Real-Time Tracking via On-line Boosting

Helmut Grabner, Michael Grabner, Horst Bischof

Graz University of Technology
Institute for Computer Graphics and Vision
Tracking Requirements

♦ Adapivity
  – 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

Tracking as binary classification problem


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

Object vs. background
**Object Detector**


**Fixed Training set**

**General object detector**

\[ \text{sign}(\alpha_1 \cdot \mathcal{I} + \alpha_2 \cdot \mathcal{F} + \alpha_3 \cdot \mathcal{B} + \ldots) \]

**Combination of simple image features using Boosting as Feature Selection**

**Object Tracker**

**On-line update**

**Object vs. Background**

**On-Line Boosting for Feature Selection**

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
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

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

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:
\[ h_{strong}(x) = \text{sign}\left( \sum_{n=1}^{N} \alpha_n \cdot h_{n}^{weak}(x) \right) \]
Given:

- set of labeled training samples
\( \mathcal{X} = \{ \langle x_1, y_1 \rangle, \ldots, \langle x_L, y_L \rangle \mid y_i \pm 1 \} \)
- weight distribution over them
\( D_0 = 1/L \)

for \( n = 1 \) to \( N \)

- train a weak classifier using samples and weight dist.
\( h_n^{\text{weak}}(x) = \mathcal{L}(\mathcal{X}, D_{n-1}) \)
- calculate error \( e_n \)
- calculate weight \( \alpha_n = f(e_n) \)
- update weight dist. \( D_n \)

next

\[ h^{\text{strong}}(x) = \text{sign} \left( \sum_{n=1}^{N} \alpha_n \cdot h_n^{\text{weak}}(x) \right) \]
From Off-line to On-line Boosting

**off-line**

Given:
- set of labeled training samples
  \[ \mathcal{X} = \{ \langle x_1, y_1 \rangle, \ldots, \langle x_L, y_L \rangle \mid y_i \pm 1 \} \]
- weight distribution over them
  \[ D_0 = 1/L \]

for \( n = 1 \) to \( N \)
- train a weak classifier using samples and weight dist.
  \[ h_n^{\text{weak}}(x) = \mathcal{L}(\mathcal{X}, D_{n-1}) \]
- calculate error \( e_n \)
- calculate weight \( \alpha_n = f(e_n) \)
- update weight dist. \( D_n \)

next

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

**on-line**

Given:

for \( n = 1 \) to \( N \)

next

\[ h^{\text{strong}}(x) = \text{sign}( \sum_{n=1}^{N} \alpha_n \cdot h_n^{\text{weak}}(x) ) \]
Given:
- set of labeled training samples
  \( \mathcal{X} = \{ \langle x_1, y_1 \rangle, \ldots, \langle x_L, y_L \rangle \mid y_i \pm 1 \} \)
- weight distribution over them
  \( D_0 = 1/L \)

for \( n = 1 \) to \( N \)
- train a weak classifier using samples and weight dist.
  \( h_{n,\text{weak}}(x) = \mathcal{L}(\mathcal{X}, D_{n-1}) \)
- calculate error \( e_n \)
- calculate weight \( \alpha_n = f(e_n) \)
- update weight dist. \( D_n \)

next

\[
h_{\text{strong}}(x) = \text{sign}( \sum_{n=1}^{N} \alpha_n \cdot h_{n,\text{weak}}(x) )
\]

Given:
- ONE labeled training sample
  \( \langle x, y \rangle \mid y \pm 1 \)
- strong classifier to update

for \( n = 1 \) to \( N \)

next

\[
h_{\text{strong}}(x) = \text{sign}( \sum_{n=1}^{N} \alpha_n \cdot h_{n,\text{weak}}(x) )
\]
Given:

- set of labeled training samples
  $$\mathcal{X} = \{\langle x_1, y_1 \rangle, \ldots, \langle x_L, y_L \rangle \mid y_i \pm 1\}$$
- weight distribution over them
  $$D_0 = 1/L$$

for $$n = 1$$ to $$N$$

- train a weak classifier using samples and weight dist.
  $$h_n^{weak}(x) = \mathcal{L}(\mathcal{X}, D_{n-1})$$
- calculate error $$e_n$$
- calculate weight $$\alpha_n = f(e_n)$$
- update weight dist. $$D_n$$

next

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

For on-line boosting:

Given:

- ONE labeled training sample
  $$\langle x, y \rangle \mid y \pm 1$$
- strong classifier to update
- initial importance $$\lambda = 1$$

for $$n = 1$$ to $$N$$

- update importance weight $$\lambda$$

next

$$h^{strong}(x) = \text{sign} \left( \sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(x) \right)$$
From Off-line to On-line Boosting

Given:
- set of labeled training samples \( \mathcal{X} = \{ \langle x_1, y_1 \rangle, ..., \langle x_L, y_L \rangle \mid y_i \pm 1 \} \)
- weight distribution over them \( D_0 = 1/L \)

for \( n = 1 \) to \( N \)
- train a weak classifier using samples and weight dist.
  \( h_n^{\text{weak}}(x) = \mathcal{C}(\mathcal{X}, D_{n-1}) \)
- calculate error \( e_n \)
- calculate weight \( \alpha_n = f(e_n) \)
- update weight dist. \( D_n \)
next

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

Given:
- ONE labeled training sample \( \langle x, y \rangle \mid y \pm 1 \)
- strong classifier to update
- initial importance \( \lambda = 1 \)

for \( n = 1 \) to \( N \)
- update the weak classifier using samples and importance
  \( h_n^{\text{weak}}(x) = \mathcal{C}(h_n^{\text{weak}}, \langle x, y \rangle, \lambda) \)
- update importance weight \( \lambda \)
next

\( h^{\text{strong}}(x) = \text{sign}(\sum_{n=1}^{N} \alpha_n \cdot h_n^{\text{weak}}(x)) \)
From Off-line to On-line Boosting

Given:
- set of labeled training samples \( \mathcal{X} = \{\langle x_1, y_1 \rangle, \ldots, \langle x_L, y_L \rangle \mid y_i \pm 1 \} \)
- weight distribution over them \( D_0 = 1/L \)

for \( n = 1 \) to \( N \)
- train a weak classifier using samples and weight dist.
  \( h_{n}^{\text{weak}}(x) = \mathcal{L}(\mathcal{X}, D_{n-1}) \)
- calculate error \( e_n \)
- calculate weight \( \alpha_n = f(e_n) \)
- update weight dist. \( D_n \)

next

\( h^{\text{strong}}(x) = \text{sign}(\sum_{n=1}^{N} \alpha_n \cdot h_{n}^{\text{weak}}(x)) \)

on-line

Given:
- ONE labeled training sample \( \langle x, y \rangle \mid y \pm 1 \)
- strong classifier to update
- initial importance \( \lambda = 1 \)

for \( n = 1 \) to \( N \)
- update the weak classifier using samples and importance
  \( h_{n}^{\text{weak}}(x) = \mathcal{L}(h_{n}^{\text{weak}}, \langle x, y \rangle, \lambda) \)
- update error estimation \( \tilde{e}_n \)
- update weight \( \alpha_n = f(\tilde{e}_n) \)
- update importance weight \( \lambda \)

next

\( h^{\text{strong}}(x) = \text{sign}(\sum_{n=1}^{N} \alpha_n \cdot h_{n}^{\text{weak}}(x)) \)
From Off-line to On-line Boosting

off-line

Given:
- set of labeled training samples
  \( \mathcal{X} = \{\langle x_1, y_1 \rangle, \ldots, \langle x_L, y_L \rangle \mid y_i \pm 1 \} \)
- weight distribution over them
  \( D_0 = 1/L \)

for \( n = 1 \) to \( N \)
- train a weak classifier using samples and weight dist.
  \( h_{n}^{\text{weak}}(x) = \mathcal{L}(\mathcal{X}, D_{n-1}) \)
- calculate error \( e_n \)
- calculate weight \( \alpha_n = f(e_n) \)
- update weight dist. \( D_n \)
next

\[ h_{\text{strong}}(x) = \text{sign}( \sum_{n=1}^{N} \alpha_n \cdot h_{n}^{\text{weak}}(x)) \]

on-line

Given:
- ONE labeled training sample
  \( \langle x, y \rangle \mid y \pm 1 \)
- strong classifier to update
- initial importance \( \lambda = 1 \)

for \( n = 1 \) to \( N \)
- update the weak classifier using samples and importance
  \( h_{n}^{\text{weak}}(x) = \mathcal{L}(h_{n}^{\text{weak}}, \langle x, y \rangle, \lambda) \)
- update error estimation \( \hat{e}_n \)
- update weight \( \alpha_n = f(\hat{e}_n) \)
- update importance weight \( \lambda \)
next

\[ h_{\text{strong}}(x) = \text{sign}( \sum_{n=1}^{N} \alpha_n \cdot h_{n}^{\text{weak}}(x)) \]
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
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
- update weak classifier using sample and importance
- update error estimation
- update weight
- update importance weight
next

Converges to the off-line results...


Result:

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

British Machine Vision Conference, Real-Time Tracking via On-line Boosting
Edinburgh, Sep. 05, 2006
Helmut Grabner
♦ Each feature corresponds to a weak classifier

\[ P(-1|f_i(x)) \]
\[ P(+1|f_i(x)) \]

\[ f_i(x) \]

♦ Features

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

Fast computation using efficient data structures

- integral images
- integral histograms

Introducing “Selector”

- selects one feature from its local feature pool

\[ \mathcal{H}^{weak} = \{ h_1^{weak}, \ldots, h_M^{weak} \} \]
\[ \mathcal{F} = \{ f_1, \ldots, f_M \} \]

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

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

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

Use a shared feature pool

$F = F_1 = \ldots = F_N$

$H_{\text{weak}} = H_{\text{weak}}^1 = \ldots = H_{\text{weak}}^N$

\[ \alpha_1, \alpha_2, \ldots, \alpha_N \]

\text{current strong classifier } h_{\text{Strong}}
Direct Feature Selection

one training sample

h1, h2, ..., hm

global weak classifier pool

hSelector1, hSelector2, ..., hSelectorN

global weak classifier pool

estimate errors
select best weak classifier

estimate errors
select best weak classifier

repeat for each training sample

alpha1

alpha2

alphaN

current strong classifier hStrong

update weight

update weight

update weight
Direct Feature Selection

one

training

sample

\( h_1 \), \( h_i \), \( h_l \), \ldots, \( h_k \), \( h_l \), \( h_m \), \ldots, \( h_M \)

global weak classifier pool

\( h_{\text{Selector}_1} \)

\( h_{\text{Selector}_2} \)

\( h_{\text{Selector}_N} \)

\( \text{estimate errors} \)

\( \text{select best weak classifier} \)

initial importance \( \lambda = 1 \)

\( \alpha_1 \)

update weight

\( \text{estimate importance} \)

\( \lambda \)

\( \alpha_2 \)

update weight

repeat for each training sample

\( \text{current strong classifier} h_{\text{Strong}} \)
Direct Feature Selection

One training sample

\[ h_1, h_i, h_i, \ldots, h_k, h_i, h_m, \ldots, h_M \]

global weak classifier pool

hSelector_1, hSelector_2, hSelector_N

estimate errors

select best weak classifier

initial importance \( \lambda = 1 \)

update weight \( \alpha_1 \)

repeat for each training sample

\[ current \] strong classifier \( h_{Strong} \)

Edinburgh, Sep. 05, 2006

Helmut Grabner
Direct Feature Selection

one training sample

hSelector1

estimate errors

select best weak classifier

initial importance \( \lambda = 1 \)

update weight \( \alpha_1 \)

repeat for each training sample

hSelector2

estimate errors

select best weak classifier

update weight \( \alpha_2 \)

hSelectorN

estimate errors

select best weak classifier

update weight \( \alpha_N \)

current strong classifier \( h_{\text{Strong}} \)
Tracking 1/2

From time t to t+1, actual object position

Evaluate classifier on sub-patches

Search Region

Analyze map and set new object position

Create confidence map

Update classifier (tracker)
On-line Feature Exchange

British Machine Vision Conference, Real-Time Tracking via On-line Boosting

Edinburgh, Sep. 05, 2006

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

**Robust online appearance models for visual tracking.** CVPR 2001.
“Tracking the Invisible”

British Machine Vision Conference, Real-Time Tracking via On-line Boosting

Edinburgh, Sep. 05, 2006

Helmut Grabner
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