Massachusetts Institute of Technology

CSAIL

Improved Robustness and Efficiency for Automatic Visual Site Monitoring

Gerald Dalley
Thesis Defense

9 June 2009

Committee: Eric Grimson, Trevor Darrell, and Bill Freeman

Collaborators: Xiaogang Wang, Josh Migdal, Kinh Tieu, Lily Lee, Tomáš Ižo,…

Commercial and Transportation Applications

- Efficiency
  - What are the traffic bottlenecks?
  - How can we coordinate arrival schedules to minimize congestion?
- Marketing
  - How do in-store marketing campaigns effect behavior?
  - Are shoppers stopping at the sales booth?
- Loss prevention
  - How can we detect customer theft?
  - How can we detect employee theft?

Security Applications

- Threat detection
  - Unauthorized access
  - Violence
  - **Theft**
  - Tailing
  - **Loitering**
  - Sudden widespread panic

- **Recognition**
  - Is this person authorized?
  - Is this a "wanted" person?

- **Activity understanding**
  - ***What are the common traffic patterns?***
  - How can we deploy security resources more effectively?

Applications & Typical Scenes

Automatic Site Monitoring Pipeline

Detection

Tracking

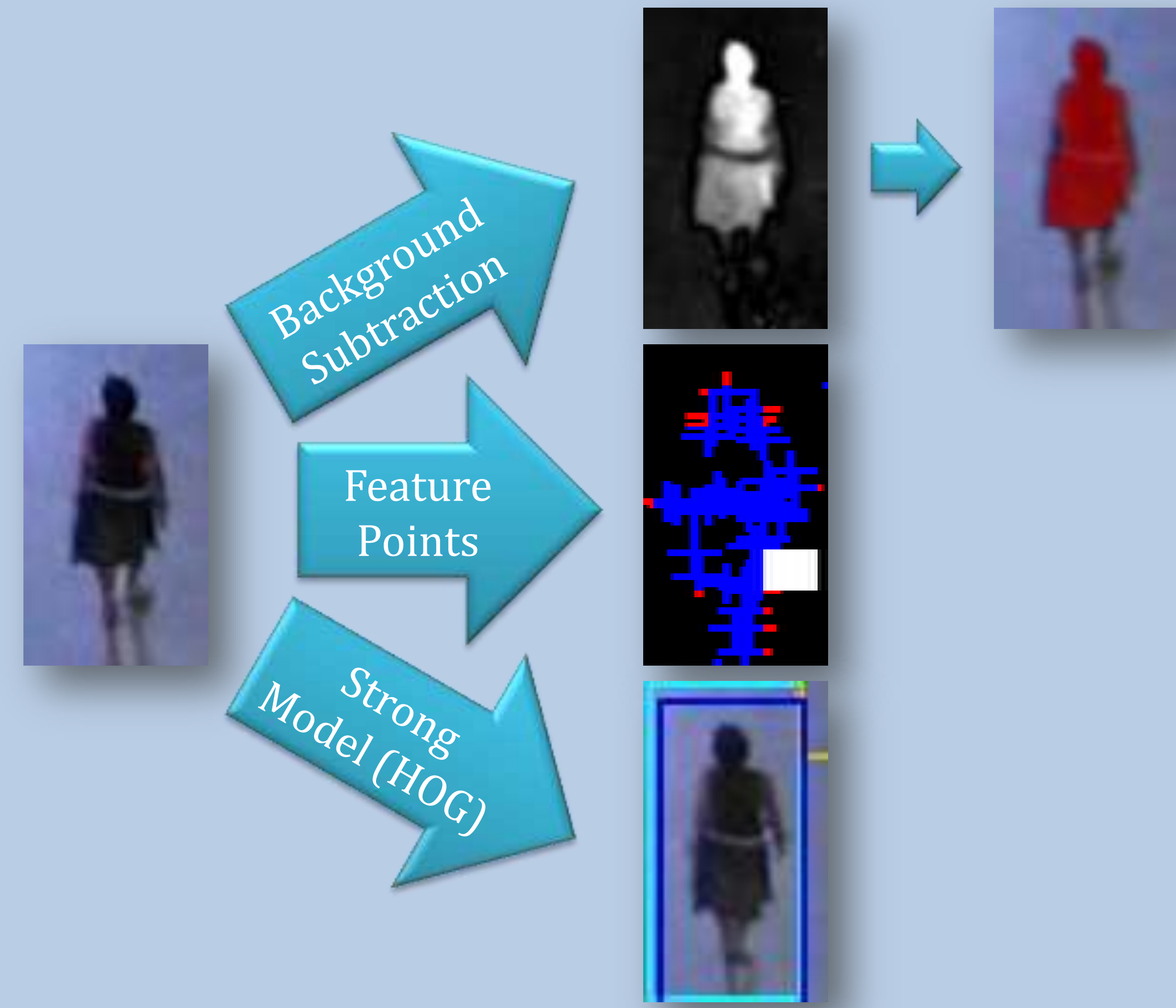
Analysis

Automatic Site Monitoring Pipeline

Detection

Tracking

Analysis



Automatic Site Monitoring Pipeline

Detection

Tracking

Analysis

- Background subtraction
  - Stauffer and Grimson, *CVPR* 1999.
  - Boykov, Veksler, and Zabih, *PAMI* 2001.
  - Mittal and Paragios, *CVPR* 2004.
  - Sheikh and Shah, *CVPR* 2005.
  - **Dalley, Migdal, and Grimson, *WACV* 2008.**
- Feature points
  - Shi and Tomasi, *CVPR* 1994.
- Strong models
  - Gavrila, *ECCV* 2000.
  - Leibe, Seeman, and Schiele, *CVPR* 2005.
  - Dalal and Triggs, *CVPR* 2005.
  - Zhu, Yeh, Cheng, and Avidan, *CVPR* 2006.
  - Wojek, Dorkó, Schulz, and Schiele, *DAGM* 2008.

Automatic Site Monitoring Pipeline

Detection

Tracking

Analysis

- **Kalman filter**
- Meanshift
- ...

Time windowing: for rendering purposes only

Automatic Site Monitoring Pipeline

Detection

Tracking

Analysis

- Identifying individual people
  - Phillips *et al*. *ICPR* 2002.
  - Sundaresan, Roy-Chowdhury, and Chellapa, *ICIP* 2003.
  - **Lee, Dalley, and Tieu, *ICCV* 2003.**
  - Veeraraghavan, Roy-Chowdhury, and Chellappa, *PAMI* 2005.
- Recognize events (loitering, theft, etc.)
  - Ivanonv and Bobick, *PAMI* 2000.
  - Vu, Bremond, and Thonnat, *ECAI* 2002.
  - PETS 2006 and PETS 2007 workshops (many papers)
    - **Dalley, Wang, and Grimson, *PETS* 2007**.
- ***Model flow patterns and site usage***
  - Stauffer, *CVPR* 1999.
  - Andrade, Blunsden, and Fisher, *ICPR* 2006.
  - Wang, Ma, and Grimson, *CVPR* 2007.
  - Wang *et al., CVPR* 2008.

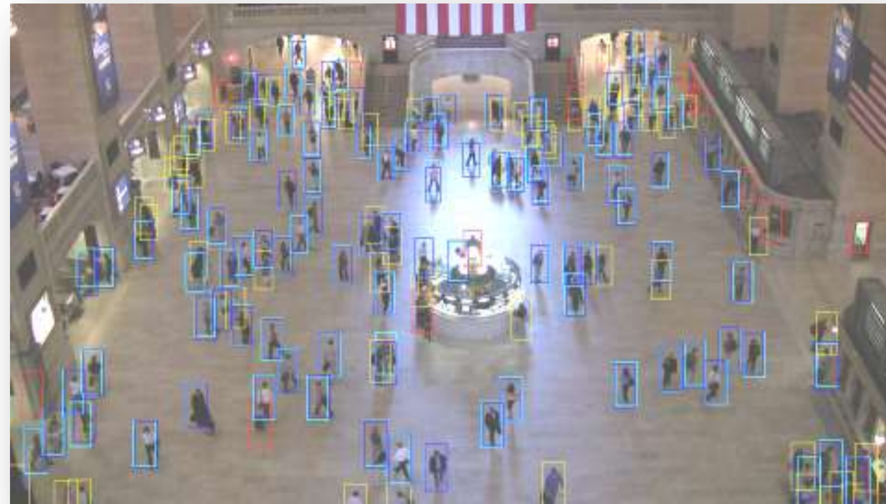
Thesis Contributions

- Background subtraction
  - Waving trees, rippling water *5.5% drop in false positive rate*
- **Large-scale monitoring**
  - **Clustering of path segments**
  - **Dalal and Triggs on a GPU** ***Up to 76x faster than CPU***
- Gait recognition
  - Model-based silhouettes *6%—44% boost in recognition rates*
- Event detection
  - Integrated detection and tracking *Only system to complete the PETS 2007 challenge*

This talk…

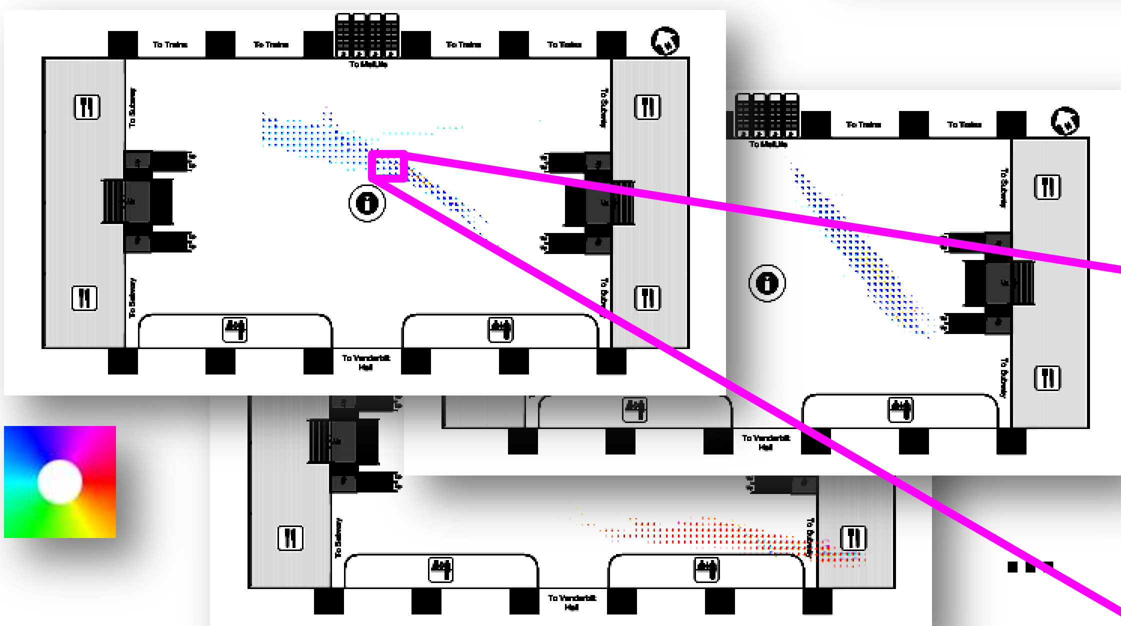
High-res Video

fast detection



Detections

good tracks



Site Activity Model

Outline

- Motivation

- Activity model overview
- Weak model detectors
- Strong model detector
- Data parallel implementation

- Summary

Outline

- Activity model overview
- Weak model detectors
- Strong model detector
- Data parallel implementation

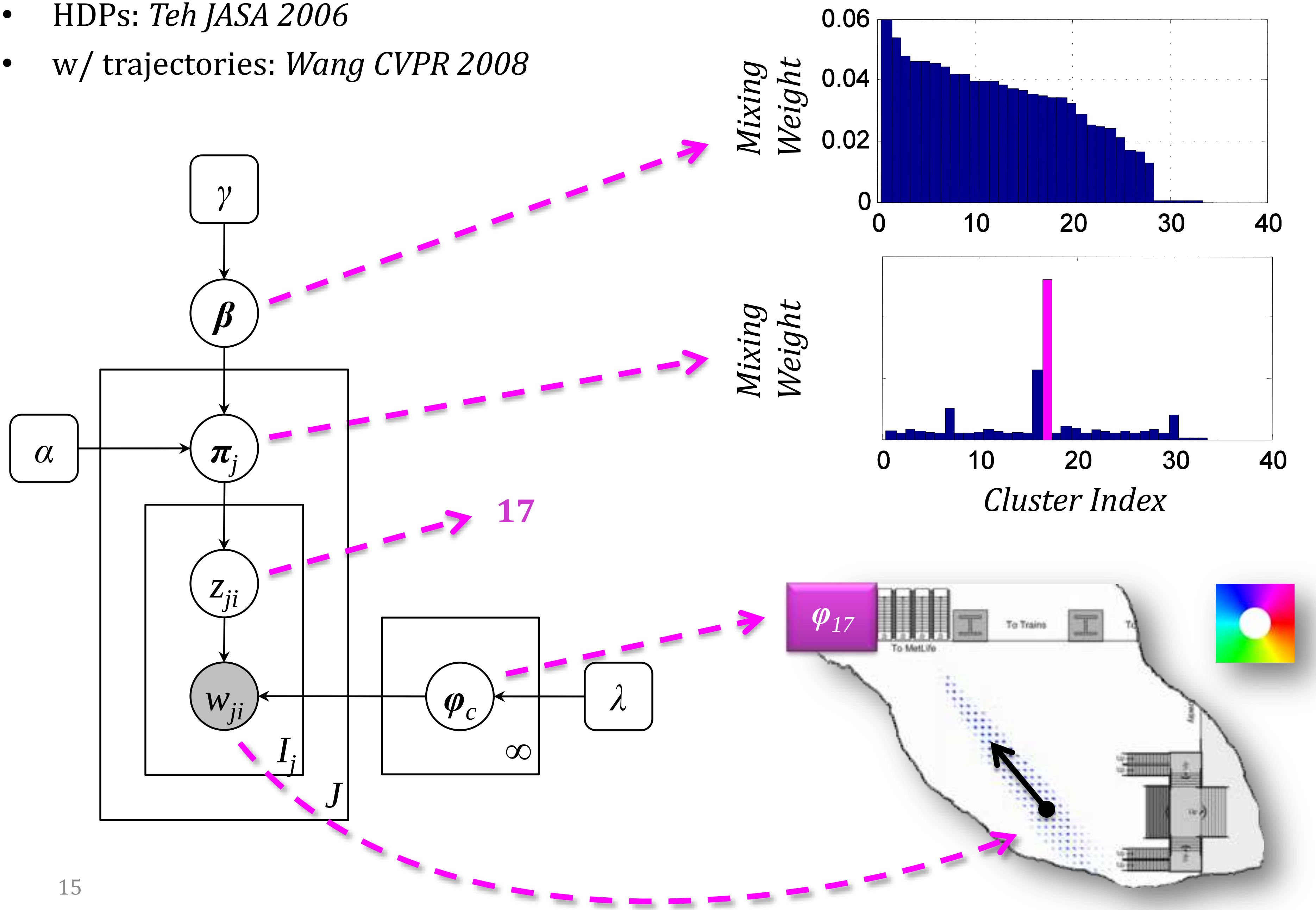
High Level

- Goal
  - Cluster trajectories to find common paths

- Approach
  - Infinite mixture model

Hierarchical Dirichlet Processes (HDPs)

- HDPs: *Teh JASA 2006*
- w/ trajectories: *Wang CVPR 2008*



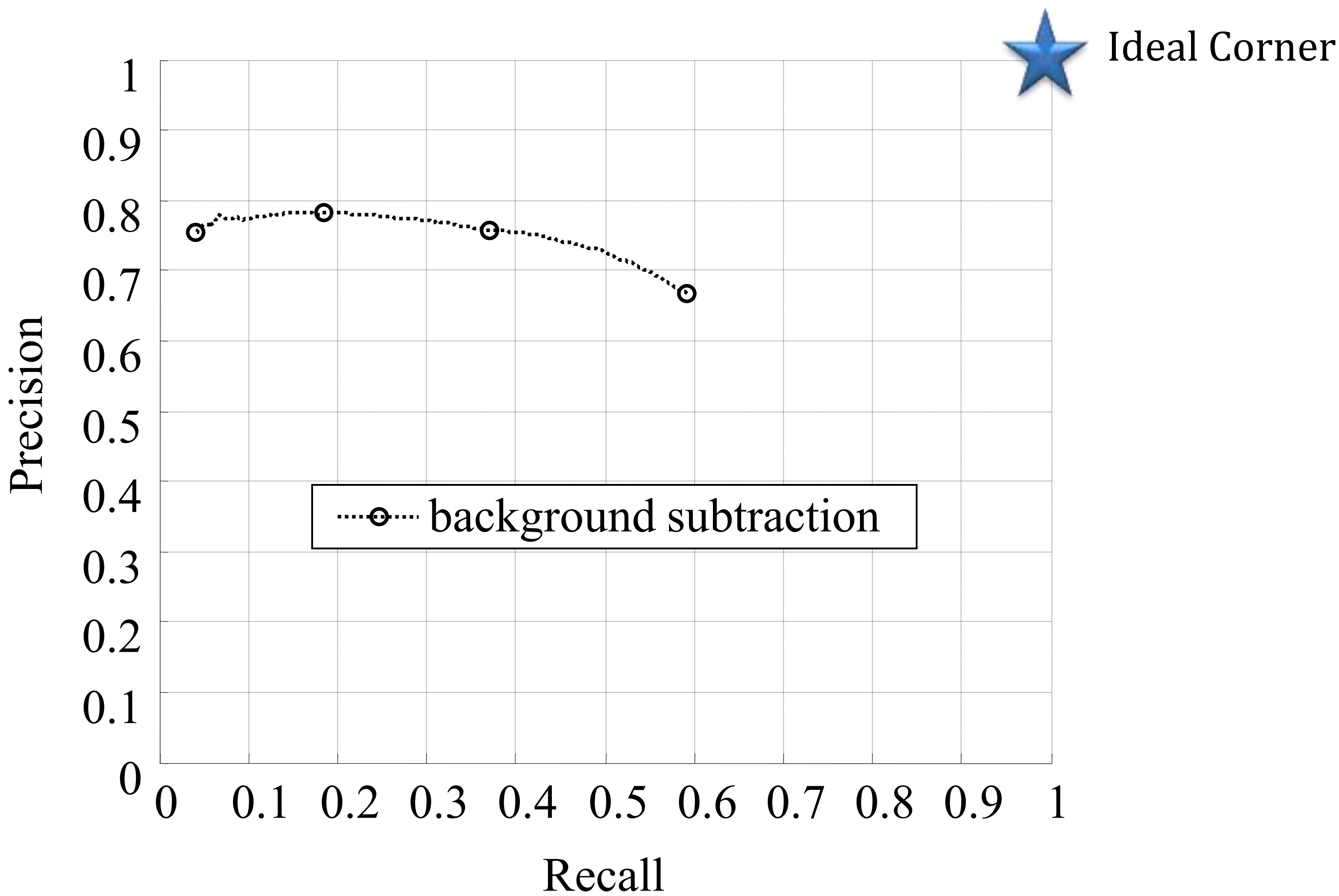
Outline

- Activity model overview
- Weak model detectors
  - Background subtraction
  - Feature point detection
- Strong model detector
- Data parallel implementation

Background Subtraction

Background Subtraction:

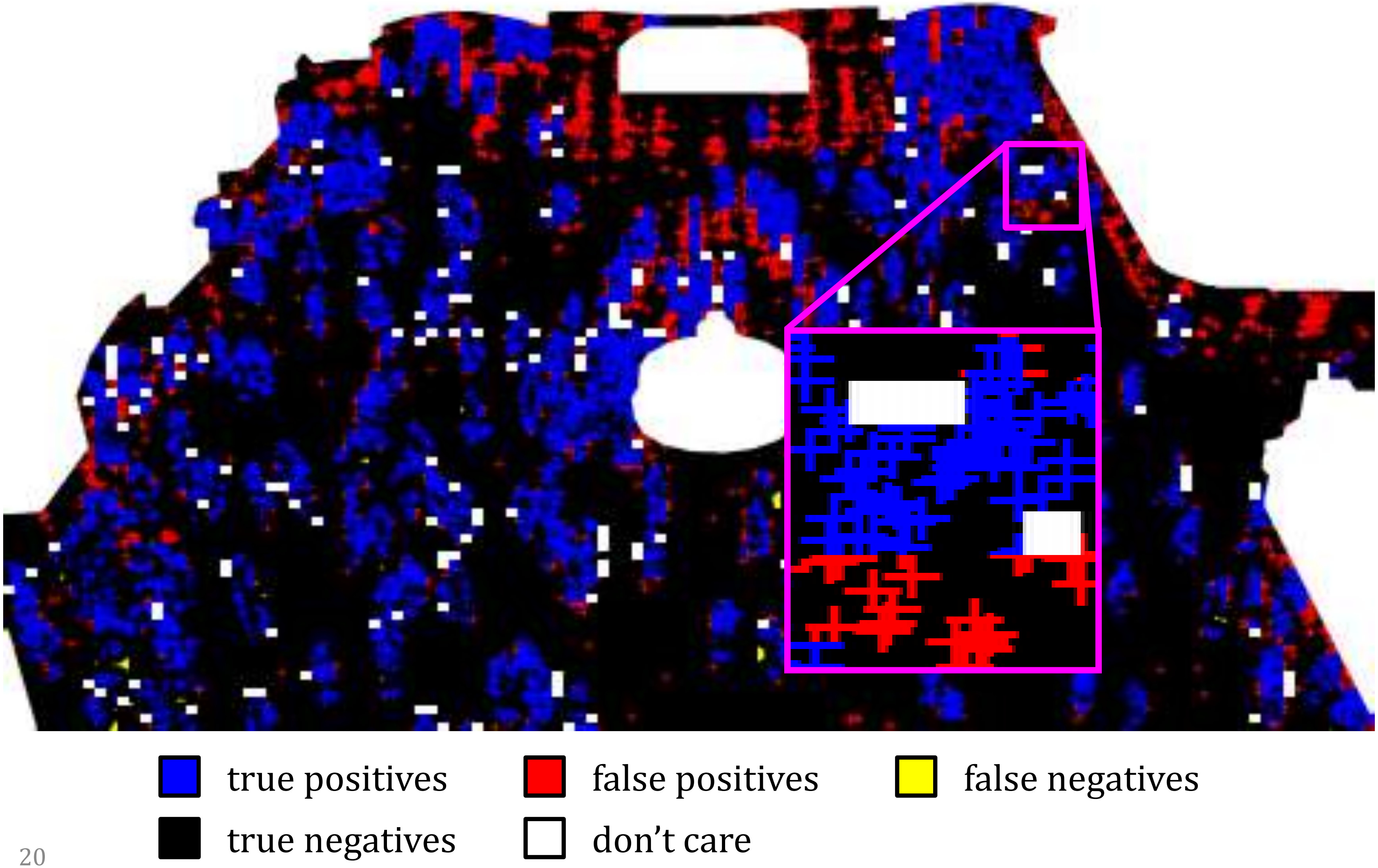
Precision-Recall



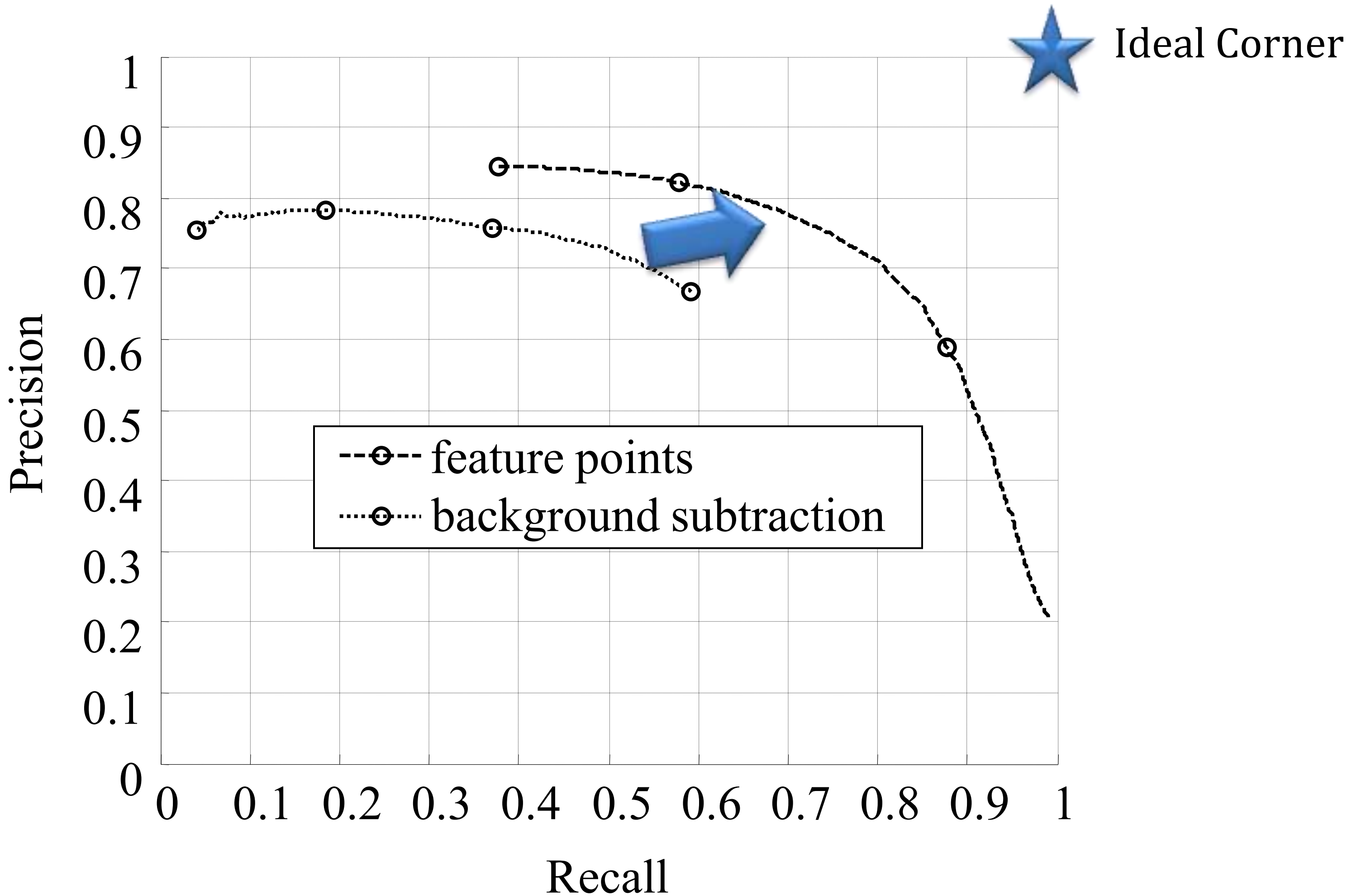
Background Subtraction:

Problems

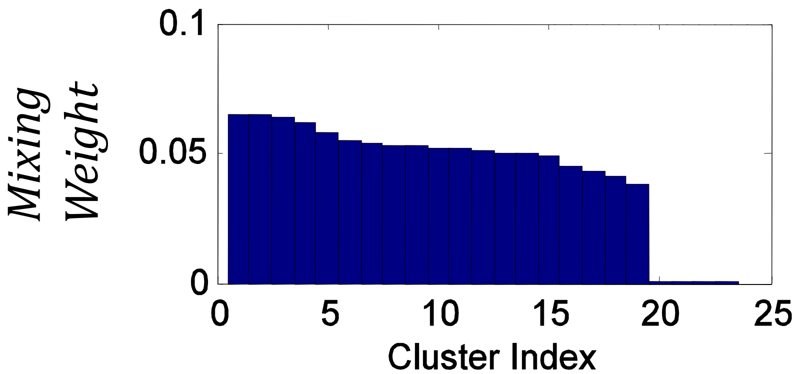
Alternative:
Shi & Tomasi Feature Point Detection



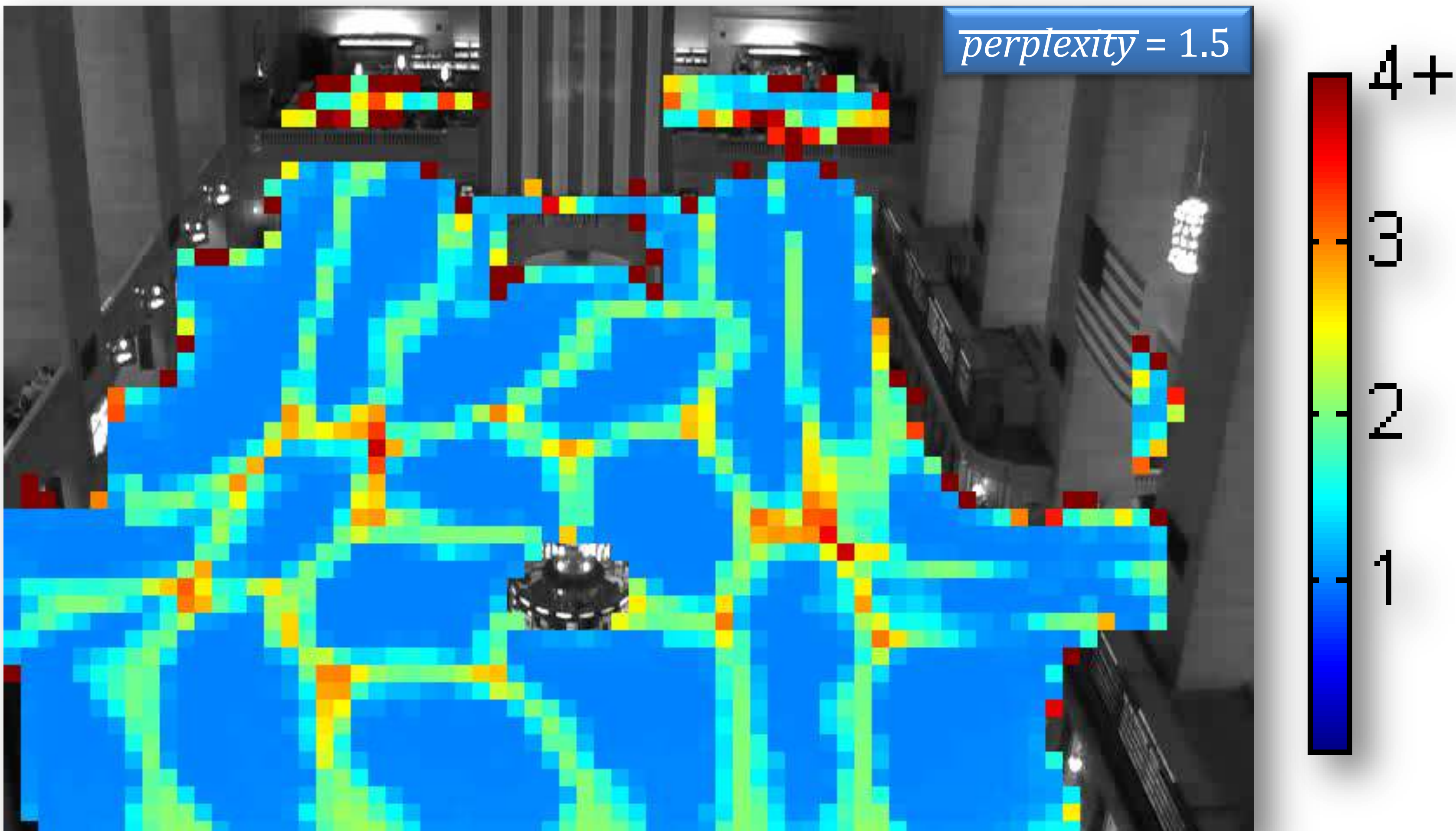
Improved Recall, but Low Precision



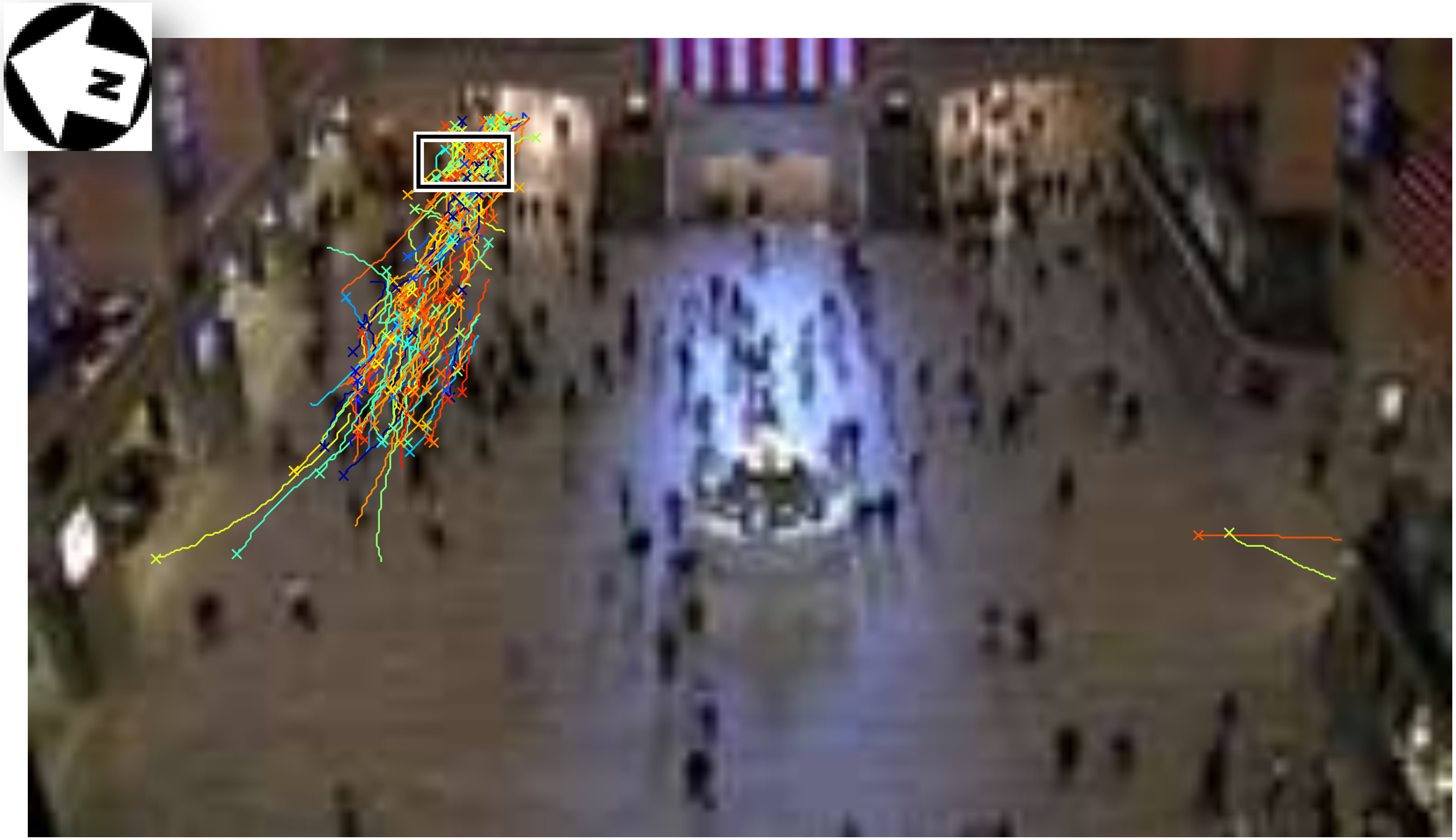
Clustering Feature Point Trajectories



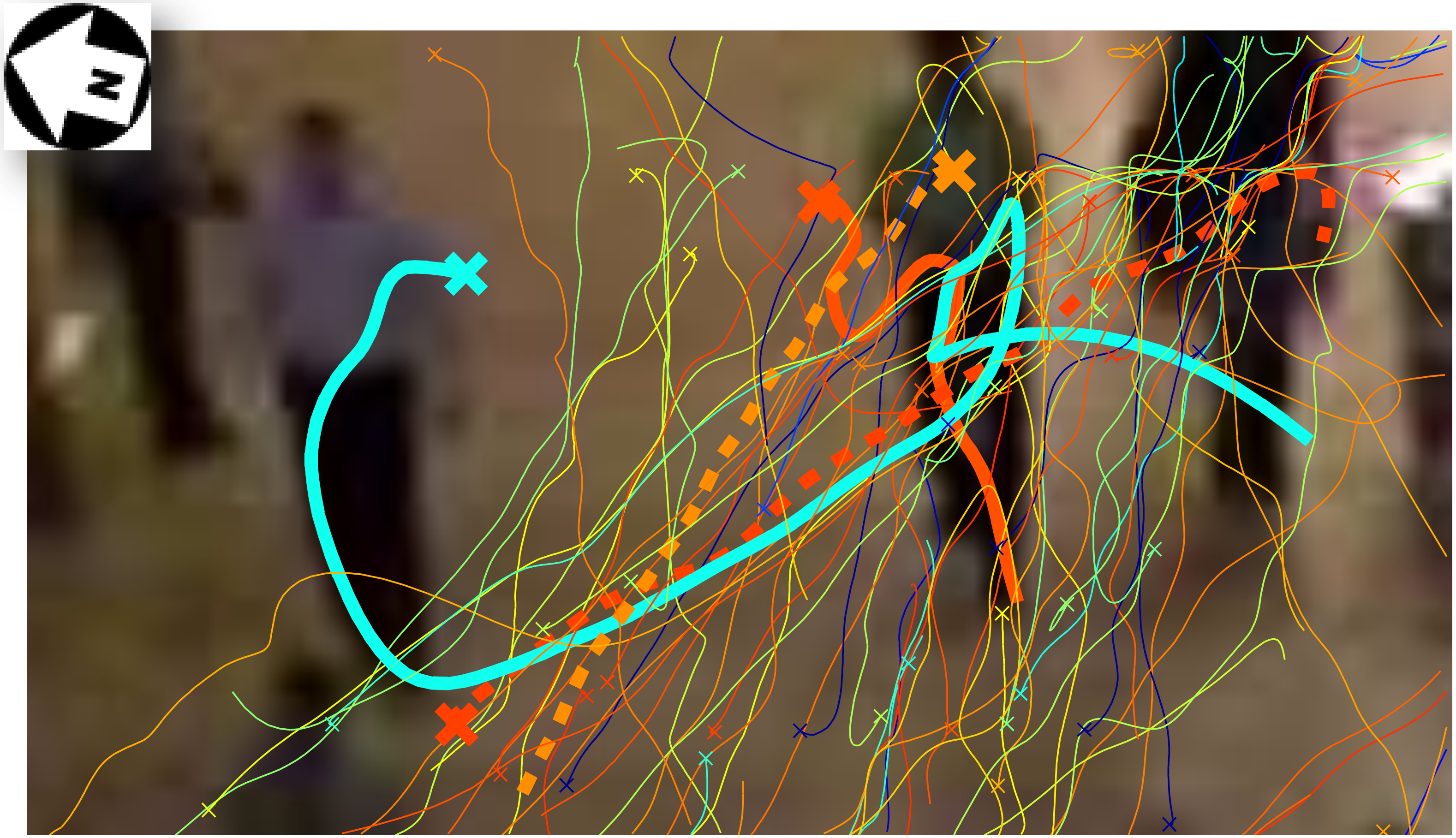
Perplexity
(cluster uncertainty given observed location)



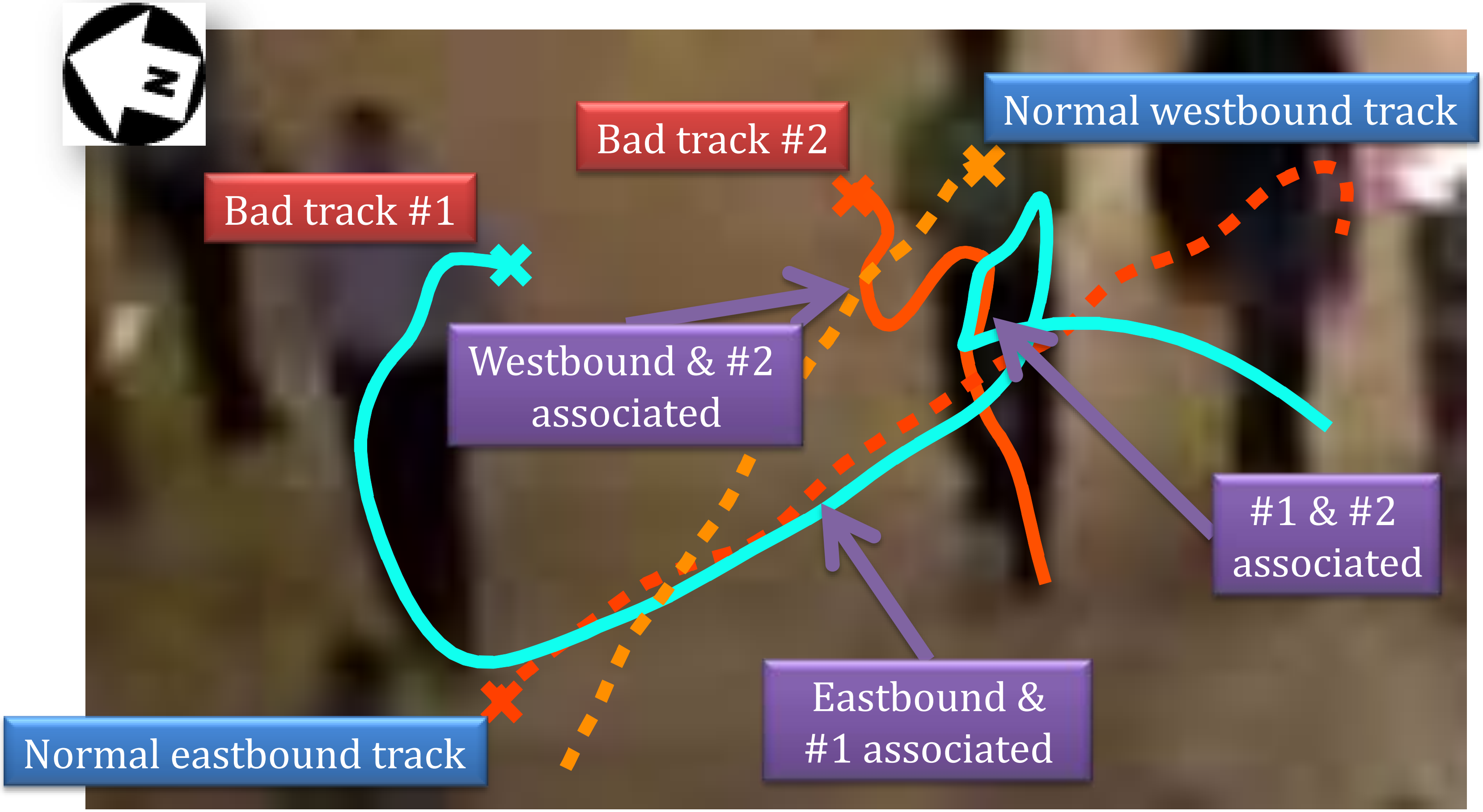
Crowded bidirectional traffic



Most Tracks Just Going East and West



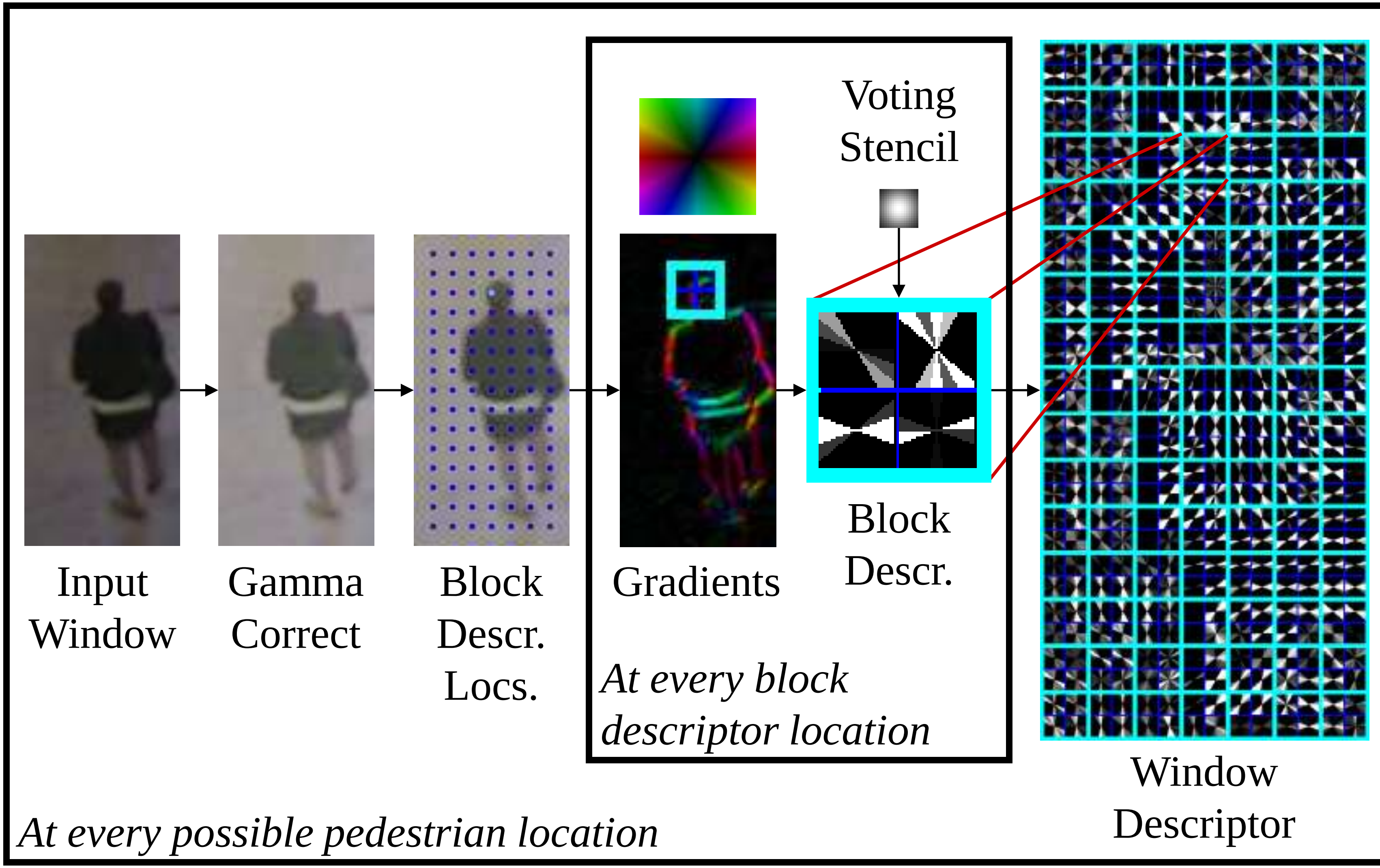
A Few Bad Tracks Couple East and West



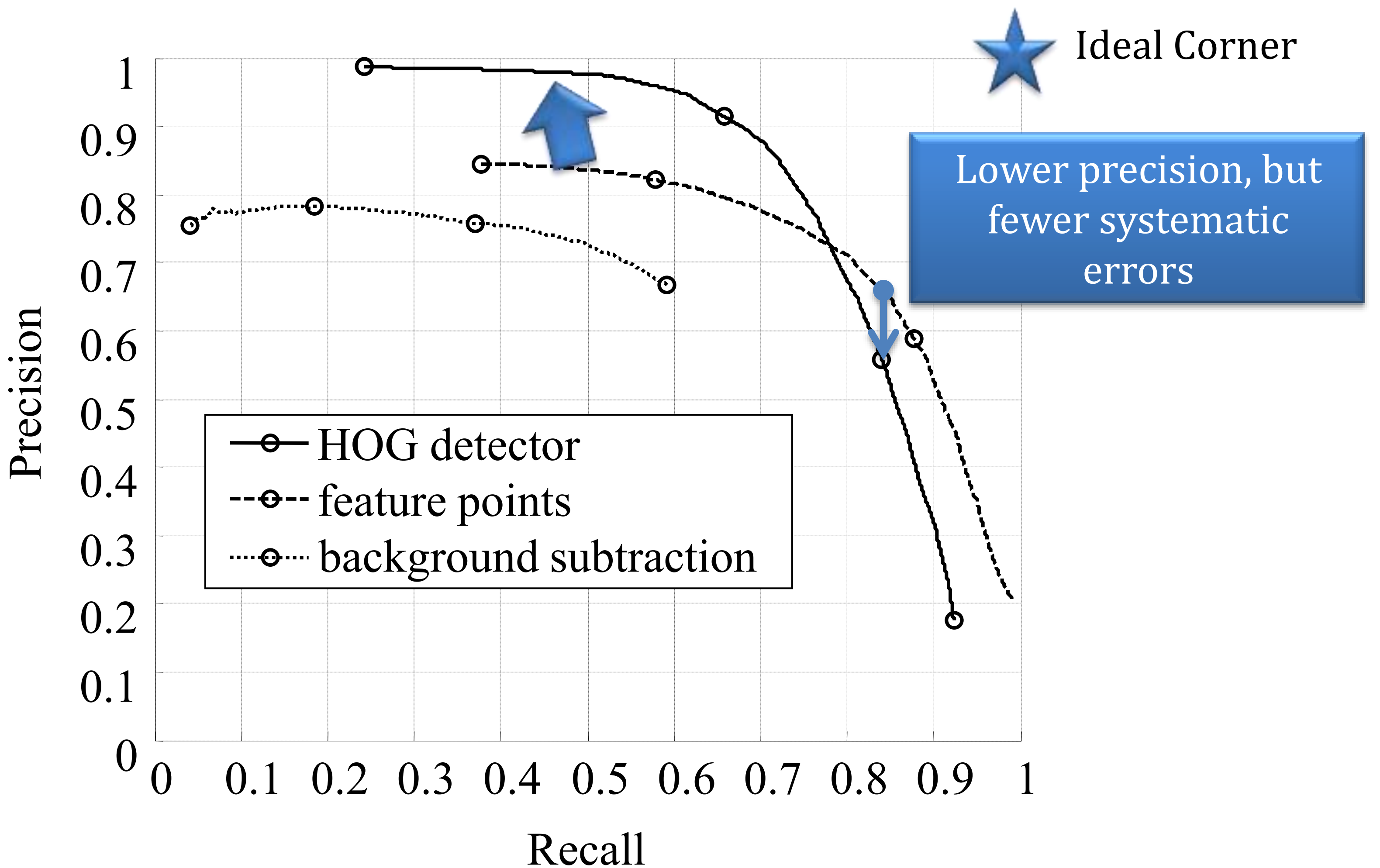
Outline

- Activity model overview
- Weak model detectors
- Strong model detector
  - Dalal and Triggs' HOG detector
  - Classification results
  - Activity modeling results
- Data parallel implementation

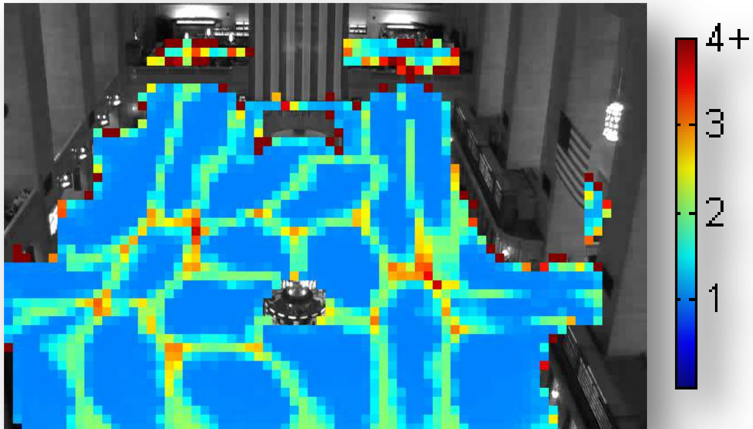
Dalal & Triggs HOG Features



Sufficient Precision and Recall



Better Perplexity



Point Tracking

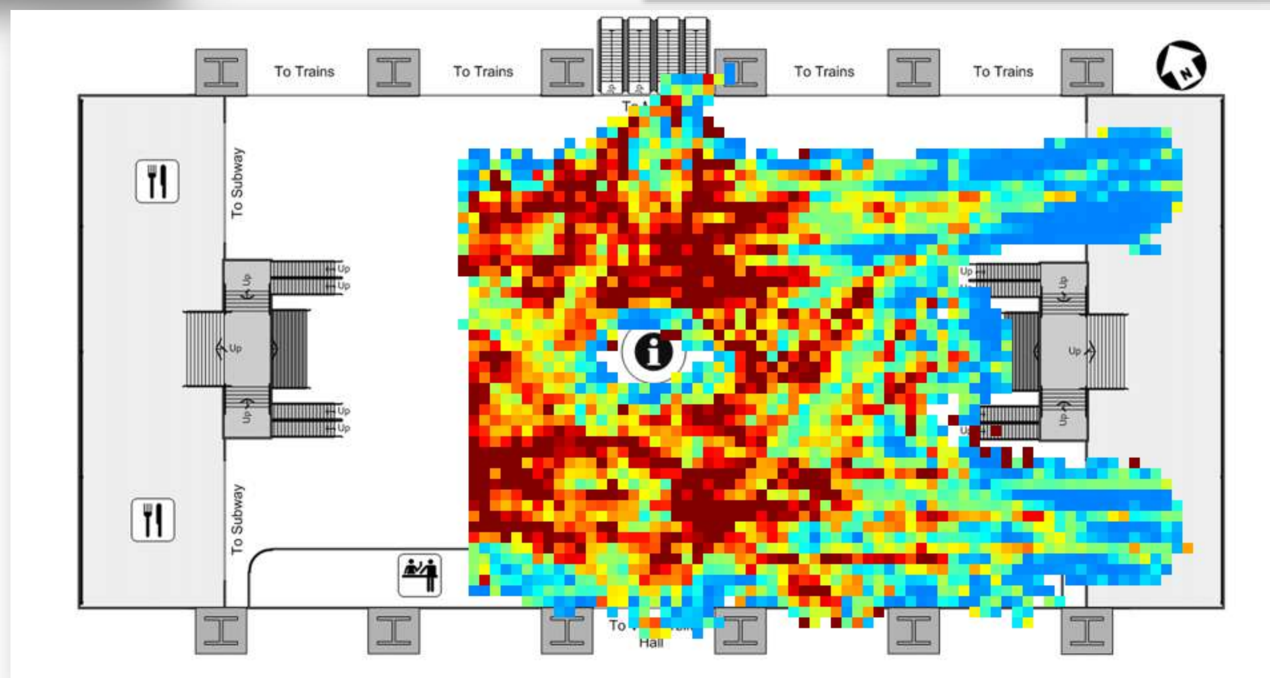
mean = 1.5

median = 1.1

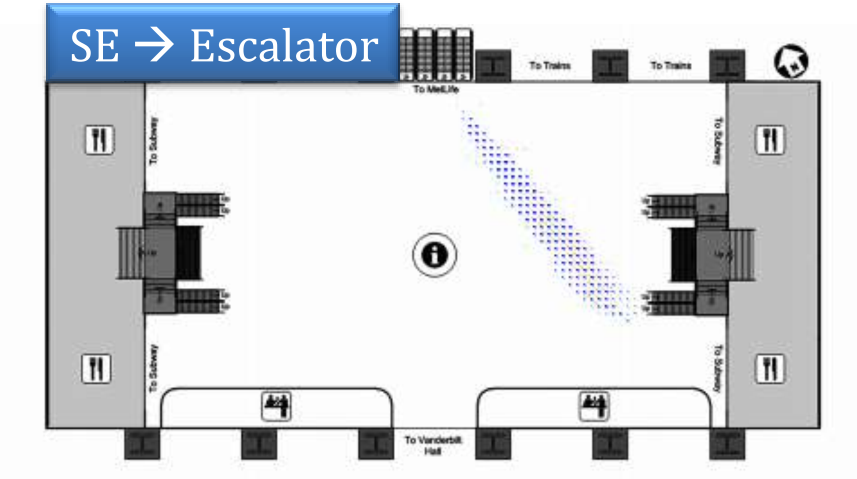
Pedestrian Detector Tracks

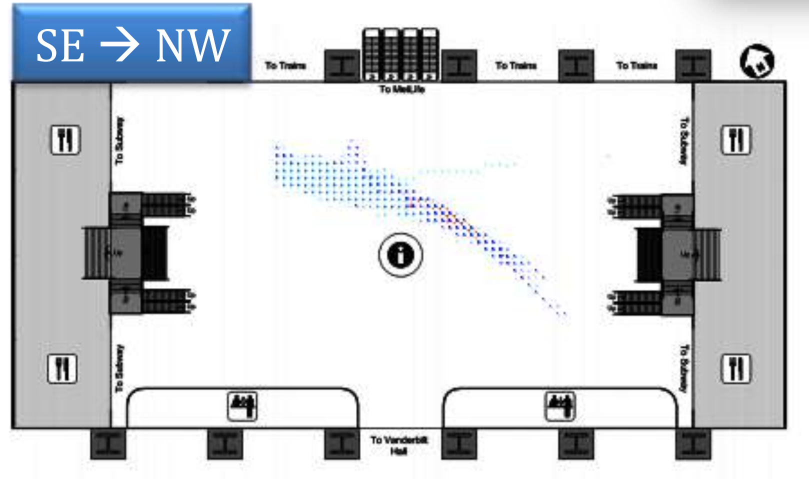
mean = 2.6

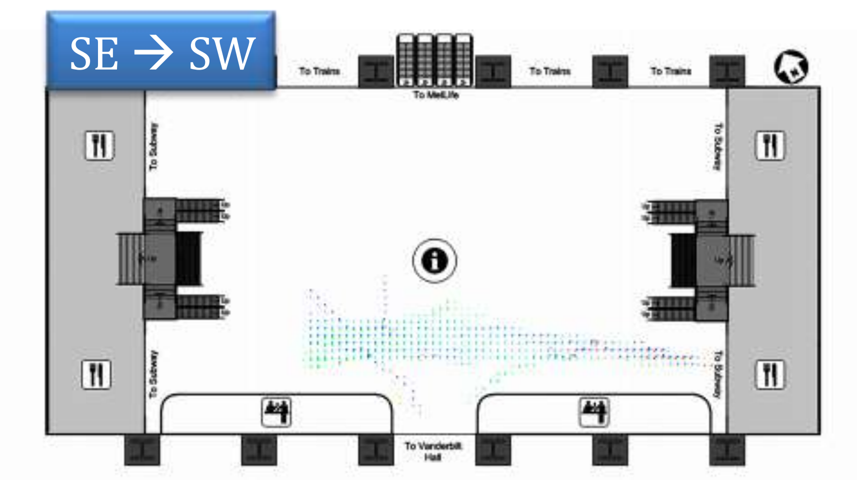
median = 2.4

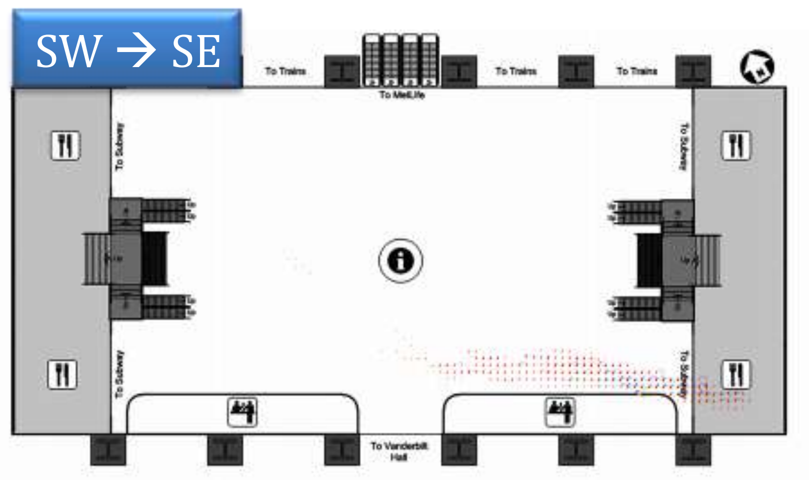


Selected Clusters



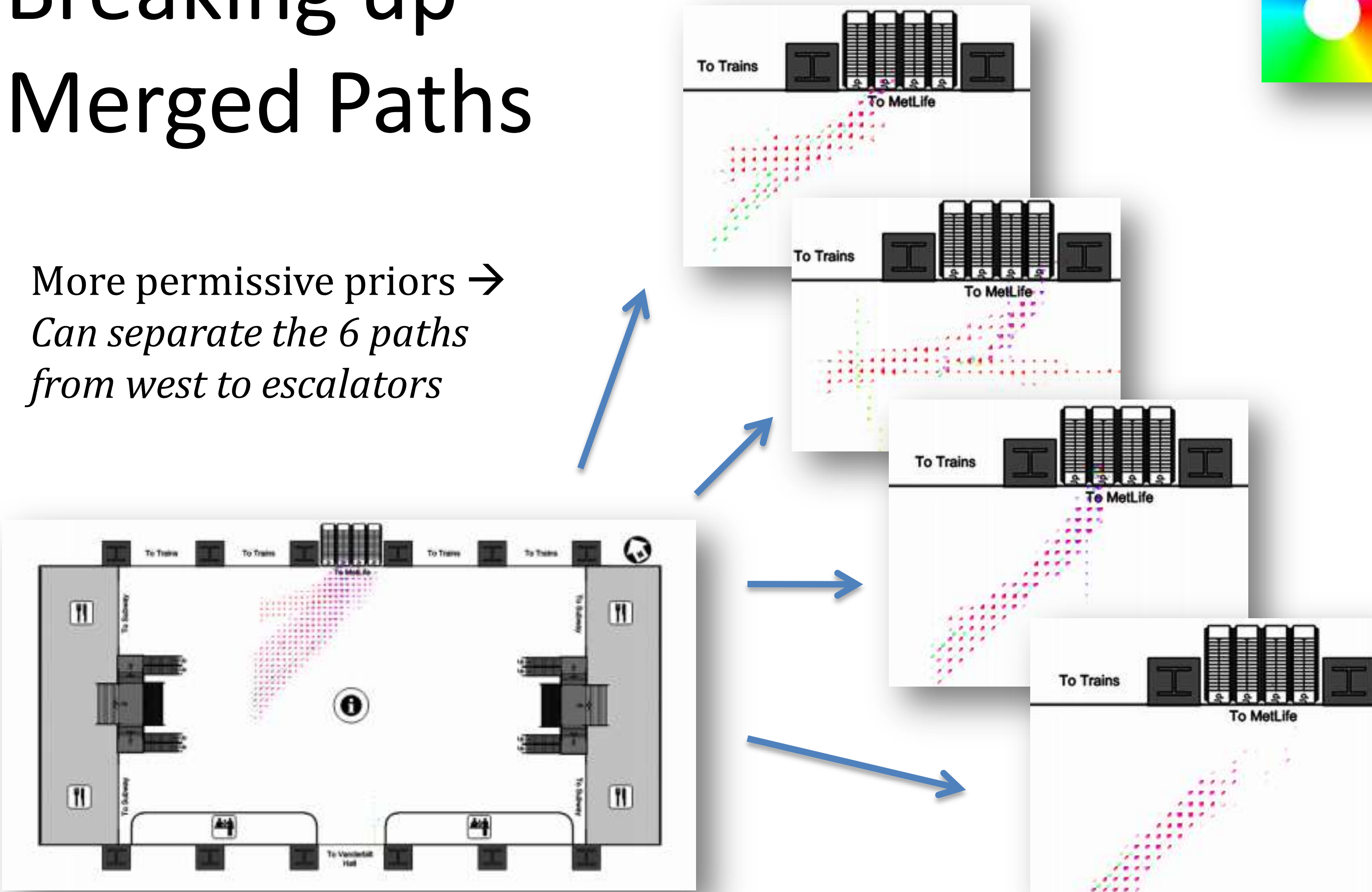






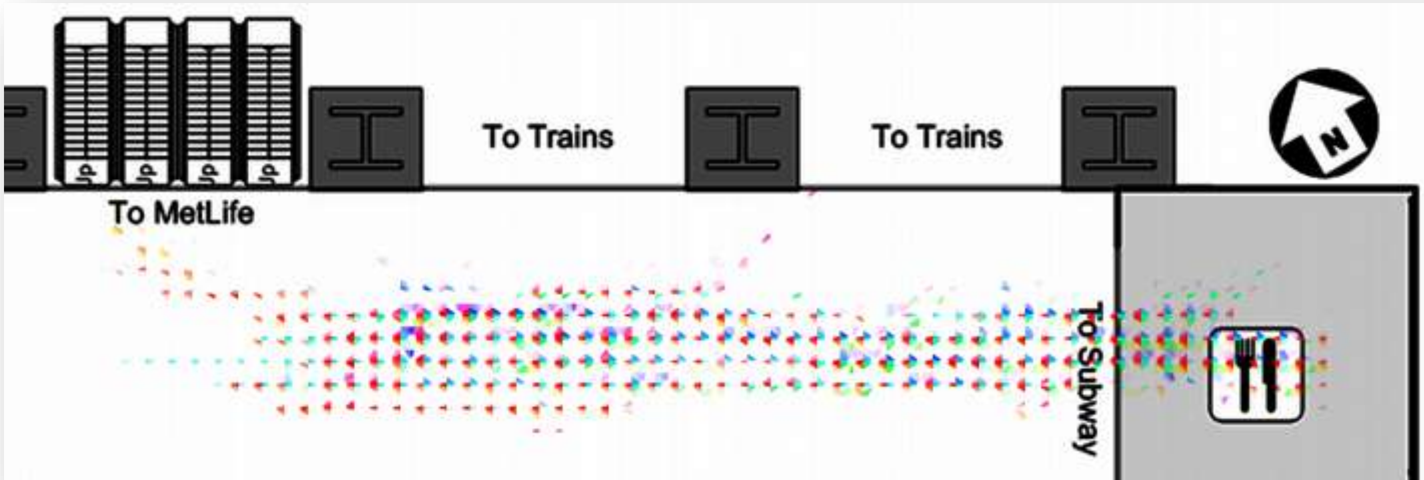
Breaking up Merged Paths

More permissive priors →
Can separate the 6 paths
from west to escalators

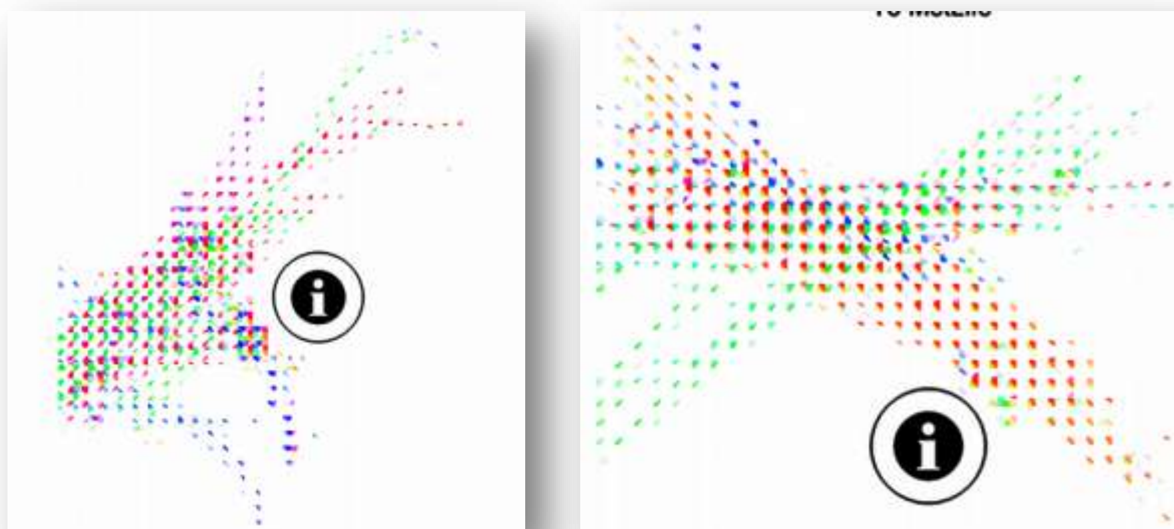


...

Some Directional Degeneracies Remain



Cause: *tracking errors*



Cause: *loitering and meandering*

Outline

- Activity model overview
- Weak model detectors
- Strong model detector
- Data parallel implementation
  - Motivation
  - GPU Intuition
  - Our design
  - Speedups

Good Results, but Too Slow

$$60\,\frac{\text{compute sec.}}{\text{frame}} \bullet 30\,\frac{\text{frames}}{\text{sec.}} \bullet 1\text{ hour of video} = 75\text{ compute days}$$

$$60\,\frac{\text{compute sec.}}{\text{frame}} \bullet 30\,\frac{\text{frames}}{\text{sec.}} \bullet \mathbf{40\ hours}\text{ of video} > \mathbf{8\ compute\ years}$$

...a little faster would be nice.

Our data:
- 40 hours
- 1920×1080 frames
    - 6.75× the pixels/frame w.r.t. 640×480
    - 27× the pixels/frame w.r.t. 320×240
- progressive scan

CPU Characteristics

- One thing fast
  - High clock speed
  - Pipelining
- Complex control flow
  - Cache
  - Branch prediction
  - Speculation
  - …

- Task parallel:
  a few *different* things fast
  - Multicore
  - Hyperthreading
  - Sophisticated caches
- Data parallel:
  Same instruction on a few data items
  - MMX, SSE, etc.

GPU Characteristics

- *Same* instruction, many data items
  - 240 “cores” or more
- Very high memory *bandwidth*
  - 10× a CPU’s

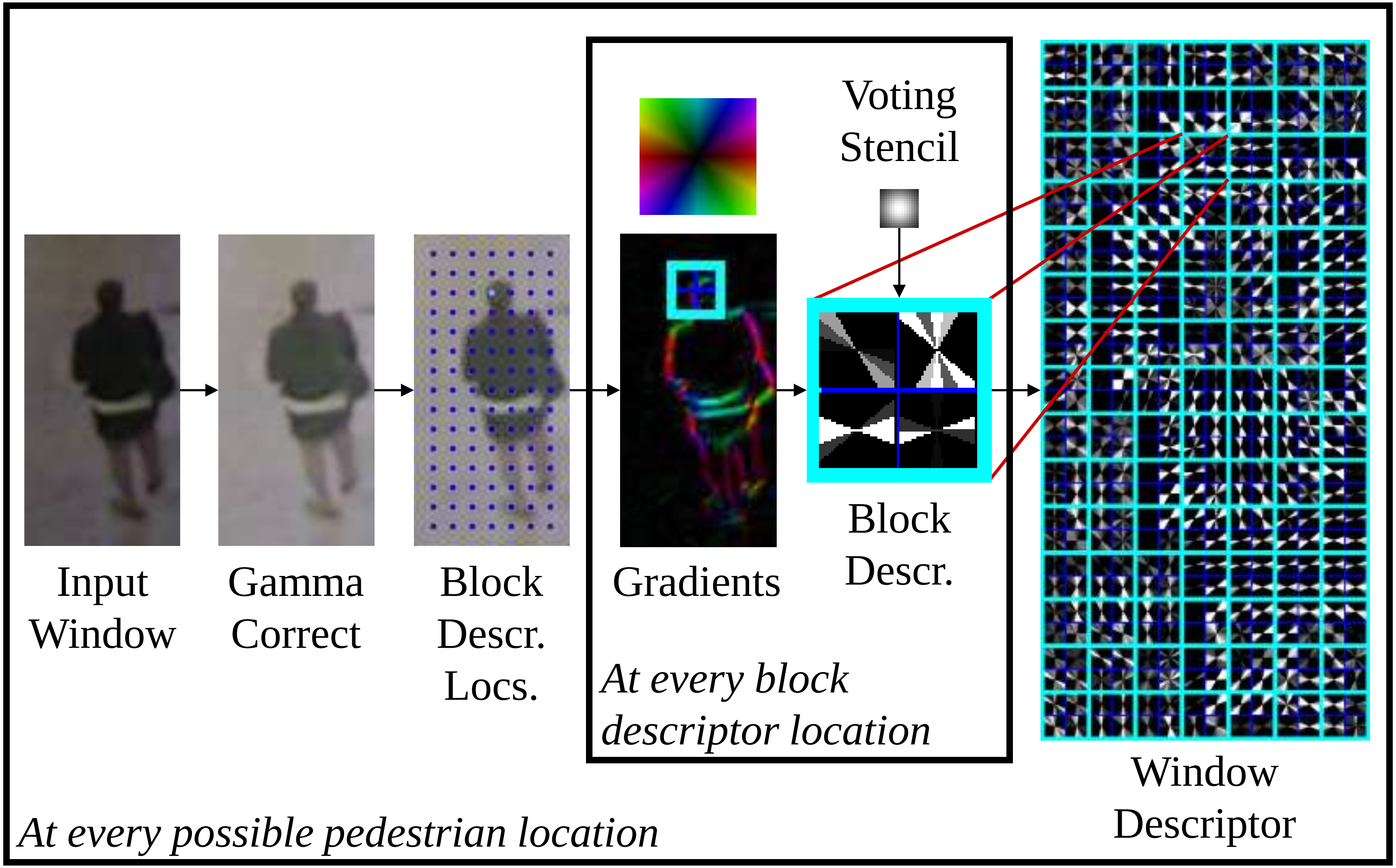
- Typical speedups:
  - 10×—100×
- Programming
  - Style: C/C++
  - Optimization effort ≈ C++ & assembly mix
- Slow if…
  - Insufficiently parallel code
  - Random memory access
  - Branching

Intuition: What Works Well on a GPU

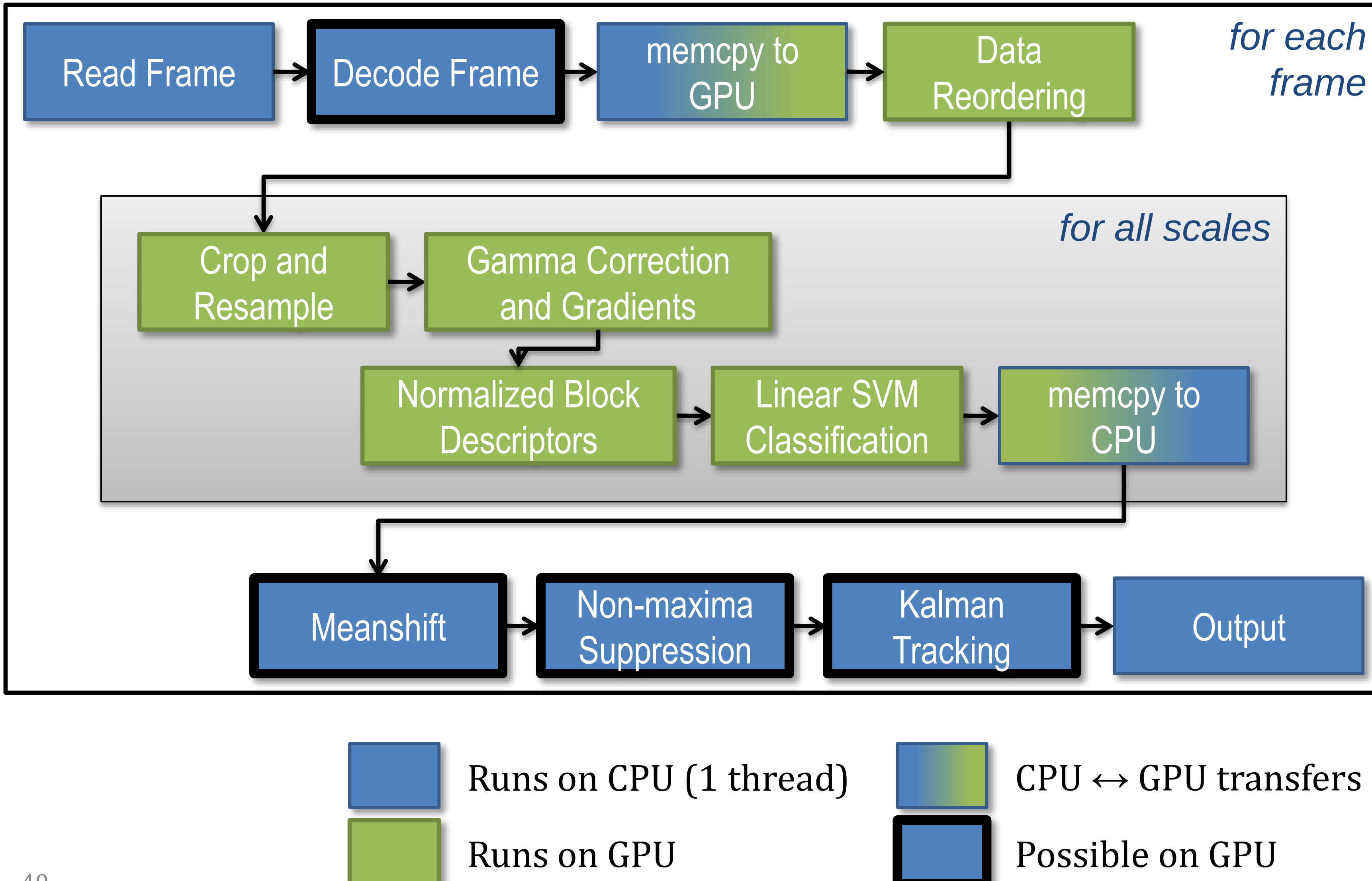
- In general
  - $10^{\text{<<<MANY>>>}}$ independent inputs and/or outputs
  - Localized memory access

- Typical applications
  - Filterbanks
  - Sliding window algorithms
  - Code that's easy to vectorize in Matlab

Dalal & Triggs HOG Features



Our CUDA Pipeline



CPU vs. GPU Times:

Results from a Simplified Profiling Application

Processing Step	CPU Implementation	GPU Implementation	GPU Speedup
Read input (CPU)	0%	**17%**	
GPU resizer setup		*5%*	
Resize	4%	11%	24.3×
Gradients	**24%**	9%	**164.0×**
Normalized block descriptors	**57%**	**35%**	**97.7×**
Window classification	14%	8%	**100.6×**
Cleanup	0%	*12%*	0.5×
Detection (CPU)	0%	4%	1.1×
TOTAL	23 seconds	0.4 seconds	58.8×

GPU Speedup Results

- Our Implementation
  - 58.8× to 76× speedup (vs. optimized CPU-only)
  - Current bottlenecks
    - Video decoding on the CPU (17%)
    - Block descriptors (35%)
    - Bookkeeping & memory transfers (17%)
- Wojek, Dorkó, Schulz, Schiele [DAGM 2008]
  - 30× speedup
  - Optimized for the previous GPU architecture
  - Less efficient usage of memory bandwidth

Summary

- Fast HOG implementation
  - 58.8× to 76× speedup
- Better clustering of trajectory flows
  - Qualitative improvements
  - Perplexity

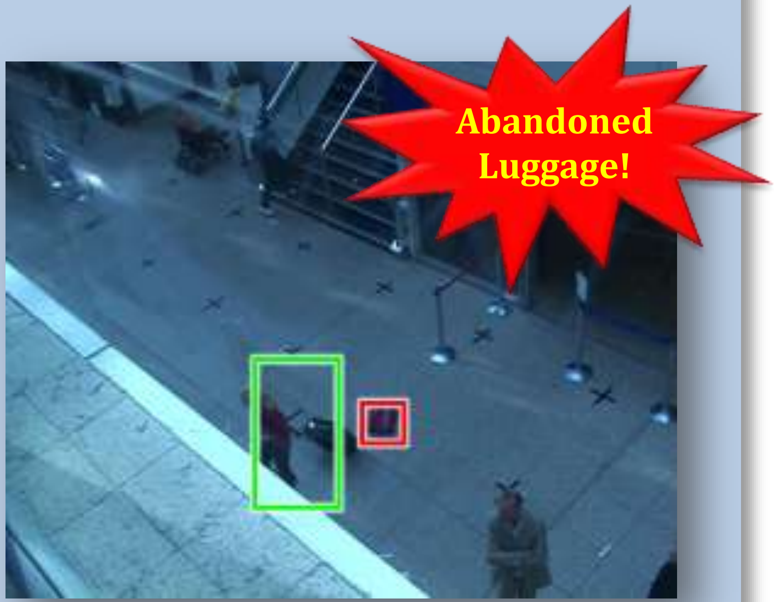
Future Work

- Scale to true HD real-time
  - Multithreaded CPU
  - Multiple GPUs
  - Asynchronous data transfers
  - More computation to GPUs
- Better HOG training
  - Explicit occlusion handling
  - Add video features (a la Dalal and Triggs 2006)
- Alternative detectors
  - Boosted cascade *on GPUs* (CPU: Avidan; Viola & Jones)
- Activity modeling
  - Learn long-term flow trends
  - Temporal dependencies (via HMMs)
- Integrate with other technologies in this thesis…

Other Potential Applications for Fast and Robust Pedestrian Detection

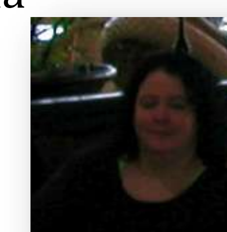
Appearance models for recognition

Multimodal tracking for event detection



Acknowledgments

- Thesis Committee
  - Eric Grimson
  - Bill Freeman
  - Trevor Darrell
- Too many friends and fellow students here at MIT to list individually…
- Collaborators
  - Xiaogang Wang
  - Josh Migdal
  - Kinh Tieu
  - Lily Lee
  - Tomáš Ižo
  - Jim Sukha, Krista Ehinger, and Geza Kovacs
- Funders
  - DARPA
  - MIT
  - Shell
  - Singapore
  - …
- Work Experiences
  - Microsoft Research
  - MERL
  - BAE Systems
  - D.E. Shaw
- My wife, Dianna



ped

Automatic Site Monitoring Pipeline

Detection

Tracking

Analysis

- Background subtraction
  - Stauffer and Grimson. Adaptive Background Mixture Models for Real-time Tracking. *CVPR*. 1999.
  - Boykov, Veksler, and Zabih. Fast Approximate Energy Minimization via Graph Cuts. *PAMI*. 2001.
  - Mittal and Paragios. Motion-based Background Subtraction using Adaptive Kernel Density Estimation. *CVPR*. 2004.
  - Migdal and Grimson. Background Subtraction using Markov Thresholds. *MVC*. 2005.
  - Sheikh and Shah. Bayesian Object Detection in Dynamic Scenes. *CVPR*. 2005.
  - **Dalley, Migdal, and Grimson. Background Subtraction for Temporally Irregular Dynamic Textures. *WACV*. 2008.**

- Feature points
  - Shi and Tomasi. Good Features to Track. *CVPR*. 1994.

- Strong models
  - Gavrila. Pedestrian Detection from a Moving Vehicle. *ECCV*. 2000.
  - Leibe, Seeman, and Schiele. Pedestrian Detection in Crowded Scenes. *CVPR*. 2005.
  - Dalal and Triggs. Histograms of Oriented Gradients for Human Detection. *CVPR*. 2005.
  - Zhu, Yeh, Cheng, and Avidan. Fast Human Detection using a Cascade of Histograms of Oriented Gradients. *CVPR*. 2006.
  - Wojek, Dorkó, Schulz, and Schiele. Sliding-windows for Rapid Object Class Localization: A Parallel Technique. *DAGM*. 2008.

Automatic Site Monitoring Pipeline

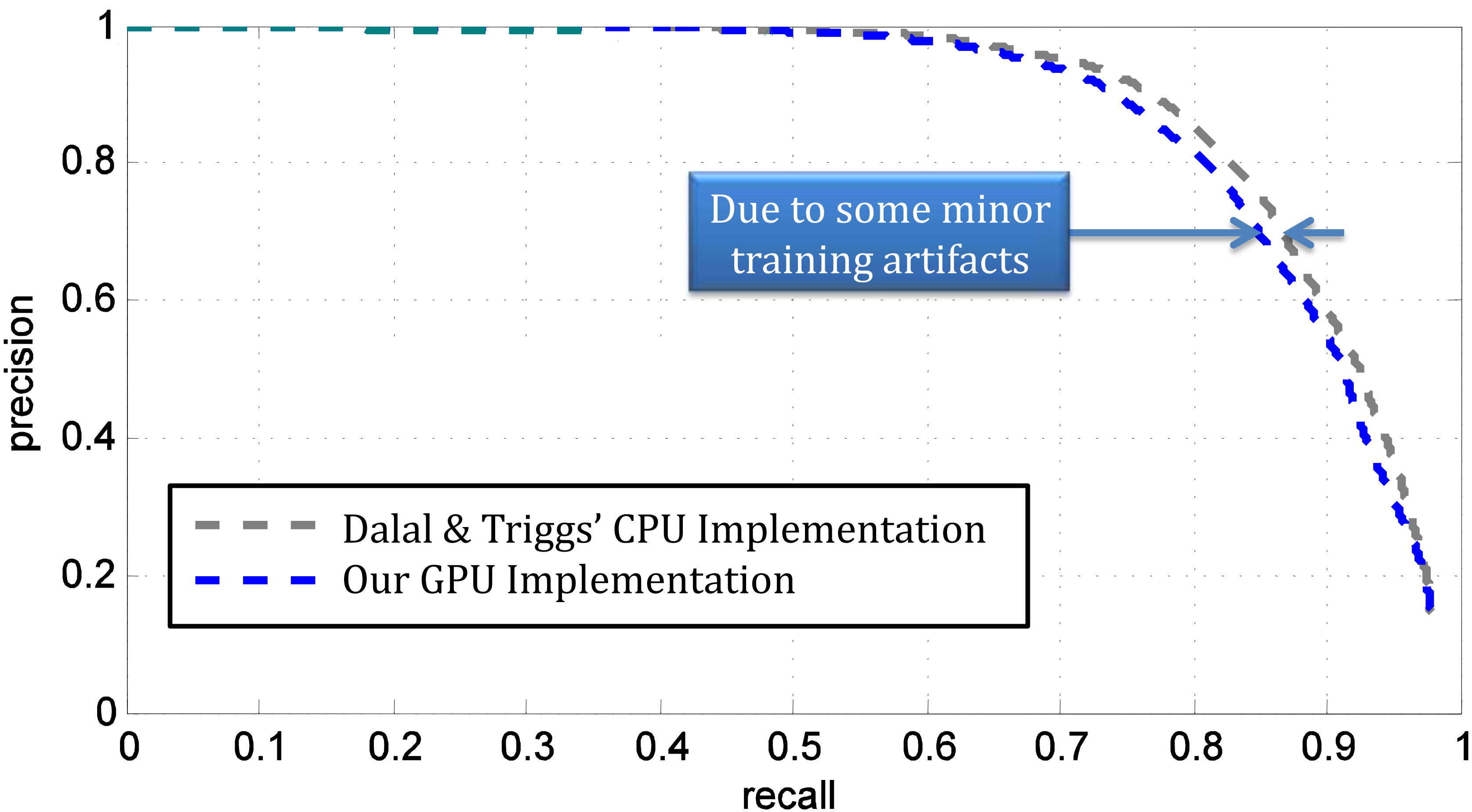
Detection

Tracking

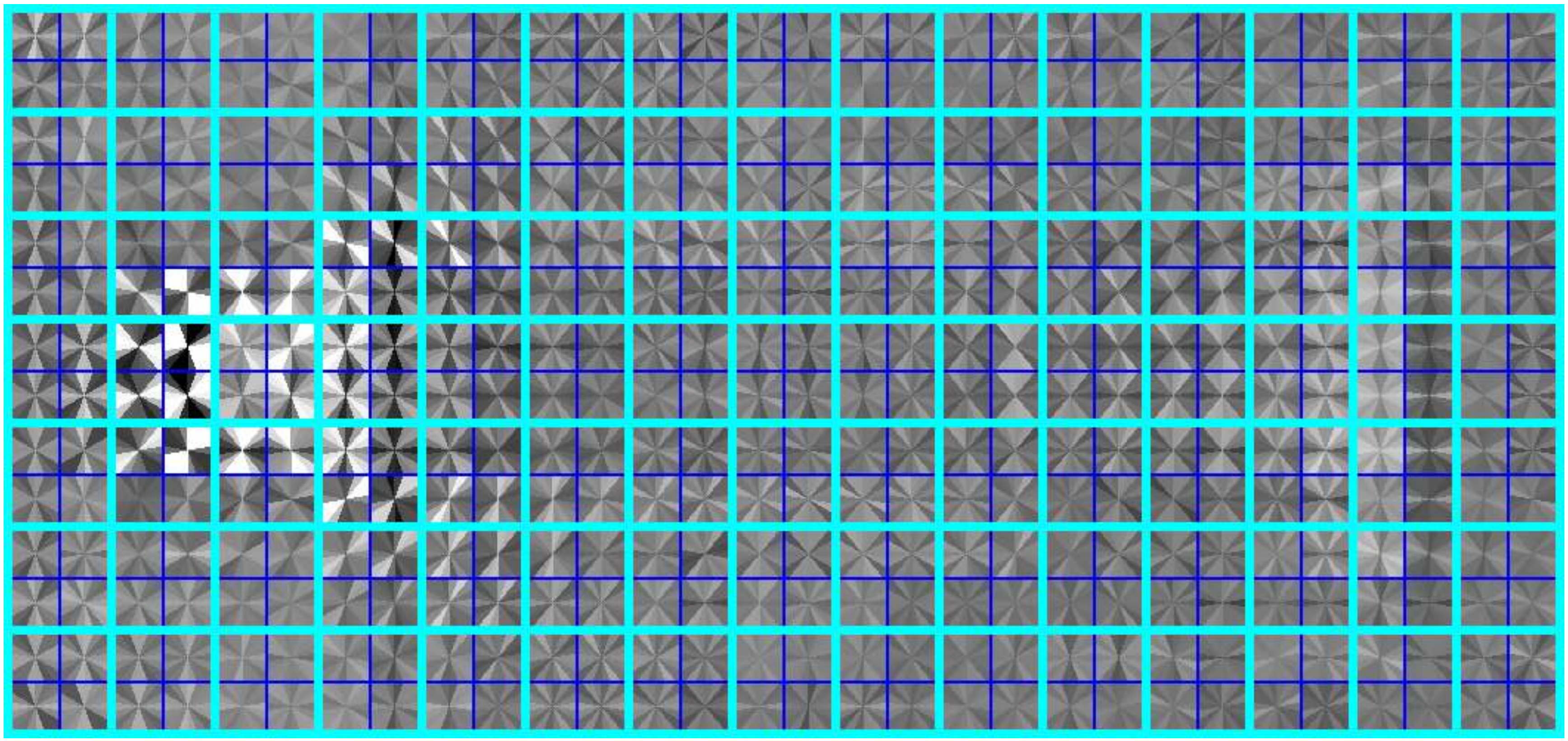
Analysis

- Identifying individual people
  - Sinha, Balas, Ostrovsky, and Russell. Face Recognition by Humans: Nineteen Results All Computer Vision Researcher Should Know About. *IEEE*. 2006.
  - Phillips *et al*. The Gait Identification Challenge Problem: Data Sets and Baseline Algorithm. *ICPR*. 2002.
  - Sundaresan, Roy-Chowdhury, and Chellapa. A Hidden Markov Model Based Framework for Recognition of Humans from Gait Sequences. *ICIP*. 2003.
  - **Lee, Dalley, and Tieu. Learning Pedestrian Models for Silhouette Refinement. *ICCV*. 2003.**
  - Veeraraghavan, Roy-Chowdhury, and Chellappa. Matching Shape Sequences in Video with Applications in Human Movement Analysis. *PAMI*. 2005.
- Recognize events (loitering, theft, etc.)
  - Ivanonv and Bobick. Recognition of Visual Activities and Interactions by Stochastic Parsing. *PAMI*. 2000.
  - Vu, Bremond, and Thonnat . Temporal Constraints for Video Interpretation. *ECAI*. 2002.
  - Francois *et al.* VERL: An Ontology Framework for Representing and Annotating Video Events. *Multimedia*. 2005.
  - PETS 2006 and PETS 2007 workshops (many papers)
  - **Dalley, Wang, and Grimson. Event Detection using an Attention-based Tracker. *PETS.* 2007.**
- Model flow patterns and site usage
  - Stauffer. Automatic Hierarchical Classification using Time-based Co-occurrences. *CVPR*. 1999.
  - Andrade, Blunsden, and Fisher. Modeling Crowd Scenes for Event Detection. *ICPR*. 2006.
  - Wang, Ma, and Grimson. Unsupervised Activity Perception by Hierarchical Bayesian Models. *CVPR*. 2007.
  - Wang *et al.* Trajectory Analysis and Semantic Region Modeling using a Nonparametric Bayesian Model. *CVPR*. 2008.

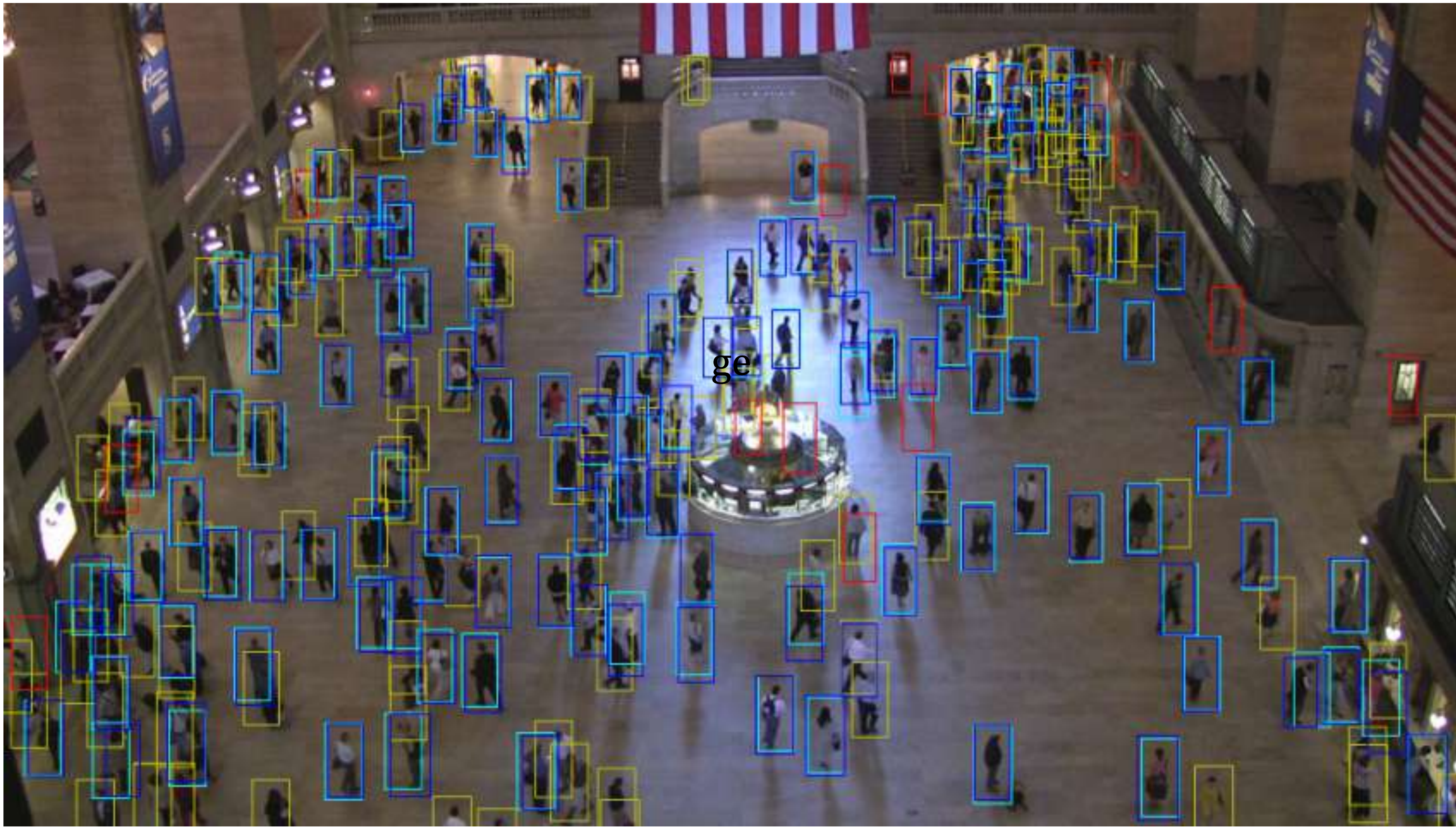
Same Quality
(*minor differences due to training tweaks*)



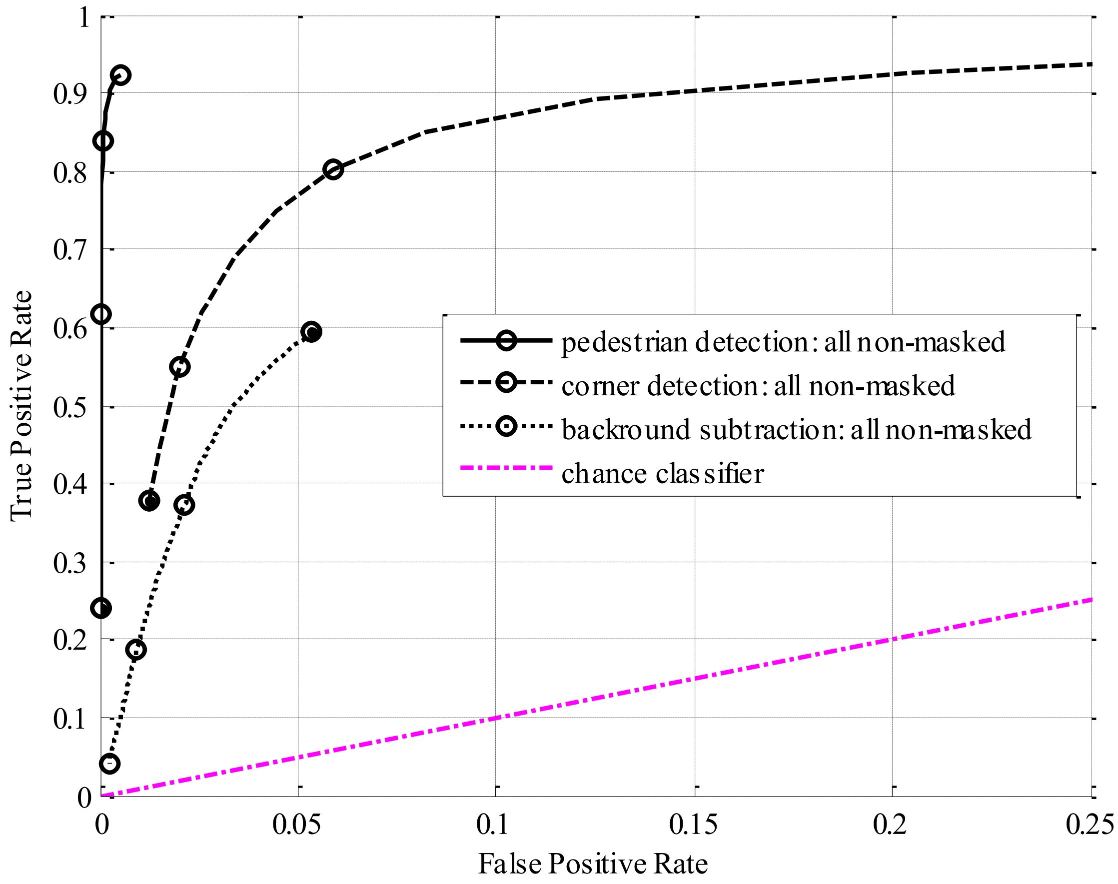
A Learned Classification Boundary (rotated)



Detections on One Frame

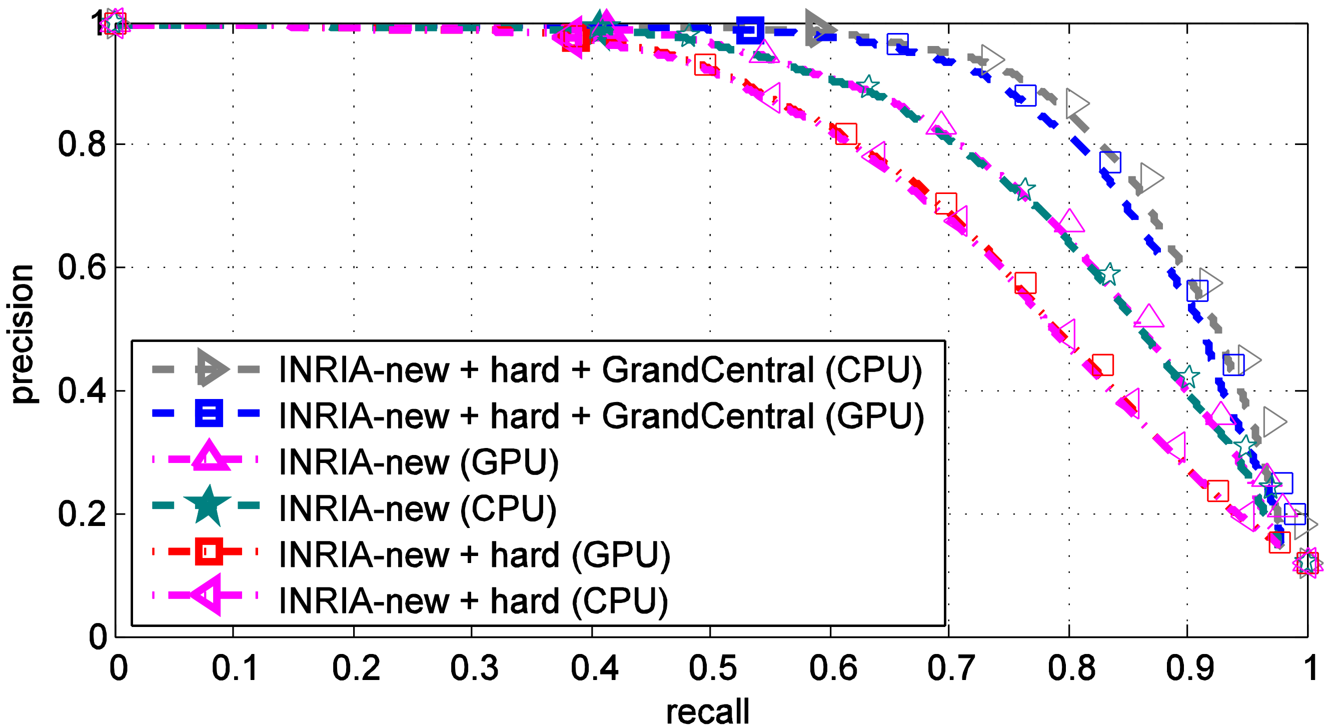


ROC Curves

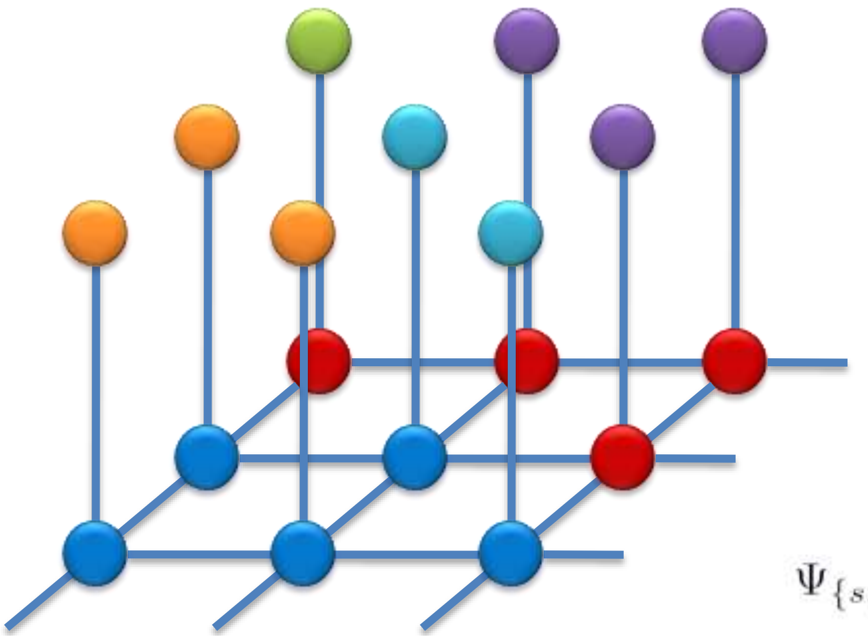


True Positive Rate
False Positive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.05
0.1
0.15
0.2
0.25
pedestrian detection: all non-masked
corner detection: all non-masked
backround subtraction: all non-masked
chance classifier

Training Set Influence



MRF equations[*MG05]*

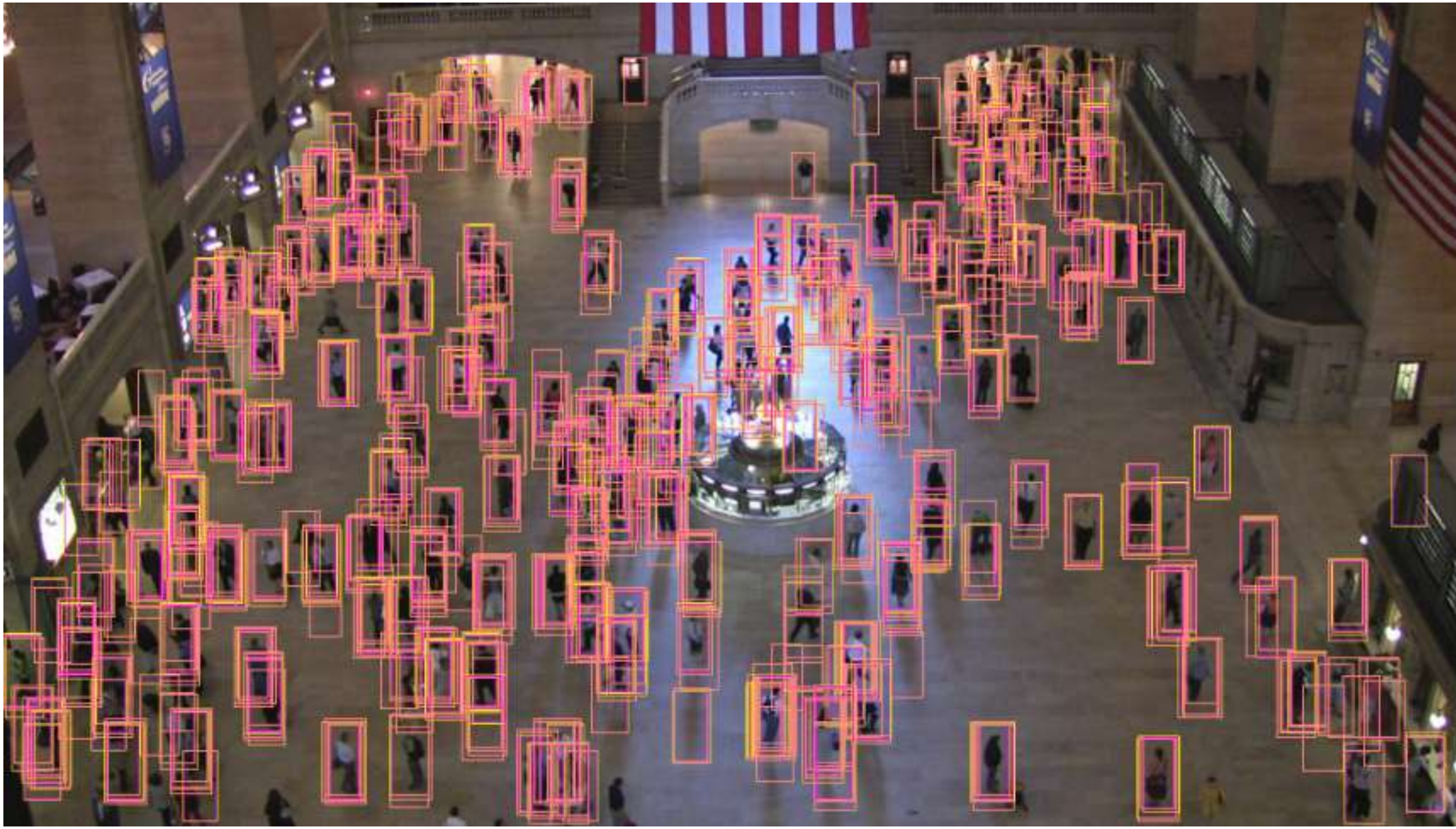


$$\Psi_{\{s,r\}}(X_s^t, X_r^t) = \begin{cases} \psi_1, & \text{if } X_s^t = X_r^t = 1 \\ \psi_2, & \text{if } X_s^t = X_r^t = 0 \\ \psi_3, & \text{if } X_s^t \neq X_r^t \end{cases}$$

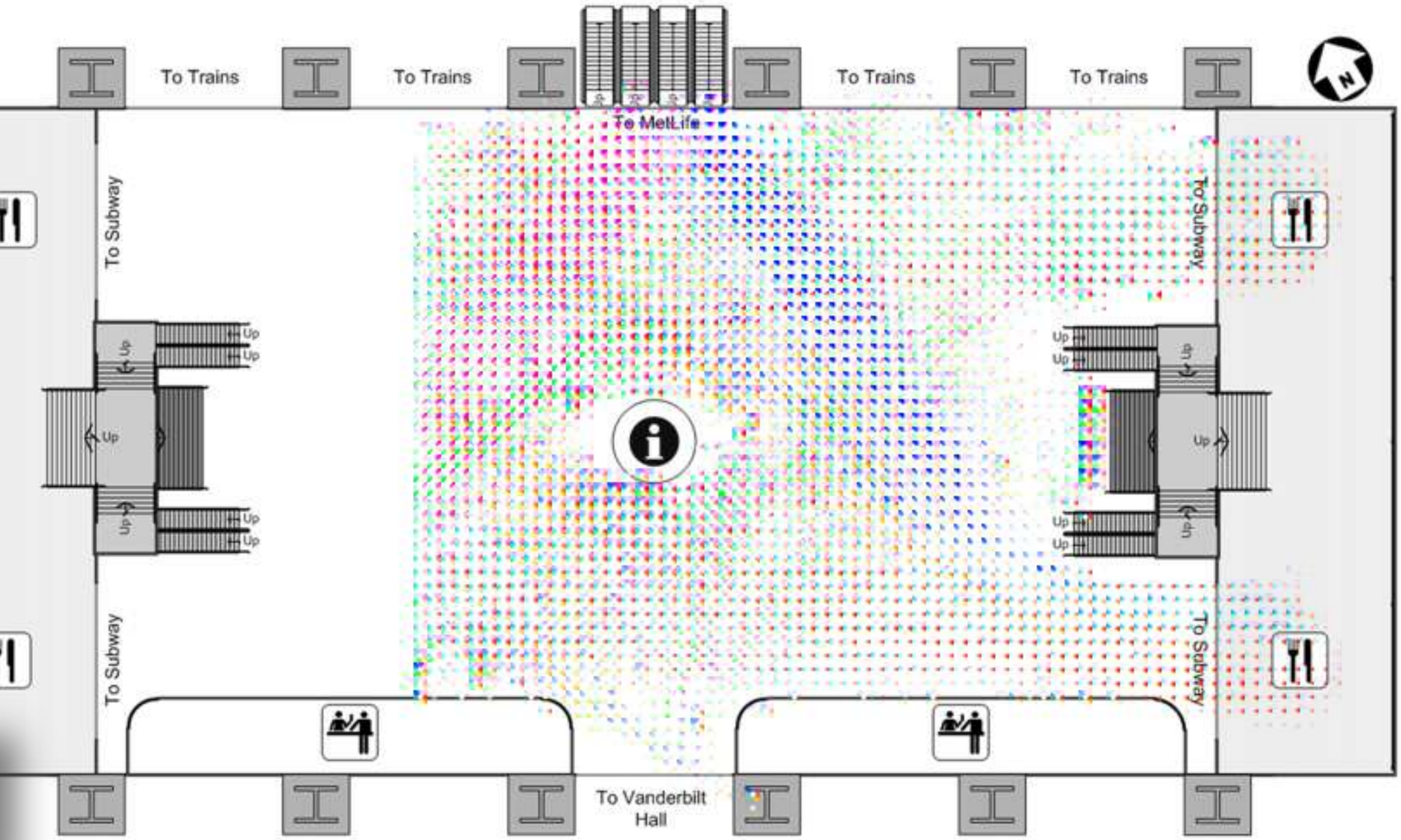
$$\Phi_{\{s\}}(X_s^t, D_s^t) = \begin{cases} \delta(d_s^t), & \text{if } X_s^t = 0 \\ \ln 2^{24}, & \text{if } X_s^t = 1 \end{cases}$$

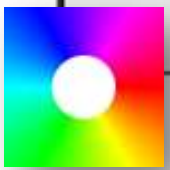
$$\Theta_{\{s^t, s^{t'}\}}(X_s^t, X_s^{t'}) = \begin{cases} \theta_1, & \text{if } X_s^t = X_s^{t'} = 1 \\ \theta_2, & \text{if } X_s^t = X_s^{t'} = 0 \\ \theta_3, & \text{if } X_s^t \neq X_s^{t'} \end{cases}$$

Detections before Global Optimization



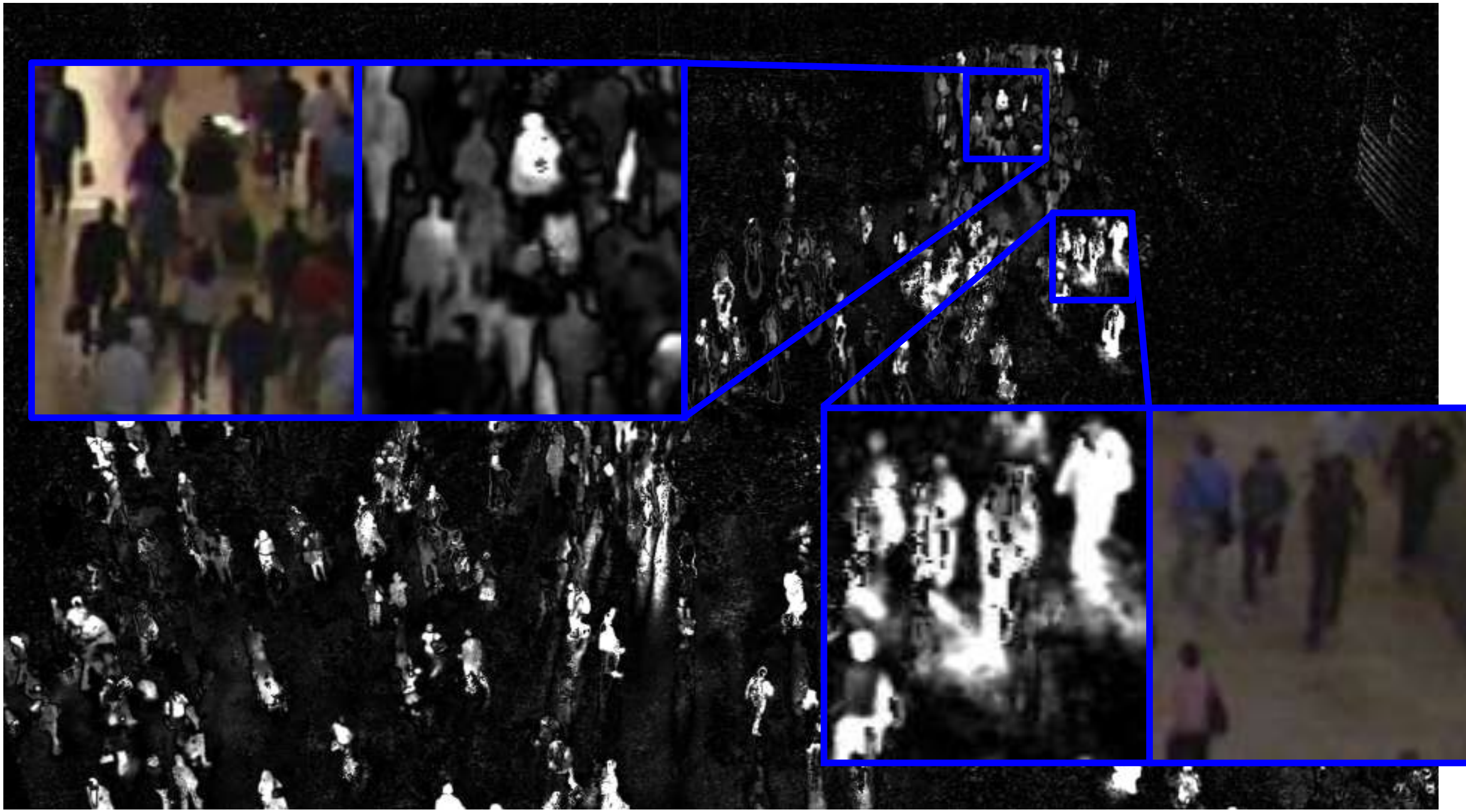
Sample Marginal of Observations





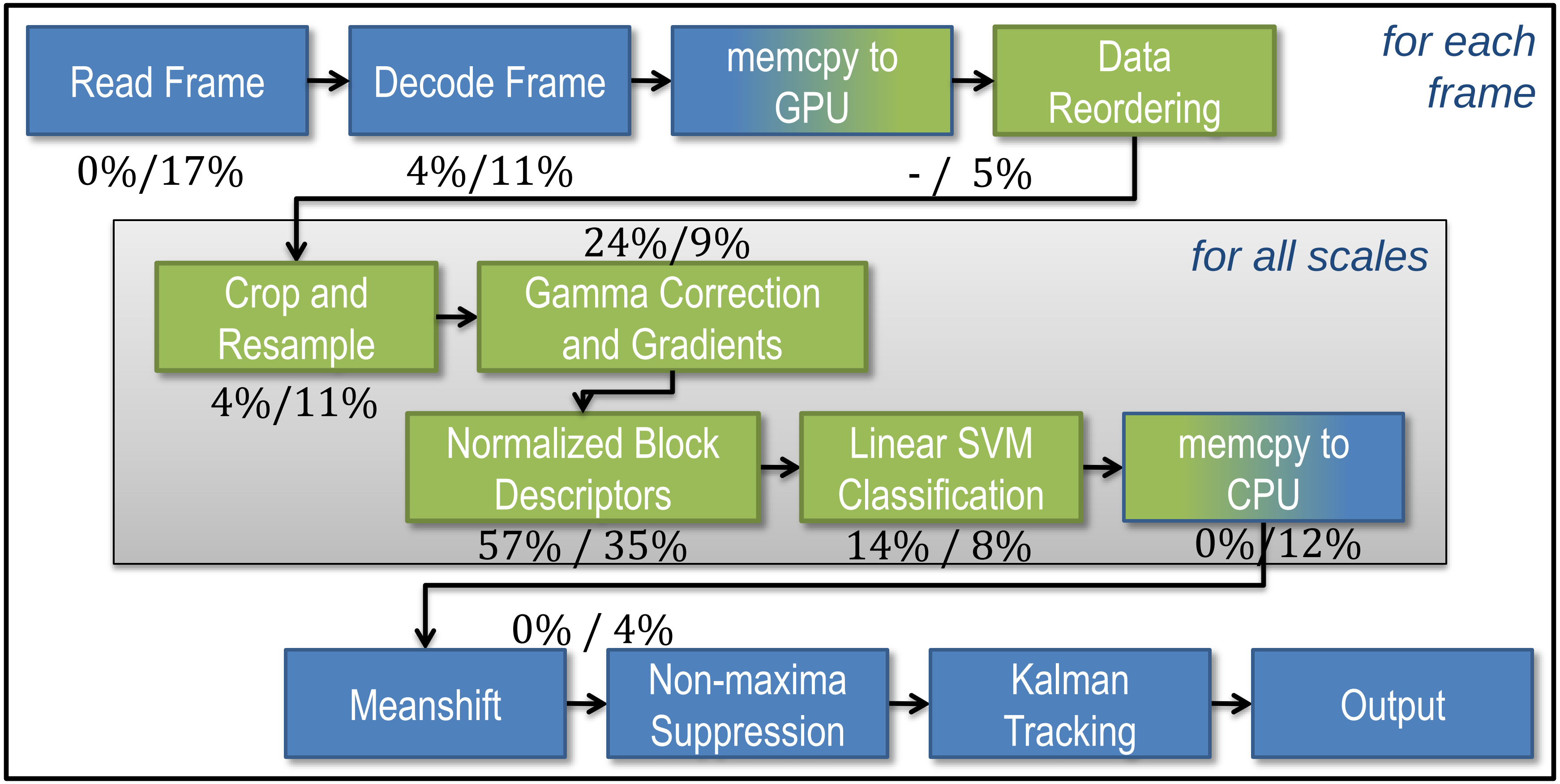
Our Data

Foreground Likelihood



Training Data

Our CUDA Pipeline: Percent Time per Module



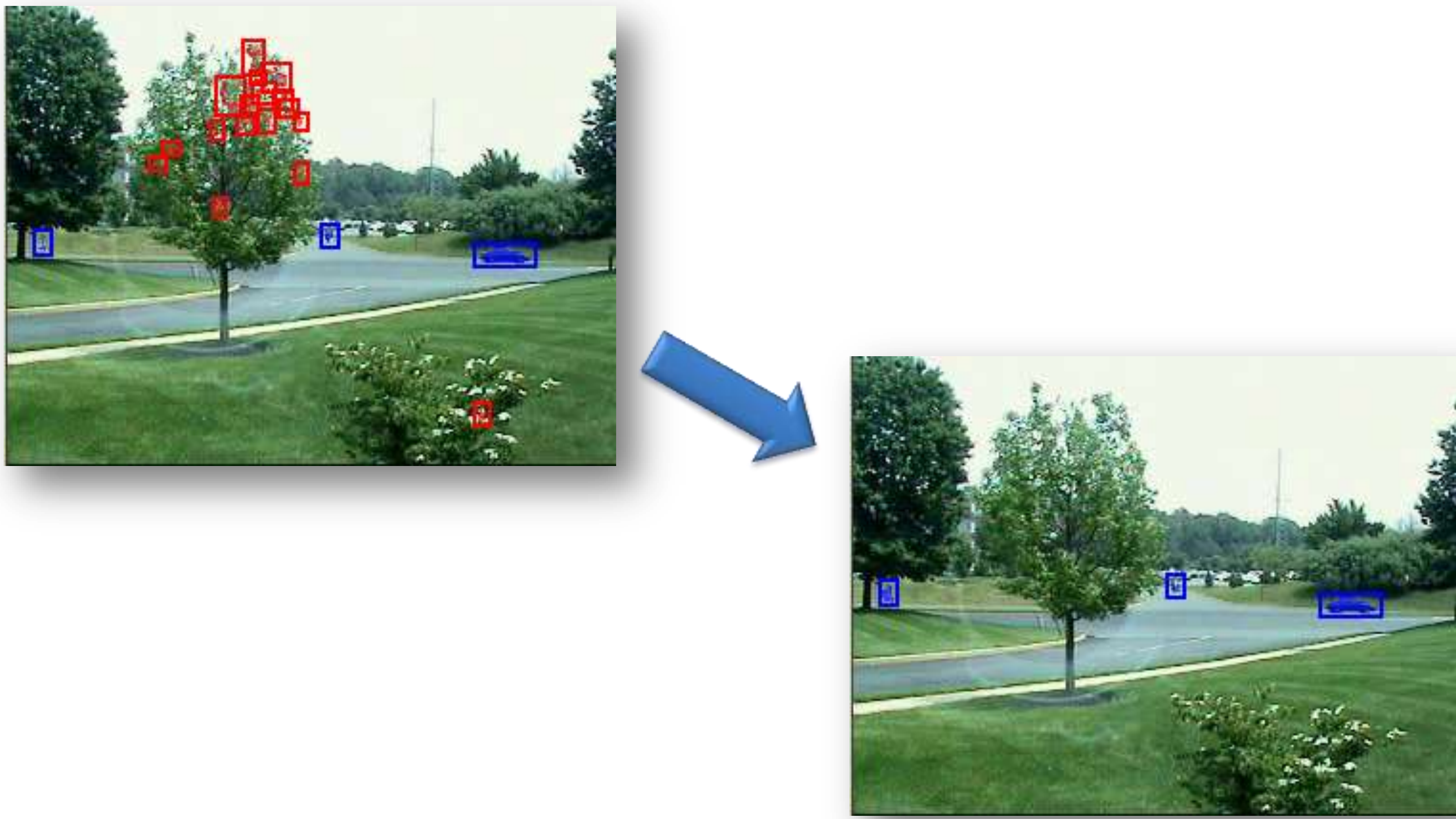
CPU vs. GPU Times:

Results from a Simplified Profiling Application

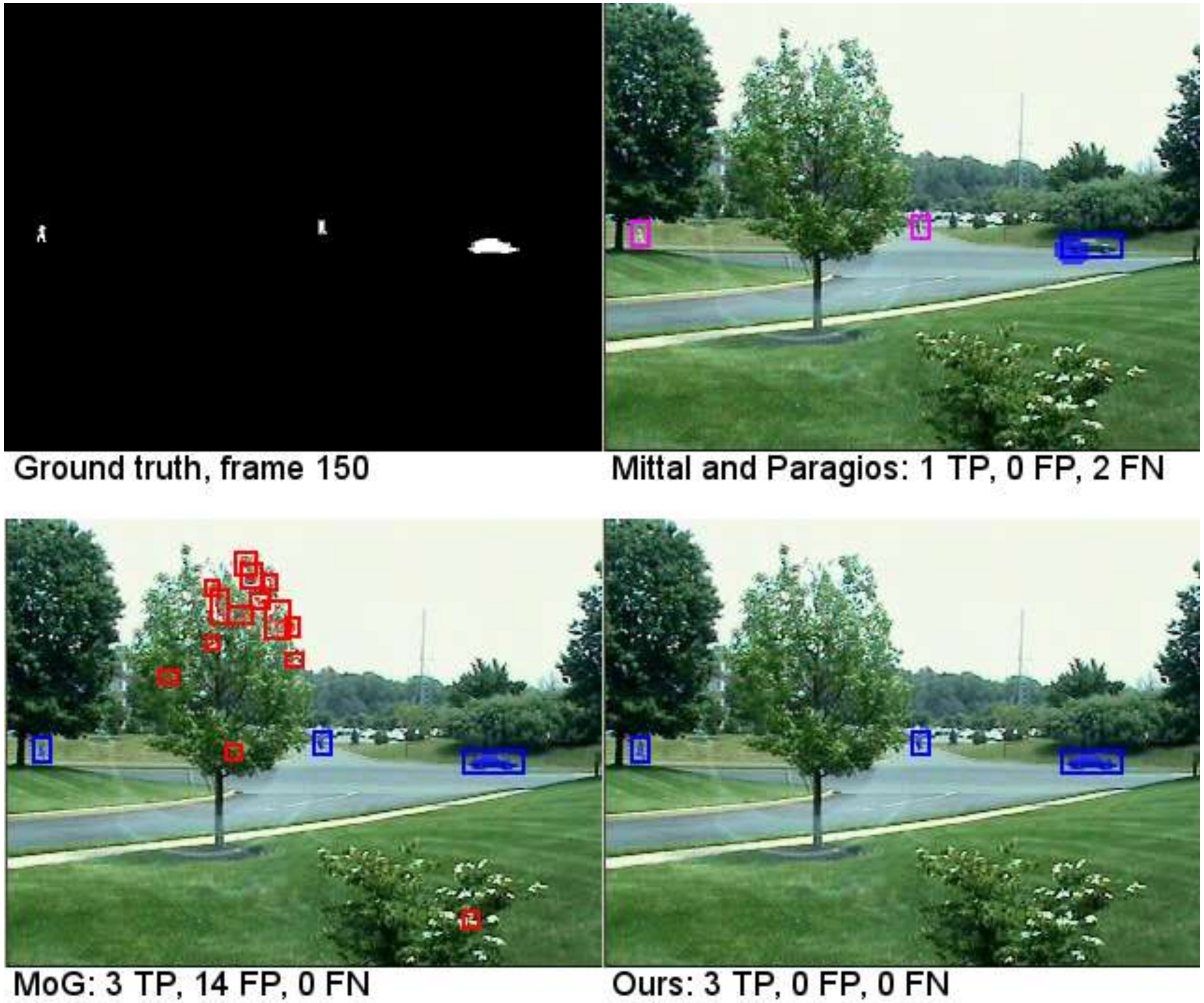
Processing Step	Time (CPU Impl.)		Time (GPU Impl.)		GPU Impl. Speedup
Read input (CPU)	68.0 ms	(0%)	68.0 ms	(17%)	
GPU resizer setup			18.1 ms	(5%)	
Resize	1,045.8 ms	(4%)	43.0 ms	(11%)	24.3×
Gradients	5,636.2 ms	(24%)	34.4 ms	(9%)	164.0×
Normalized block descriptors	13,412.3 ms	(57%)	137.2 ms	(35%)	97.7×
Window classification	3,159.1 ms	(14%)	31.4 ms	(8%)	100.6×
Cleanup	23.8 ms	(0%)	45.5 ms	(12%)	0.5×
Detection (CPU)	16.2 ms	(0%)	15.0 ms	(4%)	1.1×
TOTAL	23,082.4 ms		392.3 ms		58.8×

bgsub

Suppressing Spurious Detections

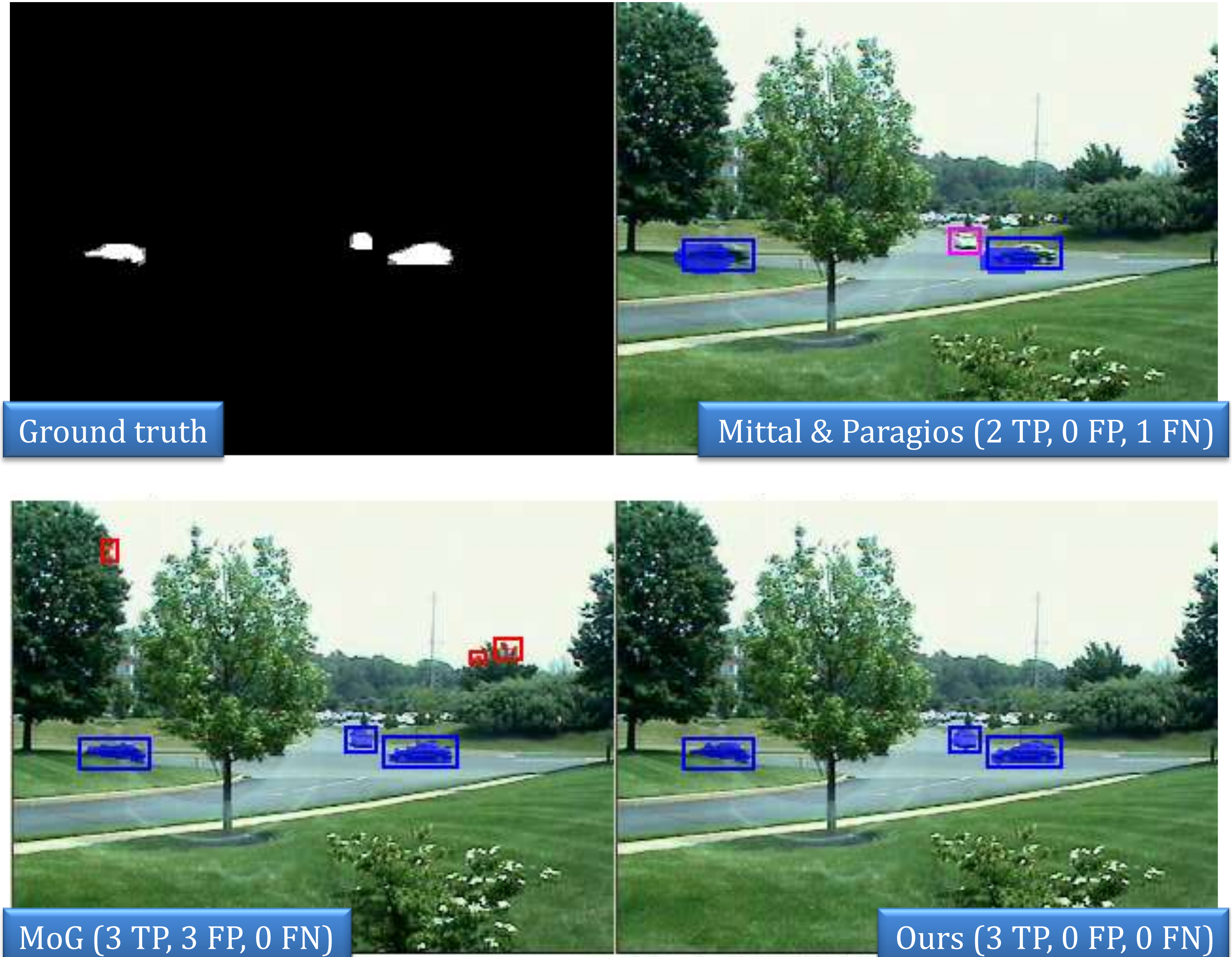


Results with Mittal & Paragios Traffic Clip



Key:

- **bboxes from ground truth**
  - **True positive**
  - **False positive**
  - **False negative**


Ground truth
Mittal & Paragios (2 TP, 0 FP, 1 FN)
MoG (3 TP, 3 FP, 0 FN)
Ours (3 TP, 0 FP, 0 FN)

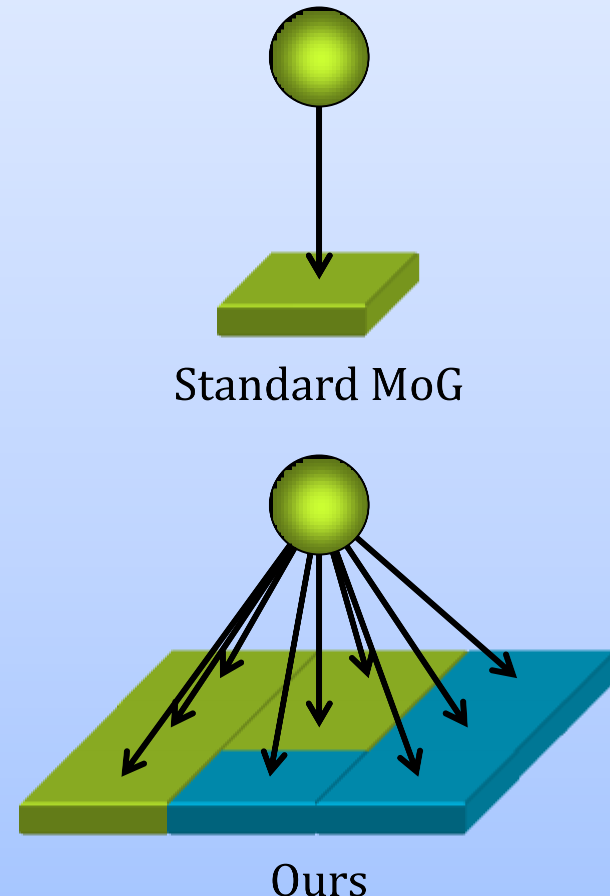
Our Model

- Mixture of Gaussians (MoG)

$$p(c_i|\Phi) \propto \sum_{\underline{j \in N_i}} w_j \mathcal{N}(c_i; \mu_j, \Sigma_j)$$

c_i	the observed color at pixel i
Φ	the model $\{w_j, \mu_j, \Sigma_j\}_j$
N_i	neighborhood of pixel i

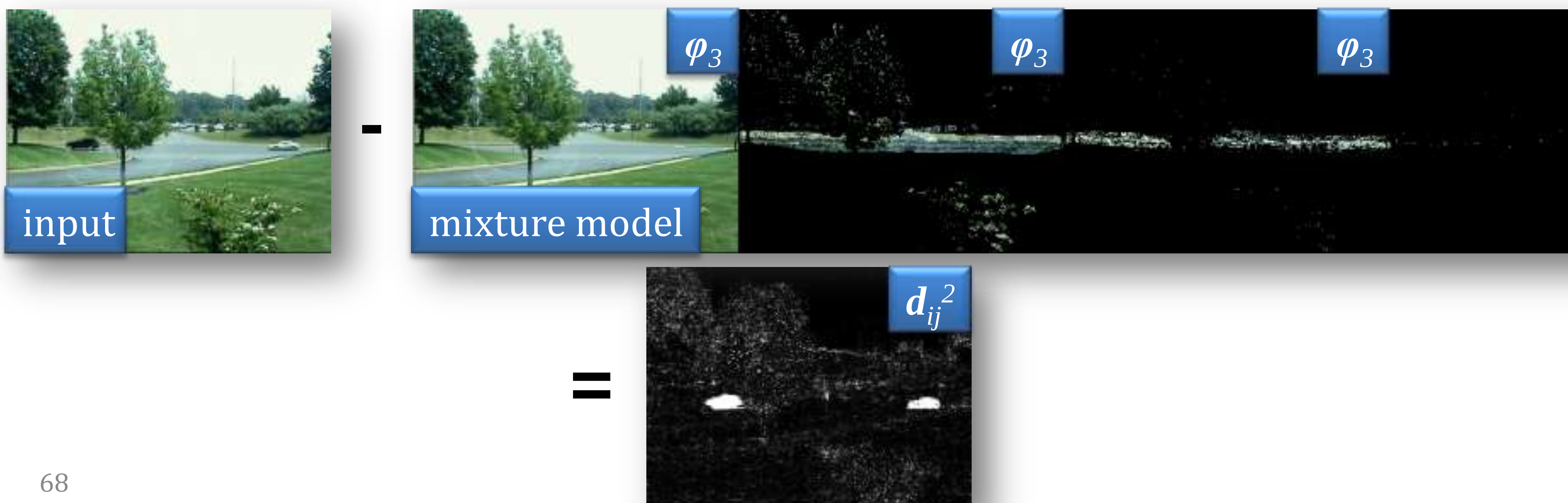
Pixels Affected by a Given Mixture Component



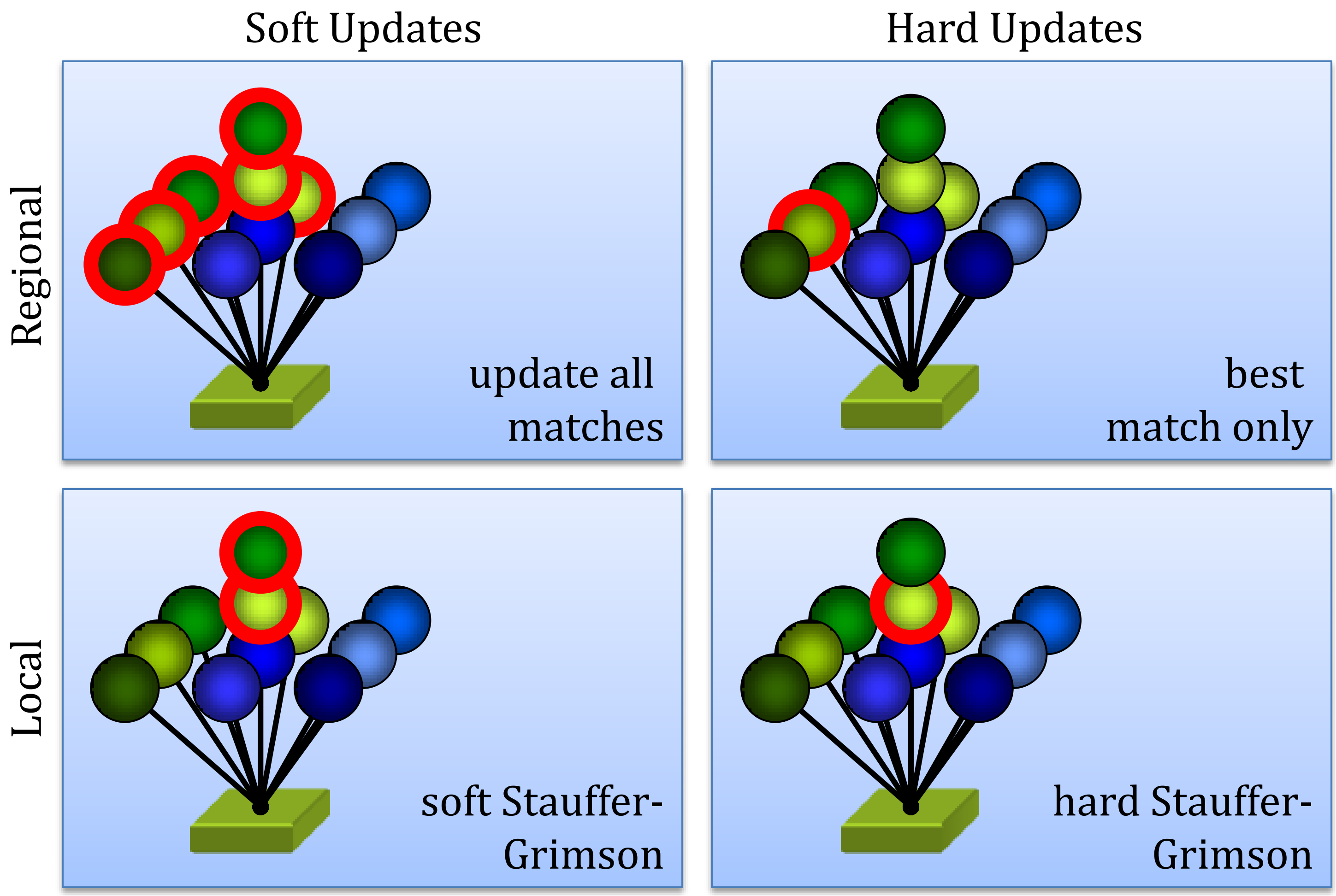
Foreground/Background Classification

- Find best matching background Gaussian, j
  - Use neighborhood
- Squared Mahalanobis Distance

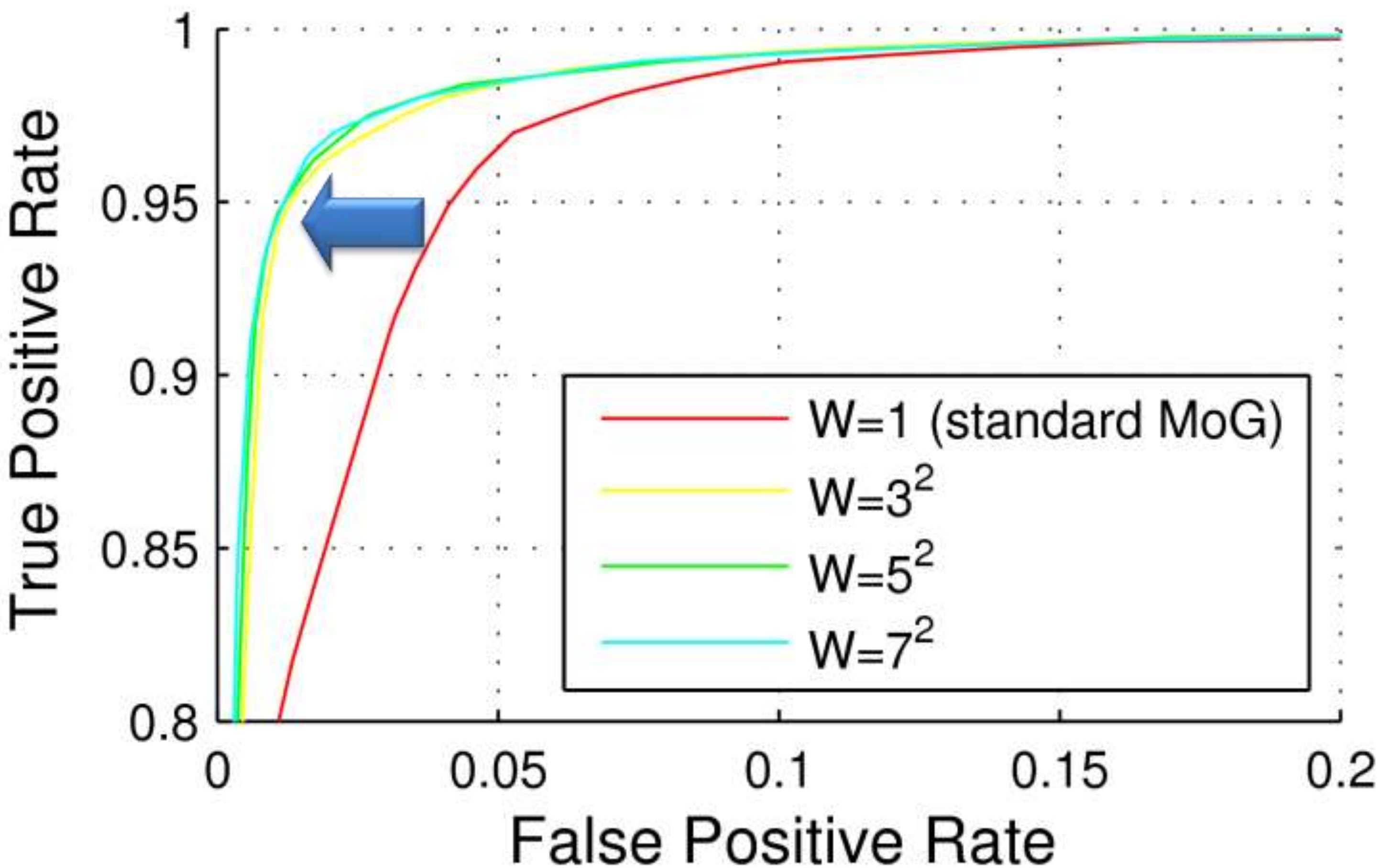
$$d_{ij} = (c_i - \mu_j)^T \Sigma_j^{-1} (c_i - \mu_j)$$



Model Update Options



ROC (Wallflower)



Parameter Sensitivity