

---

# Tracking by learning

Arnold W.M. Smeulders with Dung Chu

for SINT with Ran Tao and Efstratis Gavves

# Tracking

---

Online tracking is to *determine the location of one target in video* starting from a bounding box in the first frame.

When conceived as an instant learning problem, the task is to discriminate object from background on the basis of  $N=1$  sample (in the first frame) and  $N=k$  samples more (as long as the tracking is successful over  $k+1$  frames).





So it is a hard and complex machine learning problem.

# Tracking

---

Online tracking is to *determine the location of one target in video* starting from a bounding box in the first frame.





They consist of:

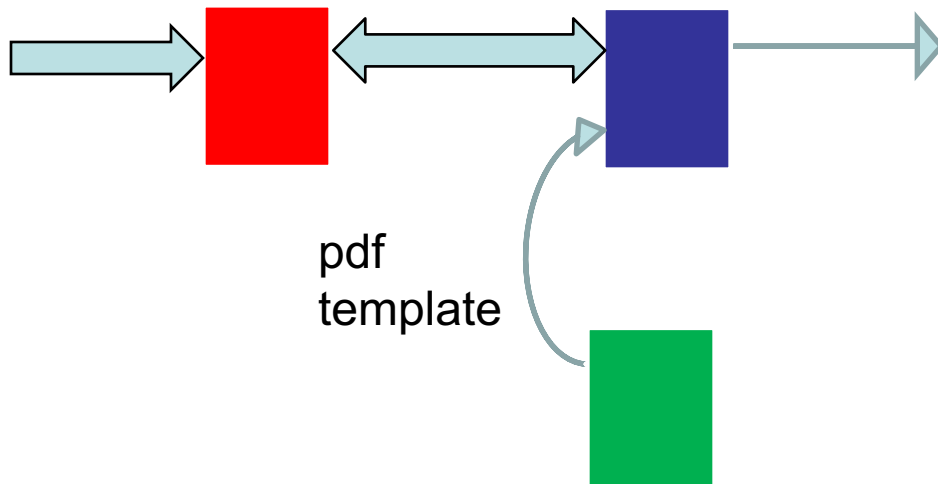
-  a module observing the features of the image.
-  a module selecting the actual motion.
-  a module holding the internal representation of the object.
-  a module updating the representation of the object.

# The *simplest* tracker: NCC

---

The oldest and still good(!) non-discriminative tracker.





-  Intensity values in the candidate box.
-  Direct target matching by Normalized Cross-Correlation.
-  Intensity values in the initial target box as template.
-  No updating of the target.

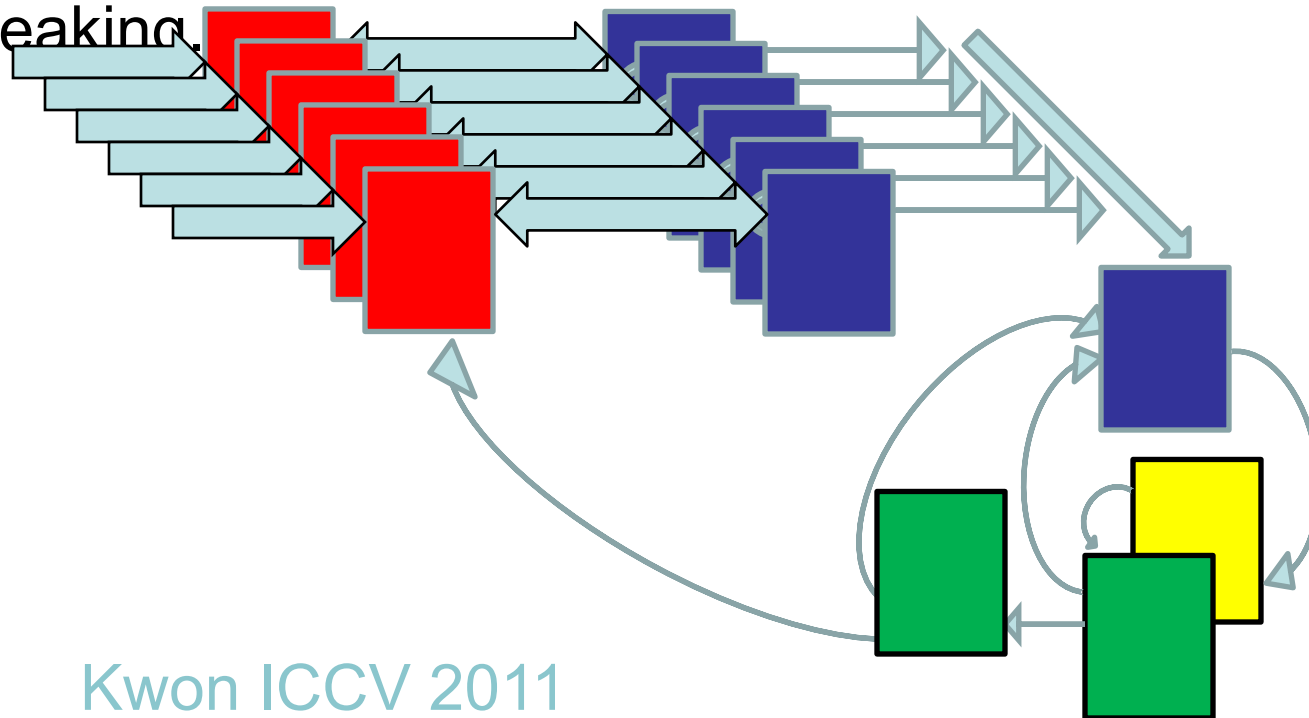




# A good tracker: TST

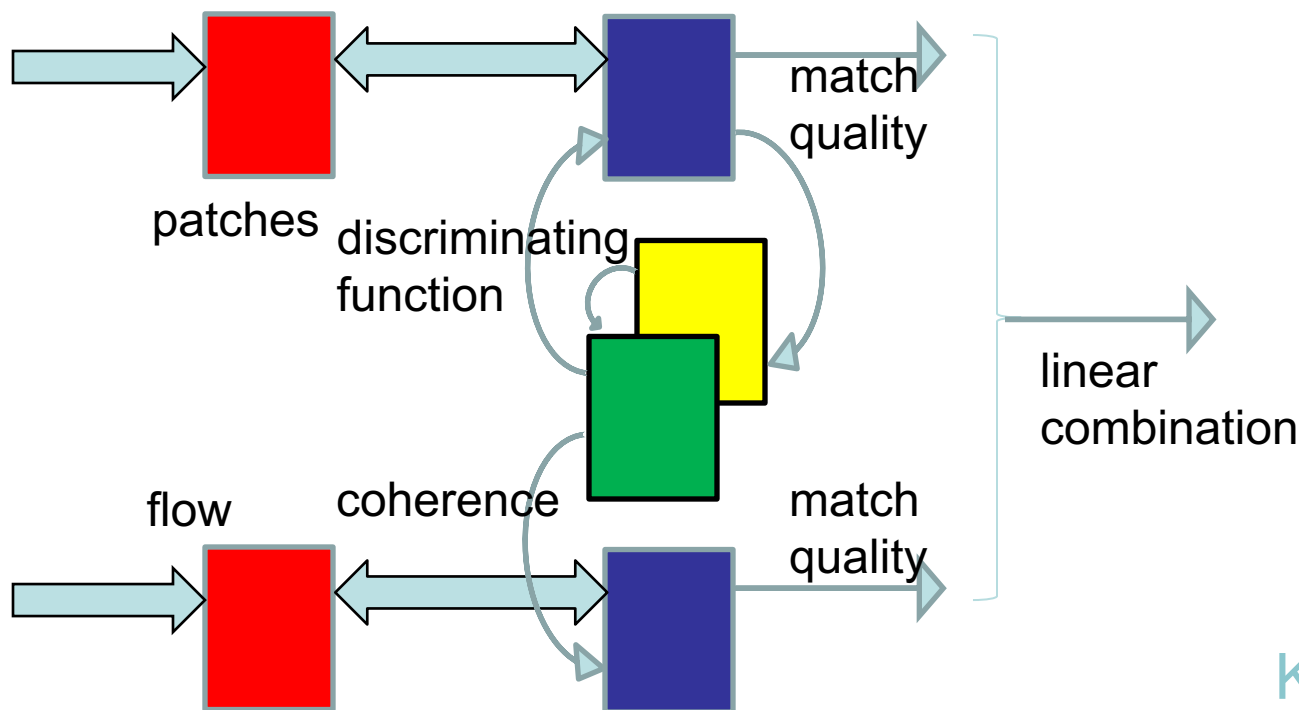
Tracking by Sampling Trackers is the best non-discriminative.

-  HIS-color edges of many different trackers.
-  Best match in image, followed by best state.
-  Trackers store eigen images. State stores  $x$ ,  $s$ , score.
-  Sparse incremental PCA image representation with leaking.



# A good tracker: TLD

- Optic flow patches + Intensity patches.
- Discriminant on median flow + Normalized Cross Correlation.
- Weights of the classifier + Template of target.
- Experts label update + Recovery when lost.



# Discriminative trackers

---

In discriminative trackers, the emphasis is on learning the current distinction between object and background.

We discuss the first: the Foreground – Background tracker.

# Discriminative Trackers

---

Minor viewpoint change



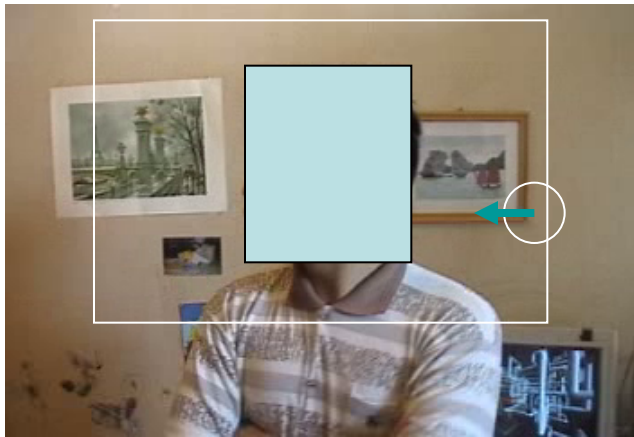
Severe viewpoint change



# Discriminative Trackers

---

The hole in the background leaves the appearance of the object entirely free: The object may change abruptly in pose.



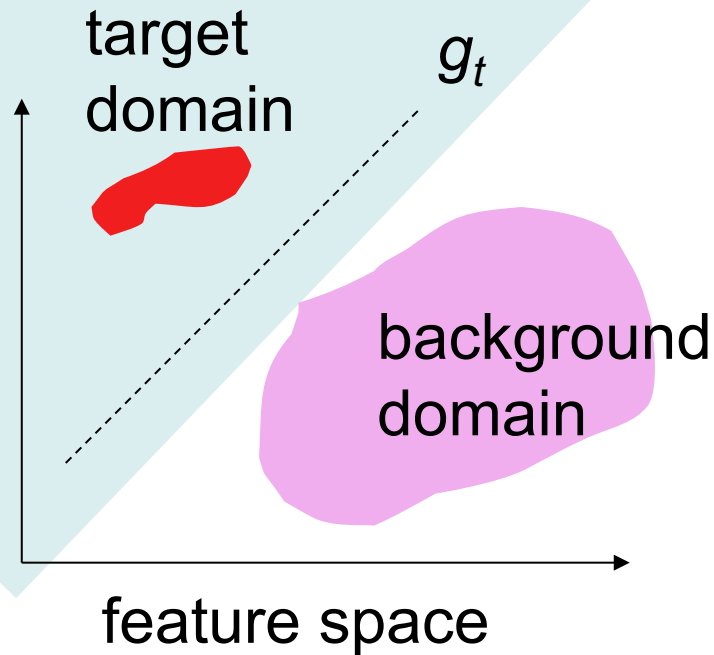
The background varies slower:  
Background is better predictable.

General scheme: Get foreground and background patches  
+ Learn a classifier + Classify patches from new image.

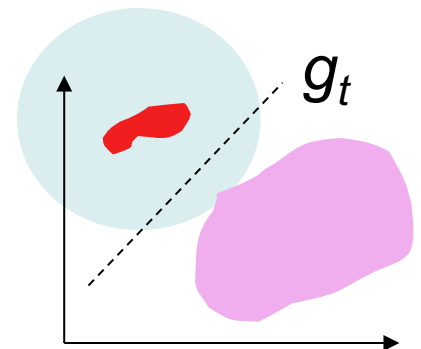
# Discriminative Tracker: FBT

---

Dynamic discrimination of the object from its background while maximizing the discriminant score of the target region.







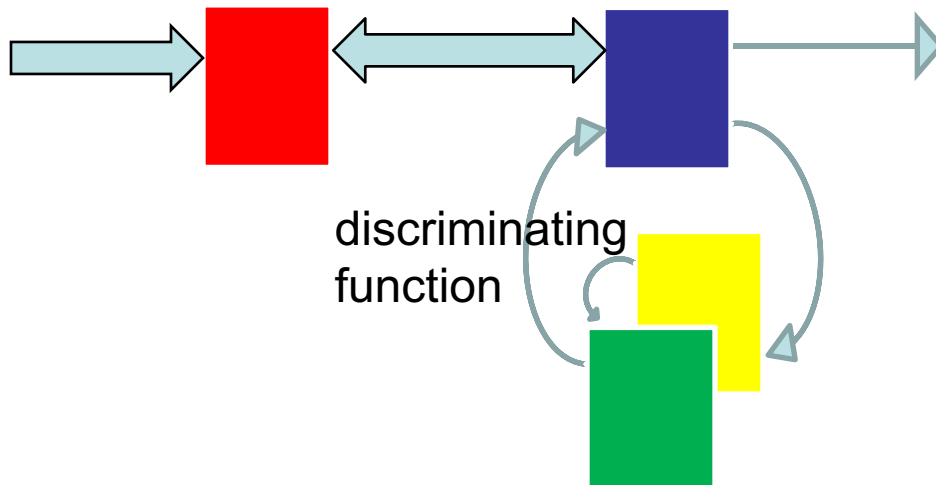
Or even better:



# Discriminative Tracker: FBT

---

-  SURF texture samples from target / background box.
-  Trains a linear discriminant classifier.
-  Classifier is foreground/background model (in feature space).
-  Updated by a leaking memory on the training data.



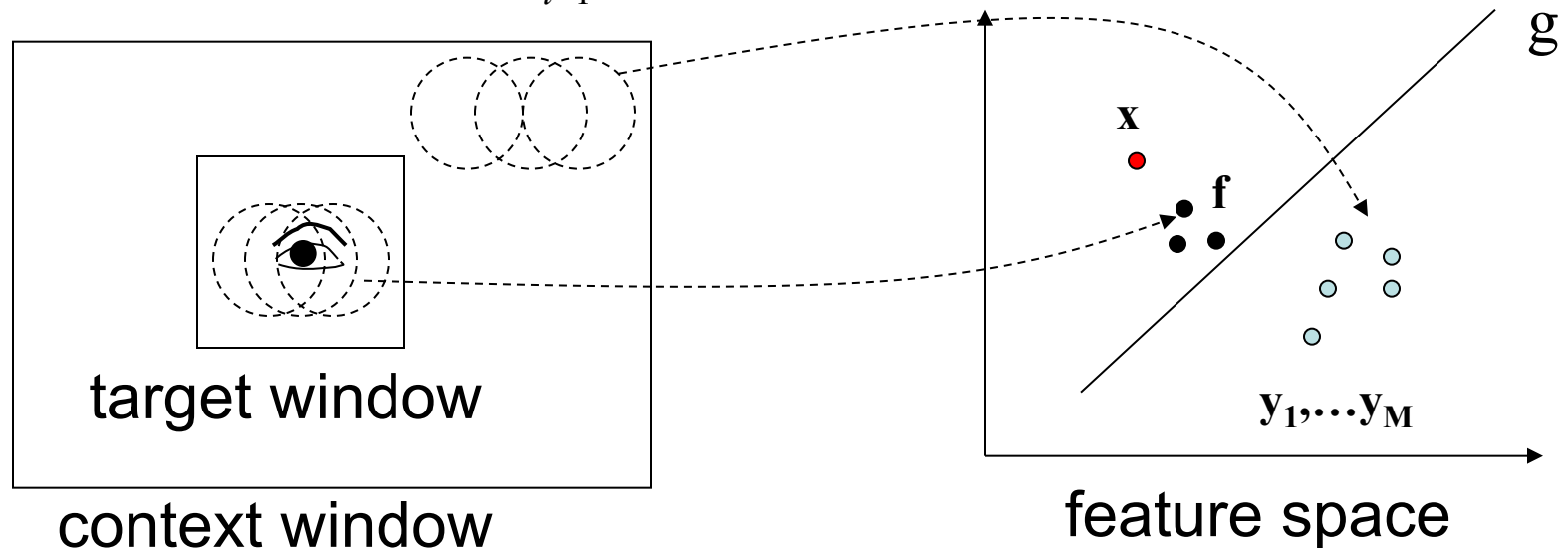
# Foreground-Background Classifier

Discriminant function

$$g(\mathbf{f}) = \mathbf{a} \cdot \mathbf{f} + b \rightarrow \max_{\text{target location}}$$

Train  $g$  by adopting linear discriminant analysis:

$$[g(\mathbf{x}) - 1]^2 + \sum_{i=1}^M \alpha_i [g(\mathbf{y}_i) + 1]^2 + \lambda \frac{|\mathbf{a}|^2}{2} \rightarrow \min_{\mathbf{a}, b}$$





# Foreground-Background Classifier

---

The solution is obtained in closed, incremental form:

$$\mathbf{a} \propto [\lambda \mathbf{I} + \mathbf{B}]^{-1} [\mathbf{x} - \bar{\mathbf{y}}]$$

The weighted mean vector of background patterns:

$$\bar{\mathbf{y}} = \sum_{i=1}^M \alpha_i \mathbf{y}_i$$

The weighted covariance matrix:

$$\mathbf{B} = \sum_{i=1}^M \alpha_i [\mathbf{y}_i - \bar{\mathbf{y}}][\mathbf{y}_i - \bar{\mathbf{y}}]^T$$

Mean and covariance can be updated incrementally.

# Foreground-Background Updating

---

The foreground template is updated in every frame:

$$\mathbf{x} = (1 - \gamma)\mathbf{x}_{prev} + \gamma\mathbf{f}_{optimal}$$

New patterns are added to the background patterns.

Background patterns are summed with leaking coefficients  $\alpha_i$ .

New and old patterns predict mean  $\mathbf{y}$  and cov  $\mathbf{B}$  incrementally.

# FBT Results

---







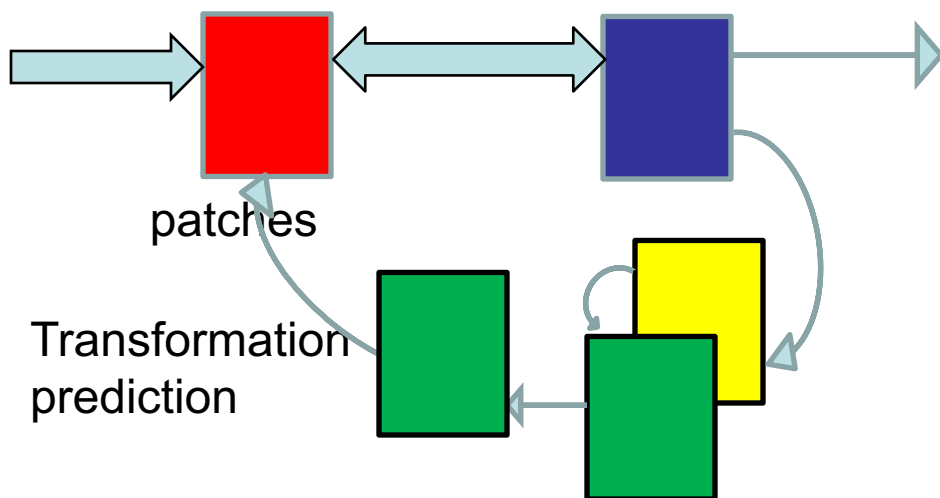
# Structured SVM Tracker

---

# STRuctured output tracking

---

-  Windows by Haar features with 2 scales.
-  Structured SVM by {appearance, translation}, no labels.
-  Structured constraints + Transformation prediction.
-  Update the constraints to stay at current  $\mathbf{x}$ .



# STRuctured output tracking

In STR the output of the classifier is the transform directly:  
 what is the effort to go from  $\mathbf{x}$  to  $\mathbf{y}$  ( $\Delta \mathbf{x}$ ,  $\Delta$  appearance,  $\Delta \dots$ )?  
 The objective function with a (non)linear kernel  $\Phi(\mathbf{x}, \mathbf{y})$  is:

SVM weights

SVM slack

$$\min_{\mathbf{w}} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i$$

$$\text{s.t.} \quad \forall i : \xi_i \geq 0$$

$$\forall i, \forall \mathbf{y} \neq \mathbf{y}_i : \langle \mathbf{w}, \delta \Phi_i(\mathbf{y}) \rangle \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i$$

Increase in transform

Loss: 0 when  $\mathbf{y}$  are equal

Can be rewritten into the online version:

$$\max_{\beta} \quad - \sum_{i, \mathbf{y}} \Delta(\mathbf{y}, \mathbf{y}_i) \beta_i^{\mathbf{y}} - \frac{1}{2} \sum_{i, \mathbf{y}, j, \bar{\mathbf{y}}} \beta_i^{\mathbf{y}} \beta_j^{\bar{\mathbf{y}}} \langle \Phi(\mathbf{x}_i, \mathbf{y}), \Phi(\mathbf{x}_j, \bar{\mathbf{y}}) \rangle$$

$$\text{s.t.} \quad \forall i, \forall \mathbf{y} : \beta_i^{\mathbf{y}} \leq \delta(\mathbf{y}, \mathbf{y}_i) C$$

$$\forall i : \sum_{\mathbf{y}} \beta_i^{\mathbf{y}} = 0$$

# STRuctured updating

---

The loss function is based on the overlap score:

$$\Delta(\mathbf{y}, \bar{\mathbf{y}}) = 1 - s_{\mathbf{p}_t}^o(\mathbf{y}, \bar{\mathbf{y}}),$$

Updating is by inserting the displacement as a positive support vector and the hardest loss as a negative.

Older support vectors are removed at random when their loss functions shows too big a deviation.

Existing support vectors are reprocessed to update their weights given the current state.

# Experimental results 2014

---

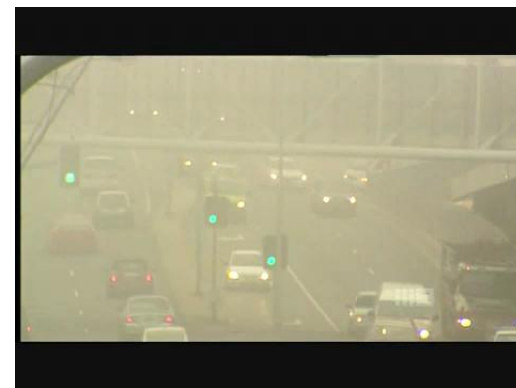
ALOV300++ dataset

Smeulders Dung et al PAMI 2014



# 13 Hard Cases for Tracking

---

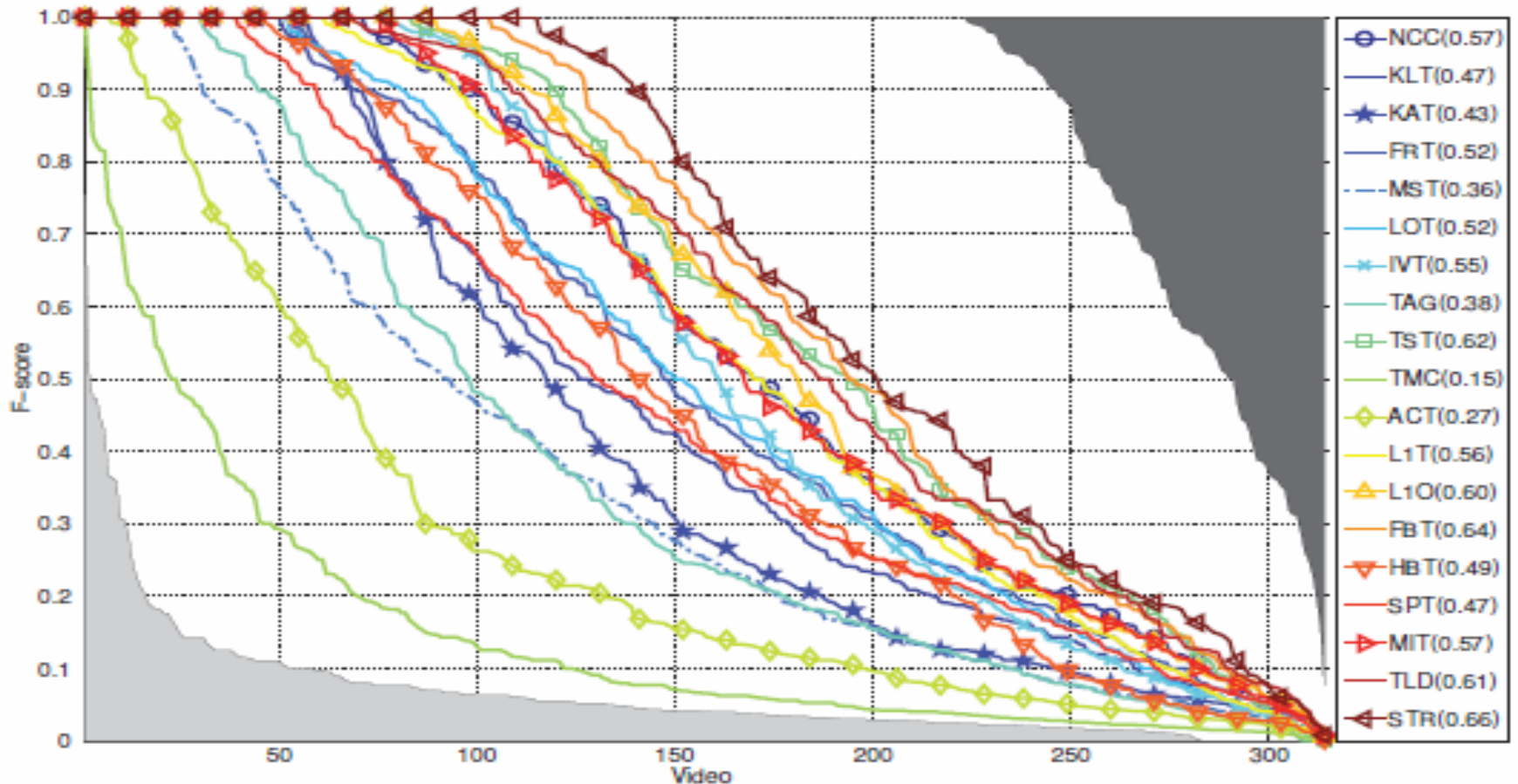


# 19 Assorted Trackers

---

1.	Normalised cross correlation	NCC	1970?
2.	Lucas Kanade tracker	LKT	1984
3.	Kalman appearance prediction tracker	KAT	2004
4.	Fragments-based tracker	FRT	2006
5.	Mean shift tracker	MST	2000
6.	Locally orderless tracker	LOT	2012
7.	Incremental visual tracker	IVT	2008
8.	Tracking on the affine group	TAG	2009
9.	Tracking by sampling trackers	TST	2011
10.	Tracking by Monte Carlo sampling	TMC	2009
11.	Adaptive Coupled-layer Tracking	ACT	2011
12.	L1-minimization Tracker	L1T	2009
13.	L1-minimization with occlusion	L1O	2011
14.	Foreground background tracker	FBT	2006
15.	Hough-based tracking	HBT	2011
16.	Super pixel tracking	SPT	2011
17.	Multiple instance learning tracking	MIT	2009
18.	Tracking, learning and detection	TLD	2010
19.	Structured output tracking	STR	2011

# Survival curves by 2014



STR (.66), FBT (.64), TLD (.62), TST (.61), all very different.

# Very hard

---



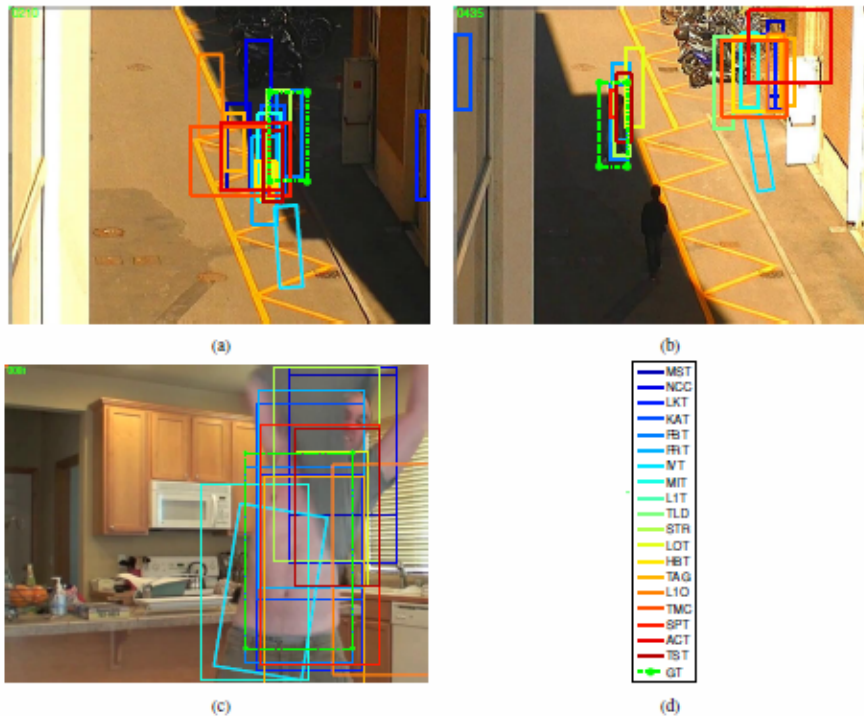


# On shadows

---

The effect of shadows.

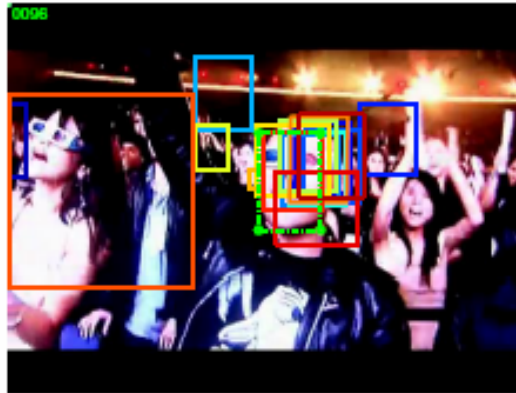
Heavy shadow has an impact almost for all.



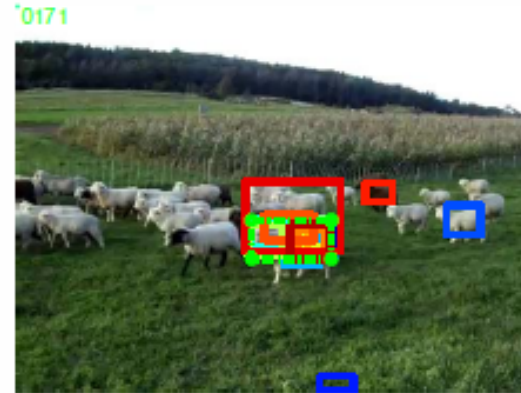
FBT (.73) performs best.

# On clutter

---



(a)



(b)



(c)

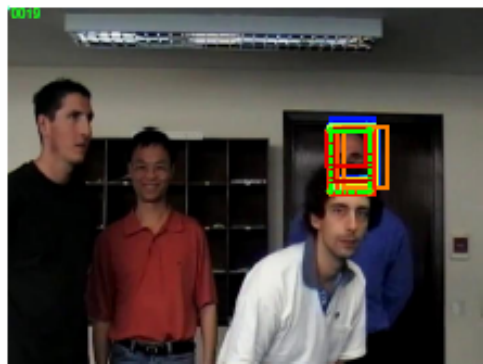


(d)

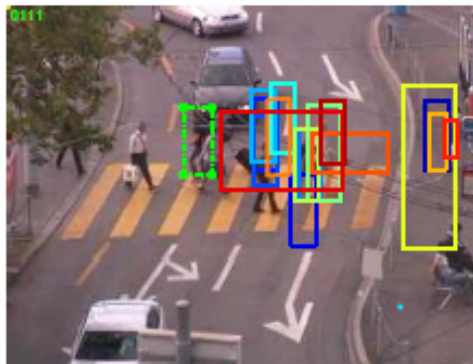
Success is better than expected even if very hard.

# On occlusion

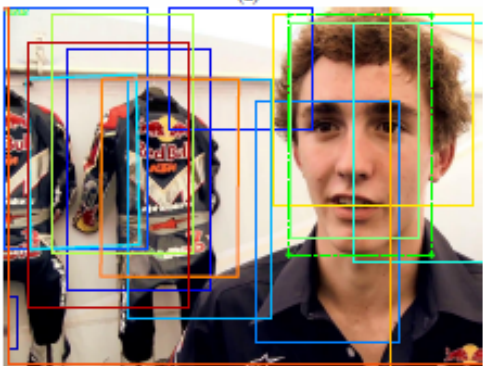
---



(a)



(b)



(c)

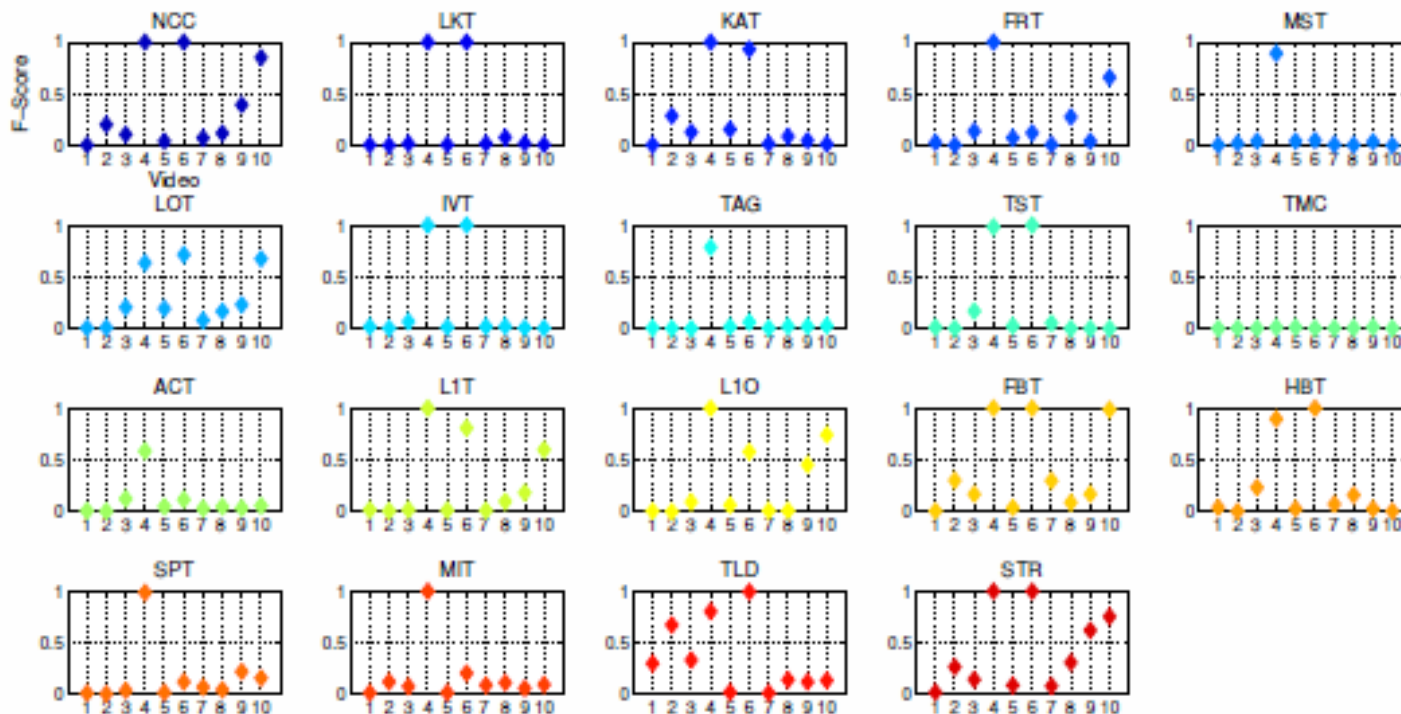


(d)

STR, FBT, and TLD are best here.  
Light occlusion is approximately solved.  
Full occlusion is still hard for most.

# On long videos

The F-score on ten 1 – 2 minute videos



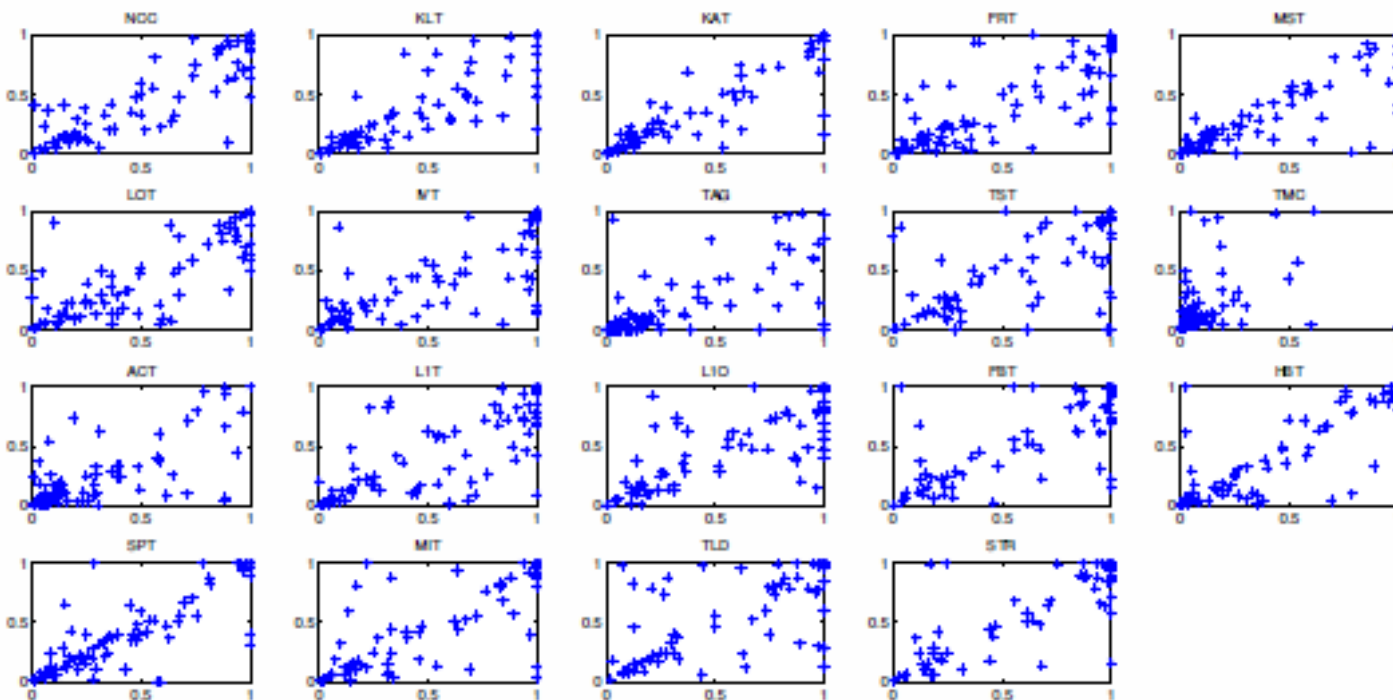
STR, FBT, NCC (no updating!), TLD perform well (!).  
TLD excels in sequence 1 which is hard.



# On stability of the initial box

---

F-scores of 20% right shift (y-axis) vs original (x-axis)



Overall loss of .05 %.

STR has a small loss.

# Outstanding results by Grubs

TABLE III: The list of outstanding cases resulted from the Grubbs' outlier test and with  $F \geq 0.5$ .

Sequence	Tracker	Sequence	Tracker	Sequence	Tracker	Sequence	Tracker
0112	TLD	0411	ACT	1102	TLD	1203	MIT
0115	STR	0510	L1T	1103	HBT	1206	STR
0116	KAT	0512	STR	1104	TLD	1210	TLD
0122	TLD	0601	STR	1107	HBT	1217	TLD
0203	FBT	0611	MST	1112	STR	1218	TLD
0301	L1T	0705	TLD	1116	TLD	1221	TLD
0305	L1T	0901	HBT	1119	TLD	1303	TLD
0312	L1T	0916	STR	1128	TLD	1402	TLD
0314	KAT	0925	STR	1129	FBT	1409	STR
0404	FBT	1020	FBT	1134	FRT		

Many excel in 1 video. (Favorable selection.)

TLD excels in camera motion, occlusion.

FBT in target appearance, light.



0916 STR



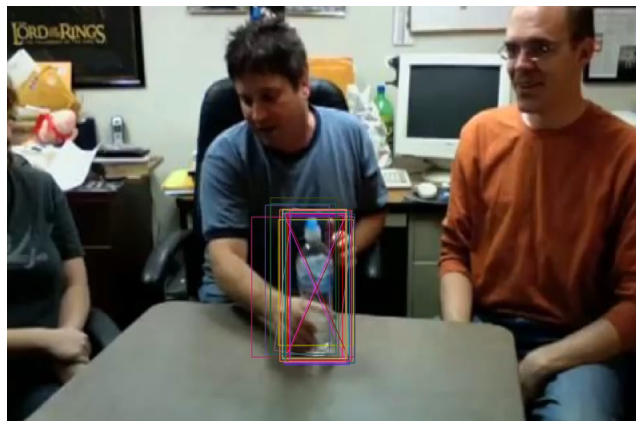
0601 STR



1107 SPT HBT



1129 FBT > FRT



0404 FBT



1402 TLD

# Siamese trackers

---

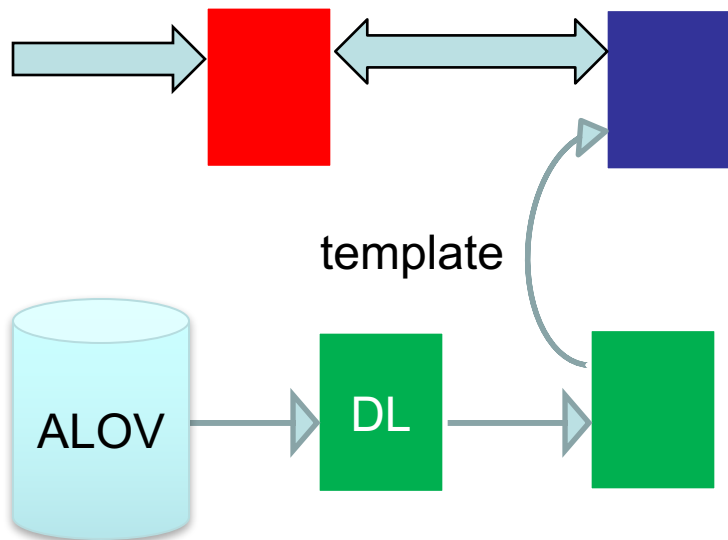
Tao, Gavves, Smeulders CVPR 2016

# Siamese instance search tracking

---

Observation: updating ruins most trackers. By learning the *general* variance in appearance *off-line*, can we avoid updating?

In other words, can we avoid the temporal aspect?



Yes we can:

Reinstall this primitive NCC-scheme.  
+ Deep learning general variance.

# Siamese instance search tracking

Yes compare with original  
No update online  
No geometric matching  
No combination of trackers  
No motion model



Initial patch



candidate  
patches



$m(\text{candidates}, \text{original})$

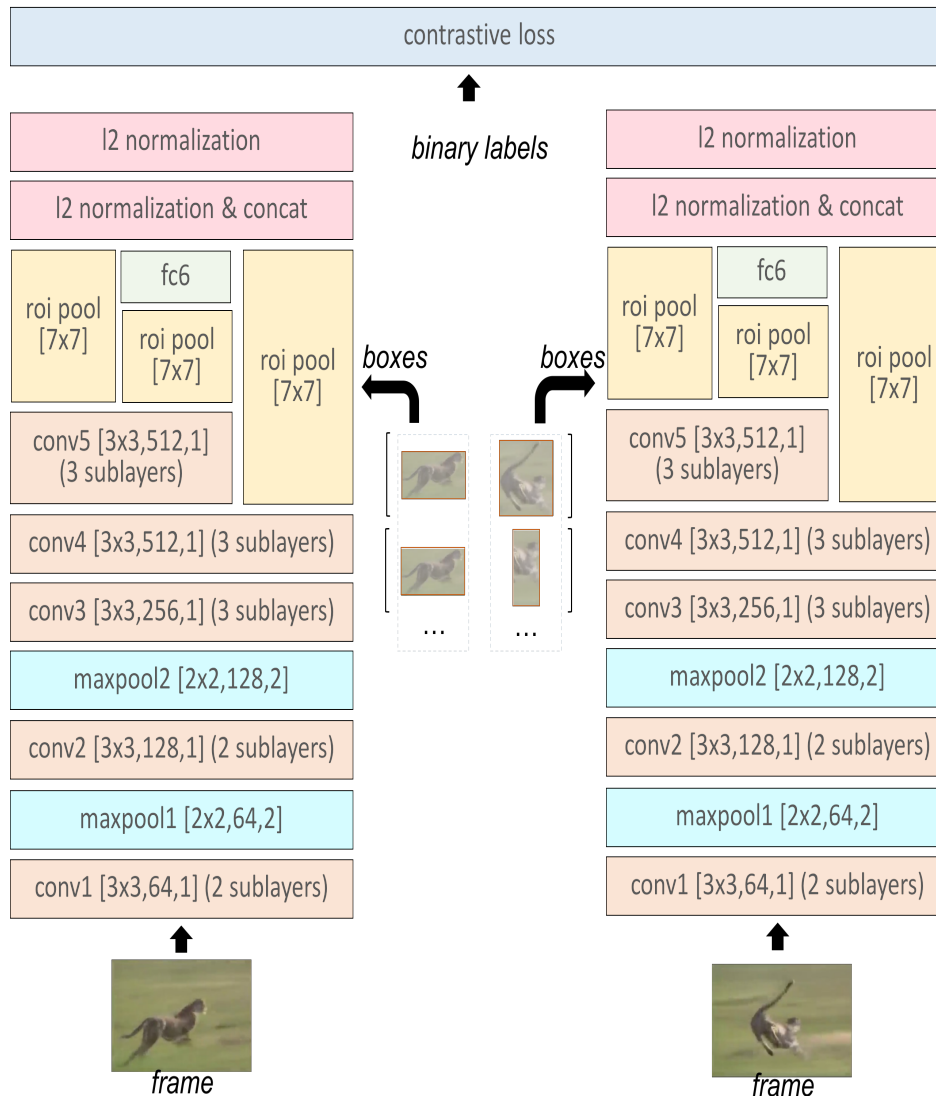


predicted  
patch



Learn the general variations  $m(\cdot)$  of object appearances offline in a Siamese network on pairs of examples.

# Siamese learning $m(.)$



Contrastive loss.

Use outputs of multiple layers to be robust in many situations.

Few max pooling to improve localization accuracy.

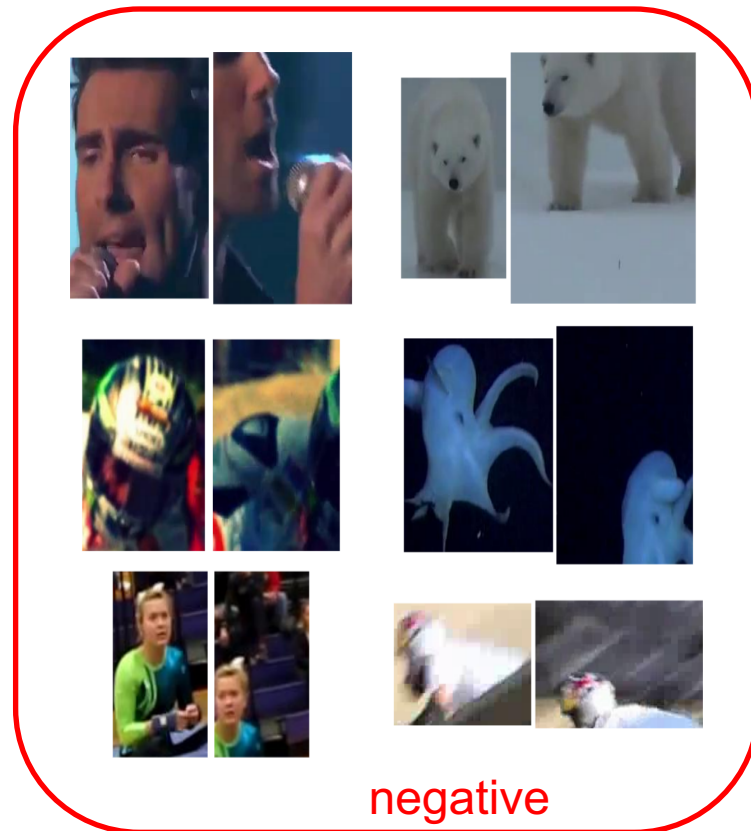
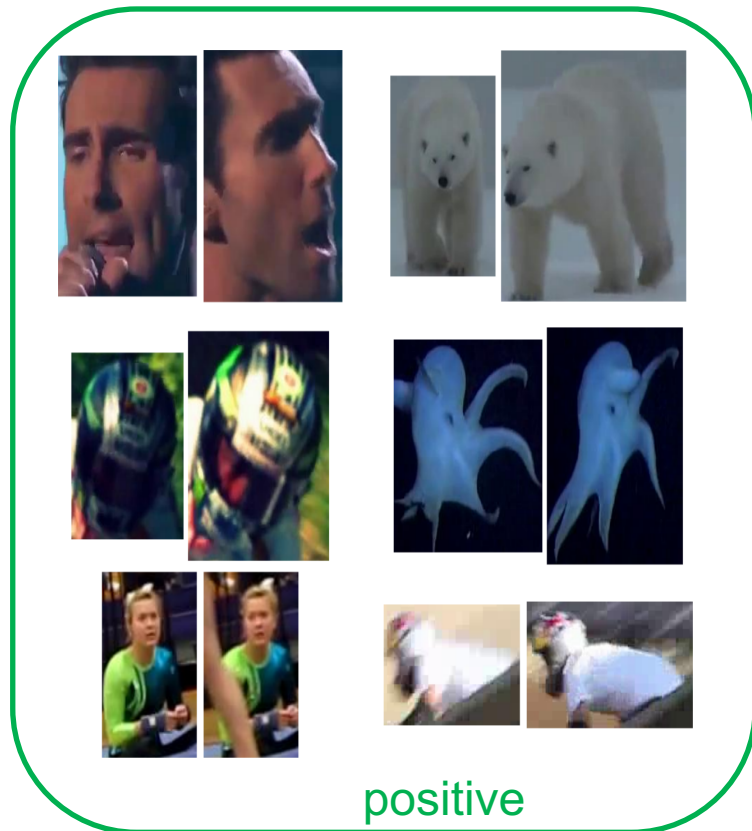
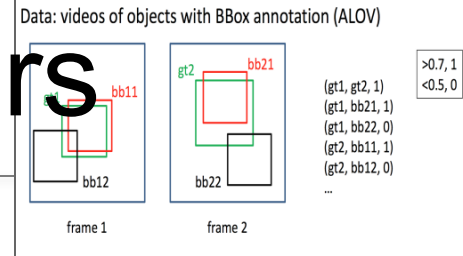
Pretrained ImageNet.

Insert correct / incorrect pairs.



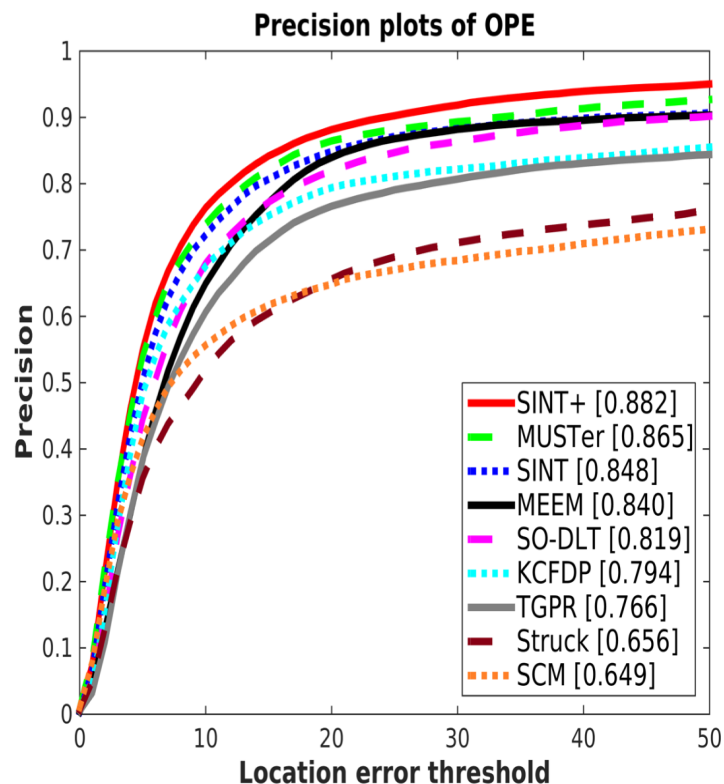
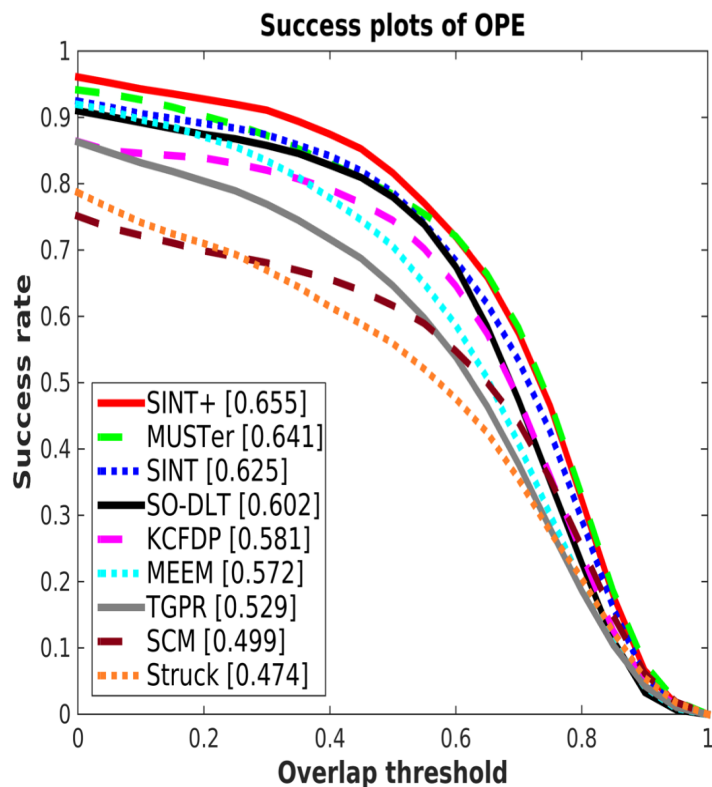
# Siamese training pairs

60,000 pairs of frames for training,  
2,000 pairs for validation  
128 pairs of boxes per pair of frames





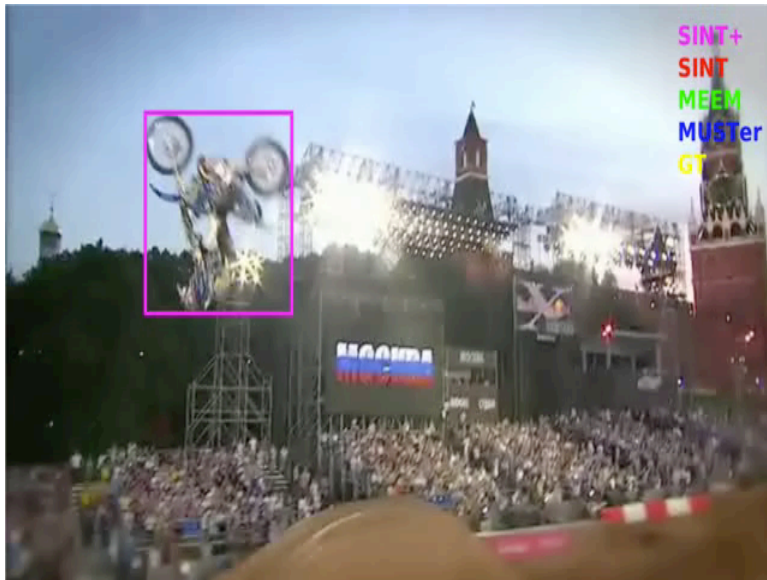
# SINT results



SINT+: adaptive sampling range [Want et al, ICCV15] & optical flow to remove motion inconsistent samples  
Results on [OTB50 Wu et al, CVPR13]

# SINT results

---



Can handle various types of appearance variations

The performance on subsequent frames will not be affected by the mistake made on the current frame.

# SINT results

Failure cases:  
confusing objects



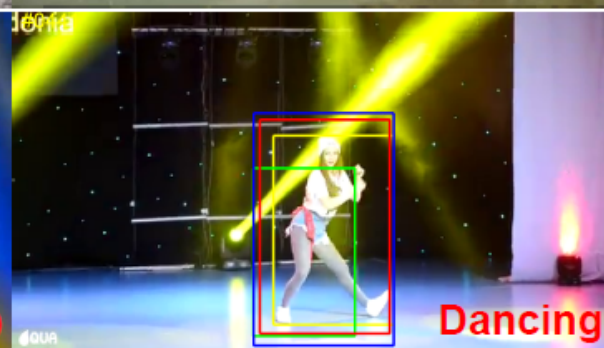
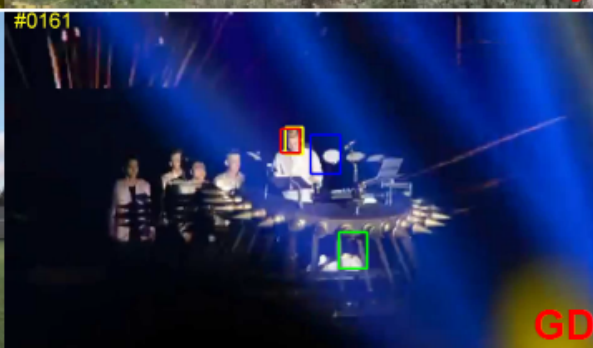
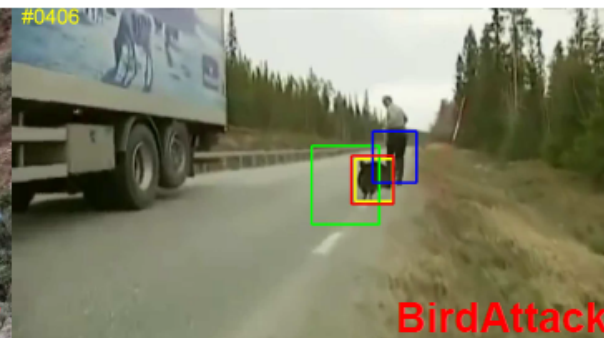
large occlusions





# SINT results

	MEEM [56]	MUSTer [18]	SINT
<i>Fishing</i>	4.3	11.2	53.7
<i>Rally</i>	20.4	27.5	53.4
<i>BirdAttack</i>	40.7	50.2	66.7
<i>Soccer</i>	36.9	48.0	72.5
<i>GD</i>	13.8	34.9	35.8
<i>Dancing</i>	60.3	54.7	66.8
mean	29.4	37.8	58.1



# SINT results: target reenters



# The hardness of tracking

---

Tracking aims to learn a target from the first few pictures; the target and the background may be dynamic in appearance, with unpredicted motion, and in difficult scenes.

Trackers tend to be under-evaluated, they tend to specialize in certain types of conditions.

Most recent trackers have learned from the oldies. We have found no definitive strategy yet, apart from *simplicity*, holding back on *updating*, apply *off-line learning* where possible.