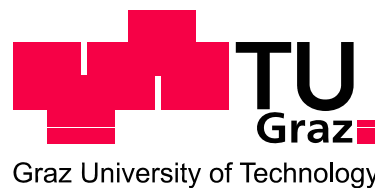


Real-Time Tracking via On-line Boosting

Helmut Grabner, Michael Grabner, Horst Bischof



Graz University of Technology
Institute for Computer Graphics and Vision





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

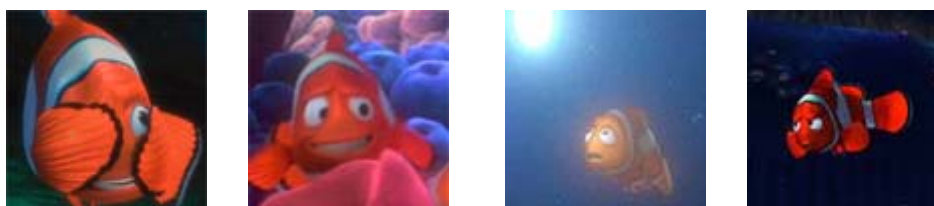
◆ Adaptivity

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



◆ Robustness

- Occlusions, cluttered background, illumination conditions



◆ Generality

- Any object



- ◆ **Tracking as Classification**

- ◆ **Boosting for Feature selection**
 - From Off-line to On-line
 - On-line Feature Selection

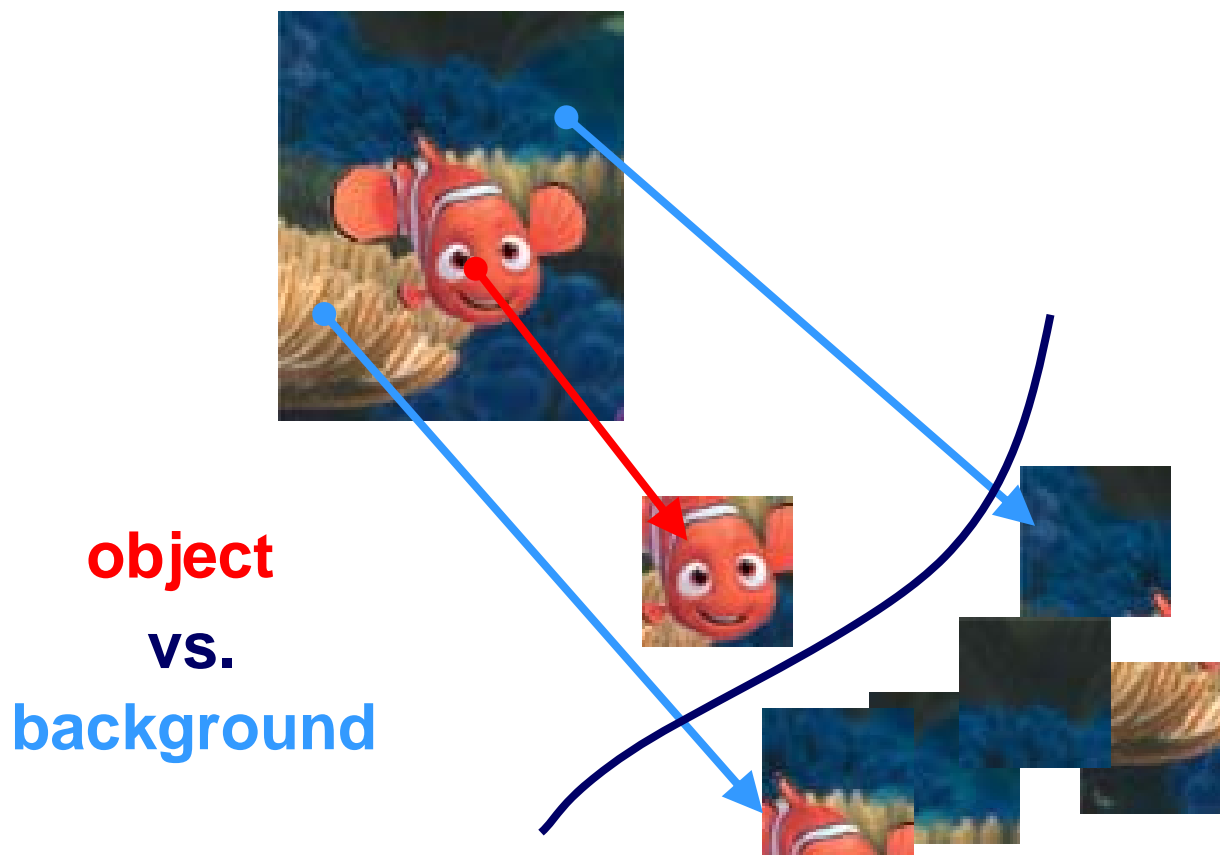
- ◆ **Tracking**

- ◆ **Experimental Results**

- ◆ **Conclusion**

◆ Tracking as binary classification

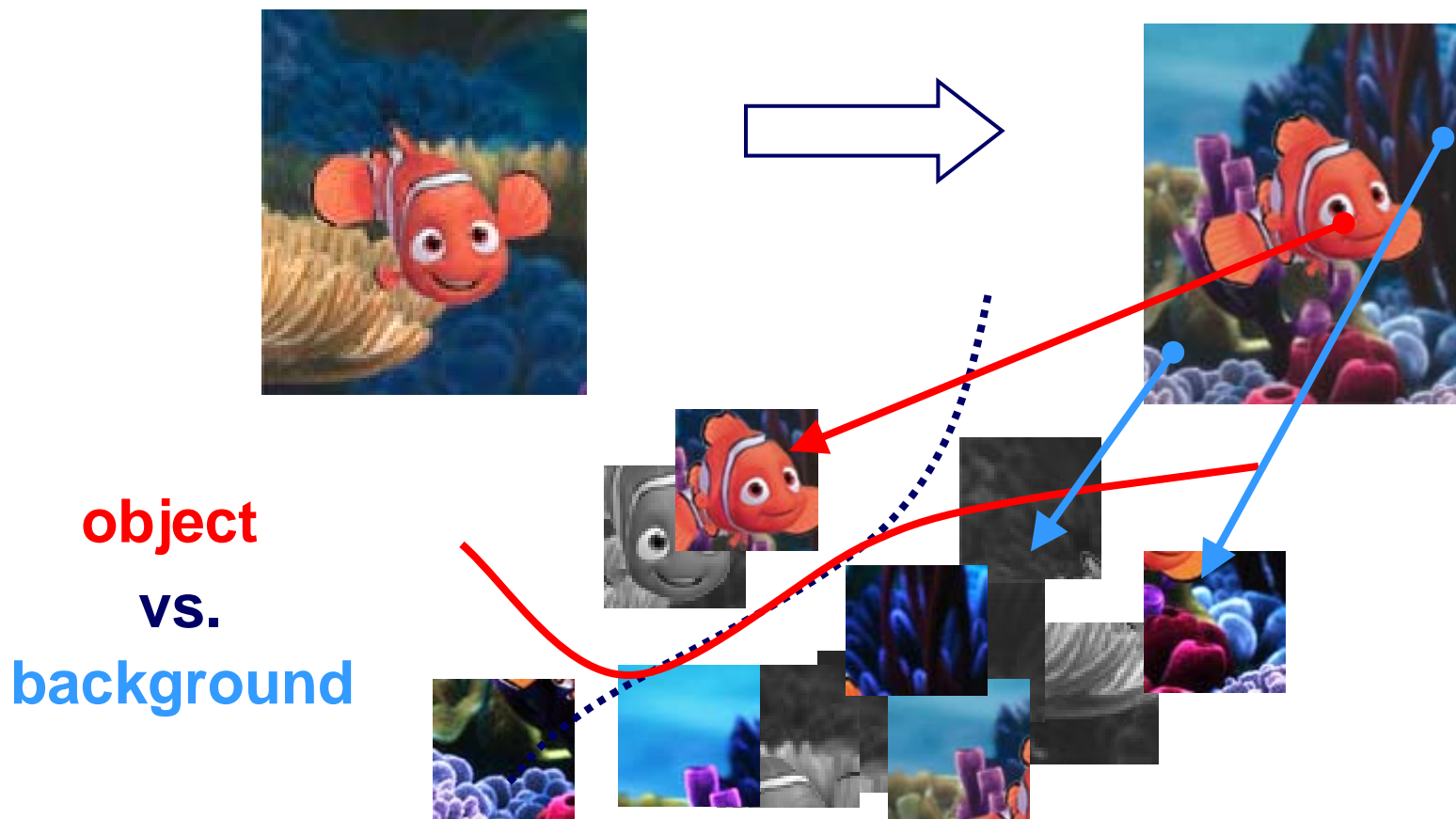
S. Avidan. **Ensemble tracking**. CVPR 2005.
 J.Wang, et al. **Online selecting discriminative tracking features using particle filter**. CVPR 2005.



◆ **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.

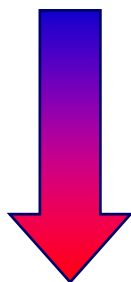
◆ **Object and background changes are robustly handled by **on-line** updating!**



Object Detector

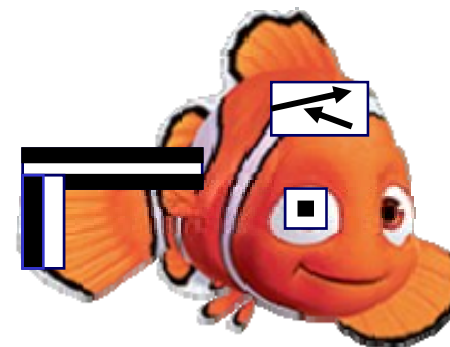
P. Viola and M. Jones. **Rapid object detection using a boosted cascade of simple features.** CVPR 2001.

Fixed Training set
General object
detector



Object Tracker

On-line update
Object vs. Background

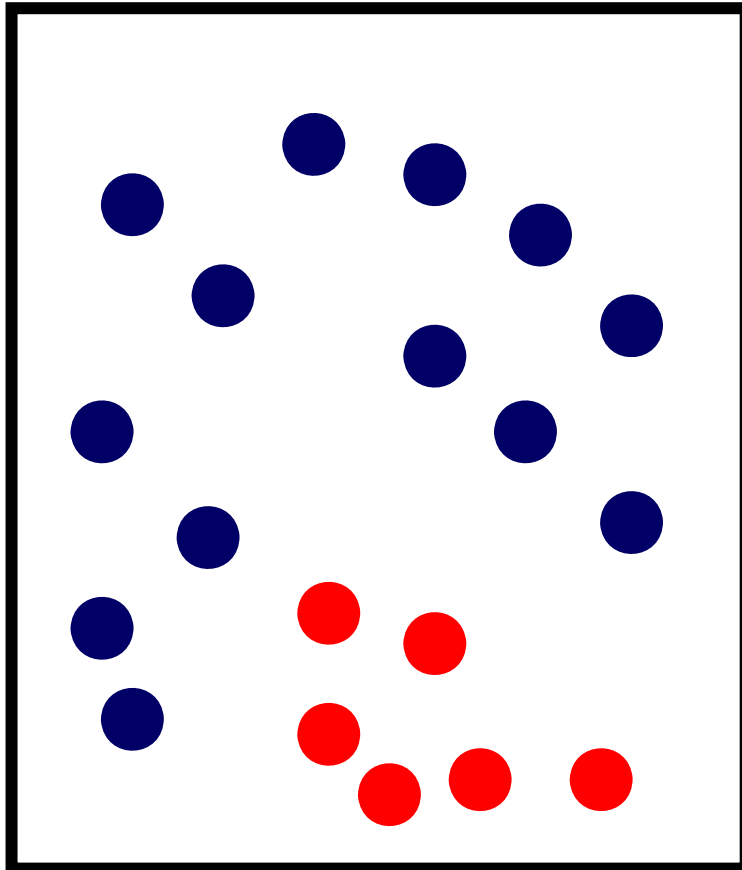


$$\text{sign}(\alpha_1 \cdot \text{[feature 1]} + \alpha_2 \cdot \text{[feature 2]} + \alpha_3 \cdot \text{[feature 3]} + \dots)$$

Combination of simple image features
using Boosting as Feature Selection

On-Line Boosting for Feature Selection

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



Given:

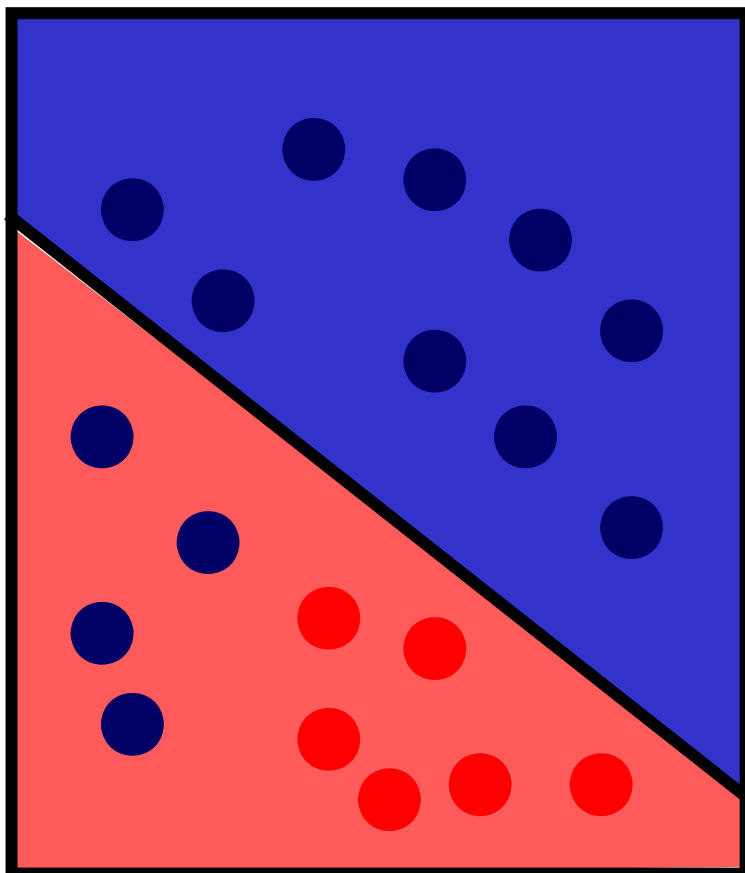
- set of labeled training samples
- weight distribution over them

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.

Given:

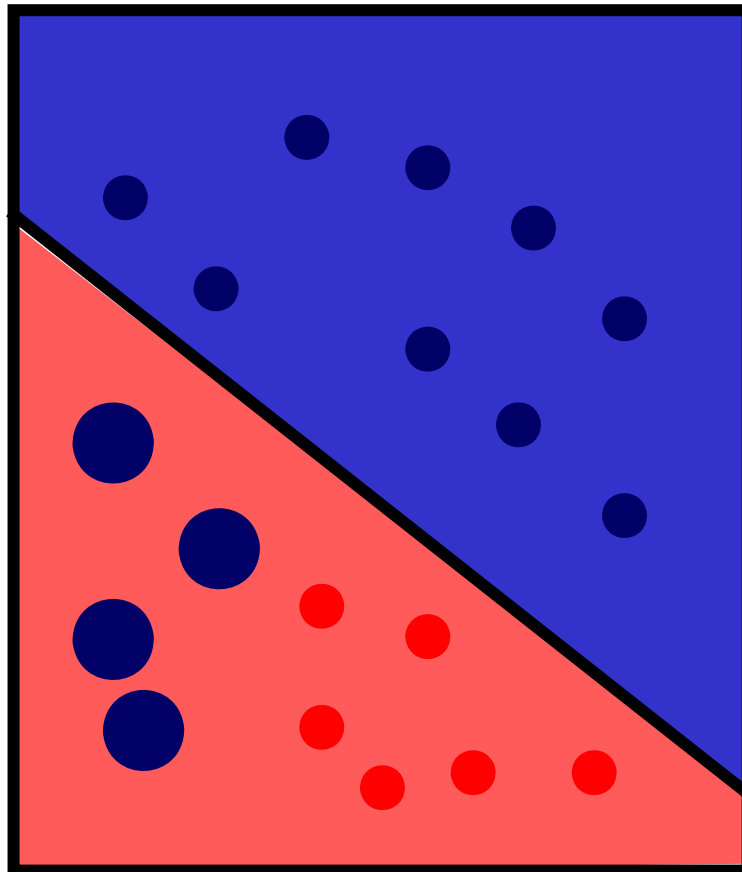
- set of labeled training samples
- weight distribution over them

Algorithm:

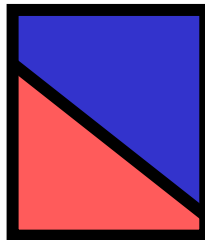
for n = 1 to N

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

next



$\alpha_1 \cdot$



Given:

- set of labeled training samples
- weight distribution over them

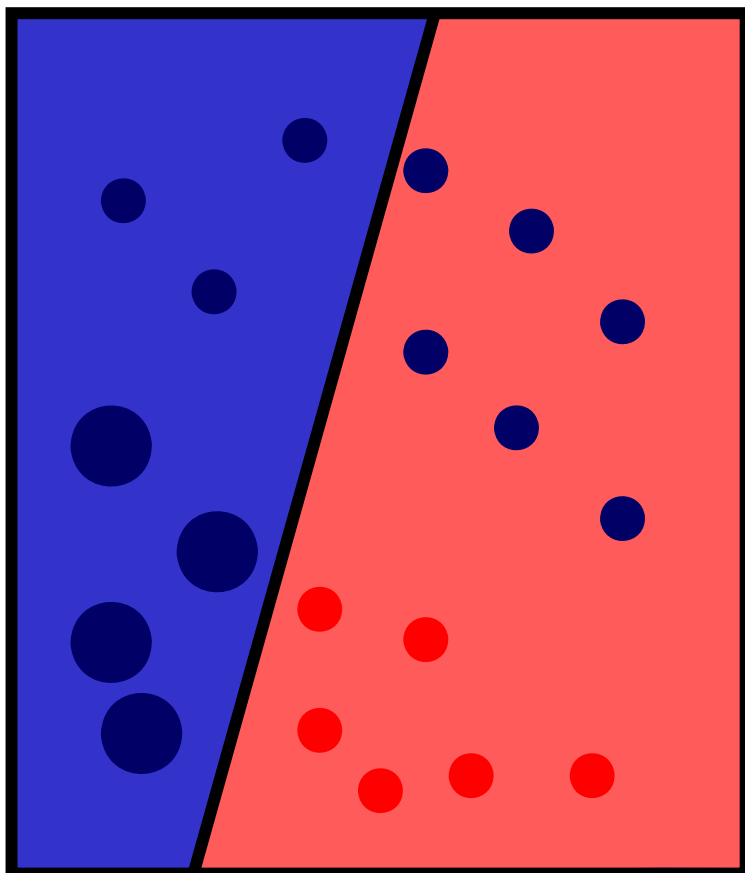
Algorithm:

for n = 1 to N

- train a weak classifier using samples and weight dist.

- calculate error
- calculate weight
- update weight dist.

next



Given:

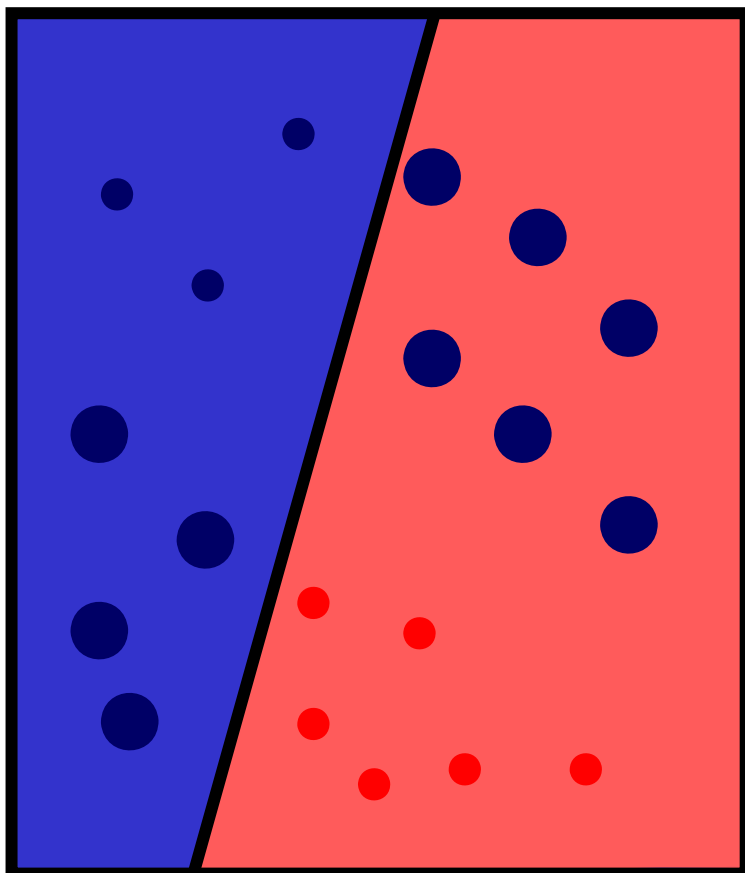
- set of labeled training samples
- weight distribution over them

Algorithm:

for n = 1 to N

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

next



Given:

- set of labeled training samples
- weight distribution over them

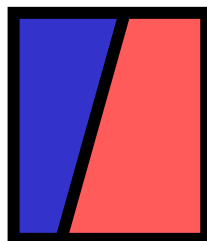
Algorithm:

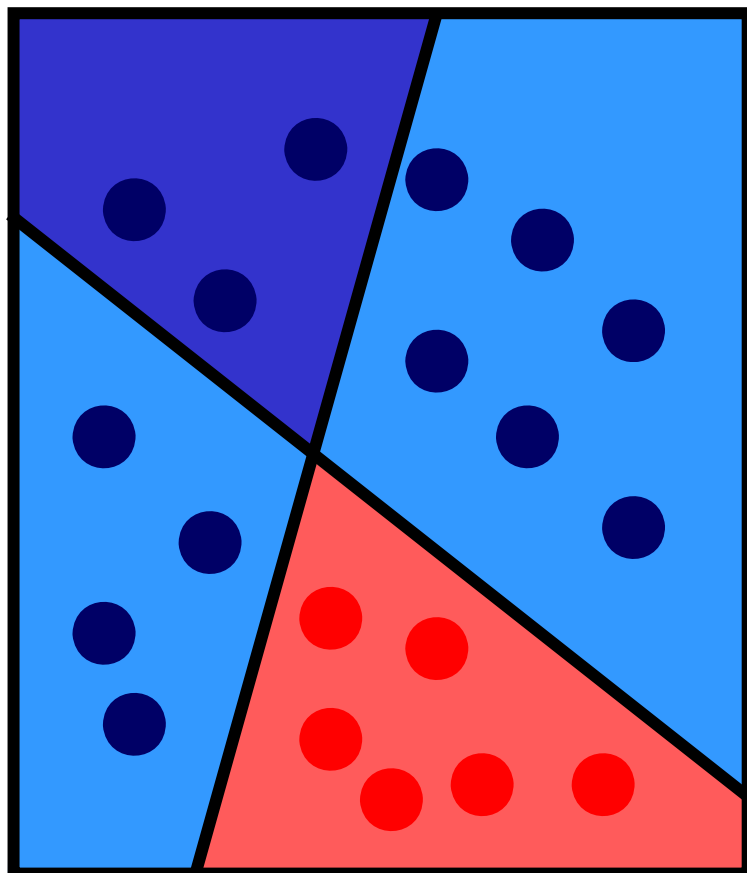
for n = 1 to N

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

next

$\alpha_2 \cdot$





$$= \alpha_1 \cdot \begin{array}{|c|} \hline \text{dark blue} \\ \hline \text{light blue} \\ \hline \end{array} + \alpha_2 \cdot \begin{array}{|c|} \hline \text{dark blue} \\ \hline \text{red} \\ \hline \end{array}$$

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:

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

off-line

Given:

- set of labeled training samples

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

- 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

off-line

Given:

- set of labeled training samples
 $\mathcal{X} = \{\langle \mathbf{x}_1, y_1 \rangle, \dots, \langle \mathbf{x}_L, y_L \rangle \mid y_i \pm 1\}$
- 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:

for $n = 1$ to N

next

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

off-line

only **one** training example
to **update** the classifier

on-line

Given:

- set of labeled training samples
 $\mathcal{X} = \{\langle \mathbf{x}_1, y_1 \rangle, \dots, \langle \mathbf{x}_L, y_L \rangle \mid y_i \pm 1\}$
- 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

for $n = 1$ to N

next

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

off-line

update importance for the current sample

on-line

Given:

- set of labeled training samples
 $\mathcal{X} = \{\langle \mathbf{x}_1, y_1 \rangle, \dots, \langle \mathbf{x}_L, y_L \rangle \mid y_i \pm 1\}$
- 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

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

off-line

online update the weak classifier

on-line

Given:

- set of labeled training samples
 $\mathcal{X} = \{\langle \mathbf{x}_1, y_1 \rangle, \dots, \langle \mathbf{x}_L, y_L \rangle \mid y_i \pm 1\}$
- 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 the weak classifier using samples and importance

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

- update importance weight λ

next

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

off-line

update errors and weights

on-line

Given:

- set of labeled training samples
 $\mathcal{X} = \{\langle \mathbf{x}_1, y_1 \rangle, \dots, \langle \mathbf{x}_L, y_L \rangle \mid y_i \pm 1\}$
- 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 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 \hat{e}_n
- update weight $\alpha_n = f(\hat{e}_n)$
- update importance weight λ

next

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

off-line

Given:

- set of labeled training samples
 $\mathcal{X} = \{\langle \mathbf{x}_1, y_1 \rangle, \dots, \langle \mathbf{x}_L, y_L \rangle \mid y_i \pm 1\}$
- 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

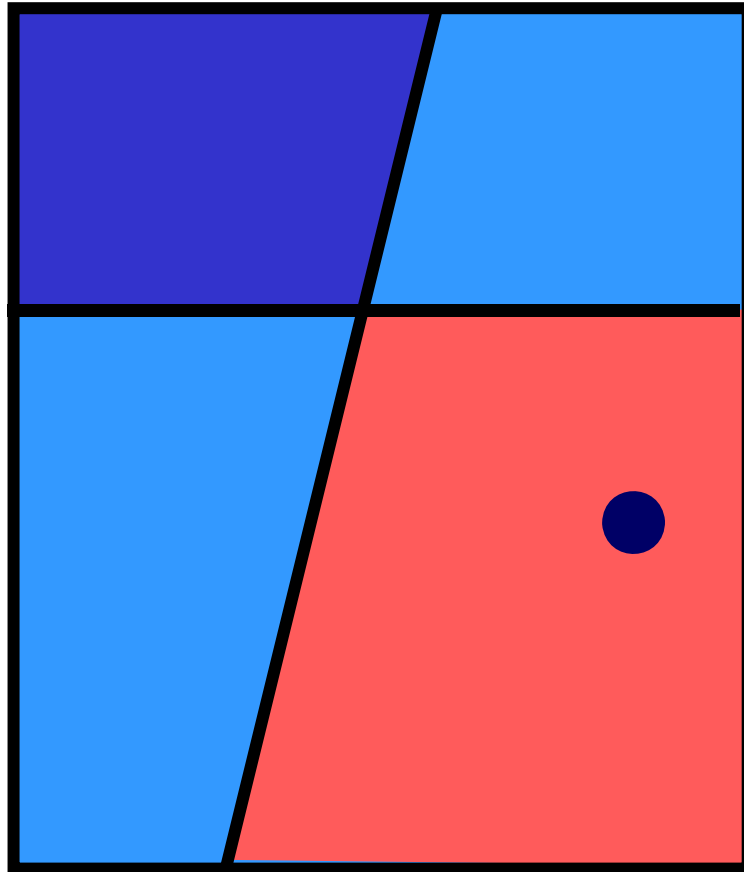
- update the weak classifier using samples and importance

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

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

next

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



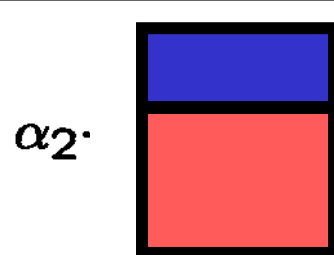
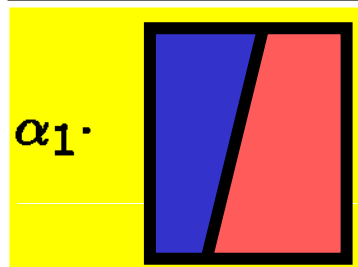
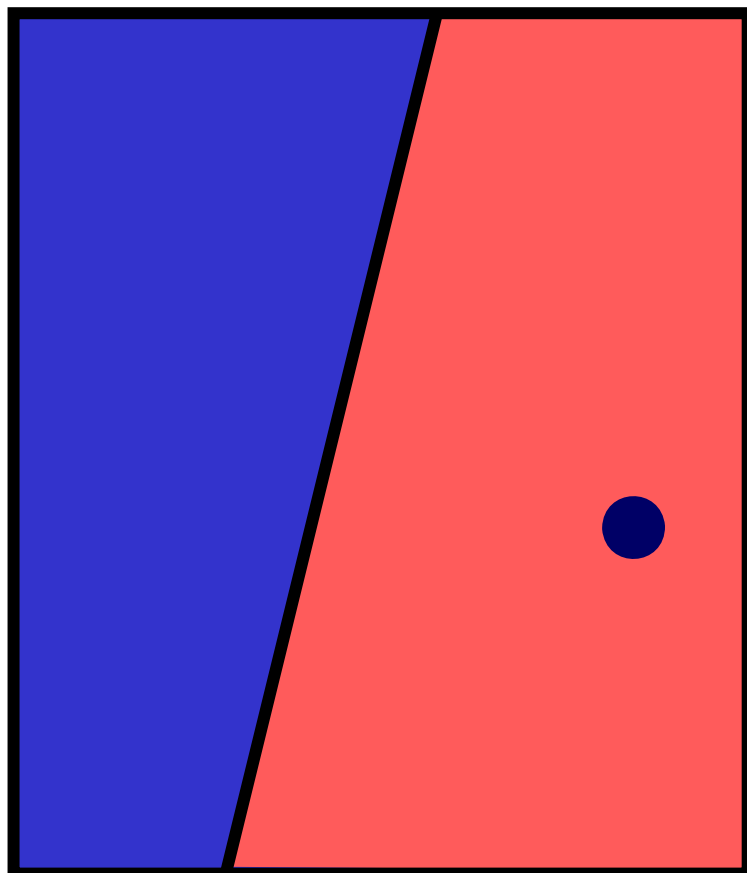
$$= \alpha_1 \cdot \left[\begin{array}{|c|} \hline \text{dark blue} \\ \hline \text{light blue} \\ \hline \end{array} \right] + \alpha_2 \cdot \left[\begin{array}{|c|} \hline \text{dark blue} \\ \hline \text{red} \\ \hline \end{array} \right]$$

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

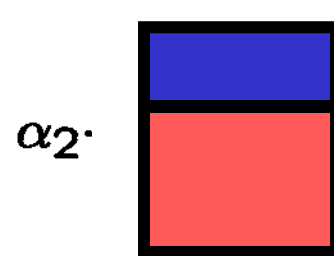
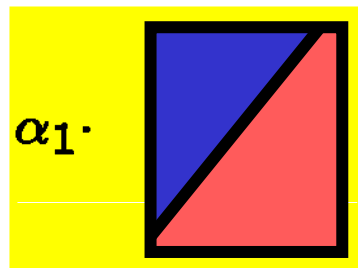
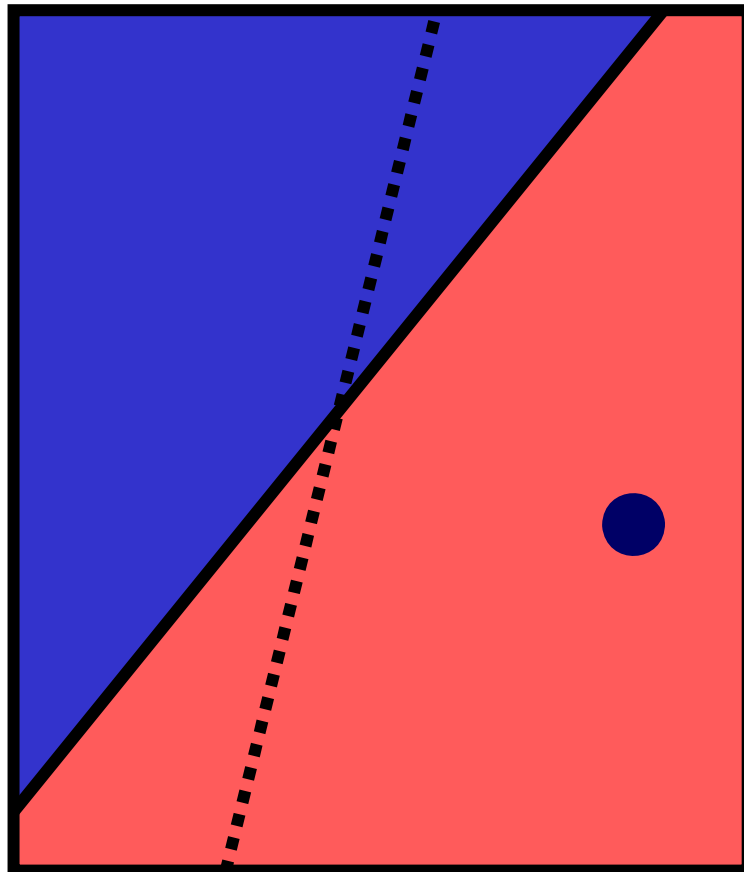
Algorithm:

- initial importance

for n = 1 to N

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

next



Given:

- ONE labeled training sample
- strong classifier to update

Algorithm:

- initial importance

for n = 1 to N

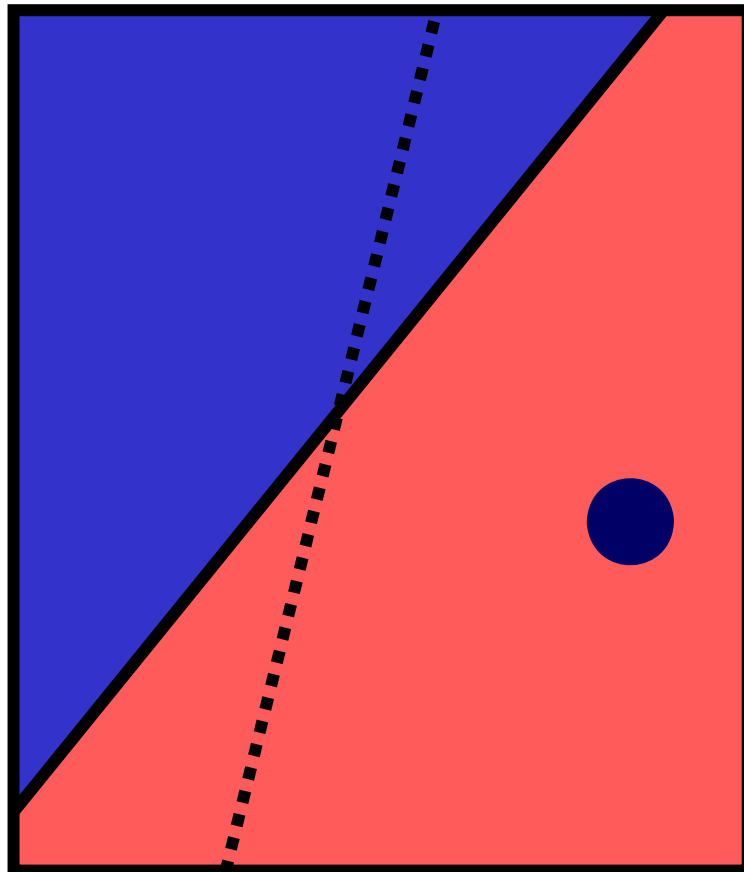
- update the weak classifier using sample and importance

- update error estimation

- update weight

- update importance weight

next



Given:

- ONE labeled training sample
- strong classifier to update

Algorithm:

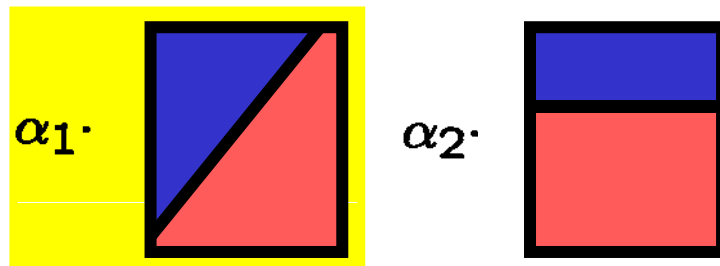
- initial importance

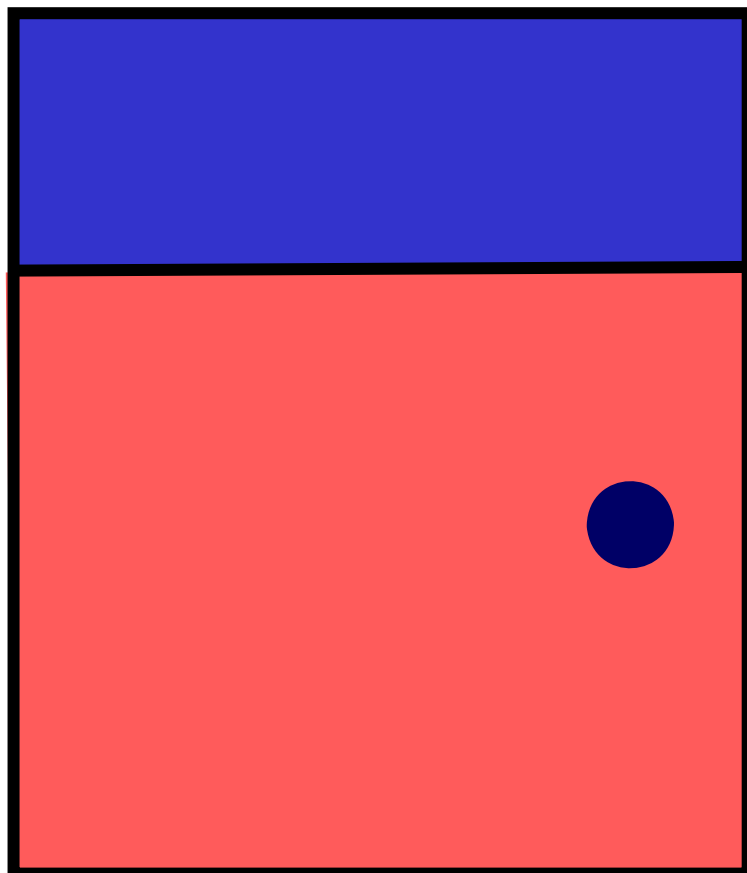
for n = 1 to N

- update the weak classifier using sample and importance

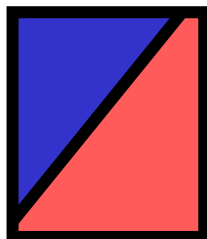
- update error estimation
- update weight
- update importance weight

next

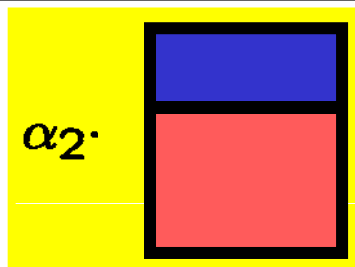




$\alpha_1 \cdot$



$\alpha_2 \cdot$



Given:

- ONE labeled training sample
- strong classifier to update

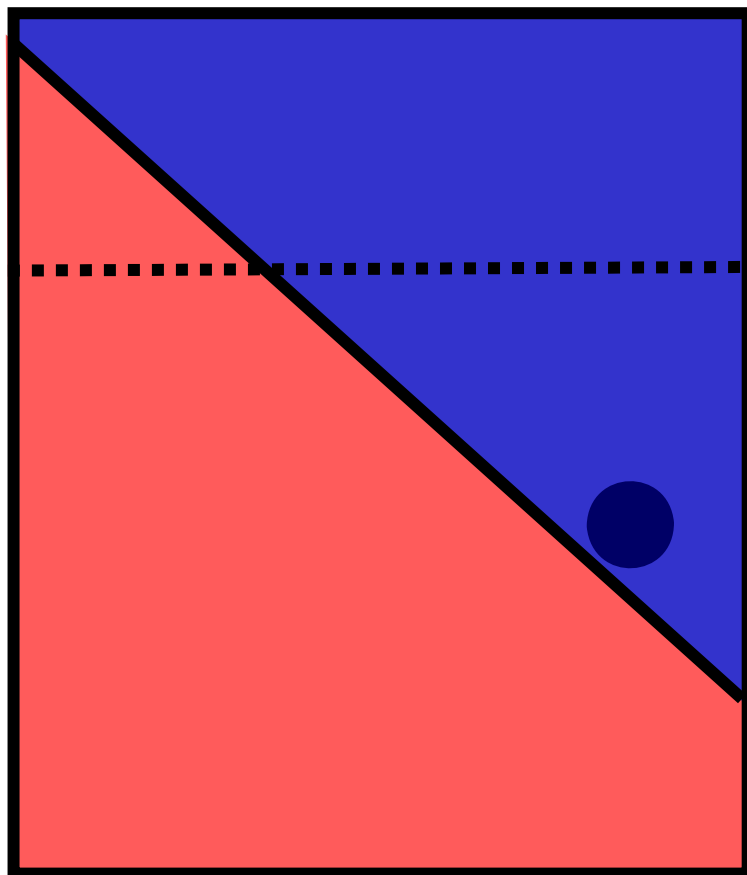
Algorithm:

- initial importance

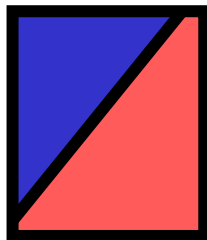
for n = 1 to N

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

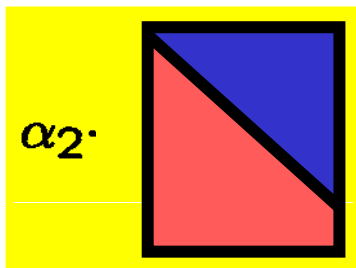
next



$\alpha_1 \cdot$



$\alpha_2 \cdot$



Given:

- ONE labeled training sample
- strong classifier to update

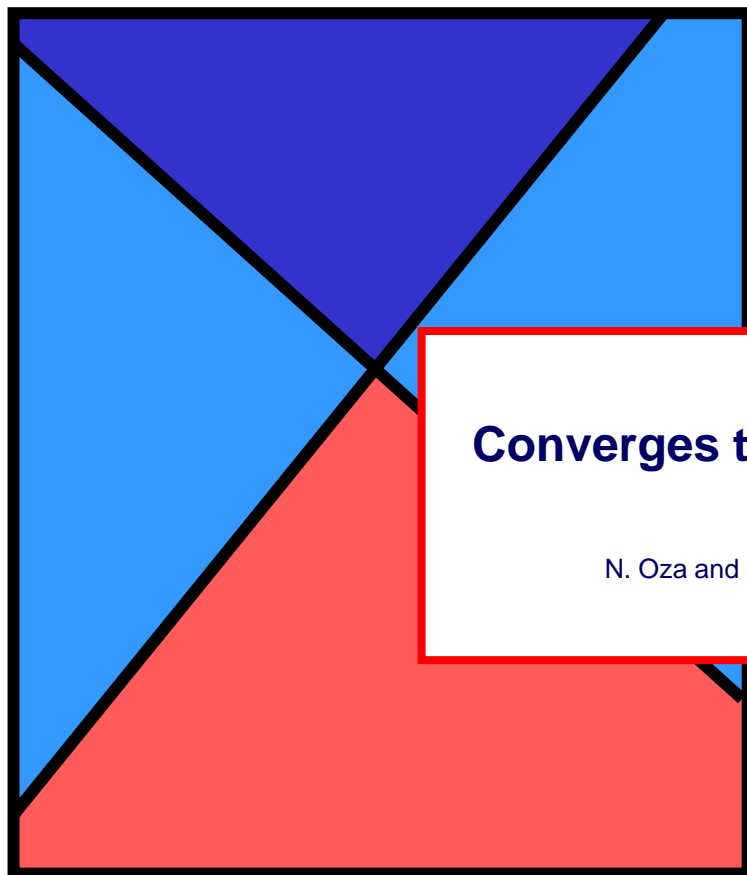
Algorithm:

- initial importance

for n = 1 to N

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

next



Converges to the off-line results...

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

$$= \alpha_1 \cdot \begin{array}{|c|} \hline \text{dark blue} \\ \hline \text{light blue} \\ \hline \end{array} + \alpha_2 \cdot \begin{array}{|c|} \hline \text{light blue} \\ \hline \text{red} \\ \hline \end{array}$$

Given:

- ONE labeled training sample
- strong classifier to update

Algorithm:

- initial importance

pick classifier using importance

estimation

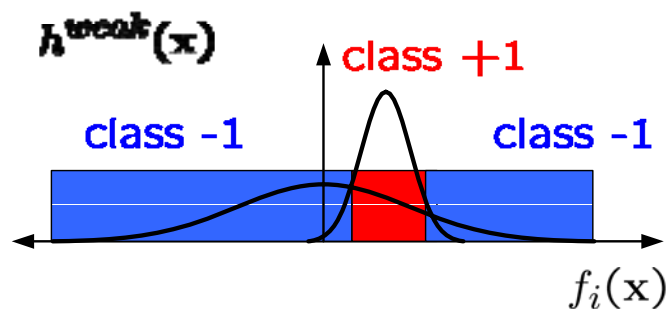
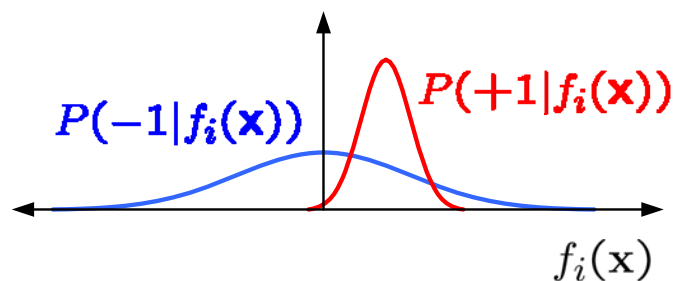
update importance weight

next

Result:

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

- ◆ 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.

◆ Introducing “Selector”

- selects **one** feature from its local feature pool

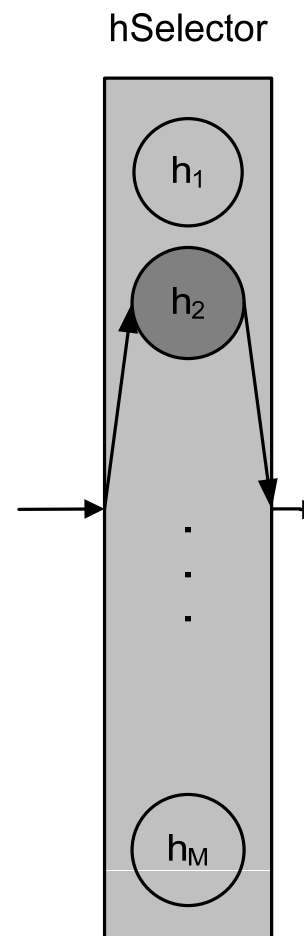
$$\mathcal{H}^{weak} = \{h_1^{weak}, \dots, h_M^{weak}\}$$

$$\mathcal{F} = \{f_1, \dots, f_M\}$$

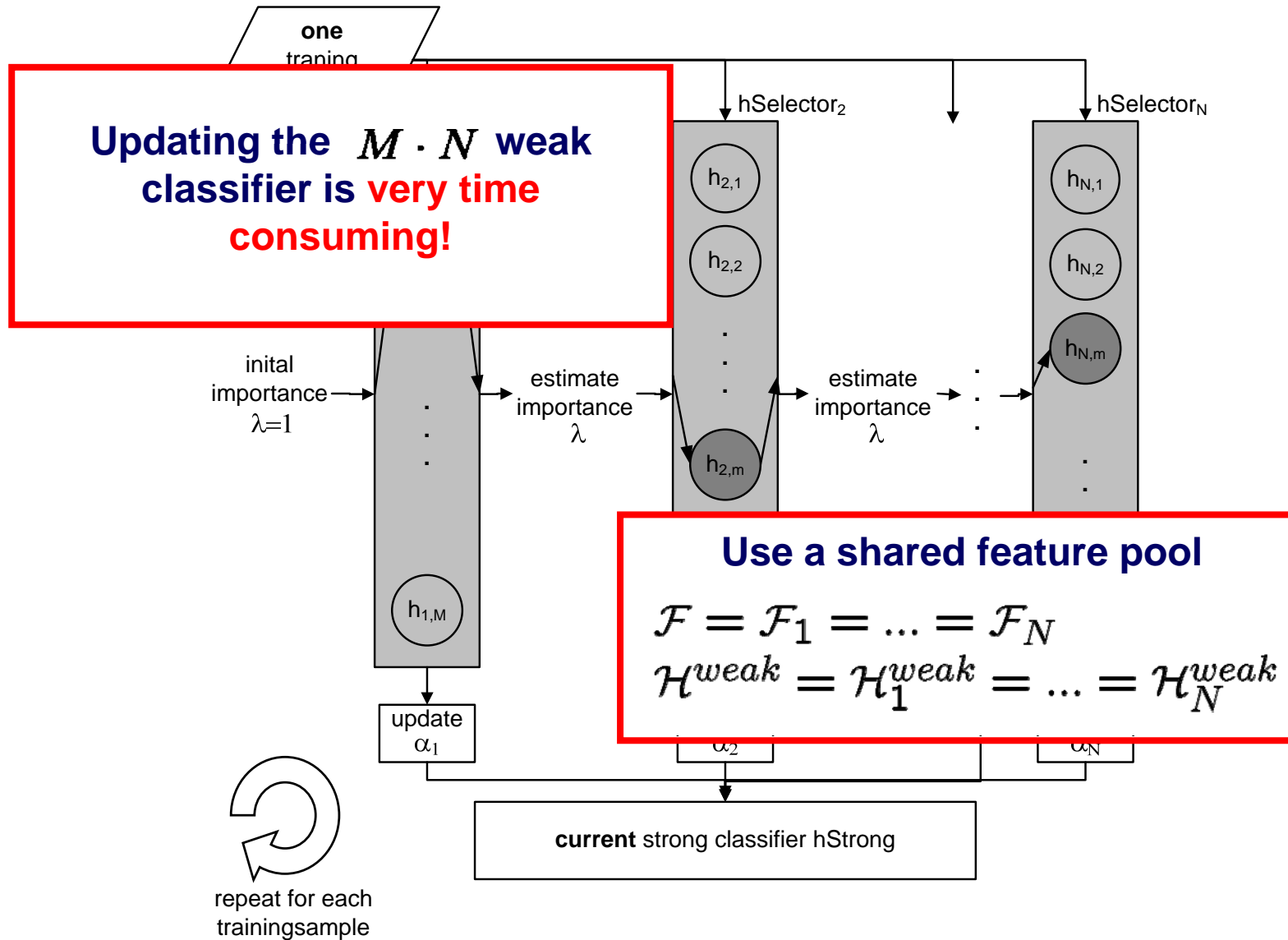
$$h^{sel}(\mathbf{x}) = h_{m^*}^{weak}(\mathbf{x})$$

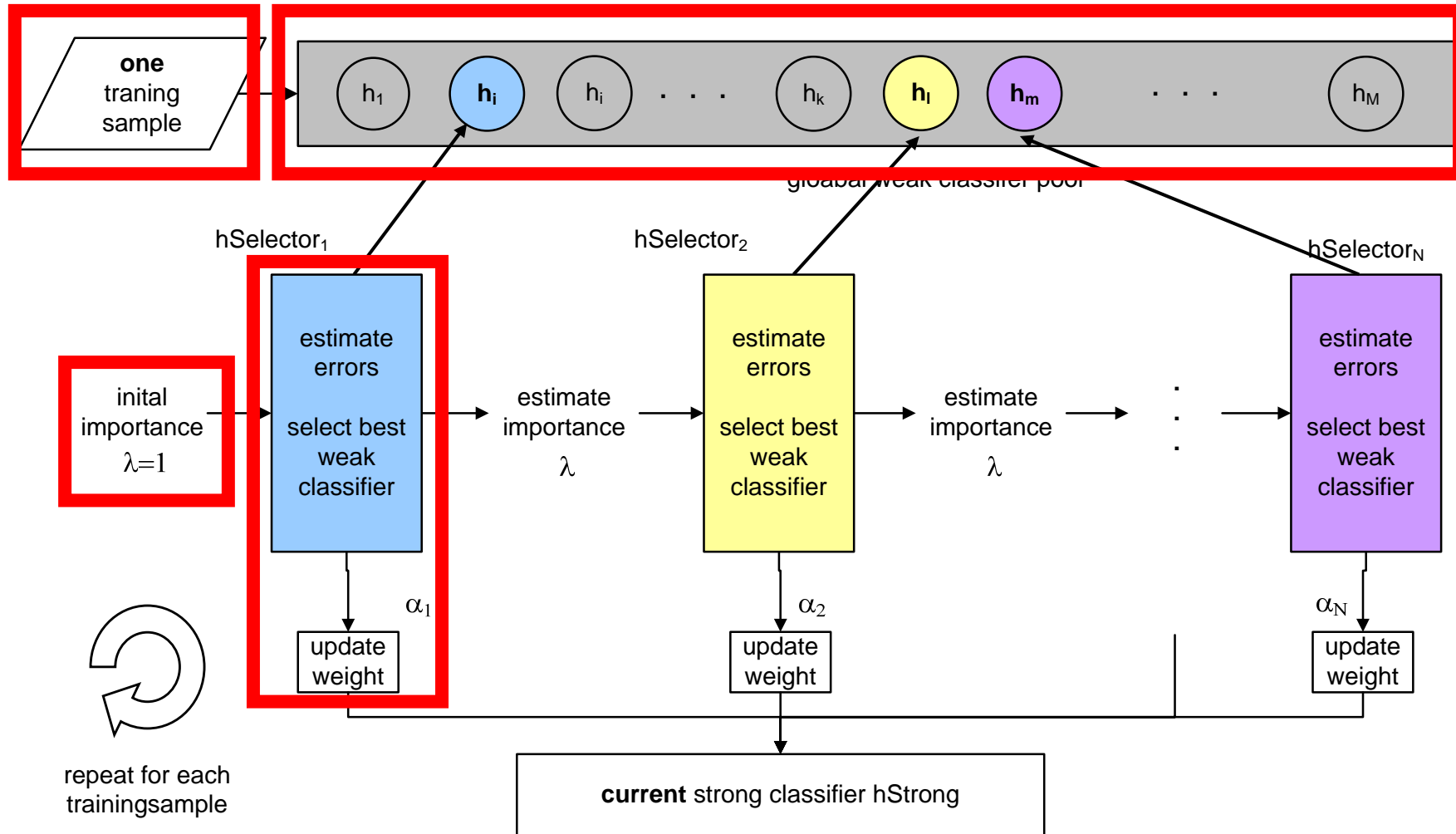
$$m^* = \arg \min_i e_i$$

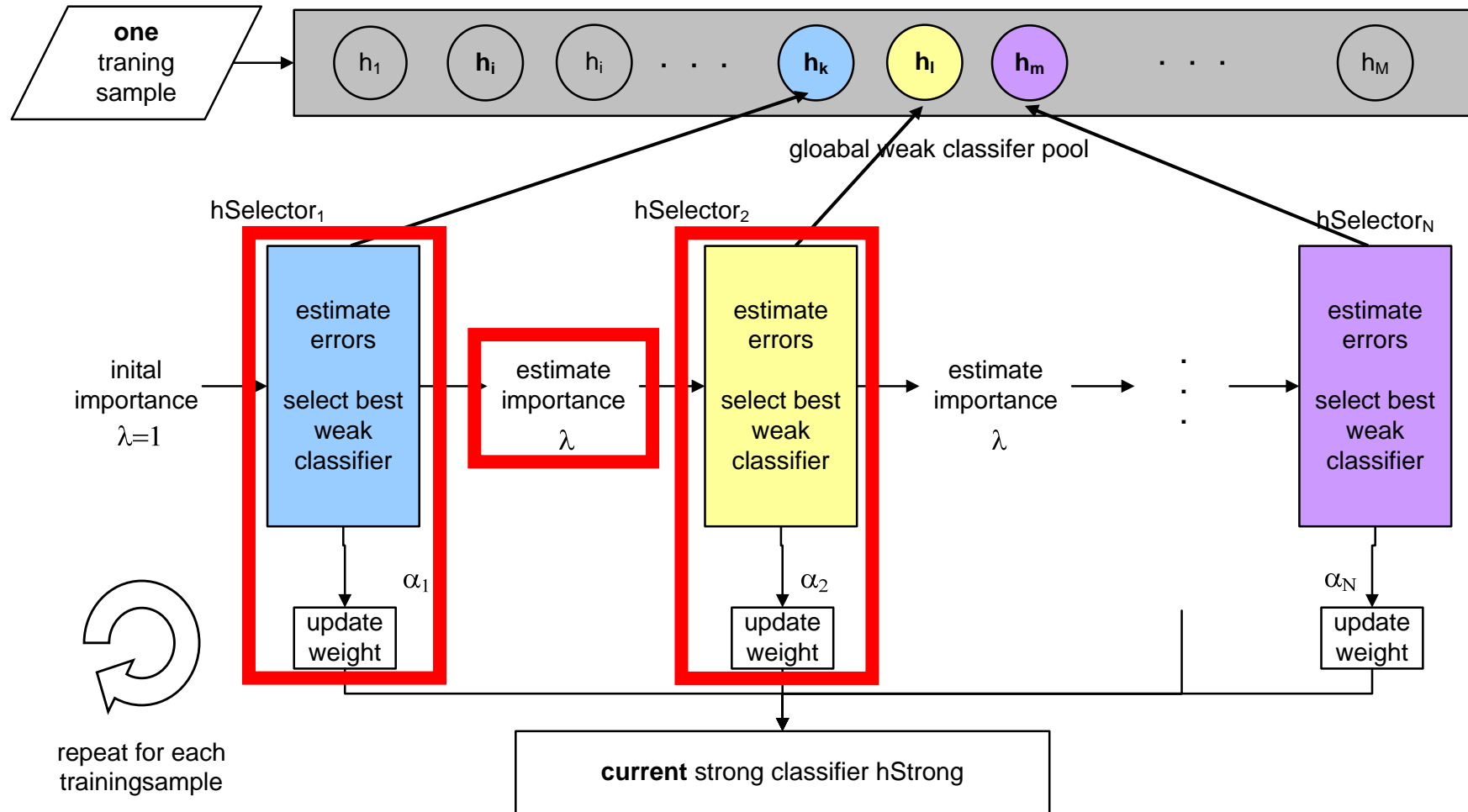
On-line boosting is performed on the **Selectors** and not on the weak classifiers directly.

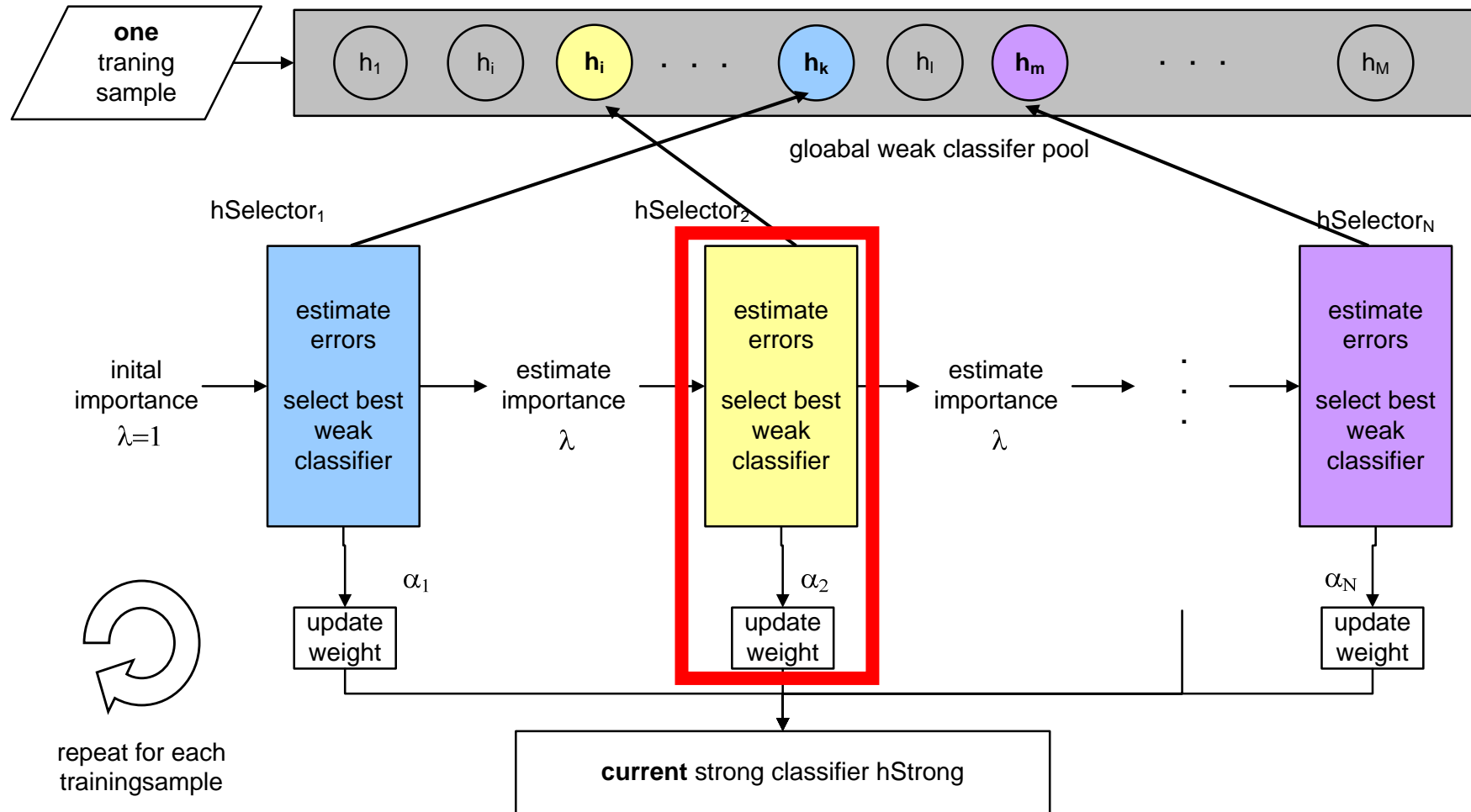


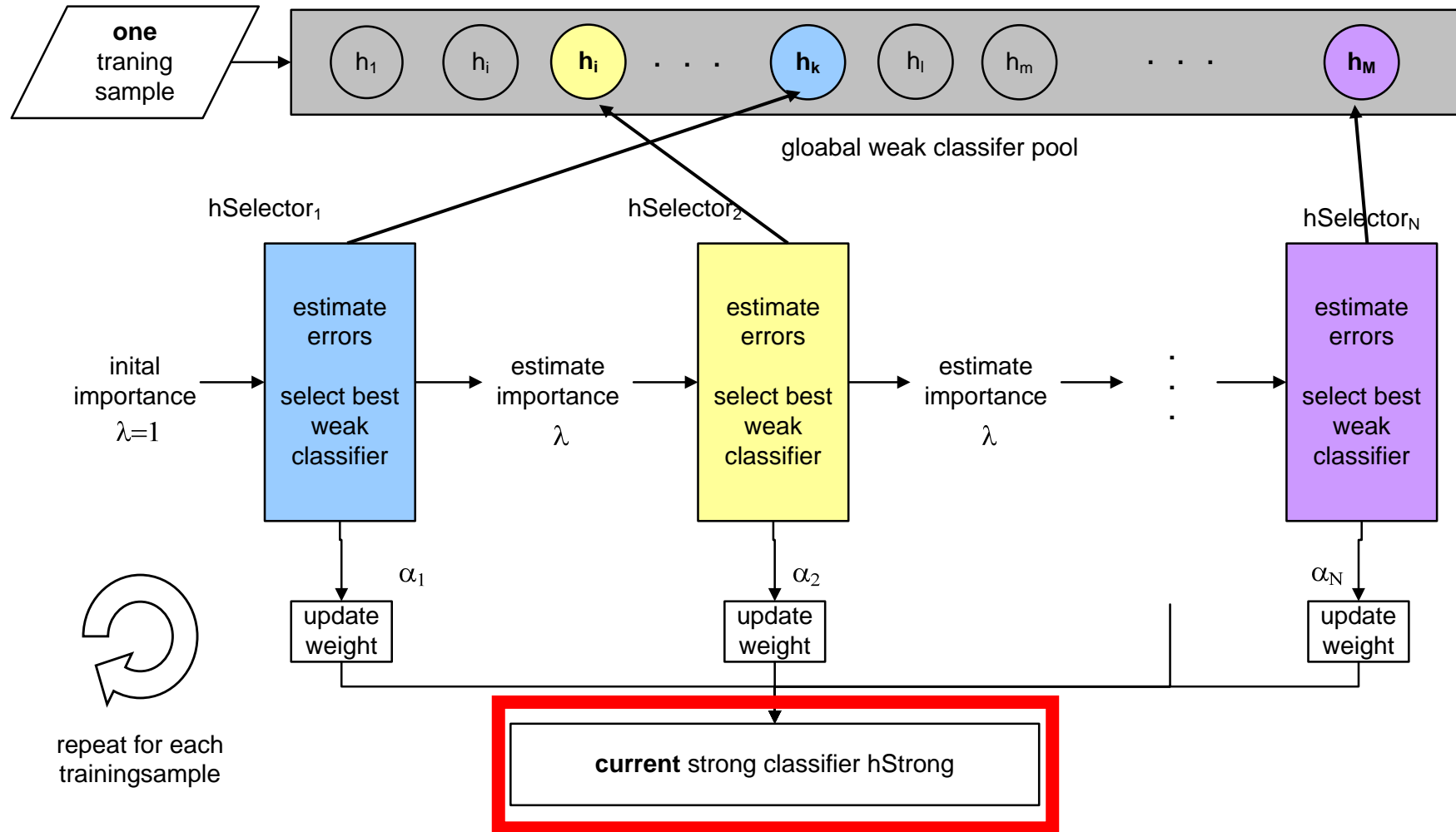
H. Grabner and H. Bischof. **On-line boosting and vision**. CVPR, 2006.

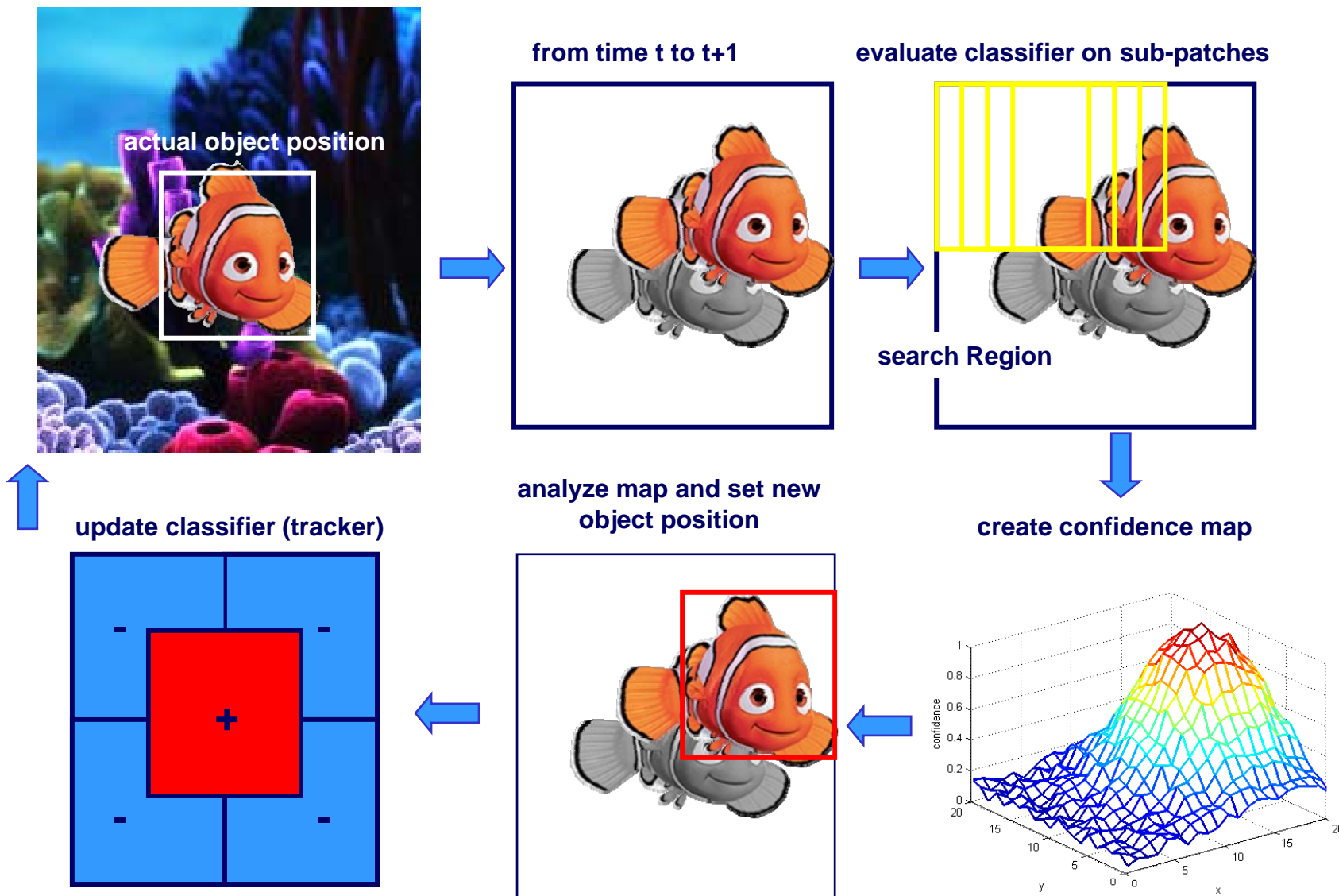




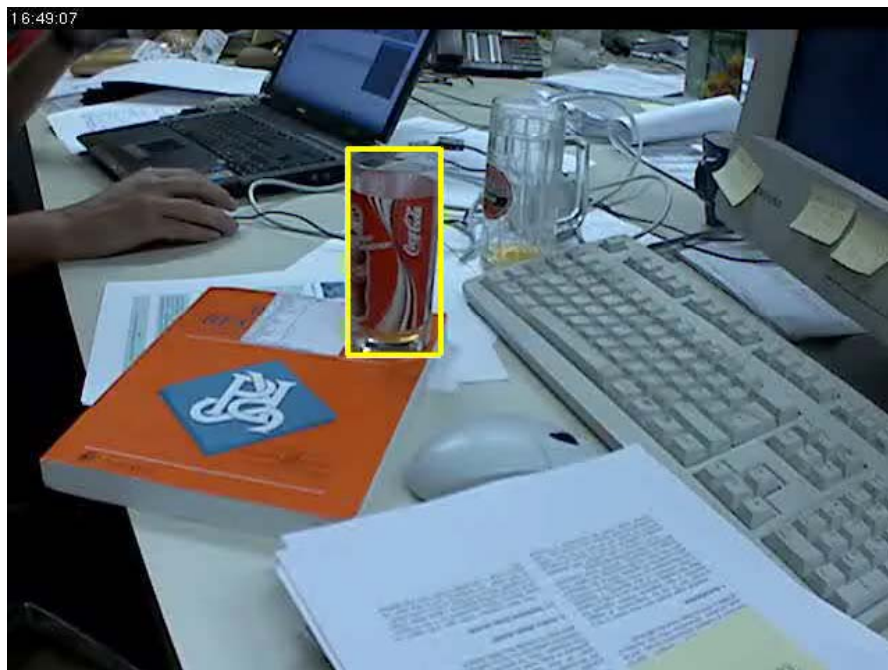




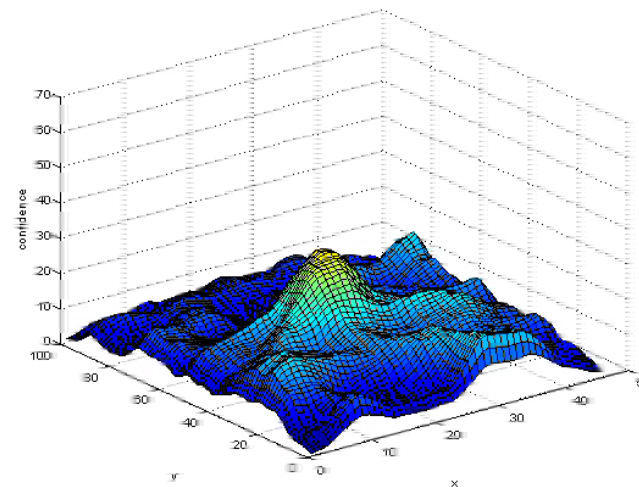




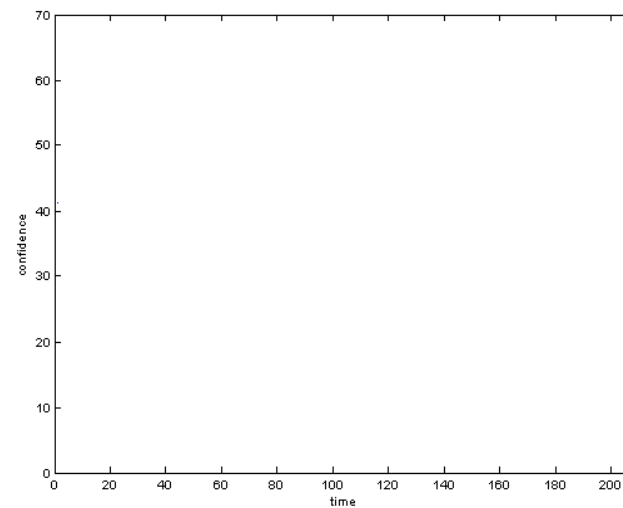
Tracking

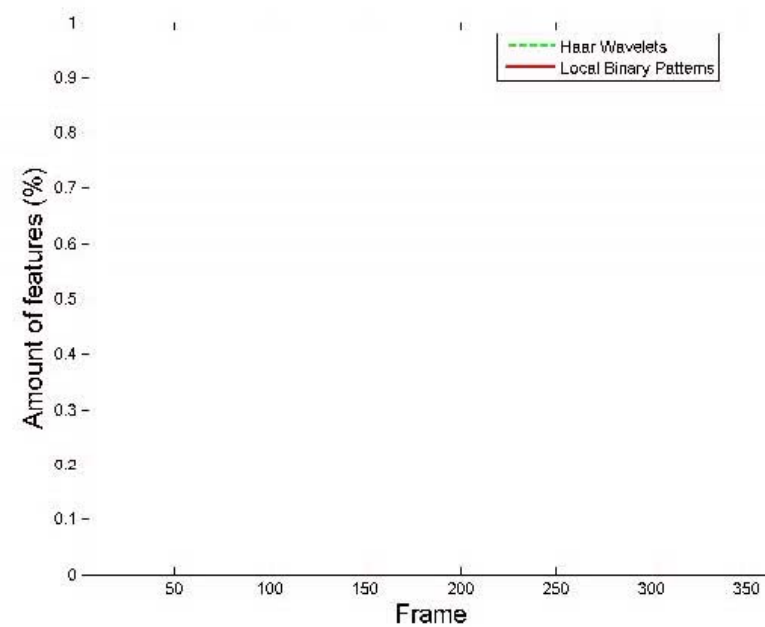
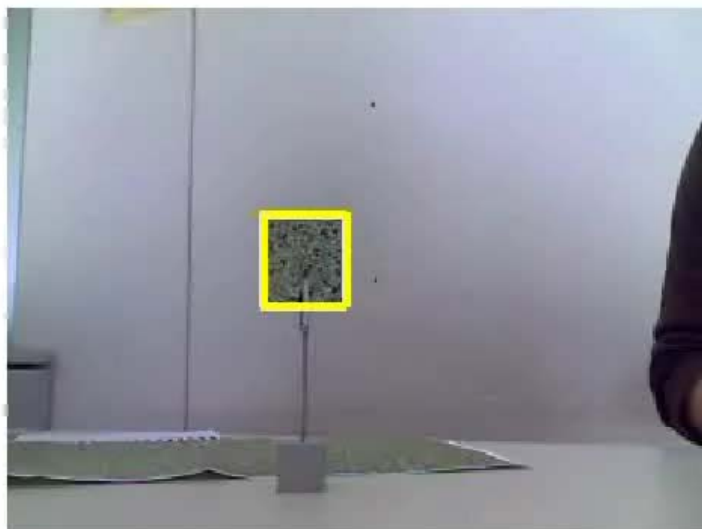


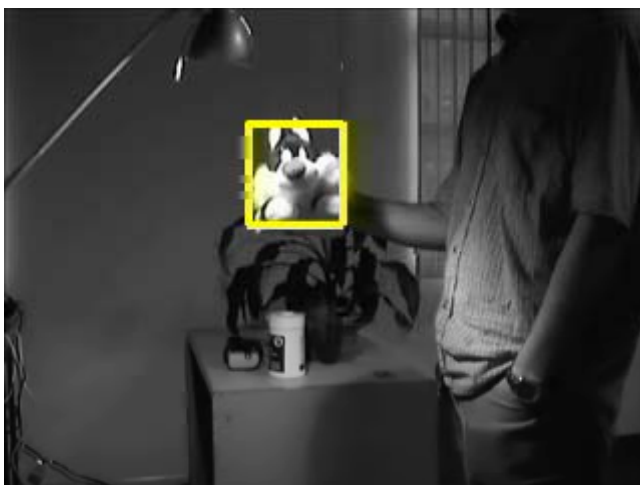
Confidence Map



Max. Confidence Value

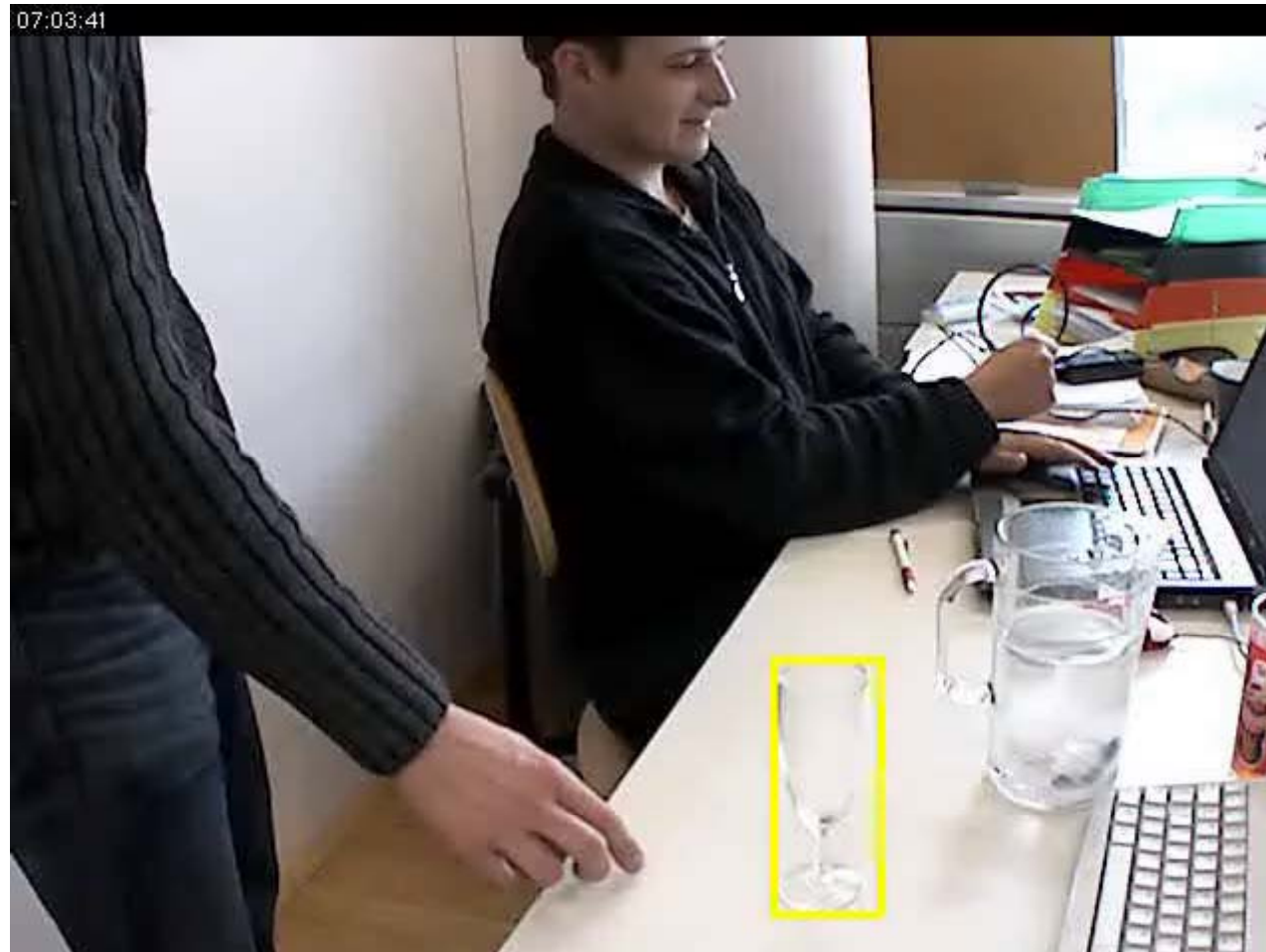






J. Lim, D. Ross, R. Lin, and M. Yang.
Incremental learning for visual tracking. NIPS 2005.

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



◆ Tracking 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