Real-Time Tracking via On-line Boosting

Helmut Grabner, Michael Grabner, Horst Bischof

Graz University of Technology

Graz University of Technology

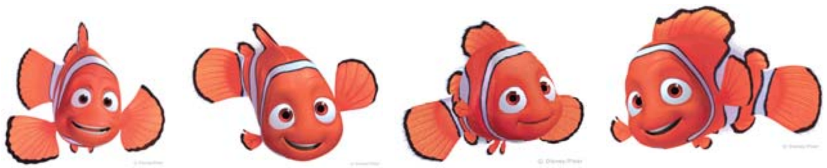
Institute for Computer Graphics and Vision

Tracking "Shrek"

M. Grabner, H. Grabner and H Bischof. **Real-time tracking with on-line feature selection.** CVPR 2006.

Tracking Requirements

- **Adaptivity**
  - Appearance changes (e.g. out of plane rotations)



- **Robustness**
  - Occlusions, cluttered background, illumination conditions

- **Generality**
  - Any object

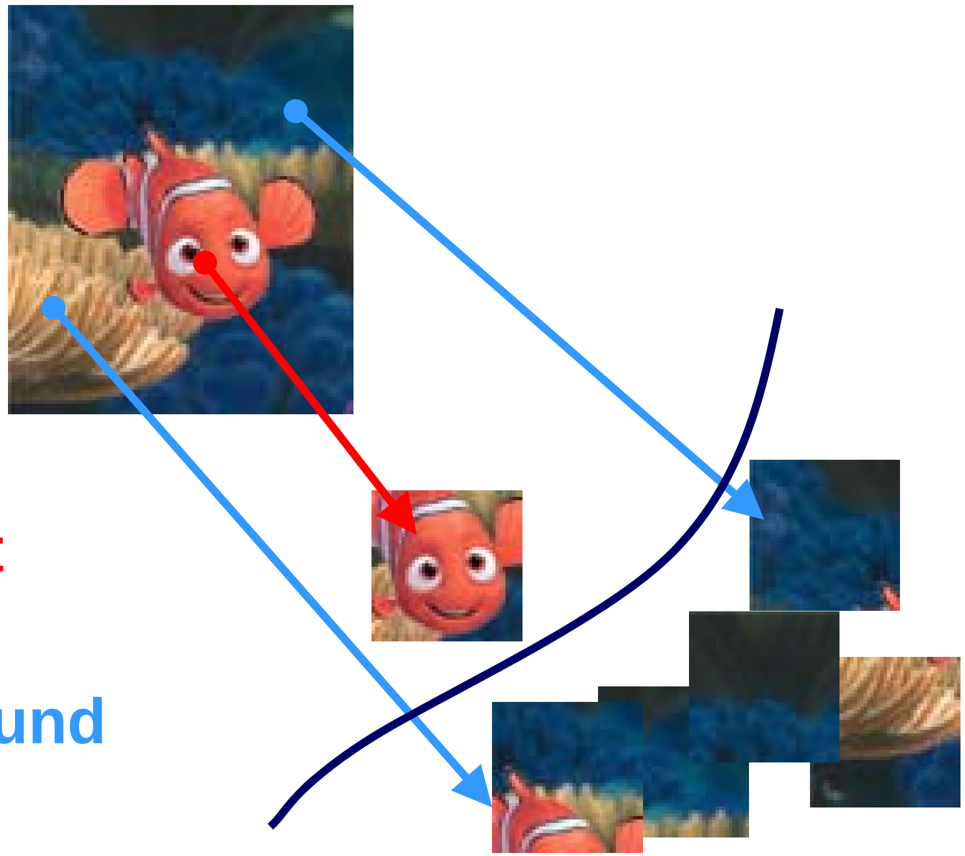
Outline

- **Tracking as Classification**
- **Boosting for Feature selection**
  - From Off-line to On-line
  - On-line Feature Selection
- **Tracking**
- **Experimental Results**
- **Conclusion**

Tracking as Classification

- **Tracking as binary classification**

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



object
vs.
background

Tracking as Classification

- **Tracking as binary classification problem**

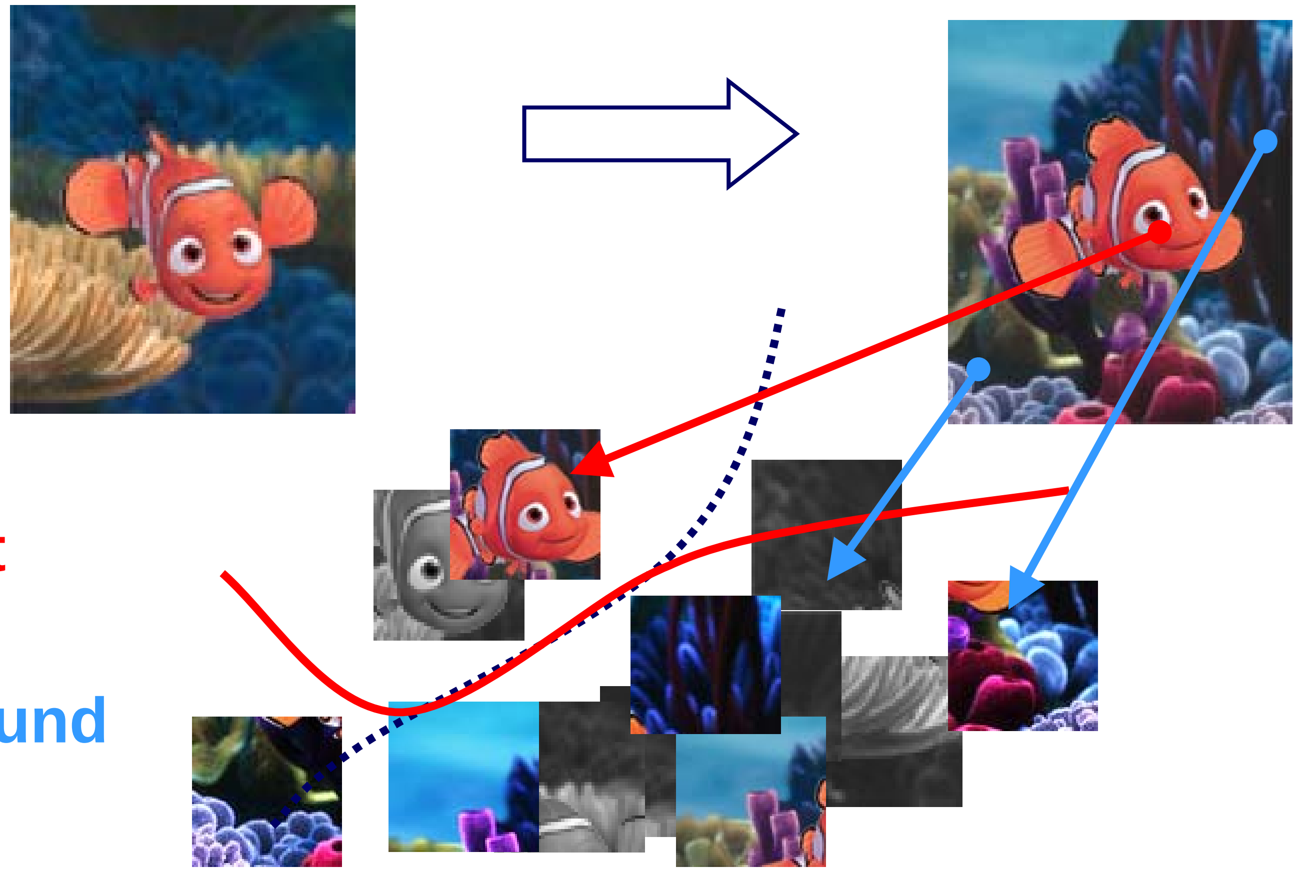
  S. Avidan. **Ensemble tracking**. CVPR 2005.
  J.Wang, et al. **Online selecting discriminative tracking features using particle filter**. CVPR 2005.

- **Object <u>and</u> background changes are robustly handled by on-line updating!**

object

vs.

background

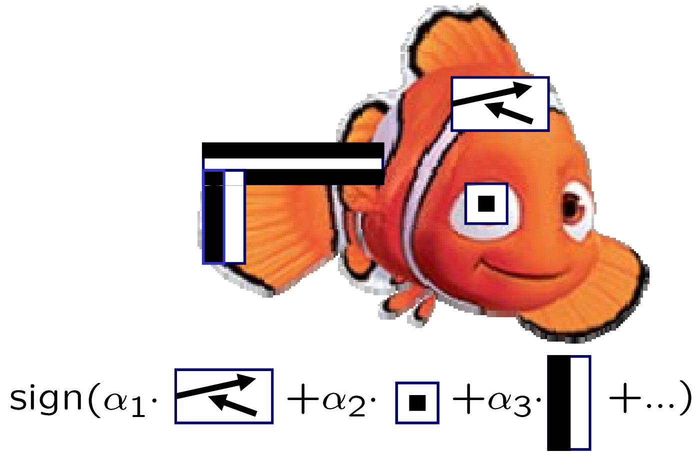


Boosting for Feature Selection

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

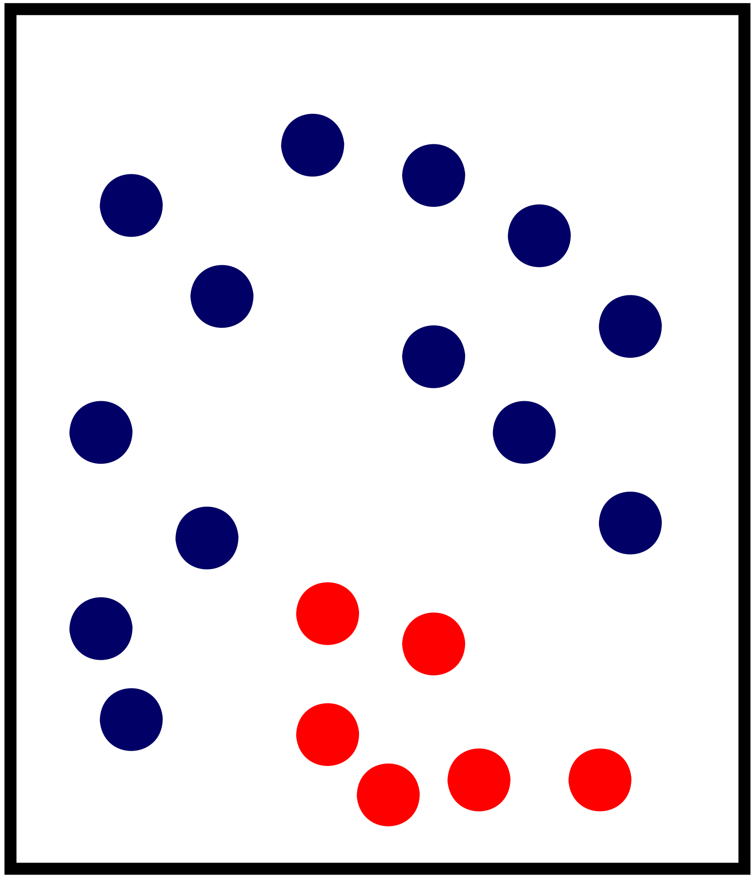
Object Tracker

On-line update
Object vs. Background

On-Line Boosting for Feature Selection

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

Off-line Boosting



Given:

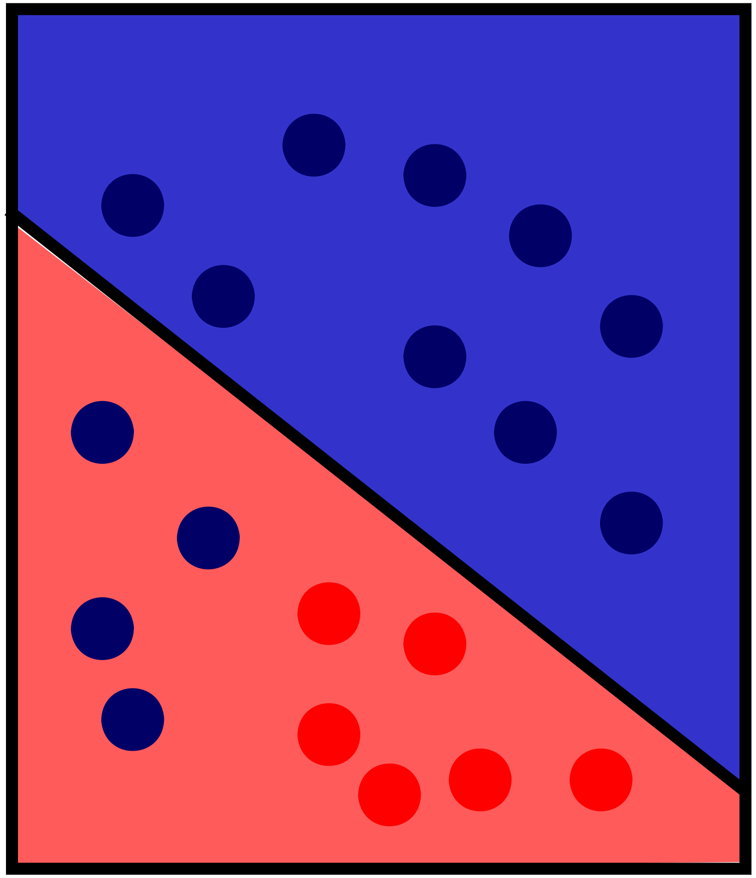
- set of labeled training samples
- weight distribution over them

Algorithm:

```
for n = 1 to N
    - train a weak classifier using
      samples and weight dist.
    - calculate error
    - calculate weight
    - update weight dist.
next
```

Y. Freund and R. Schapire. **A decision-theoretic generalization of on-line learning and an application to boosting.** Journal of Computer and System Sciences, 1997.

Off-line Boosting



Given:

- set of labeled training samples
- weight distribution over them

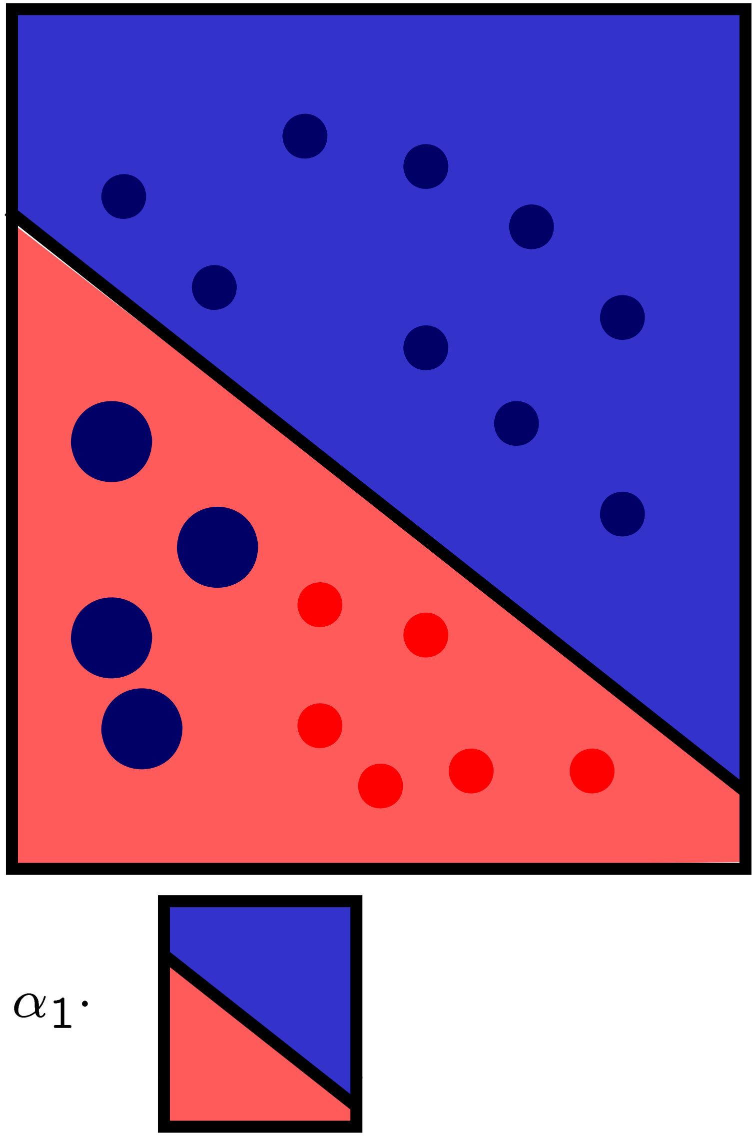
Algorithm:

for n = 1 to N

- train a weak classifier using samples and weight dist.
- calculate error
- calculate weight
- update weight dist.

next

Off-line Boosting



$\alpha_1 \cdot$

Given:

- set of labeled training samples
- weight distribution over them

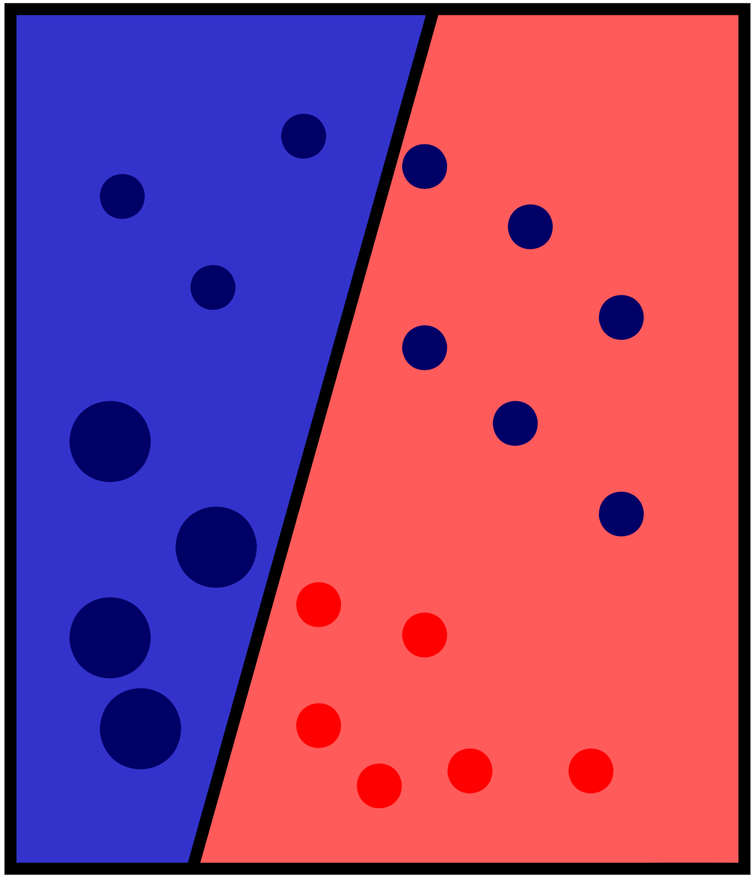
Algorithm:

for n = 1 to N

- train a weak classifier using samples and weight dist.
- calculate error
- calculate weight
- update weight dist.

next

Off-line Boosting



Given:

- **set of labeled training samples**
- **weight distribution over them**

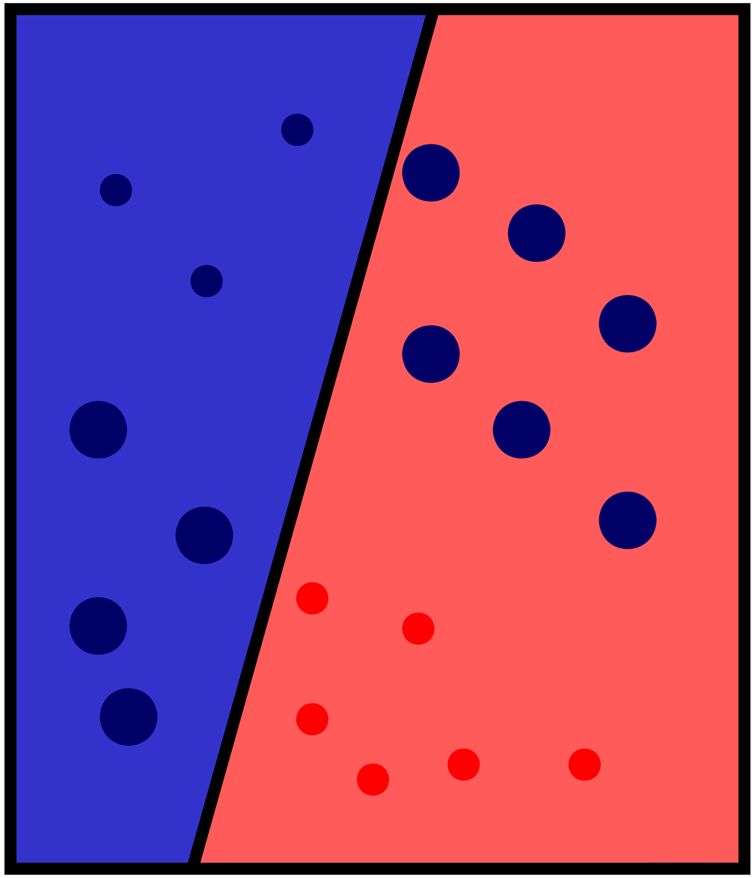
Algorithm:

for n = 1 to N

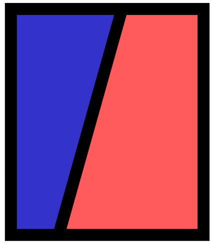
- **train a weak classifier using samples and weight dist.**
- **calculate error**
- **calculate weight**
- **update weight dist.**

next

Off-line Boosting



$\alpha_2 \cdot$



Given:

- set of labeled training samples
- weight distribution over them

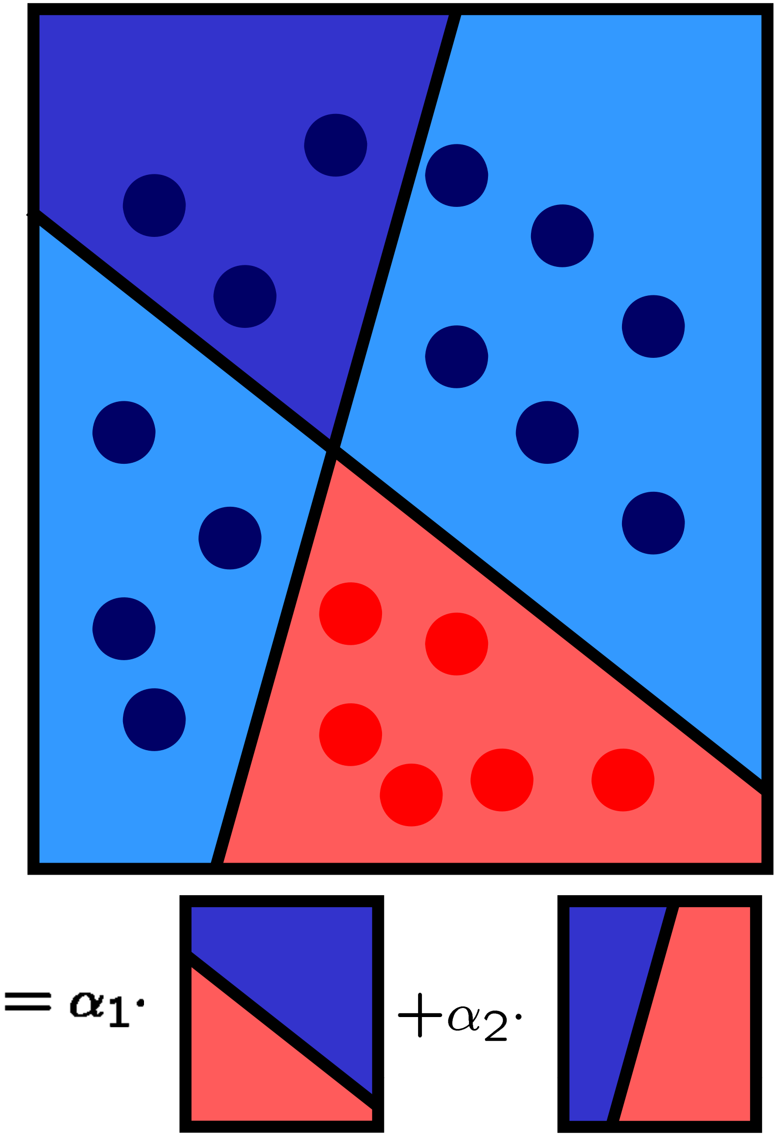
Algorithm:

for n = 1 to N

- train a weak classifier using samples and weight dist.
- calculate error
- calculate weight
- update weight dist.

next

Off-line Boosting



Given:

- set of labeled training samples
- weight distribution over them

Algorithm:

for n = 1 to N

- train a weak classifier using samples and weight dist.
- calculate error
- calculate weight
- update weight dist.

next

Result:

$$h^{strong}(\mathbf{x}) = \text{sign}\left(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(\mathbf{x}) \right)$$

From Off-line to On-line Boosting

off-line

on-line

Given:

- set of labeled training samples

$\mathcal{X} = \{\langle \mathbf{x_1}, y_1 \rangle, ..., \langle \mathbf{x_L}, y_L \rangle \mid y_i \pm 1\}$

- weight distribution over them

$D_0 = 1/L$

for n = 1 to N

- train a weak classifier using samples and weight dist.

$$h_n^{weak}(\mathbf{x}) = \mathcal{L}(\mathcal{X}, D_{n-1})$$

- calculate error e_n
- calculate weight $\alpha_n = f(e_n)$
- update weight dist. D_n

next

$$h^{strong}(\mathbf{x}) = \mathrm{sign}\left(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(\mathbf{x})\right)$$

From Off-line to On-line Boosting

off-line	on-line
Given:	**Given:**
- set of labeled training samples $\mathcal{X} = \{\langle \mathbf{x_1}, y_1 \rangle, ..., \langle \mathbf{x_L}, y_L \rangle \mid y_i \pm 1\}$	
- weight distribution over them $D_0 = 1/L$	
for n = 1 to N	**for n = 1 to N**
- train a weak classifier using samples and weight dist. $h_n^{weak}(\mathbf{x}) = \mathcal{L}(\mathcal{X}, D_{n-1})$	
- calculate error e_n	
- calculate weight $\alpha_n = f(e_n)$	
- update weight dist. D_n	
next	**next**
$h^{strong}(\mathbf{x}) = \text{sign}(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(\mathbf{x}))$	$h^{strong}(\mathbf{x}) = \text{sign}(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(\mathbf{x}))$

From Off-line to On-line Boosting

only one training example to update the classifier

off-line	on-line
Given:	**Given:**
- set of labeled training samples $\mathcal{X} = \{\langle \mathbf{x}_1, y_1 \rangle, ..., \langle \mathbf{x}_L, y_L \rangle \mid y_i \pm 1\}$	- ONE labeled training sample $\langle \mathbf{x}, y \rangle \mid y \pm 1$
- weight distribution over them $D_0 = 1/L$	- strong classifier to update
for n = 1 to N	for n = 1 to N
- train a weak classifier using samples and weight dist. $h_n^{weak}(\mathbf{x}) = \mathcal{L}(\mathcal{X}, D_{n-1})$	
- calculate error e_n	
- calculate weight $\alpha_n = f(e_n)$	
- update weight dist. D_n	
next	next
$h^{strong}(\mathbf{x}) = \text{sign}(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(\mathbf{x}))$	$h^{strong}(\mathbf{x}) = \text{sign}(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(\mathbf{x}))$

From Off-line to On-line Boosting

off-line | **update importance for the current sample** | **on-line**

Given:

- set of labeled training samples $\mathcal{X} = \{\langle \mathbf{x}_1, y_1 \rangle, ..., \langle \mathbf{x}_L, y_L \rangle \mid y_i \pm 1\}$
- weight distribution over them $D_0 = 1/L$

for n = 1 to N

- train a weak classifier using samples and weight dist.

$$h_n^{weak}(\mathbf{x}) = \mathcal{L}(\mathcal{X}, D_{n-1})$$

- calculate error e_n
- calculate weight $\alpha_n = f(e_n)$
- update weight dist. D_n

next

$$h^{strong}(\mathbf{x}) = \text{sign}\left(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(\mathbf{x}) \right)$$

Given:

- ONE labeled training sample $\langle \mathbf{x}, y \rangle \mid y \pm 1$
- strong classifier to update

- initial importance $\lambda = 1$

for n = 1 to N

- update importance weight λ

next

$$h^{strong}(\mathbf{x}) = \text{sign}\left(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(\mathbf{x}) \right)$$

From Off-line to On-line Boosting

online update the weak classifier

off-line	on-line
Given:	**Given**:
- set of labeled training samples $\mathcal{X} = \{\langle \mathbf{x}_1, y_1\rangle, ..., \langle \mathbf{x}_L, y_L\rangle \mid y_i \pm 1\}$	- ONE labeled training sample $\langle \mathbf{x}, y\rangle \mid y \pm 1$
- weight distribution over them $D_0 = 1/L$	- strong classifier to update
	- initial importance $\lambda = 1$
for n = 1 to N	for n = 1 to N
- train a weak classifier using samples and weight dist. $h_n^{weak}(\mathbf{x}) = \mathcal{L}(\mathcal{X}, D_{n-1})$	- update the weak classifier using samples and importance $h_n^{weak}(\mathbf{x}) = \mathcal{L}(h_n^{weak}, \langle \mathbf{x}, y\rangle, \lambda)$
- calculate error e_n	
- calculate weight $\alpha_n = f(e_n)$	
- update weight dist. D_n	- update importance weight λ
next	next
$h^{strong}(\mathbf{x}) = \text{sign}(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(\mathbf{x}))$	$h^{strong}(\mathbf{x}) = \text{sign}(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(\mathbf{x}))$

From Off-line to On-line Boosting

update errors and weights

off-line	on-line
Given:	**Given:**
- set of labeled training samples $\mathcal{X} = \{\langle \mathbf{x}_1, y_1 \rangle, ..., \langle \mathbf{x}_L, y_L \rangle \mid y_i \pm 1\}$	- ONE labeled training sample $\langle \mathbf{x}, y \rangle \mid y \pm 1$
- weight distribution over them $D_0 = 1/L$	- strong classifier to update
	- initial importance $\lambda = 1$
for n = 1 to N	for n = 1 to N
- train a weak classifier using samples and weight dist. $h_n^{weak}(\mathbf{x}) = \mathcal{L}(\mathcal{X}, D_{n-1})$	- update the weak classifier using samples and importance $h_n^{weak}(\mathbf{x}) = \mathcal{L}(h_n^{weak}, \langle \mathbf{x}, y \rangle, \lambda)$
- calculate error e_n	- update error estimation $\widehat{e_n}$
- calculate weight $\alpha_n = f(e_n)$	- update weight $\alpha_n = f(\hat{e}_n)$
- update weight dist. D_n	- update importance weight λ
next	next
$h^{strong}(\mathbf{x}) = \text{sign}(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(\mathbf{x}))$	$h^{strong}(\mathbf{x}) = \text{sign}(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(\mathbf{x}))$

From Off-line to On-line Boosting

off-line

Given:

- set of labeled training samples

$\mathcal{X} = \{\langle \mathbf{x}_1, y_1 \rangle, ..., \langle \mathbf{x}_L, y_L \rangle \mid y_i \pm 1\}$

- weight distribution over them

$D_0 = 1/L$

for n = 1 to N

- train a weak classifier using samples and weight dist.

$$h_n^{weak}(\mathbf{x}) = \mathcal{L}(\mathcal{X}, D_{n-1})$$

- calculate error e_n
- calculate weight $\alpha_n = f(e_n)$
- update weight dist. D_n

next

$$h^{strong}(\mathbf{x}) = \text{sign}\left(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(\mathbf{x})\right)$$

on-line

Given:

- ONE labeled training sample

$\langle \mathbf{x}, y \rangle \mid y \pm 1$

- strong classifier to update

- initial importance $\lambda = 1$

for n = 1 to N

- update the weak classifier using samples and importance

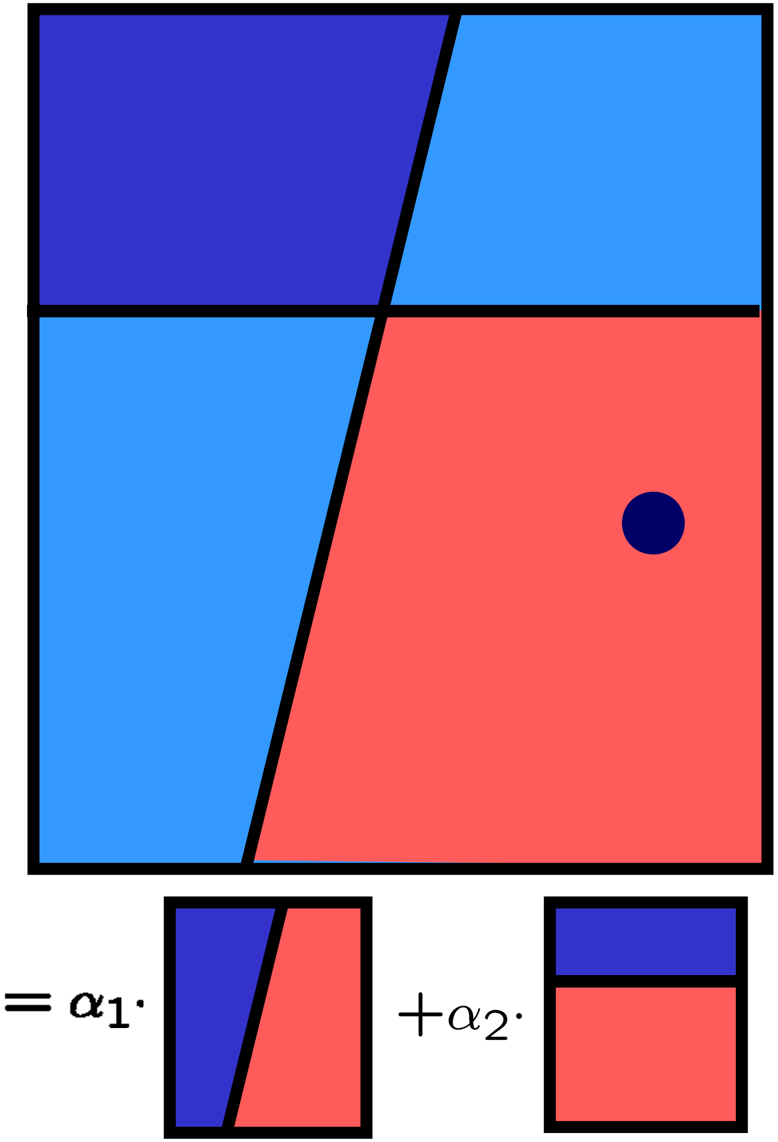
$$h_n^{weak}(\mathbf{x}) = \mathcal{L}(h_n^{weak}, \langle \mathbf{x}, y \rangle, \lambda)$$

- update error estimation $\widehat{e_n}$
- update weight $\alpha_n = f(\widehat{e}_n)$
- update importance weight λ

next

$$h^{strong}(\mathbf{x}) = \text{sign}\left(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(\mathbf{x})\right)$$

On-line Boosting



$$= \alpha_1 \cdot \quad + \alpha_2 \cdot$$

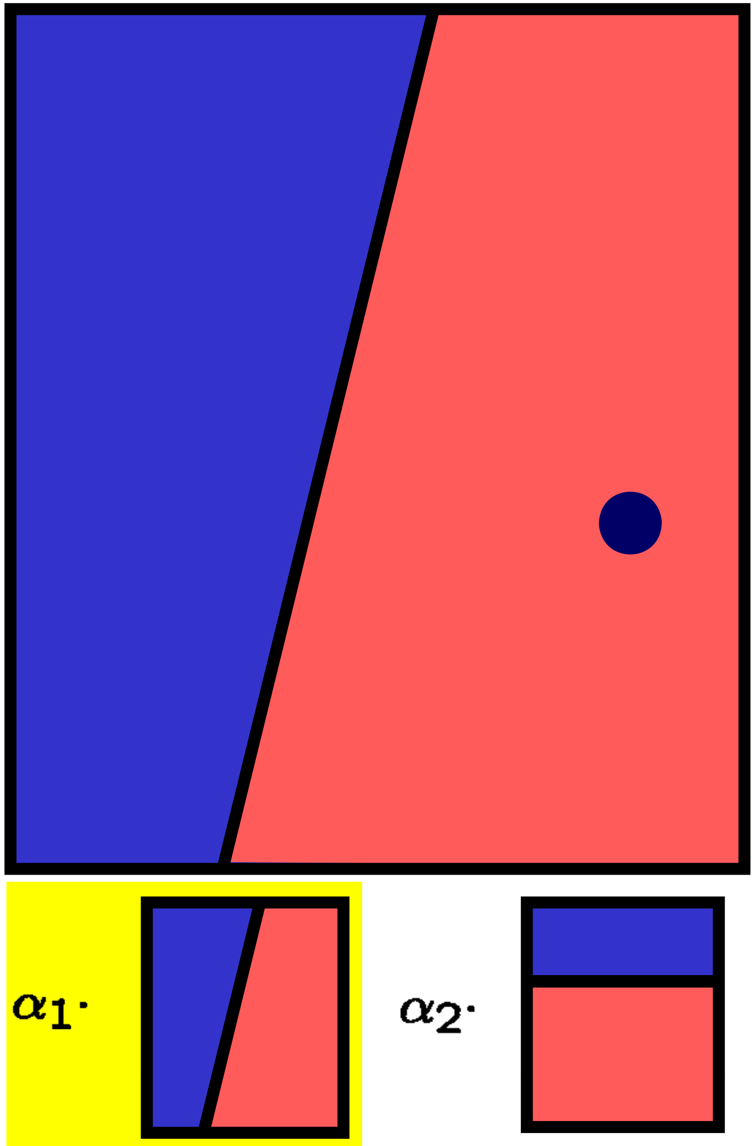
Given:

- **ONE labeled training sample**
- **strong classifier to update**

Algorithm:

```
- initial importance
for n = 1 to N
    - update the weak classifier using
      sample and importance
    - update error estimation
    - update weight
    - update importance weight
next
```

On-line Boosting



Given:

- ONE labeled training sample
- strong classifier to update

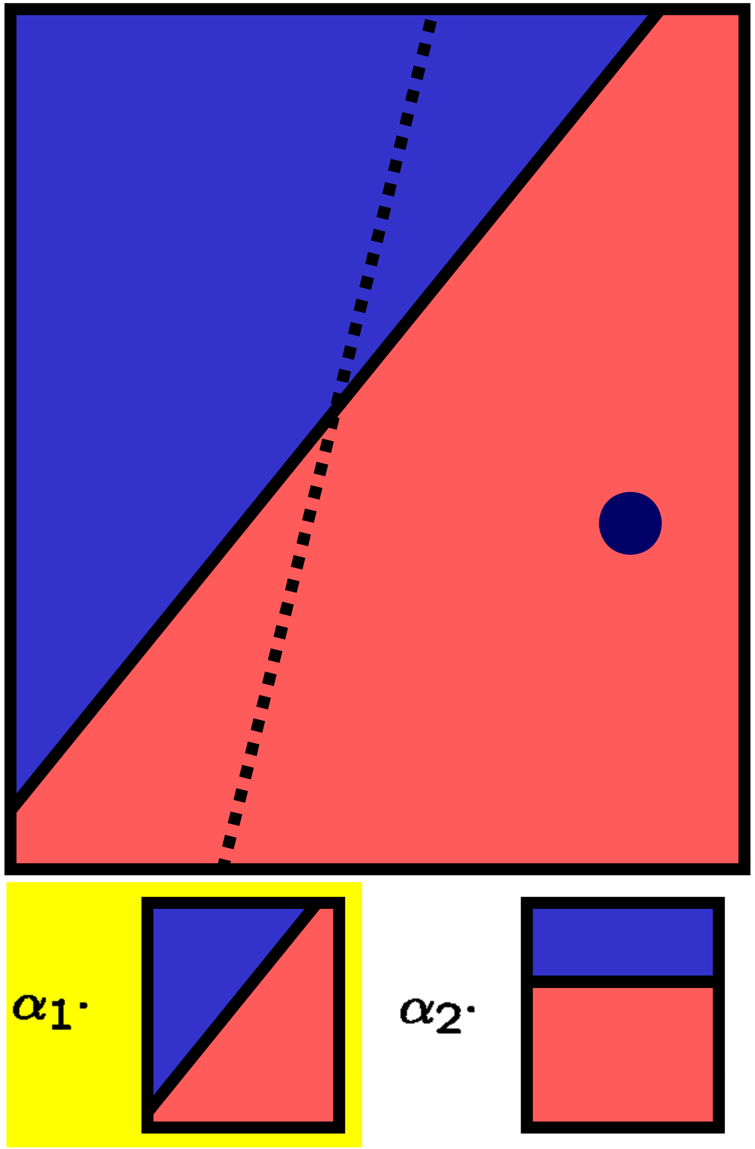
Algorithm:

- initial importance

for n = 1 to N

- update the weak classifier using sample and importance
- update error estimation
- update weight
- update importance weight

next



Given:

- ONE labeled training sample
- strong classifier to update

Algorithm:

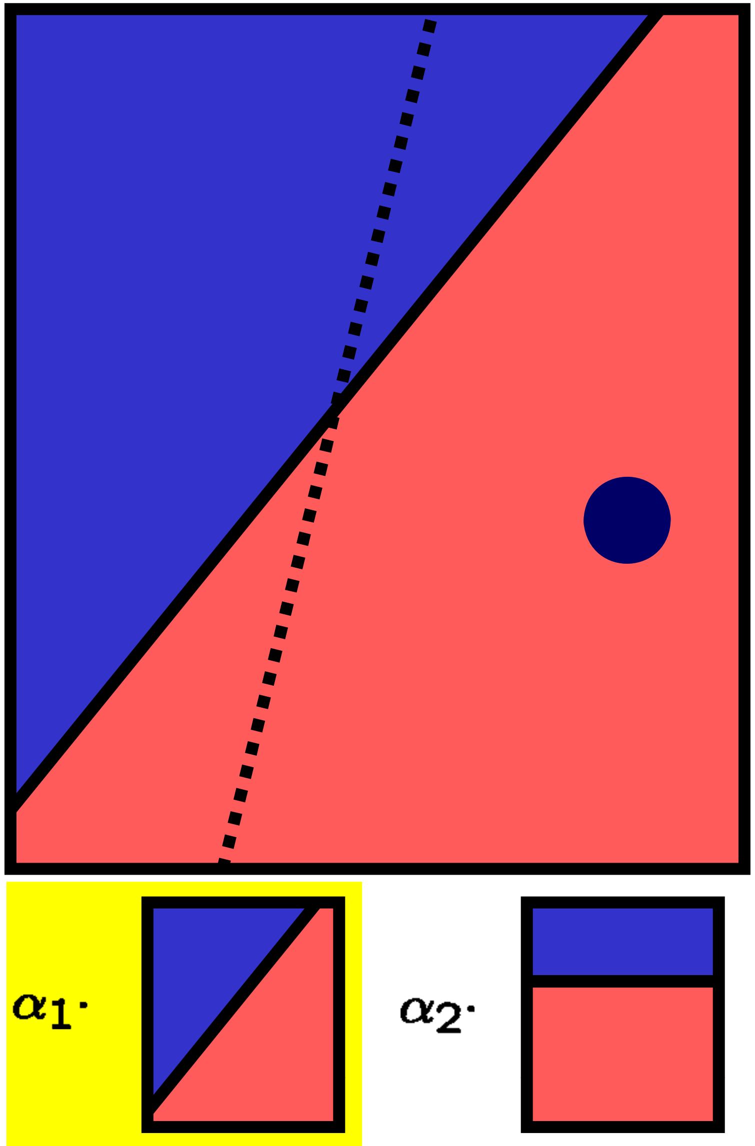
- initial importance

for n = 1 to N

- update the weak classifier using sample and importance
- update error estimation
- update weight
- update importance weight

next

On-line Boosting



Given:

- ONE labeled training sample
- strong classifier to update

Algorithm:

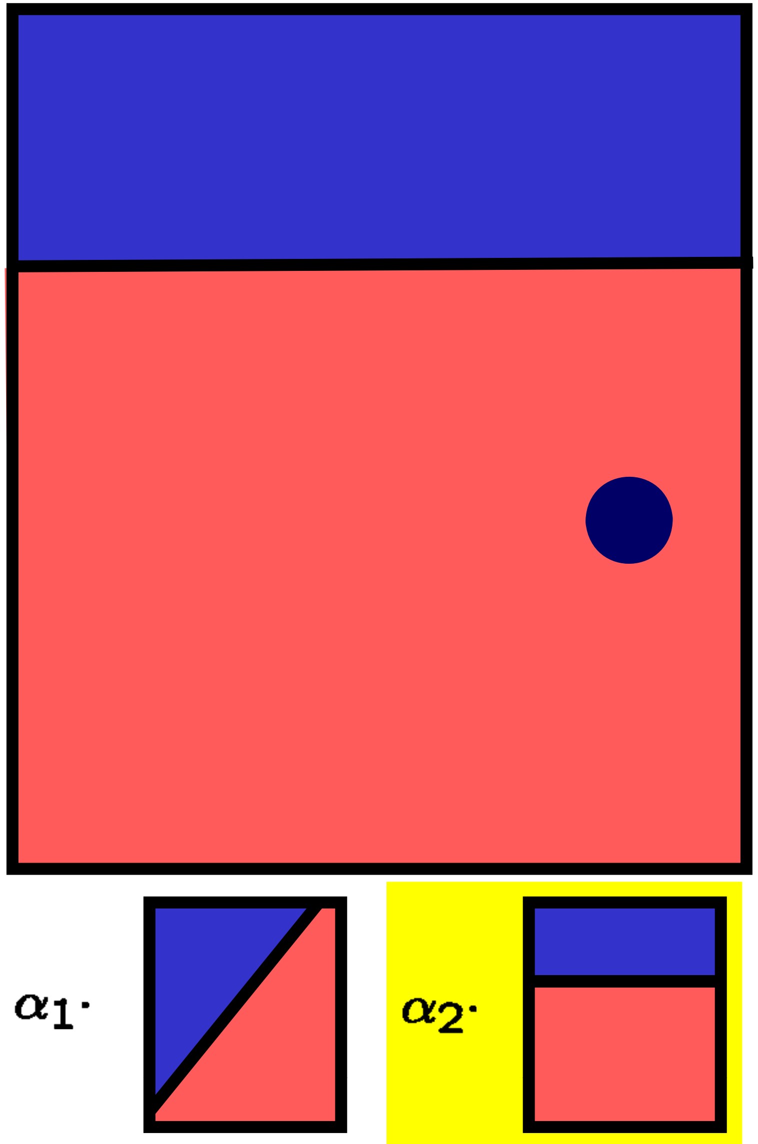
- initial importance

for n = 1 to N

- update the weak classifier using sample and importance
- update error estimation
- update weight
- update importance weight

next

On-line Boosting



Given:

- ONE labeled training sample
- strong classifier to update

Algorithm:

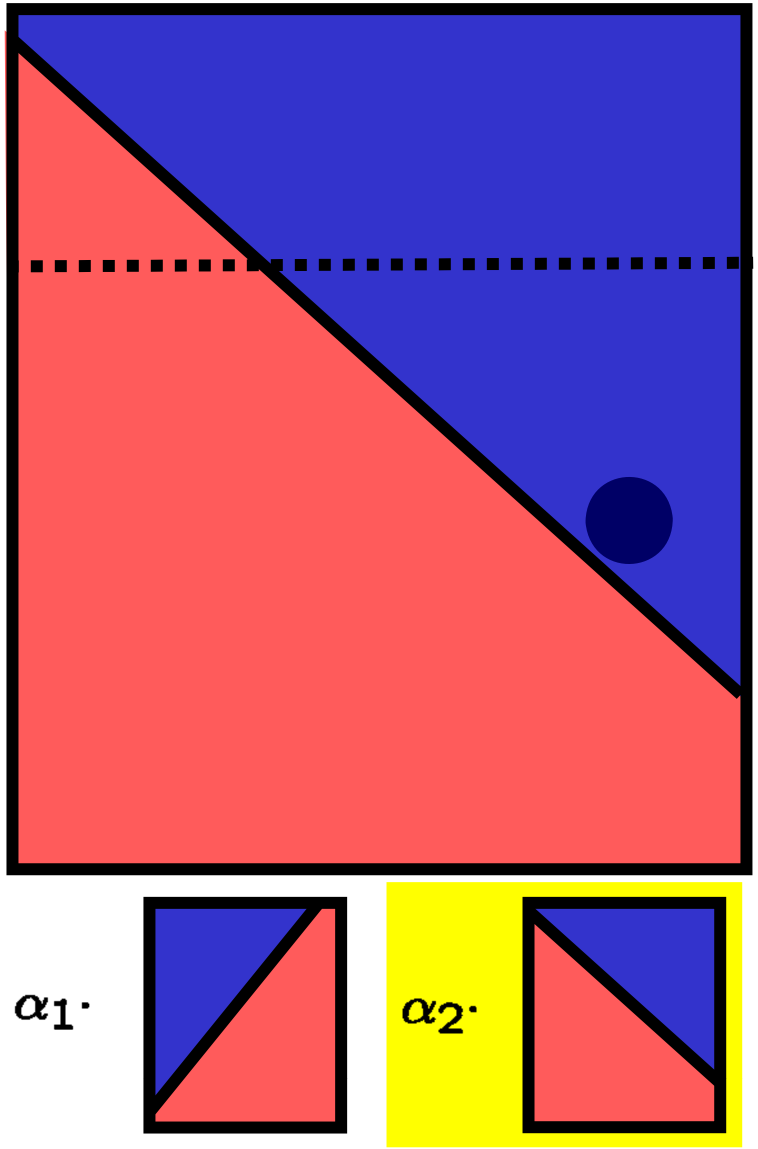
- initial importance

for n = 1 to N

- update the weak classifier using sample and importance
- update error estimation
- update weight
- update importance weight

next

On-line Boosting



Given:

- ONE labeled training sample
- strong classifier to update

Algorithm:

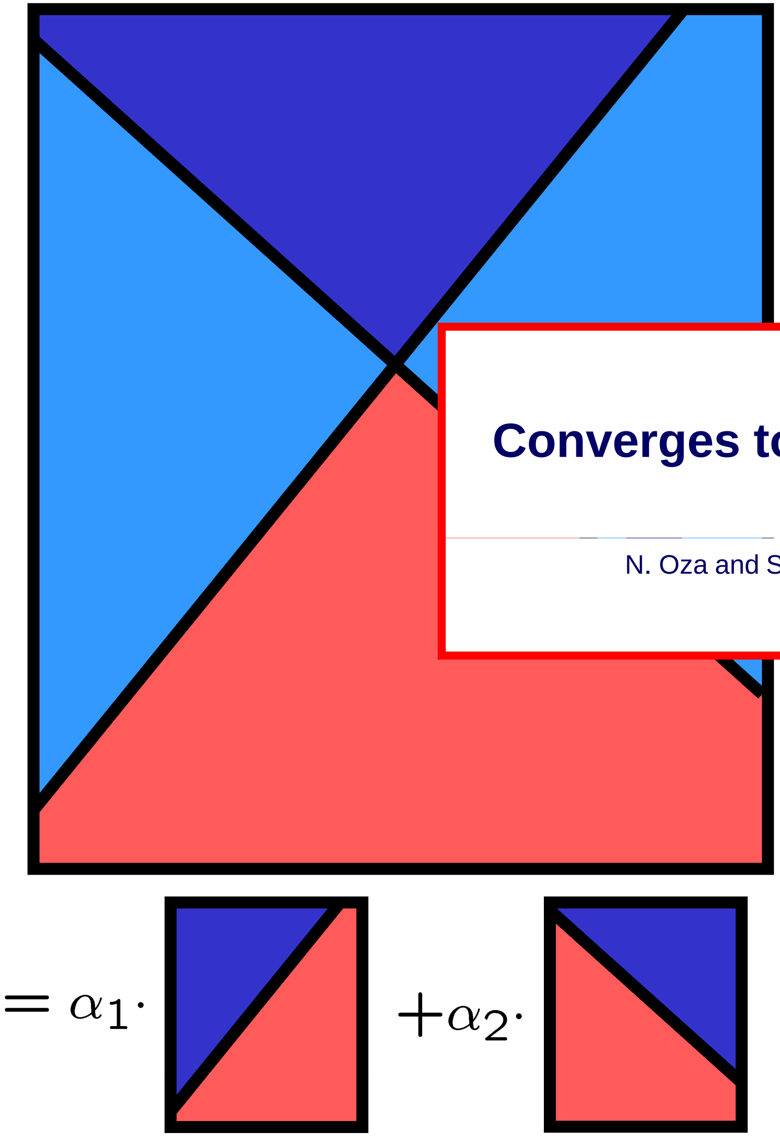
- initial importance

for n = 1 to N

- update the weak classifier using sample and importance
- update error estimation
- update weight
- update importance weight

next

On-line Boosting



Given:

- ONE labeled training sample
- strong classifier to update

Algorithm:

- initial importance

k classifier using
ortance
stimation

update importance weight

next

Converges to the off-line results...

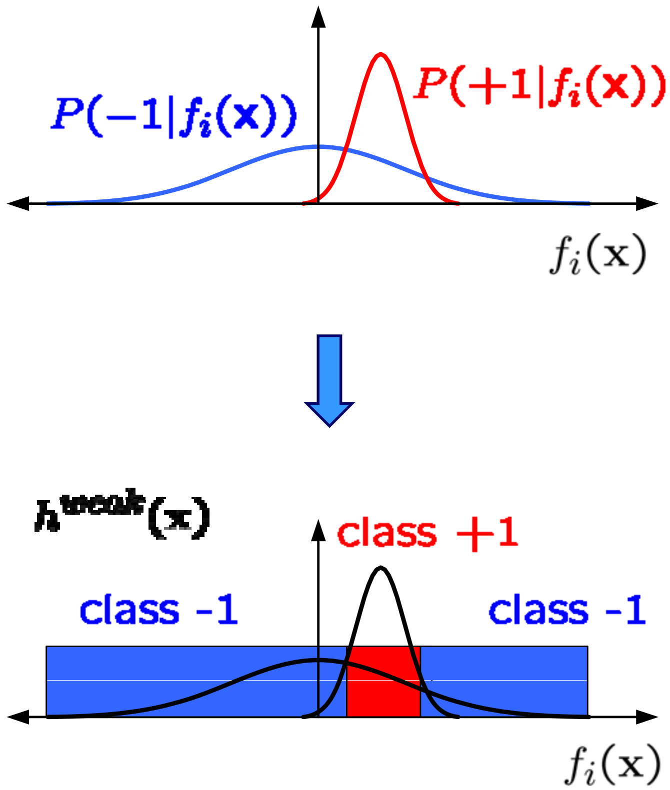
N. Oza and S. Russell. **Online bagging and boosting**. Artificial Intelligence and Statistics, 2001.

Result:

$$h^{strong}(\mathbf{x}) = \text{sign}\left(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(\mathbf{x})\right)$$

On-line Boosting for Feature Selection 1/3

- **Each feature corresponds to a weak classifier**



- **Features**
  - Haar-like wavelets
  - Orientation histograms
  - Locally binary patterns (LBP)

- **Fast computation using efficient data structures**
  - integral images
  - integral histograms

F. Porikli. **Integral histogram: A fast way to extract histograms in cartesian spaces.** CVPR 2005.

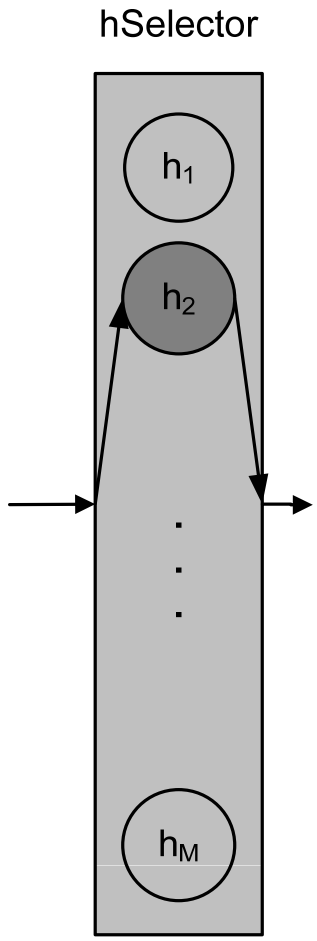
On-line Boosting for Feature Selection 2/3

- **Introducing “Selector”**
  - selects one feature from its local feature pool

$$\mathcal{H}^{weak} = \{h_1^{weak}, ..., h_M^{weak}\}$$
$$\mathcal{F} = \{f_1, ..., f_M\}$$

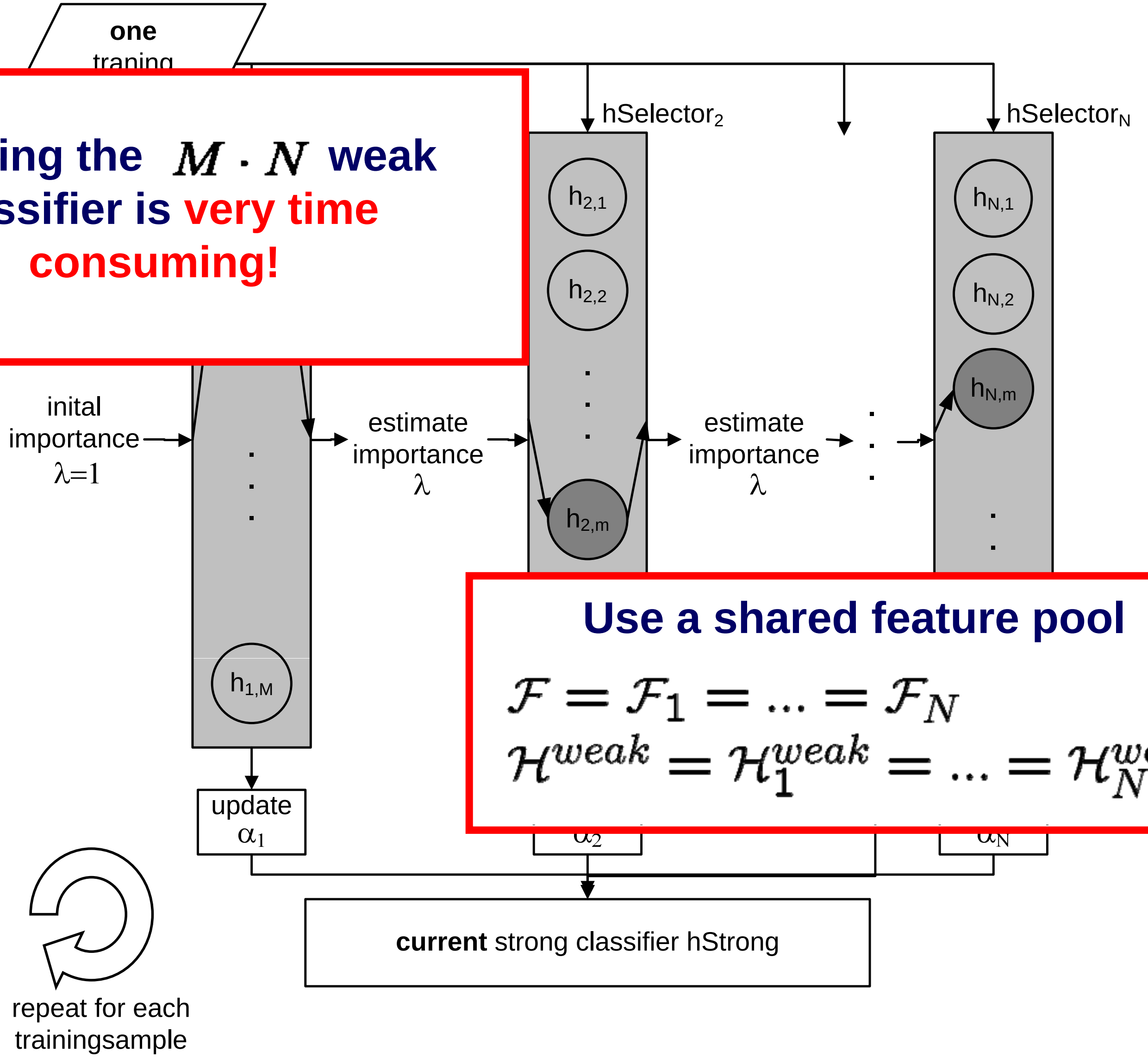
$$h^{sel}(\mathbf{x}) = h_m^{weak}(\mathbf{x})$$
$$m = \arg\min_i e_i$$

On-line boosting is performed on the Selectors and not on the weak classifiers directly.



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

On-line Boosting for Feature Selection 3/3



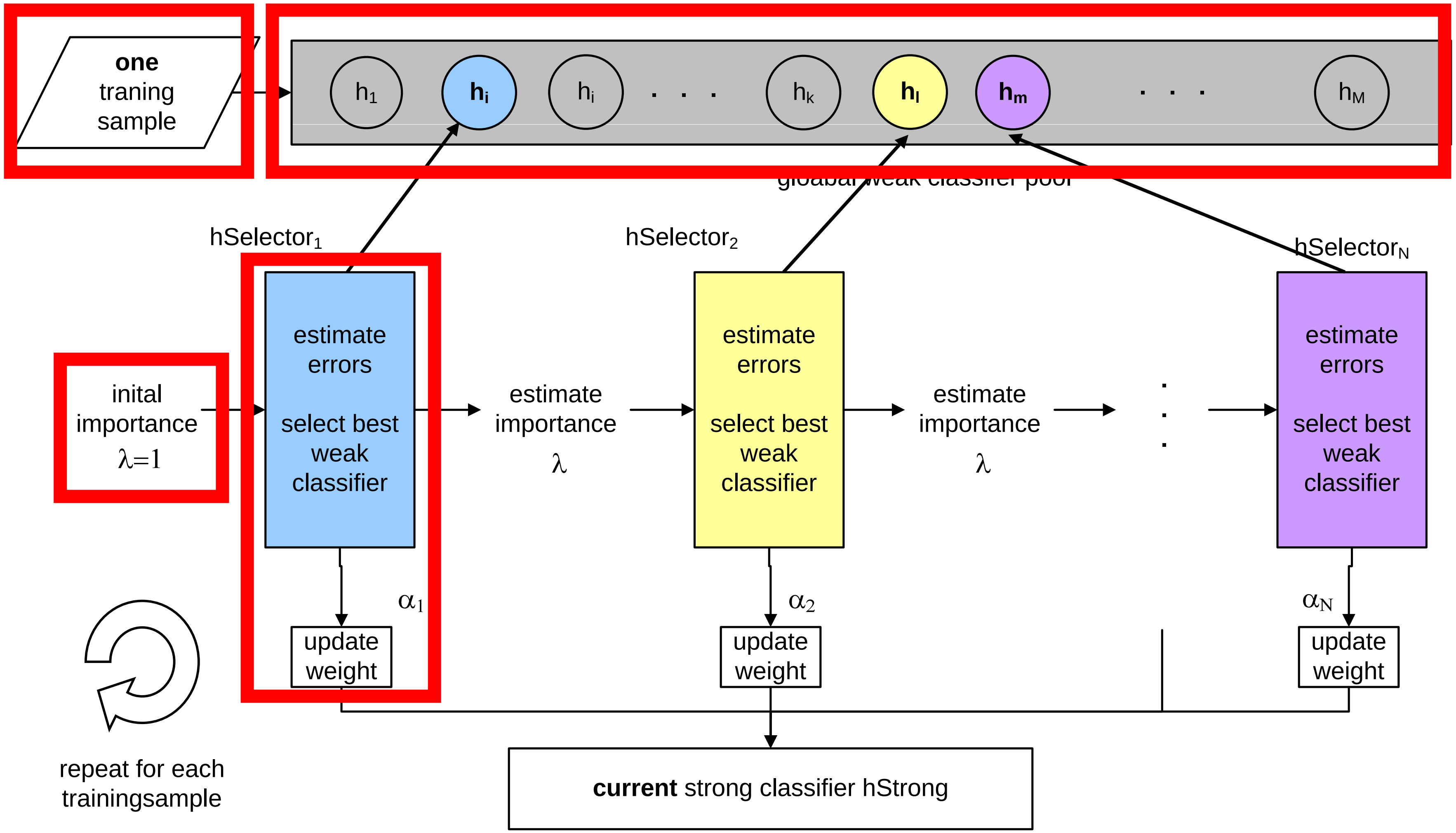
Updating the $M \cdot N$ weak classifier is very time consuming!

Use a shared feature pool

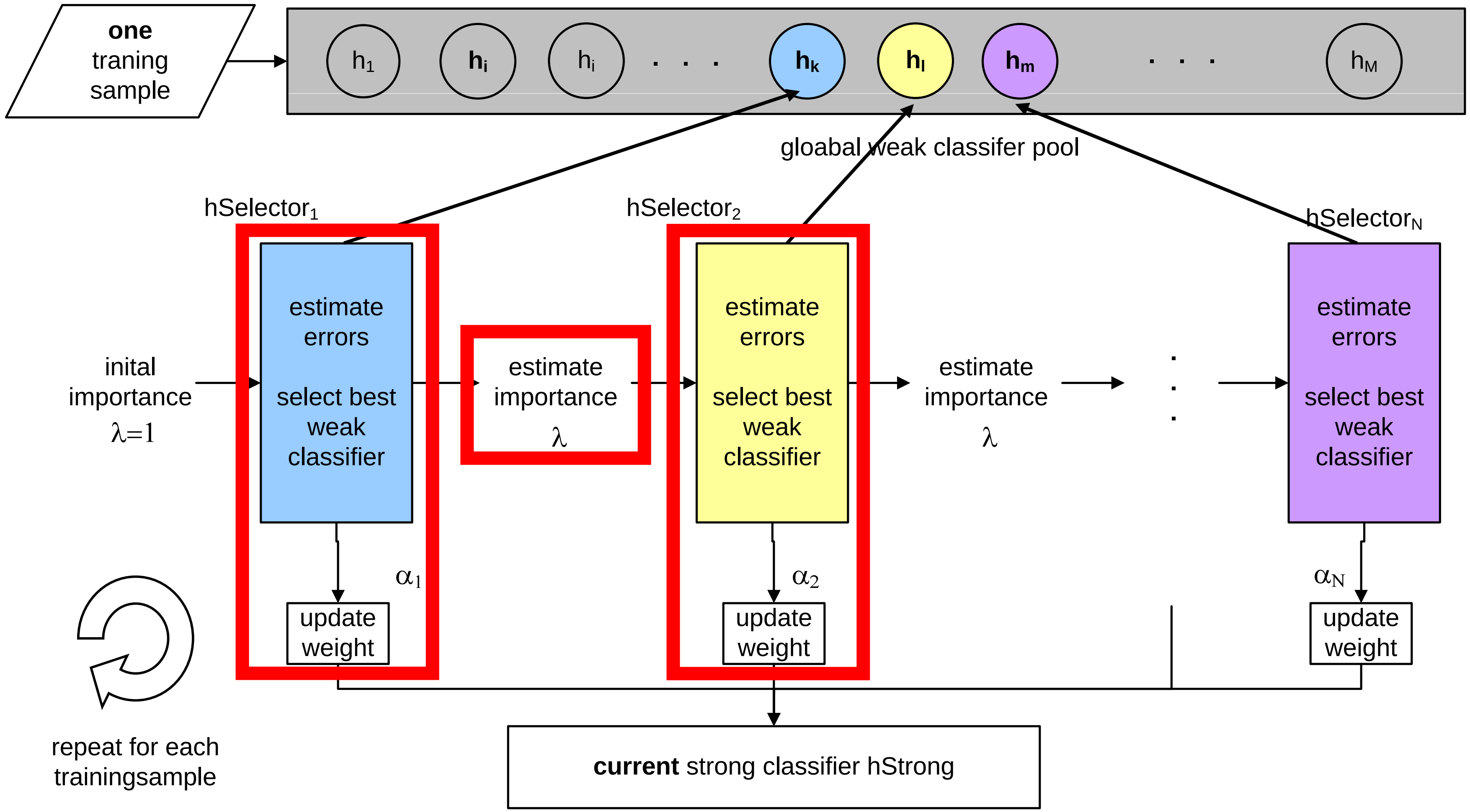
$$\mathcal{F} = \mathcal{F}_1 = \ldots = \mathcal{F}_N$$

$$\mathcal{H}^{weak} = \mathcal{H}_1^{weak} = \ldots = \mathcal{H}_N^{weak}$$

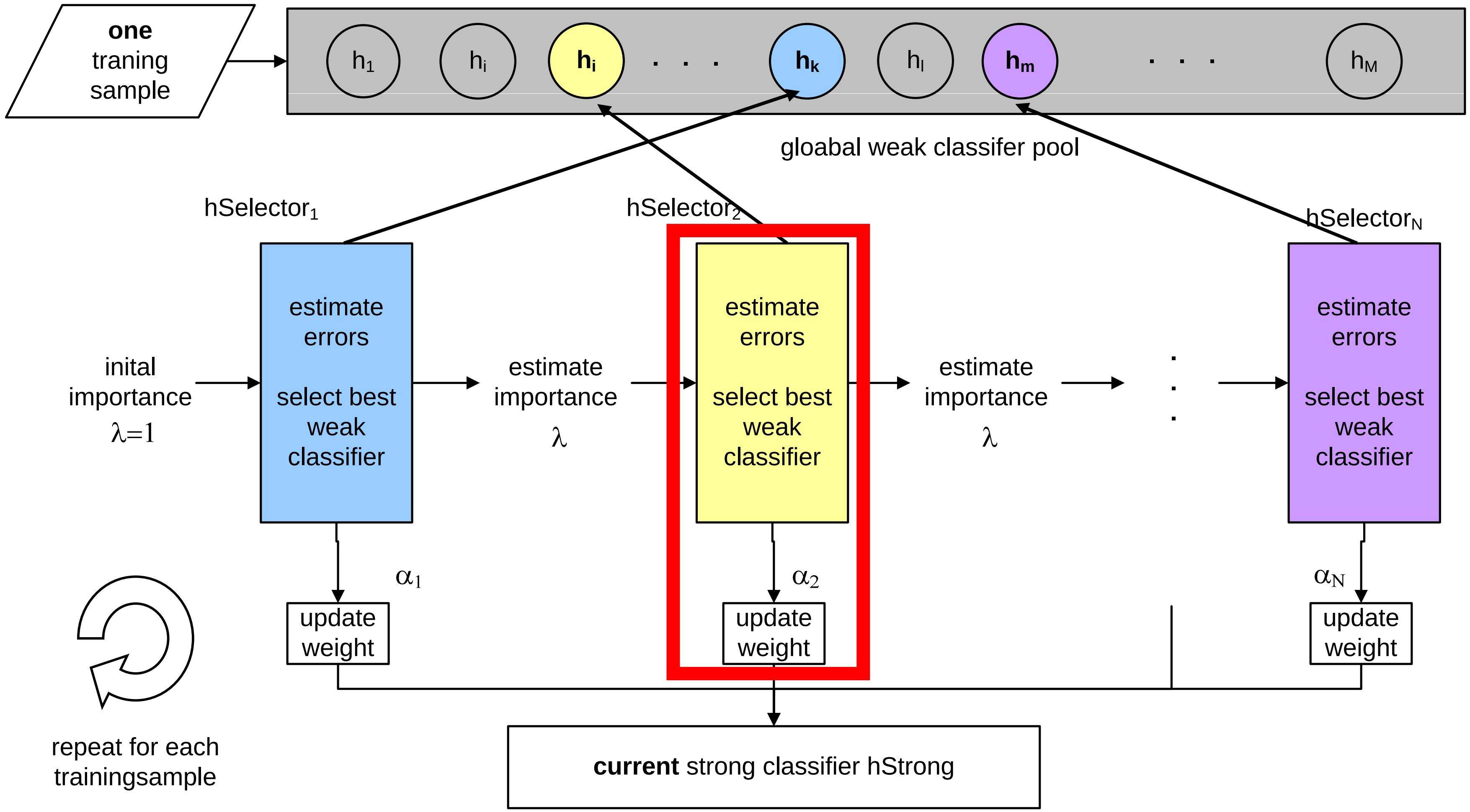
Direct Feature Selection



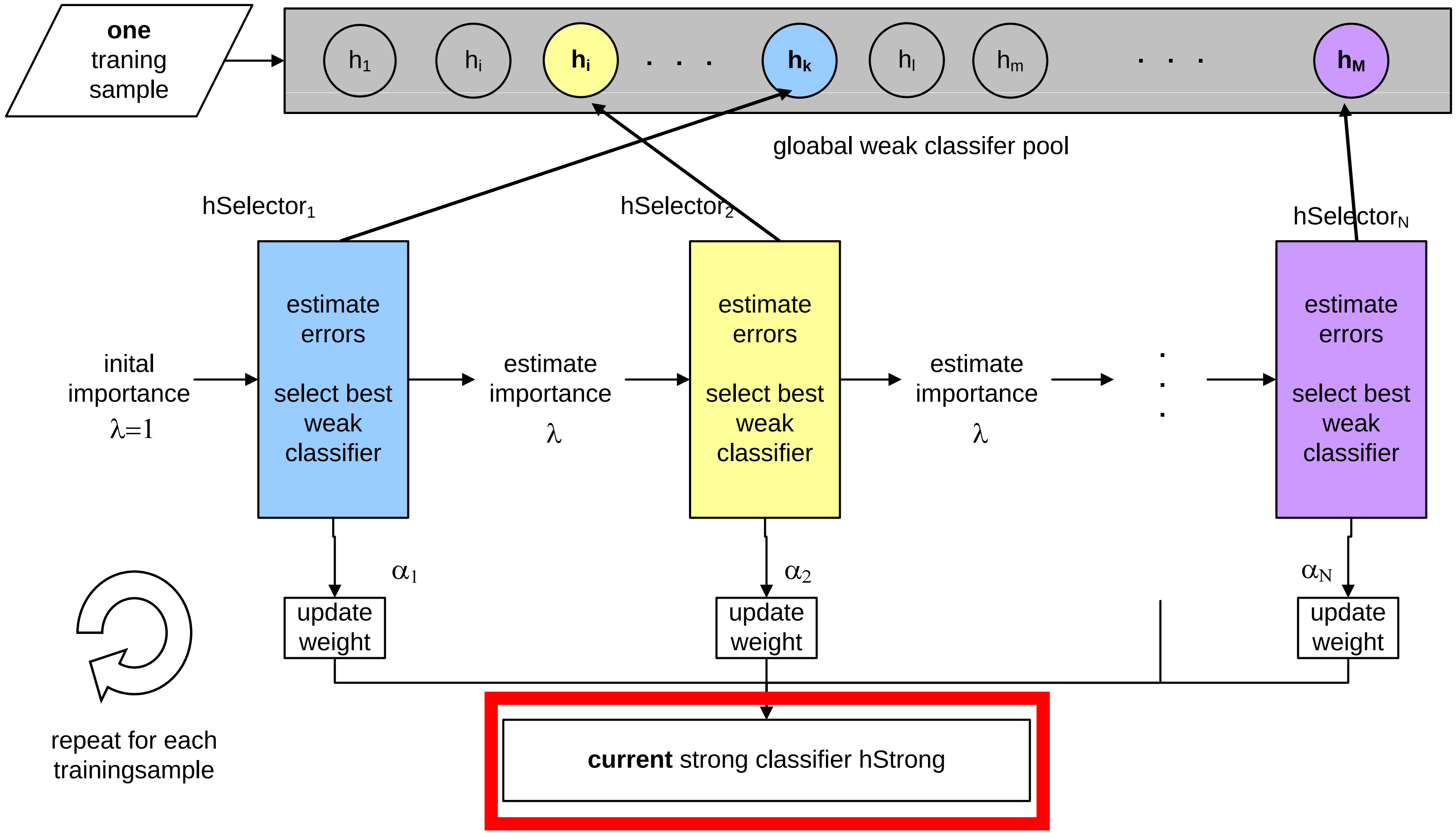
Direct Feature Selection



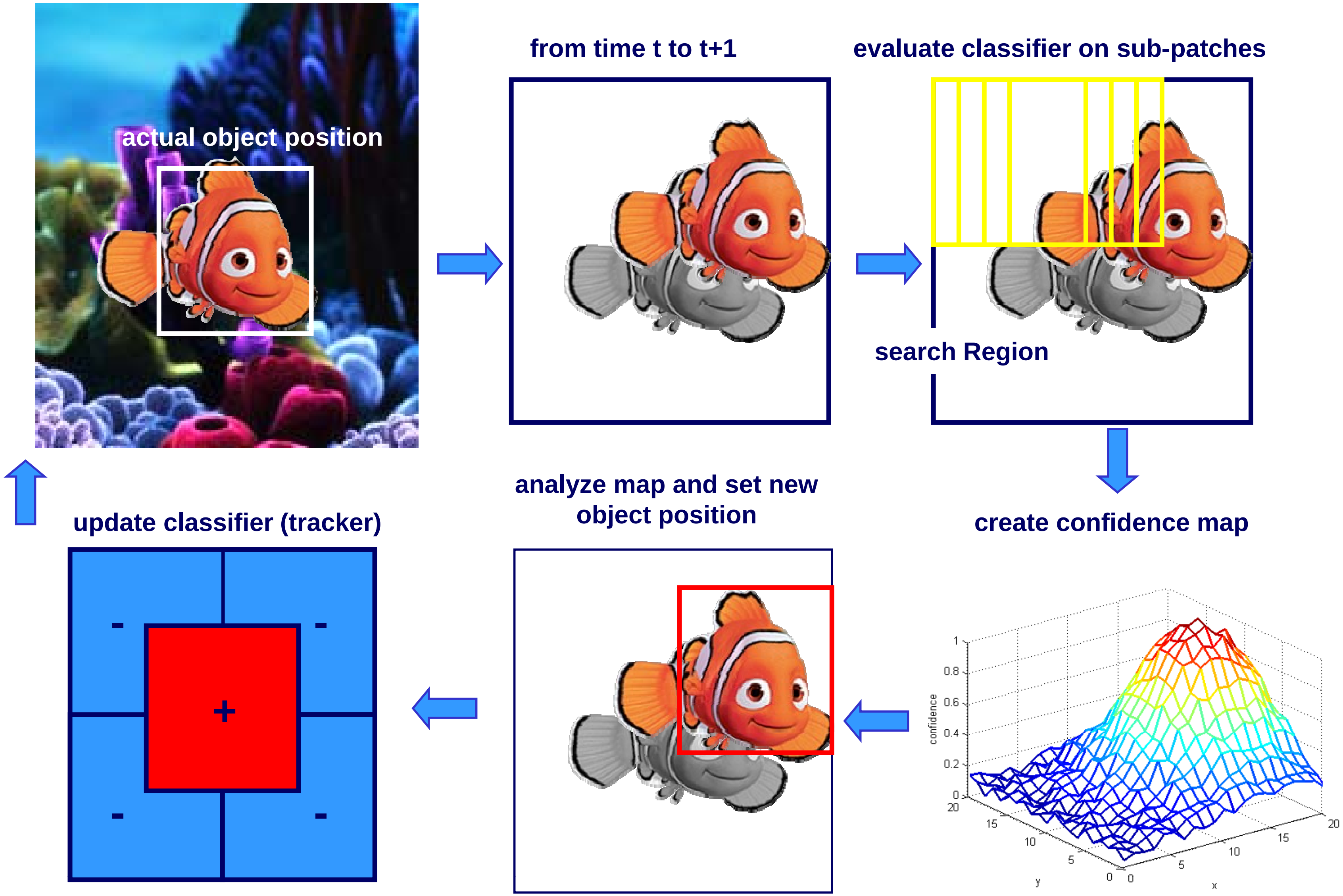
Direct Feature Selection



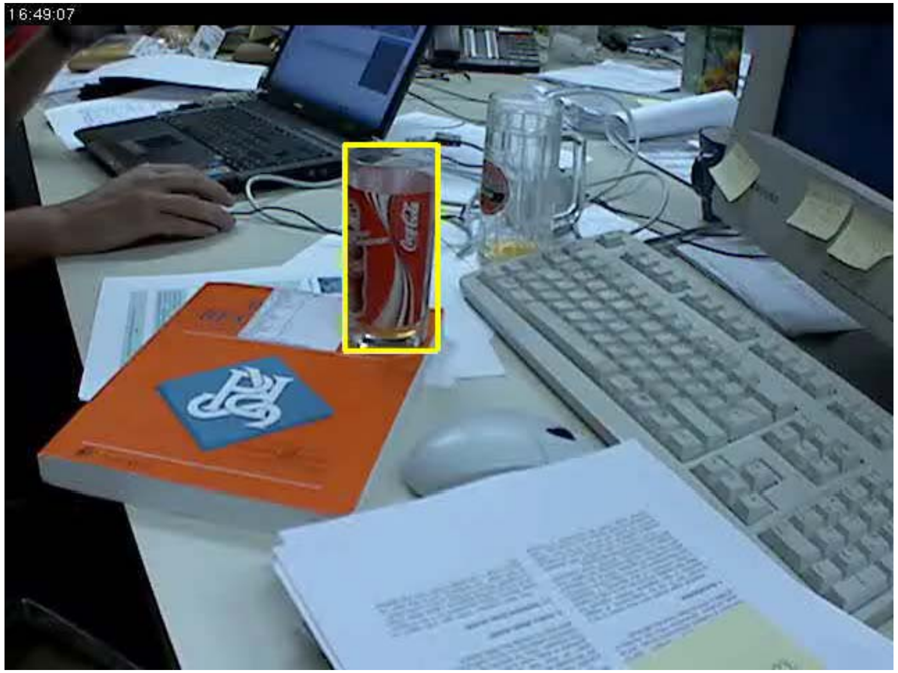
Direct Feature Selection



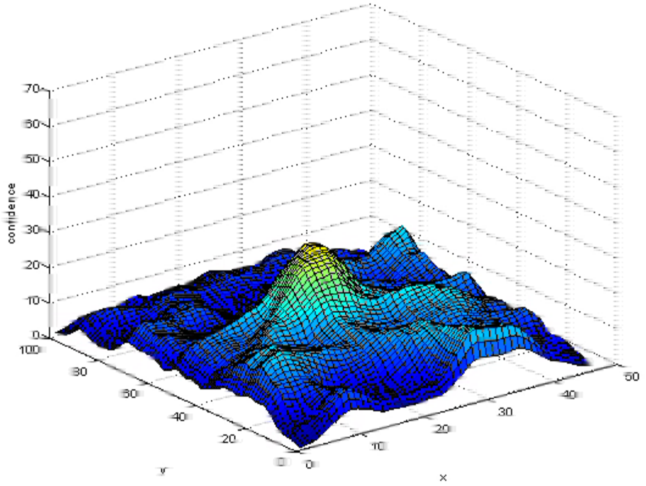
Tracking 1/2


actual object position
from time t to t+1
evaluate classifier on sub-patches
search Region
create confidence map
confidence
y
x
analyze map and set new object position
update classifier (tracker)
-
-
+
-
-

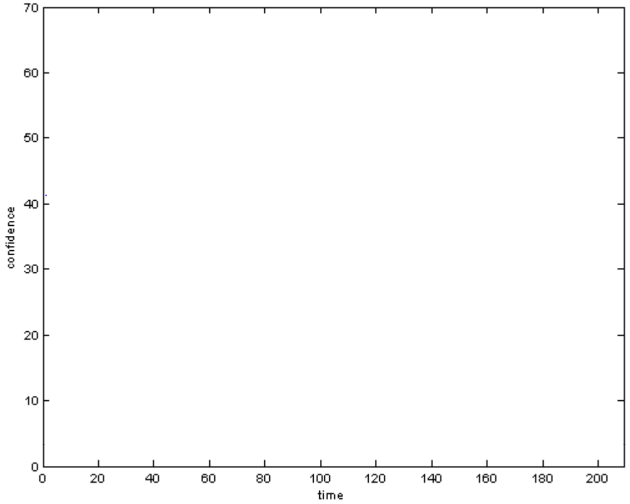
Tracking



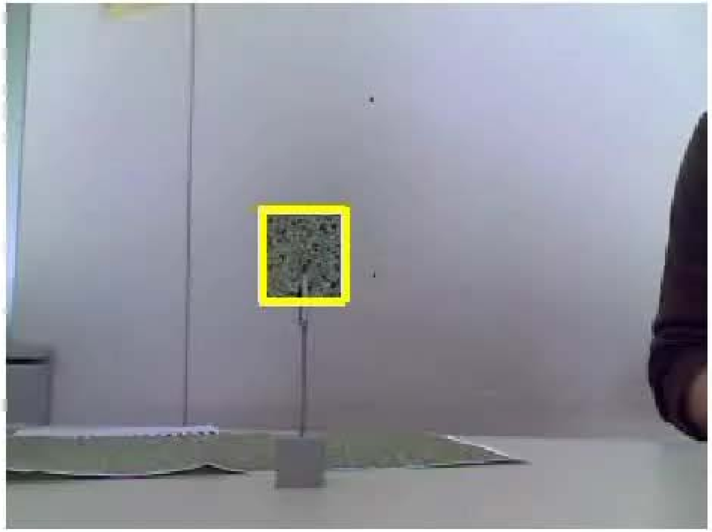
Confidence Map

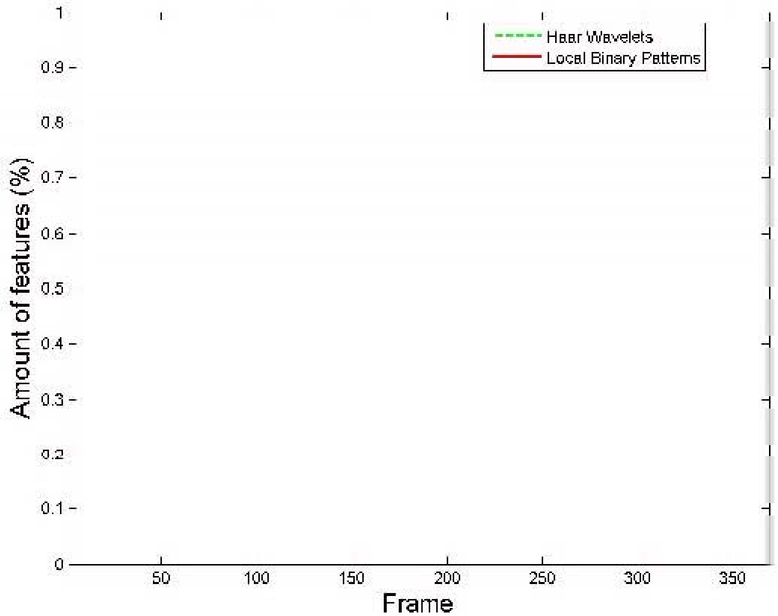


Max. Confidence Value



On-line Feature Exchange



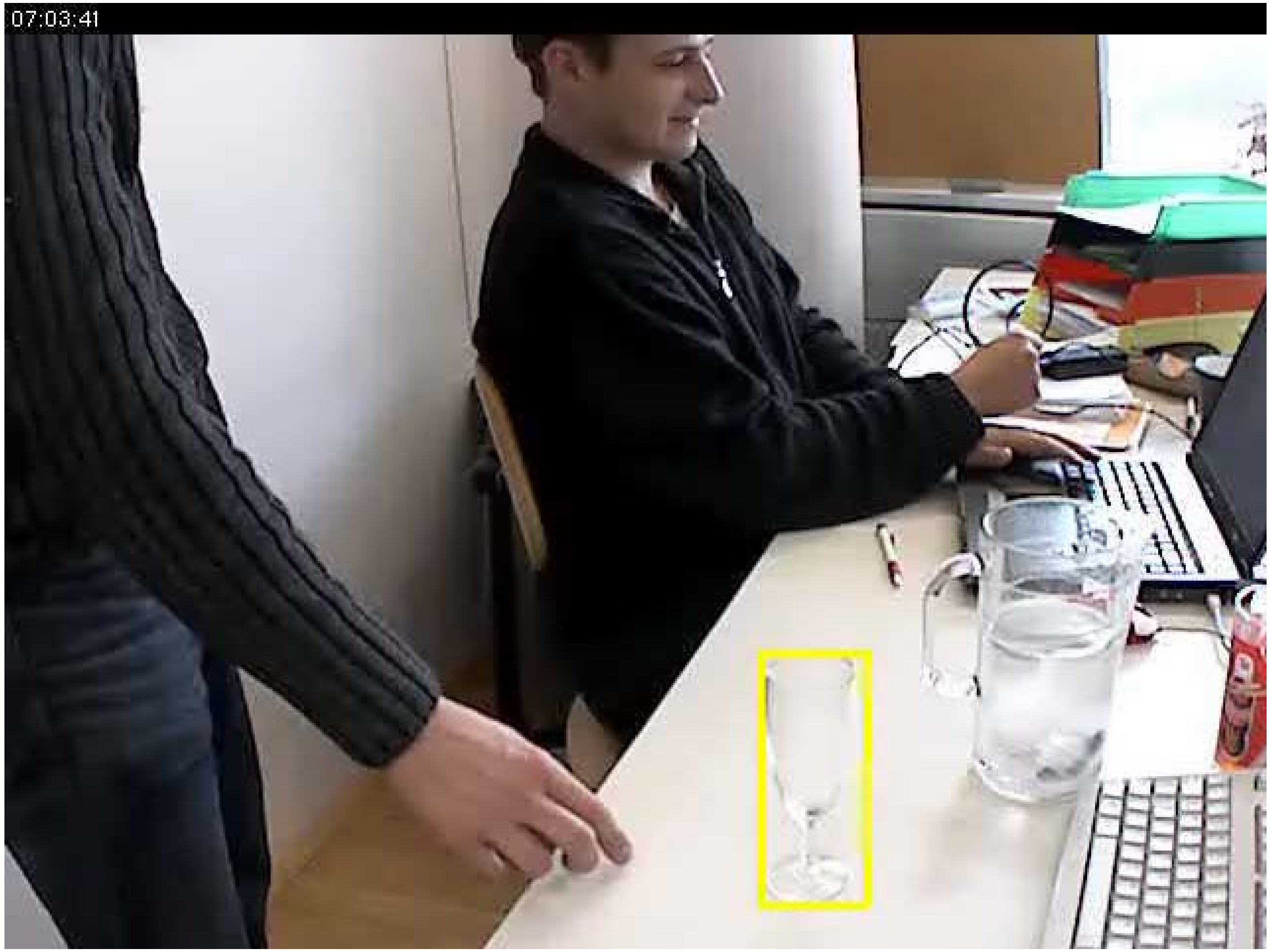


Public Sequences

J. Lim, D. Ross, R. Lin, and M. Yang.
Incremental learning for visual tracking. NIPS 2005.

A. D. Jepson, D. J. Fleet, and T.F. El-Maraghi.
Robust online appearance models for visual tracking.
CVPR 2001.

“Tracking the Invisible”



Conclusion

- **Tracking as Classification**
  - Continuously updating a classifier which discriminates the object from the background
  - Adaptivity
  - Robustness
  - Generality

- **Real-Time**
  - Efficient data structures for all basic image features types
  - Shared Feature Pool

Thank you for your attention.

Questions?

Combination: Detection, Tracking and Recognition