Univerza v Ljubljani

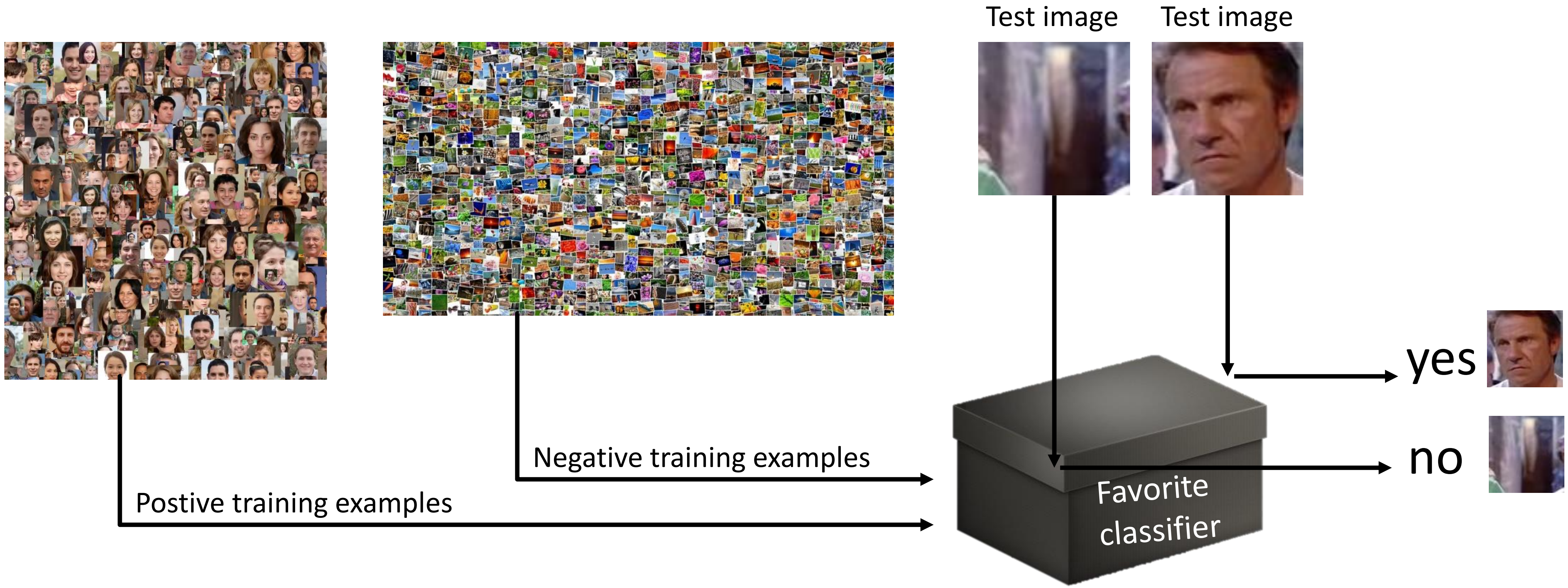
Advanced CV methods
Discriminative tracking – tracking by classifiers

Matej Kristan

Laboratorij za Umetne Vizualne Spoznavne Sisteme,
Fakulteta za računalništvo in informatiko,
Univerza v Ljubljani

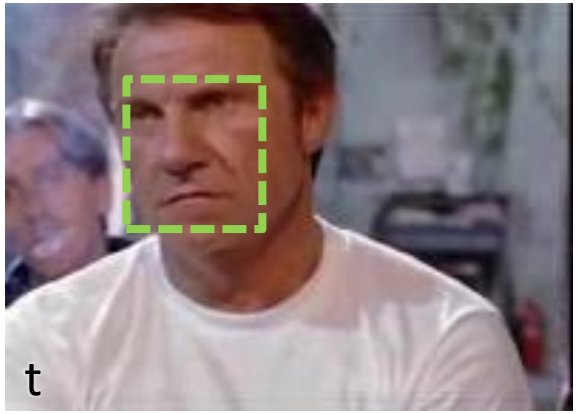
Tracking by a classifier

- A case study: tracking faces
- Take a (huge) number of cropped face images and even larger number of non-face images



Online discriminative tracking

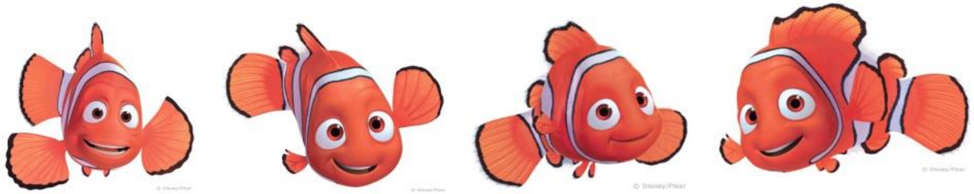
- The target does not move a lot between consecutive frames.
- Apply sliding window only within the region located at previous position.



- Choice of the type of classifier (object model) crucial for practical purpose!

Discriminative appearance model: Requirements

- Capability to adapt during tracking
  - Appearance changes (e.g. out of plane rotations)



- Appearance model robustness
  - Occlusions, cluttered background, illumination conditions

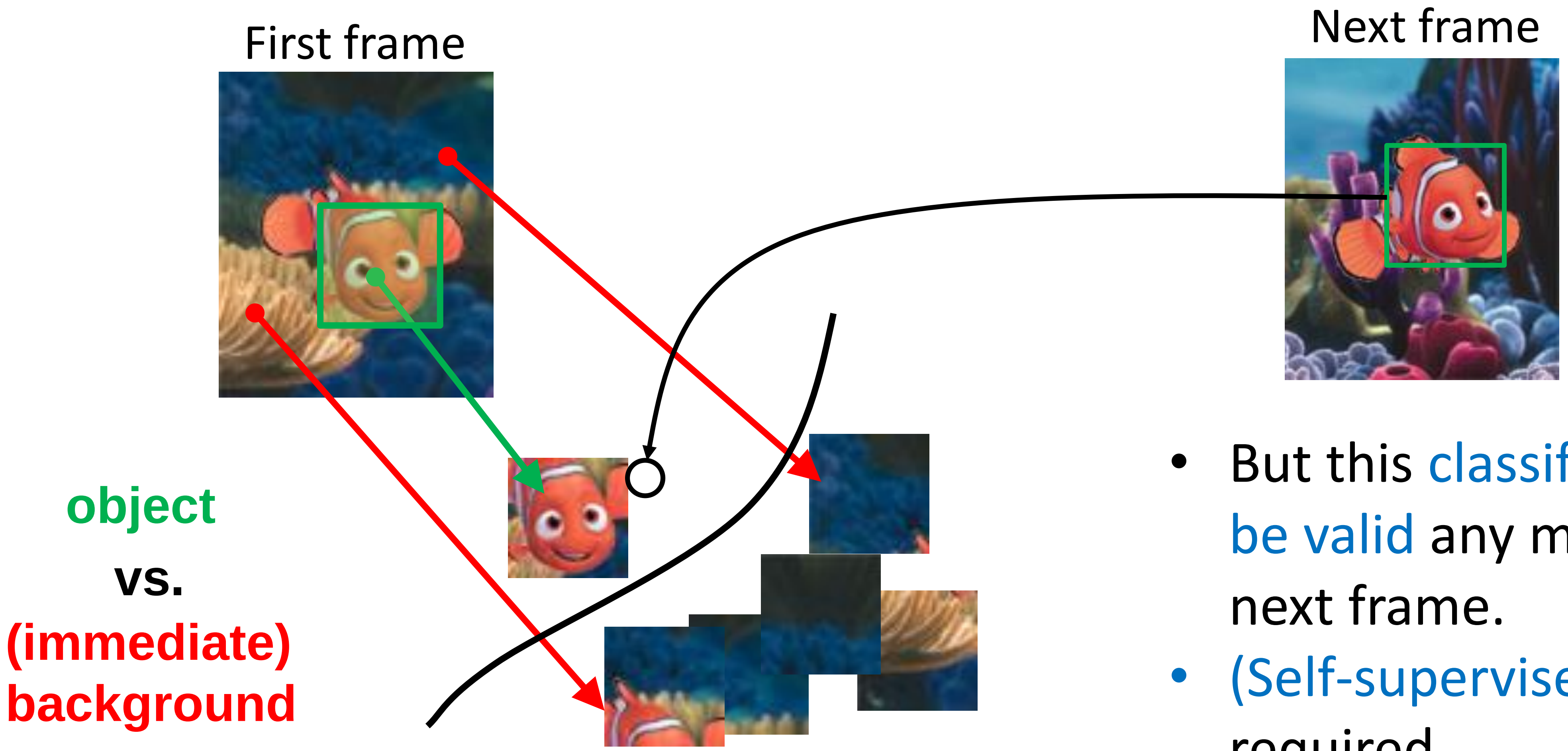
- Appearance model generality
  - Any object

Slide credit: Helmut Grabner

Tracking as a binary classification

S. Avidan. **Ensemble tracking**. CVPR 2005.
J.Wang, et al. **Online selecting discriminative tracking features using particle filter**. CVPR 2005.

- A single supervised training example provided in the first frame



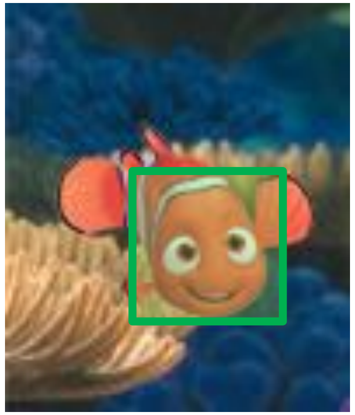
- But this classifier might not be valid any more for the next frame.
- (Self-supervised) update is required.

Tracking as a binary classification

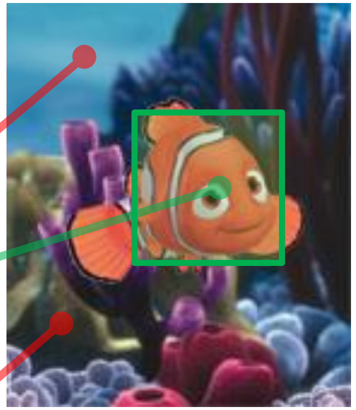
S. Avidan. **Ensemble tracking**. CVPR 2005.
J.Wang, et al. **Online selecting discriminative tracking features using particle filter**. CVPR 2005.

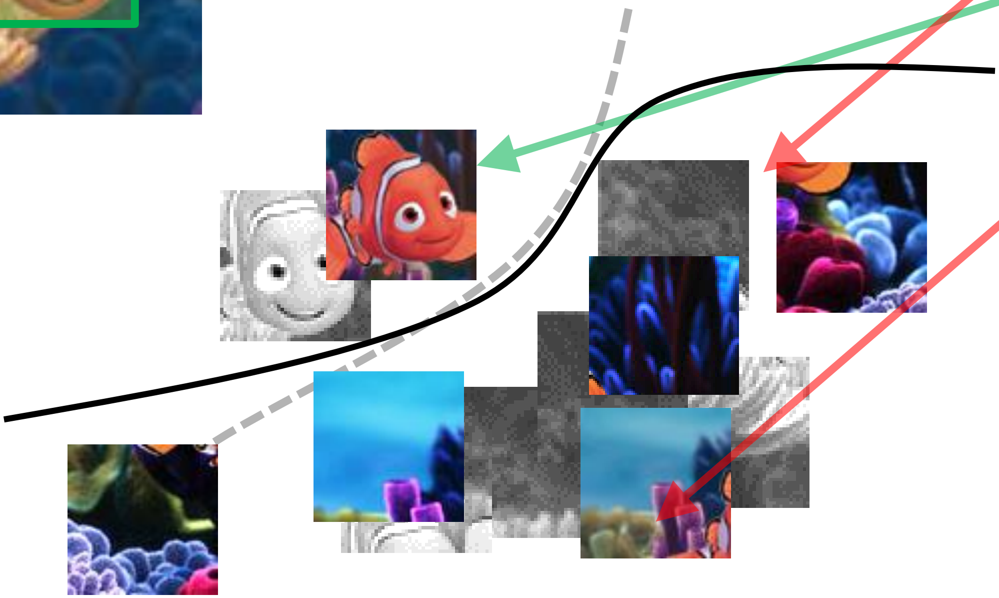
- A single supervised training example provided in the first frame

First frame



Next frame





object

vs.

(immediate) background

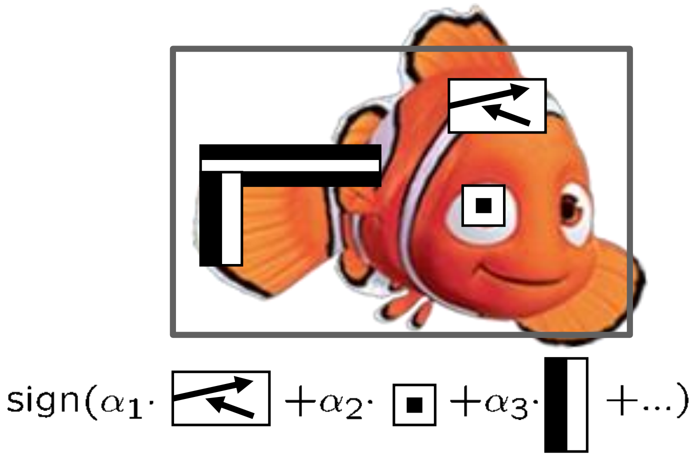
- Select positive examples from the target position.
- Select negative examples from the immediate background

Boosting for Feature Selection

Object Detector

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

Fixed Training set
General object detector



Combination of simple image features using Boosting as Feature Selection

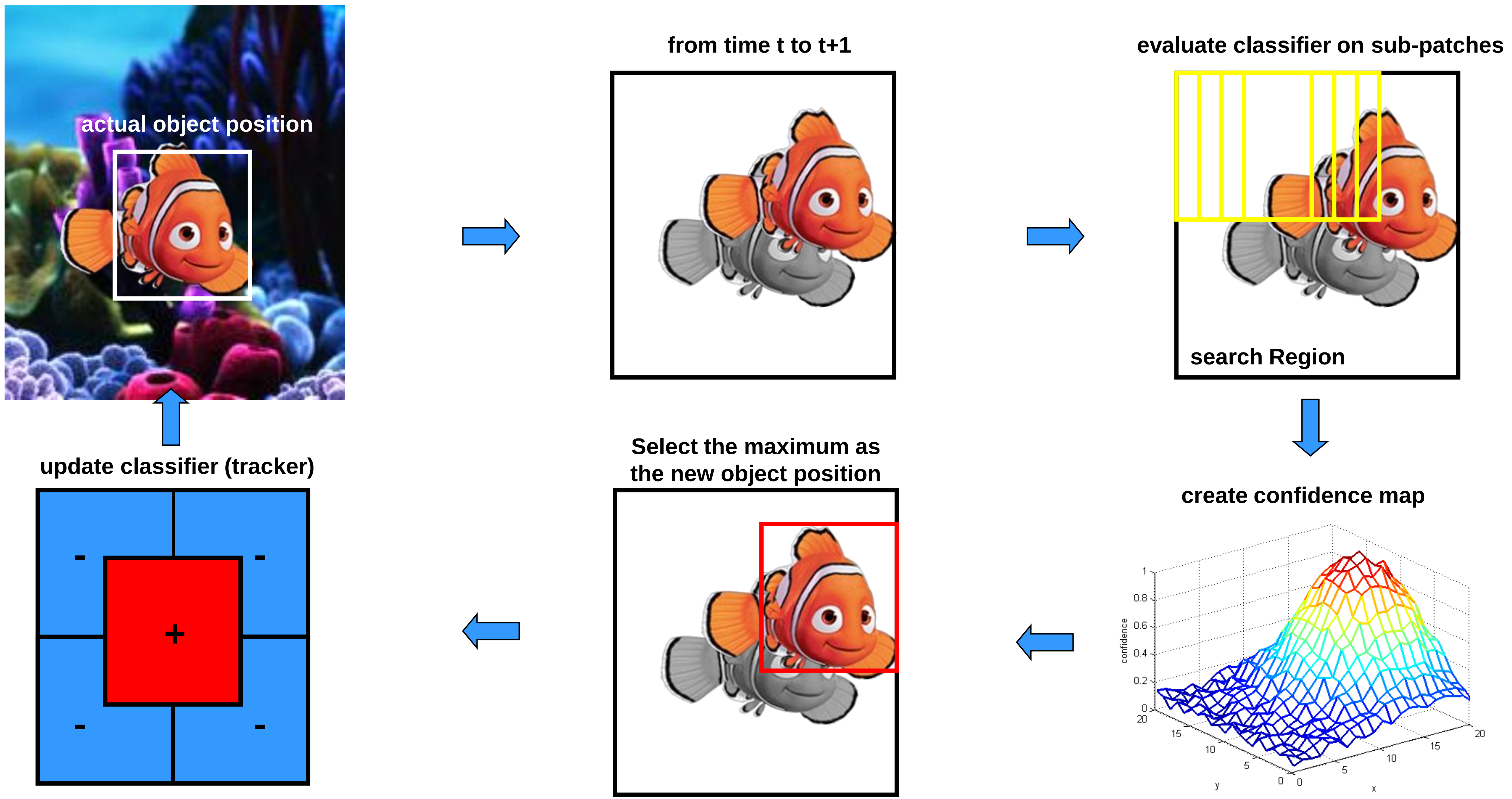
Object Tracker

On-line update
Object vs. Background

On-Line Boosting for Feature Selection

H. Grabner and H. Bischof. On-line boosting and vision. CVPR, 2006.

Tracking by online Adaboost



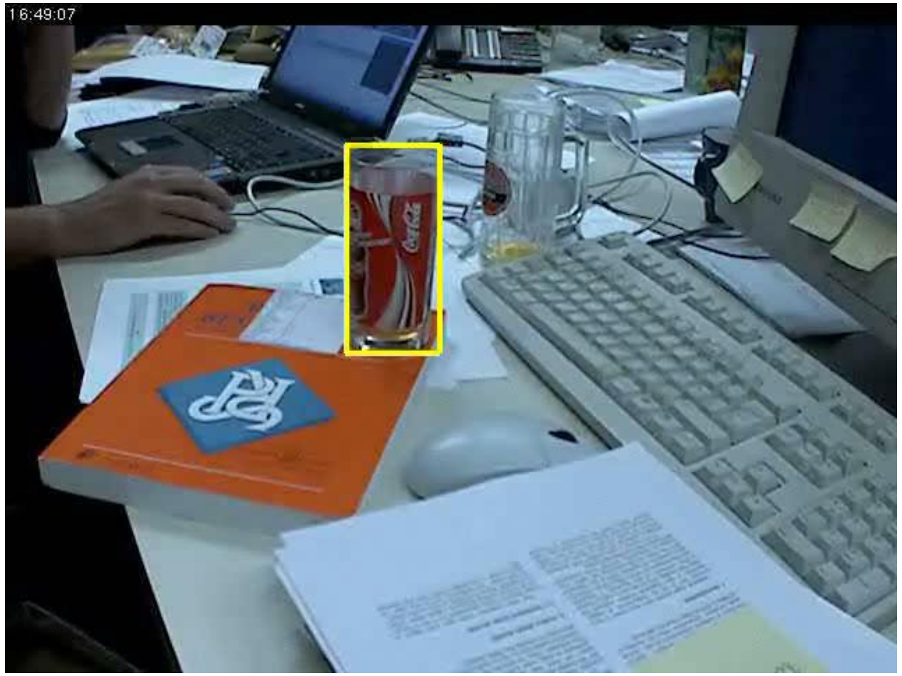
H. Grabner et. al., Real-Time Tracking via On-line Boosting . BMVC, 2006.

Slide credit: Helmut Grabner

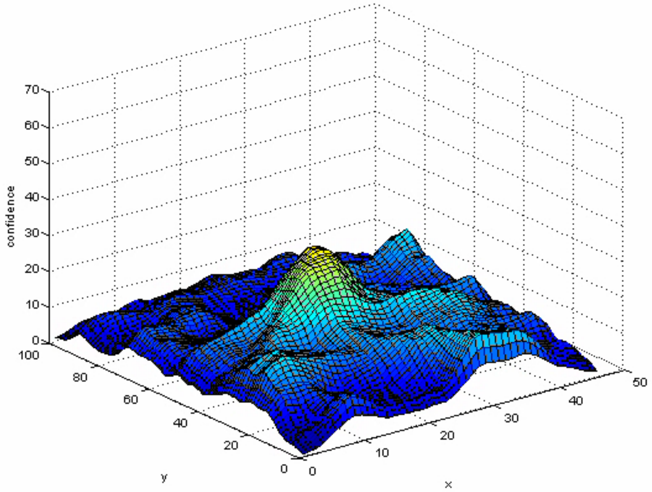
Tracking by online Adaboost

- Realtime performance
  - Fast feature computation
  - Efficient update of classifier

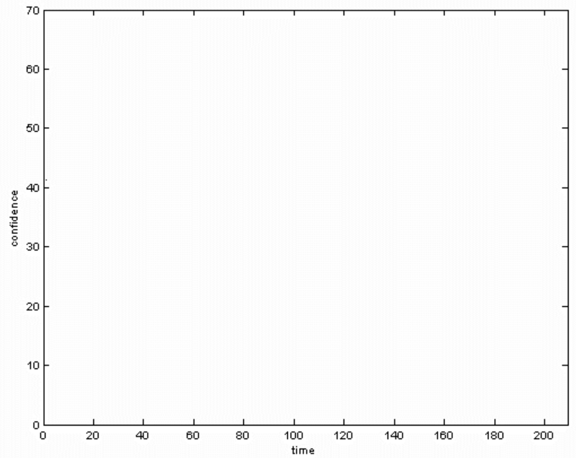
Tracking



Confidence Map

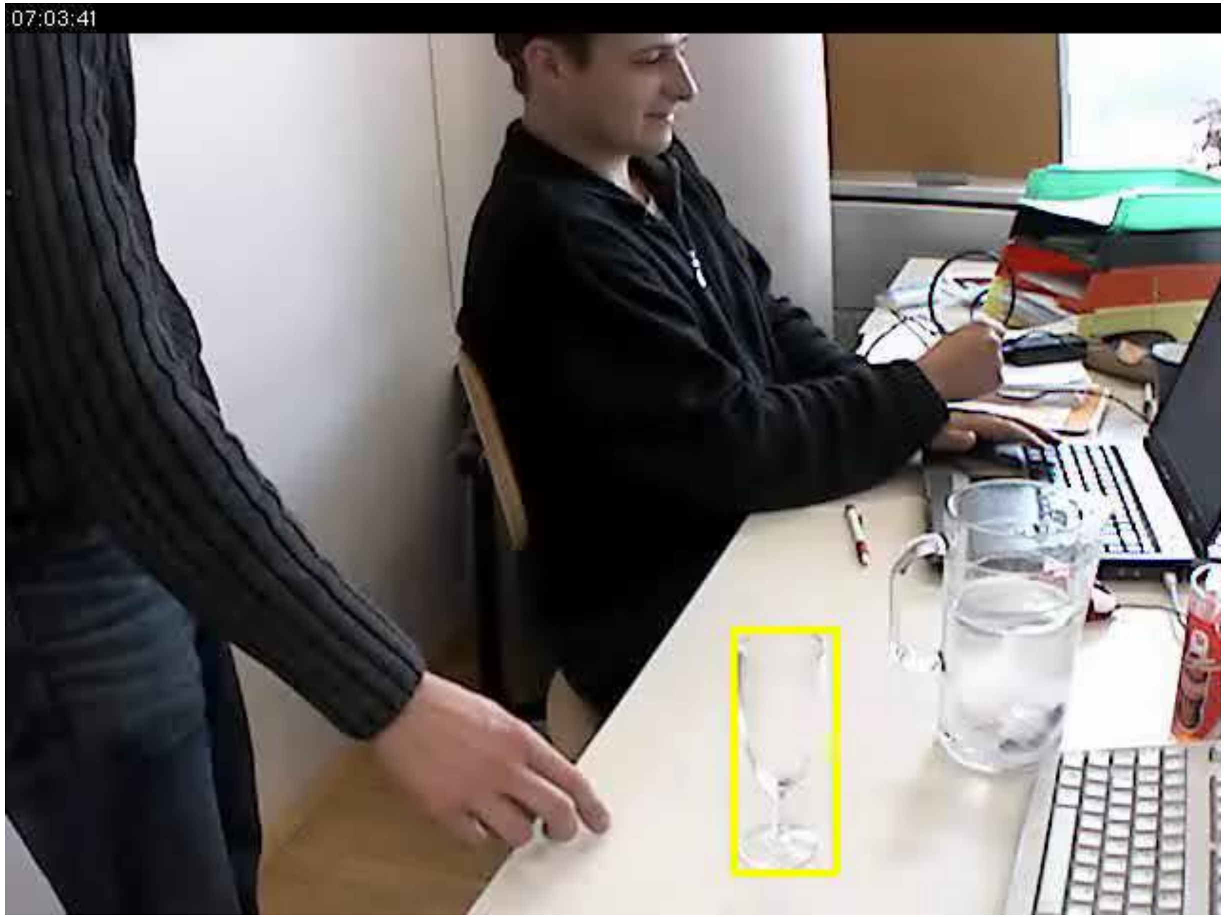


Max. Confidence Value



H. Grabner et. al., Real-Time Tracking via On-line Boosting . BMVC, 2006.

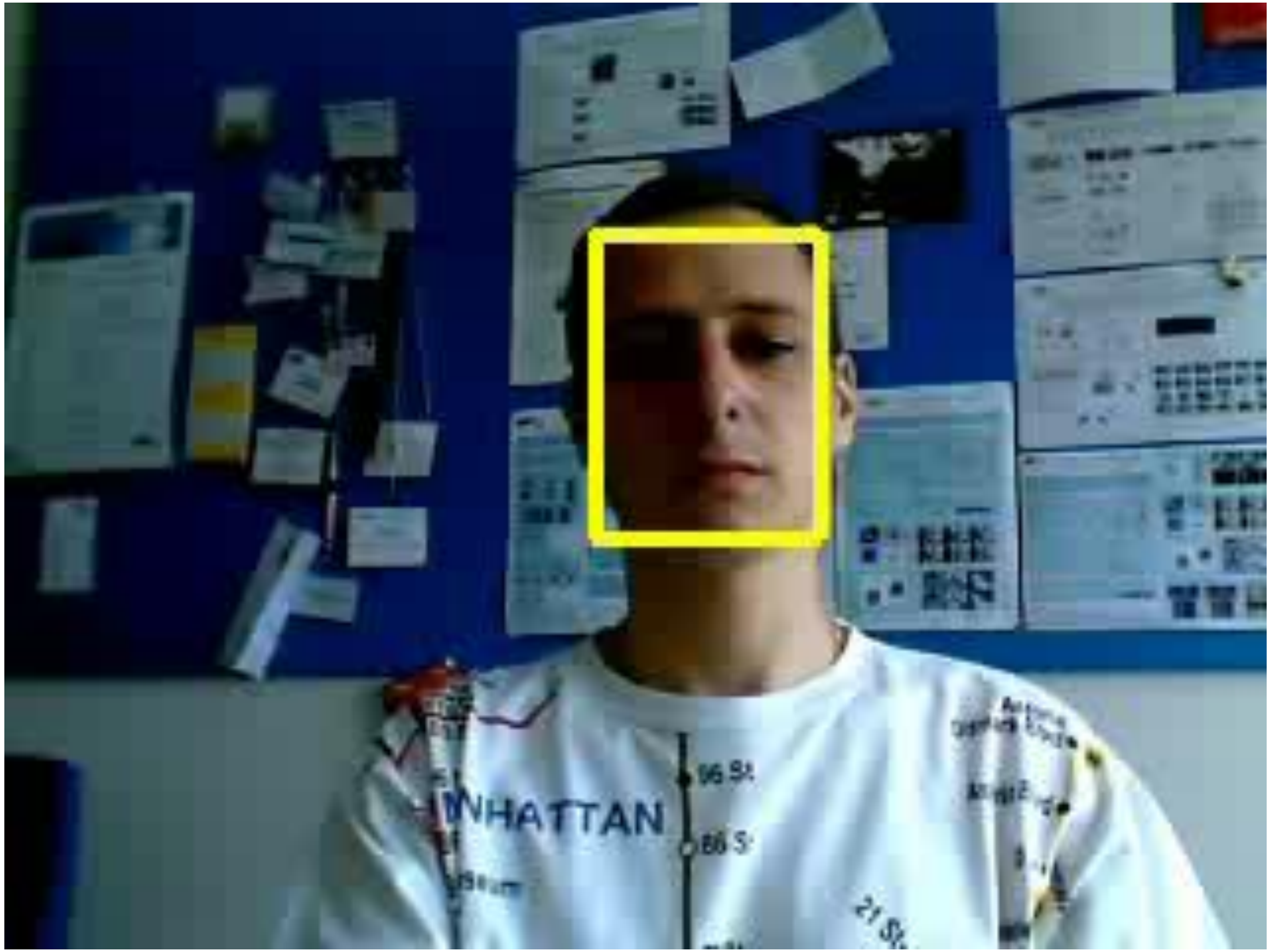
Tracking by online Adaboost

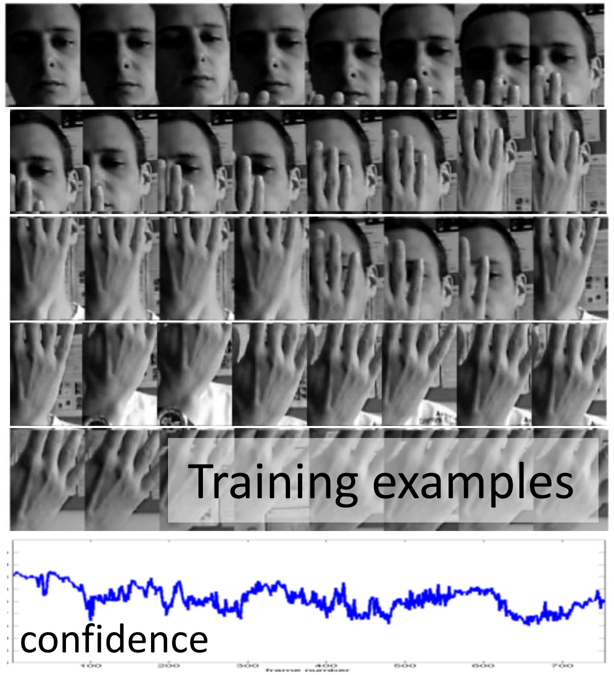


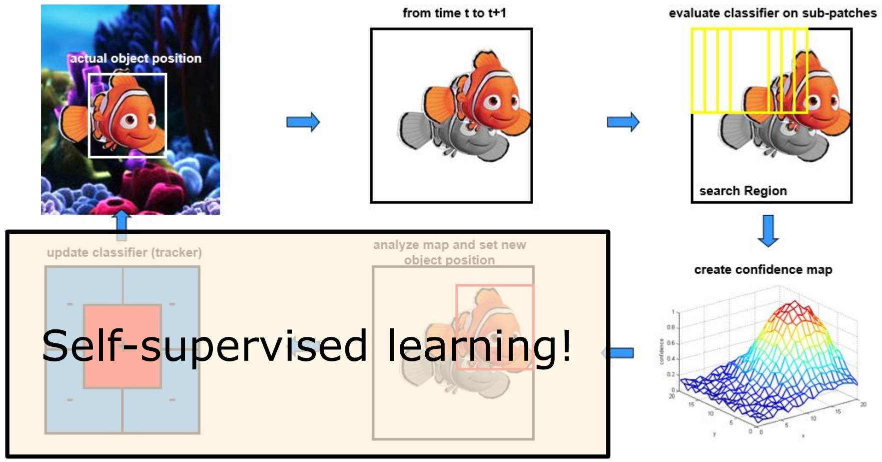
H. Grabner et. al., Real-Time Tracking via On-line Boosting . BMVC, 2006.

Slide credit: Helmut Grabner

Failure modes

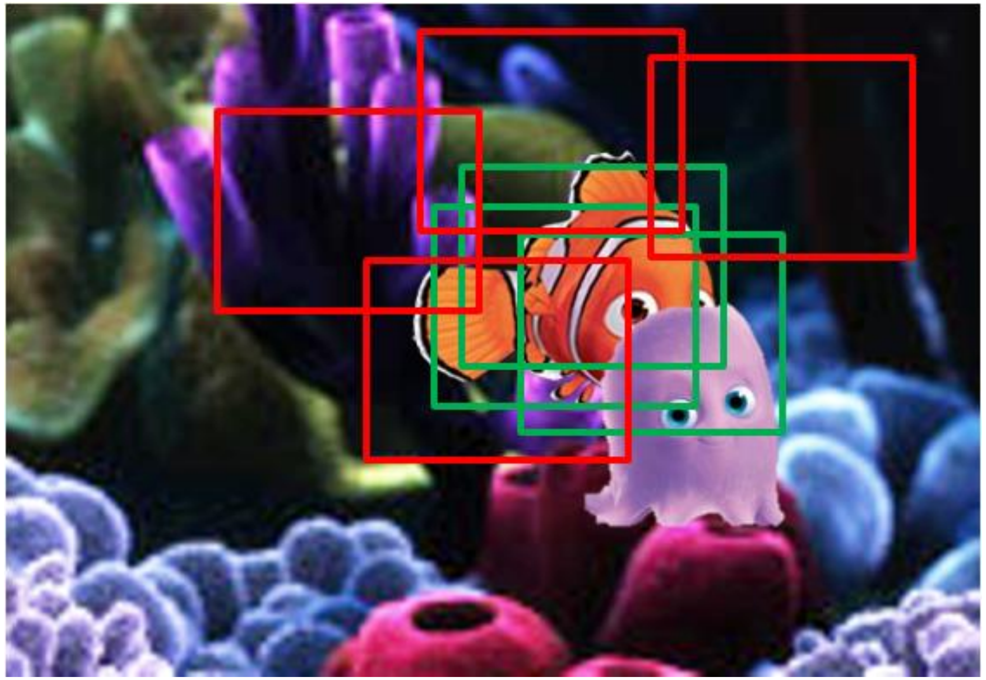




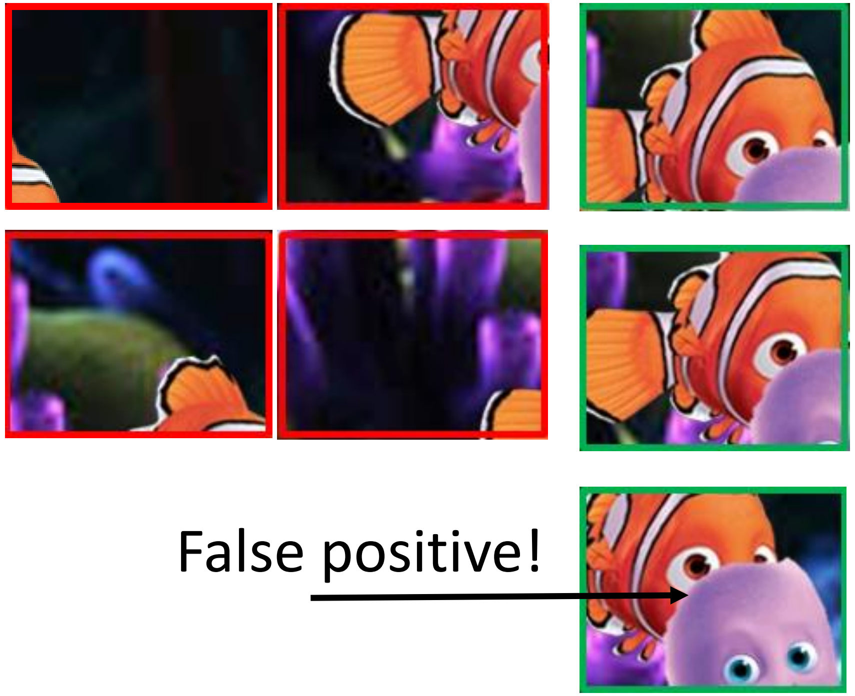


Do not trust all learning examples

Training image



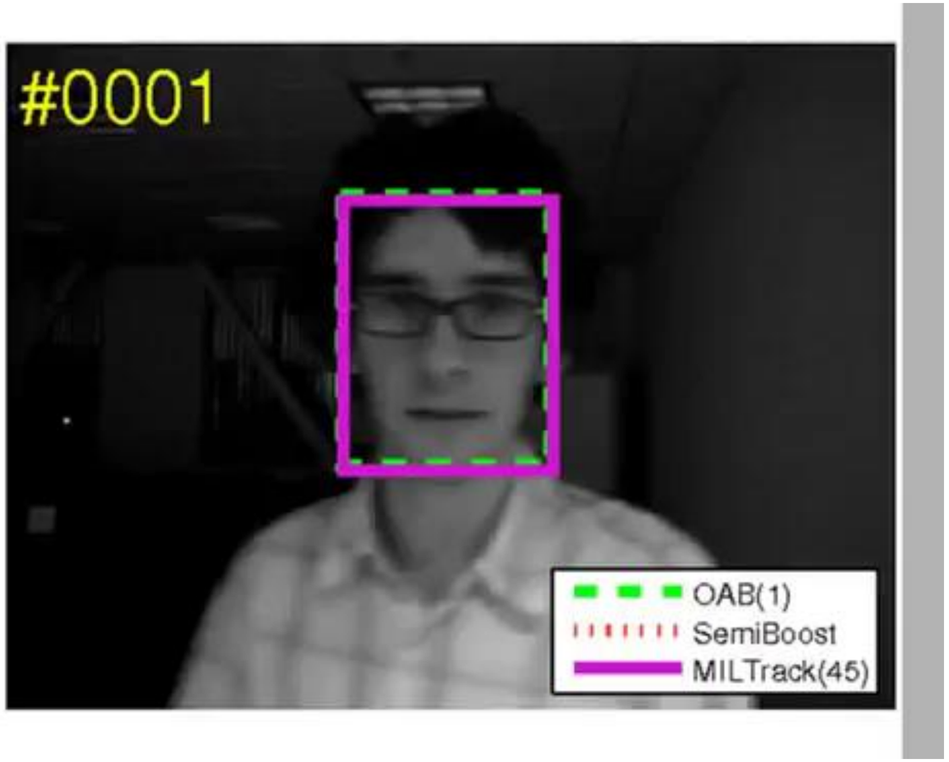
negatives positives

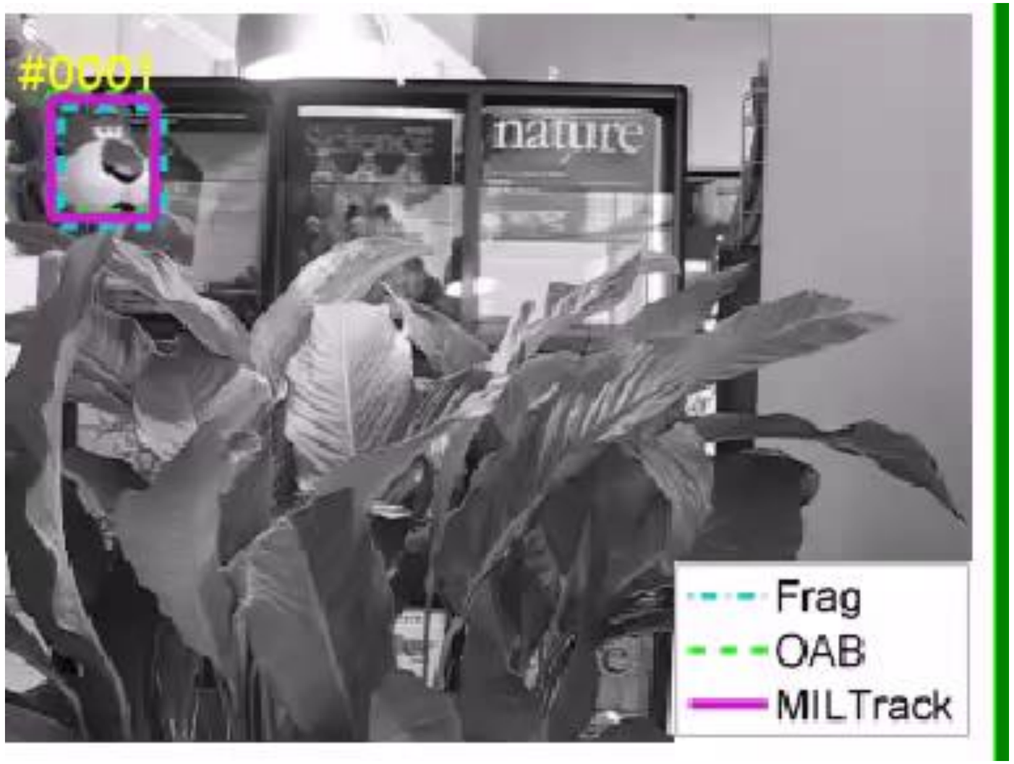


- Assume all negative examples are really negative
- Assume positive examples might contain *some* negatives
- A multiple instance learning (MIL) problem!

Babenko et al.,"Robust Object Tracking with Online Multiple Instance Learning", TPAMI2011

Do not trust all learning examples



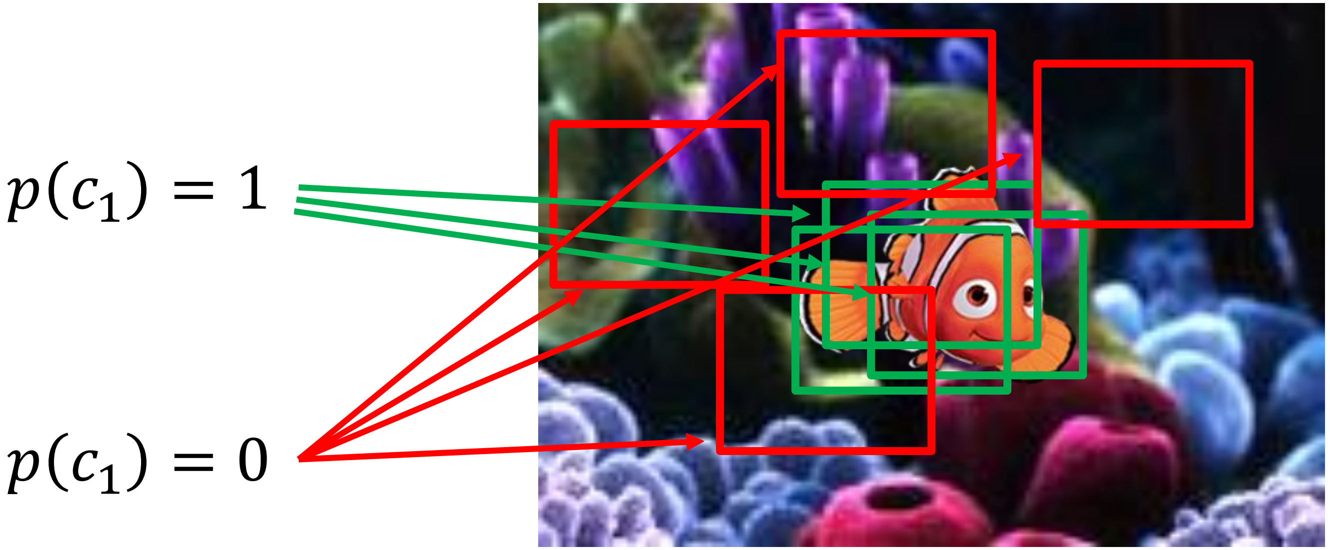


- Note that the online Adaboost failed *in this run* on the David sequence!
- Be sure that TMIL authors worked to show this, but it also says a lot about robustness of oAB to initialization!
- Code for TMIL available here.

Babenko et al.,"Robust Object Tracking with Online Multiple Instance Learning", TPAMI2011

Apply weights to training examples

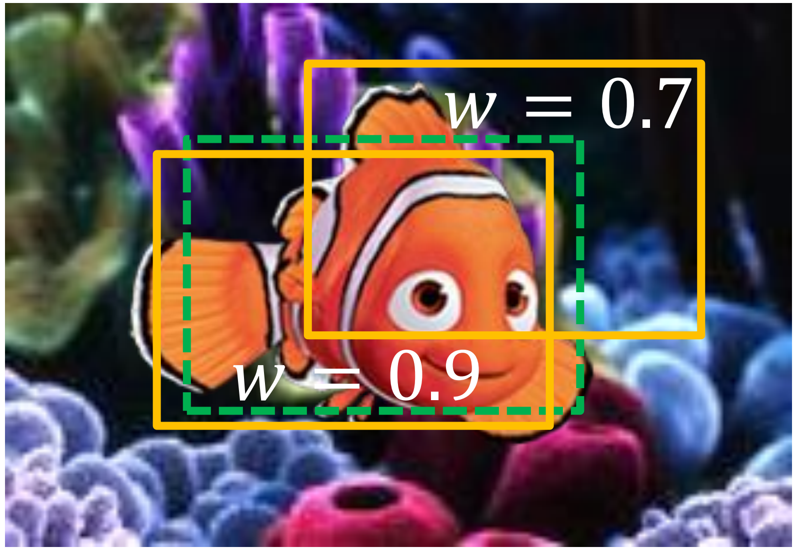
- Online AdaBoost and TMIL make hard decision on the class identity :

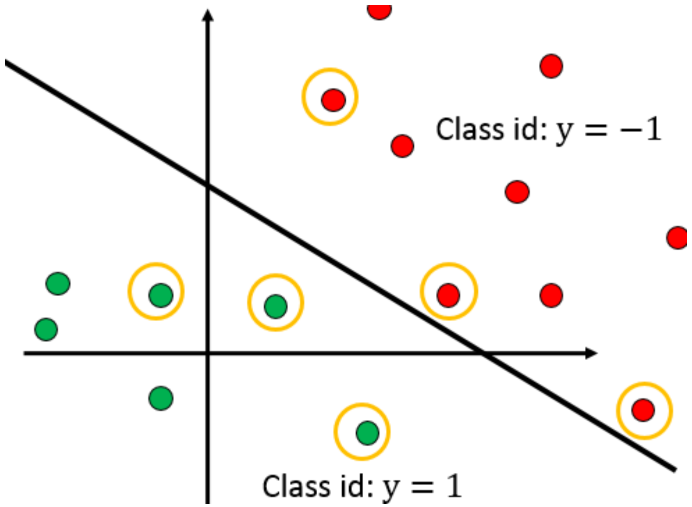


- But some positive examples are more positive than others and some negative examples are “more negative” then others…

Apply weights to training examples

- Weights proportional to estimated position overlap:

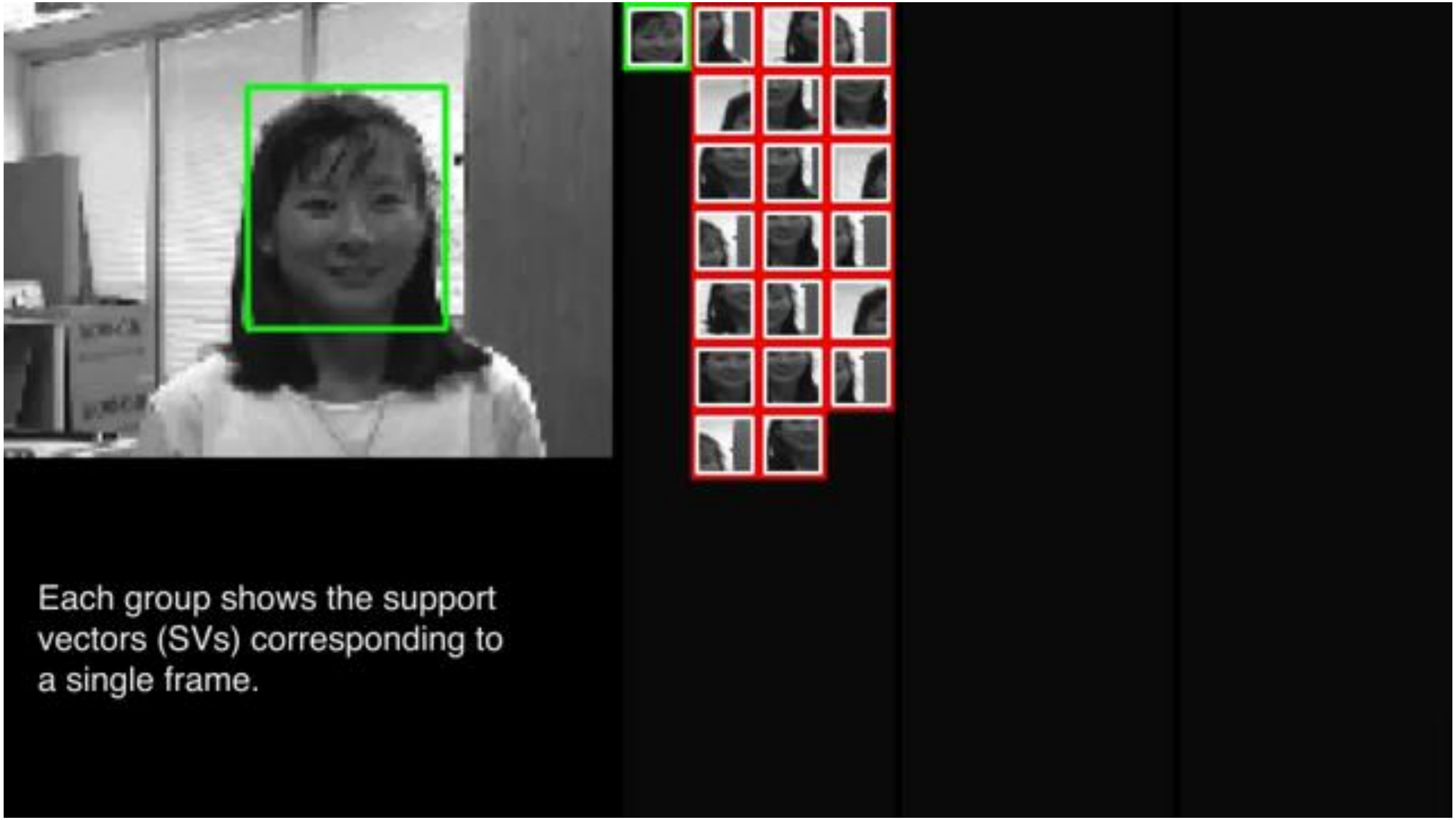




- Learning machinery:
  - Structured Support Vector Machine (online version)

Sam Hare, Amir Saffari, Philip H. S. Torr, [Struck: Structured Output Tracking with Kernels](), ICCV 2011

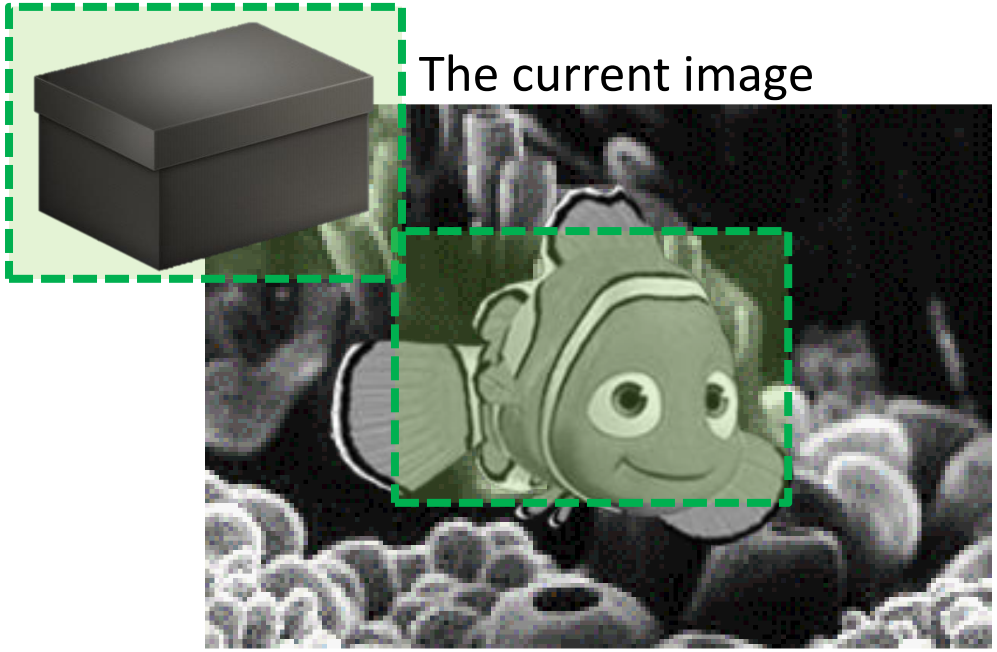
Struck tracking example

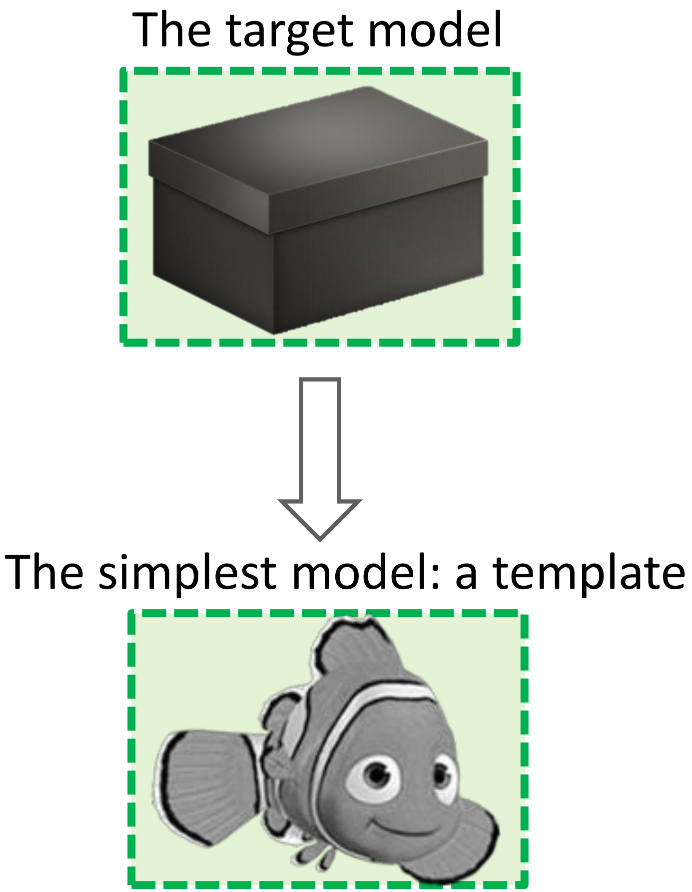


Sam Hare, Amir Saffari, Philip H. S. Torr, [Struck: Structured Output Tracking with Kernels](), ICCV 2011

Let’s take a step back…

- How is target detection carried out?

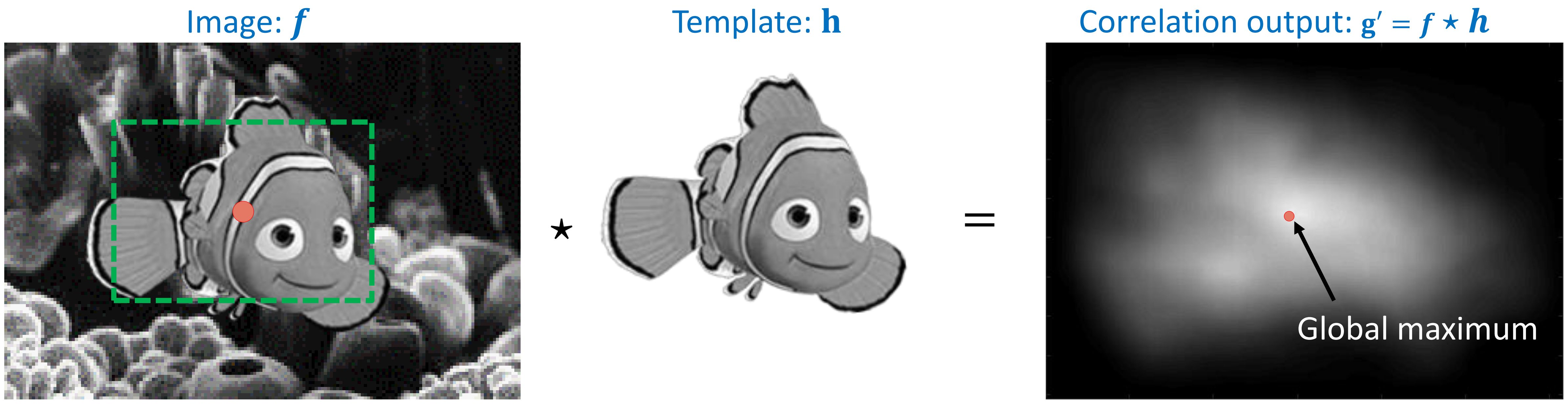




- Looks like a convolution/correlation
- A simplest model is the template

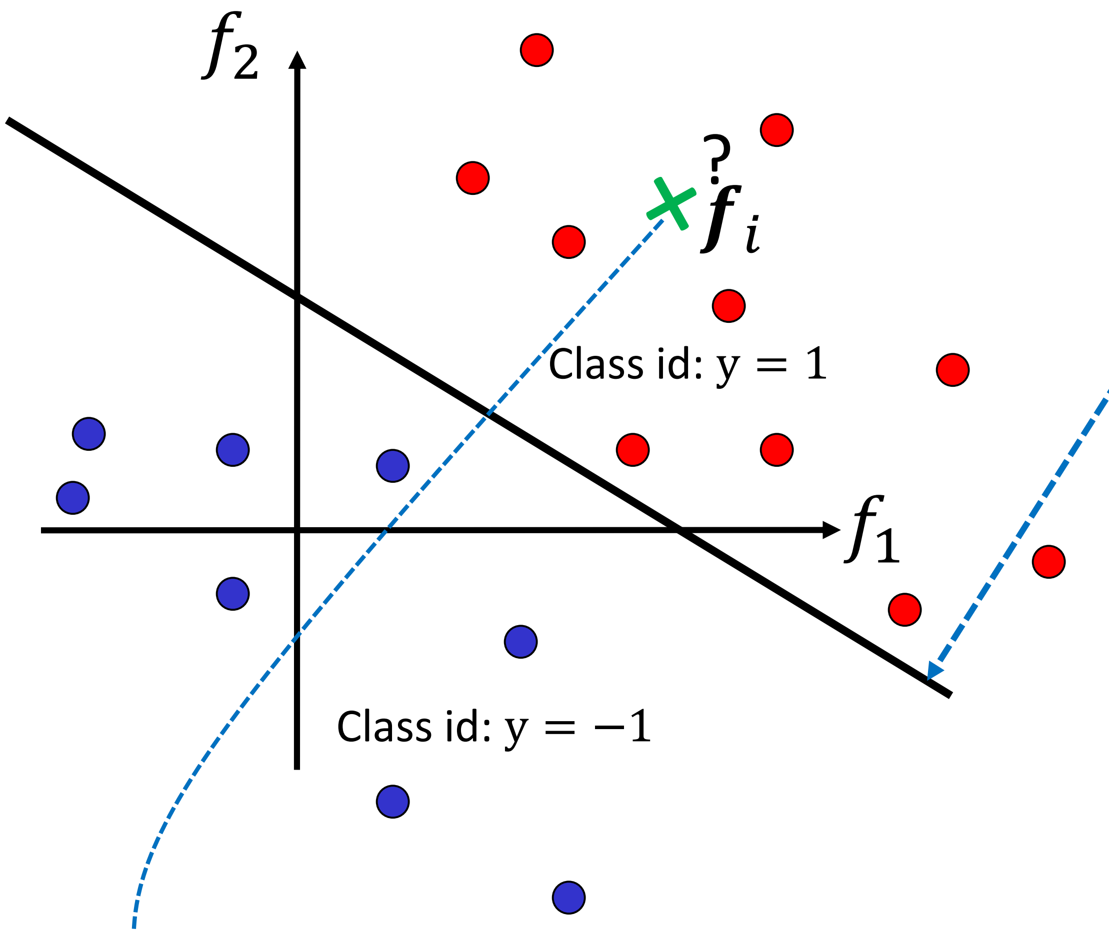
Correlation-based tracking

- Target localization: *maximum of correlation* of image $\boldsymbol{f}$ with a template $\boldsymbol{h}$.



- Recall how correlation is computed:
  - Crop a patch from $\boldsymbol{f}$, pointwise multiply with the template $\boldsymbol{h}$ and sum.
  - I.e., a dot product between the cropped patch and template.

Dot product implements a linear classifier / regressor



$$\langle \mathbf{h}, \mathbf{f}_i \rangle > 0$$ ... something very positive

A decision boundary, in general, a *hyper-plane*:

$$af_1 + cf_2 + b = 0$$

Define:

$$\mathbf{h} = \begin{bmatrix} a \\ c \\ b \end{bmatrix} \qquad \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ 1 \end{bmatrix}$$

A general hyper-plane eq:

$$\langle \mathbf{h}, \mathbf{f} \rangle = \mathbf{h}^T \mathbf{f} = 0$$

Efficient correlation computation

$\star$ $\qquad$ $*$

1. Correlation is a convolution using a flipped image:
2. Correlation equivalent to point-wise product in Fourier domain:

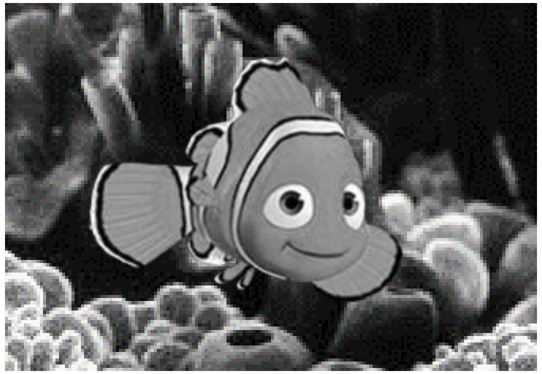
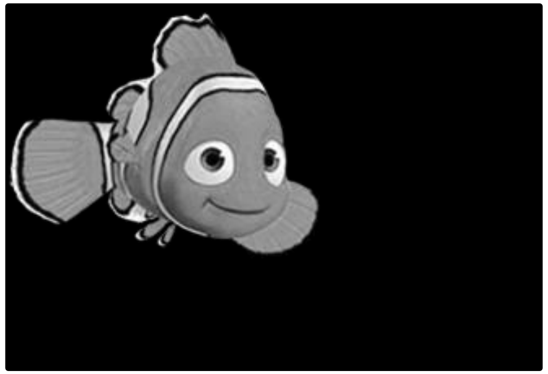
$\star$

$*$


$$\boldsymbol{g} = \boldsymbol{f} \star \boldsymbol{h} \quad \Leftrightarrow \quad \hat{\boldsymbol{g}} = \hat{\boldsymbol{f}} \odot \overline{\hat{\boldsymbol{h}}}$$

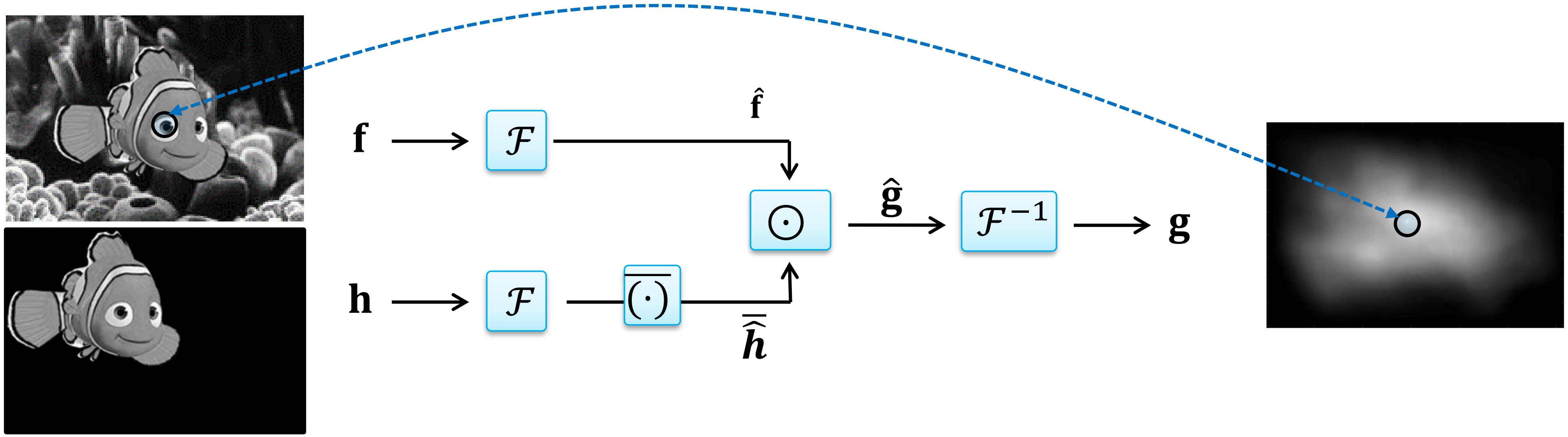
- Where:
  - $\hat{\boldsymbol{g}} = \mathcal{F}(\boldsymbol{g})$ … Fourirer transform of $\boldsymbol{g}$.
  - $\odot$ … element-wise product
  - $\overline{(\cdot)}$ … complex conjugate (i.e., imaginary part negated)
- Requirement:

  $\boldsymbol{f}$ and $\boldsymbol{h}$ must be of the same size


$\star$
pad the template with zeros


Efficient correlation computation

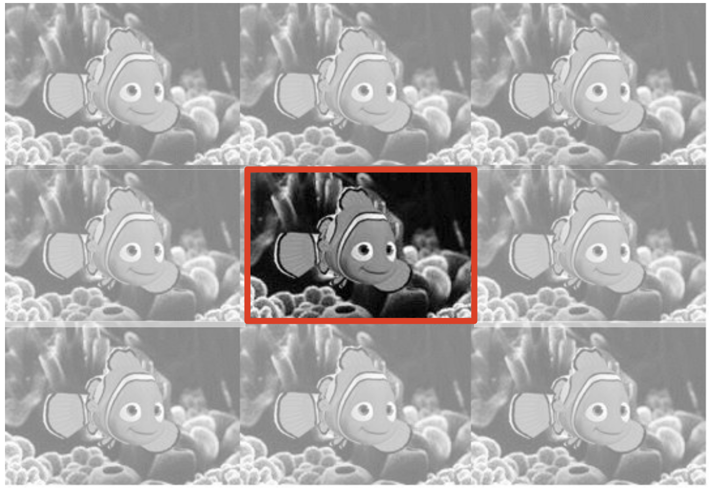
- Correlation via Fourier domain: $\boldsymbol{g} = \boldsymbol{f} \star \boldsymbol{h} \iff \hat{\boldsymbol{g}} = \hat{\boldsymbol{f}} \odot \overline{\hat{\boldsymbol{h}}}$



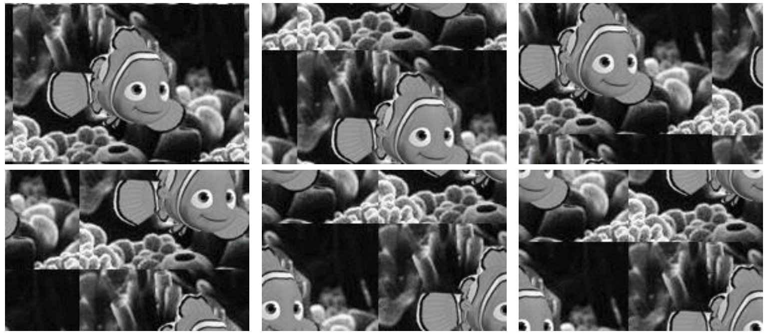
- Orders of magnitude speedup:
  - For $n \times n$ images, cross-correlation is $\mathcal{O}(n^4)$.
  - Fast Fourier Transform (and its inverse) are $\mathcal{O}(n^2 \log n)$.

Efficient correlation computation

- Correlation is *circular* in discrete Fourier transform (DFT)!

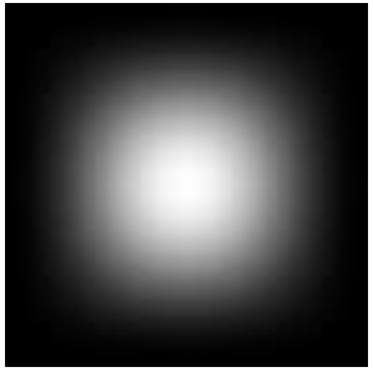


Circular shifts



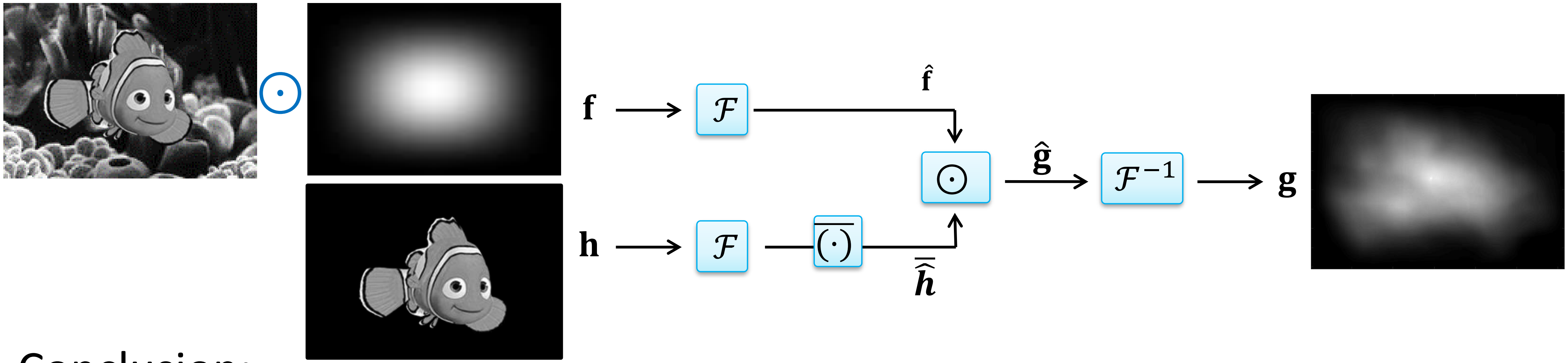
…

- To reduce the boundary effect, multiply the image $\boldsymbol{f}$ by a Hanning window:

 $\odot$  $=$ 

Efficient correlation computation

- Correlation via Fourier domain: $\boldsymbol{g} = \boldsymbol{f} \star \boldsymbol{h} \iff \hat{\boldsymbol{g}} = \hat{\boldsymbol{f}} \odot \overline{\hat{\boldsymbol{h}}}$

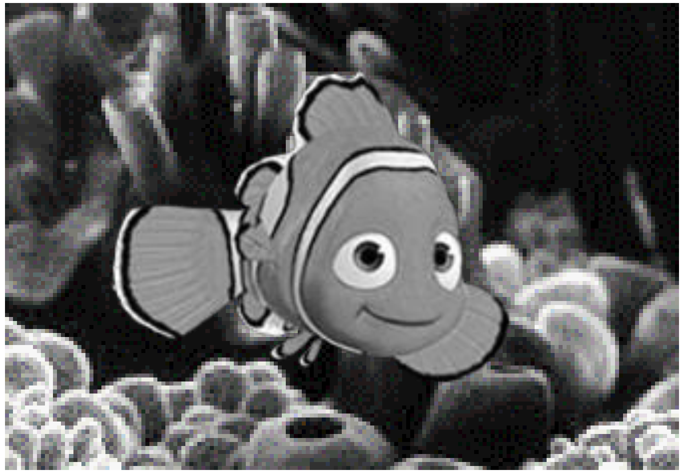


- Conclusion:
  - Correlation can be significantly accelerated by FFT
  - Since it evaluates $\langle f, h \rangle$ at all displacements it implements a fast linear classifier (regressor) evaluation at all displacements!
  - But how to learn the most suitable template $\boldsymbol{h}$?

Discriminative correlation learning

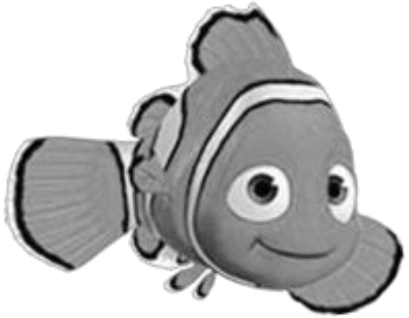
- Ideally, we would like a well expressed maximum at the object location:

Image: $\boldsymbol{f}$



$\star$

Template: $\mathbf{h}$



$=$

Correlation output: $\mathbf{g}' = \boldsymbol{f} \star \boldsymbol{h}$

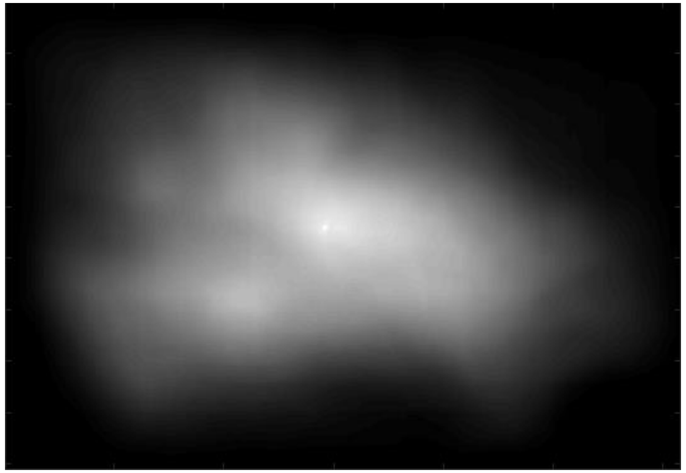
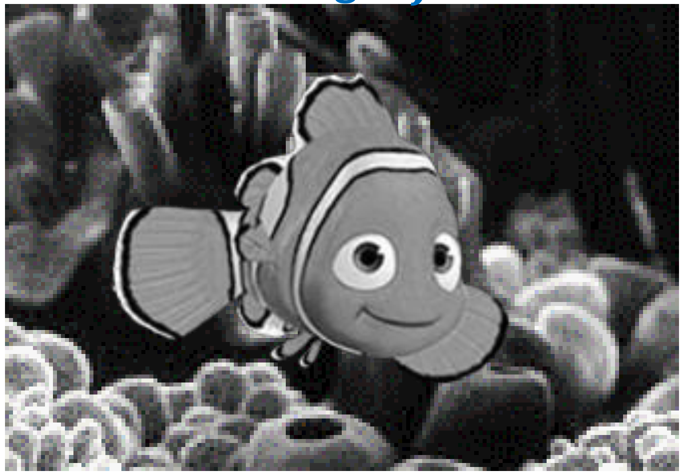


Image: $\boldsymbol{f}$



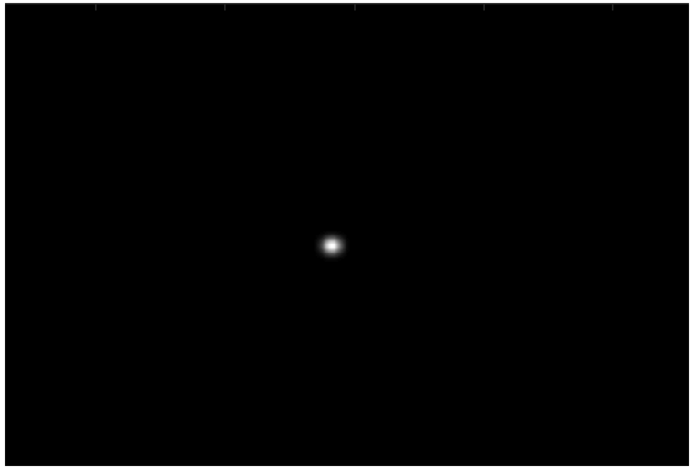
$\star$

Template: $\mathbf{h}$

$\boxed{\boldsymbol{h} = ?}$

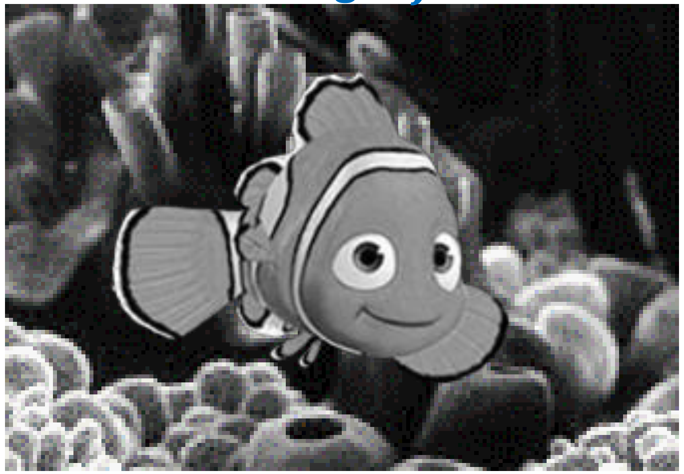
$=$

Ideal correlation output: $\mathbf{g}$



Discriminative Correlation Filters

Image: $\boldsymbol{f}$



Template: $\mathbf{h}$

$$\star \quad \boxed{\boldsymbol{h} = ?} \quad =$$

Ideal correlation output: $\mathbf{g}$

Find $\boldsymbol{h}$ that minimizes the cost ϵ.

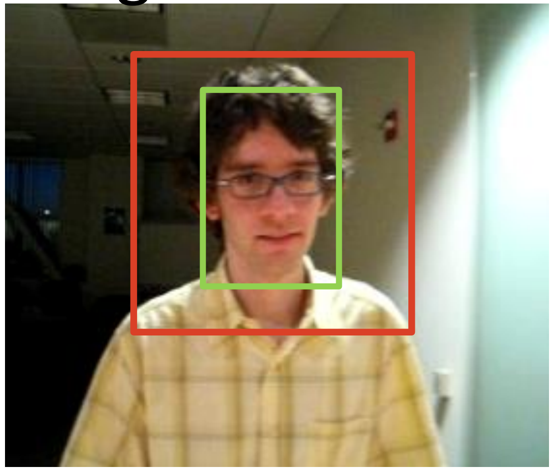
- Formalize the cost: difference between the correlation of template $\boldsymbol{h}$ with the image $\boldsymbol{f}$, i.e.,

$$\epsilon = ||\boldsymbol{f} \star \boldsymbol{h} - \boldsymbol{g}||^2 + \lambda||\boldsymbol{h}||^2.$$

- Learning: Given the image $\boldsymbol{f}$, find the filter $\boldsymbol{h}$ that minimizes ϵ.

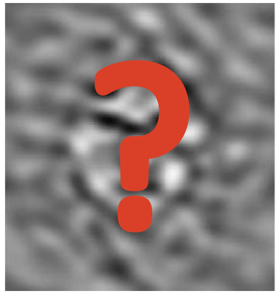
Discriminative Correlation Filters in a nutshell

Target localized



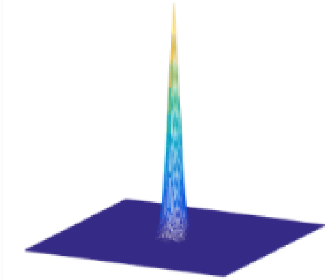
Training example: $\boldsymbol{f}$

$\star$



Template: $\boldsymbol{h}$

$=$



Desired response: $\boldsymbol{g}$

green bbox: target region, red bbox: search region

Pixel-wise product!

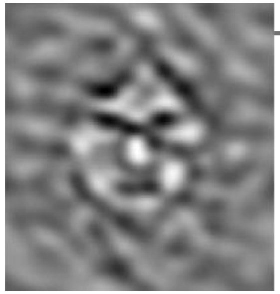
$$\arg\min_{\mathbf{h}} |\mathbf{f}\star\mathbf{h}-\mathbf{g}|^2+\lambda|\mathbf{h}|^2 = \arg\min_{\bar{\hat{\mathbf{h}}}} |\hat{\mathbf{f}}\odot\bar{\hat{\mathbf{h}}}-\hat{\mathbf{g}}|^2+\lambda|\hat{\mathbf{h}}|^2$$

Pixel-wise division!

Closed-form solution: $\bar{\hat{\mathbf{h}}} = \dfrac{\hat{\mathbf{g}}\odot\bar{\hat{\mathbf{f}}}}{\hat{\mathbf{f}}\odot\bar{\hat{\mathbf{f}}}+\lambda}$

Tracking algorithm outline

Filter: $\boldsymbol{h}, \widehat{\boldsymbol{h}}$



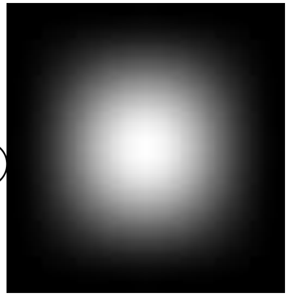
$$(\cdot)^{\dagger} = \bar{(\cdot)}$$

Localization step: Filter application

Current image

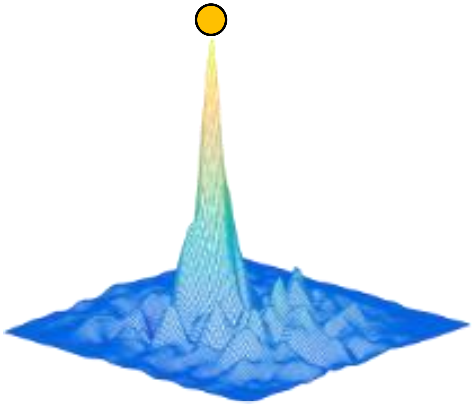
(1) Extract search region

(2) Compute FFT of the modified search region

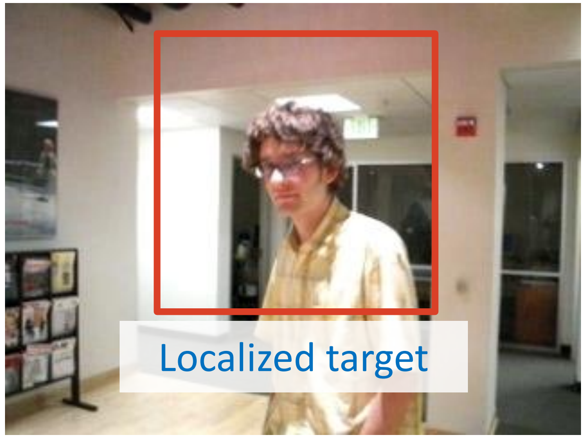
$FFT($  $\odot$  $) = \hat{\boldsymbol{f}}$

$$\mathrm{IFFT}(\hat{\mathbf{f}} \odot \hat{\mathbf{h}}^{\dagger})$$

(3) Multiply region and template and inverse FFT



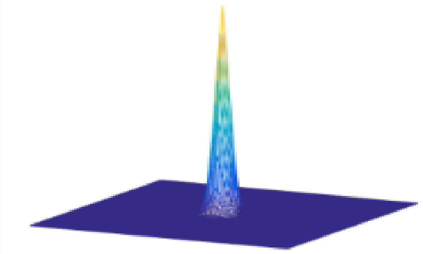
(4) Take max position as target center



Tracking algorithm outline

Update step: Filter learning

(3) Compute FFT of desired response $\hat{g} = FFT(g)$



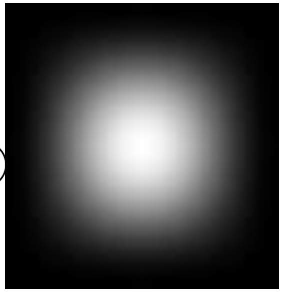
$$(\cdot)^{\dagger} = \bar{(\cdot)}$$

Current image

Localized target

(1) Extract region

(2) Compute FFT of the modified region

$FFT($  $\odot$  $) = \hat{\boldsymbol{f}}$

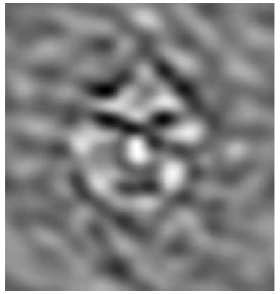
$$\hat{h}^{\dagger} = \frac{\hat{g} \odot \hat{f}^{\dagger}}{\hat{f} \odot \hat{f}^{\dagger} + \lambda}$$

(4) Compute the filter

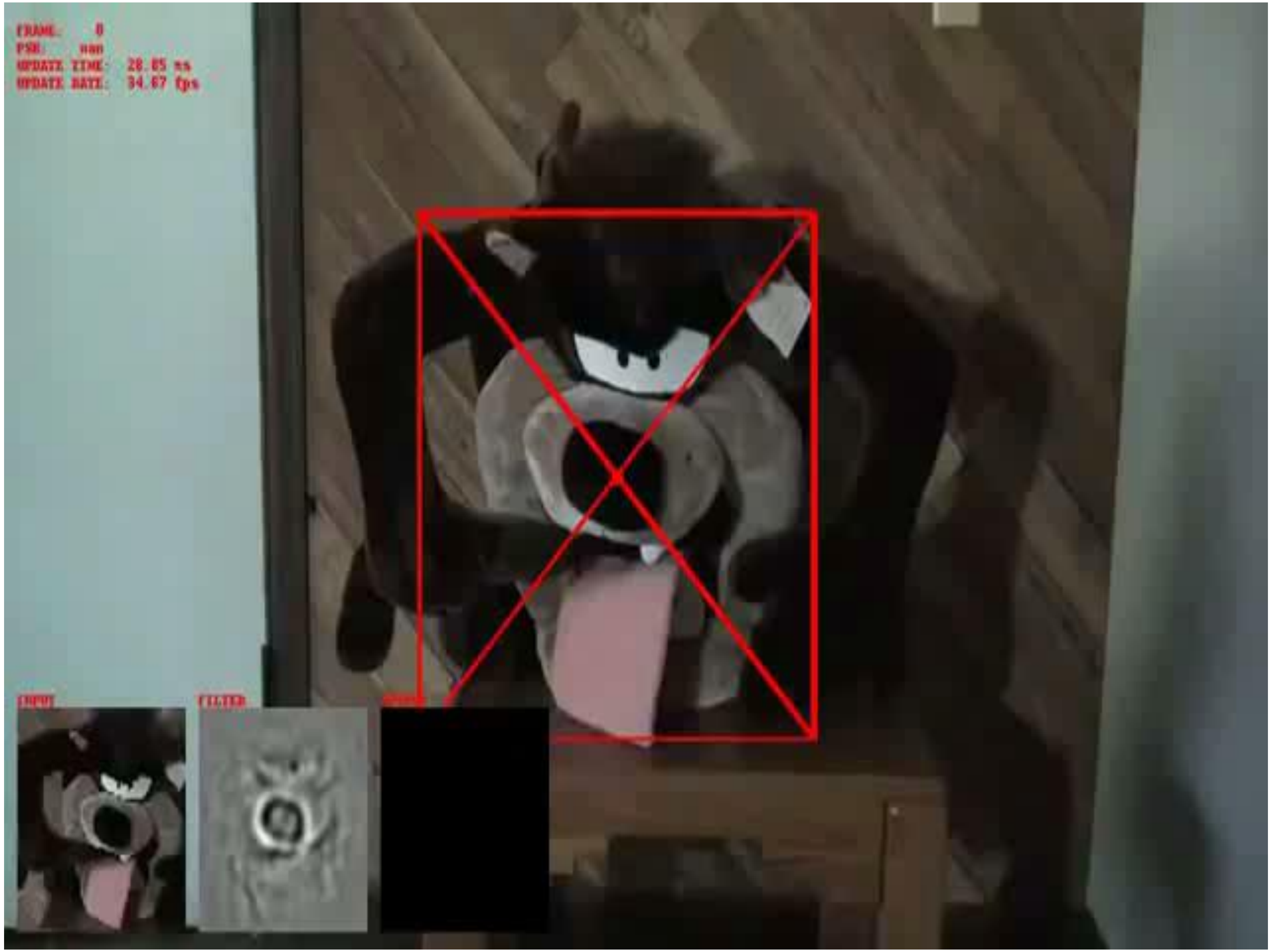
$$\hat{\boldsymbol{h}}_k = \hat{\boldsymbol{h}}_{k-1}\alpha + \hat{\boldsymbol{h}}(1-\alpha)$$

(5) Average the filter with filter from previous time-step

Final filter: $\boldsymbol{h}, \hat{\boldsymbol{h}}$



A basic CF tracker: MOSSE

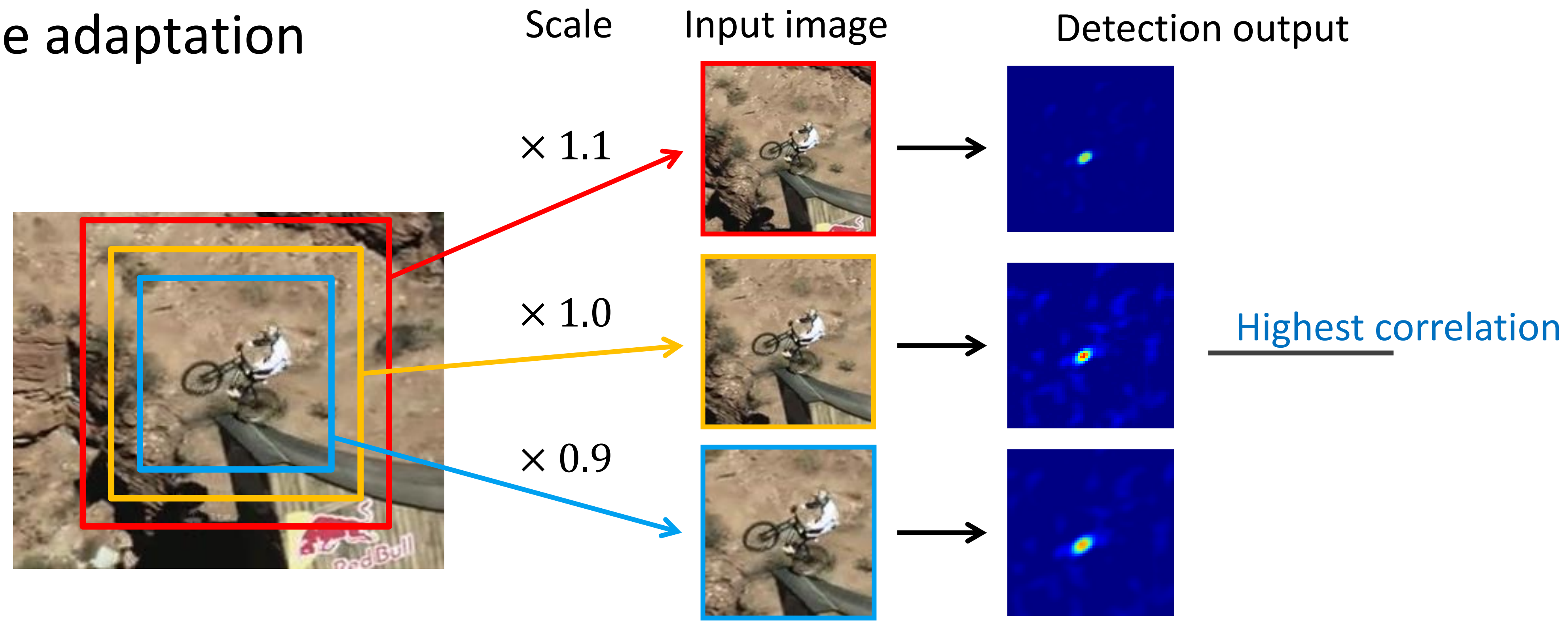


Simplest version reaches speeds approximately 300fps.

Bolme, Beveridge, Draper, Lui, Visual Object Tracking using Adaptive Correlation Filters, CVPR2010

Scale estimation during tracking

- Scale adaptation

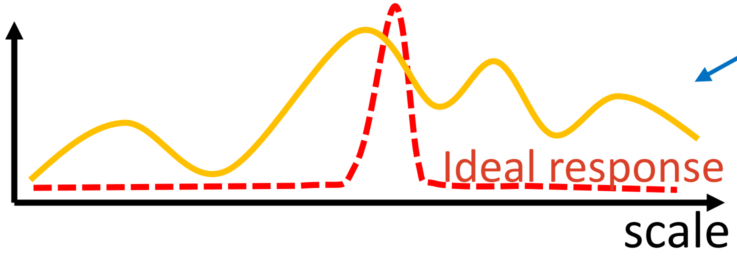


- Extract patches with different scales and normalize them to the same size
- Run classification (correlation) on all patches and output bounding box with the highest response

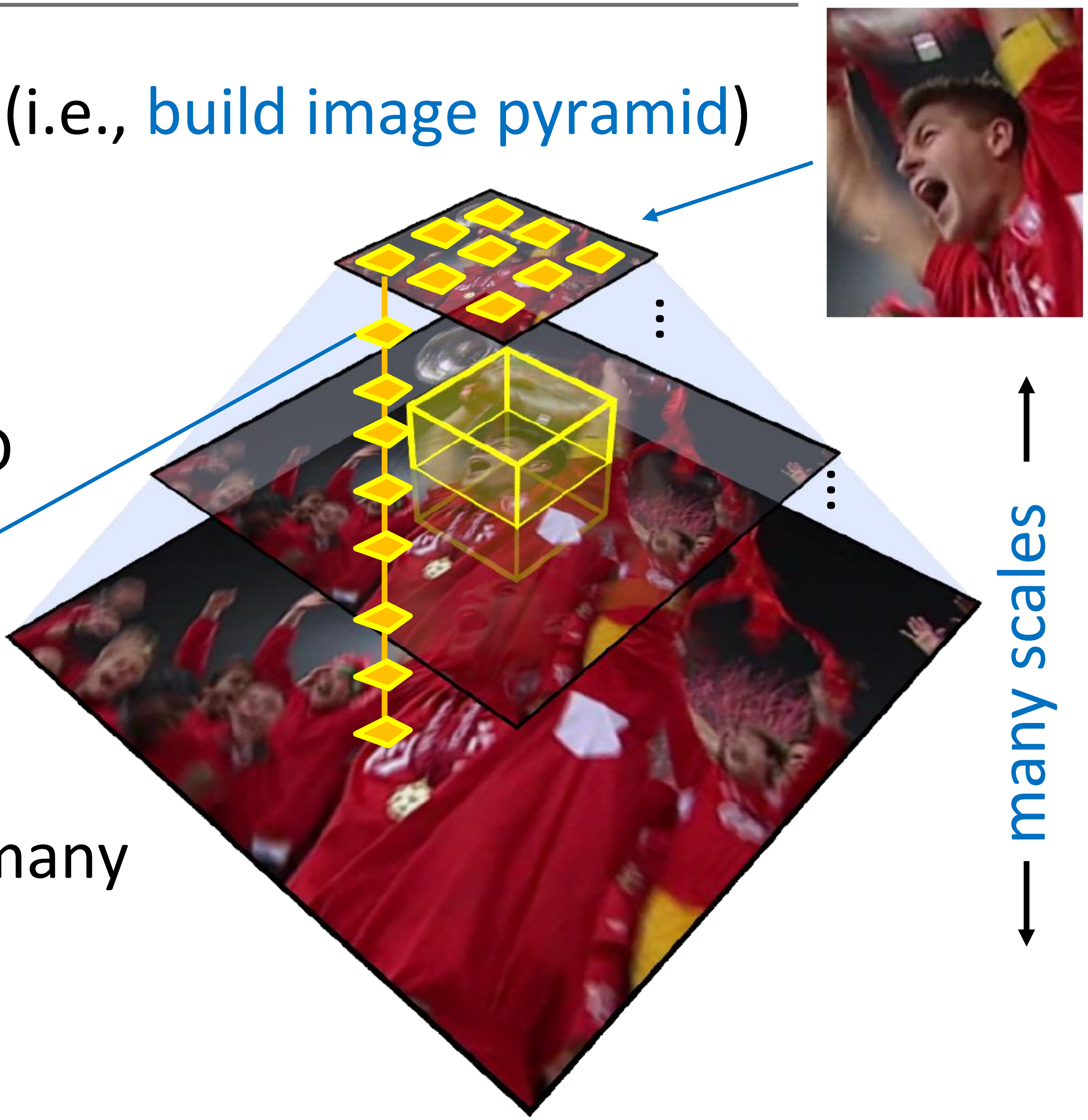
Slide credit: Jiri Matas

Scale estimation by DCF: Learning

- Resize the image patch to various sizes (i.e., build image pyramid)
- Take image intensities along each pixel through the scale-space.
- Learn a correlation filter $\boldsymbol{h}_1$ over the 1D signal



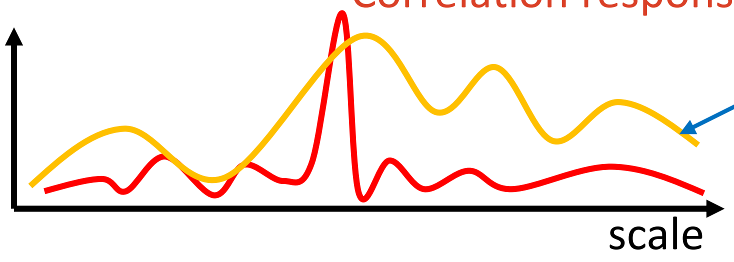
- Repeat this for all N pixels and obtain many 1D correlation filters $\{\boldsymbol{h}_i\}_{i=1:N}$.
- Multichannel version proposed in [1]



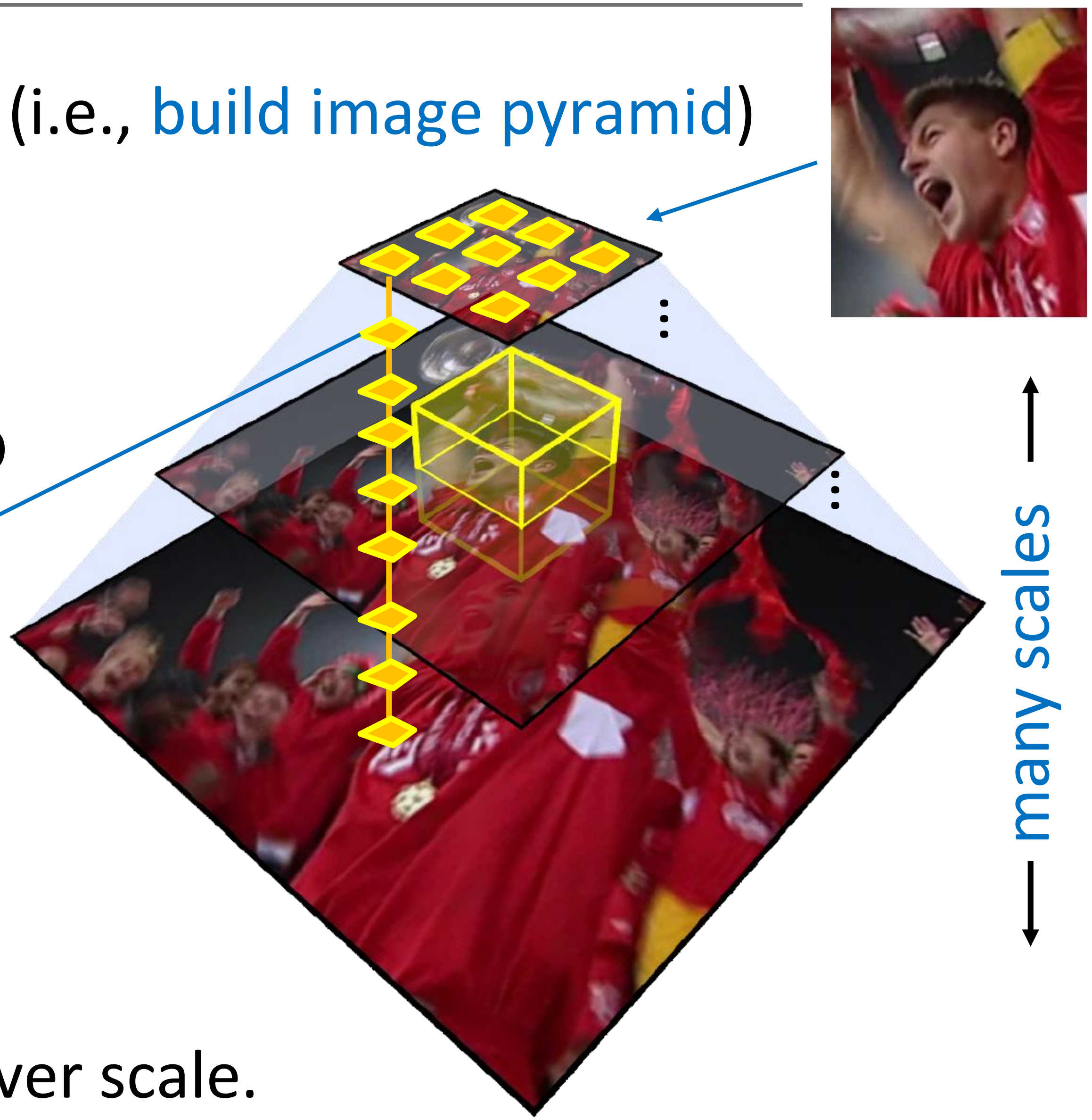
[1] Danelljan et al.,.: Accurate scale estimation for robust visual tracking. BMVC2014

Scale estimation by DCF: Estimation

- Resize the image patch to various sizes (i.e., build image pyramid)
- Take image intensities along each pixel through the scale-space.
- Apply the corresponding filter $\boldsymbol{h}_i$ on 1D signal

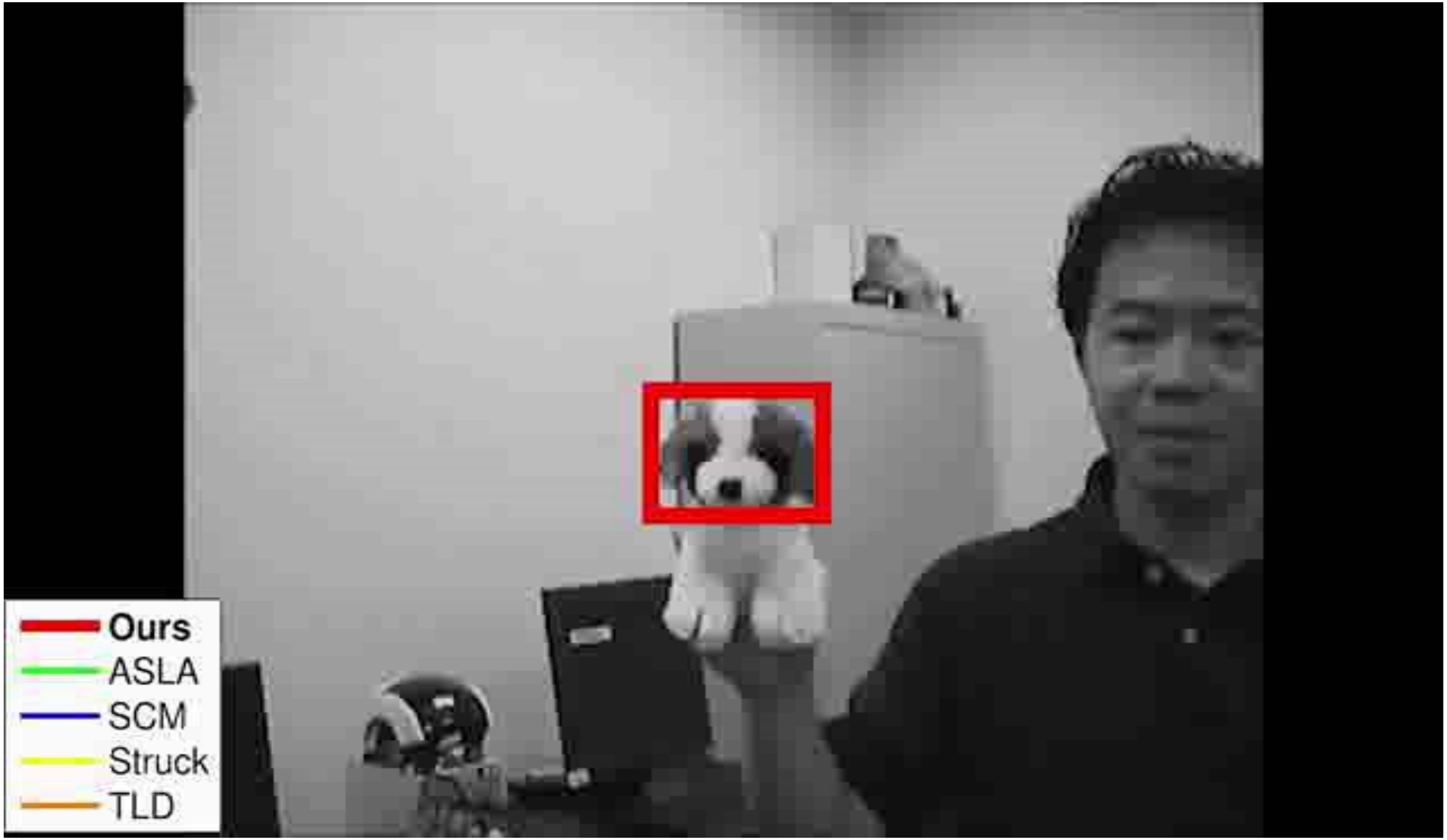


- Repeat for all 1D signals at other N locations.
- Average the responses, take the max over scale.



[1] Danelljan et al.,.: Accurate scale estimation for robust visual tracking. BMVC2014

Scale estimation by DCF



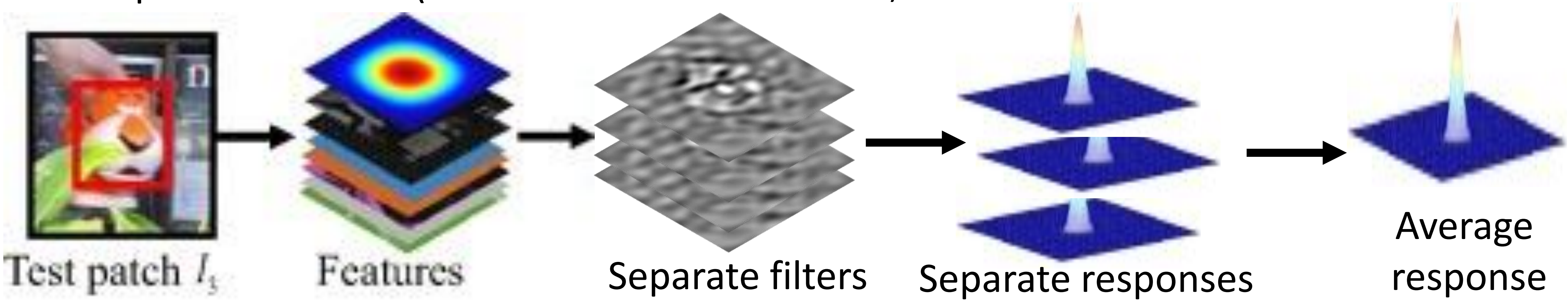
1. Localize (standard DCF)
2. Estimate scale (scale DCF)

Danelljan, M., Hager, G., Khan, F.S., Felsberg, M.: Accurate scale estimation for robust visual tracking. BMVC2014

Multichannel formulations

Multichannel formulation

- Henriques et al. – KCF (HoG 31-multi-channel features)



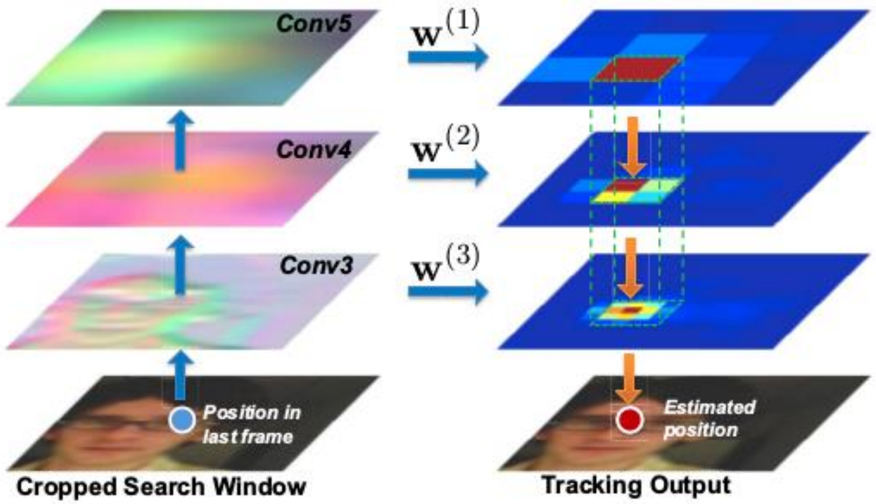
Further work

- Li et al. A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration, ECCVW2014:
  - HoG (31), color-naming (11 dimensional color representation) and grayscale pixels features
  - Quantize scale space and normalize each scale to a single (common) size by bilinear interpolation → only one filter on normalized size

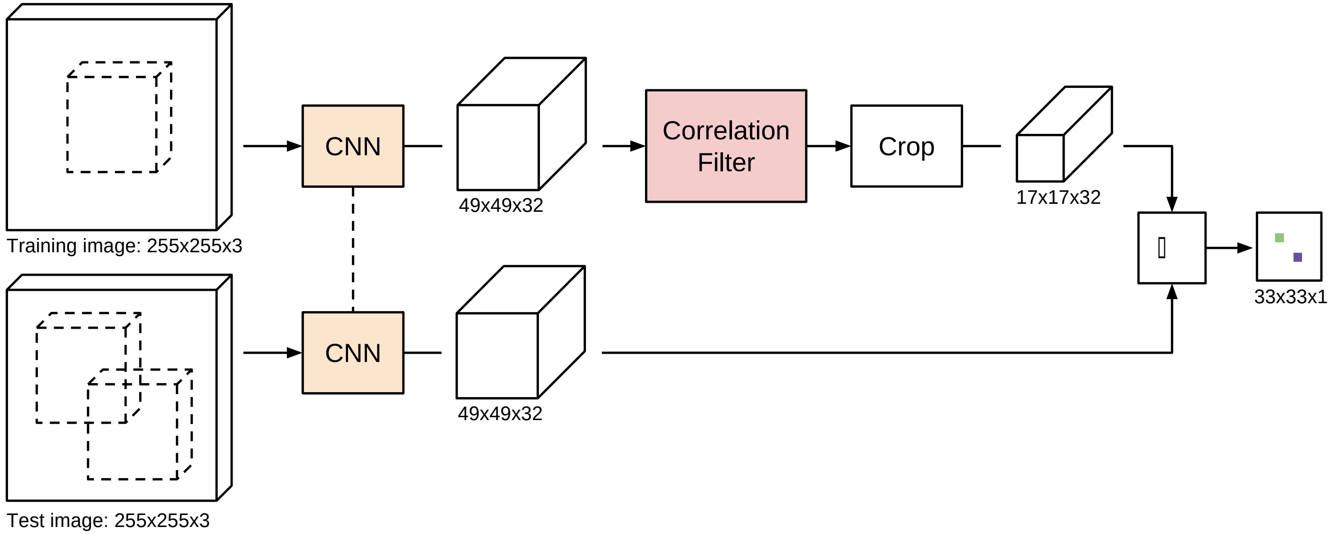
Better channel features

CNN-based Correlation Trackers

- Bhat et al. (ECCV 2018)
  Goutam Bhat et al. “Unveiling the Power of Deep Tracking”, ECCV *2018.*
    - features: VGG-Net pretrained on ImageNet dataset extracted from several layers
    - Fusion of different feature channels into a single response



- Valmadre et al. (CVPR 2017)
    - Learn CNN features for DCF



Pictures were obtained from authors publication:
- Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang, "Hierarchical Convolutional Features for Visual Tracking," International Conference on Computer Vision (ICCV), 2015.
- Valmadre, Bertinetto, Henriques, Vedaldi, Torr, End-to-end representation learning for Correlation Filter based tracking,CVPR2017

Issues with standard DCFs: search region

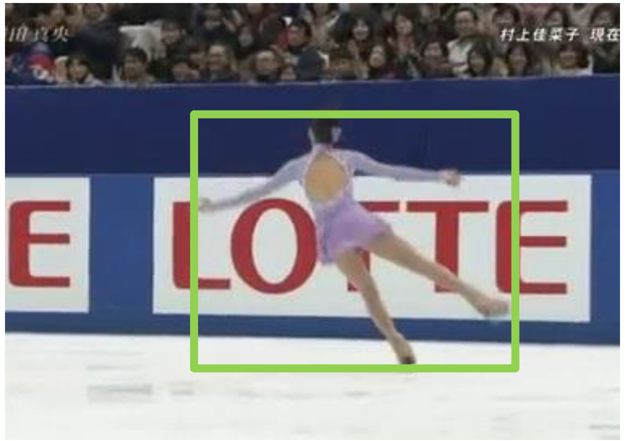
Filter learned from cyclic shifts

Unrealistic training examples

Search region size equal to filter size

Difficult to address large displacements

Poor approximation with bbox



Background enters filter

CSRDCF: Constrained filter learning

- Discriminative Correlation Filter with Channel and Spatial Reliability

  Lukežič, Čehovin, Vojir, Matas, Kristan, Discriminative Correlation Filter with Channel and Spatial Reliability, CVPR2017 (extended/updated version IJCV2019)

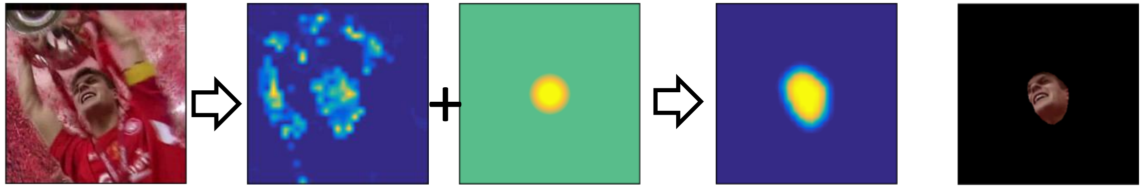
- State-of-the-art results, outperformed even trackers based on CNN
- Simple features:
  - HoG features (18 contrast sensitive orientation channels)
  - binarized grayscale channel (1 channel)
  - color names (~mapping of RGB to 10 channels)
- Single-CPU single-thread, Matlab implementation @13 fps,

  C++ realtime ; part of OpenCV & NVIDIA embedded lib; over 1.2k citations

CSRDCF outline

Training:

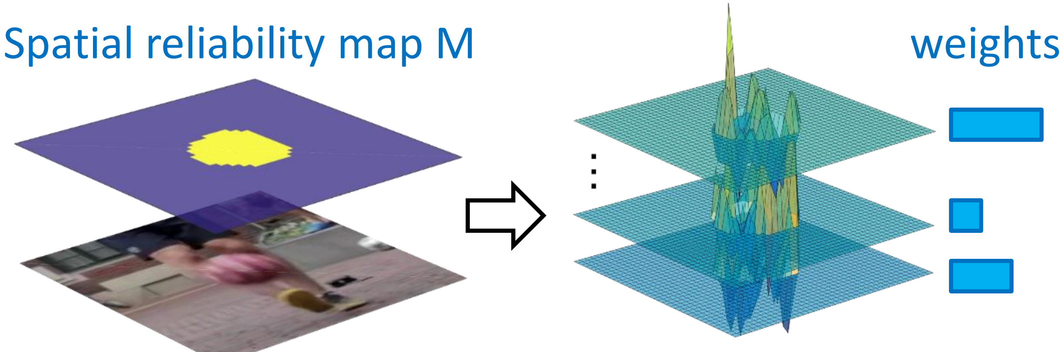
- Estimate object segmentation → object mask

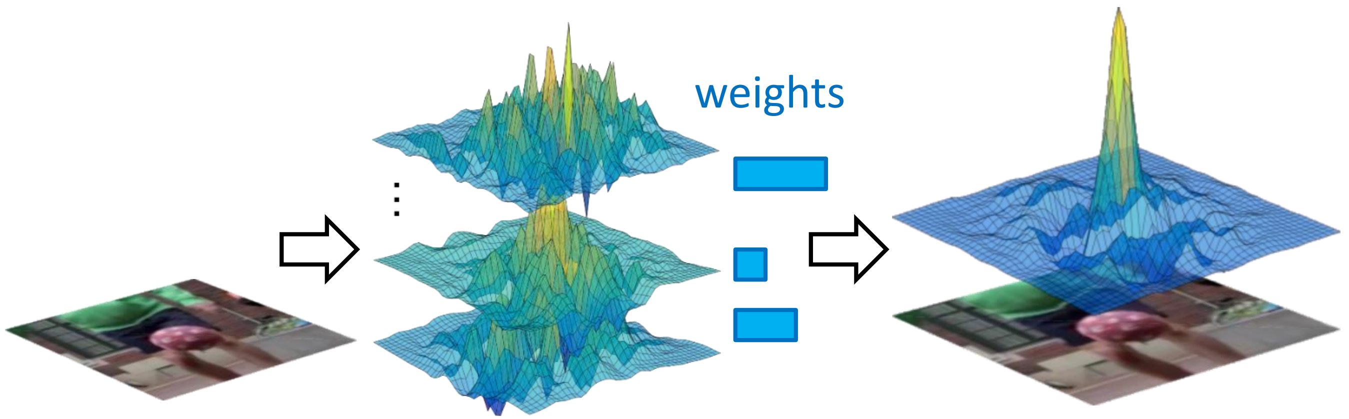


- Learn correlation filter using the object mask as constraints
- Estimate weights of the feature channels

Localization:

- Compute response map from the weighted feature channels responses
- Estimate best position
- Estimate scale (standard approach in correlation tracking)





CSRDCF computational challenges

- The cost function becomes complicated when filter masking is considered

$$\epsilon = ||\hat{\mathbf{f}} \odot \bar{\hat{\mathbf{h}}} - \hat{\mathbf{g}}||^2 + \lambda||\hat{\mathbf{h}}||^2 \ ; \ \mathbf{h} = \mathbf{h} \odot \mathbf{m}$$

- A closed-form solution does not exist, but the problem can be re-formulated and solved by Alternate Direction Method of Multipliers (ADMM).

See these papers for a practical example of ADMM uses:

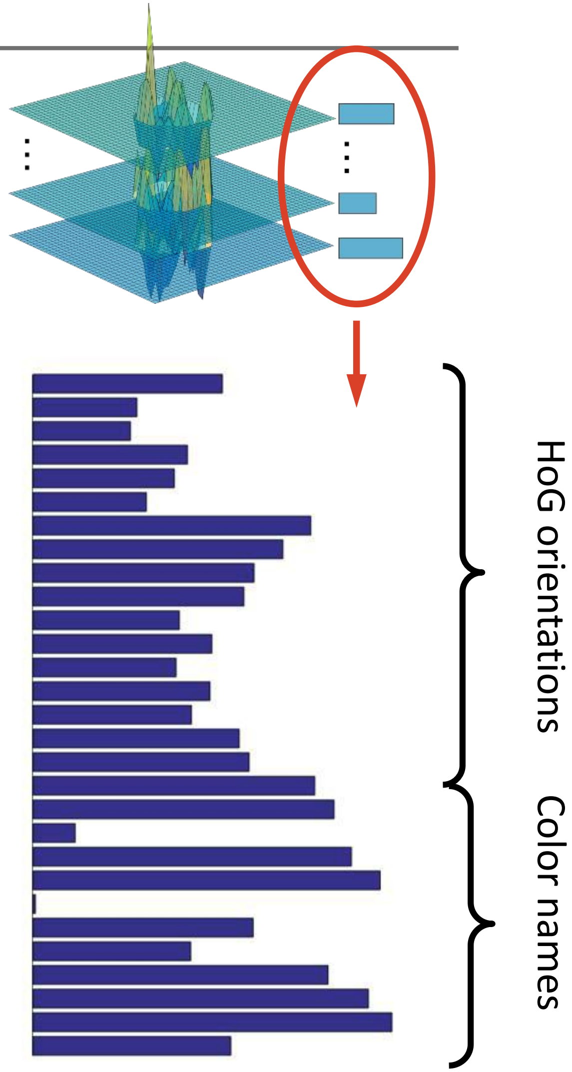
(full derivation in the appendix of [1])

[1] Lukežič, Čehovin, Vojir, Matas, Kristan, Discriminative Correlation Filter with Channel and Spatial Reliability, CVPR2017 (extended/updated version IJCV2019)

[2]Lukežič, Čehovin Zajc, Kristan, Fast Spatially Regularized Correlation Filter Tracker, ERK 2018

CSRDCF – example

Tracking result



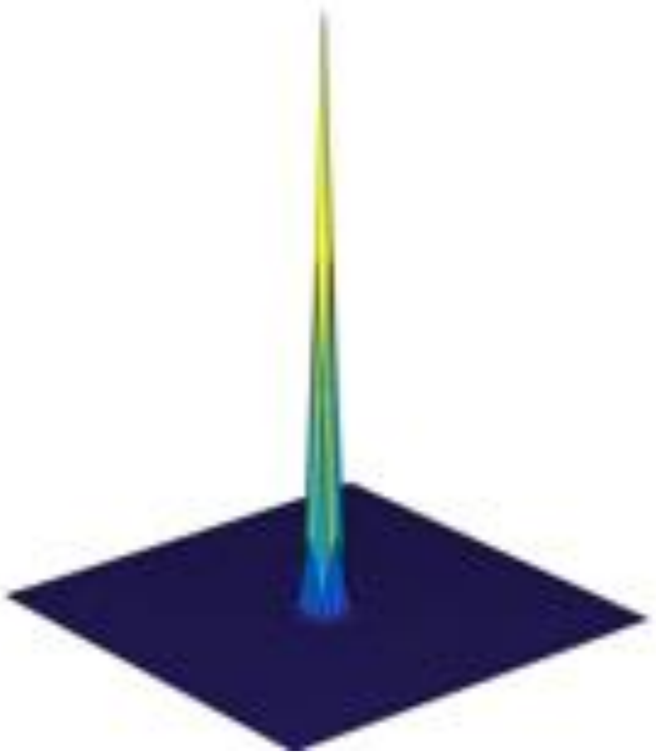
Channel reliability weights

CSRDCF – segmentation mask

Lukežič, Vojíř, Čehovin, Matas, Kristan, *Discriminative Correlation Filter with Channel and Spatial Reliability*, CVPR 2017.

CSRDCF – nonrigid target

Input image



CF response

Lukežič, Vojíř, Čehovin, Matas, Kristan, *Discriminative Correlation Filter with Channel and Spatial Reliability*, CVPR 2017.

Applications: AR.Drone [1]

Alan Lukežič, Jon Natanael Muhovič, Tina Strgar

[1] M. Kristan, Računalniški vid v avtonomnih robotskih sistemih, Noč raziskovalcev (Ljubljana, September 2015)

Alternative constrained filter learning approaches

- Constrained filter learning has been explored before:
  - [1] Danelljan, Häger, Khan, Felsberg, Learning Spatially Regularized Correlation Filters for Visual Tracking. ICCV 2015
  - [2] Hamed Kiani Galoogahi, Terence Sim, Simon Lucey, Correlation Filters with Limited Boundaries. CVPR 2015

- A followup continuous formulation:
  - [3] Martin Danelljan, Goutam Bhat, Fahad Khan, Michael Felsberg, ECO: Efficient Convolution Operators for Tracking, CVPR 2017

Discriminative tracking – summary

- Optimization technique:

  various, some in closed form, some as efficient variants of gradient descent

- Cost functions:

  Discriminative – foreground/background differentiability maximized!

- Attractive properties:
  - Potentially fast learning and fast application (e.g., ~300fps MOSE)
  - Performance may be boosted in straight-forward manner by better features.
  - Further boosts by *learning* the best features for tracking

References

- Online Adaboost for tracking:
  - H. Grabner et. al., Real-Time Tracking via On-line Boosting . BMVC, 2006.
- Multiple instance learning for tracking:
  - Babenko et al.,"Robust Object Tracking with Online Multiple Instance Learning", TPAMI2011
- Structured SVM tracking:
  - Hare, Saffari, Torr, Struck: Structured Output Tracking with Kernels, ICCV 2011
- Correlation filter tracking:
  - Bolme, Beveridge, Draper, and Y. M. Lui. Visual Object Tracking using Adaptive Correlation Filters, CVPR 2010.
  - Henriques, Caseiro, Martins, Batista, High-Speed Tracking with Kernelized Correlation Filters, TPAMI2015
  - Danelljan, M., Hager, G., Khan, F.S., Felsberg, M.: Accurate scale estimation for robust visual tracking, BMVC2014
  - Danelljan, Häger, Khan, Felsberg, Learning Spatially Regularized Correlation Filters for Visual Tracking. ICCV 2015
  - Lukežič, Vojíř, Čehovin, Matas, Kristan, *Discriminative Correlation Filter with Channel and Spatial Reliability*, CVPR 2017
  - Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang, Hierarchical Convolutional Features for Visual Tracking, ICCV 2015
  - Valmadre, et al., End-To-End Representation Learning for Correlation Filter Based Tracking, CVPR2017