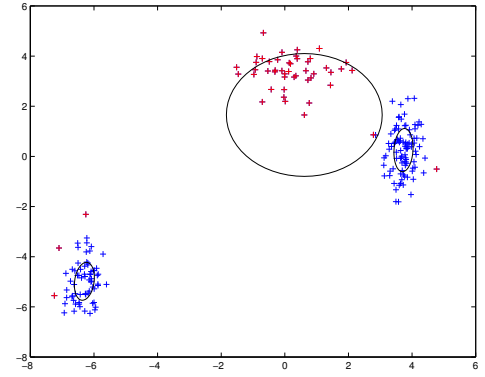
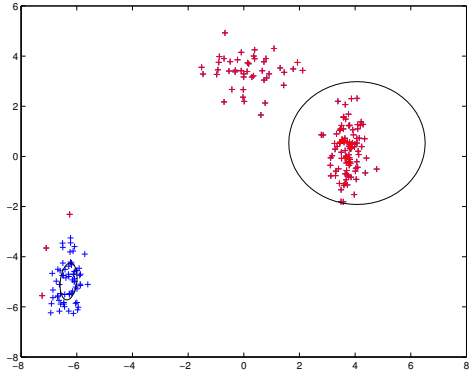
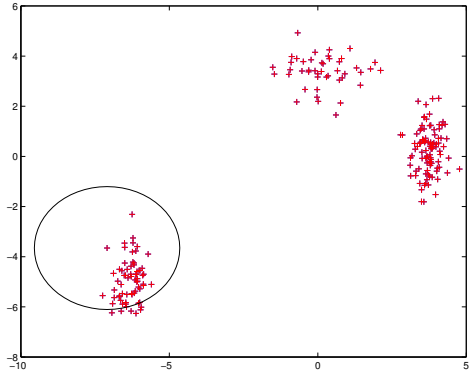
$$L_j = \sum_{n=1}^N z_{j-1}^n \log \left\{ \pi_j N(\mathbf{x}^n; \mu_j, \Sigma_j) + (1 - \sum_{i=1}^j \pi_i) U(x^n) \right\}$$

$$z_j^n = \frac{(1 - \sum_{i=1}^j \pi_i) U(x^n)}{\sum_{i=1}^j \pi_i N(\mathbf{x}^n; \mu_i, \Sigma_i) + (1 - \sum_{i=1}^j \pi_i) U(x^n)}$$

end for

output $\sum_{j=1}^J \pi_j^* N(\mathbf{x}, \mu_j^*, \Sigma_j^*)$

Can terminate when (almost) all datapoints are explained

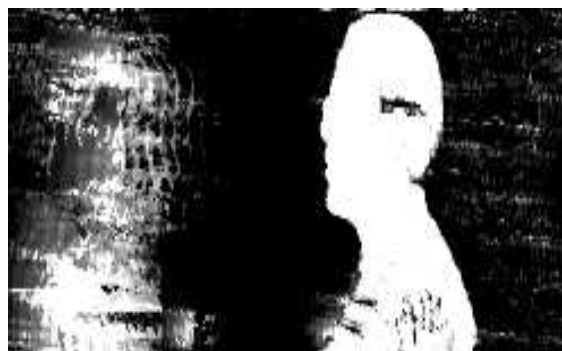


Speeding it up with tracking

- For image *sequences*, it is wasteful to repeatedly search the whole image for an object when its inter-frame motion is small
- First track background, then subtract this out and track foreground objects
- Initialize b to first frame, consider small motions (15×15) window to find best transformation, then update b , and so on
- Once b is learned, delete these pixels from all images and focus on foreground objects
- Speed up: From 80 hours to 3 minutes



$t = 1$



$t = 10$



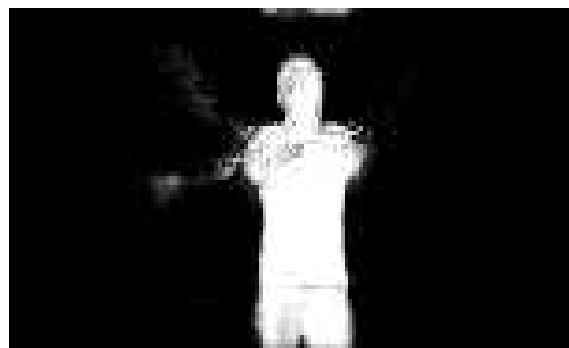
$t = 20$



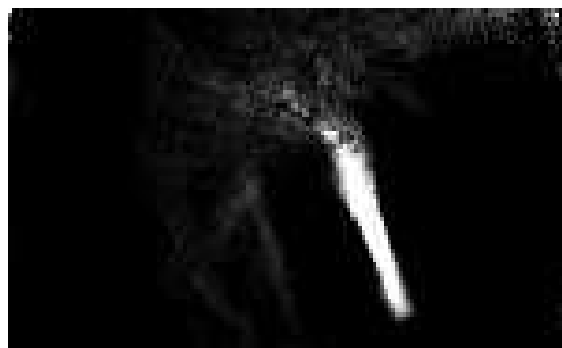
Learning Parts



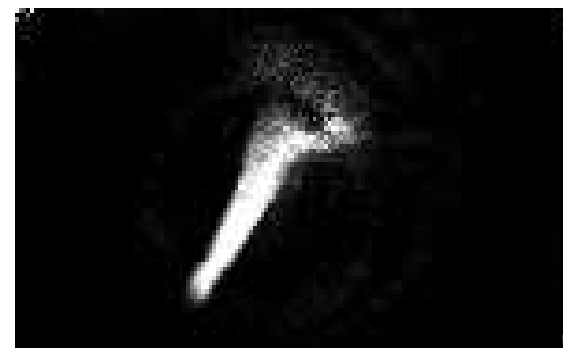
- Use both rotations and translations
- Use 15×15 window of translations and 23 rotations at 2° spacing



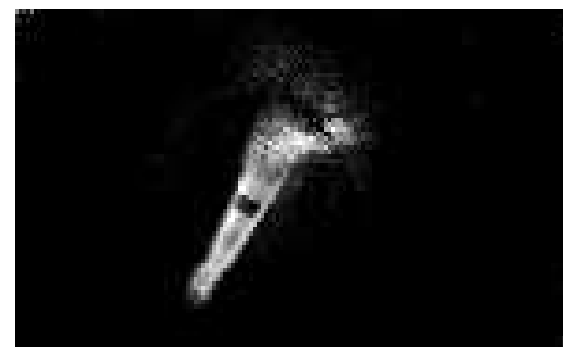
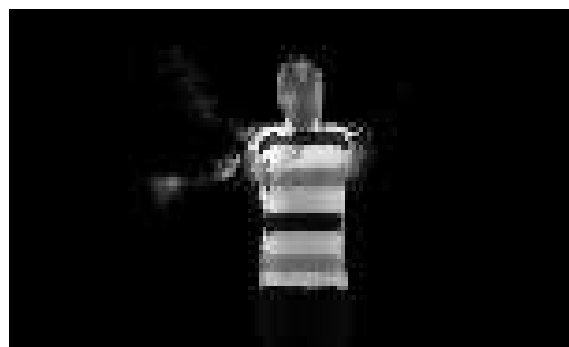
head/torso



left arm



right arm



Fast learning using affine-invariant features



Affine-invariant features: e.g. Schmid & Mohr (1997), Lowe (1999). We use Lowe's SIFT features (128-d)

Feature Matching between Frames



Match by distance in feature space, only accept matches where there is a single good candidate

Grouping features into objects

- Use RANSAC to find distinct motion clusters in each pair of frames
- For each feature through time, generate a vector of motion cluster labels
- Run k -means style algorithm to find a class centre corresponding to each object, allocate each feature to a class
- Videos

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
- Computer vision approaches e.g. Wang and Adelson (1994), Irani et al (1994) 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
- 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

Summary

- The sequential approach works, making use of the combination-by-occlusion regularity
- Sequential methods can also be used, e.g. for fitting Gaussian mixture models, or for sequences (HMMs)
- Can be speeded up by tracking for image sequences
- Learning works for articulated parts
- Tracking with “smart” features to initialize search

References

- C. K. I. Williams and M. K. Titsias, *Greedy Learning of Multiple Objects in Images using Robust Statistics and Factorial Learning*, Neural Computation 16(5) 1039-1062, 2004.
- M. K. Titsias, C. K. I. Williams, *Fast Unsupervised Greedy Learning of Multiple Objects and Parts from Video*, Proc. Generative-Model Based Vision Workshop, June 2004.