

# Simple Features, AdaBoost und Kaskaden

Niklas Beuter

10. Mai 2006

# Idee

Paper: *Viola & Jones, Rapid Object Detection using a Boosted Cascade of Simple Features*

- Gesichtserkennung
- Möglichst gute Erkennungsrate, wenige *False-Positives*

- 1 Inhalt
  - Umsetzung
- 2 Erklärung
  - Simple Features
  - Integralbild
  - AdaBoost
  - Kaskaden
- 3 Zusammenfassung
  - Ergebnisse
  - Analyse

# Umsetzung

## Wichtige Beiträge zur Umsetzung

- Suchfenster zum Auffinden des gesuchten Objekts
- Objektdetektion durch *Simple Features*
- Schnelle Berechnung durch *Integralbild*
- Lernalgorithmus *Adaboost*
- Verschaltung durch *Kaskaden*

# Umsetzung

## Wichtige Beiträge zur Umsetzung

- Suchfenster zum Auffinden des gesuchten Objekts
- Objektdetektion durch *Simple Features*
- Schnelle Berechnung durch *Integralbild*
- Lernalgorithmus *Adaboost*
- Verschaltung durch *Kaskaden*

# Umsetzung

## Wichtige Beiträge zur Umsetzung

- Suchfenster zum Auffinden des gesuchten Objekts
- Objektdetektion durch *Simple Features*
- Schnelle Berechnung durch *Integralbild*
- Lernalgorithmus *Adaboost*
- Verschaltung durch *Kaskaden*

# Umsetzung

## Wichtige Beiträge zur Umsetzung

- Suchfenster zum Auffinden des gesuchten Objekts
- Objektdetektion durch *Simple Features*
- Schnelle Berechnung durch *Integralbild*
- Lernalgorithmus *Adaboost*
- Verschaltung durch *Kaskaden*

# Umsetzung

## Wichtige Beiträge zur Umsetzung

- Suchfenster zum Auffinden des gesuchten Objekts
- Objektdetektion durch *Simple Features*
- Schnelle Berechnung durch *Integralbild*
- Lernalgorithmus *Adaboost*
- Verschaltung durch *Kaskaden*

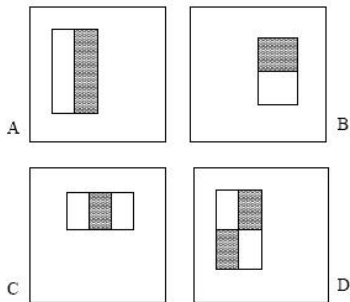


# Umsetzung

## Wichtige Beiträge zur Umsetzung

- Suchfenster zum Auffinden des gesuchten Objekts
- Objektdetektion durch *Simple Features*
- Schnelle Berechnung durch *Integralbild*
- Lernalgorithmus *Adaboost*
- Verschaltung durch *Kaskaden*

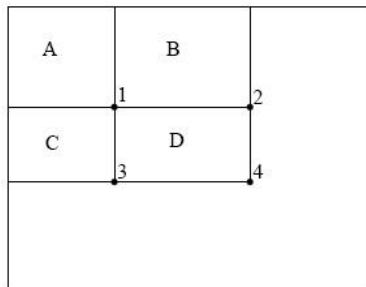
# Simple Features



- Rechtecke robuster als Pixel
- Einfach zu skalieren
- Einfacher Vergleich der dunklen und hellen Flächen

Abbildung: 4 einfache Merkmale

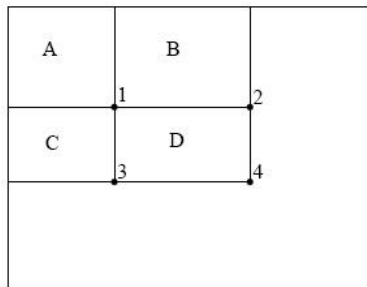
# Integralbild



- Schnelle Berechnung der Rechtecke durch Integralbild
- Konstanter Zeitaufwand
  - z.B. bei zwei gleichen Rechtecken nur noch 6 Rechenschritte

Abbildung: 4 Berechnungen pro Rechteck

# Integralbild



- Schnelle Berechnung der Rechtecke durch Integralbild
- Konstanter Zeitaufwand
  - z.B. bei zwei gleichen Rechtecken nur noch 6 Rechenschritte

Abbildung: 4 Berechnungen pro Rechteck

# Boosting

- Auffinden der aussagekräftigen Merkmale , sonst über 160.000 Merkmale für ein 24x24 Pixel Suchfenster
- Vorher: Bildnormalisierung  $\Rightarrow$  Kontrasterhöhung, Beleuchtungsinvarianz
- Beruht auf Boosting (Zweiklassenproblem)
- Basisklassifikator  $\rightarrow$  *weak classifier*
- Geschicktes Trainieren mehrerer *weak classifier* erbringt starken Klassifikator

# Boosting

- Auffinden der aussagekräftigen Merkmale , sonst über 160.000 Merkmale für ein 24x24 Pixel Suchfenster
- Vorher: Bildnormalisierung  $\Rightarrow$  Kontrasterhöhung, Beleuchtungsinvarianz
- Beruht auf Boosting (Zweiklassenproblem)
- Basisklassifikator  $\rightarrow$  *weak classifier*
- Geschicktes Trainieren mehrerer *weak classifier* erbringt starken Klassifikator

# AdaBoost

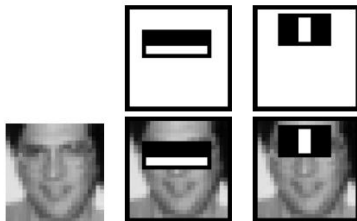
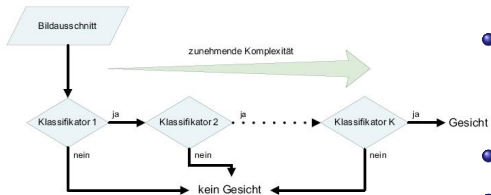


Abbildung: Features selected by AdaBoost

- AdaBoost erzeugt beliebig viele *weak classifier*, die durch Linearkombination den endgültigen Klassifikator bilden
- gewichtetes Training, verlagert sich auf Extremfälle
- Trainingsfehler kann beliebig verringert werden
- Optimierungsproblem für jedes Merkmal

# Kaskaden



- Linearer Entscheidungsbaum mit aufsteigender Komplexität
- Alle positiv  $\rightarrow$  Gesicht
- Training durch *Bootstrapping*

Abbildung: K Einzelklassifikatoren



# Ergebnisse

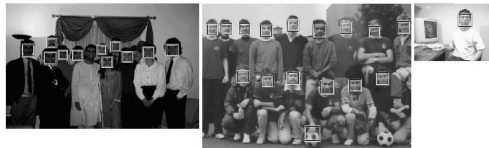


Abbildung: Test images from the MIT+CMU

- 4916 Gesichter, 9544 Hintergründe
- Fenster mit  $24 \times 24$  Pixel
- 700 Mhz P III,  $384 \times 288$  Pixel
- 15 mal schneller als jeder bisherige
- Pro Bild ca 0.7 Sekunden

# Ergebnisse

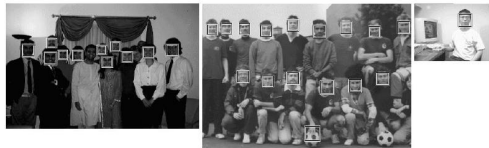


Abbildung: Test images from the MIT+CMU

- 4916 Gesichter, 9544 Hintergründe
- Fenster mit  $24 \times 24$  Pixel
- 700 Mhz P III,  $384 \times 288$  Pixel
- 15 mal schneller als jeder bisherige
- Pro Bild ca 0.7 Sekunden

# Analyse

Detector \ False detections	10	31	50	65	78	95	167
Viola-Jones	76.1%	88.4%	91.4%	92.0%	92.1%	92.9%	93.9%
Viola-Jones (voting)	81.1%	89.7%	92.1%	93.1%	93.1%	93.2%	93.7%
Rowley-Baluja-Kanade	83.2%	86.0%	-	-	-	89.2%	90.1%
Schneiderman-Kanade	-	-	-	94.4%	-	-	-
Roth-Yang-Ahuja	-	-	-	-	(94.8%)	-	-

Abbildung: Detektionsrate auf MIT+CMU test set [130 Bilder, 507 Gesichter]

- Sehr schnell
- Gute Erkennungsrate
- Im Prinzip auf jedes Objekt anwendbar
- Große Bilddatenbank, viele Subwindows  $\Rightarrow$  langes Lernen
- Viele Parameter  $\Rightarrow$  schwierige Einstellungen

# Analyse

Detector \ False detections	10	31	50	65	78	95	167
Viola-Jones	76.1%	88.4%	91.4%	92.0%	92.1%	92.9%	93.9%
Viola-Jones (voting)	81.1%	89.7%	92.1%	93.1%	93.1%	93.2%	93.7%
Rowley-Baluja-Kanade	83.2%	86.0%	-	-	-	89.2%	90.1%
Schneiderman-Kanade	-	-	-	94.4%	-	-	-
Roth-Yang-Ahuja	-	-	-	-	(94.8%)	-	-

Abbildung: Detektionsrate auf MIT+CMU test set [130 Bilder, 507 Gesichter]

- Sehr schnell
- Gute Erkennungsrate
- Im Prinzip auf jedes Objekt anwendbar
- Große Bilddatenbank, viele Subwindows  $\Rightarrow$  langes Lernen
- Viele Parameter  $\Rightarrow$  schwierige Einstellungen