

# Gesichtserkennung

Marcel Bacher, Christoph Koller, Lukas Ott

Universität Salzburg

25. Jänner 2019



- 1 Einleitung
- 2 Technik
- 3 Anwendungen
- 4 Vor- Nachteile

## 1 Einleitung

## 2 Technik

## 3 Anwendungen

## 4 Vor- Nachteile

# Was ist Gesichtserkennung?

- Kategorie: biometrische Software

# Was ist Gesichtserkennung?

- Kategorie: biometrische Software
- Geschichte der Gesichtserkennung
  - sehr junge Wissenschaft (etwas mehr als 10 Jahre)
  - Ursprünge im US-amerikanischen "Department of Defence"
  - anfängliche nur technische Versuchsversionen
  - mittlerweile im echten Einsatz

# Was ist Gesichtserkennung?

- Kategorie: biometrische Software
- Geschichte der Gesichtserkennung
  - sehr junge Wissenschaft (etwas mehr als 10 Jahre)
  - Ursprünge im US-amerikanischen "Department of Defence"
  - anfängliche nur technische Versuchsversionen
  - mittlerweile im echten Einsatz
- Grobe Vorgehensweise:
  - Personen in Kameraaufnahmen oder Bildern erkannt
  - charakteristische Merkmale werden digitalisiert
  - mit zuvor gespeicherten Gesichtsbildern verglichen

# Was ist Gesichtserkennung?

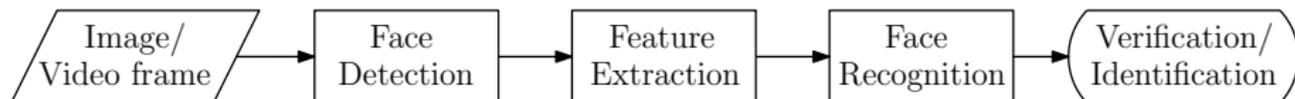
- Kategorie: biometrische Software
- Geschichte der Gesichtserkennung
  - sehr junge Wissenschaft (etwas mehr als 10 Jahre)
  - Ursprünge im US-amerikanischen "Department of Defence"
  - anfängliche nur technische Versuchsversionen
  - mittlerweile im echten Einsatz
- Grobe Vorgehensweise:
  - Personen in Kameraaufnahmen oder Bildern erkannt
  - charakteristische Merkmale werden digitalisiert
  - mit zuvor gespeicherten Gesichtsbildern verglichen
- Themenfeld entwickelt sich sehr rasch weiter
  - zB China
  - Smartphones, Überwachung, Videospiele (Kinect)

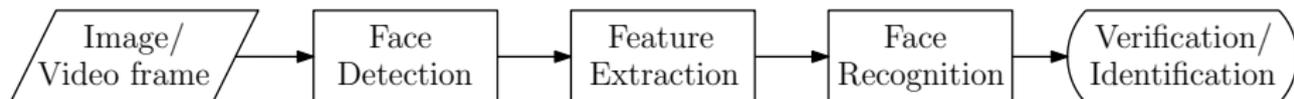
1 Einleitung

**2 Technik**

3 Anwendungen

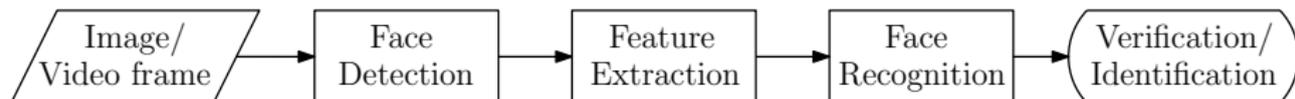
4 Vor- Nachteile





## Face Detection:

- Erkennung ob und wie viele Gesichter sich im Bild befinden
- Preprocessing:
  - vor: Gray-scale, Normalization (Anpassen von Kontrast, Licht, ...)
  - nach: Gesichter zuschneiden, skalieren, drehen, ...



## Face Detection:

- Erkennung ob und wie viele Gesichter sich im Bild befinden
- Preprocessing:
  - vor: Gray-scale, Normalization (Anpassen von Kontrast, Licht, ...)
  - nach: Gesichter zuschneiden, skalieren, drehen, ...

**Knowledge-based:** basierend auf Regeln

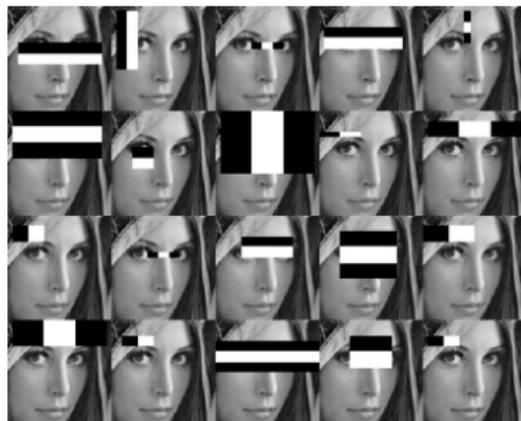
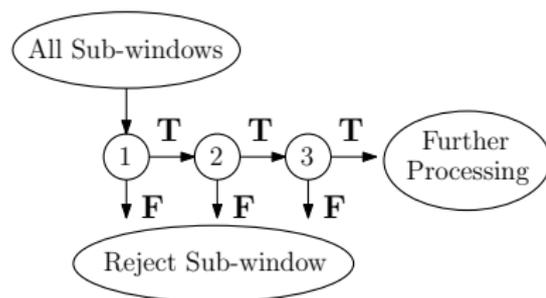
**Feature-based:** über Erkennung von Augen, Nase, Mund, etc.

**Template-matching:** Vergleich mit Beispielgesichtern/-features

**Appearance-based:** Statistische Analyse und Machine Learning

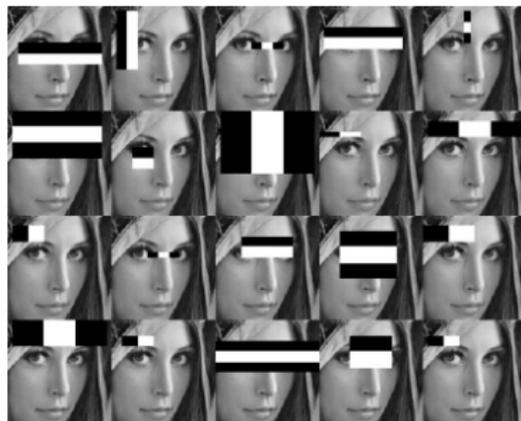
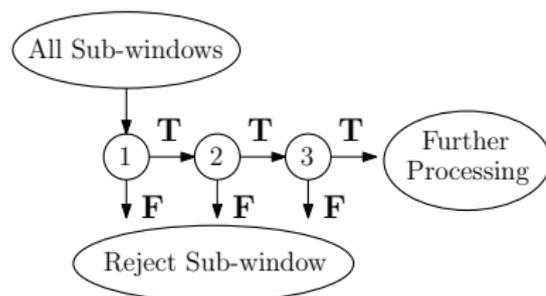
# Viola-Jones Algorithmus

- Findet Features mittels “Haar-like Features”
- Lernt, welche Features wichtig sind
- kaskadierendes Klassifizieren der Features



# Viola-Jones Algorithmus

- Findet Features mittels “Haar-like Features”
- Lernt, welche Features wichtig sind
- kaskadierendes Klassifizieren der Features



- Andere Techniken: Neuronale Netze, Support Vector Machines, Eigenfaces, ...

# Eigenfaces

Die Idee ist, dass jedes Gesicht eine Linearkombination von "Standardgesichtern" ist, sog. "Eigenfaces".

# Eigenfaces

Die Idee ist, dass jedes Gesicht eine Linearkombination von "Standardgesichtern" ist, sog. "Eigenfaces".

## Linearkombination

Sei  $V$  ein Vektorraum über  $\mathbb{R}$ . Seien  $v_1, \dots, v_n \in V$  und  $x_1, \dots, x_n \in \mathbb{R}$ . Dann heißt  $v = x_1 v_1 + \dots + x_n v_n = \sum_{i=1}^n x_i v_i$  eine Linearkombination von  $v_1, \dots, v_n$ .

# Eigenfaces

Die Idee ist, dass jedes Gesicht eine Linearkombination von "Standardgesichtern" ist, sog. "Eigenfaces".

## Linearkombination

Sei  $V$  ein Vektorraum über  $\mathbb{R}$ . Seien  $v_1, \dots, v_n \in V$  und  $x_1, \dots, x_n \in \mathbb{R}$ . Dann heißt  $v = x_1 v_1 + \dots + x_n v_n = \sum_{i=1}^n x_i v_i$  eine Linearkombination von  $v_1, \dots, v_n$ .

D.h. ein Gesicht besteht zu 20% aus  $EF_1$ , 7% aus  $EF_2$ , ...

# Eigenfaces

Die Idee ist, dass jedes Gesicht eine Linearkombination von "Standardgesichtern" ist, sog. "Eigenfaces".

## Linearkombination

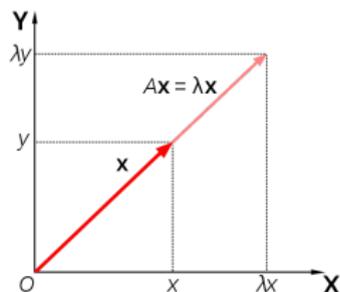
Sei  $V$  ein Vektorraum über  $\mathbb{R}$ . Seien  $v_1, \dots, v_n \in V$  und  $x_1, \dots, x_n \in \mathbb{R}$ . Dann heißt  $v = x_1 v_1 + \dots + x_n v_n = \sum_{i=1}^n x_i v_i$  eine Linearkombination von  $v_1, \dots, v_n$ .

D.h. ein Gesicht besteht zu 20% aus  $EF_1$ , 7% aus  $EF_2$ , ...

Ein "Eigenface" ist ein Eigenvektor:

## Eigenvektor

Sei  $v \in \mathbb{R}^k$  und  $A$  eine  $k \times k$ -Matrix. Ist  $Av = \lambda v$  mit  $\lambda \in \mathbb{R}$ , heißt  $v$  Eigenvektor.



# Eigenfaces

Die Idee ist, dass jedes Gesicht eine Linearkombination von "Standardgesichtern" ist, sog. "Eigenfaces".

## Linearkombination

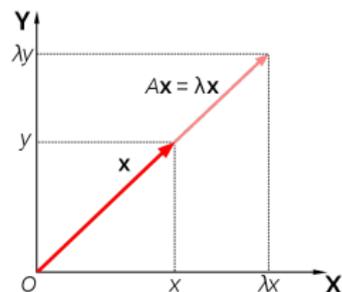
Sei  $V$  ein Vektorraum über  $\mathbb{R}$ . Seien  $v_1, \dots, v_n \in V$  und  $x_1, \dots, x_n \in \mathbb{R}$ . Dann heißt  $v = x_1 v_1 + \dots + x_n v_n = \sum_{i=1}^n x_i v_i$  eine Linearkombination von  $v_1, \dots, v_n$ .

D.h. ein Gesicht besteht zu 20% aus  $EF_1$ , 7% aus  $EF_2$ , ...

Ein "Eigenface" ist ein Eigenvektor:

## Eigenvektor

Sei  $v \in \mathbb{R}^k$  und  $A$  eine  $k \times k$ -Matrix. Ist  $Av = \lambda v$  mit  $\lambda \in \mathbb{R}$ , heißt  $v$  Eigenvektor.



Eigenfaces werden mittels Principal Component Analysis (PCA) berechnet.

# Berechnung der Eigenfaces

- 1 Gegeben sind  $N$  Training-Bilder (gleiche Auflösung,  $Z \times S = P$ ).
- 2 Forme jedes Bild in  $P \times 1$ -Vektor um (Zeilen konkatenieren).

# Berechnung der Eigenfaces

- 1 Gegeben sind  $N$  Training-Bilder (gleiche Auflösung,  $Z \times S = P$ ).
- 2 Forme jedes Bild in  $P \times 1$ -Vektor um (Zeilen konkatenieren).
- 3 Bilder zu  $P \times N$ -Matrix  $T$  verbinden (ein Bild ist eine Spalte).
- 4 Mittelwert  $M$  der Bilder berechnen und von jedem Bild in  $T$  subtrahieren.

# Berechnung der Eigenfaces

- 1 Gegeben sind  $N$  Training-Bilder (gleiche Auflösung,  $Z \times S = P$ ).
- 2 Forme jedes Bild in  $P \times 1$ -Vektor um (Zeilen konkatenieren).
- 3 Bilder zu  $P \times N$ -Matrix  $T$  verbinden (ein Bild ist eine Spalte).
- 4 Mittelwert  $M$  der Bilder berechnen und von jedem Bild in  $T$  subtrahieren.
- 5 Eigenvektoren  $V_i$  und Eigenwerte  $\lambda_i$  für Kovarianzmatrix  $S = TT^T$  berechnen. Eigenvektoren sind  $P \times 1$ -Vektoren, und deshalb Eigenfaces.

# Berechnung der Eigenfaces

- 1 Gegeben sind  $N$  Training-Bilder (gleiche Auflösung,  $Z \times S = P$ ).
- 2 Forme jedes Bild in  $P \times 1$ -Vektor um (Zeilen konkatenieren).
- 3 Bilder zu  $P \times N$ -Matrix  $T$  verbinden (ein Bild ist eine Spalte).
- 4 Mittelwert  $M$  der Bilder berechnen und von jedem Bild in  $T$  subtrahieren.
- 5 Eigenvektoren  $V_i$  und Eigenwerte  $\lambda_i$  für Kovarianzmatrix  $S = TT^T$  berechnen. Eigenvektoren sind  $P \times 1$ -Vektoren, und deshalb Eigenfaces.
- 6 Sortiere Eigenvektoren nach deren Eigenwerten.
- 7 Wähle  $k$  erste Eigenvektoren, die  $\frac{\lambda_1 + \lambda_2 + \dots + \lambda_k}{\lambda_1 + \lambda_2 + \dots + \lambda_N} > \epsilon$  erfüllen.

Identifizierung eines neuen Bildes  $U$ , mithilfe des Mittelwerts  $M$  und den  $k$  Eigenfaces  $V_i$ :

Identifizierung eines neuen Bildes  $U$ , mithilfe des Mittelwerts  $M$  und den  $k$  Eigenfaces  $V_i$ :

- 1 Berechne Gewichte  $w_i = V_i^T (U - M)$ . Gewichte formen Vektor  $W = (w_1, \dots, w_k)$ .

# Identifizierung

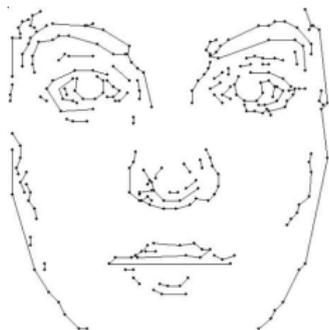
Identifizierung eines neuen Bildes  $U$ , mithilfe des Mittelwerts  $M$  und den  $k$  Eigenfaces  $V_i$ :

- 1 Berechne Gewichte  $w_i = V_i^T (U - M)$ . Gewichte formen Vektor  $W = (w_1, \dots, w_k)$ .
- 2 Berechne die euklidische Distanz zwischen den Gewichtsvektor  $W$  und gespeicherten Gewichtsvektor  $W_l$  für die  $l$ -te Gesichtsklasse.  
 $\epsilon_l = \|W - W_l\|$ .
- 3 Ist  $\min \epsilon_l < \Theta_\epsilon$ , dann gehört das Bild  $U$  zur Gesichtsklasse  $l$ , sonst wird  $U$  als unbekannt klassifiziert.

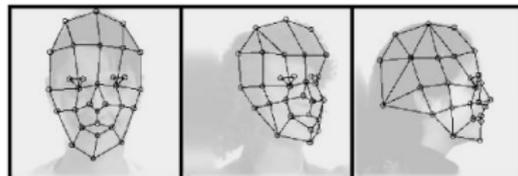


**Abbildung 1:** Eigenface-Rekonstruktion

# Weitere Methoden



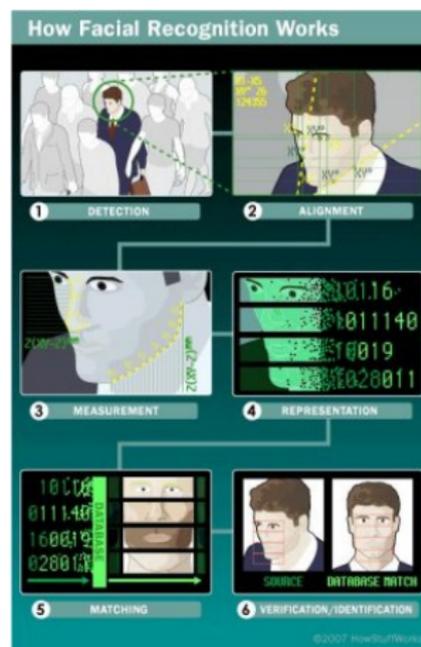
**Abbildung 2:** Line Edge Map



**Abbildung 3:** Elastic Bunch Graph

# 3D Gesichtserkennung

- 1 Erkennung
- 2 Auslegen
- 3 Messungen
- 4 Darstellung
- 5 Abgleich
- 6 Verifikation und Identifikation



**Abbildung 4:** Prozess einer Gesichtserkennung

1 Einleitung

2 Technik

**3 Anwendungen**

4 Vor- Nachteile

# Anwendungen

- Smartphone
- Gaming
- Social Media
- Sicherheit

# Anwendungen

- Smartphone
- Gaming
- Social Media
- Sicherheit



**Abbildung 5:** Ausschnitt aus Star Citizen

1 Einleitung

2 Technik

3 Anwendungen

**4 Vor- Nachteile**

# Vor- und Nachteile

## Vorteile:

- + Verbesserung der Sicherheit
- + hohe Genauigkeit
- + komplett automatisiert
- + leichte Integration in vorhandene Software
- + Einsatzbereich

# Vor- und Nachteile

## Nachteile:

- Voraussetzungen
- Verarbeitung und Speicher
- Sicherheit?
- Privatsphäre



Danke für Ihre Aufmerksamkeit!

- Biometrie - Gesichtserkennung
- Face Detection for Beginners
- Viola, Jones (2001), Rapid Object Detection using a Boosted Cascade of Simple Features
- Turk, Pentland (1991), Face Recognition using Eigenfaces
- 3D Gesichtserkennungsverfahren