Real-Time Tracking via On-line Boosting

Helmut Grabner, Michael Grabner, Horst Bischof
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 binary classification

Tracking as Classification

- Tracking as binary classification problem

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

object vs. background

British Machine Vision Conference, Real-Time Tracking via On-line Boosting
Edinburgh, Sep. 05, 2006
Helmut Grabner
Object Detector


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

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
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
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:

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

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

on-line

Given:

for n = 1 to N
next

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

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

British Machine Vision Conference, Real-Time Tracking via On-line Boosting
Edinburgh, Sep. 05, 2006
Helmut Grabner
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.
  \[ h_n^{weak}(x) = \mathcal{L}(\mathcal{X}, D_{n-1}) \]
- calculate error \( e_n \)
- calculate weight \( \alpha_n = f(e_n) \)
- update weight dist. \( D_n \)

next

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

For on-line:
- 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

\[ h^{strong}(x) = \text{sign}(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(x)) \]
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
  - 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 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.
  \[ h_n^{weak}(x) = \mathcal{L}(\mathcal{X}, D_{n-1}) \]
- calculate error \( e_n \)
- calculate weight \( \alpha_n = f(e_n) \)
- update weight dist. \( D_n \)

next

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

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
  \[ h_n^{weak}(x) = \mathcal{L}(h_n^{weak}, \langle x, y \rangle, \lambda) \]
- update error estimation \( \tilde{e}_n \)
- update weight \( \alpha_n = f(\tilde{e}_n) \)
- update importance weight \( \lambda \)

next

\[ h_n^{strong}(x) = \text{sign}( \sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(x)) \]
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 \( \widehat{e}_n \)
- update weight \( \alpha_n = f(\widehat{e}_n) \)
- 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
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
Given:
- ONE labeled training sample
- strong classifier to update

Algorithm:
- initial importance

Result:
\[ h_{\text{strong}}(x) = \text{sign} \left( \sum_{n=1}^{N} \alpha_n \cdot h_{n}\text{weak}(x) \right) \]
Each feature corresponds to a weak classifier

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

\[h^{\text{weak}}(x)
\]

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

\[ \mathcal{H}_{\text{weak}} = \{ h_1^{\text{weak}}, \ldots, h_M^{\text{weak}} \} \]

\[ F = \{ f_1, \ldots, f_M \} \]

\[ h_{\text{sel}}(x) = h_m^{\text{weak}}(x) \]

\[ m = \arg \min_i e_i \]

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

\[
\mathcal{F} = \mathcal{F}_1 = \ldots = \mathcal{F}_N
\]

\[
\mathcal{H}^{\text{weak}} = \mathcal{H}_1^{\text{weak}} = \ldots = \mathcal{H}_N^{\text{weak}}
\]
Direct Feature Selection

- Initial importance \( \lambda = 1 \)
- One training sample
- Global weak classifier pool
- Estimate errors
- Select best weak classifier
- \( h_1 \) to \( h_M \)
- \( h_{\text{Selector}_1} \) to \( h_{\text{Selector}_N} \)
- Update weight
- \( \alpha_1 \) to \( \alpha_N \)
- Current strong classifier \( h_{\text{Strong}} \)

British Machine Vision Conference, Real-Time Tracking via On-line Boosting
Edinburgh, Sep. 05, 2006
Helmut Grabner
Direct Feature Selection

**one training sample**

- **h₁, h₂, h₃, ..., hₖ, hₗ, ..., hₘ, ..., hₘ**
- **global weak classifier pool**

- **hSelector₁**
  - estimate errors
  - select best weak classifier
  - initial importance \( \lambda = 1 \)
  - update weight \( \alpha₁ \)

- **hSelector₂**
  - estimate errors
  - select best weak classifier
  - estimate importance \( \lambda \)
  - update weight \( \alpha₂ \)

- **hSelectorₙ**
  - estimate errors
  - select best weak classifier
  - estimate importance \( \lambda \)
  - update weight \( \alphaₙ \)

**current strong classifier hStrong**

repeat for each trainingsample

British Machine Vision Conference, Real-Time Tracking via On-line Boosting
Edinburgh, Sep. 05, 2006
Helmut Grabner
Direct Feature Selection

one training sample

\[ h_1, h_2, \ldots, h_k, h_{l}, \ldots, h_m, \ldots, h_M \]

global weak classifier pool

\[
\text{hSelector}_1 \quad \text{estimate errors} \quad \text{select best weak classifier} \\
\text{estimate importance} \quad \lambda = 1 \\
\text{update weight} \quad \alpha_1
\]

\[
\text{hSelector}_2 \quad \text{estimate errors} \quad \text{select best weak classifier} \\
\text{estimate importance} \quad \lambda \\
\text{update weight} \quad \alpha_2
\]

\[
\text{hSelector}_N \quad \text{estimate errors} \quad \text{select best weak classifier} \\
\text{estimate importance} \quad \lambda \\
\text{update weight} \quad \alpha_N
\]

\[
\text{current strong classifier } h_{\text{Strong}}
\]

repeat for each training sample

British Machine Vision Conference, Real-Time Tracking via On-line Boosting
Edinburgh, Sep. 05, 2006
Helmut Grabner
Direct Feature Selection

one training sample

\[ h_1 \quad h_2 \quad h_3 \quad \ldots \quad h_k \quad h_{k+1} \quad \ldots \quad h_M \]

global weak classifier pool

\( h_{Selector_1} \)
\( h_{Selector_2} \)
\( h_{Selector_N} \)

estimate errors
select best weak classifier

initial importance \( \lambda = 1 \)

update weight \( \alpha_1 \)

repeat for each training sample

\[ \text{current strong classifier } h_{Strong} \]
Tracking 1/2

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

British Machine Vision Conference, Real-Time Tracking via On-line Boosting
Edinburgh, Sep. 05, 2006
Helmut Grabner
Tracking

Confidence Map

Max. Confidence Value

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

Edinburgh, Sep. 05, 2006

Helmut Grabner
British Machine Vision Conference, Real-Time Tracking via On-line Boosting  
Edinburgh, Sep. 05, 2006  
Helmut Grabner
Public Sequences


“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