Project 2

Student presentations

TLD

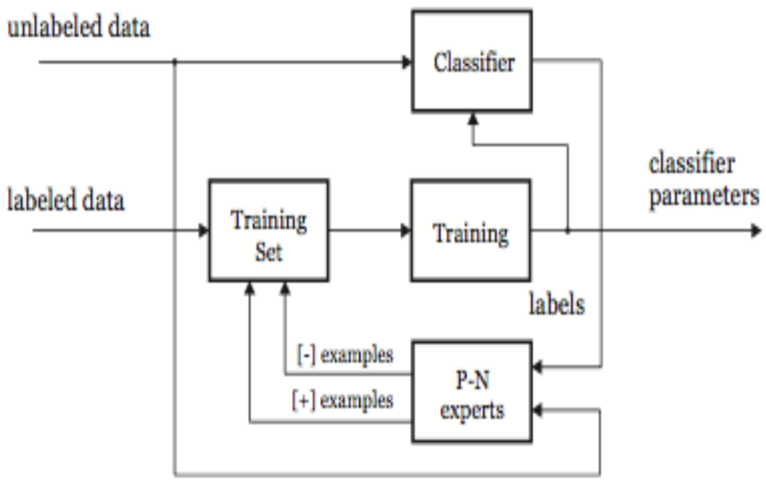
CS231b 2015 Project 2

Link

Iretiayo Akinola
Josh Tennefoss

Challenges

- Getting Matlab code to run properly
- Understanding the strange TLD code structure
- Speeding up runtime
- Accuracy

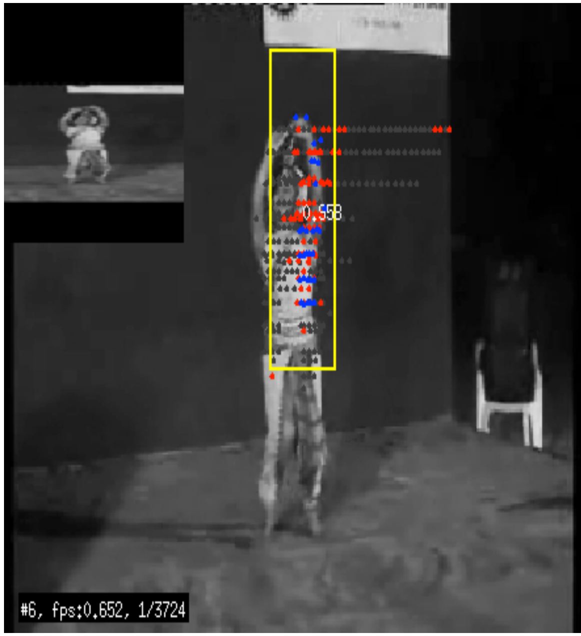


Our Experiments

- Detector Algorithm
  - 10-NN
  - SVM
- Training KNN
  - Set maximum number to keep
    - Keep most recent
    - Randomly keep 100, remove 100 per frame
- Combine Detector and Tracker
  - Penalize detector boxes if they are far from tracker

Detector - NN

1. First filter by variance
2. Use FERN features, 20 of them
3. 10-NN for pos and neg
4. Similarity = # places that are same, over NNs
5. Calculate confidences, C
6. If C's differ enough, output the higher one

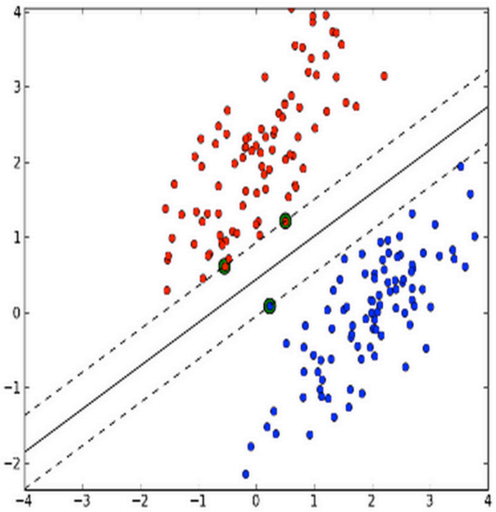


$$C_{pos} = \frac{\mathrm{Sim}_{pos}}{\mathrm{Sim}_{pos} + \mathrm{Sim}_{neg}}$$

$$C_{neg} = \frac{\mathrm{Sim}_{neg}}{\mathrm{Sim}_{pos} + \mathrm{Sim}_{neg}}$$

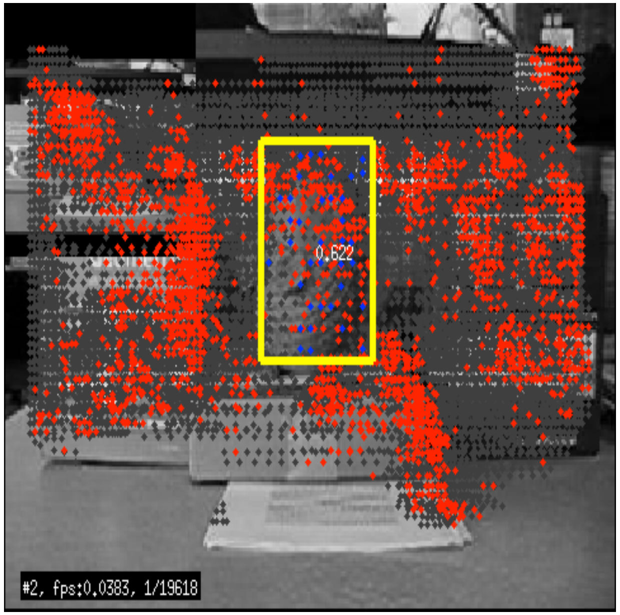
Detector - SVM

- SVM Classification
  - Keeps only updated support vectors from new frame.
  - Confidence score: fits sigmoid curve on margin of the SVM model.
  - Learning: updates model to handle false positives and false negative in new frame.



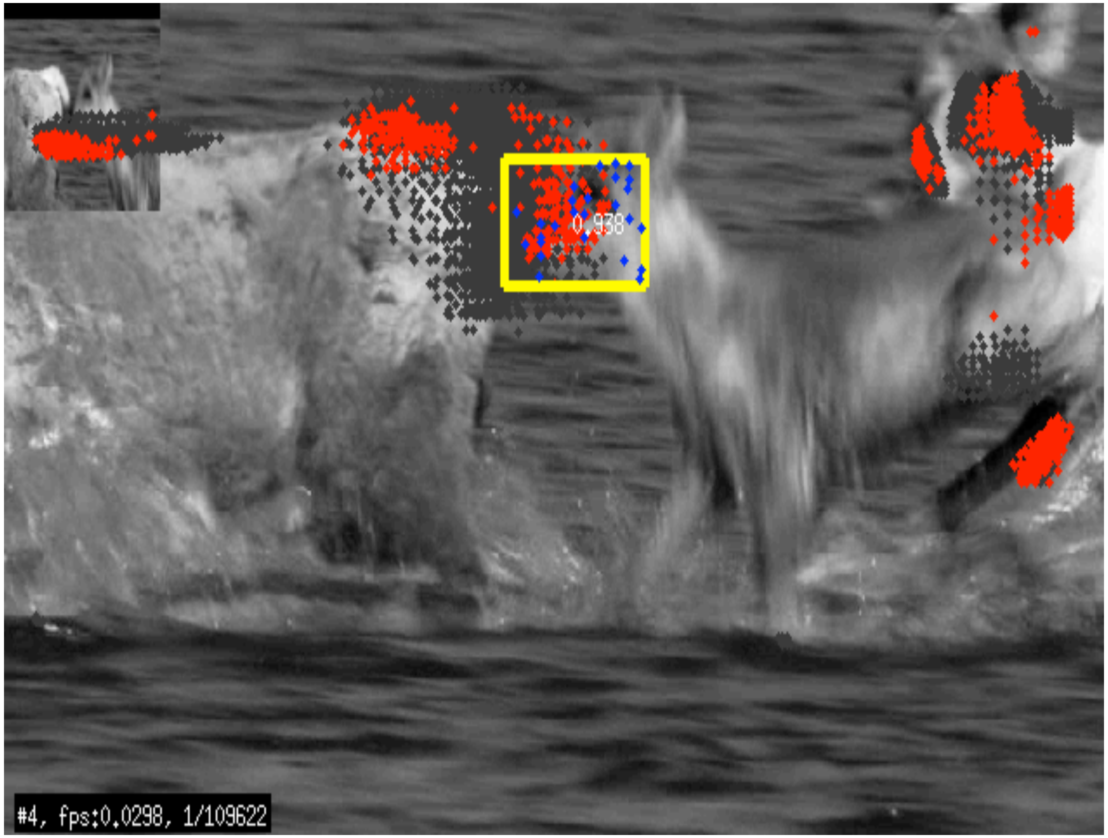
Training KNN

- Set maximum number to keep, MAX
  - Used 200, 500, 1000
- Attempt 1: Keep MAX most recent
- Attempt 2: randomly keep 10, remove 10 per frame to stay at MAX



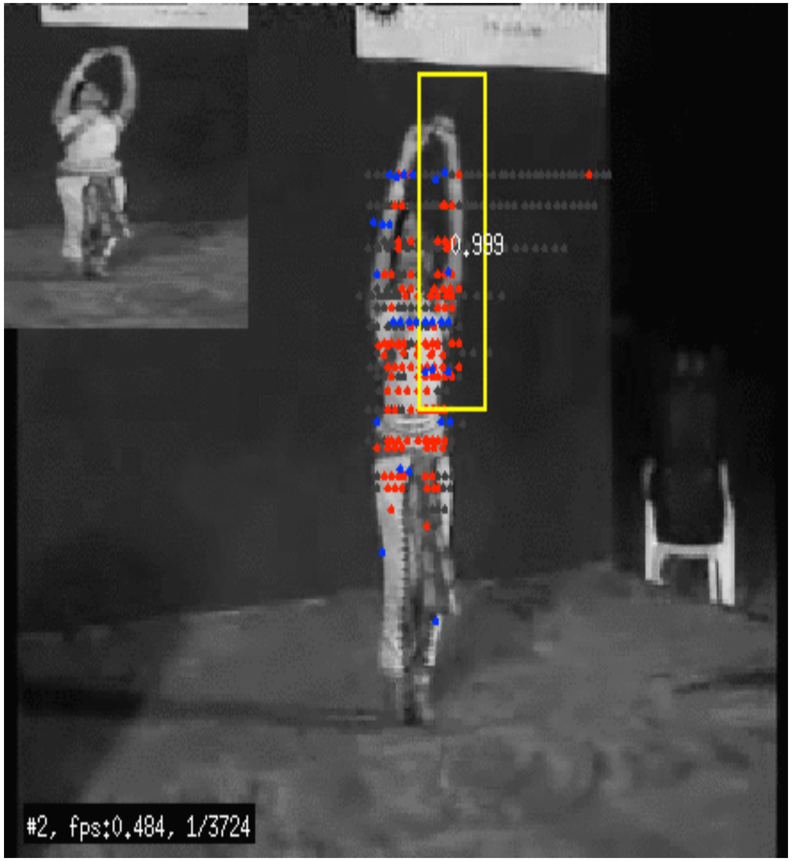
Combining Tracker and Detector

- Penalize detector boxes if they are far from tracker



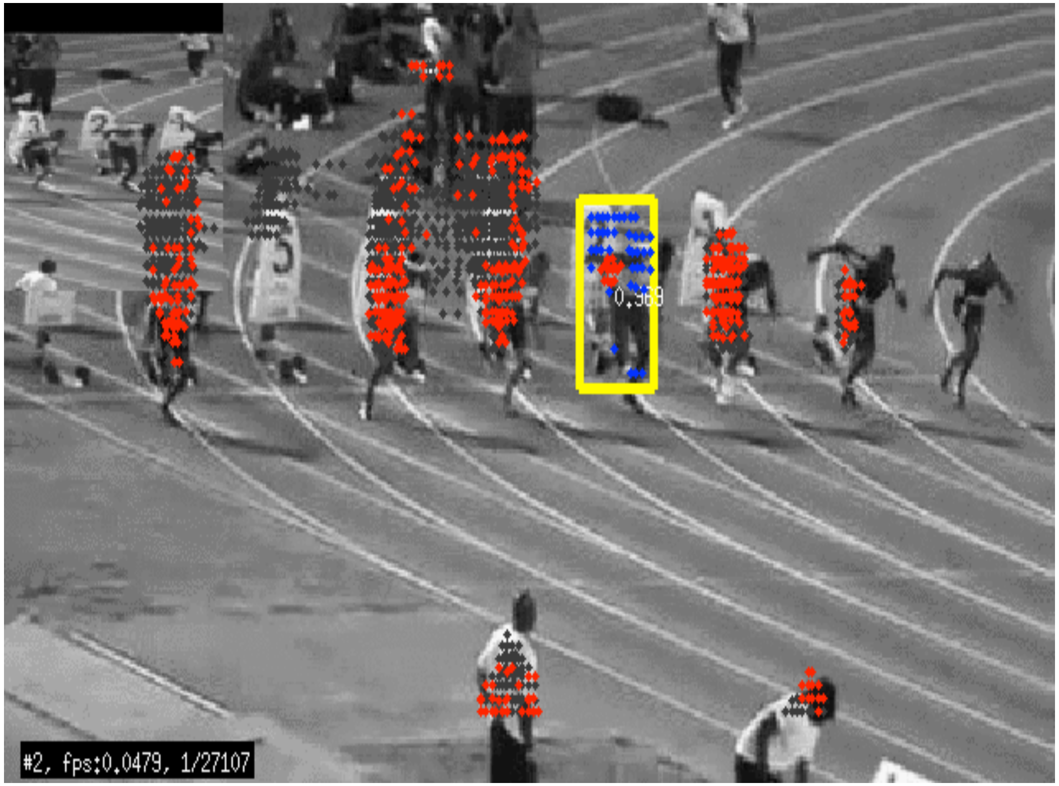
Results

link



Results

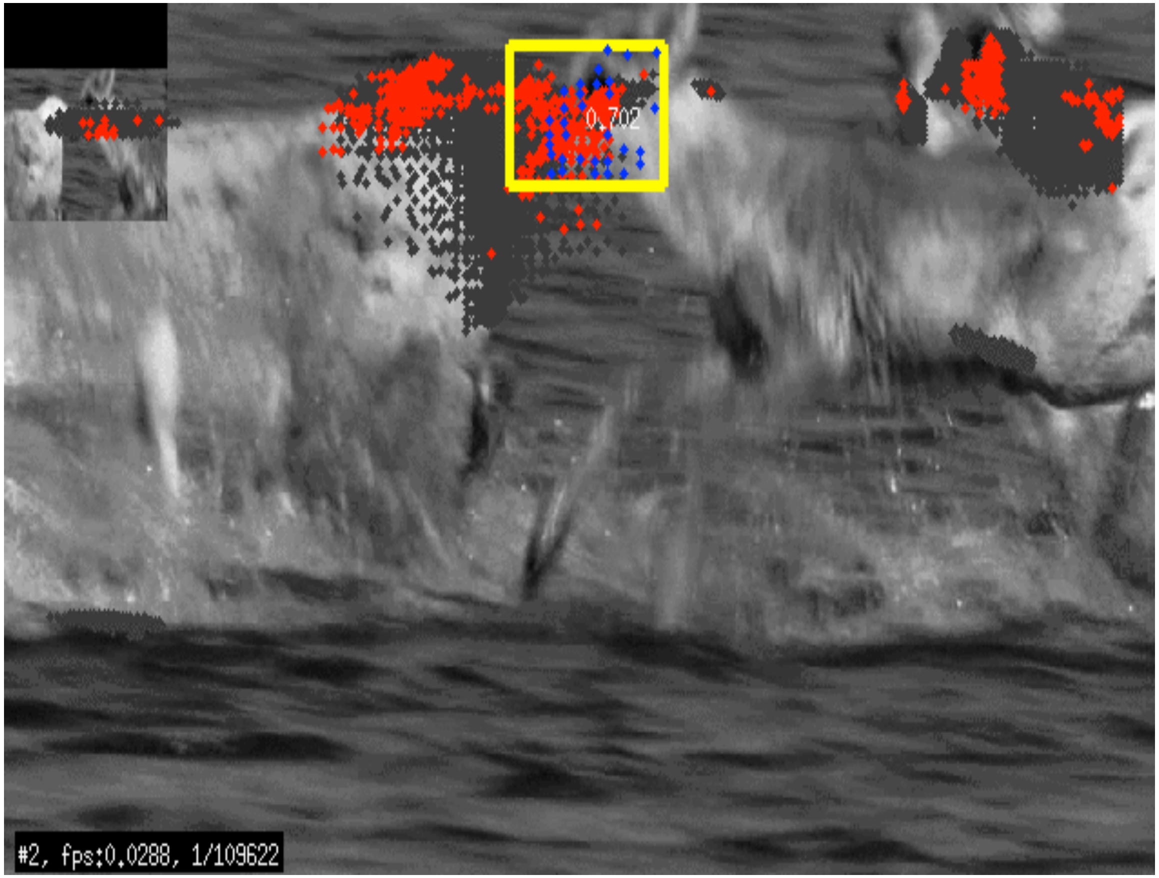
- We are still working on… late days :)



Hopefully Bolt is quicker than our algorithm thinks...

Results

Oops, where's the box?

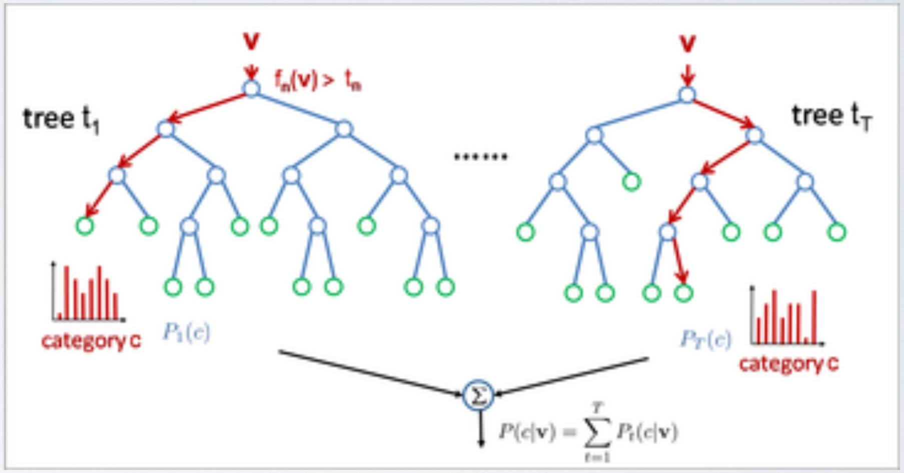


Tracking-Learning-Detection:
An Integrated Approach for Robust Tracking

Amani V. Peddada
amanivp@cs.stanford.edu

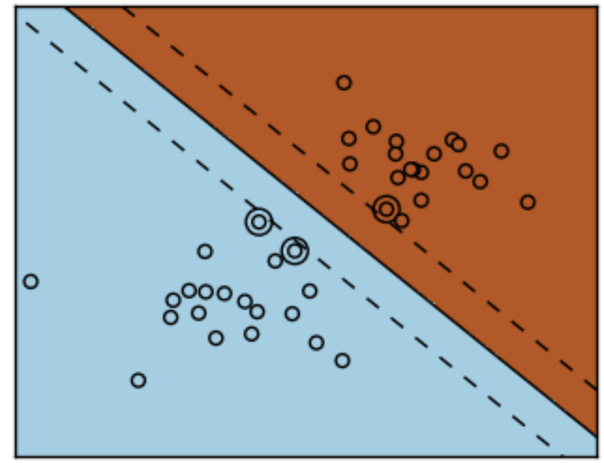
Implementation

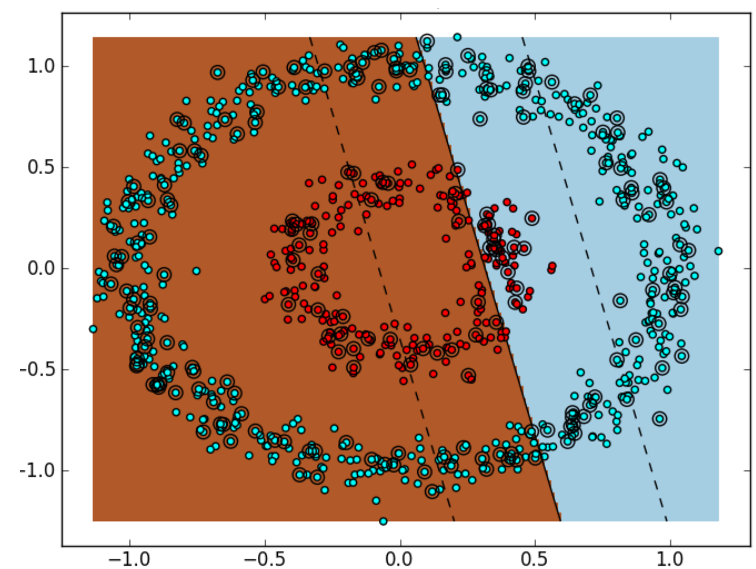
- Detector: Random Fern Forest + Nearest Neighbor
  - 10 trees, 9 comparisons
  - 50 trees, 6 comparisons
  - Filter by overlap
- LK Tracker
- Integrator that weights scores of tracker and detector.



Extension: Support Vector Machine

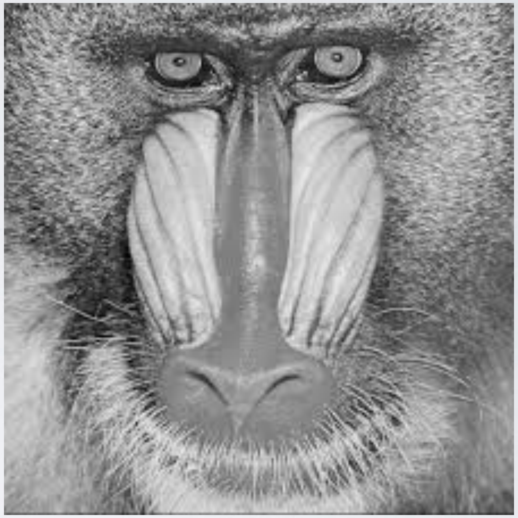
- Max-Margin Binary Classifier
- Trained on linear, quadratic, polynomial, and Gaussian Kernels.
- Superior Performance

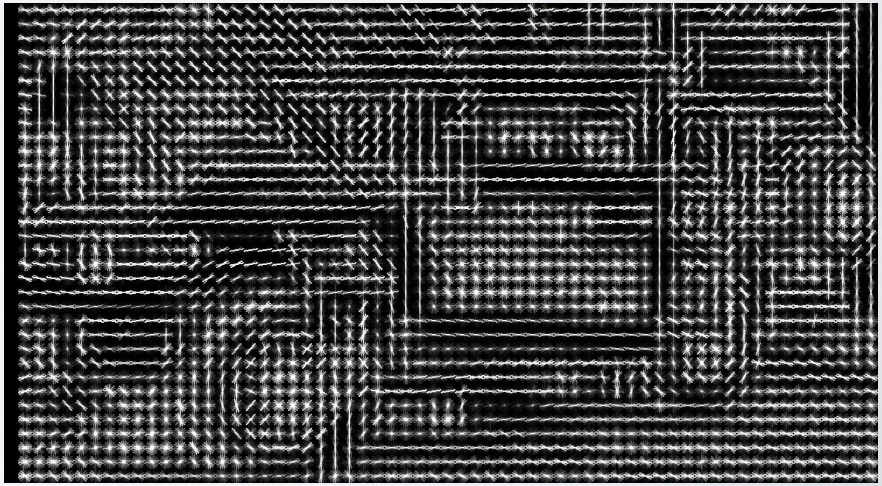




Features & Input Data

- Normalized, resized patch
- HOG Features + SVM — noisy performance
- HOG Features + Patch intensity — accurate, inefficient





The Integrator

- Finding balance between detection and tracking output is key
- Use confidences as measure of accuracy
- Strategies:
  - Use tracking prediction unless the max detection confidences is larger by a margin
  - Utilize a weighted average of bounding boxes based on confidences

Results

	Dancer2	Deer	Car4	Bolt2
SVM	57.19	51.7	50.1	21.0
Fern Forest	64.77	16.9	14.1	31.2

Average Overlap

Average MAP

	Dancer2	Deer	Car4	Bolt2
SVM	57.4	54.2	46.6	2.90
Fern Forest	89.4	14.45	3.1	5.87

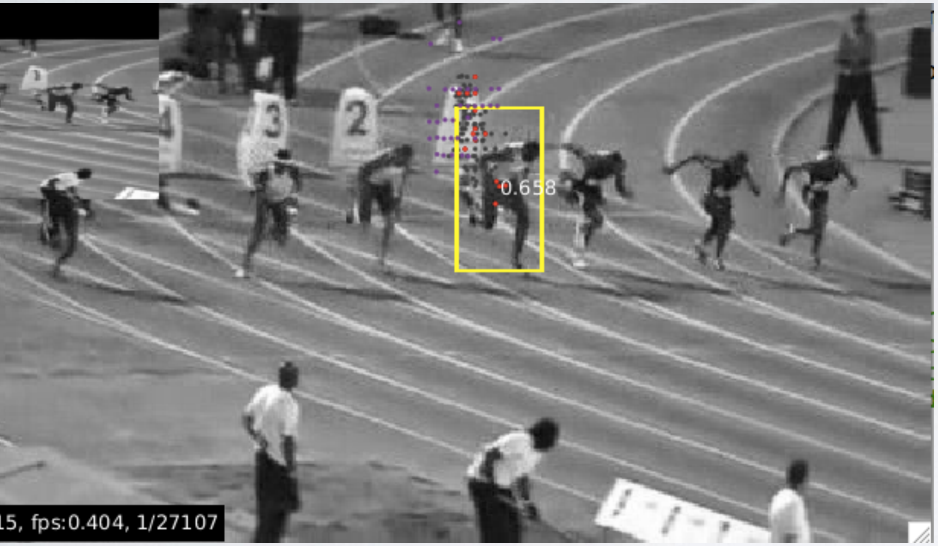
Analysis

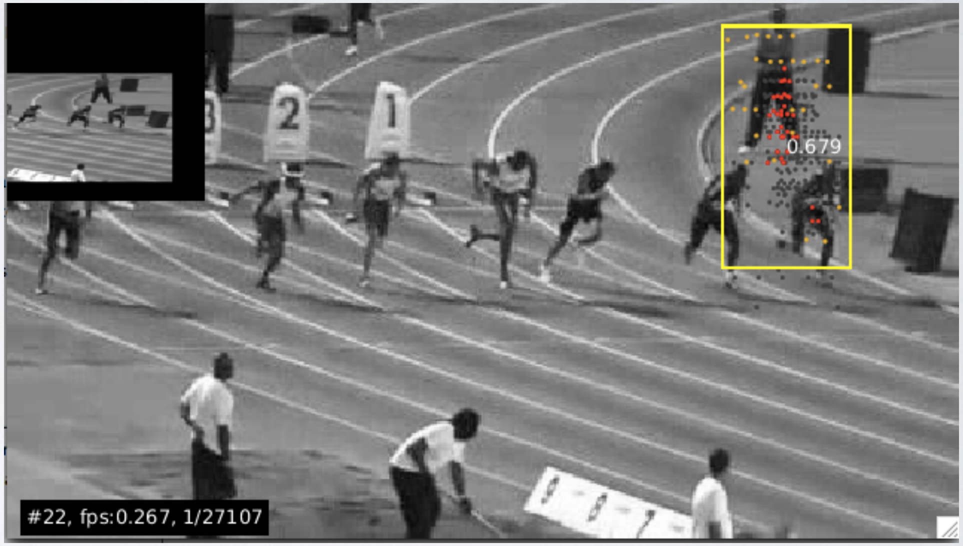
9 comparisons

6 comparisons

Analysis

HOG Features





Large jumps between frames

Analysis

SVM vs. Fern Forest

SVM - with Patch features

Fern Forest - 60 classifiers, 5 comparisons

Analysis

SVM vs. Fern Forest

SVM - with Patch features

Fern Forest - 60 classifiers,
5 comparisons

Further Extensions

- Information Gain to determine optimal tree structure
- Difference between mean values of sub-patches as binary tests - less noisy.
- Other discriminative classifiers — feed-forward Neural Networks (trained less often)

Thank you!

TLD

Bryan Anenberg & Michela Meister
10 May 2015

Results

Standard Implementation

varied performance across different videos ranging from <20% avg. overlap to >65%

Extensions

random fern classifier

strategies for generating positive and negatives

strategies for fusing the detection bounding boxes from your learned detector, and the boxes obtained from the KLT tracker

Standard Implementation

Bolt2: evaluation values: average-overlap=0.137347, success auc=0.160500, map=0.027264
Car4: evaluation values: average-overlap=0.653871, success auc=0.654000, map=0.838637
Deer: The evaluation values: average-overlap=0.302865, success auc=0.320423, map=0.323944
Dancer2: evaluation values: average-overlap=0.672883, success auc=0.674000, map=0.992122

Tracking Project

Eric Holmdahl

231B

TLD Tracking: Results (no extensions)

- First 20 frames of Car4:
  - mAP: 1.0
  - Average overlap: .86
- Full Car4:
  - mAP: .79
  - Average overlap: .70

Using BRIEF Features

More specifically, we define test τ on patch $\mathbf{p}$ of size $S \times S$ as

$$\tau(\mathbf{p}; \mathbf{x}, \mathbf{y}) := \begin{cases} 1 & \text{if } \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases} , \tag{1}$$

where $\mathbf{p}(\mathbf{x})$ is the pixel intensity in a smoothed version of $\mathbf{p}$ at $\mathbf{x} = (u, v)^\top$. Choosing a set of n_d $(\mathbf{x}, \mathbf{y})$-location pairs uniquely defines a set of binary tests. We take our BRIEF descriptor to be the n_d-dimensional bitstring

$$f_{n_d}(\mathbf{p}) := \sum_{1 \leq i \leq n_d} 2^{i-1} \, \tau(\mathbf{p}; \mathbf{x}_i, \mathbf{y}_i) \ . \tag{2}$$

Pyramid Sampling

- Instead of static 15x15 patch, take increasing size patches (30x30, 60x60, etc) to try and improve resolution
- Similar to pyramid-style SIFT feature extraction

Extension Results

- Should have by class Monday!

CS231B Project #2:

Tracking – Learning - Detection

Tugce Tasci

Stanford Universtiy

05/11/2015

Object Detection

Variance Filter → Ensemble Classifier → Nearest Neighbor Classifier

If variance

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2,$$

is smaller than a threshold, patch fails

Probability P_{pos} = $P(y=1|F)$ is calculated with random fern classification. If $P_{pos}<0.5$, patch fails

Relative similarity of the current patch and previous patches is calculated (online learning). If $S^r<0.6$, patch fails.

Integrator

Decision is based on the number of detections, their confidence values and the confidence of the tracking result

```
If T ~=0
    if |D|==1 && conf(D)>conf(T)
        result = D
    else
        result = T
else if |D| == 1
    result = D
```

For all other cases, object is assumed invisible.

Learning, P/N experts

for all patches B

if overlap>0.6 and classifyPatch(B)<0.5

calculate and update features for all ferns

#of (+) patches +=1

else if overlap<0.2 and classifyPatch(B)>0.5

calculate and update features for all ferns

#of (-) patches +=1

if conf(result)>thr^{-}

add it to (-) patches

If conf(result)<thr^{+}

add it to (+) patches

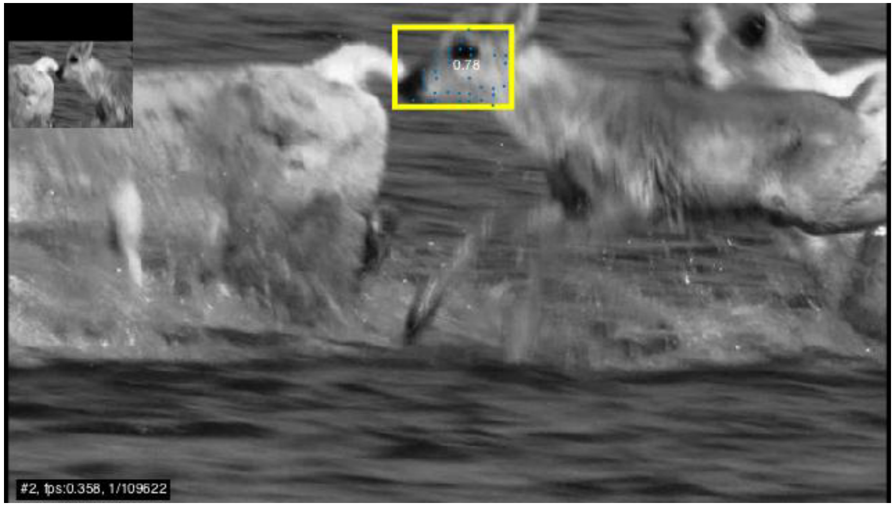
Results

Dancer

average-overlap=0.668031,
success auc=0.670667,
map=0.977073

Elapsed time is 0.72538 seconds.

Deer



average-overlap=0.573483,
success auc=0.585211,
map=0.600965

Elapsed time is 2.18415 seconds.

Tracking Project

Jasper Lin

My Implementation

- 10 Fern classifiers with 13 comparisons each
- Limit to 600 positive examples
- Achieved better than baseline for both Car4 and Dancer2

Preliminary Results with Datasets

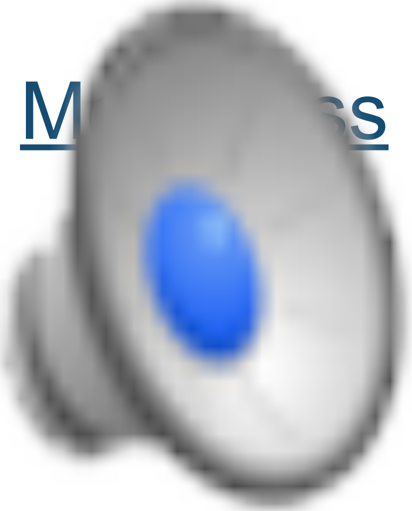
	Overlap	mAP	Notes
Car4	0.747	0.925	Loses in shadow
Dancer2	0.705	0.894	Occassionaly jumps
Bolt2	~	~	Does not track runner
Deer (10 frames)	0.863	0.870	Always fails on 10th frame even though learning

Preliminary results: Dancer2

Overlap: 0.705, mAP: 0.89

Outside Example

M[illegible]ss



Debugging?

- Focus on implementing and testing a variety of integrators
  - Motocross examples shows need to re-initialize tracker

Debugging?

- Focus on implementing and testing a variety of integrators
  - Motocross examples shows need to re-initialize tracker

- Model doesn't track Bolt2
  - Seems like a P-N expert issue in the learning model
  - Tracks other background features, not Bolt

Other Thoughts

- Model tends to fail during transitions in lighting (e.g. Car4 and Human8)

Other Thoughts

- Model tends to fail during transitions in lighting (e.g. Car4 and Human8)
  - Introduce different image warps that also change illumination

Other Thoughts

- Model tends to fail during transitions in lighting (e.g. Car4 and Human8)
  - Introduce different image warps that also change illumination

- Local Binary Pattern Alternative seems to have minimal impact on results

Other Thoughts

- Model tends to fail during transitions in lighting (e.g. Car4 and Human8)
  - Introduce different image warps that also change illumination
- Local Binary Pattern Alternative seems to have minimal impact on results
  - Naive Implementation isn’t much slower than vector implementation for small patches

Other Thoughts

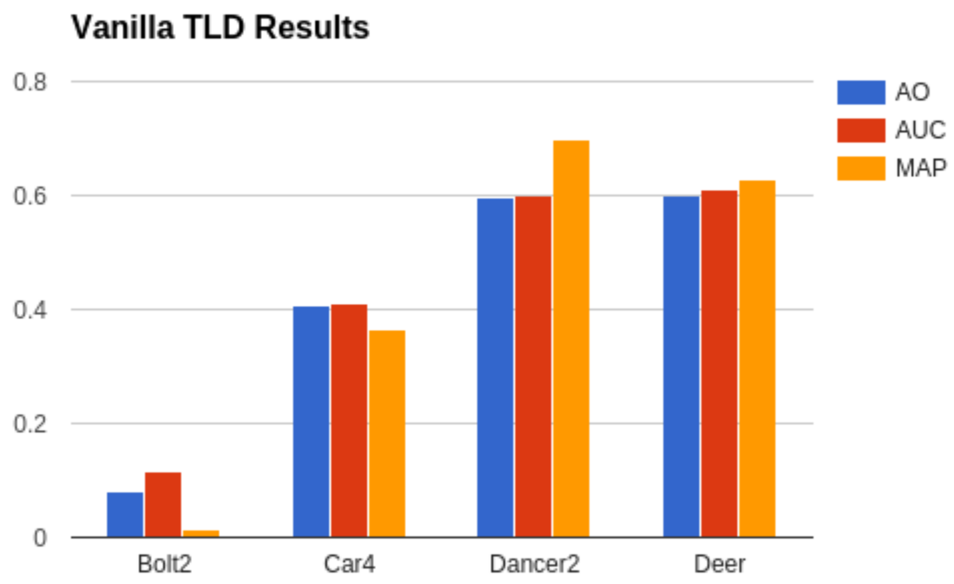
- Model tends to fail during transitions in lighting (e.g. Car4 and Human8)
  - Introduce different image warps that also change illumination

- Local Binary Pattern Alternative seems to have minimal impact on results
  - Naive Implementation isn't much slower than vector implementation for small patches
  - Explore using larger patterns to represent patches more accurately

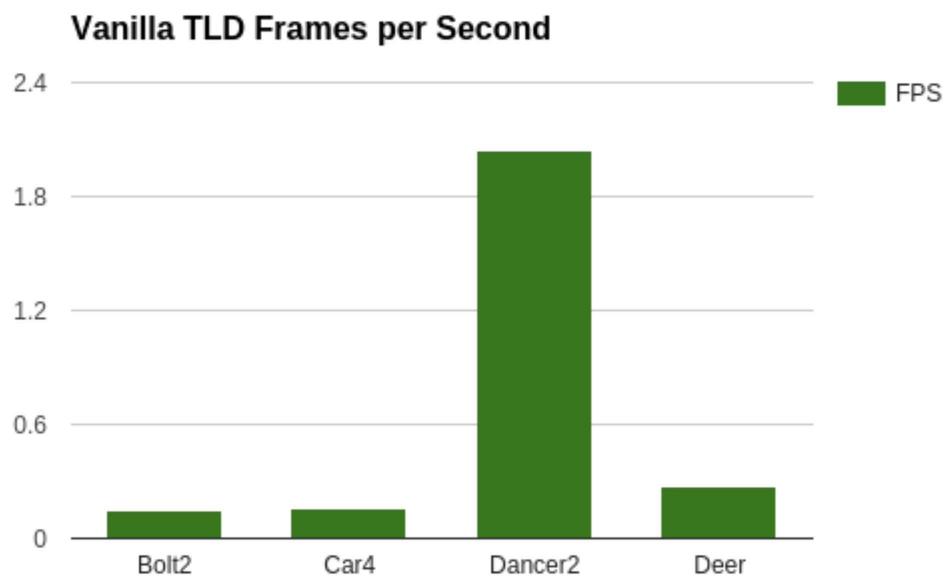
TLD Tracker

Dylan Rhodes

Vanilla Results



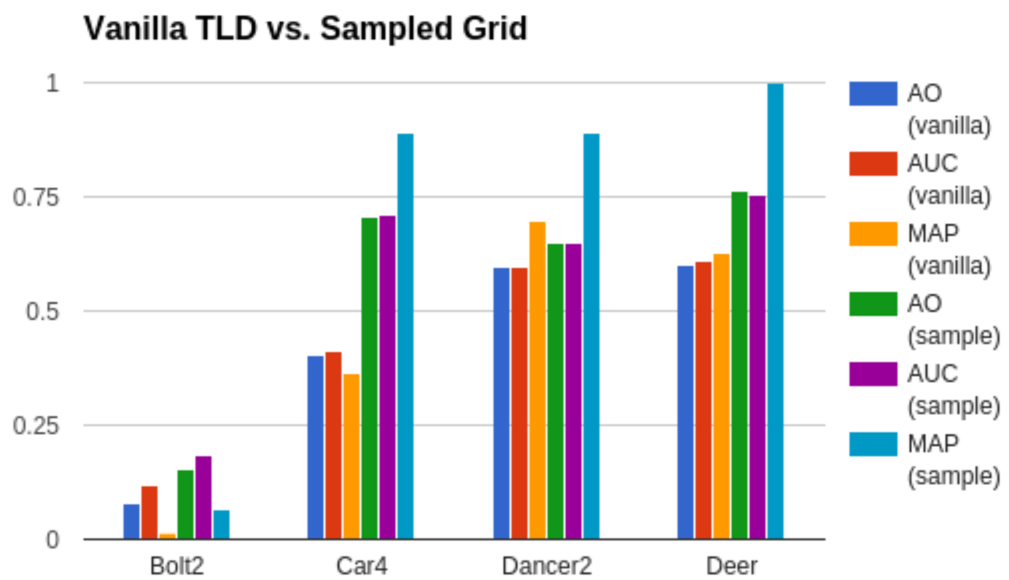
Vanilla Results



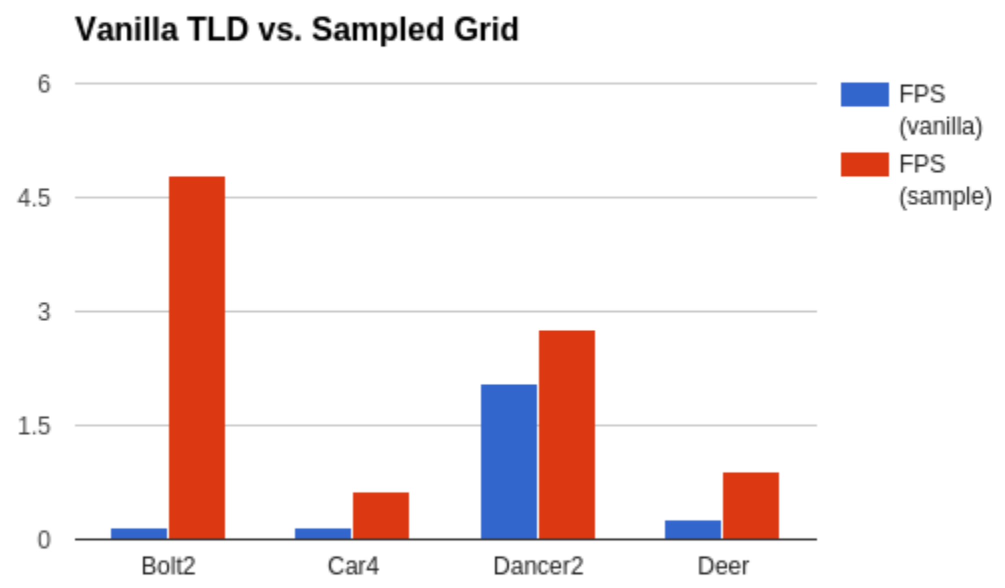
Sampling Motivation

- Trajectories should be fairly stable
  - Boxes which overlap with the current one more will produce a smaller change between frames
  - Subsampling boxes should speed the algorithm
- Sample boxes from grid based on their overlap with the last box and a random coefficient drawn from a gaussian

Sampling Results

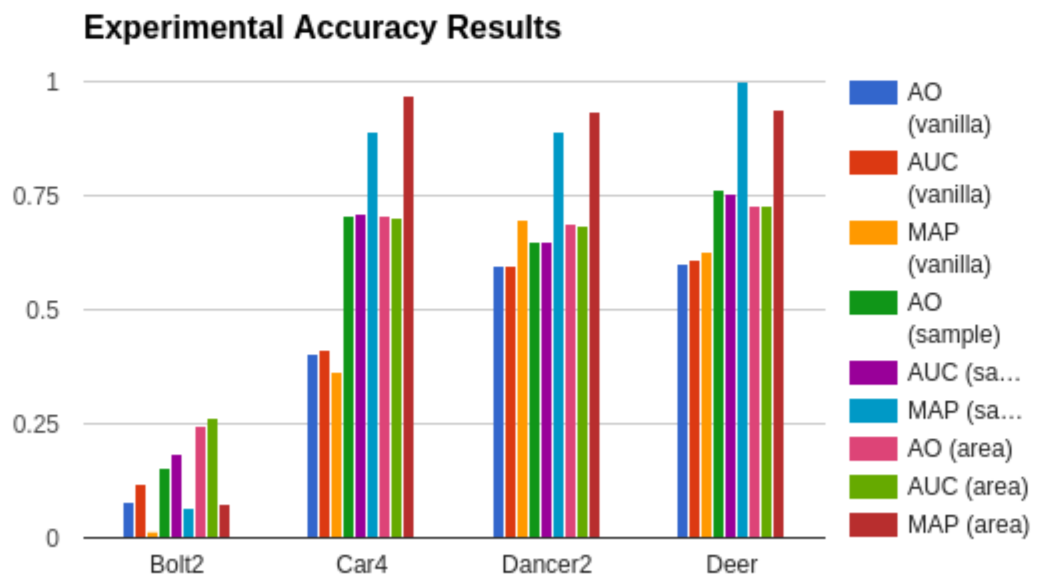


Sampling Results

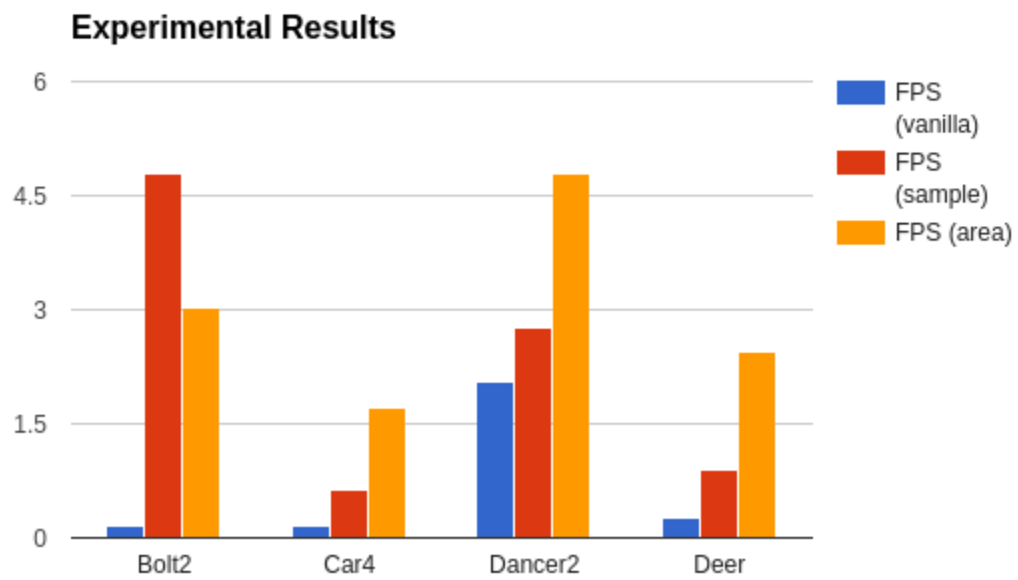


Area Prior Motivation

Area Prior Results



Area Prior Results



Single Object Tracking with TLD, Convolutional Networks and AdaBoost

Albert Haque and Fahim Dalvi

May 11, 2015

Outline

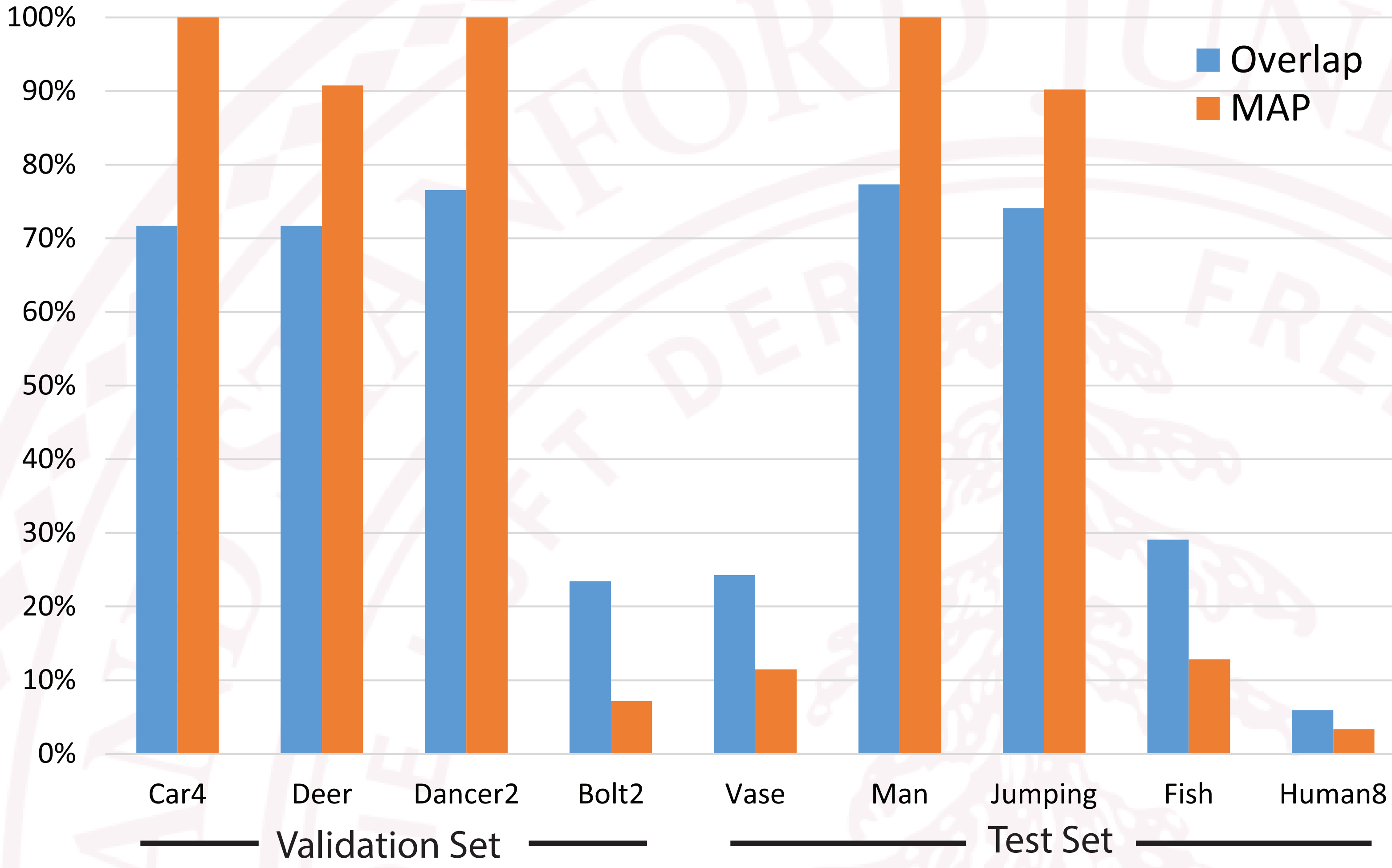
- Patch Features
  - Raw Pixels, HOG, CNN
- Learning Methods
  - SVM, AdaBoost
- Tracker Regularization
- Quantitative Results


Frame: 2
Features: Raw Pixels
Learning Model: SVM


Frame: 2
Features: HOG
Learning Model: SVM


Frame: 2
Features: VGG16-fc7 (4096)
Learning Model: SVM

SVM with Raw Pixels



Convolutional Network Feature Extraction

- VGG-16 architecture using fc7 non-rectified features
- GTX Titan X
- Patches resized to 256x256
- Test time batch size of 200
- Overhead: 2 seconds per frame

Tracking-Learning-Detection with HOG/SVM & RCNN and Spatial Priors

Ranjay Krishna

Extensions & Experiments

1. Pixel Values + SVM
2. HOG features + SVM
3. Selective Search
4. Spatial Prior
   a. Size Delta
   b. Overlap Threshold
5. RCNN Features
6. Generalizing Detections

1: Using Pixels + SVM

	Average Overlap	Success AUC	MAP	Time per Video (s)
Car4	0.58	0.55	0.9	255
Deer	0.64	0.63	0.66	253
Dancer2	0.67	0.67	0.74	196
Bolt2	0.02	0.06	0.01	114
Fish	0.74	0.75	0.77	142
Human8	0.09	0.12	0.06	200
Jumping	0.24	0.27	0.19	176
Man	0.65	0.64	0.98	115
Vase	0.59	0.59	0.55	142

1: Using Pixels + SVM

Example with Deer. Pixels do not capture the face very well and we lose the box for multiple frames when the detector gets confused.

Example of Deer with pixels: link

2: HOG + SVM

	Average Overlap	Success AUC	MAP	Time per Video (s)
Car4	0.65	0.65	0.92	254
Deer	0.66	0.66	0.67	150
Dancer2	0.76	0.77	0.95	176
Bolt2	0.01	0.06	0.01	142
Fish	0.80	0.81	0.88	156
Human8	0.23	0.25	0.19	191
Jumping	0.46	0.46	0.27	188
Man	0.87	0.87	1.00	115
Vase	0.56	0.56	0.63	160

2: HOG + SVM

Performance on Vase video goes down because of the large difference in pixels between the object and the background. So, the pixel features perform really well.

Example of the deer now with HOG: [link]

3. Selective Search

	Average Overlap	Success AUC	MAP	Time per Video (s)
Car4	0.65	0.65	0.92	44
Deer	0.66	0.66	0.67	48
Dancer2	0.76	0.77	0.95	46
Bolt2	0.01	0.05	0.01	42
Fish	0.80	0.81	0.88	90
Human8	0.20	0.21	0.10	90
Jumping	0.42	0.42	0.24	82
Man	0.83	0.87	1.00	15
Vase	0.56	0.56	0.61	53

4. Spatial Priors

1. **Size Delta**
   The object doesn't change in size too much between consecutive frames. So, my integrator checks and rejects boxes that differ in size from the previous detections.

2. **Overlap Threshold**
   Similarly, my integrator checks and only considers detections that have an overlap with previous detections. Prevents detections from jumping around.

Improved results with the Deer: link

5. RCNN Person Detector

Hog Failure on Human8: link

RCNN performs better on Human8: link

Perfect Example with Dancer2: link

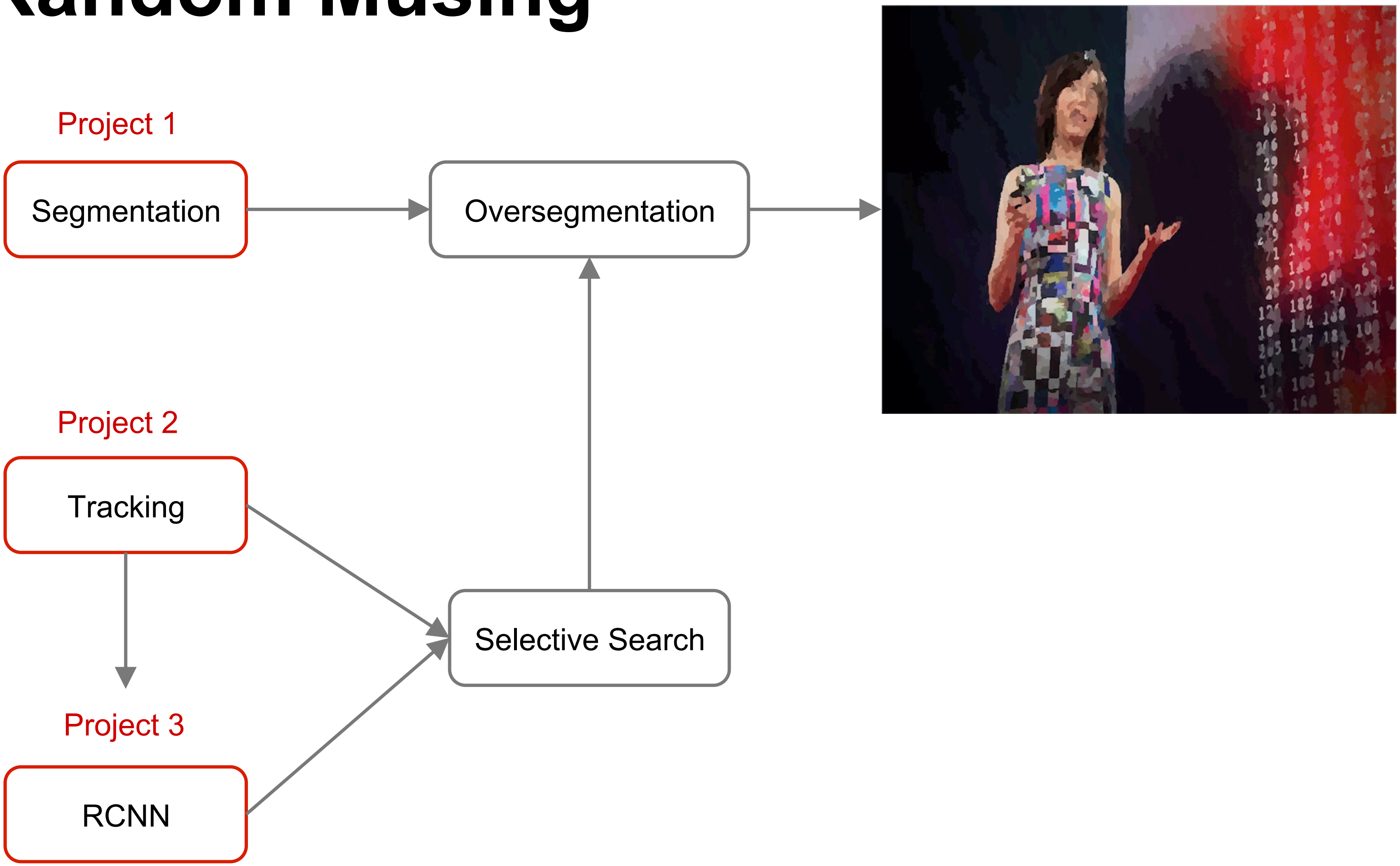
6. Generalizing Detections

What happens if we don’t warp our positive detection examples?

Without warps: [link]

With warps: [link]

Random Musing



Project 2: TLD

Kelsie Zhao

Contents

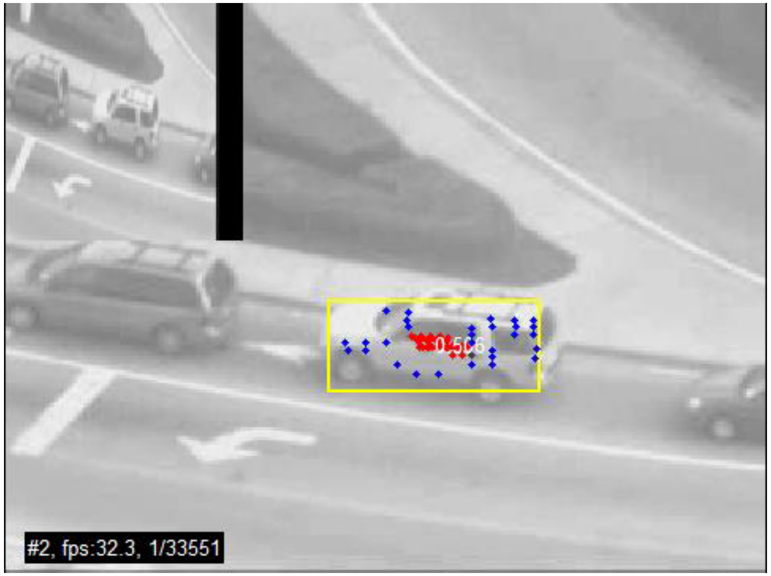
Results

- Fair ones
  - Car4: average-overlap=0.74, map=0.97
  - Dancer2: average-overlap=0.78, map=1.00
  - Fish: average-overlap=0.88, map=1.00

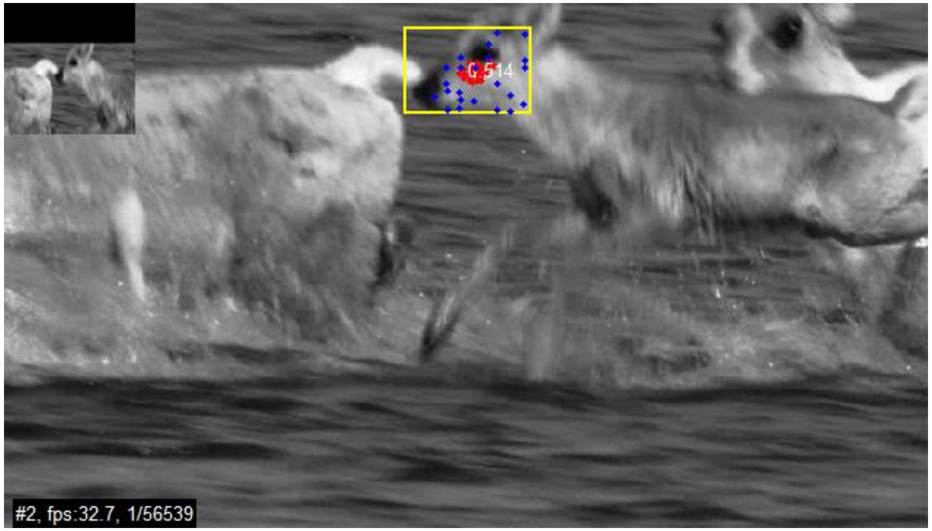
Slow motion, low appearance variance

- Unsatisfactory ones: Human8, Bolt2
  - Bounding box not following

Fast motion or Sudden change of appearance


0.506
#2, fps:32.3, 1/33551


0.509
#2, fps:34.6, 1/6841


0.514
#2, fps:32.7, 1/56539


0.507
#2, fps:36, 1/22490

Some Problems

- NN classifier produces probability always around 0.5
  - For positives, 0.5043; for negatives, 0.4902
- Cannot handle fast motion
  - Scan a larger region vs Speed
- Cannot handle fast illumination variances
  - Fern might not handle uneven illumination change

Extensions:

- Detection Strategy
  - Run Classifiers on bounding boxes within a region of the last bounding box
- Priori for detection
  - Penalize the confidences of the detected bounding boxes which experienced a sudden change in bounding box size.
- HOG & SVM
  - Use HOG feature and SVM in place of Fern + NN

Thank You!
Q&A

TLD tracking project

Meng Wu

Main components

- KL tracker -> direction
- SVM classifier -> robustness
- NN classifier -> confirmation
- Integrator
  1. SVM rejects wrong detections
  2. score = NN conf + SVM score + overlap * KL conf

Parameter setting

- Patch size: 24 x 24
- Pattern size: 24 x 24
- SVM:
  - Linear kernel without auto-scale
  - Average 50 – 80 supporting vectors

- 200 Positive/Negative Examples
  - Always keep the original examples
  - Keep positive examples most away from negative examples
  - Randomly replace 100 with new negative examples

Results

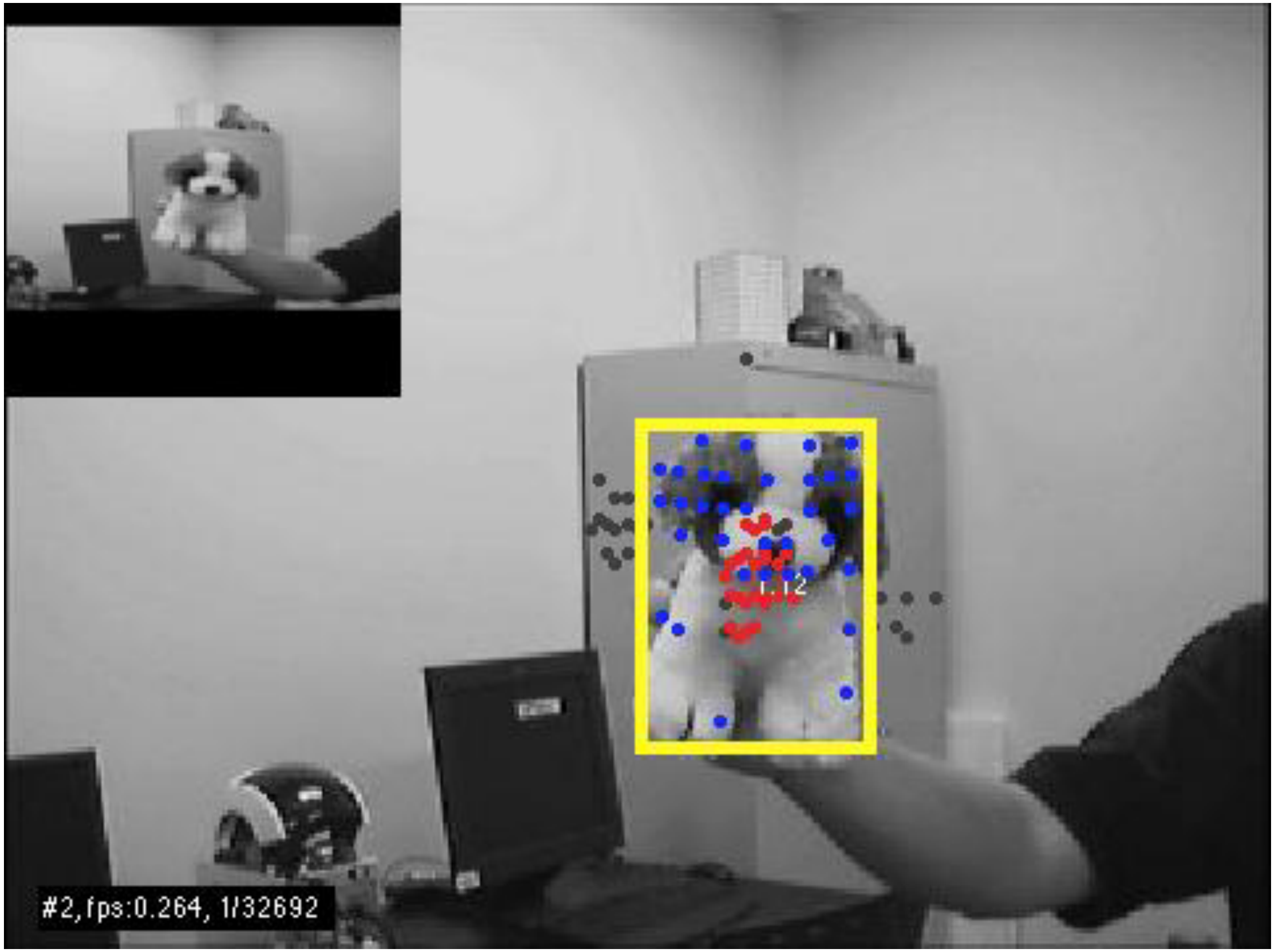
	Bolt2	Car4	Deer	Dancer2	Fish
Average overlap	0.601	0.712	0.690	0.764	0.668
Average precision	0.602	0.712	0.686	0.761	0.667
Map	0.765	0.752	0.87	1.00	0.913
Frame rate	0.46	0.35	0.35	0.16	0.23

Because adding the SVM scores, the average precision bad.

Some observations

- Important to keep the original positive examples
- Resizing patches takes most of time
- Reduce number of bonding boxes
  - Search in the neighborhood
  - Similar sizes
- Only update the NN datasets when you are sure


#2,fps:0.348, 1/41534


#2,fps:0.264, 1/32692

TLD
Implementation and Evaluation

Lyne P. Tchapmi
Stanford University/CS231B

Building Blocks

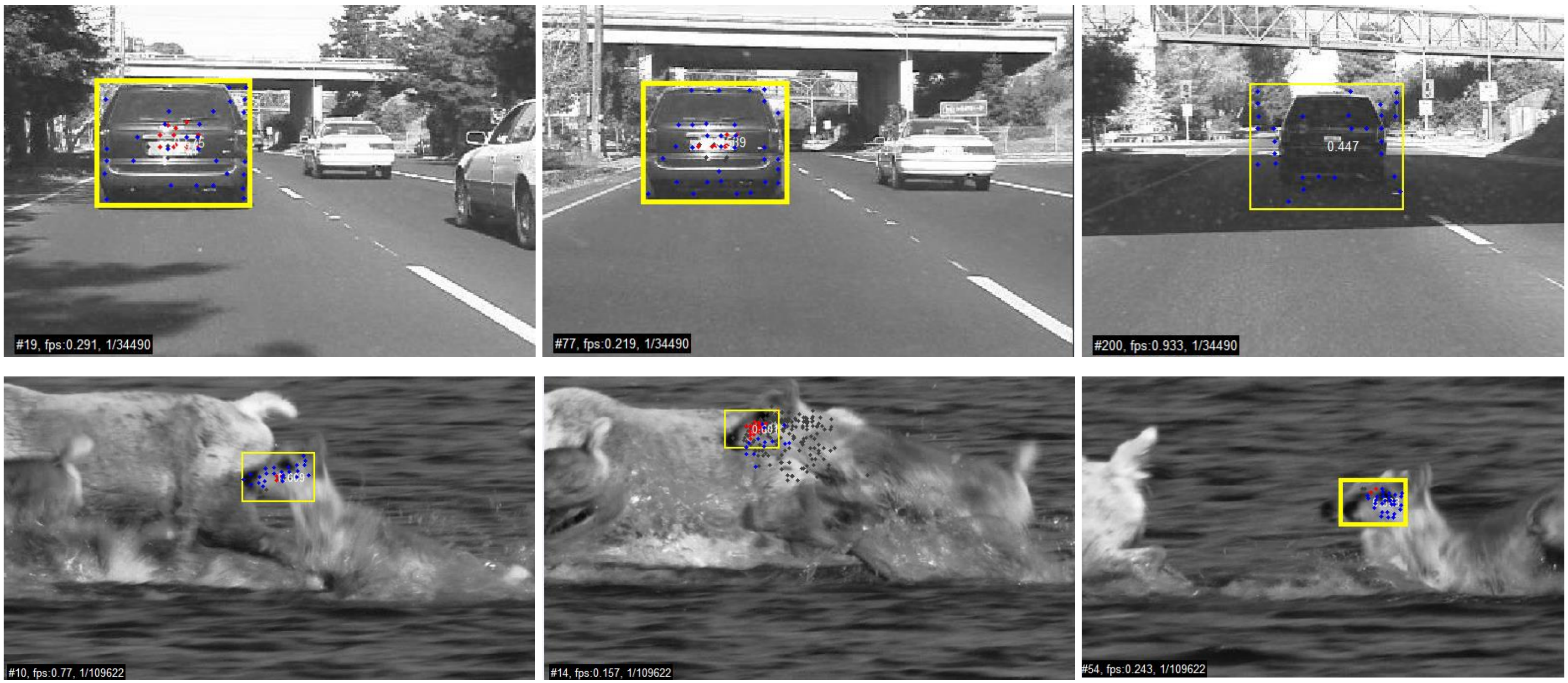
- Classifier
  - FERN
  - SVM

- Features
  - ZMUV
  - BRIEF-16
  - BRIEF-32
- Integrator
- Pattern generator

Evaluation

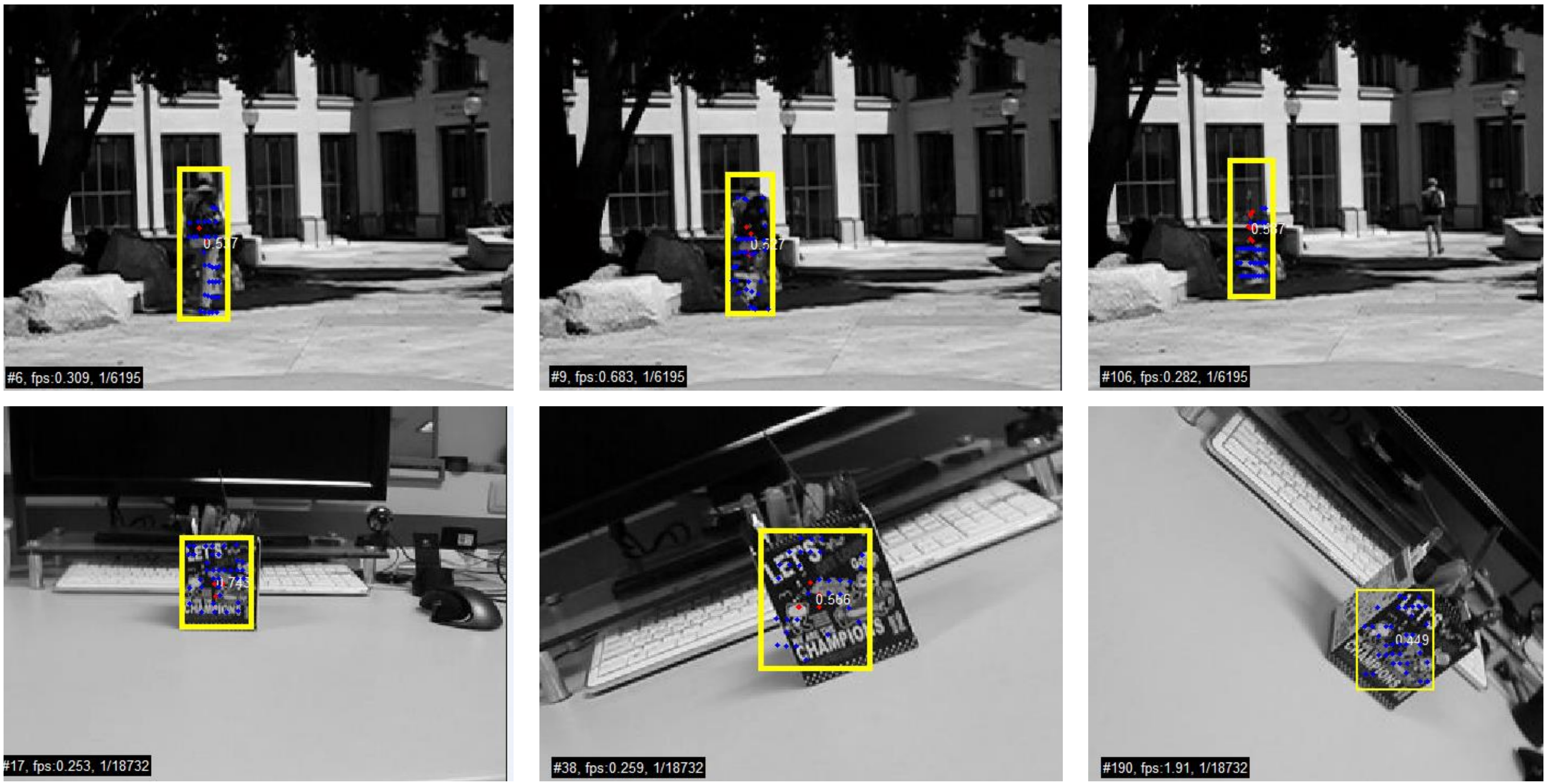
Sequence	SVM+BRIEF-32	SVM+BRIEF-16	SVM+ZMUV	FERN+BRIEF-16	FERN+BRIEF-32	FERN+ZMUV
Bolt2	0.01/0.01	**0.02/0.01**	0.01/0.00	**0.02/0.01**	0.02/0.00	0.01/0.00
Car4	0.63/0.67	0.59/0.56	0.76/1.00	0.71/1.00	**0.79/1.00**	0.79/0.98
Dancer2	0.66/0.92	0.63/0.92	0.58/0.71	**0.70/1.00**	0.63/0.97	0.62/0.75
Deer	0.29/0.10	0.04/0.03	0.60/0.82	0.11/0.03	0.17/0.06	**0.68/0.91**
Fish	0.57/0.42	0.58/0.77	**0.83/1.00**	0.49/0.23	0.44/0.11	0.74/1.00
Human8	0.13/0.10	0.15/0.03	0.06/0.01	0.12/0.02	0.10/0.06	0.09/0.02
Jumping	0.12/0.02	0.17/0.04	**0.23/0.15**	**0.27/0.13**	0.10/0.07	**0.23/0.15**
Man	0.85/1.00	**0.87/1.00**	0.80/1.00	0.45/0.07	0.53/0.18	0.67/1.00
Vase	0.34/0.11	**0.54/0.33**	0.48/0.25	0.44/0.22	0.32/0.07	0.51/0.31

FERN+ZMUV



#19, fps:0.291, 1/34490
#77, fps:0.219, 1/34490
0.447
#200, fps:0.933, 1/34490
#10, fps:0.77, 1/109622
#14, fps:0.157, 1/109622
#54, fps:0.243, 1/109622

FERN+ZMUV


#6, fps:0.309, 1/6195
#9, fps:0.683, 1/6195
#106, fps:0.282, 1/6195
#17, fps:0.253, 1/18732
0.566
CHAMPIONS
#38, fps:0.259, 1/18732
0.449
#190, fps:1.91, 1/18732