

Learning Sprites

Chris Williams



School of Informatics, University of Edinburgh, UK

What is a sprite?

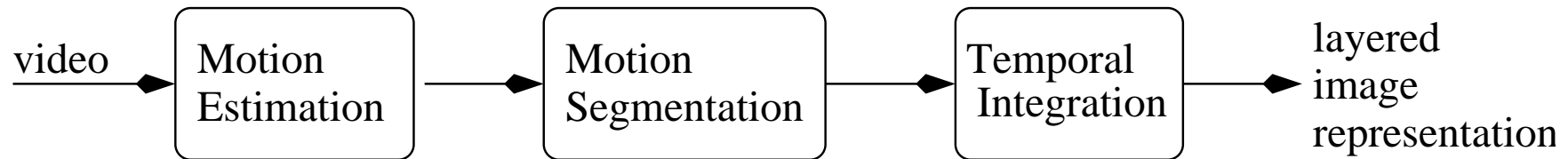
- A graphics term for a “cardboard cutout” model having a shape and appearance
- We will see how to frame the problem of learning sprites from data as a probabilistic modelling/machine learning problem

Overview

- Wang and Adelson (1994)
- Frey and Jojic (2001)
- Williams and Titsias (2004)

Lots of work in the computer vision literature, Wang and Adelson (1994) is an early reference

Layered Image Representation



- Motion estimation using optic flow data (square regions)
- Affine motion segmentation (k -means clustering in affine parameter space plus region splitter, region filter)
- Optic flow estimates and segmentations are iteratively refined
- Temporal integration by inverse warping of regions and median filtering
- Depth ordering is obtained in a verification stage

Uses of the Layered Representation

- Image Coding
- Mid-level language for sequences—coherent moving objects, depth ordering, occlusion, object tracking
- Sequence synthesis

Strengths/weaknesses

- Optic flow estimates are poor in regions of low texture, or if the frame rate is low relative to the motion
- Can handle affine motions
- Lack of a probabilistic framework for the problem

Frey and Jojic: Flexible Sprites

- Learning One Object
- Learning Many Objects
- Results

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

- Foreground/background only

$$\mathbf{x} = \pi. * \mathbf{f} + (1 - \pi). * \mathbf{b} + \text{noise}$$

cf α -matte in graphics



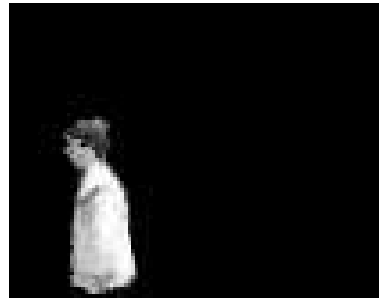
foreground



mask

- Coping with transformations

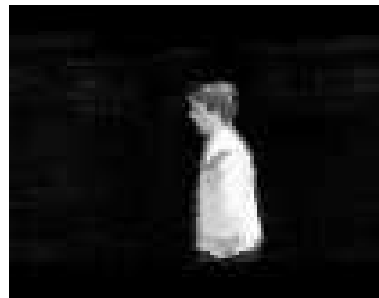
$$\mathbf{x} = T\boldsymbol{\pi} * T\mathbf{f} + T(1 - \boldsymbol{\pi}) * \mathbf{b} + \text{noise}$$



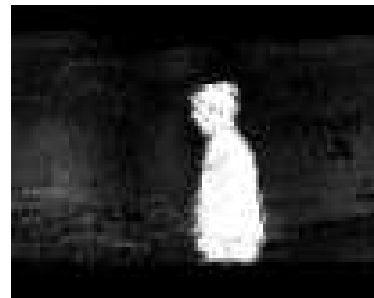
foreground (original)



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.

Coping with multiple objects

$$\mathbf{x} = T_2 \pi_2 . * T_2 \mathbf{f}_2 +$$

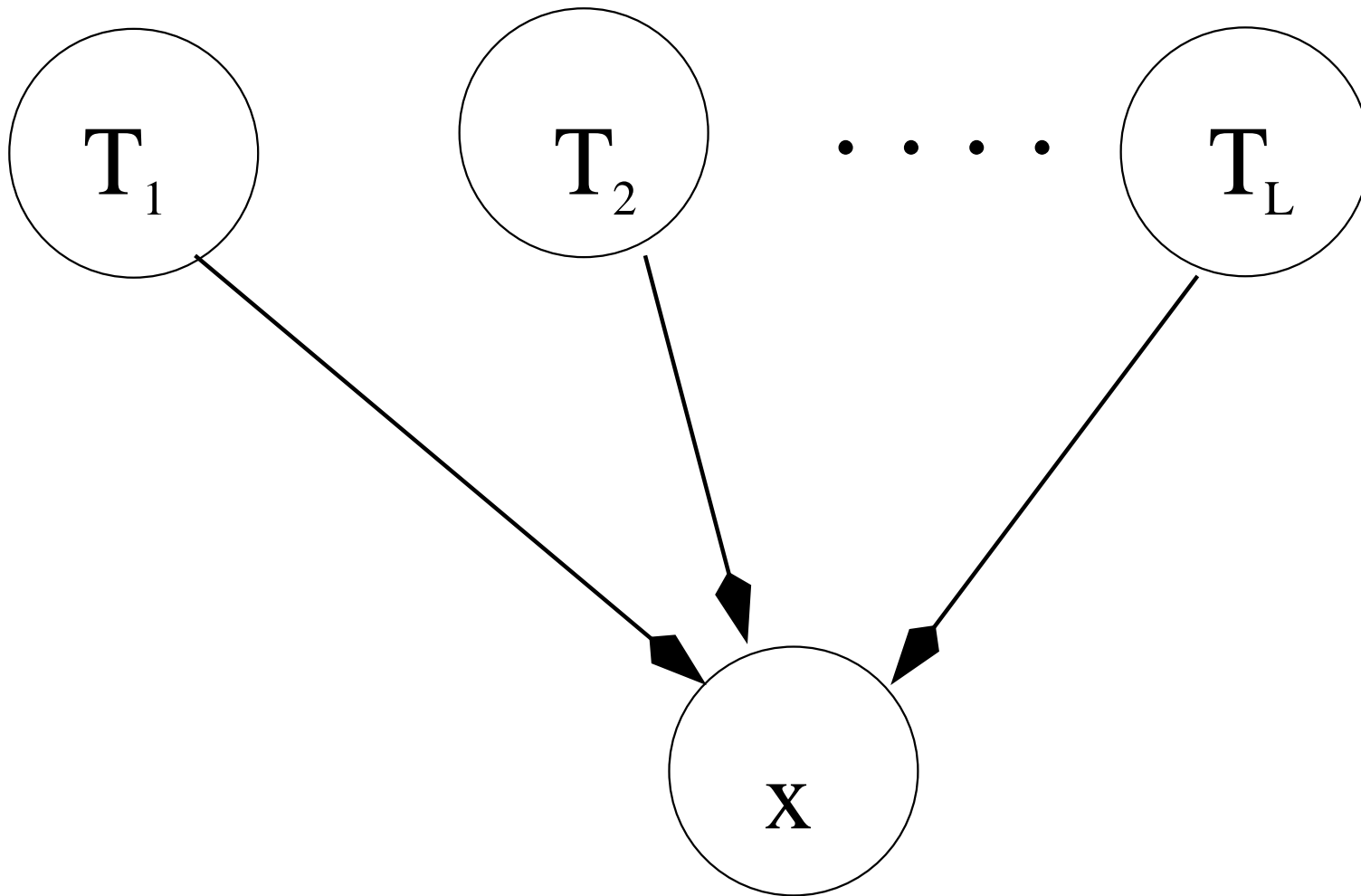
$$T_2(1 - \pi_2) . * (T_1 \pi_1 . * T_1 \mathbf{f}_1 + T_1(1 - \pi_1) . * \mathbf{b}) + \text{noise}$$

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

In general

$$\mathbf{x} = \sum_{\ell=0}^L \left(\left(\prod_{i=\ell+1}^L T_i(1 - \pi_i) \right) . * T_\ell \pi_\ell . * T_\ell \mathbf{f}_\ell \right) + \text{noise}$$

where $\mathbf{f}_0 = \mathbf{b}$, $\pi_0 = 1$ and T_0 is the identity



Flexible Sprites

- Allow some variability of the appearance and shape of the sprite relative to its template

$$p(\boldsymbol{\pi}, \mathbf{f}) = N(\boldsymbol{\pi}; \boldsymbol{\mu}_{\boldsymbol{\pi}}, \boldsymbol{\Sigma}_{\boldsymbol{\pi}})N(\mathbf{f}; \boldsymbol{\mu}_{\mathbf{f}}, \boldsymbol{\Sigma}_{\mathbf{f}})$$

Inference

- Can't afford to deal with multiple objects exactly due to the combinatorial explosion $O(J^L)$ (and sprite flexibility if this is present)
- Jojic & Frey (2001) use variational inference

$$p(\{T_\ell, \mathbf{f}_\ell, \boldsymbol{\pi}_\ell\} | \mathbf{x}) \simeq \prod_{\ell=1}^L q(T_\ell)q(\mathbf{f}_\ell)q(\boldsymbol{\pi}_\ell)$$

and minimize the KL-divergence $D(Q||P)$, using a discrete distribution for each $q(T_\ell)$ and Gaussian distributions for each $q(\mathbf{f}_\ell)$, $q(\boldsymbol{\pi}_\ell)$

- Use nonlinear optimization to minimize $D(Q||P)$ and FFTs for convolutions

Learning

$$F = D(Q||P) + \log p(\mathbf{x})$$

- F is a lower bound on the log likelihood
- Iterate until convergence
 1. Generalized E-step: Increase F wrt one set of variational parameters for each video frame
 2. Generalized M-step: Increase F wrt the model parameters

Results

Many examples of the use of learning sprites, including

- 2 people moving against a stationary background
- A person moving against a moving background (camera motion)



Greedy Learning of Multiple Objects

- Our data is images containing multiple objects and the 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
- The method can be speeded up for image *sequences* using (approximate) tracking

Learning One Object

The mask π specifies the probability that a pixel is from the foreground or background.

- Foreground/background only

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

- Coping with transformations

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

Fitting the model to data

- \mathbf{f} , \mathbf{b} , $\boldsymbol{\pi}$, σ_f^2 , σ_b^2 can be learned by EM
- In comparison to Jojic and Frey (2001) note that $\boldsymbol{\pi}$ defines a mixture model, not a matte. This means that an exact M-step can be used

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

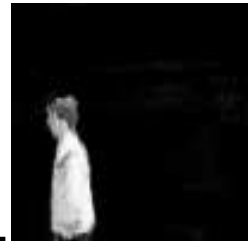
Mask



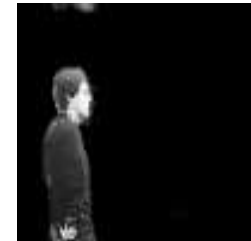
Mask



Mask * Foreground



Mask * Foreground



Background



- Consider two people comoving—what happens?

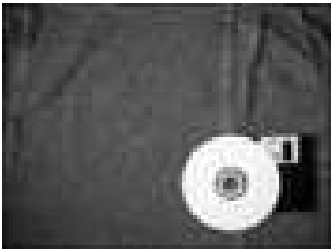
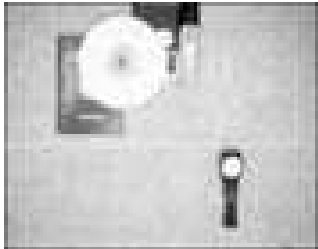
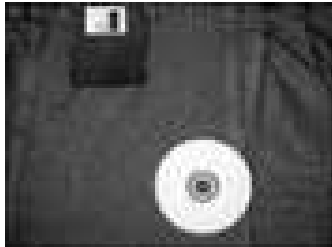
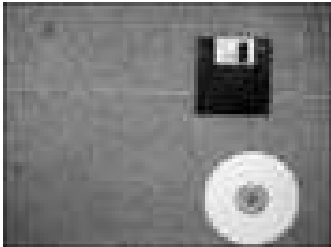
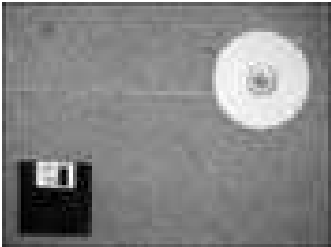
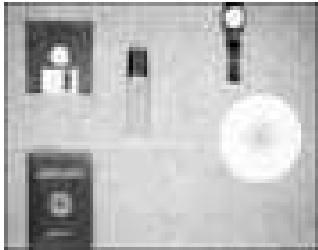
Frey and Jojic Video Sequences

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

2 Objects and Moving Background

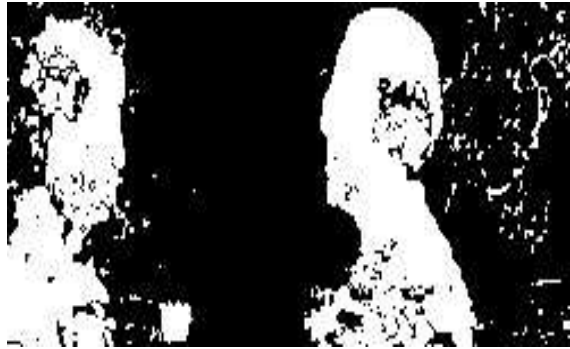


Further Examples

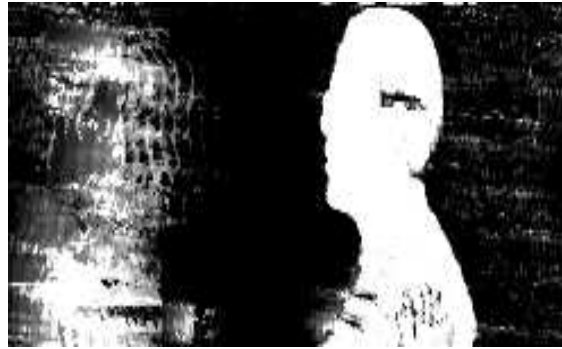


Speeding it up with tracking

- 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



$t = 1$



$t = 10$



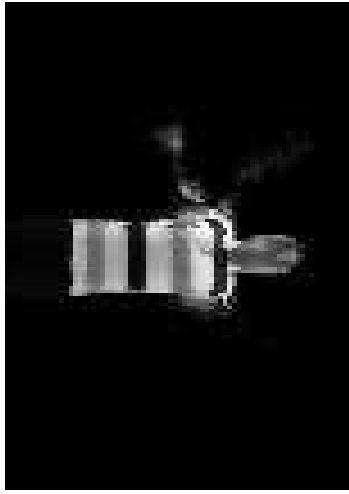
$t = 20$



Learning Parts



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



Discussion

- Layered representations can be learned in a variety of ways
- Probabilistic framework is attractive, but a very large number of transformations may have to be considered (cf affine motion of Wang and Adelson)
- Tracking method can be used to speed up learning in sequences (richer transformations)
- Is a layered representation sufficient? (3-d vs multiple views)

References

- J. Y. A. Wang and E. H. Adelson, *Representing Moving Images with Layers*, IEEE Trans Image Processing 3(5) 625-638, 1994.
- N. Jojic and B. Frey, *Learning Flexible Sprites in Video Layers*, CVPR 2001.
- 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.