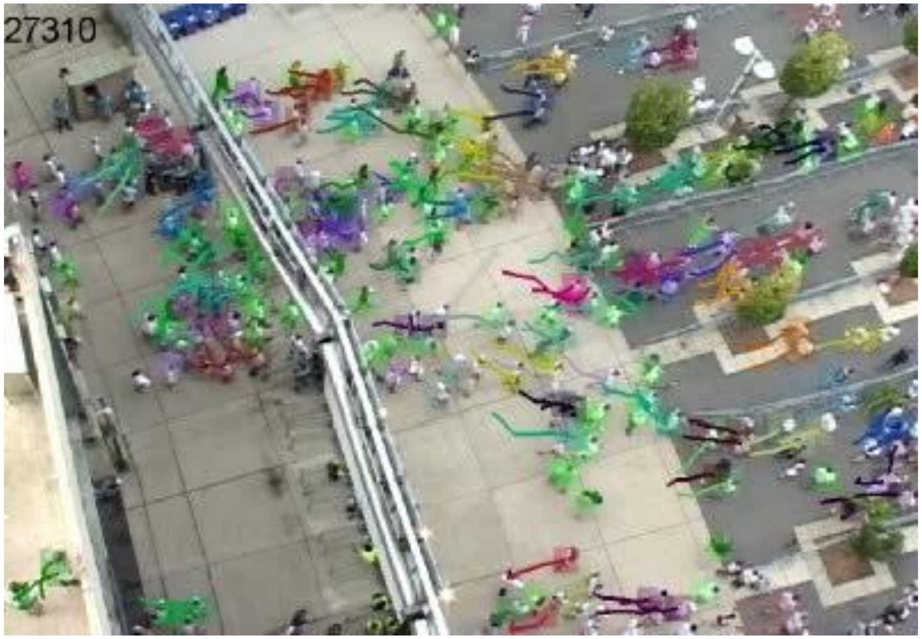
- Using computer vision tools to look at people in public places

- Real-time monitoring
  - situation awareness
  - notifications/alarms
- After-action review
  - traffic analysis



Things we might want to know:

- How many people are there?
- How to track specific individuals?
- How to determine who is with whom?

Challenges:

Crowd scenes tend to have low resolution.
You rarely see individuals in isolation.
Indeed, there are frequent partial occlusions.

Crowd Counting

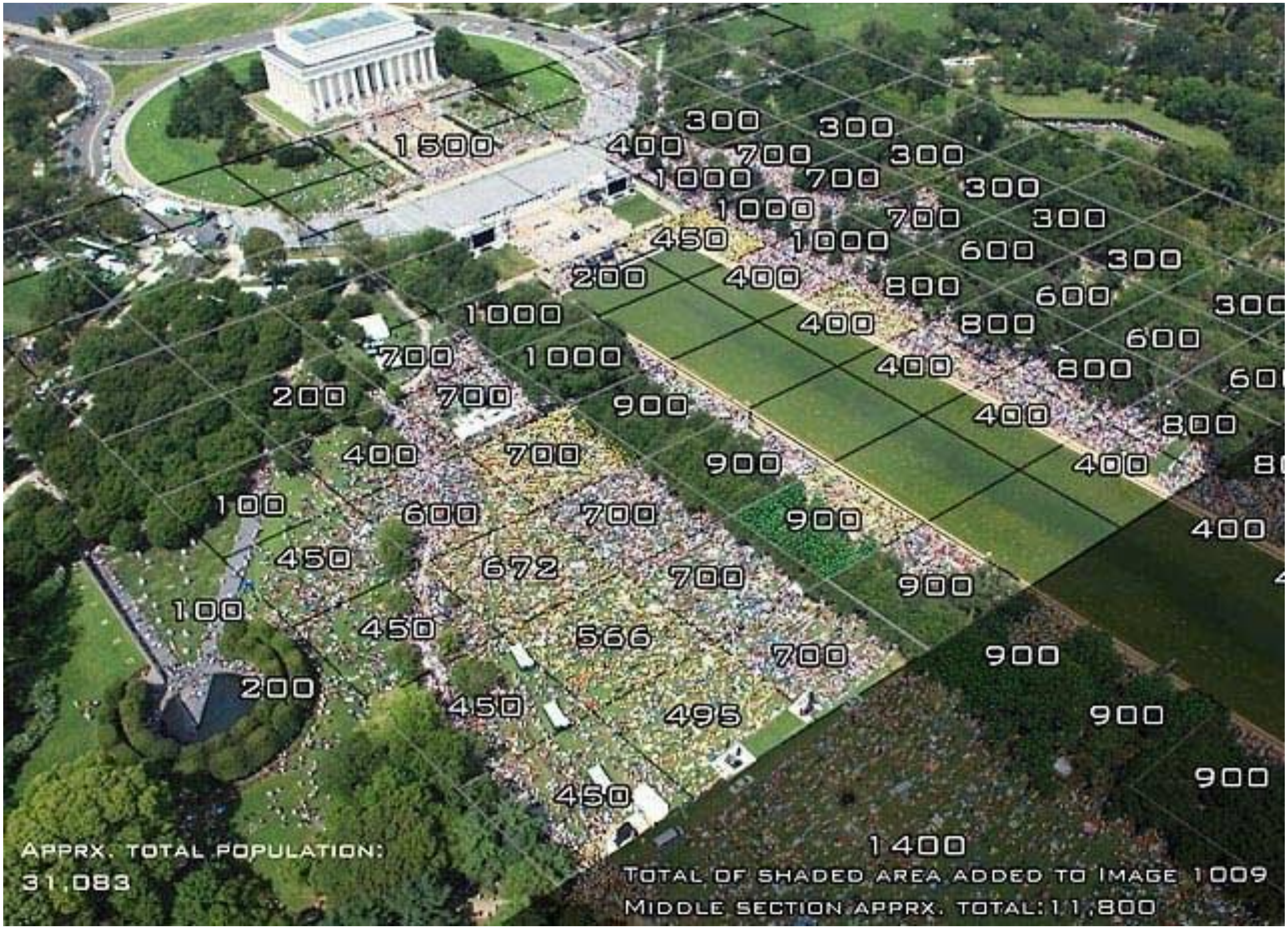
FAQ: How many people participated in ...

- Tahrir Square Protests
- Obama’s inaguration
- Occupy Wall Street
- Kumbh Mela

Jacob’s Method

- Herbert Jacobs, Berkeley, 1960s
- count = area * density
  - 10 sqft/person – loose crowd (arm’s length from each other)
  - 4.5 sqft/person – more dense
  - 2.5 sqft/person – very dense (shoulder-to-shoulder)
- Problem: Pedestrians do not uniformly distribute over a space, but clump together into groups or clusters.
- Refinement: break area into a grid of ground patches and estimate a different density in each small patch. Accumulate these counts over whole area.

Example of Jacob’s Method



Computer Vision Could do Better!

Cavaet: nobody really wants accurate counts

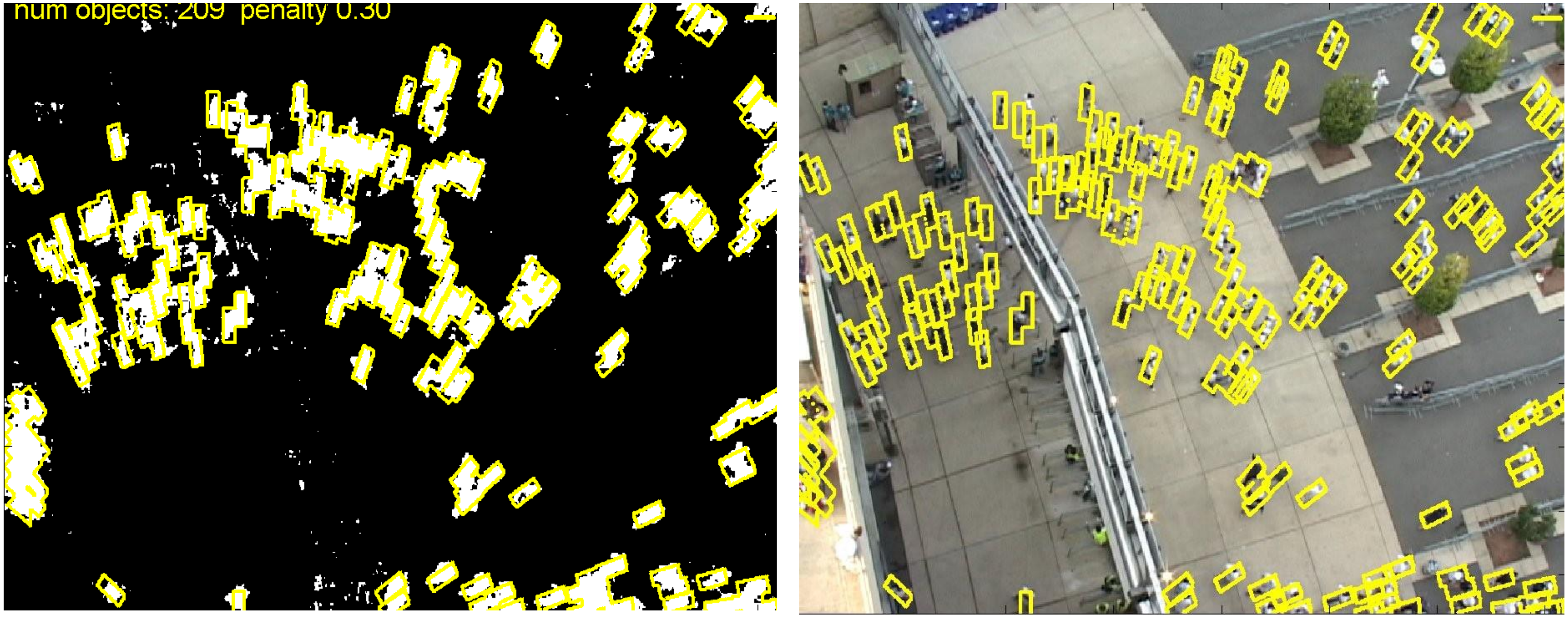
e.g. organizers of the “Million Man March” in Washington DC threatened to sue the National Park Service for estimating that only 400K people attended.

Vision-based Counting

- detection and tracking (light density)
- clustering feature trajectories that move coherently (moderate density)
- treat crowd as a dynamic texture and compute regression estimates based on measured properties (heavy density)

Detecting and Counting Individuals

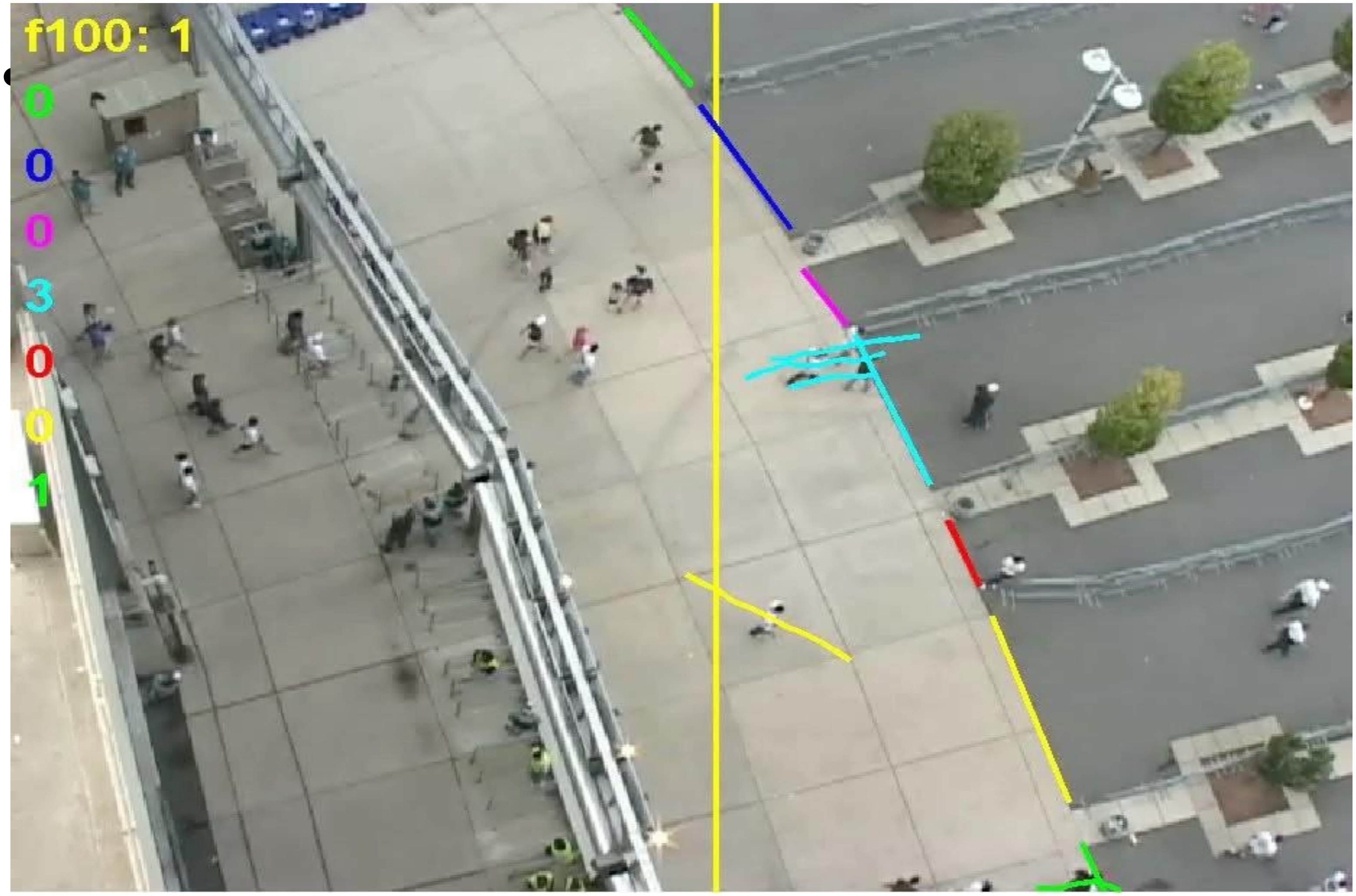
Ge and Collins, "Marked Point Processes for Crowd Counting," *IEEE Computer Vision and Pattern Recognition (CVPR'09)*, Miami, FL, June 2009, pp.2913-2920.



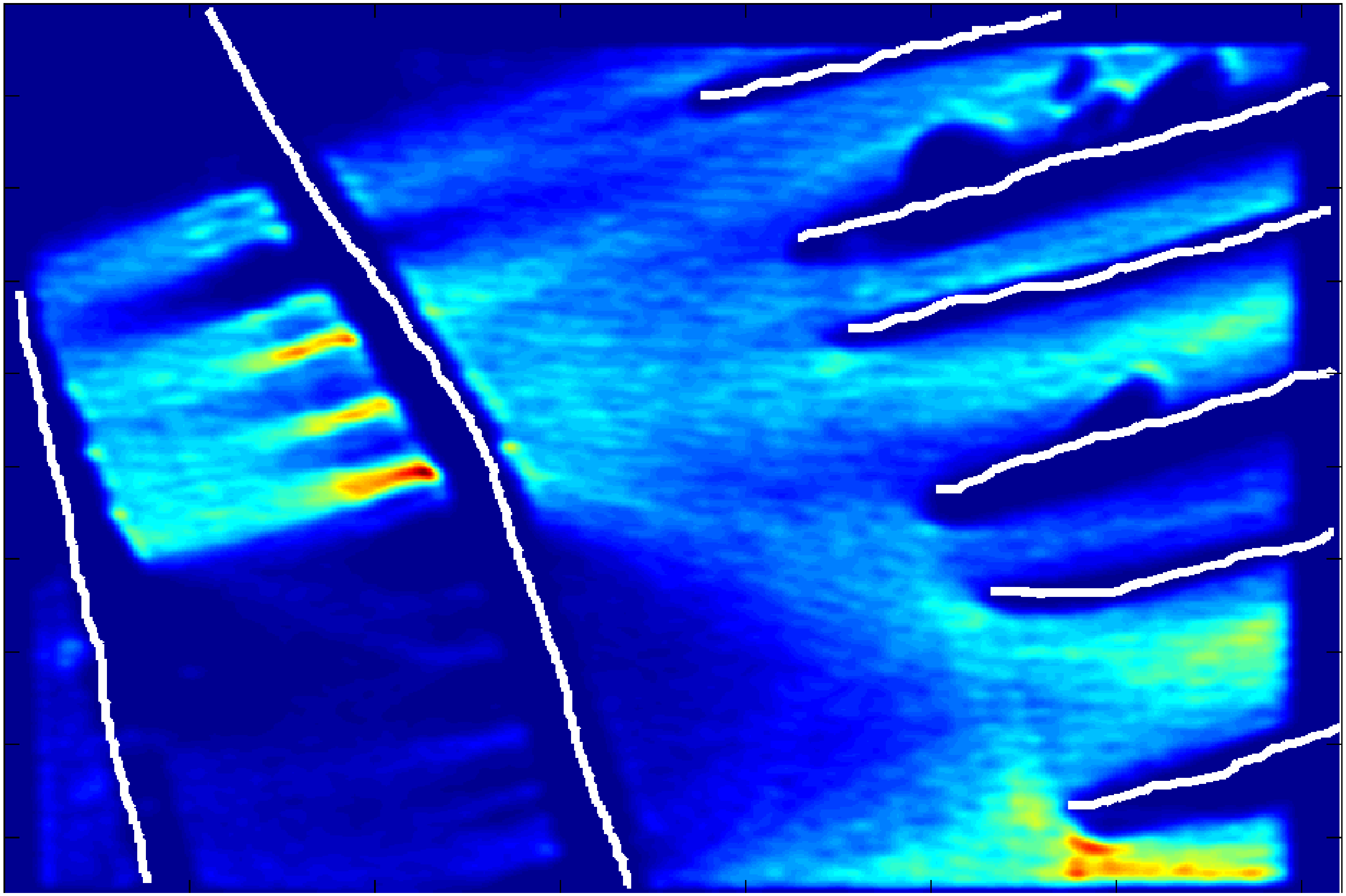
Good for low-resolution / wide-angle views.
Relies on foreground/background segmentation.
Not appropriate for very high crowd density or stationary people.

GateA Path Counts

movie

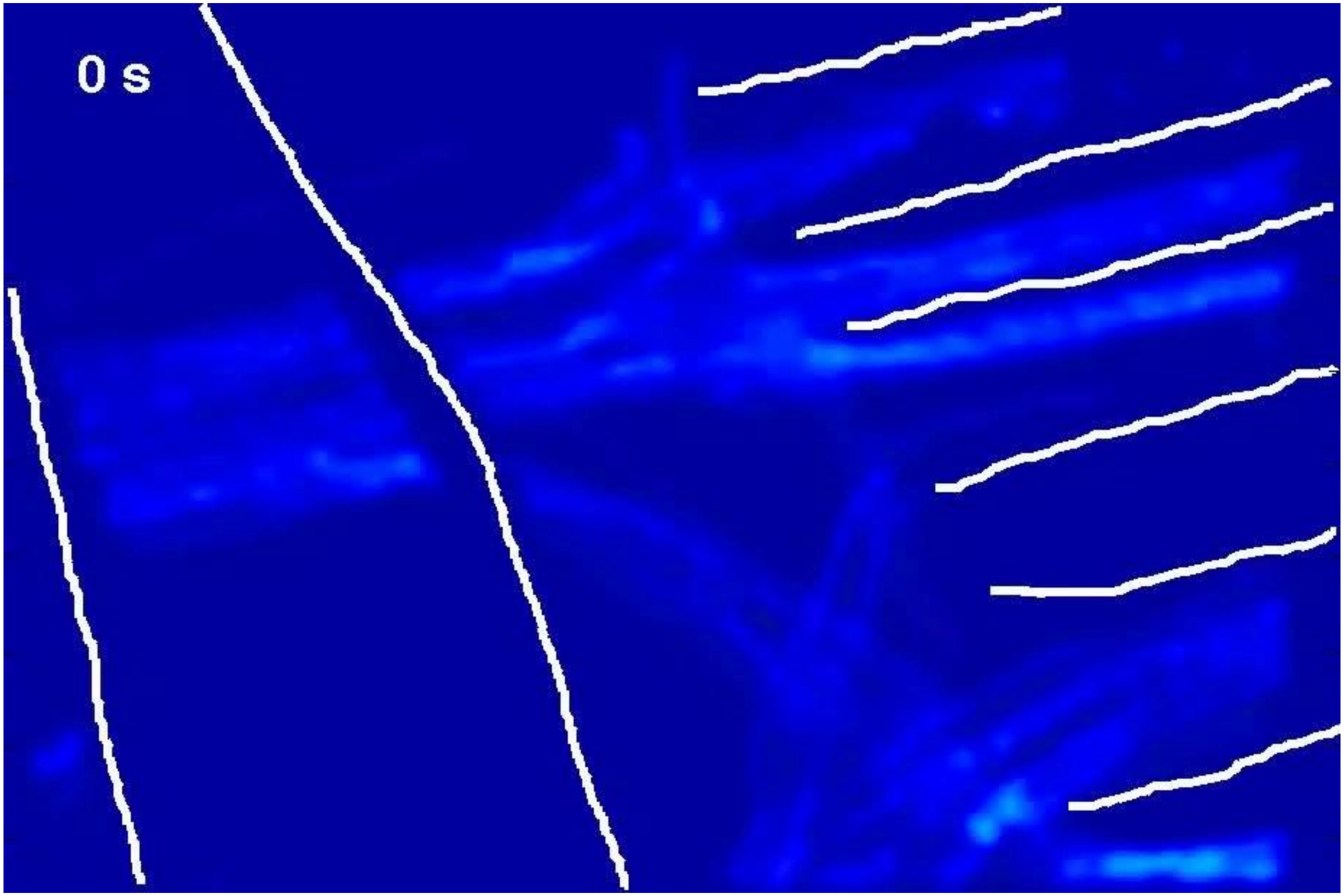


Maintain a running count of number of people whose trajectories cross a set of user-specified lines (color-coded).



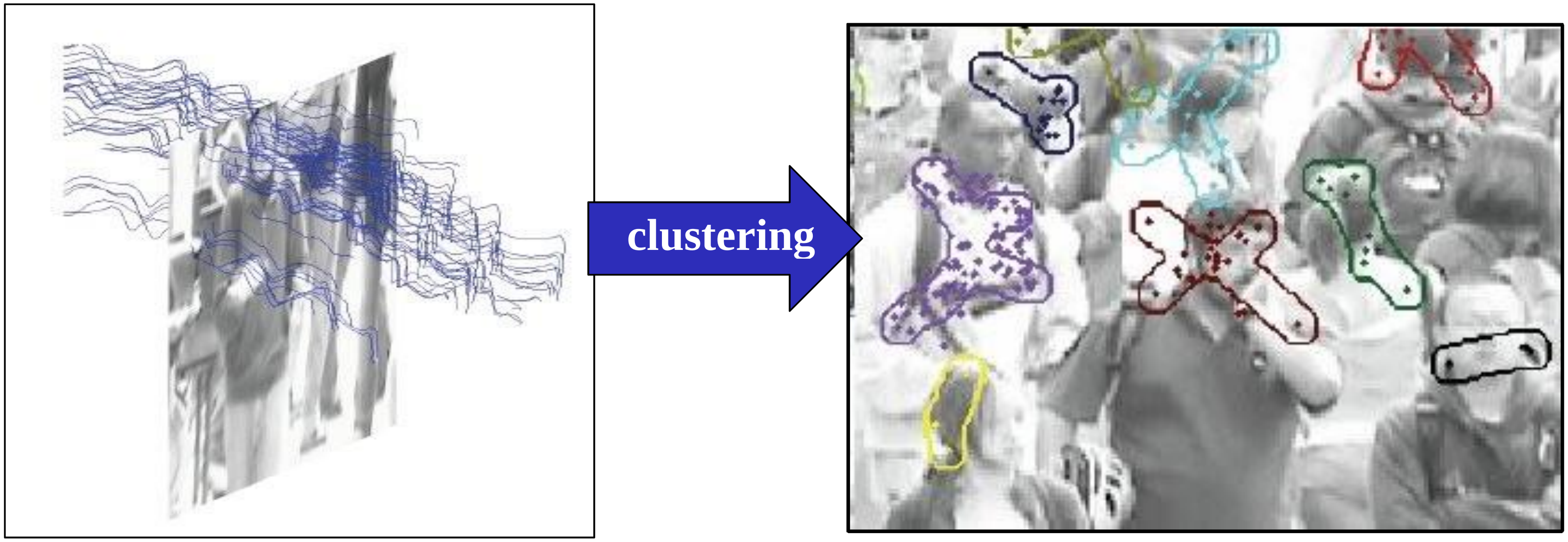
30 minute period

movie



Time Lapse. Integrated over spatial/temporal windows.

Idea: track many small features (e.g. corners) over time and cluster sets of features that have similar motion.

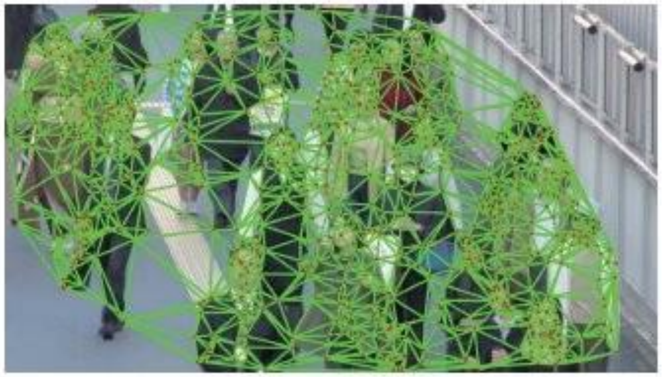


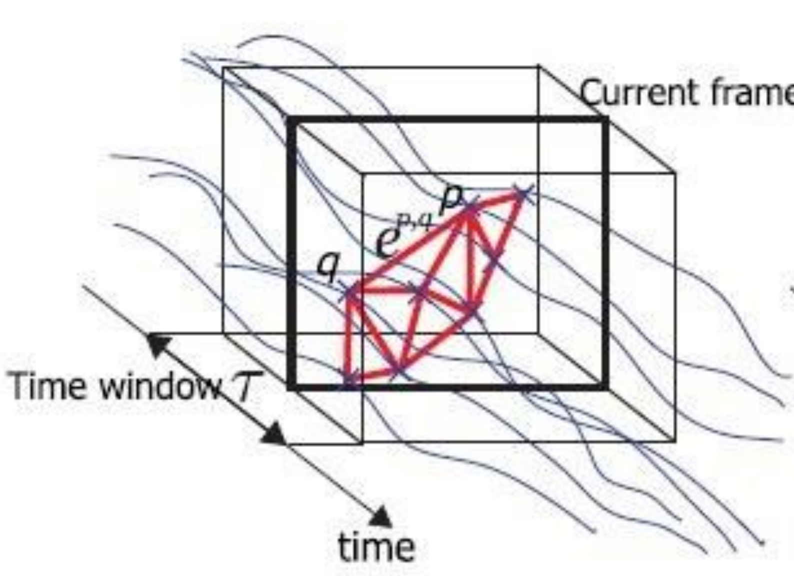
corner trajectories

independently moving objects

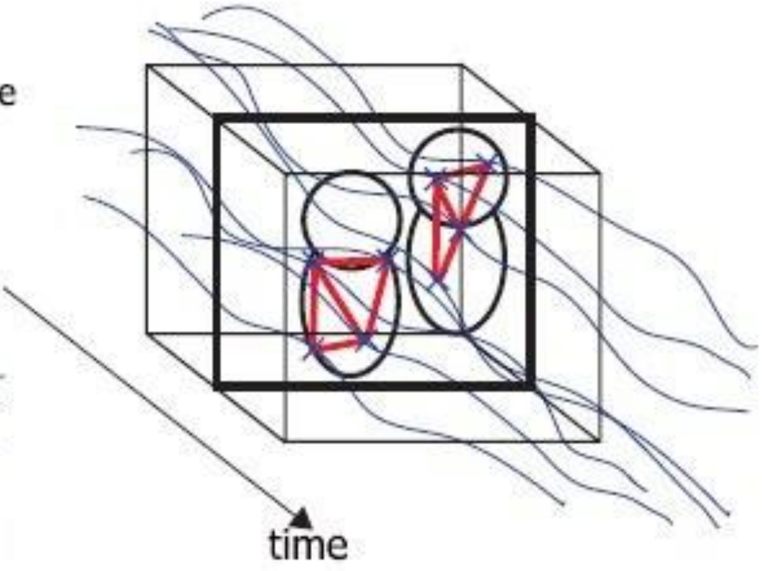
- G. J. Brostow and R. Cipolla, “Unsupervised bayesian detection of independent motion in crowds,” in IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 594–601.
- V. Rabaud and S. Belongie, “Counting crowded moving objects,” in IEEE Computer Vision and Pattern Recognition, New York City, 2006, pp. 705–711.
- D. Sugimura, K. Kitani, T. Okabe, Y. Sato, and A. Sugimoto, “Using individuality to track individuals: Clustering individual trajectories in crowds using local appearance and frequency trait,” in International Conference on Computer Vision, 2009, pp. 1467–1474.

Basic steps: Form a corner connectivity graph. Assign each edge a dissimilarity score based on distance and motion coherence of trajectories. Prune edges with high scores. The remaining connected components are the independent objects.



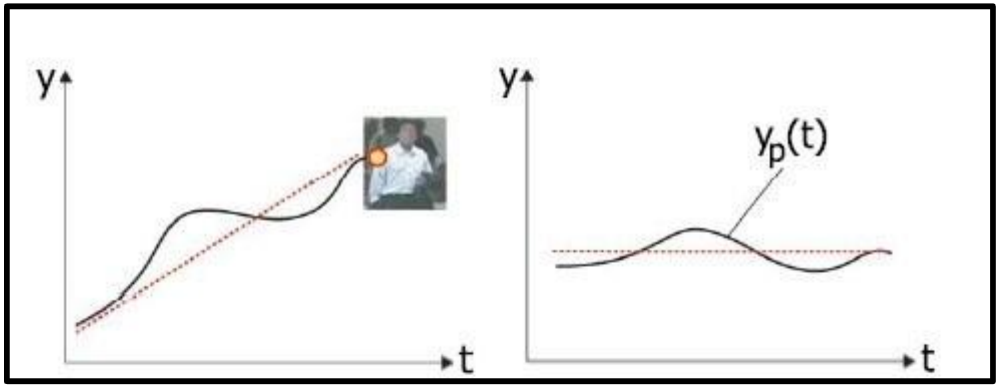


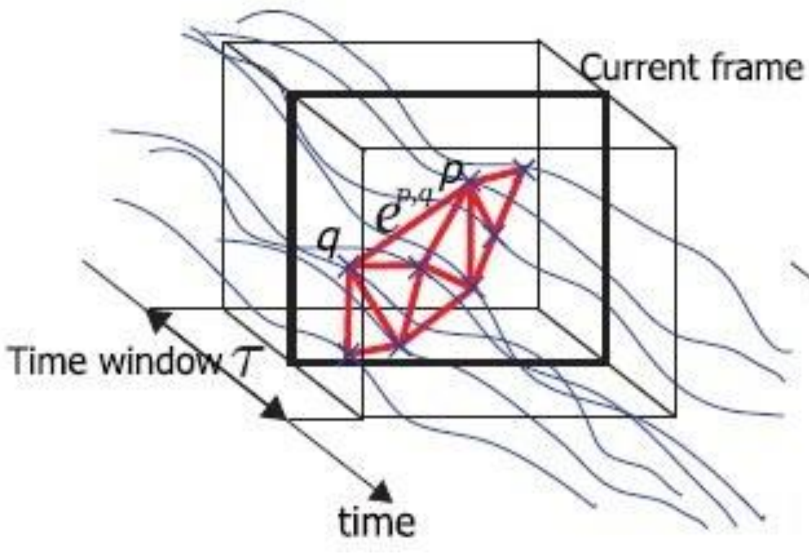
connectivity graph



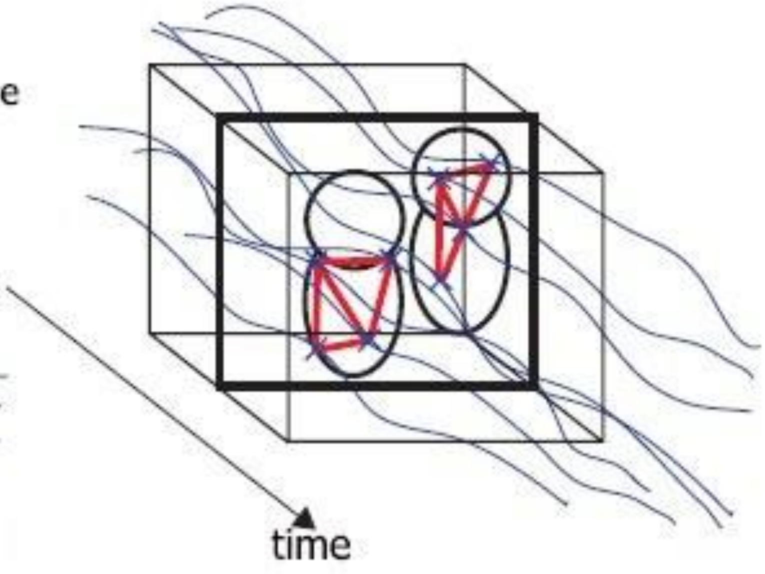
connected components after pruning

Note: Sugimara et.al. add a feature based on gait periodicity to help disambiguate nearby people.



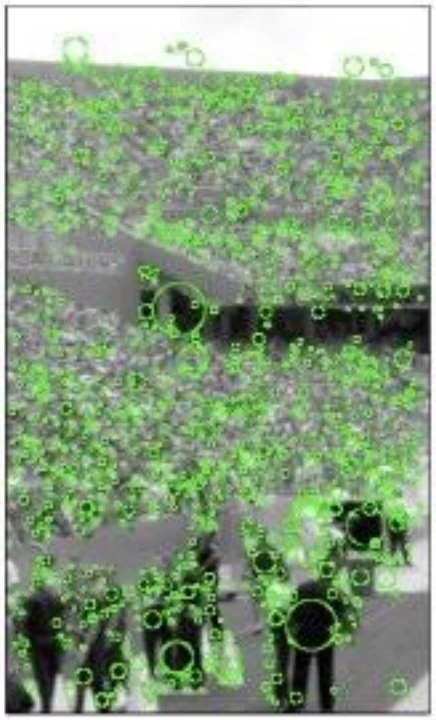


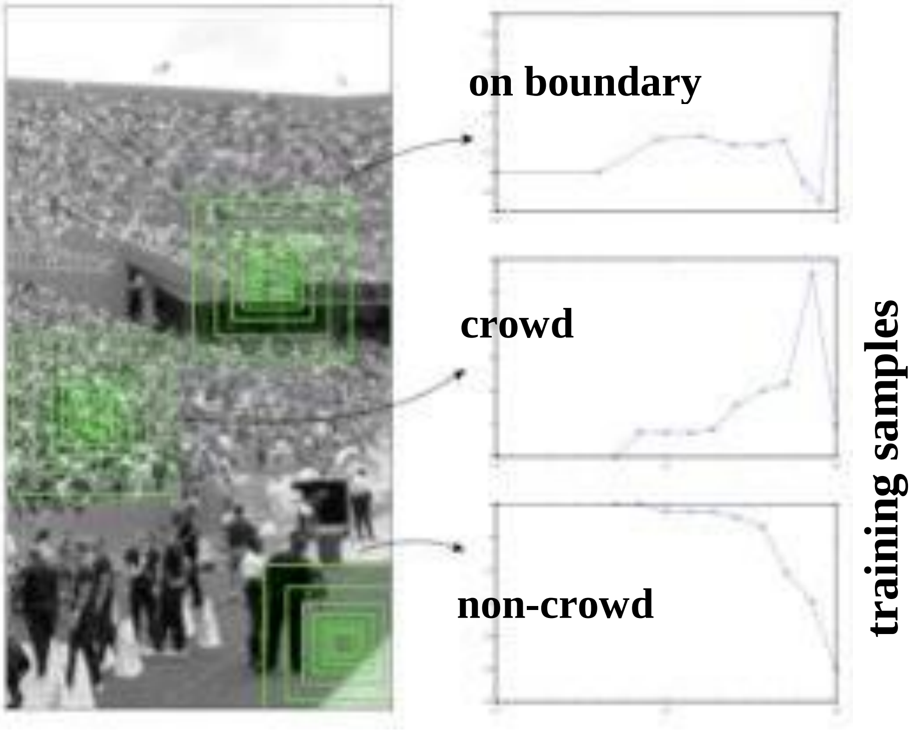
connectivity graph



connected components after pruning

Arandjelovic, “Crowd Detection from Still Images,” BMVC 2008

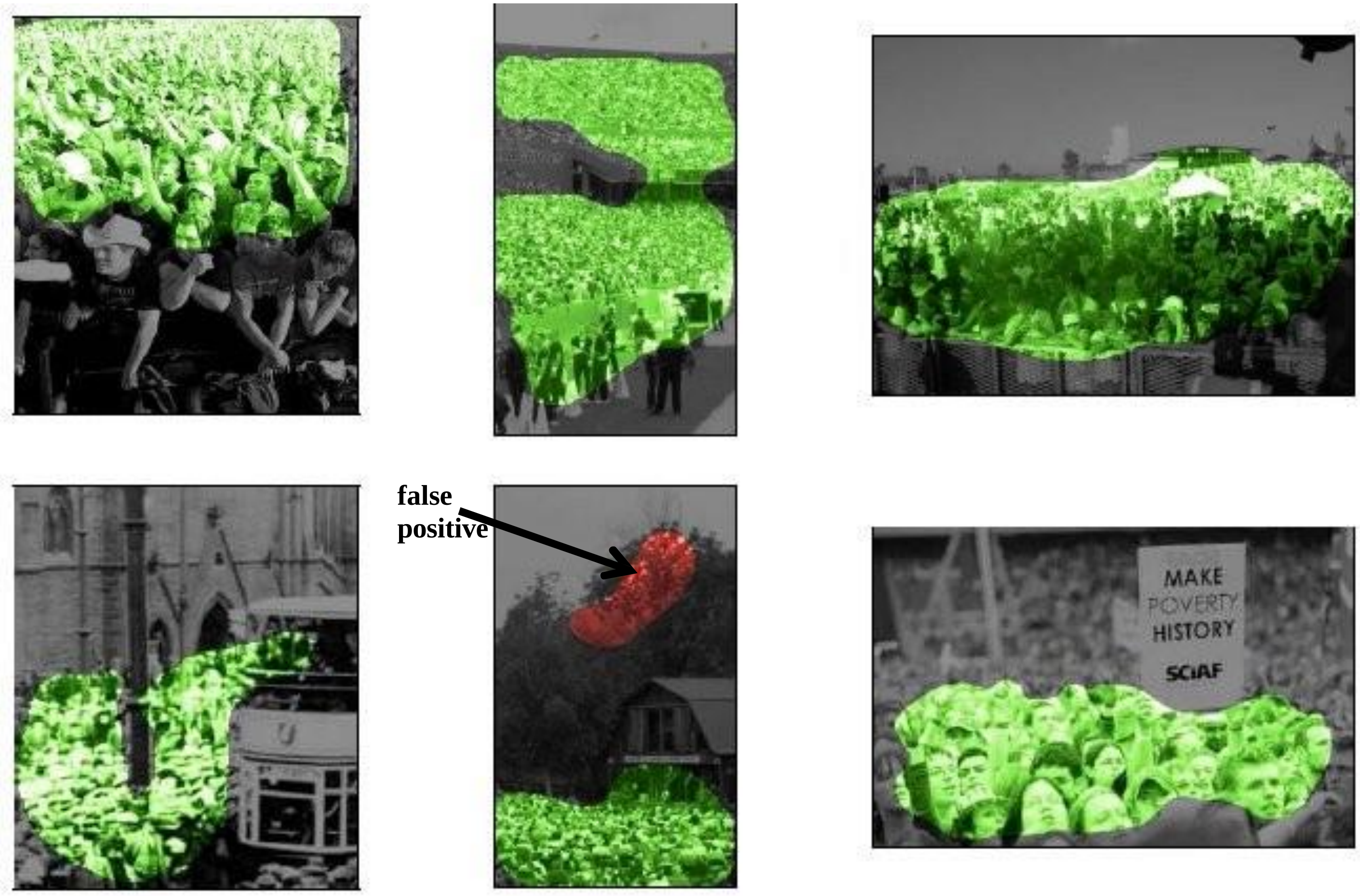


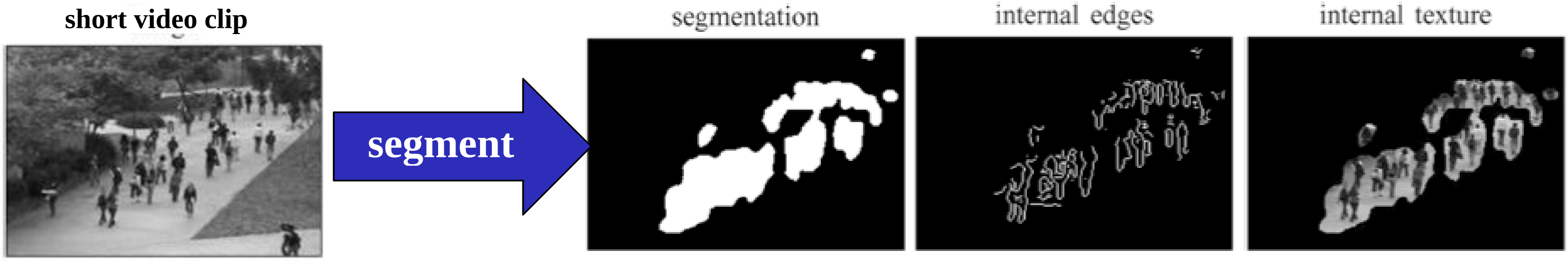


- SIFT descriptors
- K-means clustering to form “SIFT-Words”

- Likelihood ratio of distributions of word counts over 10 patch sizes yields 10-D feature vector
- Radial basis SVM for classification into crowd / non-crowd

Sparse classifications turned into dense segmentation using graph cuts. Unary costs based on SVM output and pairwise costs based on magnitude of patch likelihood scores (small magnitudes indicate interclass boundaries).

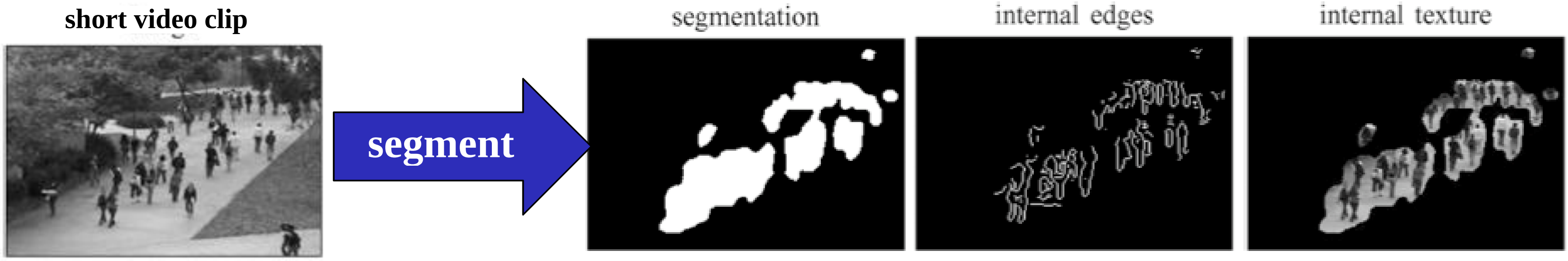




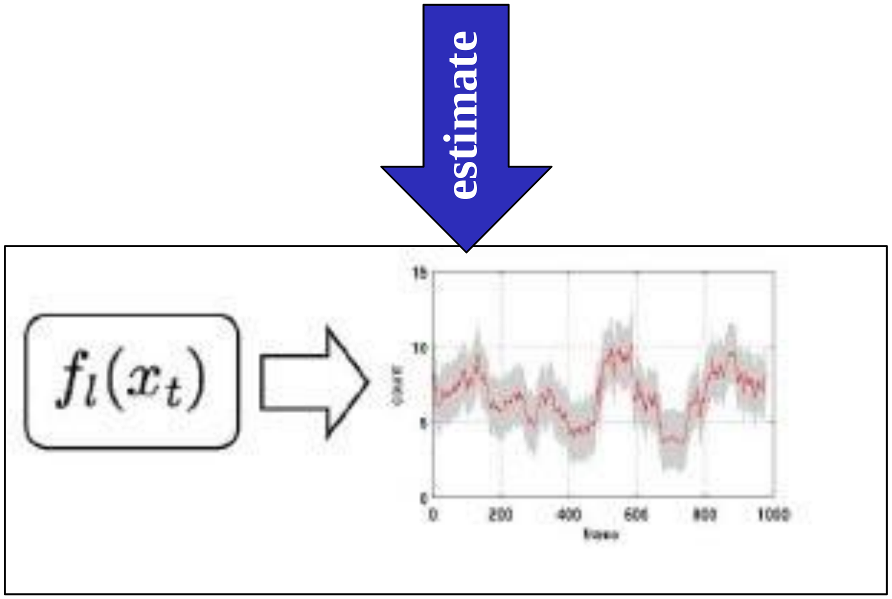
motion segmentation using dynamic textures

Extract feature vector for each frame:

- **region features**
  e.g. area, perimeter, num connected components...
- **internal edge features**
  e.g. num edges, histogram of orientations
- **grey-level texture features**
  e.g. homogeneity, energy, entropy

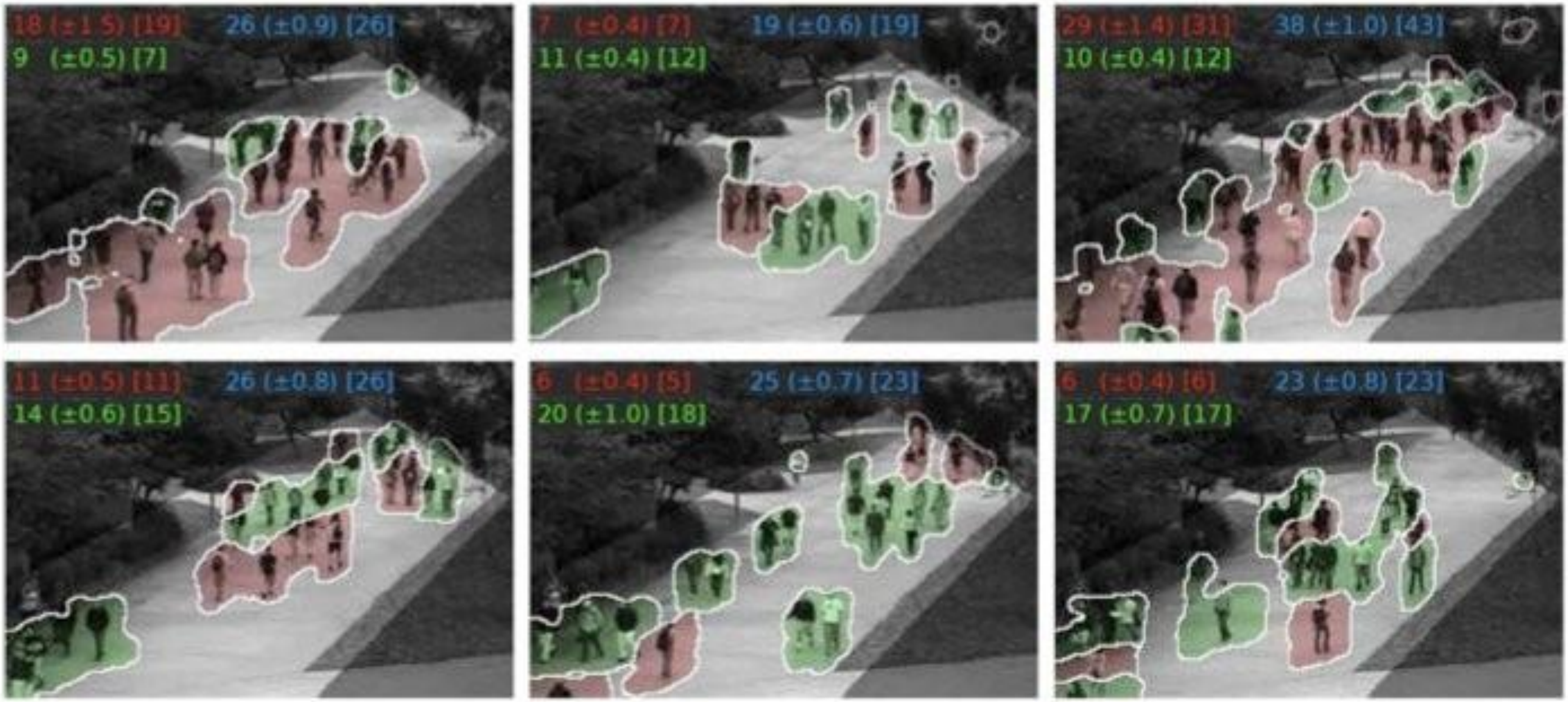


Extract feature vector for each frame:



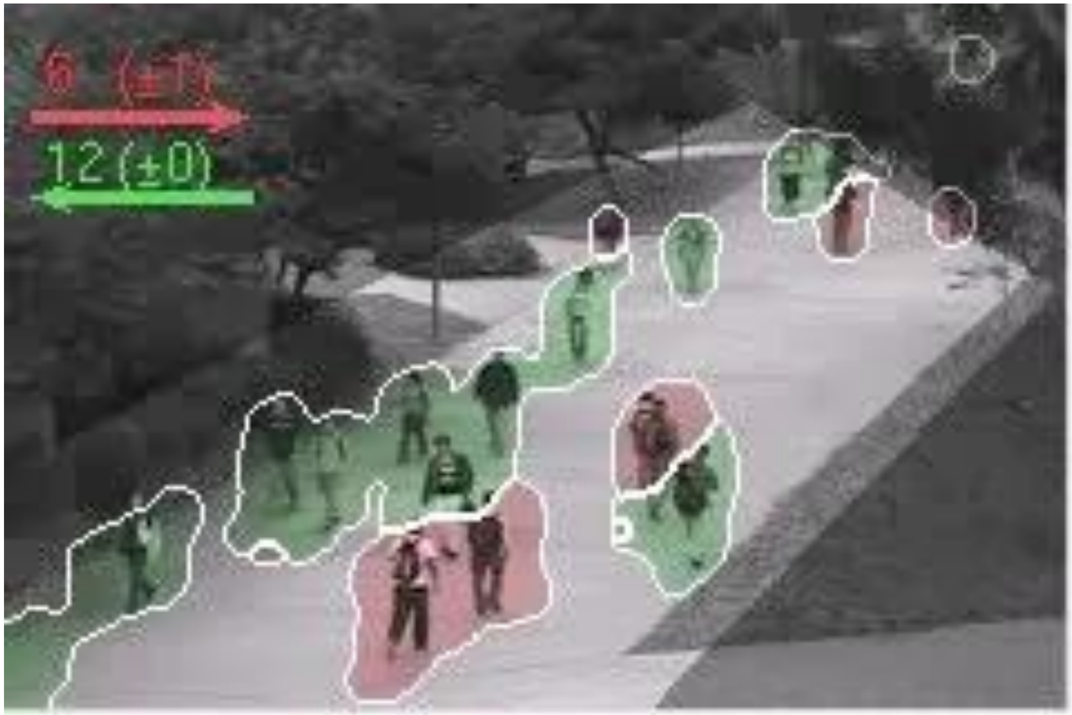
estimate counts using learned regression function

UTS, Sean. He



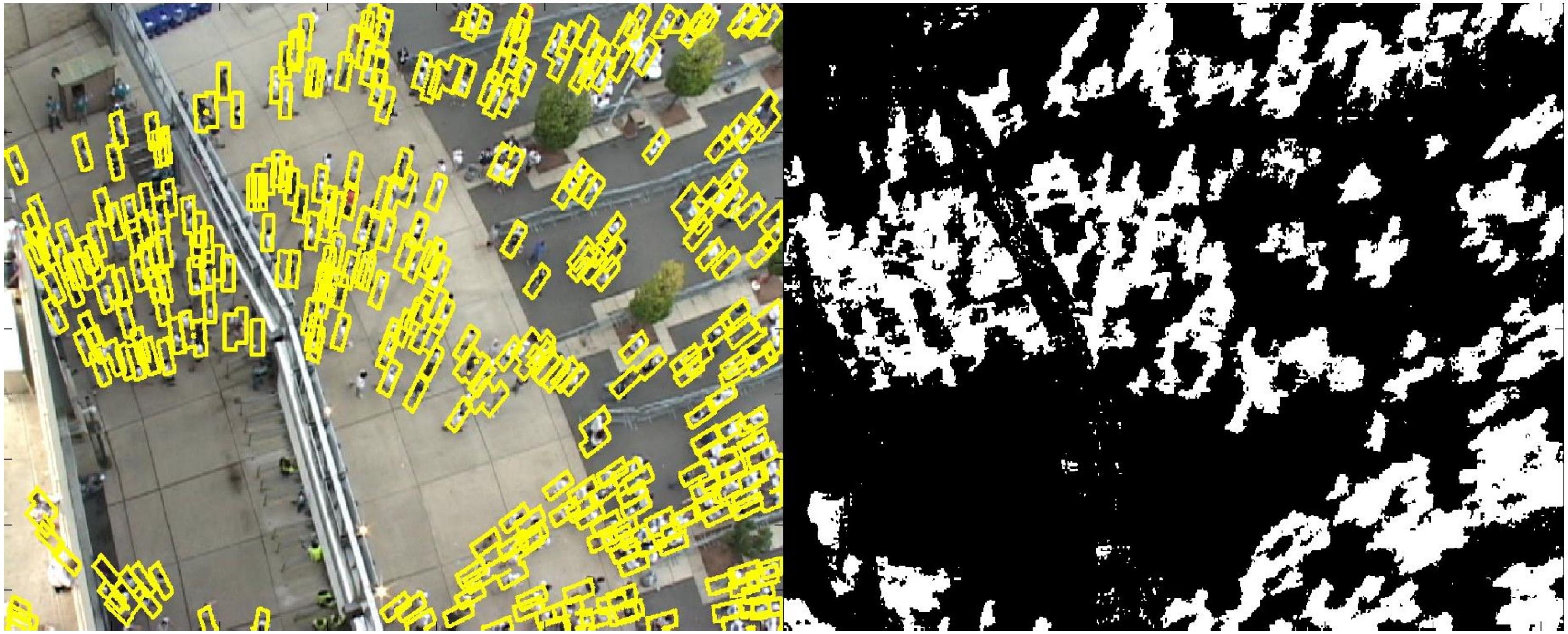
green/red = crowd walking towards/away   blue = total
numeric results formatted as: estimated count (uncertainty) [true count]

UTS, Sean. He



green/red = crowd walking towards/away
numeric results formatted as: estimated count (uncertainty)

Our Approach*



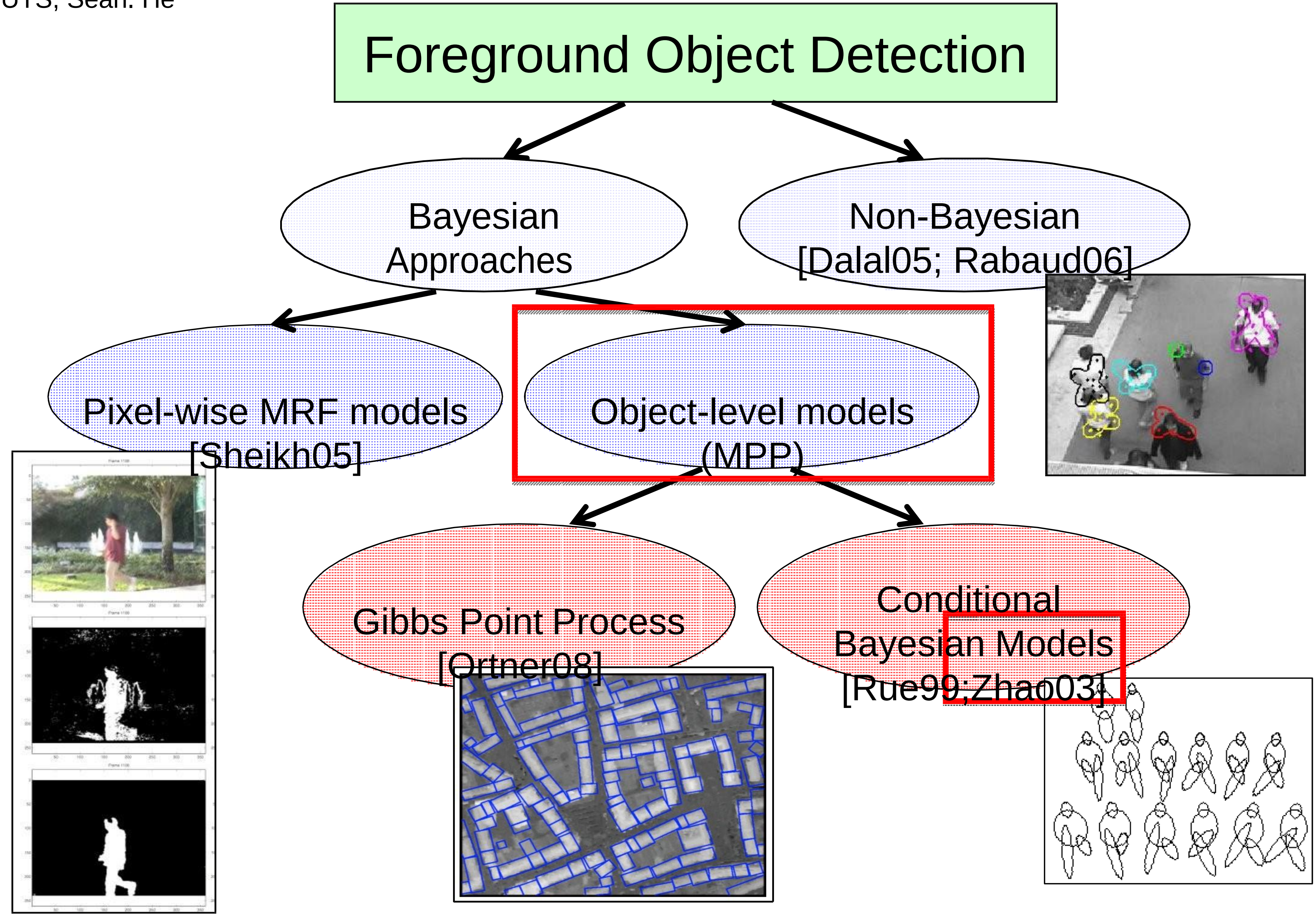
Bayesian Marked Point Process

- the prior models expected size/shape of people, indexed by location
- the likelihood measures how well a proposed configuration explains the data
- MAP estimate of number/configuration of people is found using RJ-MCMC

*Weina Ge and R.Collins, “Marked Point Processes for Crowd Counting,”

UTS, Sean. He

Foreground Object Detection



Bayesian Approaches

Non-Bayesian [Dalal05; Rabaud06]

Pixel-wise MRF models [Sheikh05]

Object-level models (MPP)

Gibbs Point Process [Ortner08]

Conditional Bayesian Models [Rue99;Zhao03]

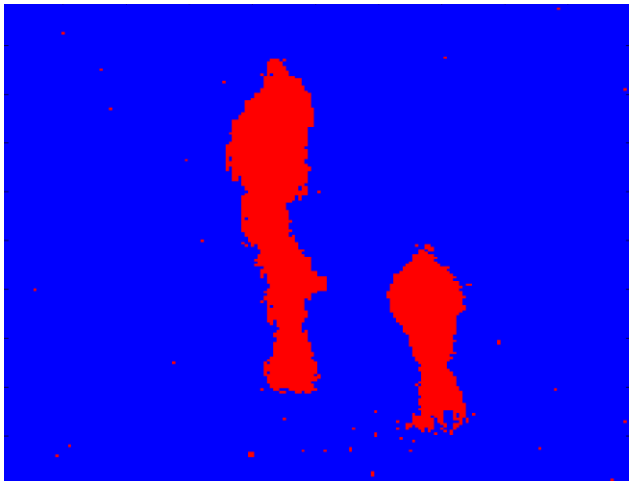
Marked Point Process

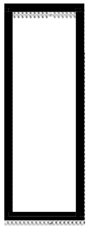
- Spatial Point Process
  - Distribution of a set of points in a bounded space
- Marked Point Process (MPP)
  - A spatial point process + a “mark” process
- Examples
  - Spatial process could model tree locations in a forest and the mark could be tree height
  - Spatial process could model cell locations on a microscope slide and the mark could be a polygonal representation of the cell boundary [Rue and Hurn, 1999]

Simplified Problem Statement

Given a foreground image, find a configuration of bounding boxes* that cover a majority of foreground pixels while leaving a majority of background pixels uncovered.

foreground image





person-sized bounding box

As an MPP:

- Spatial process models number and (x,y) locations of bounding boxes
- Mark process models (height,width,orientation) of each bounding box.

*Footnote: We will add more realistic “shape” models in a moment

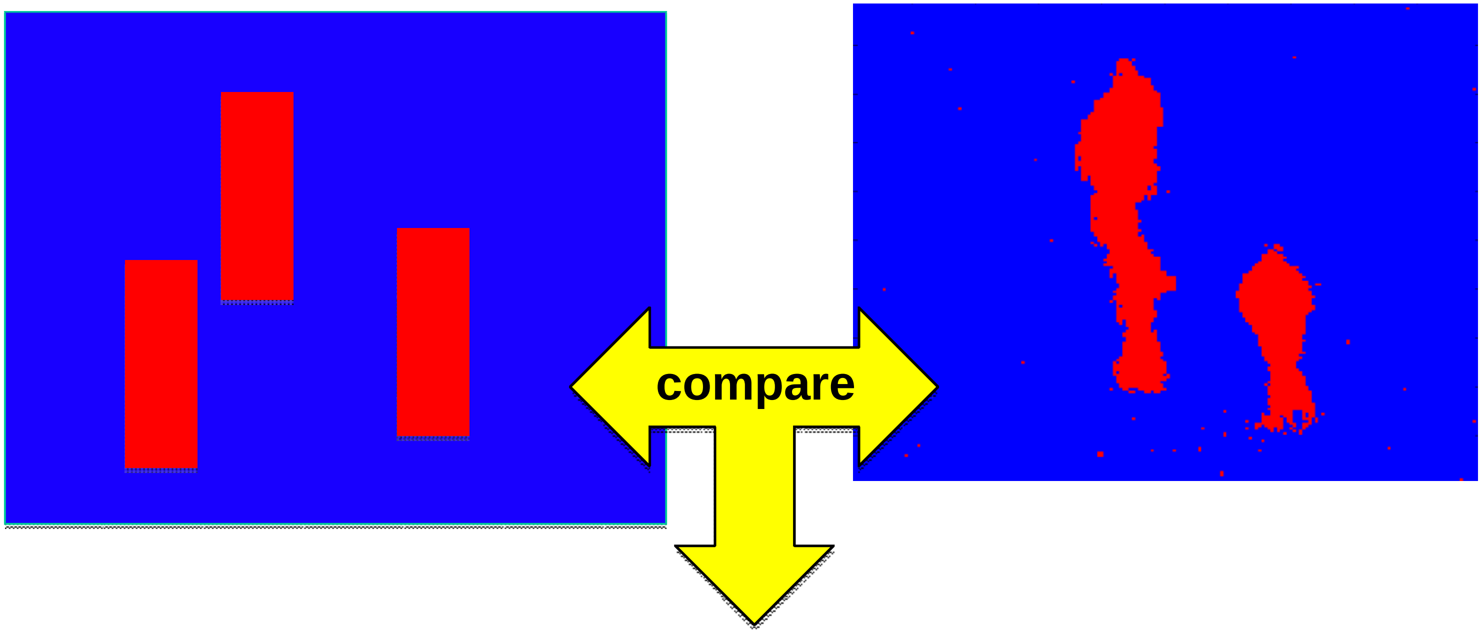
Likelihood Score

To measure how “good” a proposed configuration is, we generate a foreground image from it and compare with the observed foreground image to get a likelihood score.

config = $\{\{x_1,y_1,w_1,h_1,theta_1\},\{x_2,y_2,w_2,h_2,theta_2\},\{x_3,y_3,w_3,h_3,theta_3\}\}$

generated foreground image

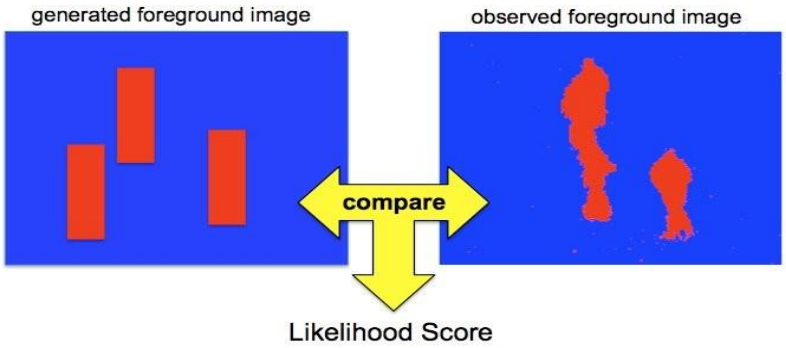
observed foreground image



compare

Likelihood Score

Likelihood Score



Bernoulli distribution model

$$\begin{aligned}
p_{00} &= p(y_i=0|x_i=0) &= \text{prob of observing background given a label of background}\\
p_{01} &= p(y_i=0|x_i=1) &= \text{prob of observing background given a label of foreground}\\
p_{10} &= p(y_i=1|x_i=0) &= \text{prob of observing foreground given a label of background}\\
p_{11} &= p(y_i=1|x_i=1) &= \text{prob of observing foreground given a label of foreground}
\end{aligned}$$

$$\begin{aligned}
c_{00} &= \text{count of pixels where observation is background and label is background}\\
c_{01} &= \text{count of pixels where observation is background and label is foreground}\\
c_{10} &= \text{count of pixels where observation is foreground and label is background}\\
c_{11} &= \text{count of pixels where observation is foreground and label is foreground}
\end{aligned}$$

likelihood

$$L(Y|X) = \prod_{i=1}^{N} p(y_i|x_i) = p_{00}{}^{c_{00}} p_{01}{}^{c_{01}} p_{10}{}^{c_{10}} p_{11}{}^{c_{11}}$$

simplify, by assuming

$$p_{00} = p_{11} = \mu \qquad \text{and} \qquad p_{01} = p_{10} = 1-\mu$$

log likelihood

$$\begin{aligned}
logL(Y|X) &= (c_{00}+c_{11})\log\mu + (c_{01}+c_{10})\log(1-\mu)\\
&= [N-(c_{01}+c_{10})]\log\mu + (c_{01}+c_{10})\log(1-\mu)\\
&= N\log\mu - \boxed{(c_{01}+c_{10})}[\log\mu - \log(1-\mu)]
\end{aligned}$$

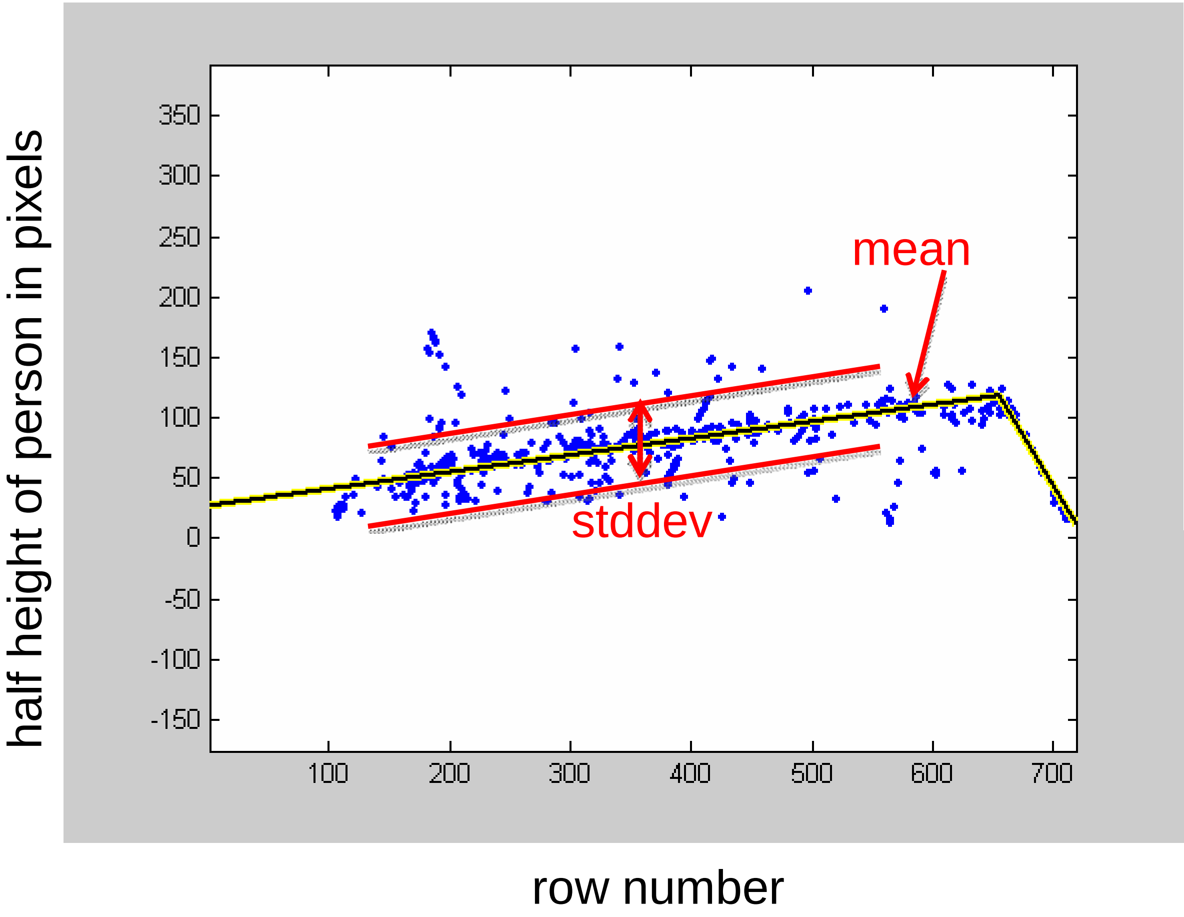
Number of pixels that disagree!

Prior Model

We use a prior to model our expectations about bounding box configurations in the image

$$\underbrace{\pi(\mathbf{o}_i)}_{\substack{\text{Prior for} \\ \text{bounding} \\ \text{box i}}} = \underbrace{\pi(p_i)}_{\substack{\text{Prior on location} \\ \text{(center point)}}} \underbrace{\pi(w_i, h_i, \theta_i | p_i)}_{\substack{\text{Prior on bounding box} \\ \text{height, width and orientation,} \\ \text{conditioned on center location.}}}$$

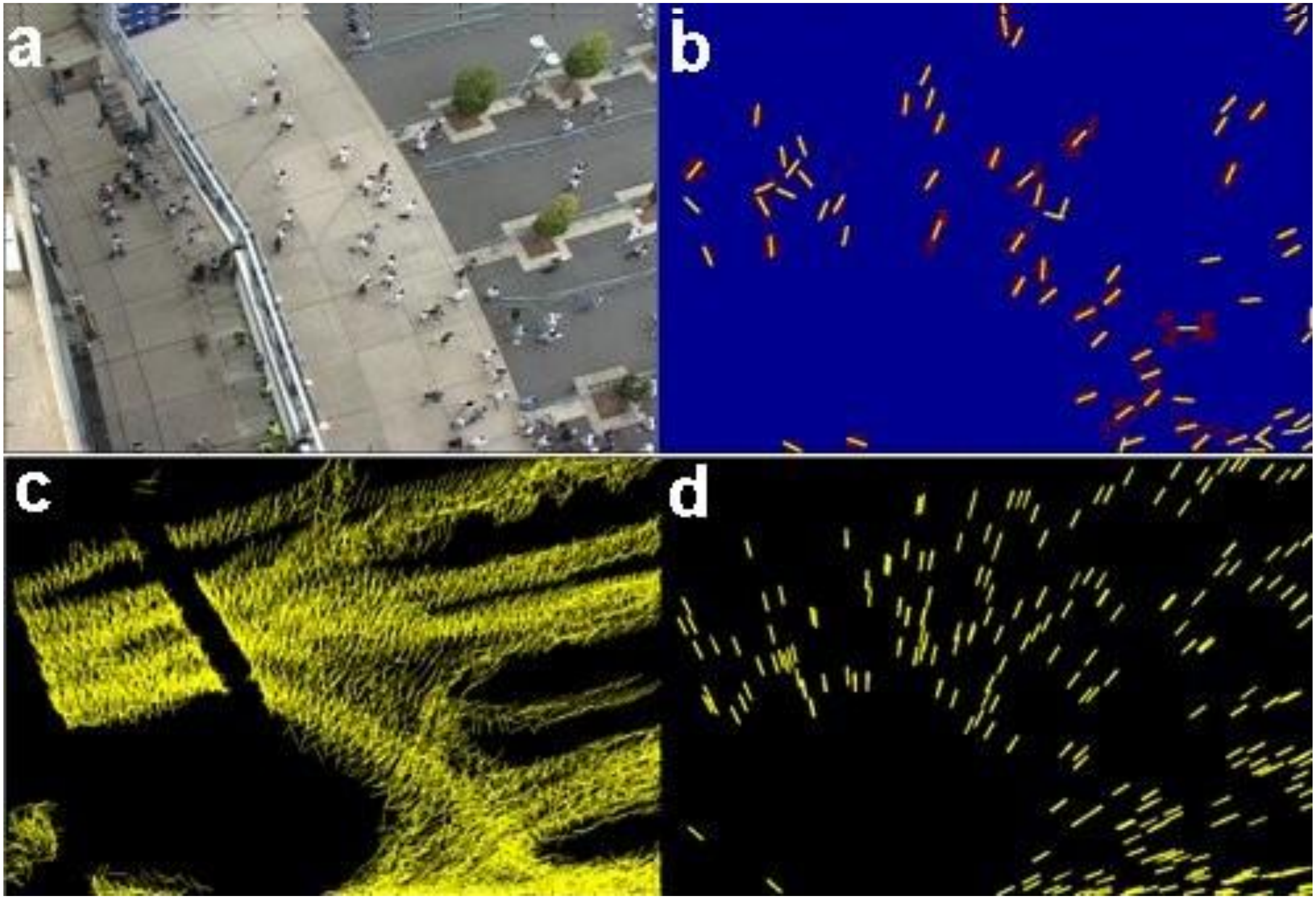
Example: learning height distribution as a function of image row



Example: learning orientation as a function of image location

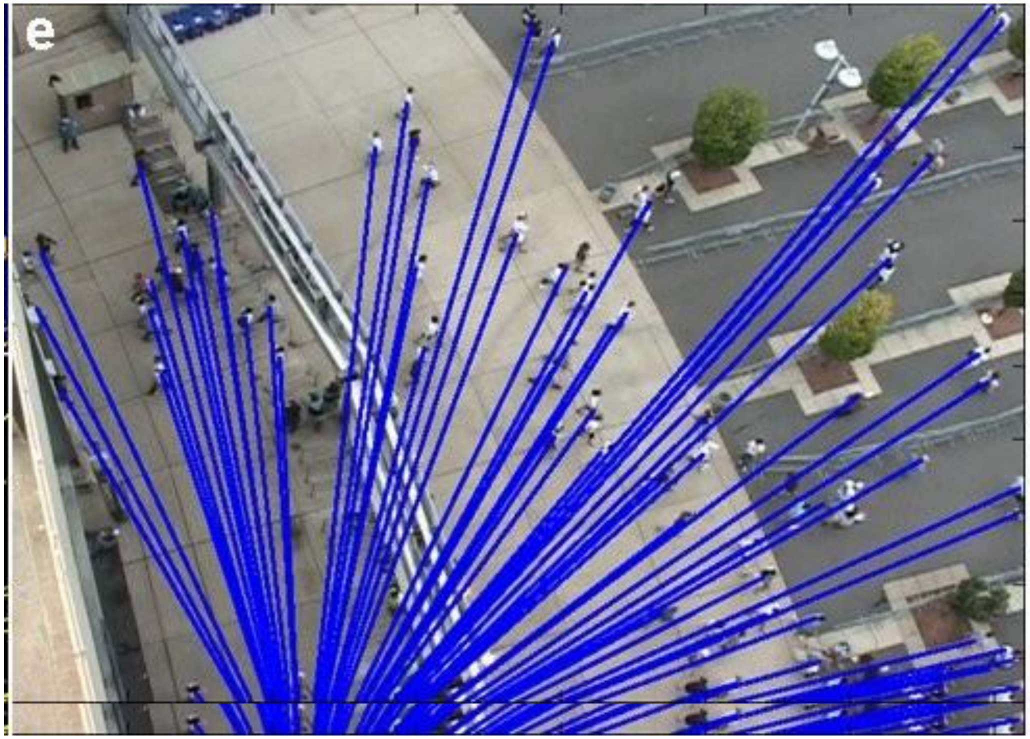
sample frame

extracted blobs / axes

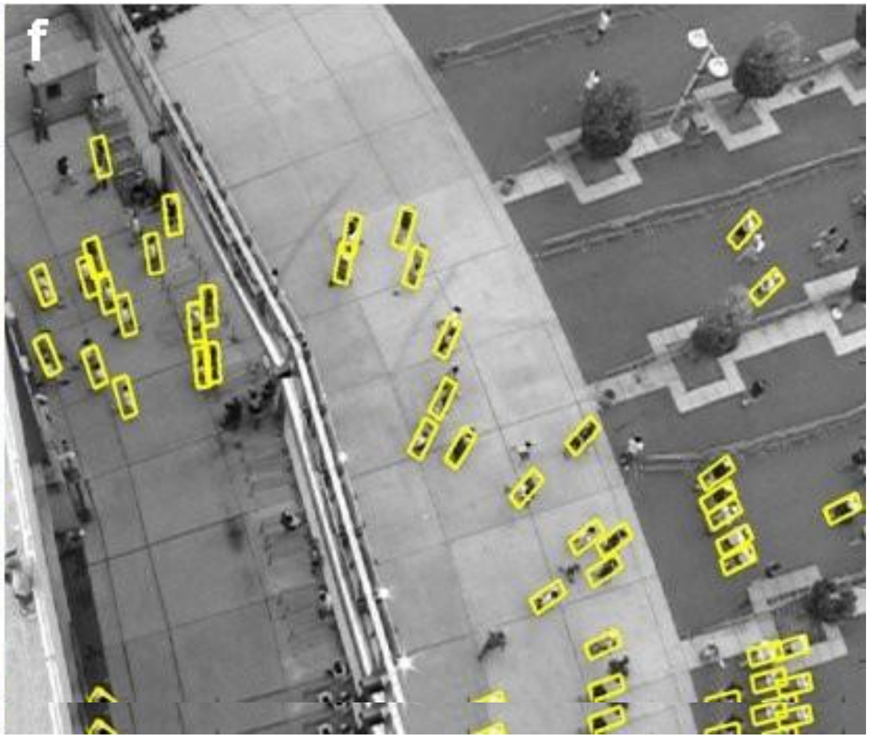


extracted from training sequence

inliers for VP estimation



estimated vanishing point



Scaled, oriented rectangles

Bottom line: it is not difficult to estimates priors on location,
size and orientation of people as seen from a specific viewpoint.

The space of configurations is very large. We can't exhaustively search for the max likelihood configuration. We can't even really uniformly sample the space to a reasonable degree of accuracy.

$$\text{config}_k = \{\{x_1,y_1,w_1,h_1,\text{theta}_1\},\{x_2,y_2,w_2,h_2,\text{theta}_2\},\ldots,\{x_k,y_k,w_k,h_k,\text{theta}_k\}\}$$

Let N = number of possible locations for (x_i,y_i) in a k-person configuration.

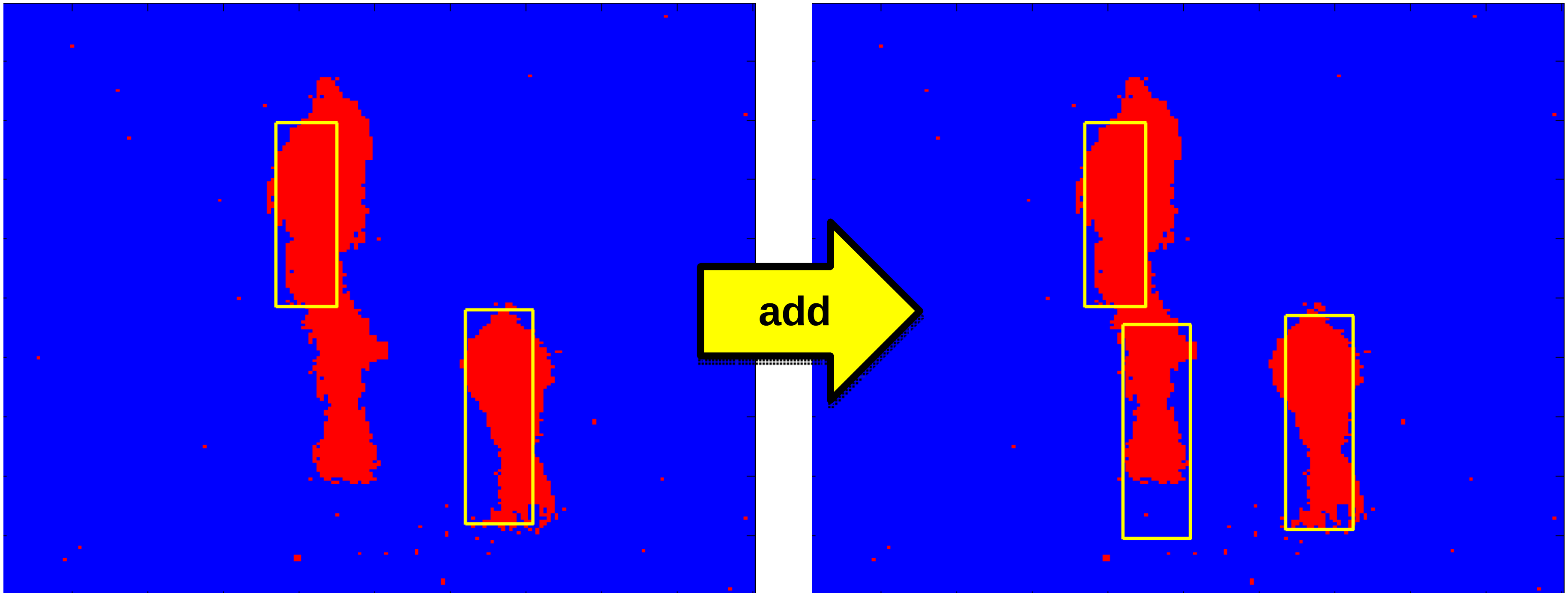
Size of config_k = N^k
And we don't even know how many people there are...

Size of config space = $N^0 + N^1 + N^2 + N^3 + \ldots$

If we also wanted to search for width, height and orientation, this space would be even more huge.

- Local Search Approach
  - Given a current configuration, propose a small change to it
  - Compare likelihood of proposed config with likelihood of the current config
  - Decide whether to accept the change

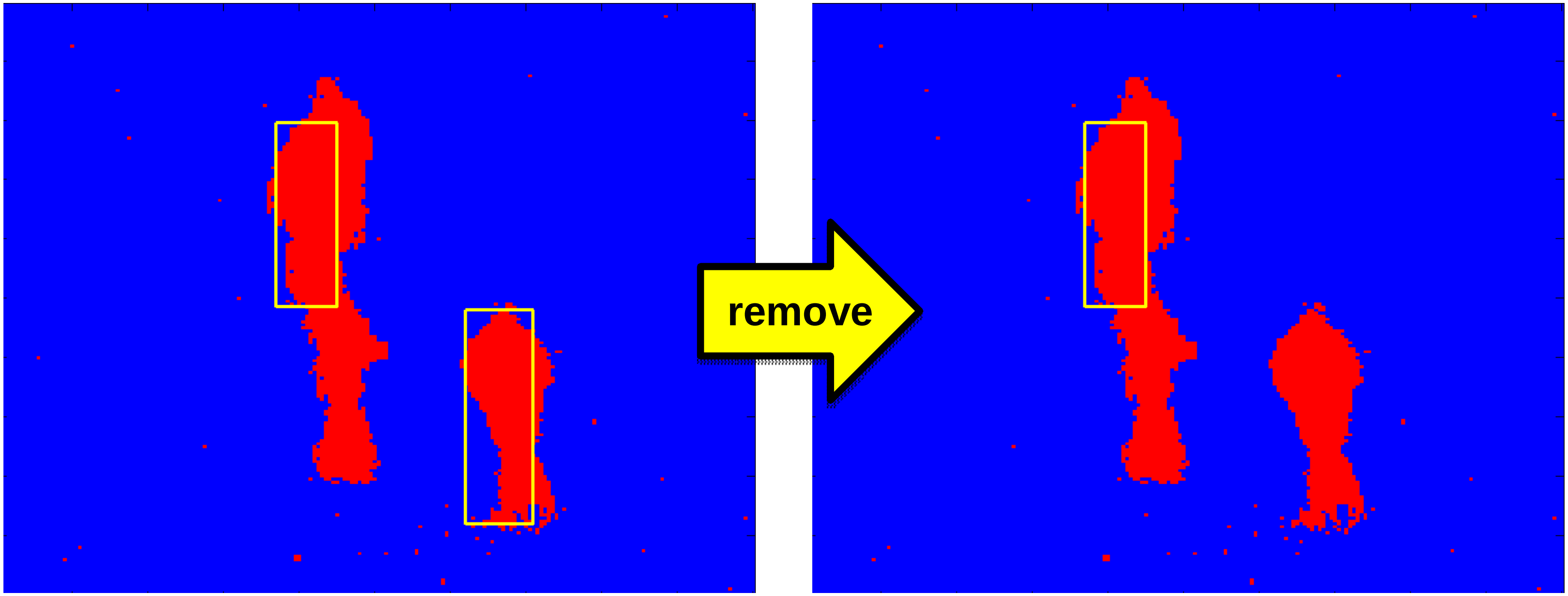
- Add a rectangle (birth)



current configuration

proposed configuration

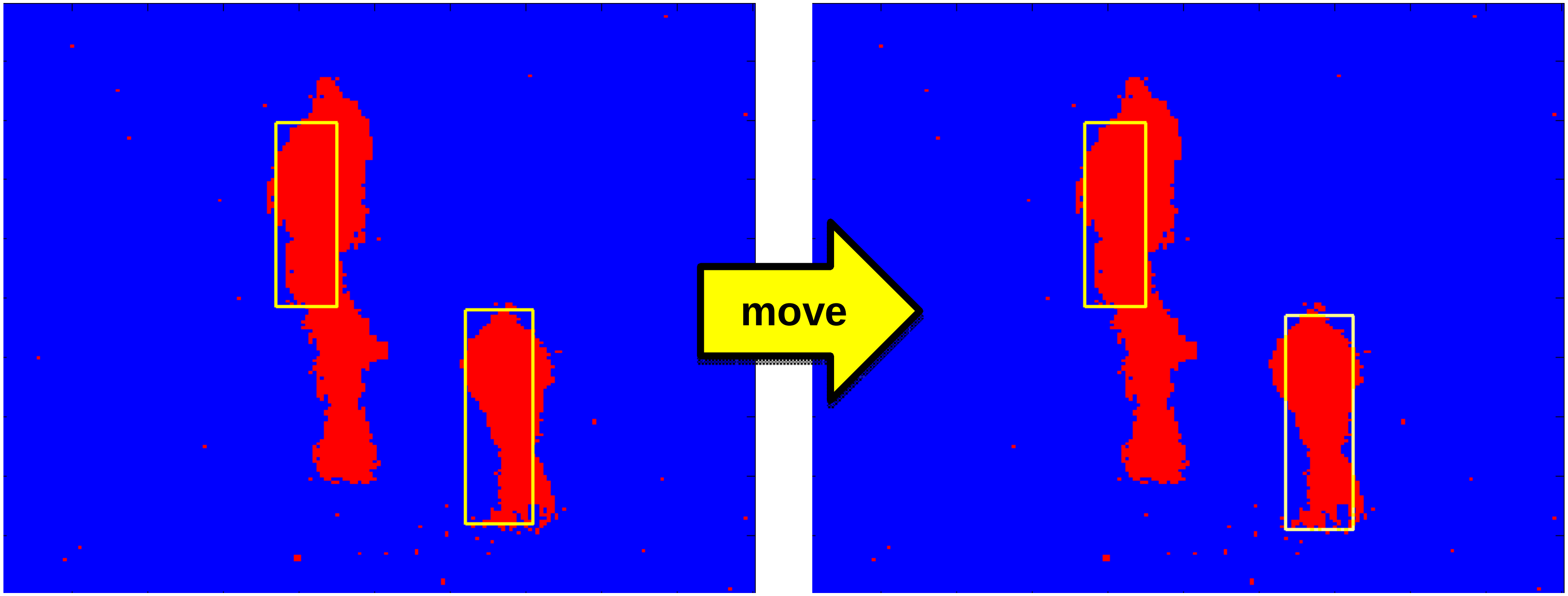
- Remove a rectangle (death)



current configuration

proposed configuration

- Move a rectangle



current configuration

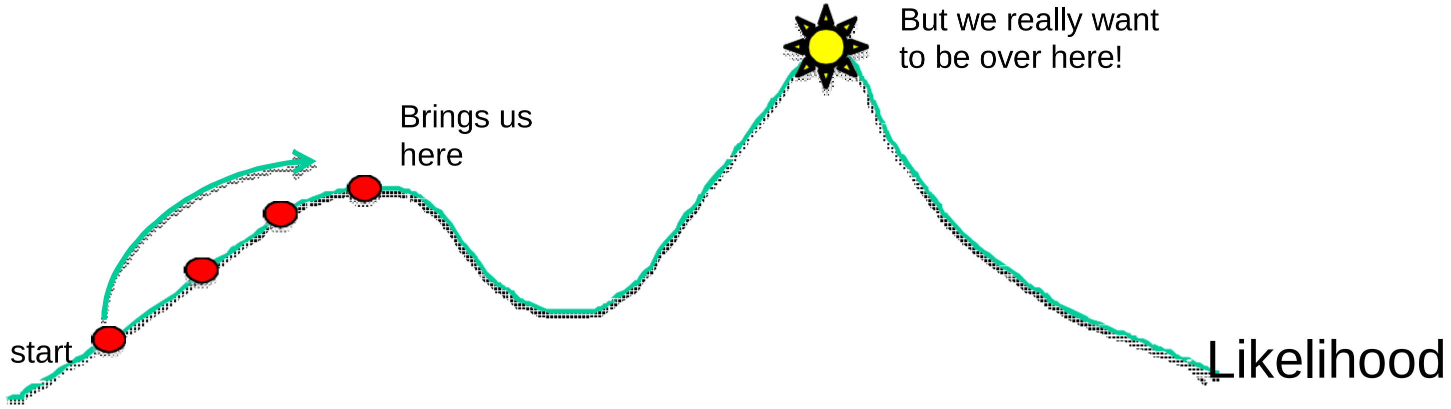
proposed configuration

- Naïve Acceptance
  - Accept proposed configuration if it has a larger likelihood score, i.e.

$$\text{Compute } a = \frac{L(\text{proposed})}{L(\text{current})}$$

$$\text{Accept if } a > 1$$

  - Problem: leads to hill-climbing behavior that gets stuck in local maxima



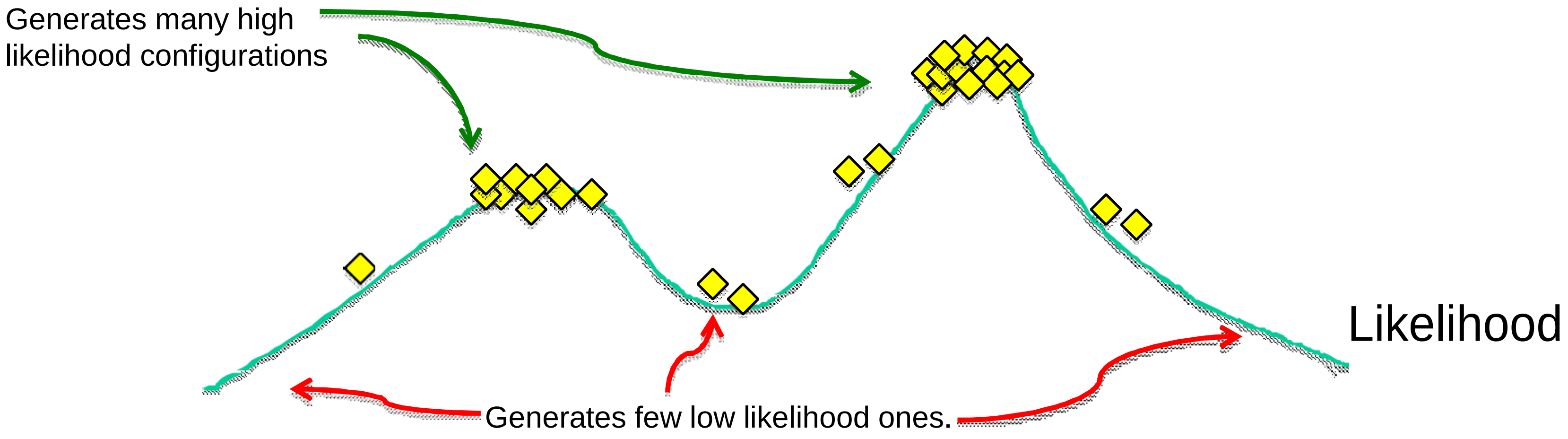
But we really want to be over here!

Brings us here

start

Likelihood

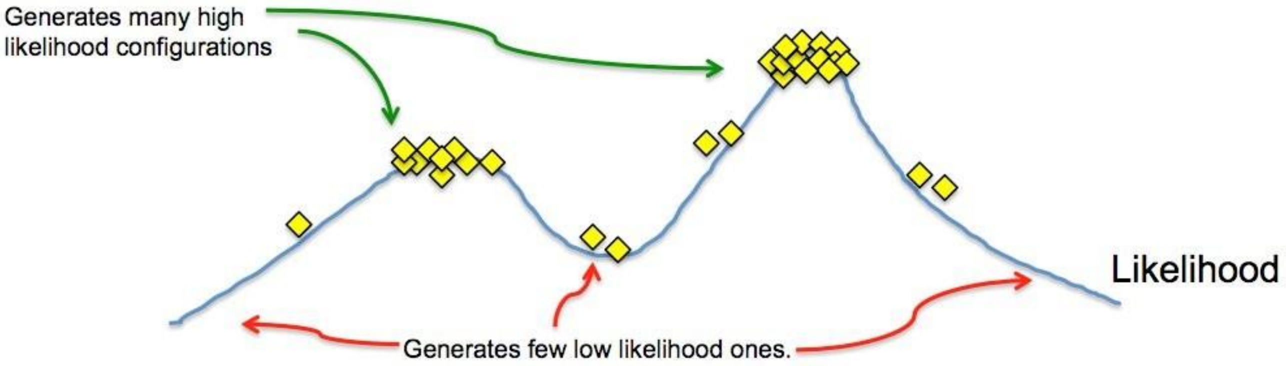
- The MCMC approach
  - Generate random configurations from a distribution proportional to the likelihood!



Generates many high likelihood configurations

Generates few low likelihood ones.

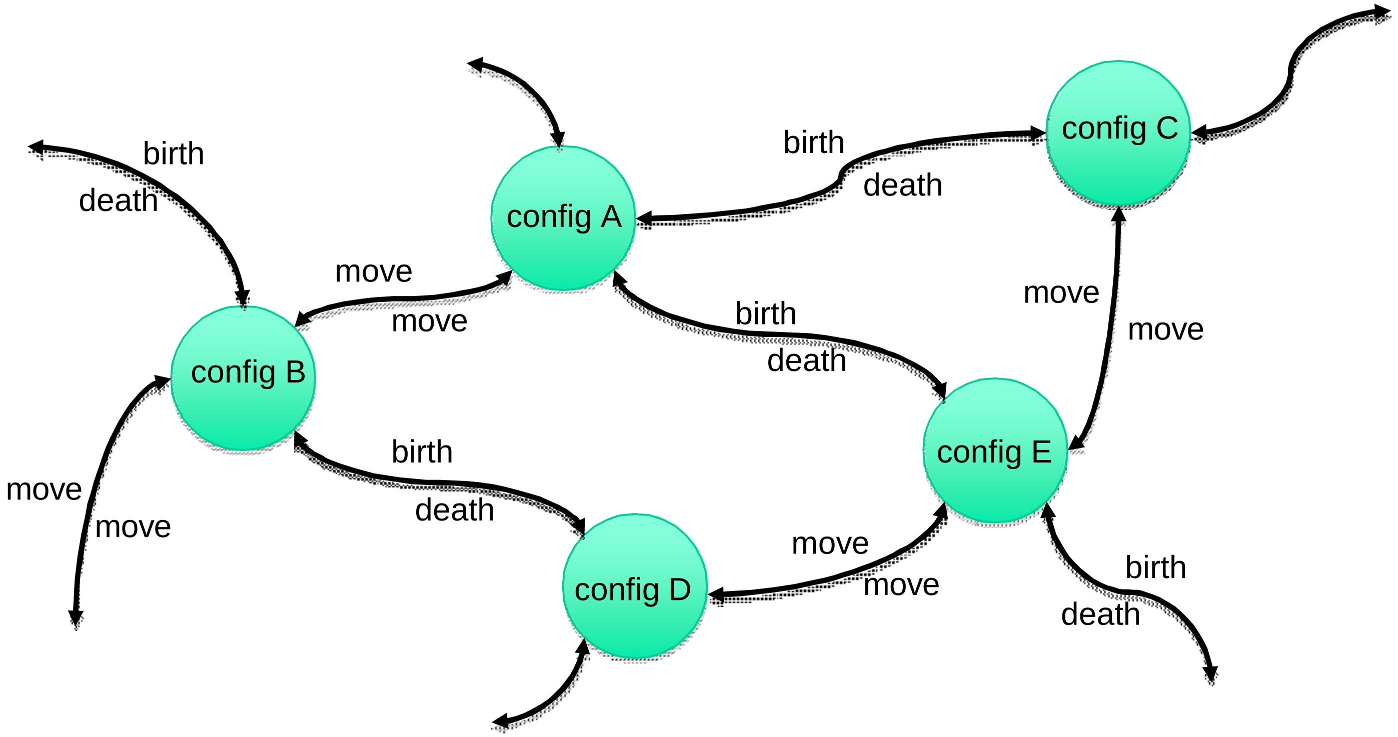
- The MCMC approach
  - Generate random configurations from a distribution proportional to the likelihood!


Generates many high likelihood configurations
Likelihood
Generates few low likelihood ones.

  - This searches the space of configurations in an efficient way.
  - Now just remember the generated configuration with the highest likelihood.

Sounds good, but how to do it?

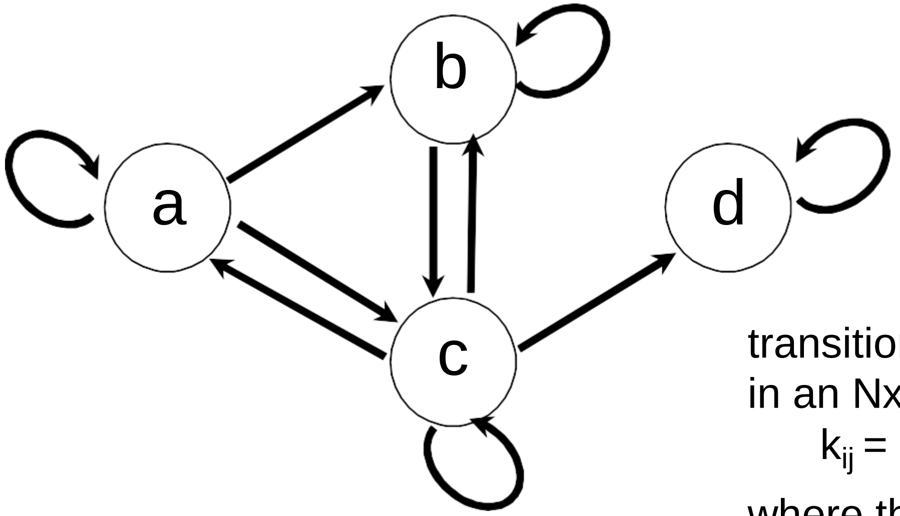
- Think of configurations as nodes in a graph.
- Put a link between nodes if you can get from one config to the other in one step (birth, death, move)



Recall: Markov Chains

Markov Chain:

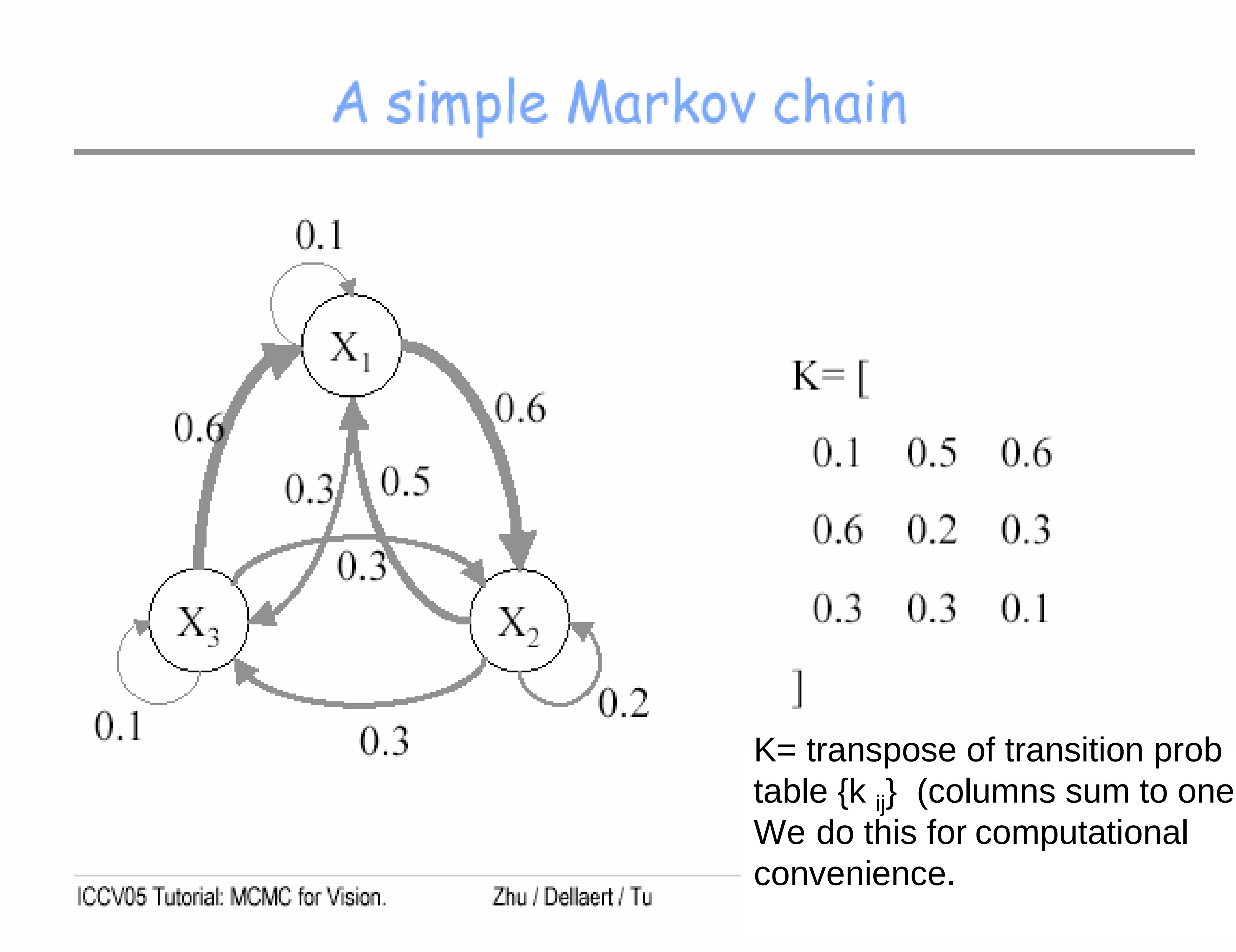
- A sequence of random variables X_1,X_2,X_3,...
- Each variable is a distribution over a set of states (a,b,c...)
- Transition probability of going to next state only depends on the current state. e.g. $P(X_{n+1} = a \mid X_n = b)$



transition probs can be arranged in an NxN table of elements

$k_{ij} = P(X_{n+1}=j \mid X_n = i)$

where the rows sum to one



A simple Markov chain

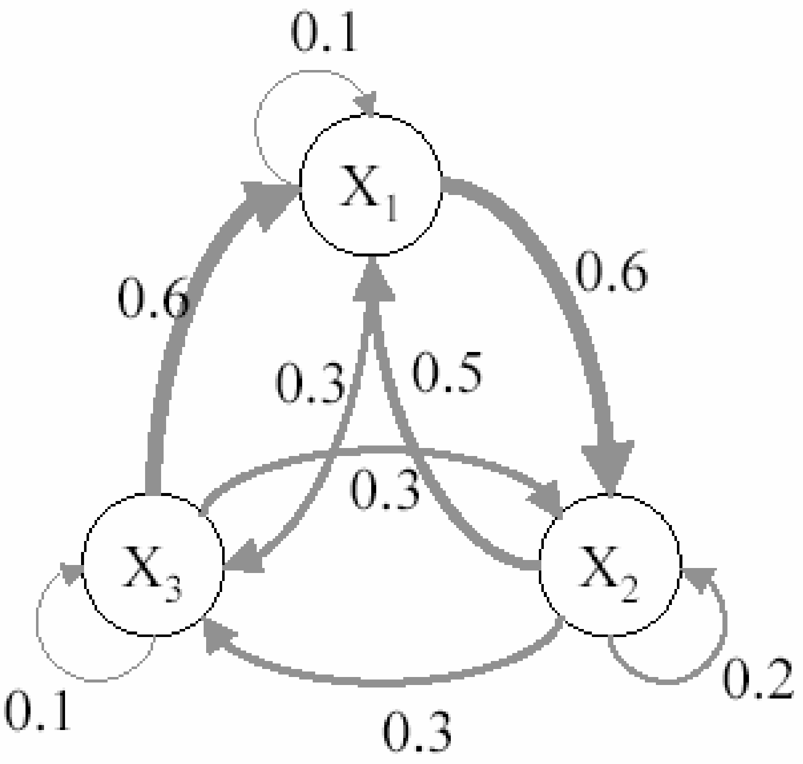
$$K = \begin{bmatrix} 0.1 & 0.5 & 0.6 \\ 0.6 & 0.2 & 0.3 \\ 0.3 & 0.3 & 0.1 \end{bmatrix}$$

ICCV05 Tutorial: MCMC for Vision. Zhu / Dellaert / Tu

K= transpose of transition prob table $\{k_{ij}\}$ (columns sum to one. We do this for computational convenience.

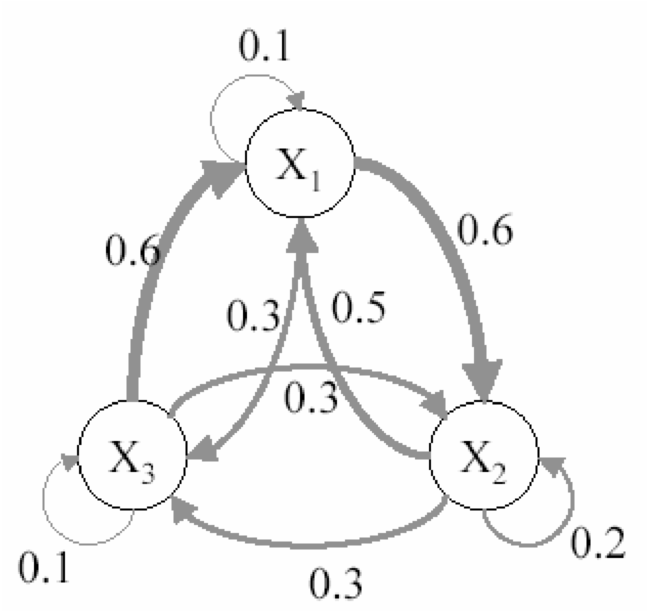
Question:

Assume you start in some state, and then run the simulation for a large number of time steps. What percentage of time do you spend at X1, X2 and X3?



$$K = \begin{bmatrix} 0.1 & 0.5 & 0.6 \\ 0.6 & 0.2 & 0.3 \\ 0.3 & 0.3 & 0.1 \end{bmatrix}$$

Experimental Approach



Start in some state, and then run the simulation for some number of time steps. After you have run it “long enough” start keeping track of the states you visit.

{... X1 X2 X1 X3 X3 X2 X1 X2 X1 X1 X3 X3 X2 ...}

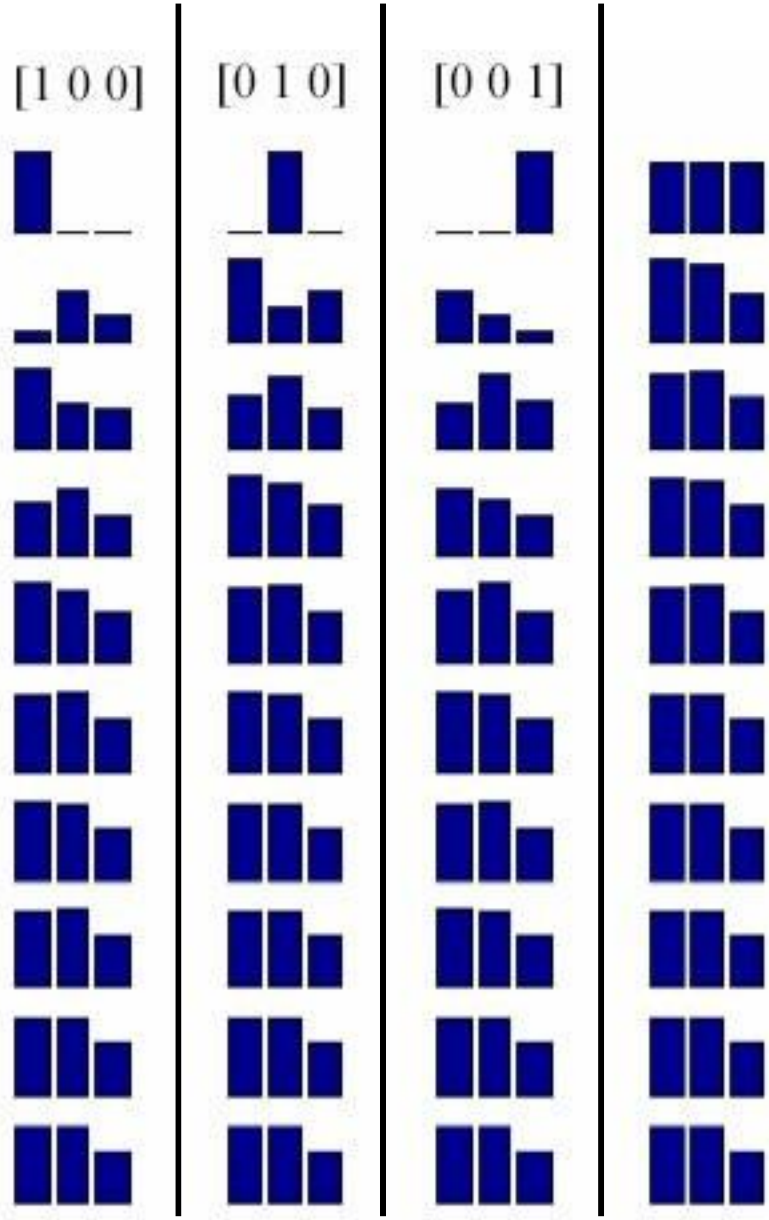
These are samples from the distribution you want, so you can now compute any expected values with respect to that distribution empirically.

UTS, Sean. He

Analytic Approach

four possible initial distributions

[.33 .33 .33]



initial distribution

distribution after one time step

q_0

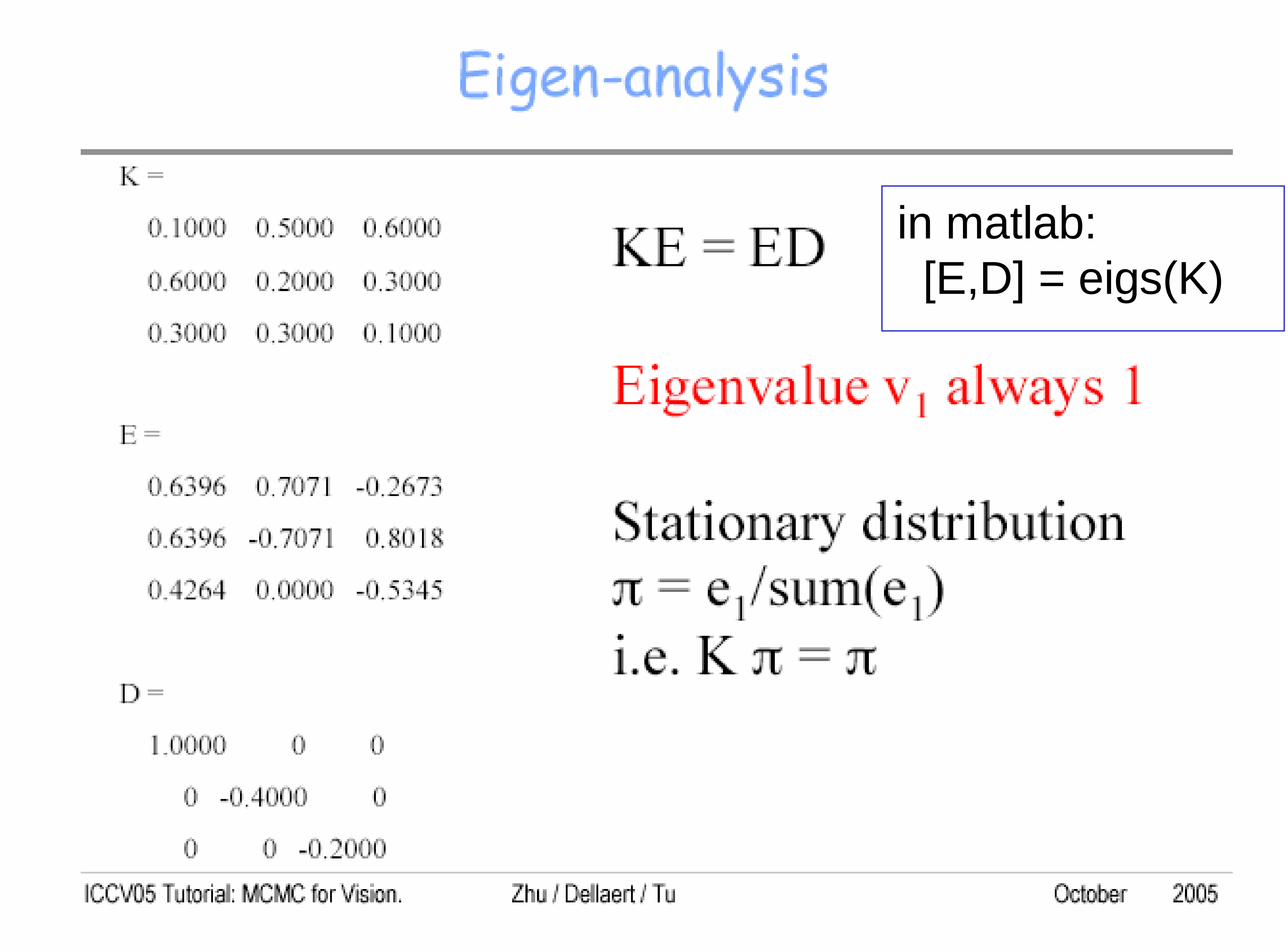
$q_1 = K\ q_0$

$q_2 = K\ q_1 = K^2\ q_0$

$q_3 = K\ q_2 = K^2\ q_1 = K^3\ q_0$

$q_{10} = K\ q_9 = \ldots\ K^{10}\ q_0$

all eventually end up with same distribution -- this is the stationary distribution!



Eigen-analysis

```
K =

   0.1000    0.5000    0.6000

   0.6000    0.2000    0.3000

   0.3000    0.3000    0.1000


E =

   0.6396    0.7071   -0.2673

   0.6396   -0.7071    0.8018

   0.4264    0.0000   -0.5345


D =

   1.0000         0         0

        0   -0.4000         0

        0         0   -0.2000
```

$KE = ED$

in matlab:
[E,D] = eigs(K)

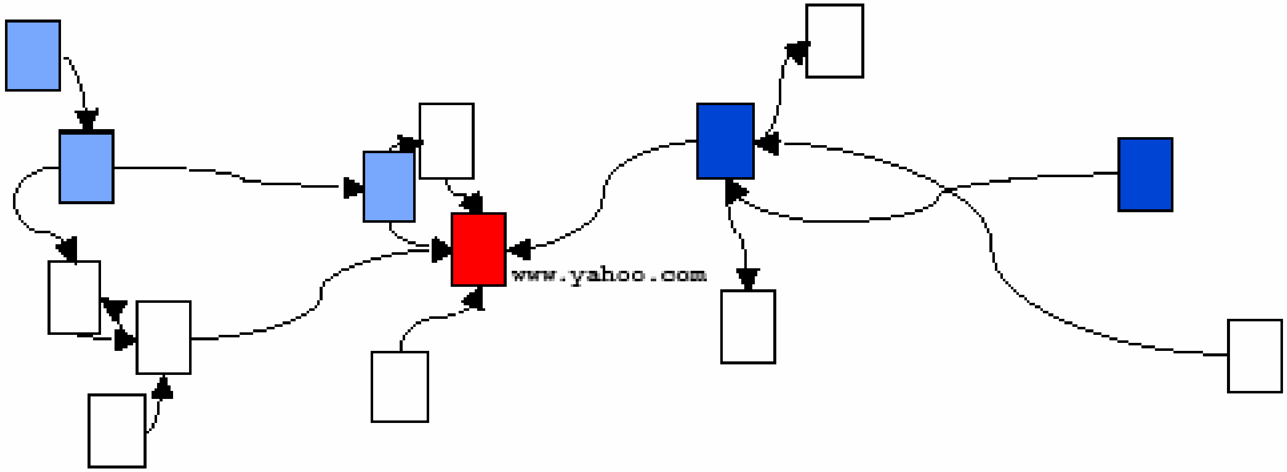
Eigenvalue v_1 always 1

Stationary distribution

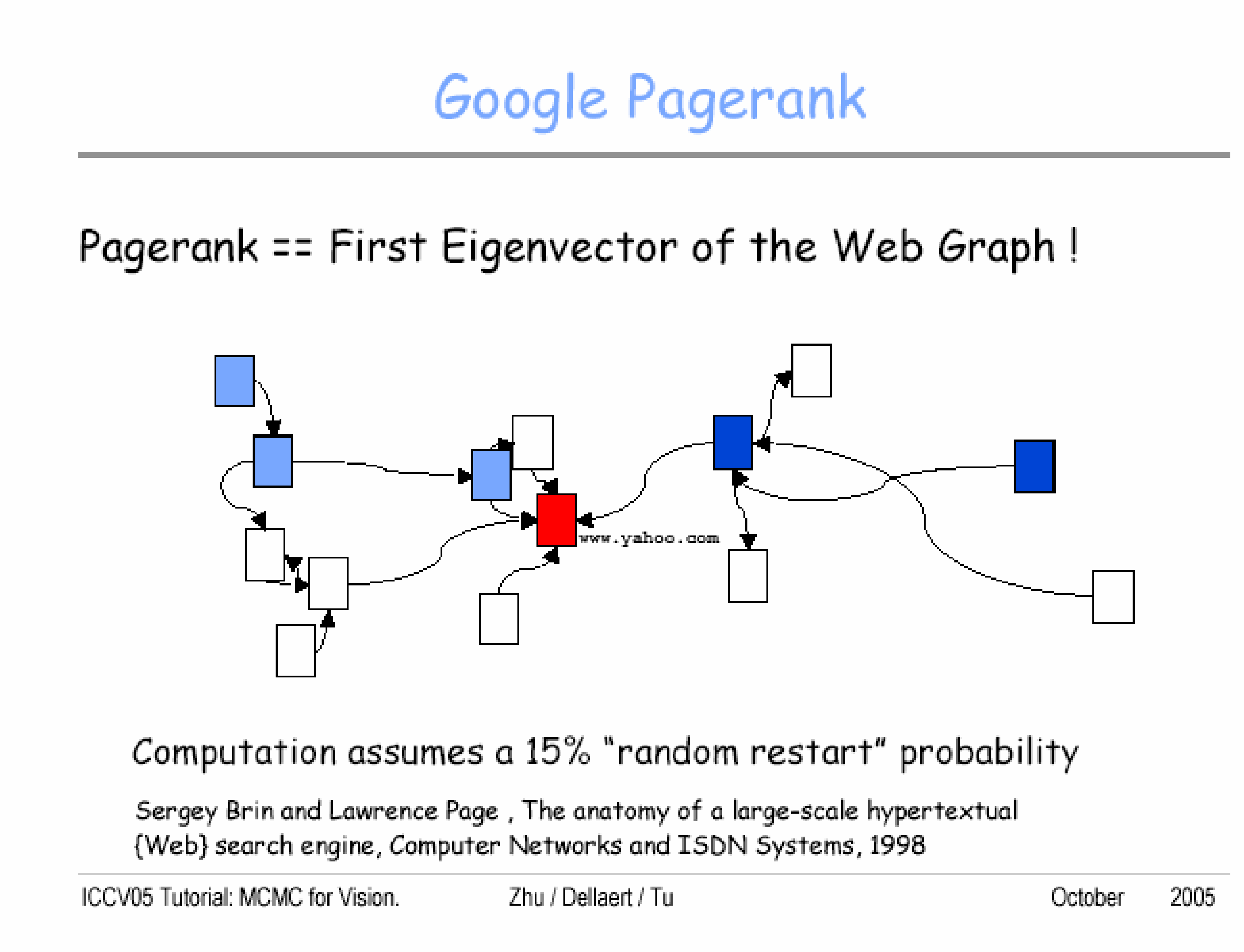
$\pi = e_1/\text{sum}(e_1)$

i.e. $K\,\pi = \pi$

The Web as a Markov Chain



The PageRank of a webpage as used by Google is defined by a Markov chain. It is the probability to be at page *i* in the stationary distribution on the following Markov chain on all (known) webpages. If *N* is the number of known webpages, and a page *i* has *ki* links then it has transition probability (1-*q*)/*ki* + *q*/*N* for all pages that are linked to and *q*/*N* for all pages that are not linked to. The parameter *q* is taken to be about 0.15.



Google Pagerank

Pagerank == First Eigenvector of the Web Graph !

Computation assumes a 15% "random restart" probability

Sergey Brin and Lawrence Page , The anatomy of a large-scale hypertextual {Web} search engine, Computer Networks and ISDN Systems, 1998

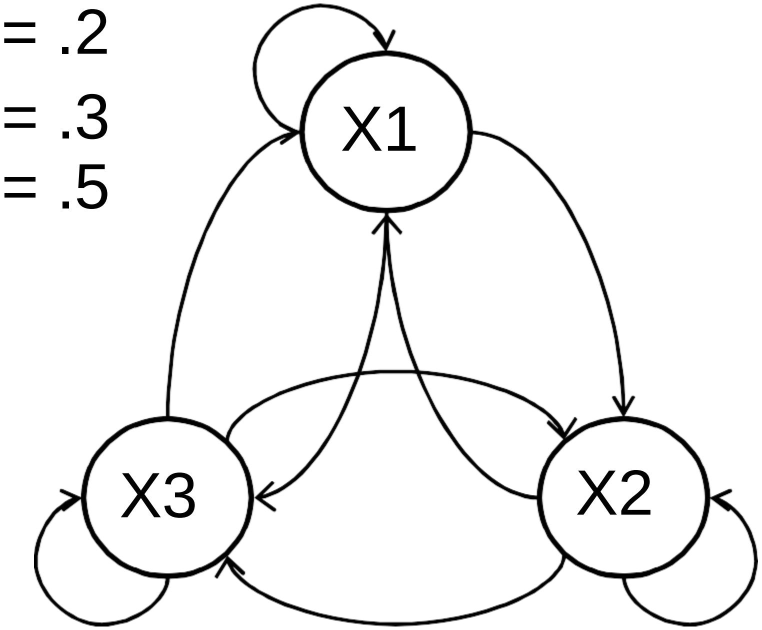
How to Design a Chain?

Assume you want to spend a particular percentage of time at X1, X2 and X3. What should the transition probabilities be?

P(x1) = .2

P(x2) = .3

P(x3) = .5



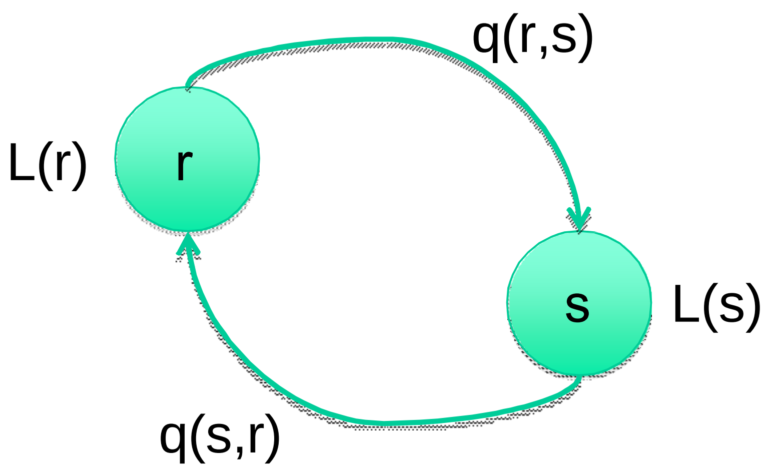
$$K = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix}$$

- Consider a pair of configuration nodes r,s
- Want to generate them with frequency relative to their likelihoods L(r) and L(s)
- Let q(r,s) be relative frequency of proposing configuration s when the current state is r (and vice versa)

A sufficient condition to generate r,s with the desired frequency is

$$L(r)\,q(r,s) \;=\; L(s)\,q(s,r)$$

“detailed balance”



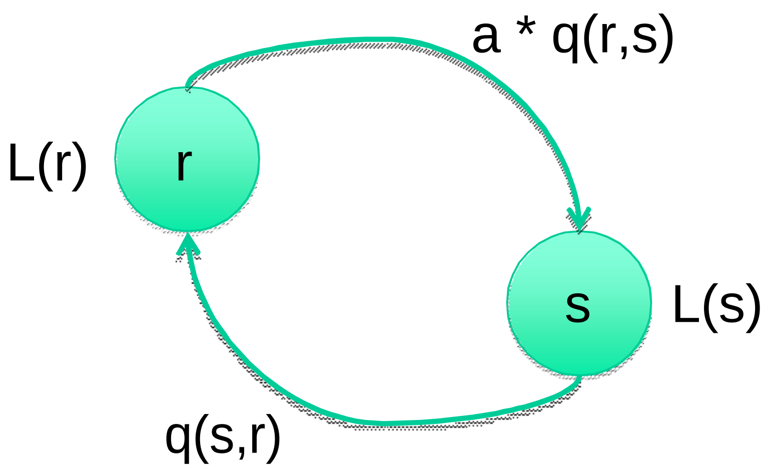
- Typically, your proposal frequencies do NOT satisfy detailed balance (unless you are extremely lucky).
- To “fix this”, we introduce a computational fudge factor a

Detailed balance:

$$a^* L(r)\, q(r,s) = L(s)\, q(s,r)$$

Solve for a:

$$a = \frac{L(s)\, q(s,r)}{L(r)\, q(r,s)}$$



MCMC Sampling

- Metropolis Hastings algorithm

Propose a new configuration

Compute $a = \frac{L(\text{proposed})}{L(\text{current})} \; \frac{q(\text{proposed,current})}{q(\text{current,proposed})}$

Accept if a > 1

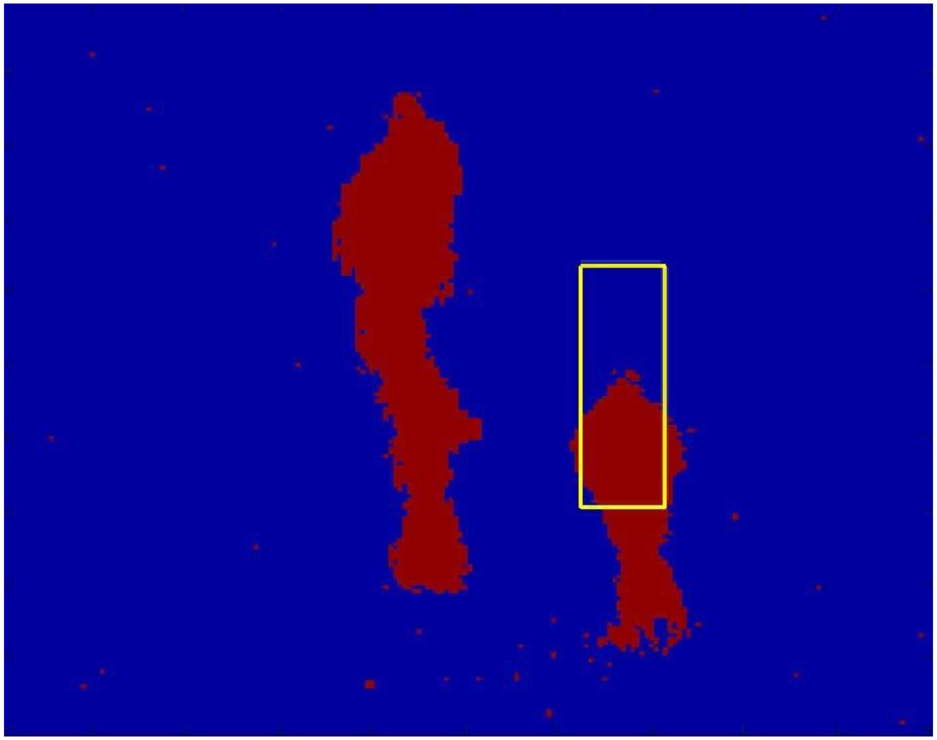
Else accept anyways with probability a

Difference from Naïve algorithm

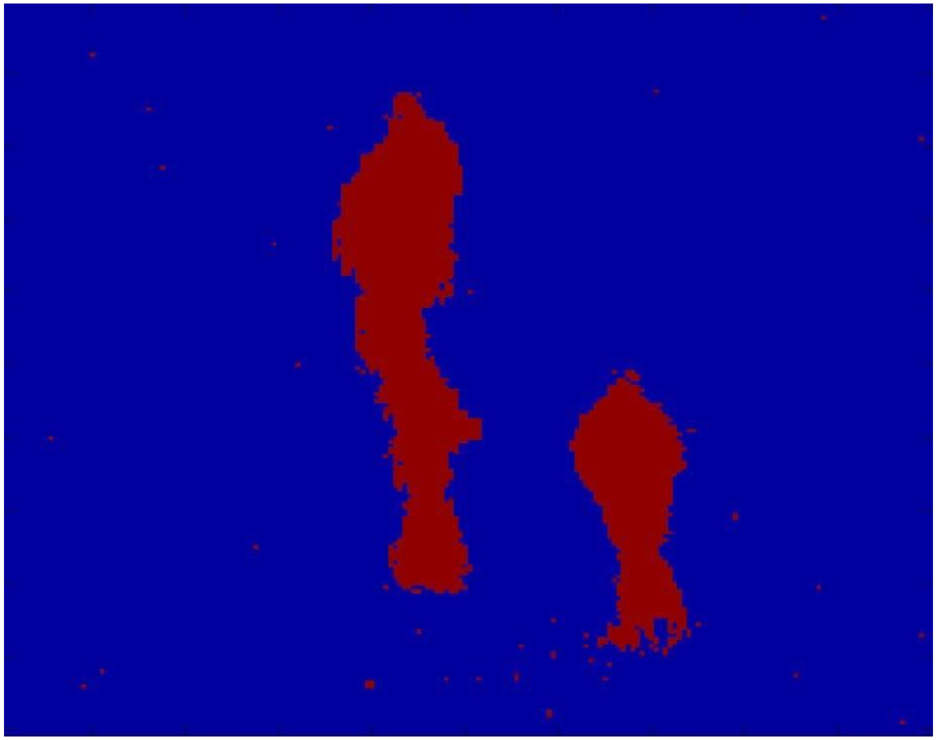
Trans-dimensional MCMC

- Green’s reversible-jump approach (RJMCMC) gives a general template for exploring and comparing states of differing dimension (diff numbers of rectangles in our case).
- Proposals come in reversible pairs: birth/death and move/move.
- We should add another term to the acceptance ratio for pairs that jump across dimensions. However, that term is 1 for our simple proposals.

Sequence of proposed configurations



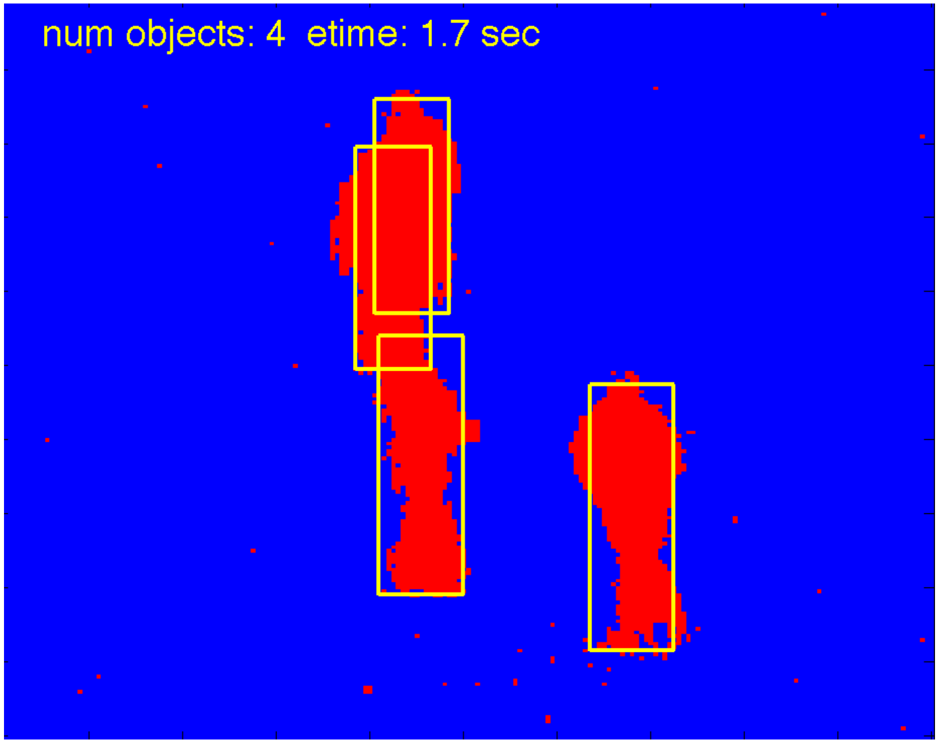
Sequence of “best” configurations



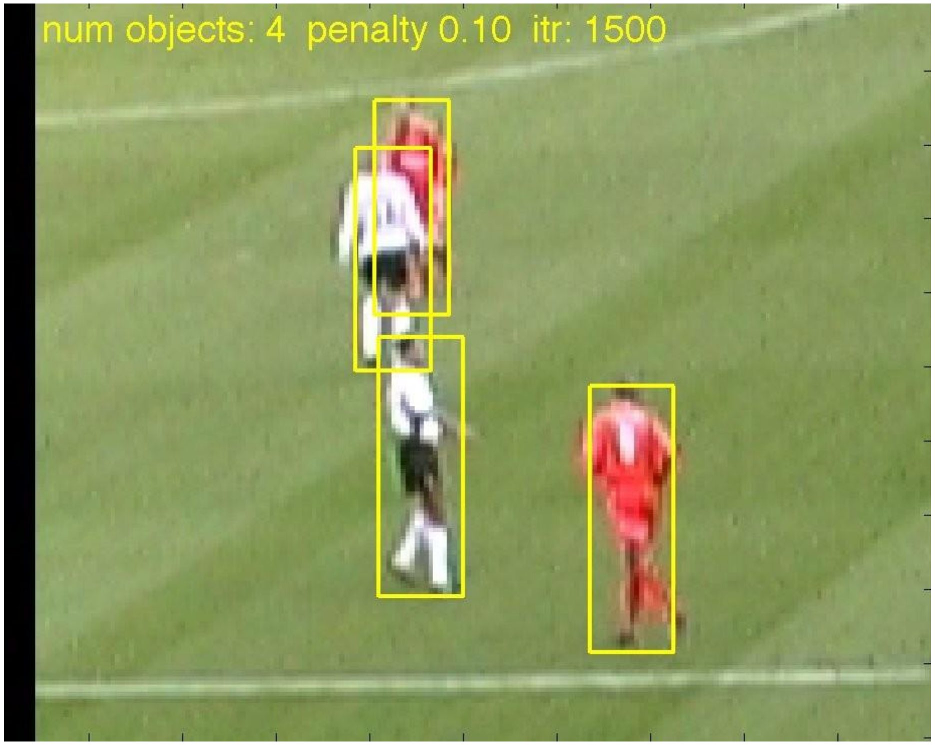
movies

UTS, Sean. He

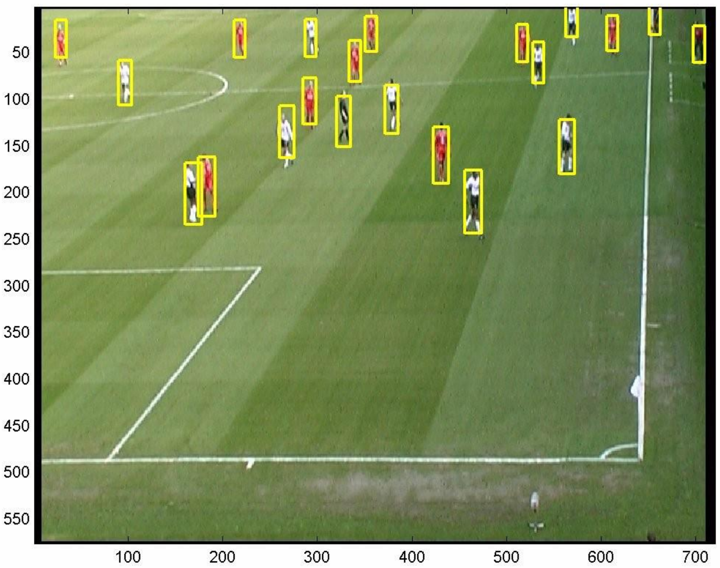
MAP configuration

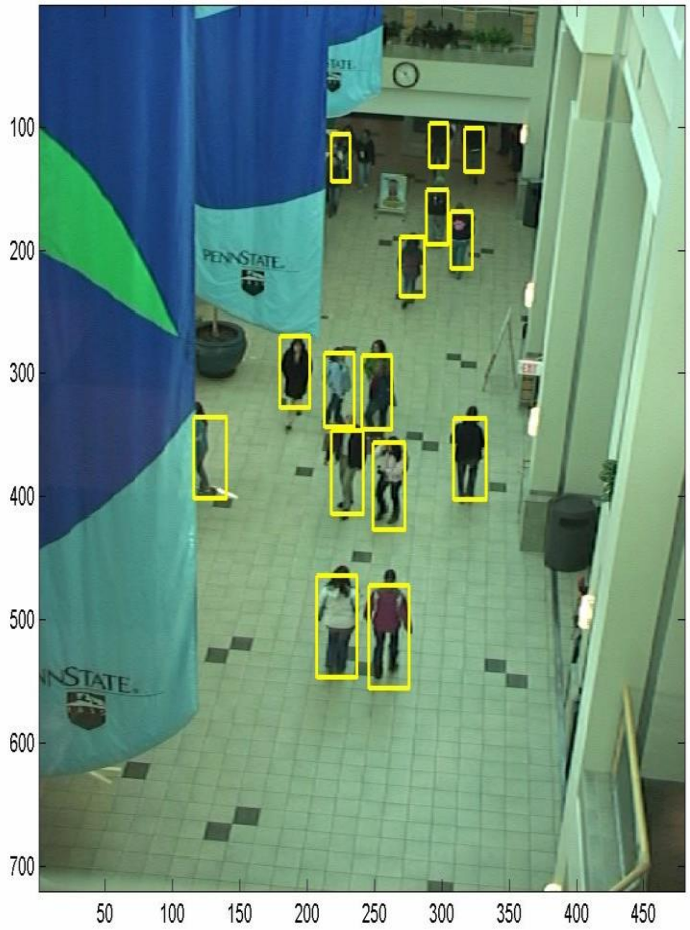


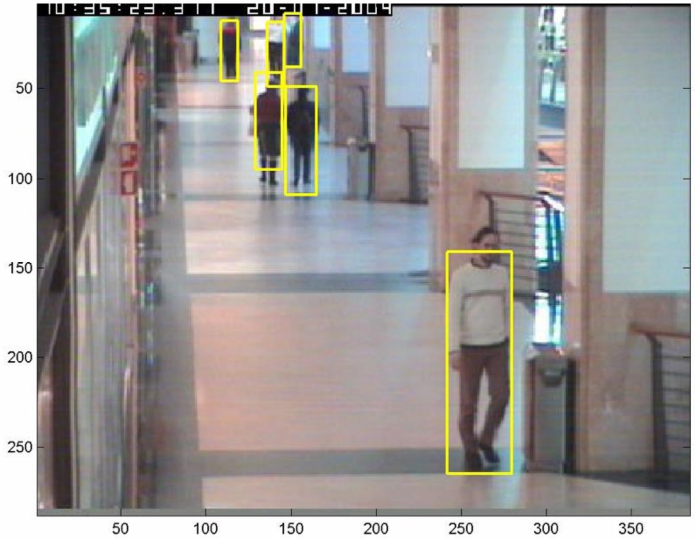
Looking good!



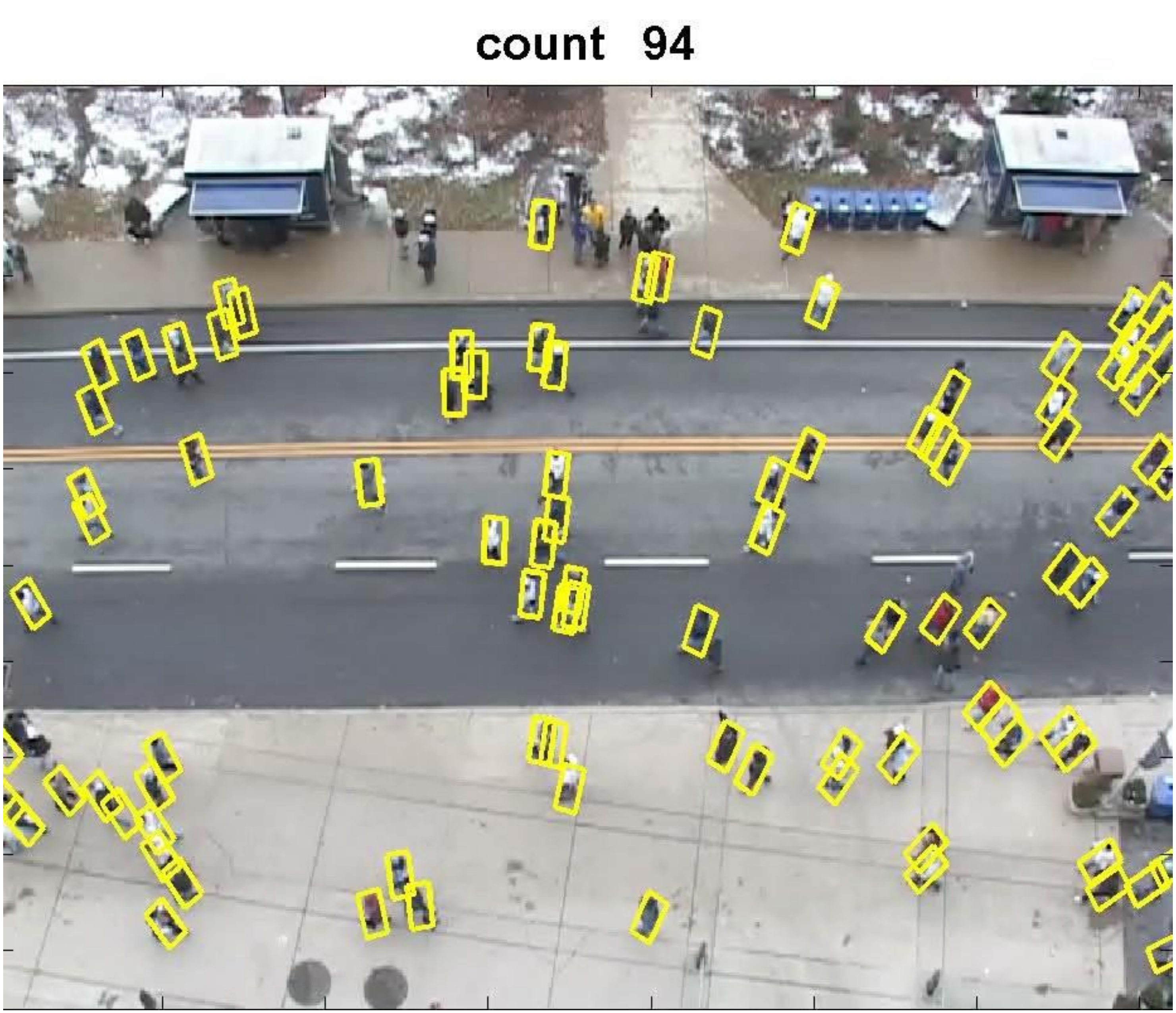
Examples


50
100
150
200
250
300
350
400
450
500
550
100
200
300
400
500
600
700

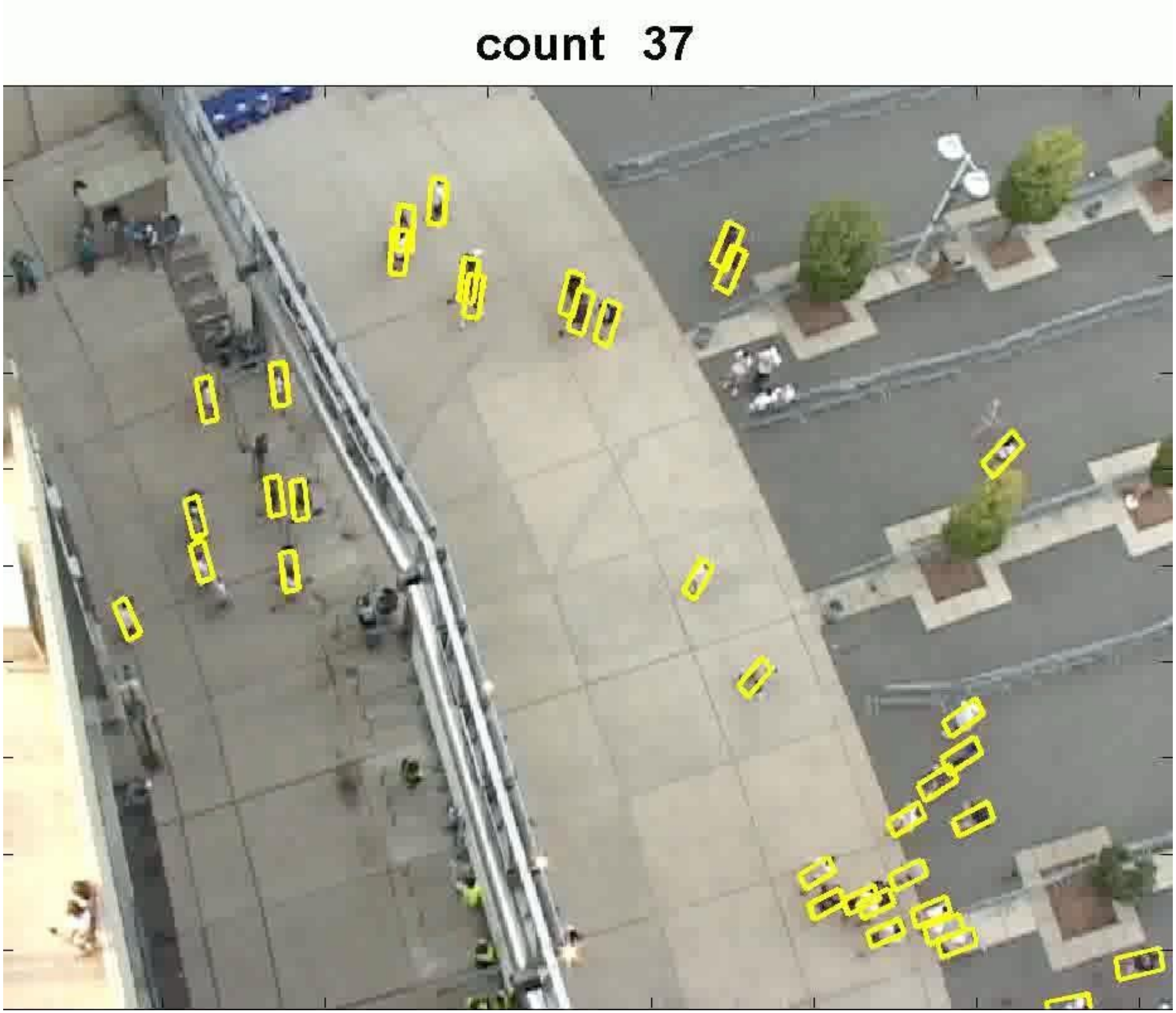

100
200
300
400
500
600
700
50
100
150
200
250
300
350
400
450


50
100
150
200
250
50
100
150
200
250
300
350

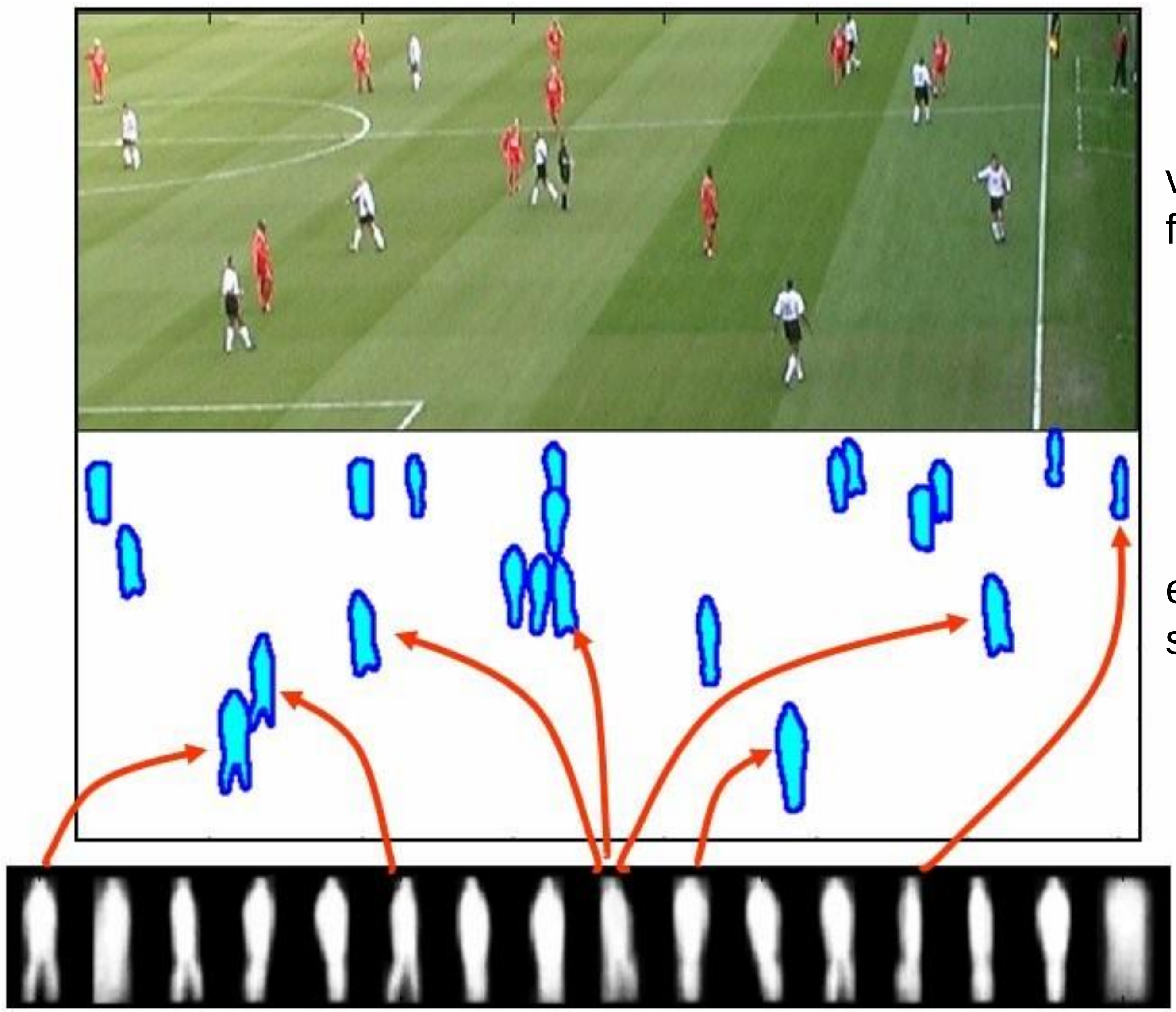
Example: Nov 22, Curtin Road



Example: Sep 6, Gate A



Adding Shape to the Estimation

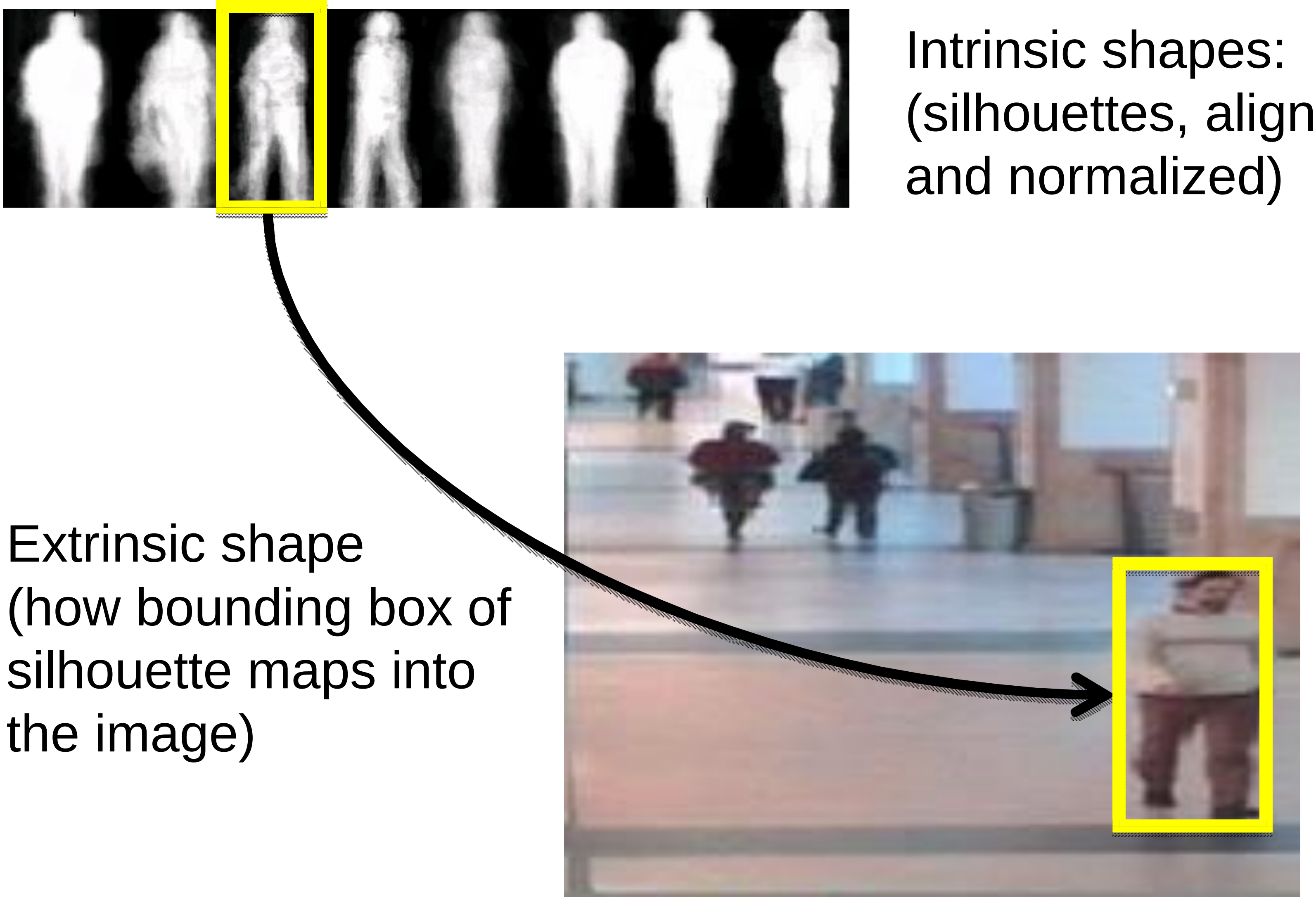


video frame

estimated state

library of candidate shapes

Intrinsic vs Extrinsic Shape



Intrinsic shapes:
(silhouettes, aligned
and normalized)

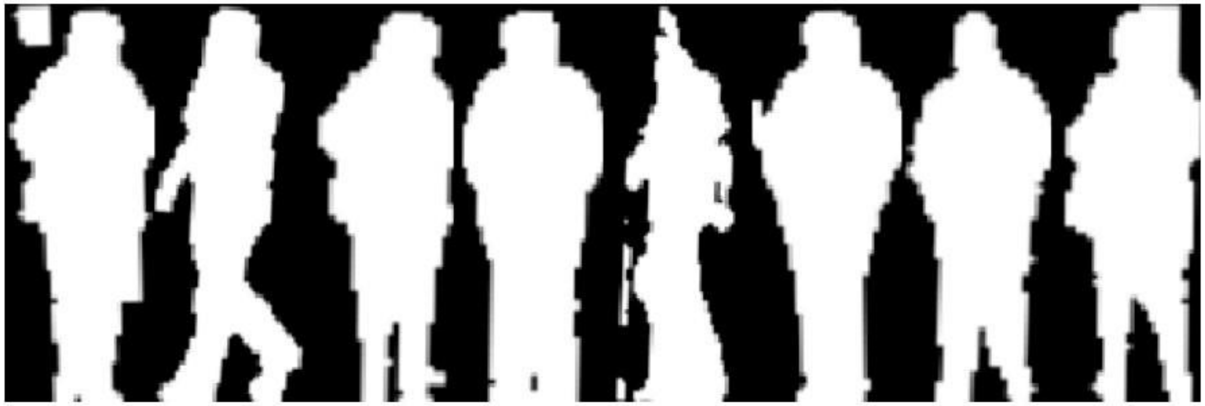
Extrinsic shape
(how bounding box of
silhouette maps into
the image)

Revised Prior Model

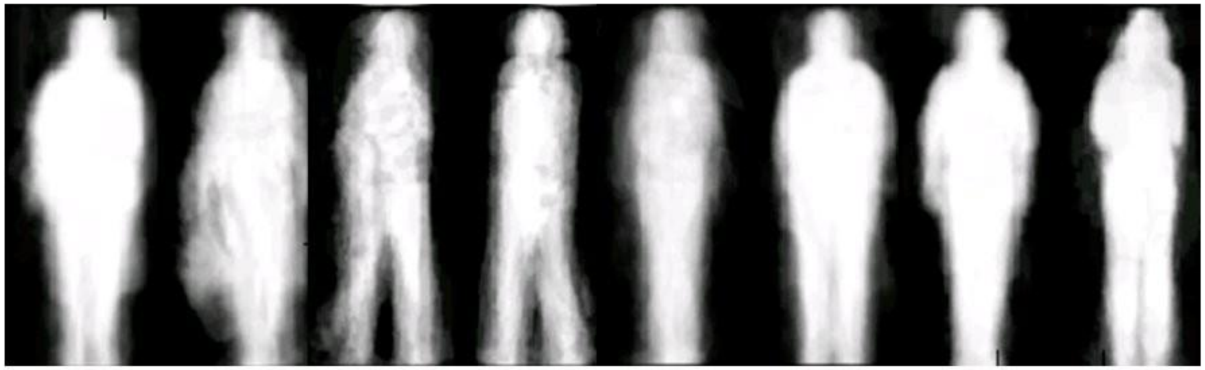
$$\underbrace{\pi(\mathbf{o}_i)}_{\substack{\text{Prior for}\\ \text{object i}}} = \underbrace{\pi(p_i)\pi(w_i, h_i, \theta_i|p_i)}_{\substack{\text{Prior on extrinsic shape}\\ \text{(location + bounding box}\\ \text{height, width and orientation)}}}\underbrace{\pi(s_i)}_{\substack{\text{Prior on intrinsic}\\ \text{shape selection}}}$$

Silhoutte shape represented as a Bernoulli mixture model

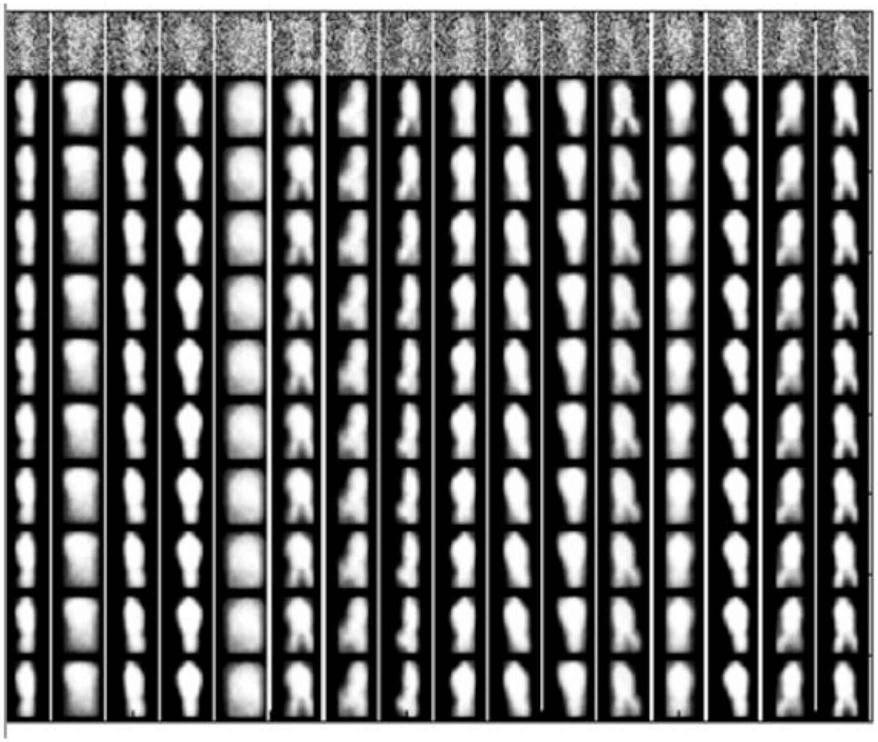
$$p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\pi}) = \textstyle\sum_{k=1}^{K} \pi_k p(x|\boldsymbol{\mu}_k)$$



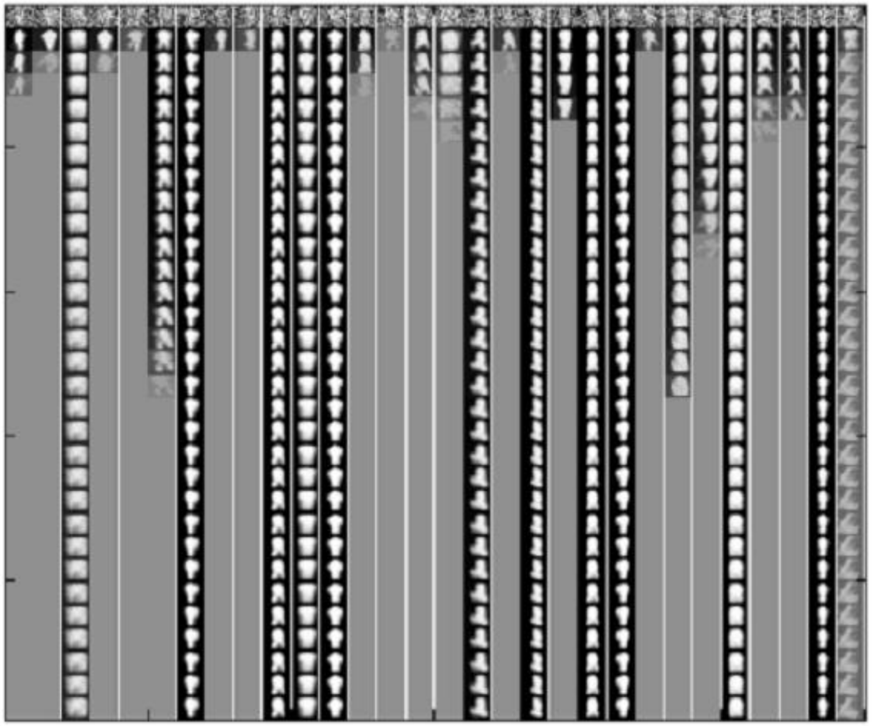
Training shapes (foreground masks)



Learned Bernoulli mixture model (“soft” silhouettes) “standard” EM



Bayesian EM with Dirichlet prior

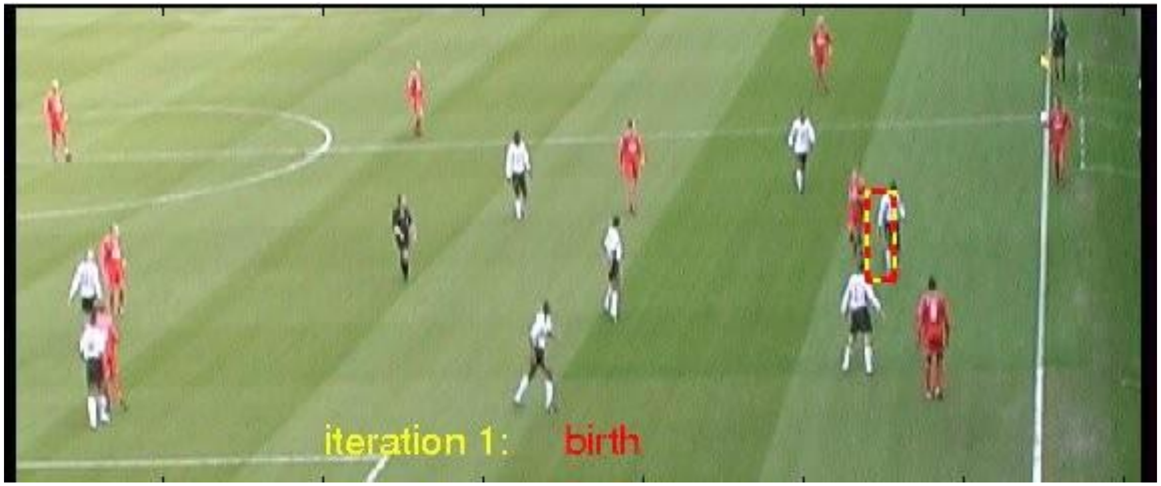


Provides automatic selection of number of mixture components

UTS, Sean. He

Adding Shape to the Estimation

MCMC iterations



movie

MCMC proposes changes to current configuration

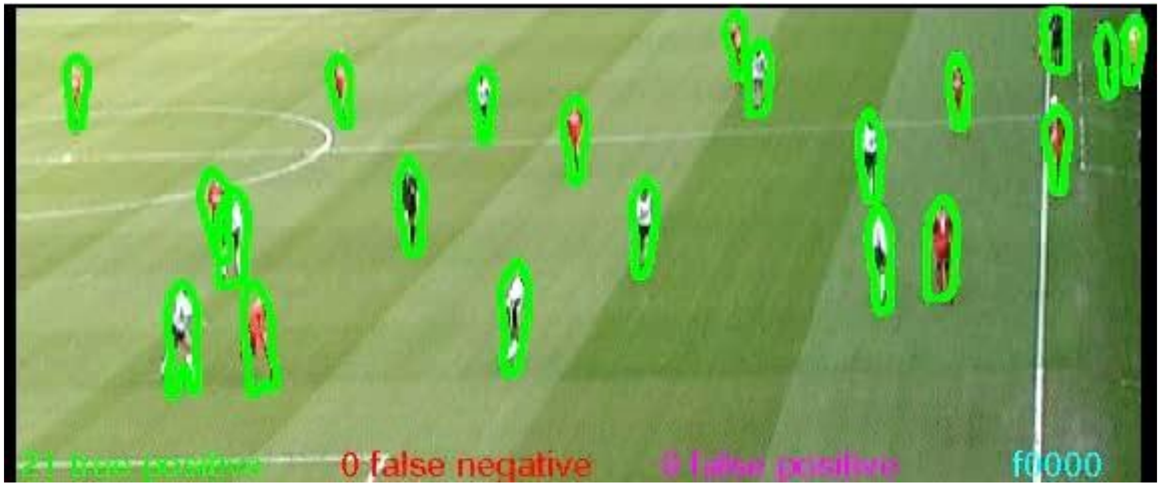
- add/remove a person
- shift location of person
- **change their shape**

UTS, Sean. He

VSPETS Soccer Dataset

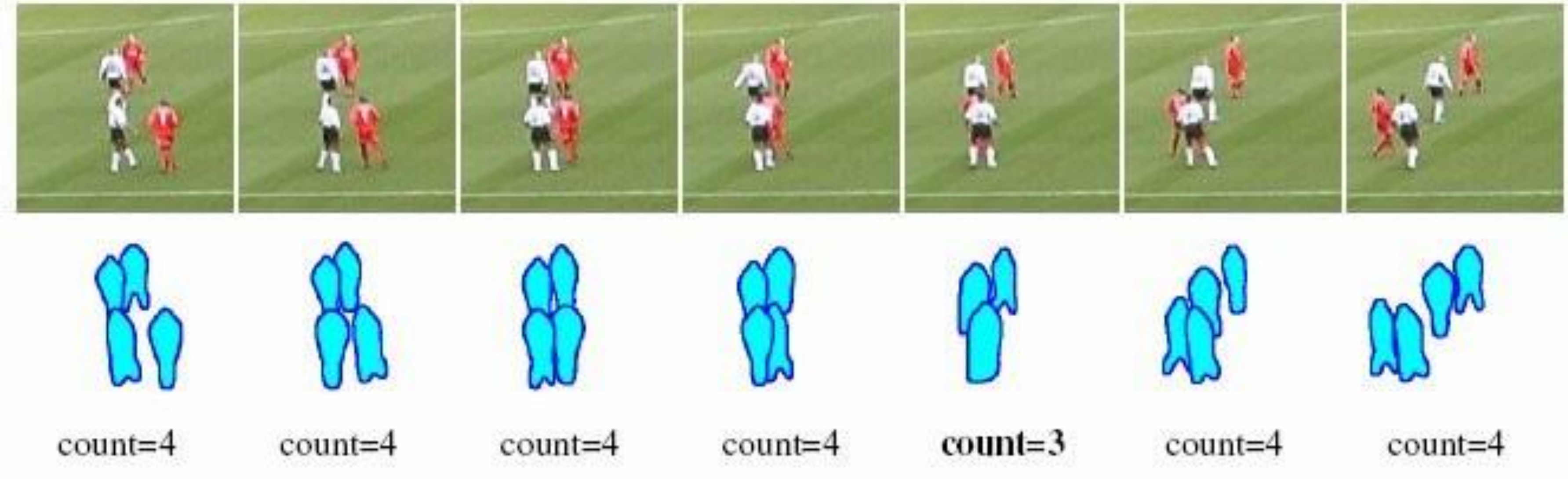
Evaluation run on every 10th frame

green contours: true positives
red boxes: false negatives
pink contours: false positives



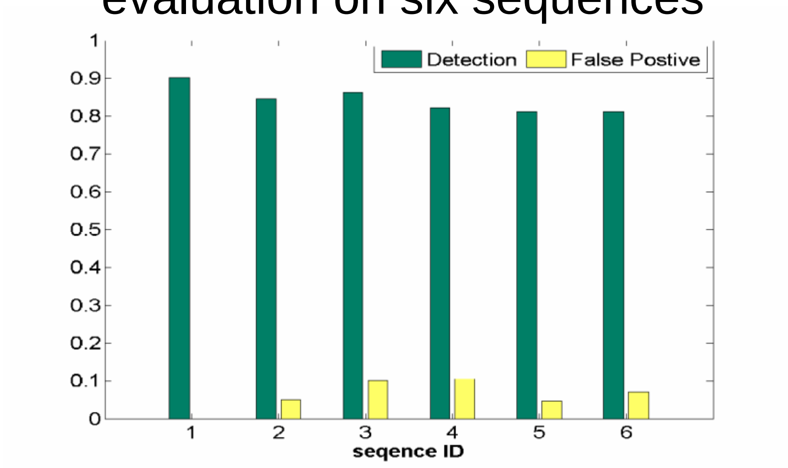
movie

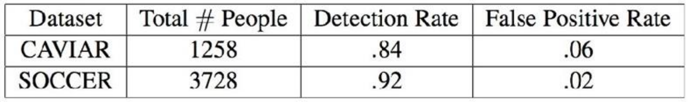
detailed view



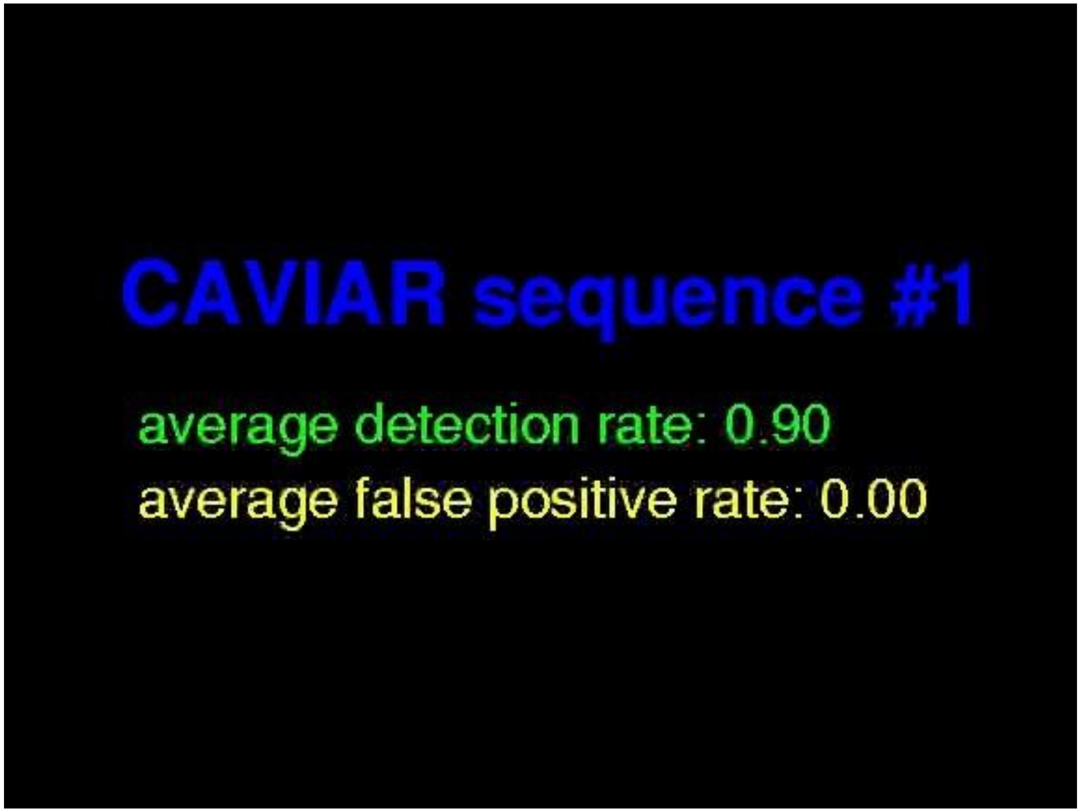
Caviar Dataset

evaluation on six sequences





Dataset	Total # People	Detection Rate	False Positive Rate
CAVIAR	1258	.84	.06
SOCCER	3728	.92	.02



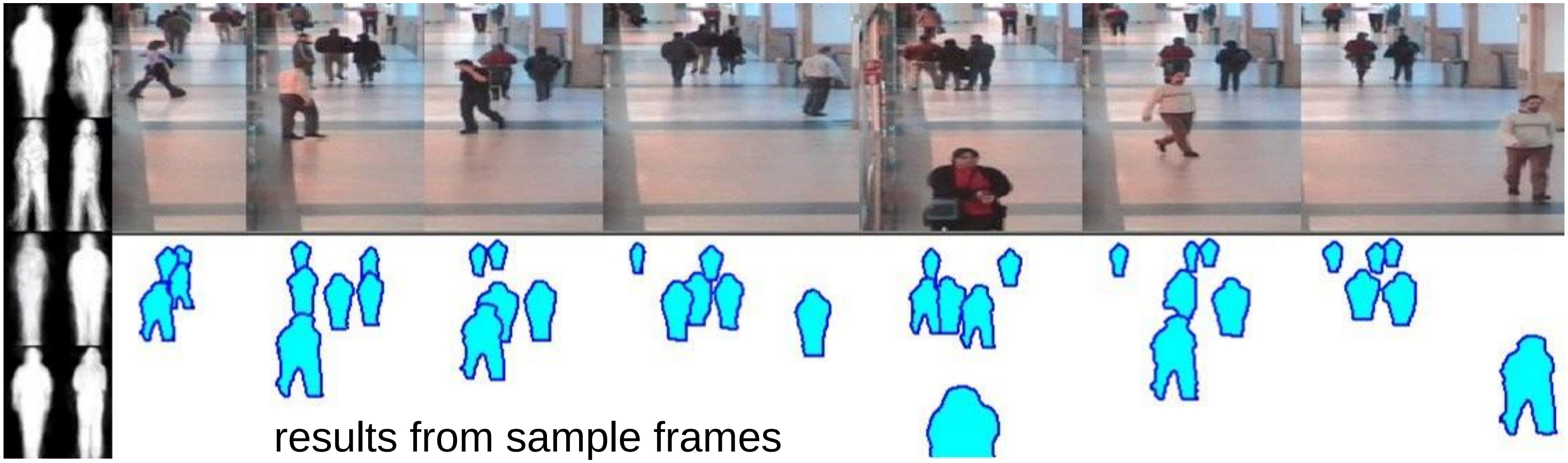
CAVIAR sequence #1

average detection rate: 0.90

average false positive rate: 0.00

UTS, Sean. He

Learned shapes



results from sample frames

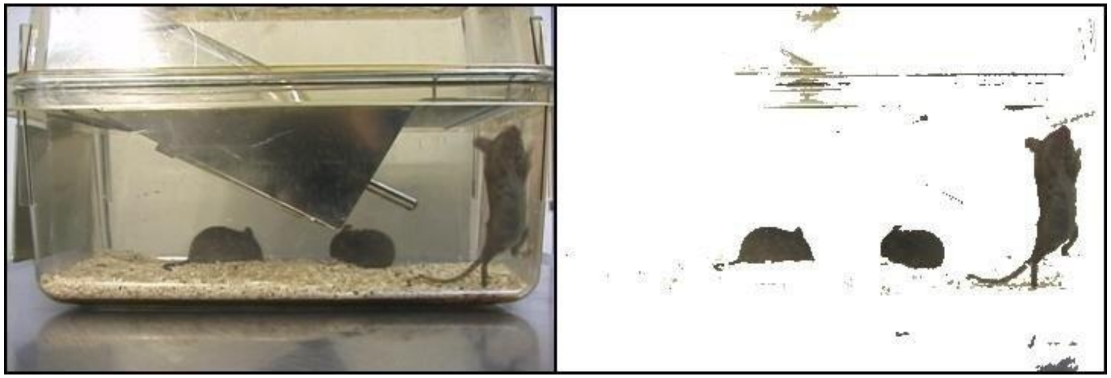
Contributions

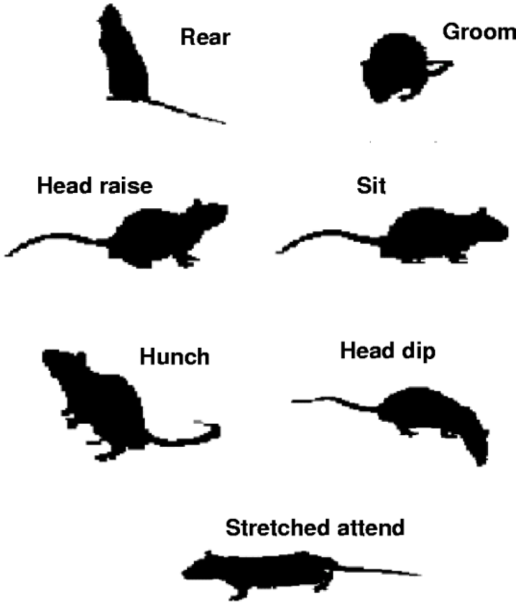
- We introduce a Marked Point Process framework for detecting people in crowds
- Conditional mark process models known correlations between bounding box size/orientation and image location
- Extrinsic and intrinsic shape models are learned automatically from training data
- Bayesian EM is used to automatically determine the number of components in the mixture of Bernoulli model for intrinsic shape

Research Directions

Idea: might be good to try other human shapes (activities)

or other animals/objects.




Rear
Groom
Head raise
Sit
Hunch
Head dip
Stretched attend