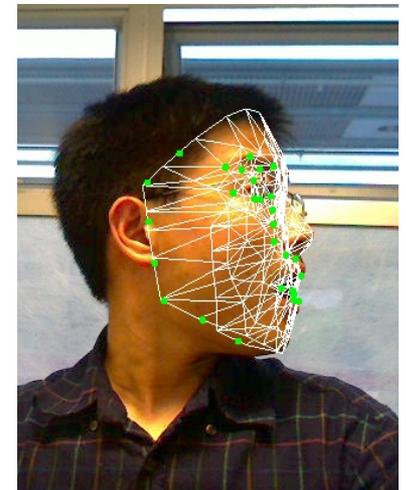
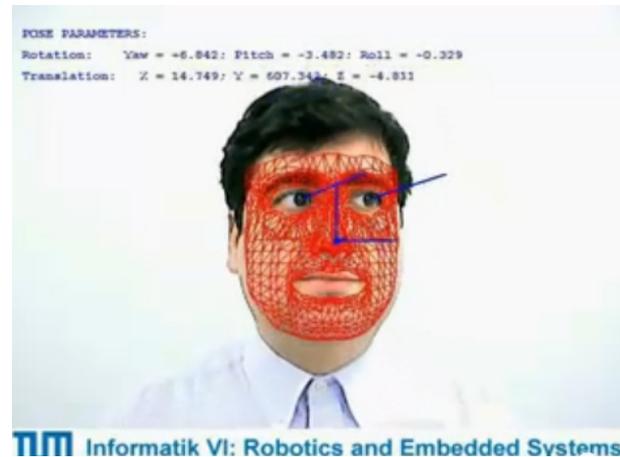
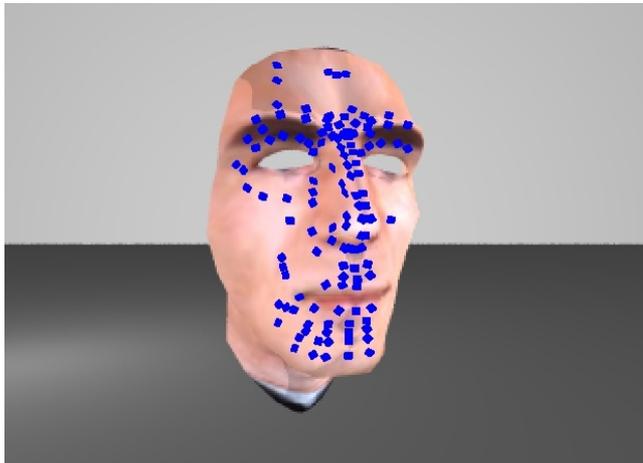


# Hauptseminar Computer Vision & Visual Tracking for Robotic Applications SS2012

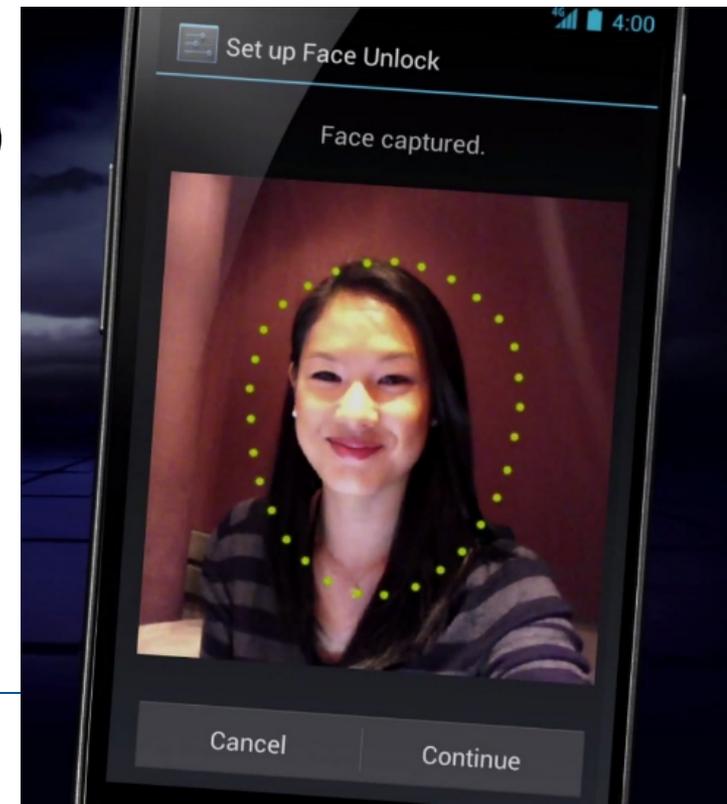
## Monocular 6D realtime face tracking



**Vortragende: Alina Roitberg**  
**Betreuer: Dr. Claus Lenz**

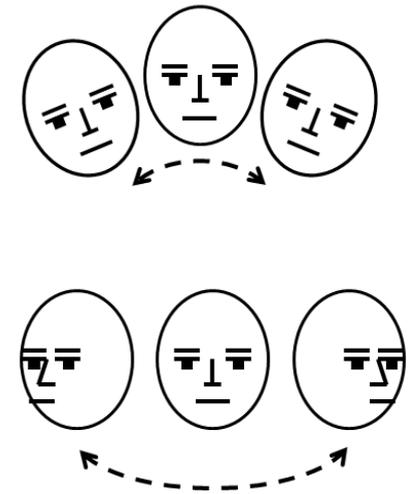
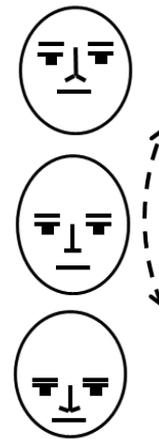
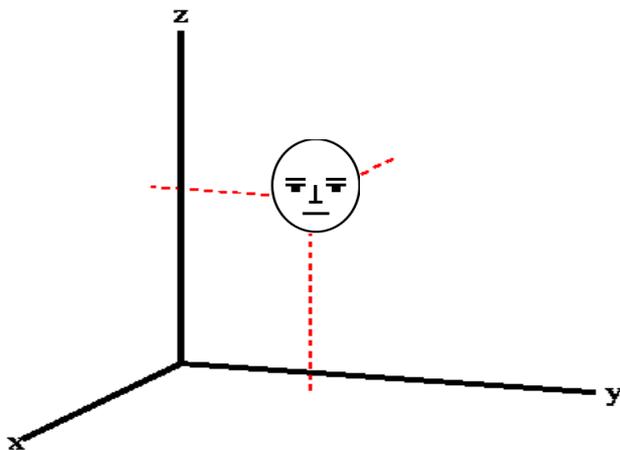
# Anwendungen

- Biometrie (Verifikation von Personen in Sicherheitssystemen)
- Digitale Kameras
- Video Conferencing
- Entertainment (Computerspiele etc.)



# Position

- Warum 6D?
  - Verschiedene Positionen des Kopfes
  - drei Raumdimensionen
  - drei Drehachsen des Kopfes



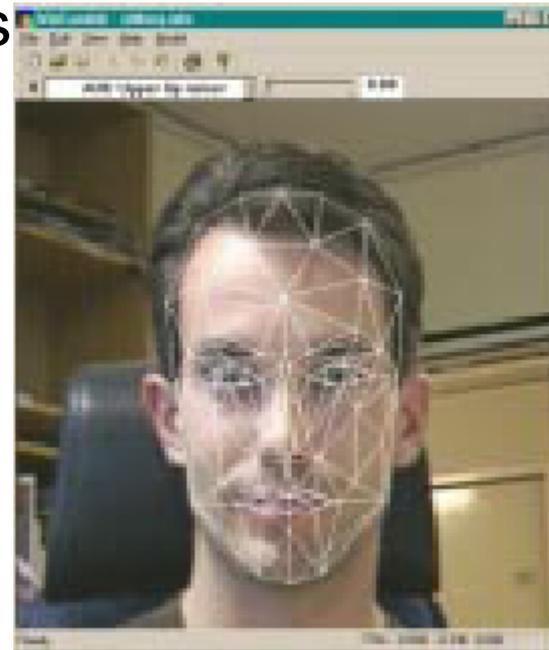
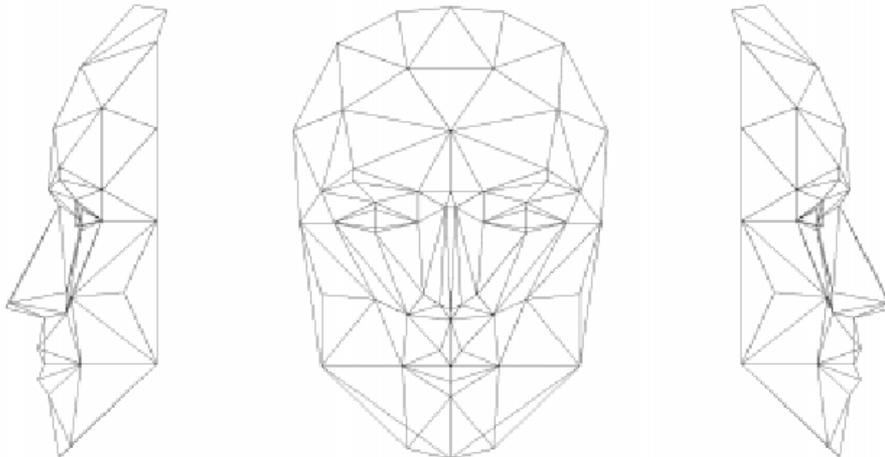
# Variationen zwischen verschiedenen Personen

- Sehr viele verschiedene Gesichter
- Variationen in Kopfform, Hautfarbe etc.



# Freiheitsgrade

- 6-D Position
- Abstand zwischen den Augen
- Abstand zwischen Mund und Nase
- Länge und breite des Kopfes
- Und noch viel viel mehr!



Shape Unit	
0	Head height
1	Eyebrows, vertical position
2	Eyes, vertical position
3	Eyes, width
4	Eyes, height
5	Eye separation distance
6	Cheeks z
7	Nose z-extension
8	Nose vertical position
9	Nose, pointing up
10	Mouth vertical position
11	Mouth width

# Mimik und andere Gesichtsveränderungen

- nahezu unendlich Anzahl an Gesichtsausdrücken
- Lachen, Gähnen, Blinken etc
- Accessoires (Sonnenbrille, Mütze etc.)
- Altern (Falten, Strukturveränderung)



# Variationen in Erscheinung und Mimik

Lösungsansätze:

- Charakterisierung der Gesichtern durch spezifische Merkmale (feature-based)
- Beschreibung eines Gesichts durch ein Durchschnittsgesicht + Freiheitsgrade (appearance-based)





# Methoden

- Feature-Based:
  - Besimme relevante Punkte
  - Diese Punkte sollen lokal, unverwechselbar, und robust gegenüber Veränderungen sein
  - Durchsuche das Bild nach diesen Merkmalen
  - Mapping von 3D Features auf eine 2D Referenz
  
- Appearance-Based

# Methoden

- Feature-Based
- Appearance-Based
  - Das gesamte Aussehen wird Trainiert
  - Training-Phase anhand einer Gesichterdatenbank
  - Bilden von einem Modell für das Gesicht (oft Durchschnittsgesicht)
  - Modellierung des Aussehens und der Textur
  - Bestimme die wichtigsten Parameter (meistens mit PCA)

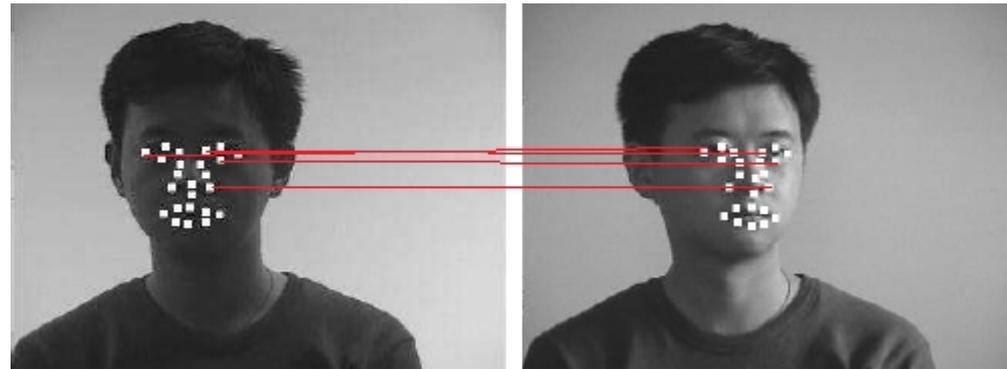
# Feature-Based-Verfahren

Was ist ein ein Merkmal?

- Kanten
- Ecken
- lokale Änderungen der Intensität in mind. eine Dimension
- Blobs
  - Regionen, die dunkler/heller sind als ihre Umgebung
- Keypoints
- Haar wavelets

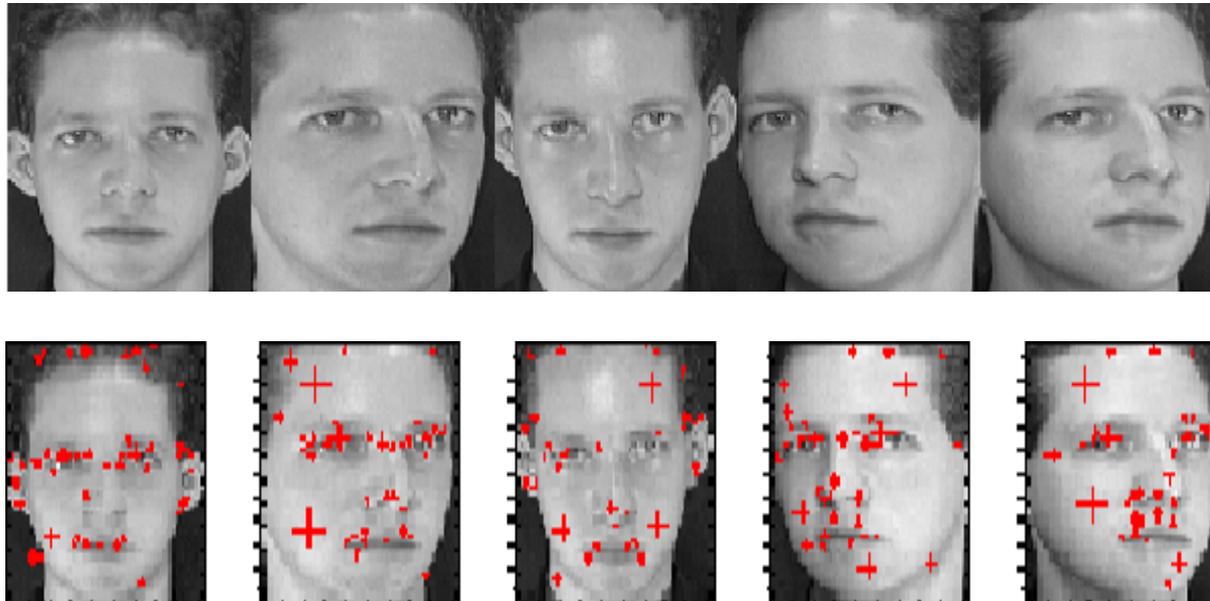
# Feature-Based-Verfahren: Keypoint-Matching

- Bestimmen von Keypoints
  - zufällig wählen (SIFT)
  - mit einer Training-Phase “antrainieren”
    - Keypoint-Deskriptor
- Bestimme welche Anordnung von Keypoints einem Gesicht entspricht
  - Classifier
- 3D Keypoints-Modell
- Bei einem neuen Frame:
  - Finde Keypoints
  - Ordne sie den Key-Points aus dem Modell zu
  - Berechne die Transformation



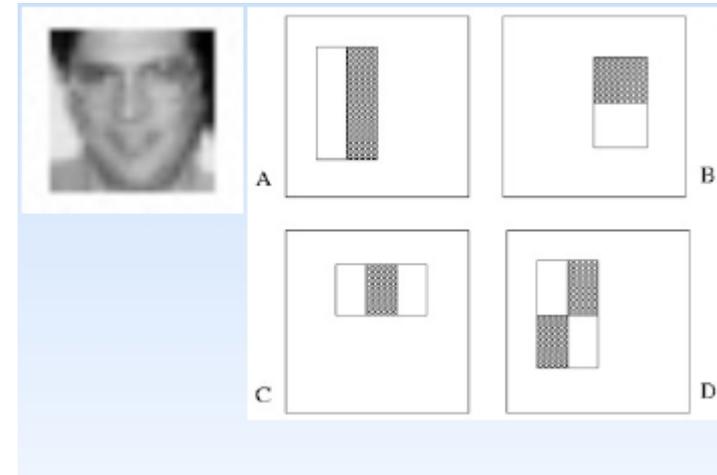
# Feature-Based-Verfahren: Keypoint-Matching

- Scale Invariant Feature Transform (SIFT)
- Die extrahierte Merkmale sind unveränderlich im Bezug auf Skalierung, Rotation, Translation und Filtern



# Feature-Based-Verfahren: Viola-Jones

- Basierend auf den sog. Haar-Like features
- Ein Merkmal besteht aus mehreren dunklen und hellen benachbarten Rechtecken
- Der Wert eines Merkmals ist bestimmt durch:

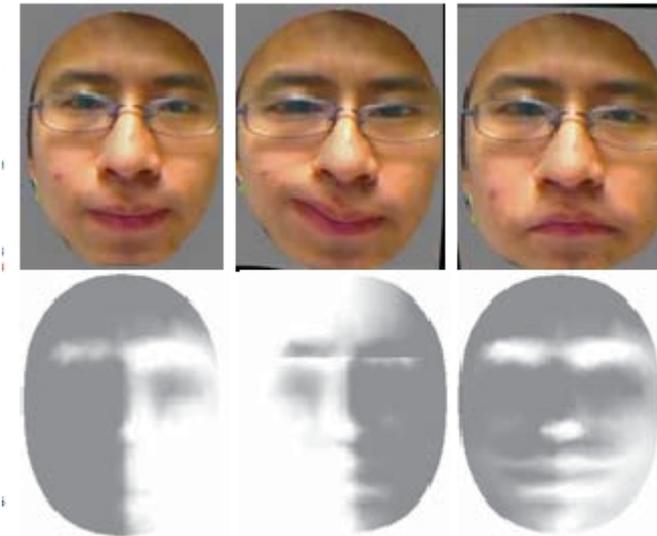


$$\sum \text{Pixel aus den hellen Rechtecken} - \sum \text{Pixel aus den dunklen Rechtecken}$$

- Trainings-Phase zum bestimmen von wichtigen Merkmalen eines Gesichts

# Beispiel: 3D Head Tracking via Invariant Keypoint Learning

- Hauptidee: “Lerne” welche Keypoints sich bei Transformationen nicht verändern
  - 3D Modell des Gesichts wird verschiedenen Störungen ausgesetzt
  - Das Gesicht wird immer wieder erstellt, nur unter anderen Sichtwinkeln, Verzerrungen etc.
  - Die Transformationen sind zufällig



# Beispiel: 3D Head Tracking via Invariant Keypoint Learning

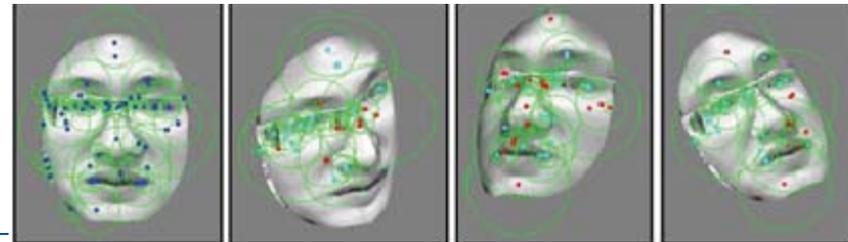
- Hauptidee: “Lerne” welche Keypoints sich bei Transformationen nicht verändern
  - invariante (unveränderliche) Keypoints werden bestimmt
  - Invarianz-Kriterium:  
 $u = u(k; M, A, L)$   
M ist eine Rotationsmatrix, A ist eine affine Transformation und L eine nichtlineare photometrische Transformation
  - Lockerung durch  
 $u - u(k; M, A, L) < \eta$ ,  $\eta$  ist der größte tolerierte lokale Fehler



# Beispiel: 3D Head

## Tracking via Invariant Keypoint Learning

- Hauptidee: “Lerne” welche Keypoints sich bei Transformationen nicht verändern
  - invariante (unveränderliche) Keypoints werden bestimmt
  - Invarianz-Kriterium:  
 $u = u(k; M, A, L)$   
M ist eine Rotationsmatrix, A ist eine affine Transformation und L eine nichtlineare photometrische Transformation
  - Lockerung durch  
 $u - u(k; M, A, L) < \eta$ ,  $\eta$  ist der größte tolerierte lokale Fehler



# Beispiel: 3D Head

## Tracking via Invariant Keypoint Learning

- Problem: sehr schwierig Merkmale zu finden, die sich bei starken Rotationen nicht verändern
- Lösung: Klassifizierung von Merkmalen
  - Nicht nach Merkmalen suchen, die sich bei Rotationen nicht verändern, sondern
  - Unterteilen aller Merkmale in Klassen
  - Deskriptoren einer Klasse repräsentieren denselben Merkmal bei verschiedenen Rotationen
  - In einem neuen Bild kann ein Deskriptor entweder
    - entweder als eine neue Variation einer existierenden Klasse
    - oder als die erste Variation einer alten Klasse zugeordnet werden

# Appearance-Based-Verfahren

- Lernen, welche Form- und Texturattribute für ein Gesicht typisch sind
- Modell für das gesamte Aussehen
- Training-Phase:
  - Analyse von einer großen Menge von Gesichtern
  - Erstellen eines Durchschnittsmodells
  - Bestimmen von wichtigen Parametern (PCA)
  - Anzahl von Parametern: Laufzeit gegen Genauigkeit

# Appearance-Based-Verfahren: Eigengesichter

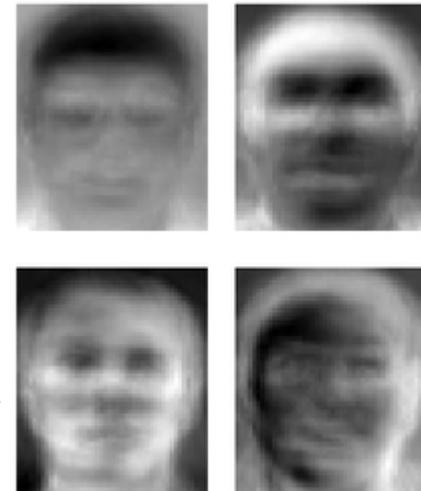
## Eigenfaces

Active Shape Model

Active Appearance Model

3D Morphable Model

- Eines der ersten Ansätze (Anfang 80er)
- Training:
  - Einlesen von Gesichtern als Vektoren
  - Anwenden von PCA
    - Erstelle eine Matrix bestehend aus Gesichtsvektoren
    - Ziehe von jeder Reihe das Durchschnittsgesicht ab
    - Bilde Kovarianzmatrix
    - Bestimme die Eigenwerte der Kovarianzmatrix



# Appearance-Based-Verfahren: Eigengesichter

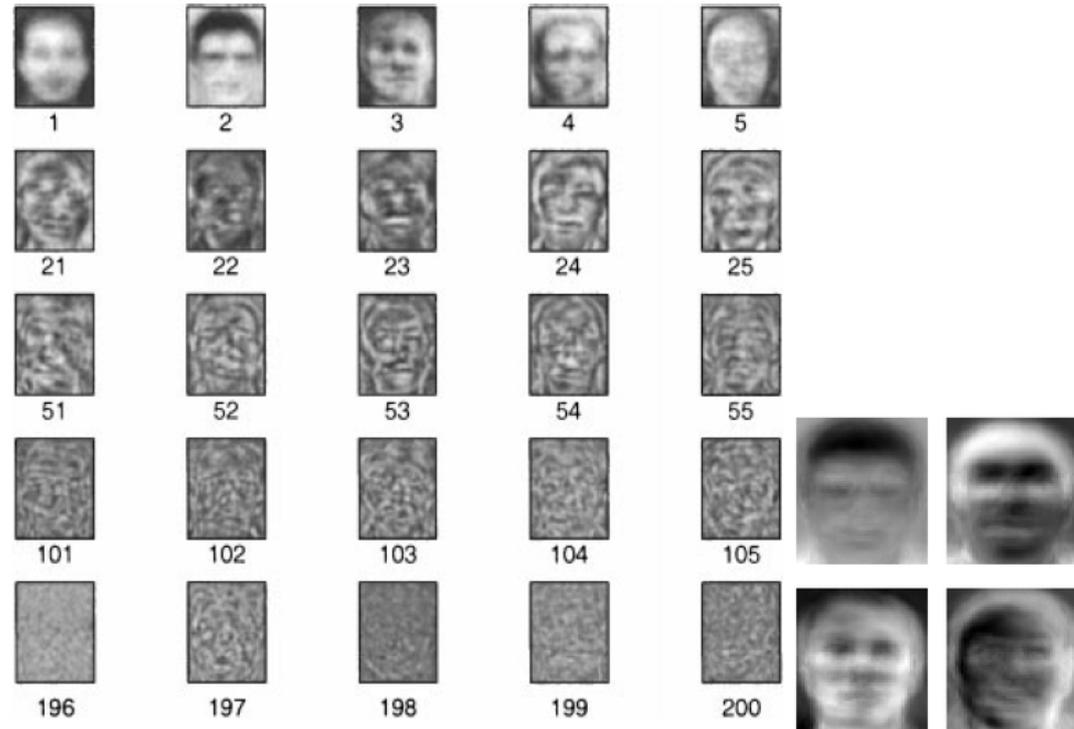
## Eigenfaces

- Eigenwerte der Kovarianzmatrix = Richtungen der größten Varianz
- 100x100 Bild: 10 000 Eigenvektoren → behalten von Eigenvektoren mit den größten Eigenwerten

Active Shape Model

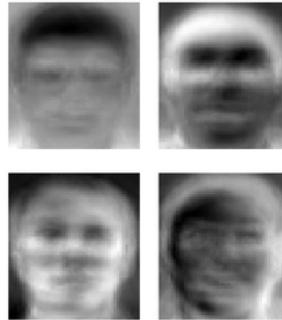
Active Appearance Model

3D Morphable Model



# Appearance-Based-Verfahren: Eigengesichter

## Eigenfaces



- Neue Gesichter werden als Linearkombination der Eigengesichter dargestellt
- Vorteile:
  - schnell
  - gute Genauigkeit bei gleicher Position und Beleuchtung
- Nachteile:
  - große Probleme bei Position- und Lichtveränderungen

Active Shape Model

Active Appearance Model

3D Morphable Model

# Appearance-Based-Verfahren: Active Shape Model (ASM)

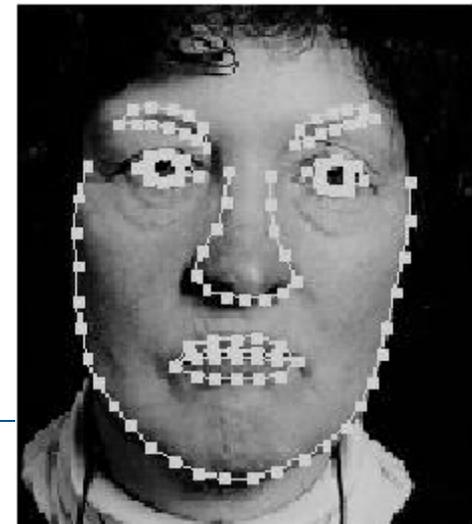
Eigenfaces

**Active Shape Model**

Active Appearance Model

3D Morphable Model

- Statistische Verteilung von Punkten in getracktem Objekt
- Training:
  - Labelling von spezifischen Punkten
  - Erstellung des Durchschnittsmodells
- Tracking:
  - Das Durchschnittsmodell wird angenommen
  - Die Position der Punkte wird aktualisiert



# Appearance-Based-Verfahren: Active Shape Model (ASM)

Eigenfaces

**Active Shape Model**

Active Appearance Model

3D Morphable Model

- Vorteile:
  - simpel
  - schnell
- Nachteile
  - nicht robust genug
  - Informationen über die Textur nicht im Betracht gezogen

# Appearance-Based-Verfahren: Active Appearance Model (AAM)

Eigenfaces

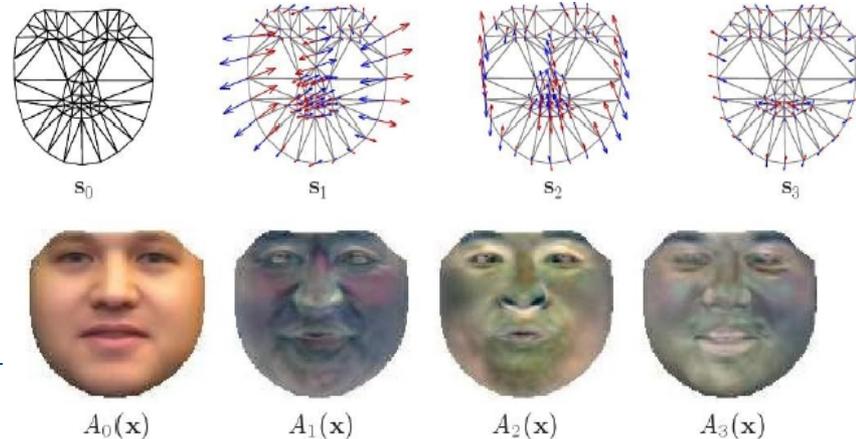
Active Shape Model

**Active Appearance Model**

3D Morphable Model

- Mitte der 90er von Tim Cootes und Chris Taylor entwickelt
- Weiterentwicklung von ASM
- Bilde ein Aussehens-Modell für die Form UND die Textur
- Modell eines Kopfes: Linearkombination der Form- und Texturparameter:

$$S = s_0 + \sum p_i * s_i \quad A = A_0 + \sum \lambda_i * A_i$$



# Appearance-Based-Verfahren: Active Appearance Model (AAM)

- Ablauf:
  - Generiere AAM (Offline Training-Phase)
  - Anpassen des Modells an das eingegebene Bild (Fitting)
  - Auswerten der Modell-Parameter

Eigenfaces

Active Shape Model

**Active Appearance Model**

3D Morphable Model



# Appearance-Based-Verfahren: Active Appearance Model (AAM)

Eigenfaces

Active Shape Model

**Active Appearance Model**

3D Morphable Model

- Vorteile:
  - schnell
  - robust
- Nachteile
  - gutes Anfangsmodell erforderlich



$[-60^\circ, -30^\circ]$



$[-30^\circ, 30^\circ]$



$[30^\circ, 60^\circ]$

# Appearance-Based-Verfahren: 3D Morphable Models (3D-MM)

Eigenfaces

Active Shape Model

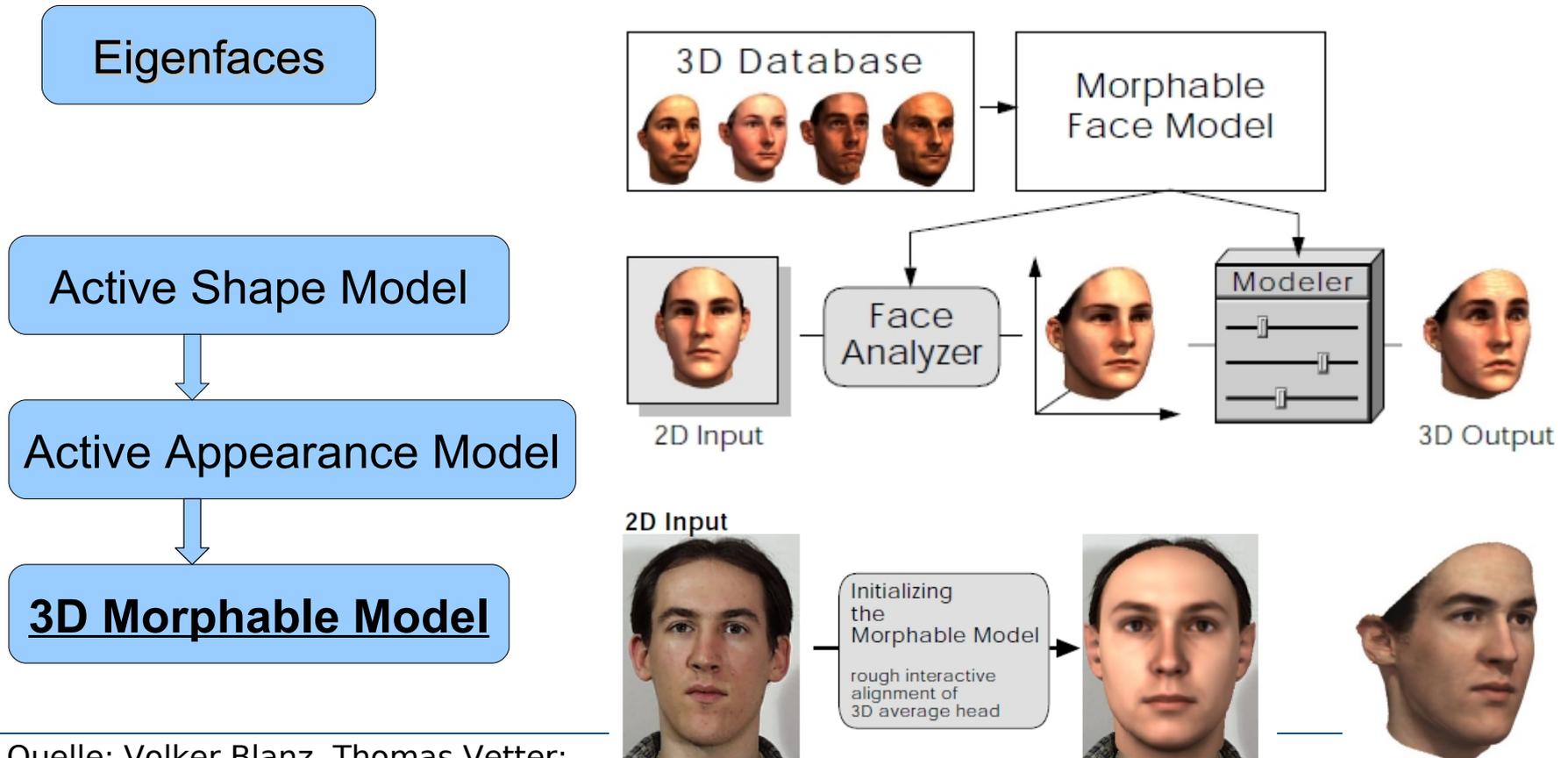
Active Appearance Model

**3D Morphable Model**

- Dreidimensionales Modell des Gesichts
- Jeder Punkt des Modells hat 3D Koordinaten  
→ Besser im Bezug auf starke Rotation



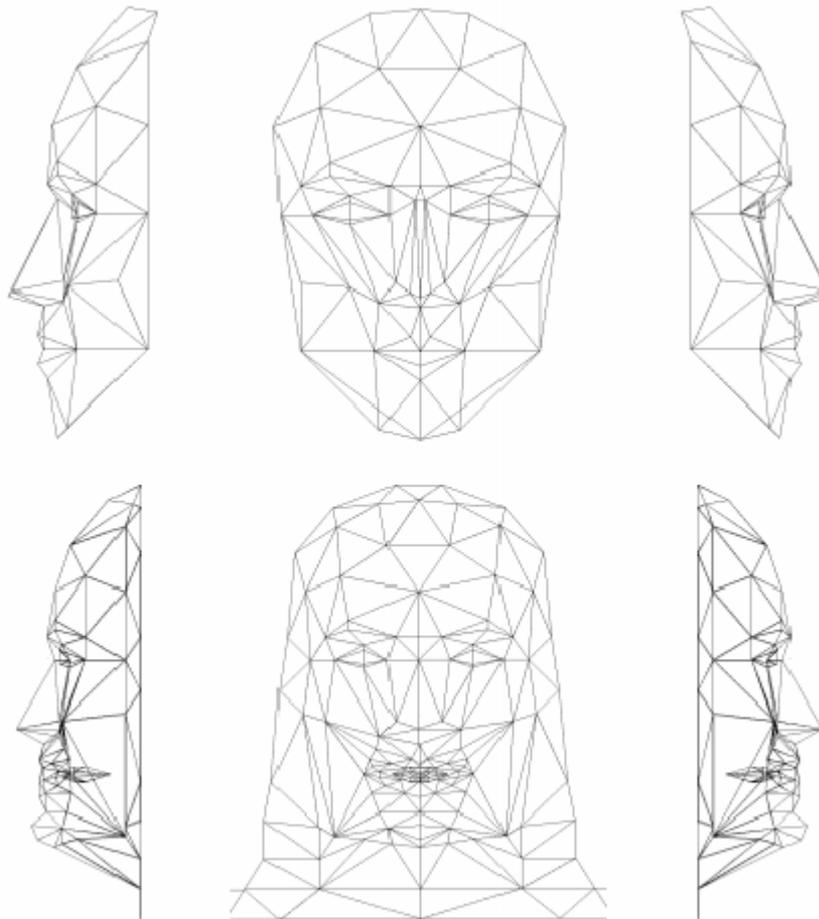
# Appearance-Based-Verfahren: 3D Morphable Models (3D-MM)



# Beispiel: Extended 2D+3D AAMs based Face Tracking

- Training: Candide-3 (parametrisiertes Gesichts-Modell)
- Ein Gesicht wird beschrieben durch die Summe von:
  - Standard-Modell
  - Form-Vektor
  - Appearance-Vektor
- Ablauf:
  - Initialisiere ein Start-Modell
  - Aktualisiere die Parameterwerte um die Fehlerfunktion zu minimieren
  - Inverse Compositional Parameter Update Technik

# Beispiel: Extended 2D+3D AAMs based Face Tracking



Shape Unit		Comment
0	Head height	Does not influence eyes, mouth, ...
1	Eyebrows, vertical position	
2	Eyes, vertical position	
3	Eyes, width	
4	Eyes, height	
5	Eye separation distance	
6	Cheeks z	z-extension of the cheek bone.
7	Nose z-extension	z-extension of the nose.
8	Nose vertical position	
9	Nose, pointing up	Vertical position of nose tip.
10	Mouth vertical position	
11	Mouth width	

- Training: Candide-3 (parametrisiertes Gesichts-Modell)
- Ein Gesicht wird beschrieben durch die Summe von:
  - Standard-Modell
  - Form-Vektor
  - Appearance-Vektor
- Ablauf:
  - Initialisiere ein Start-Modell
  - Aktualisiere die Parameterwerte um die Fehlerfunktion zu minimieren
  - Inverse Compositional Parameter Update Technik

# Beispiel: Extended 2D+3D AAMs based Face Tracking

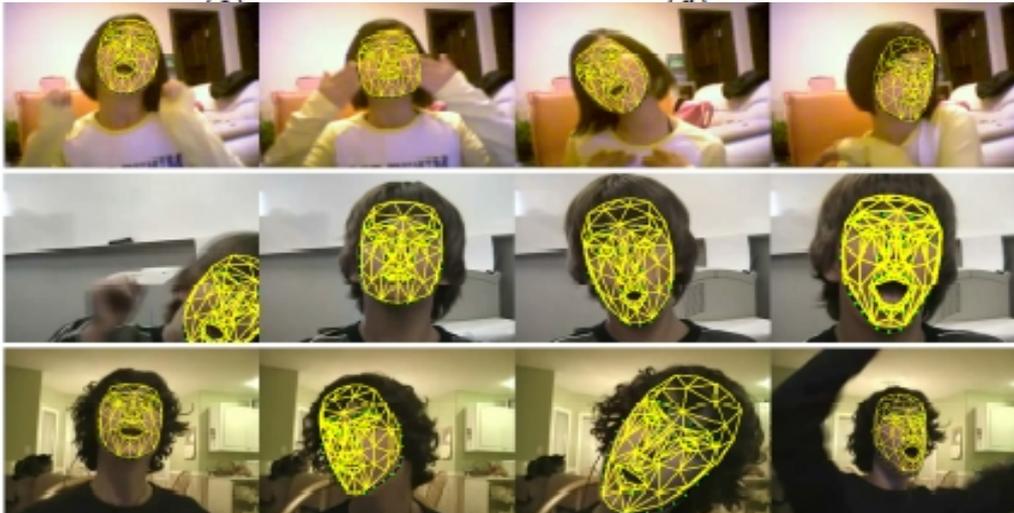
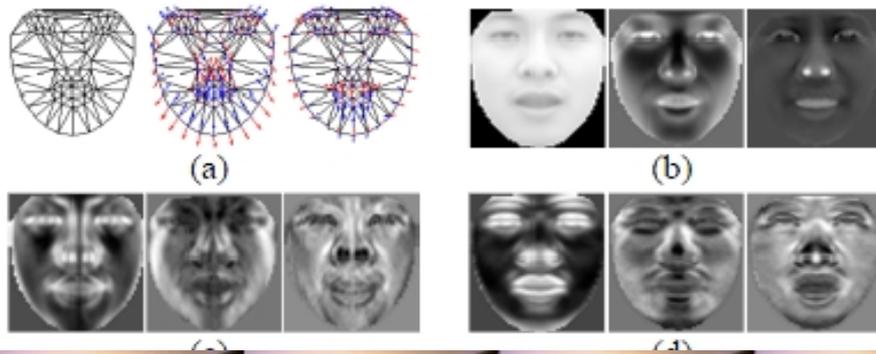
- Hauptidee: beschreibe ein 3D-Modell mithilfe eines 2-D AAM
- Möglich durch zusätzliche Parameter und
- Einschränkungen des Modells (nur die Parameterwerte sind zulässig, die ein dreidimensionales Modell darstellen)
- Recovery: Übersteigt der Wert der Fehlerfunktion einen bestimmten Wert, wird der Zustand verworfen und neu initialisiert

→ Robuster gegenüber starker Rotation

# Beispiel: Extended 2D+3D AAMs based Face Tracking

- Hauptidee: beschreibe ein 3D-Modell mithilfe eines 2-D AAM
- Möglich durch zusätzliche Parameter und
- Einschränkungen des Modells (nur die Parameterwerte sind zulässig, die ein dreidimensionales Modell darstellen)
- Recovery: Übersteigt der Wert der Fehlerfunktion einen bestimmten Wert, wird der Zustand verworfen und neu initialisiert

# Beispiel: Extended 2D+3D AAMs based Face Tracking



# Vergleich von Verfahren

- Feature-Based-Verfahren
  - +Kein Training erforderlich
  - +Manche Verfahren (wie SIFT) sind stabil und robust
  - oft nicht schnell genug für die Echtzeitanwendungen
  - falsch erkannte Keypoints oder zu wenige Keypoint führen zur falschen Positionsbestimmung
- Appearance-Based-Verfahren
  - +stabil und genau
  - +gute Laufzeit
  - sehr Abhängig von dem Anfangsmodell
  - gute Training-Daten erforderlich

Danke für Ihre Aufmerksamkeit!

Fragen?