

Gesichtsidentifikation mit dem Erkennungssystem FaceVACS SDK der Firma Cognitec

Zwischenpräsentation

Gliederung

- Aufgabenstellung und Randbedingungen
- Ablauf und Verfahren der Gesichtsidentifikation
- Anwendungen für die Gesichtserkennung bzw. -identifikation
- Vorstellung des FaceVACS SDK
- Aktueller Stand des Projektes
- Further Work

Aufgabenstellung und Randbedingungen

Entwicklung einer Applikation (SCITOS) zur Gesichtsidendifikation auf der Basis eines 2D-Bildes unter Verwendung der FaceVACS SDK der Firma Cognitec

- Überblick Stand der Technik
- Einarbeitung in das FaceVACS SDK
- 2D Kamera Inbetriebnahme und Konfiguration unter Linux

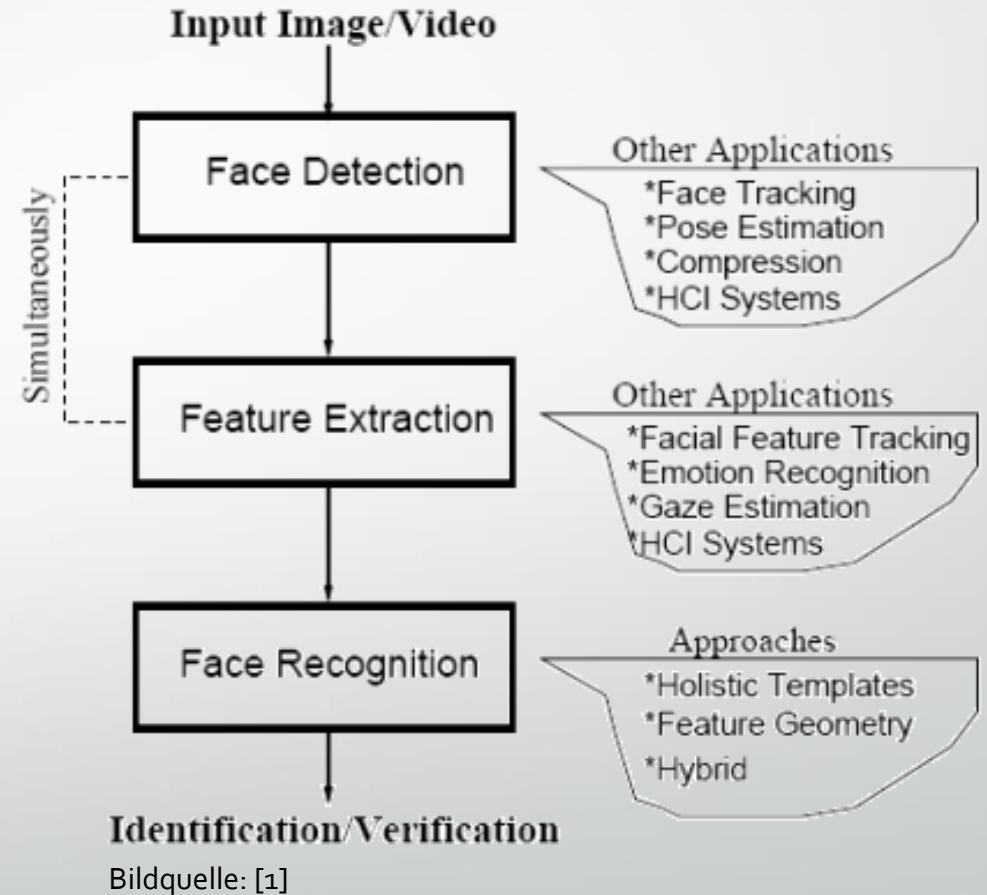
Gliederung

- Aufgabenstellung und Randbedingungen
- **Ablauf und Verfahren der Gesichtsidendifikation**
- Anwendungen für die Gesichtserkennung bzw. -identifikation
- Vorstellung des FaceVACS SDK
- Aktueller Stand des Projektes
- Further Work

Ablauf einer Gesichtsidentifikation

Aufteilung in zwei Phasen:

1. Erstellung Datenbank, mit bekannten Gesichtern & Identitätszuordnung
2. Überprüfung ob in Kamerabild bekannte Gesichter vorhanden
 - a) Finden des Gesichts im Bild
 - b) Extrahieren der Merkmale
 - c) Ermittlung der Identität



Verfahren zur Gesichtsidentifikation

- **Merkmalsbasierte Verfahren**

Merkmalsvektor aus z.B. 35 Merkmalen wie Distanzen, Flächen und Winkel. Klassifizierung durch Pattern-Matching mit euklidischer Distanz (z.B. Elastic Bunch Graph Matching)[1].

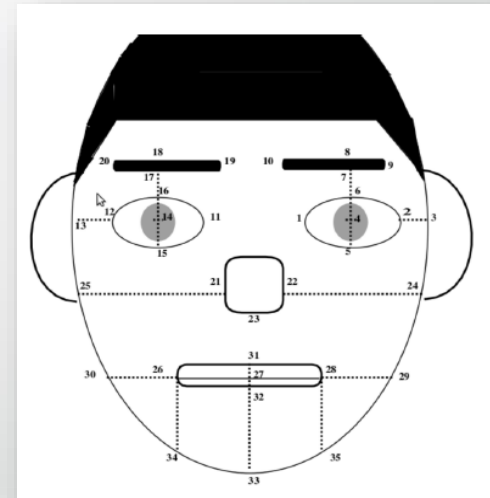
- **Holistische Verfahren**

Erkennung von Gesichtern als Untermenge der Erkennung von Objekten. Holistische Ansätze „wissen“ nichts über das Gebiet auf das sie angewendet werden. Es entstehen hochdimensionale Vektorräume die erst in niedrigdimensionale Unterräume reduziert werden müssen [1].

- **Hybride Ansätze**

- **Sonstige Ansätze**

Betrachtung von Gesichtern als Texturen von 3D-Körpern. Höheninformationen als eindeutige Identifizierungsmerkmale[1].



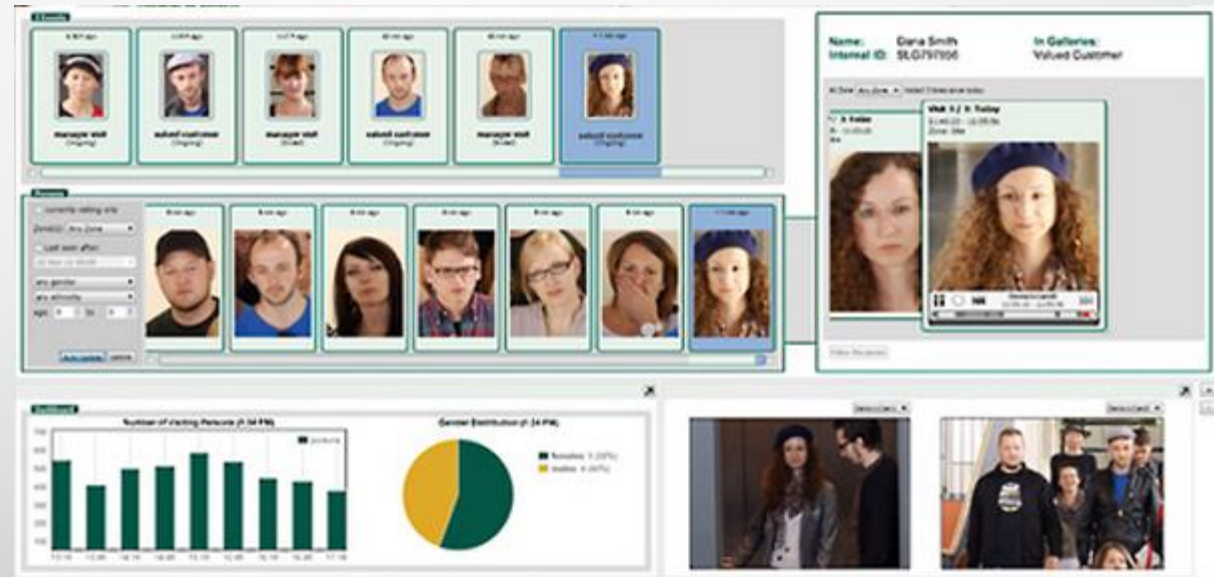
Bildquelle: [2]

Gliederung

- Aufgabenstellung und Randbedingungen
- Ablauf und Verfahren der Gesichtsidentifikation
- **Anwendungen für die Gesichtserkennung bzw. -identifikation**
- Vorstellung des FaceVACS SDK
- Aktueller Stand des Projektes
- Further Work

Anwendungen für die Gesichtserkennung/ -identifikation

- Zutrittskontrolle / Ausweiskontrolle
- Polizei / Militär
- Werbung
- Fotosammlungen
- Service-Roboter Systeme

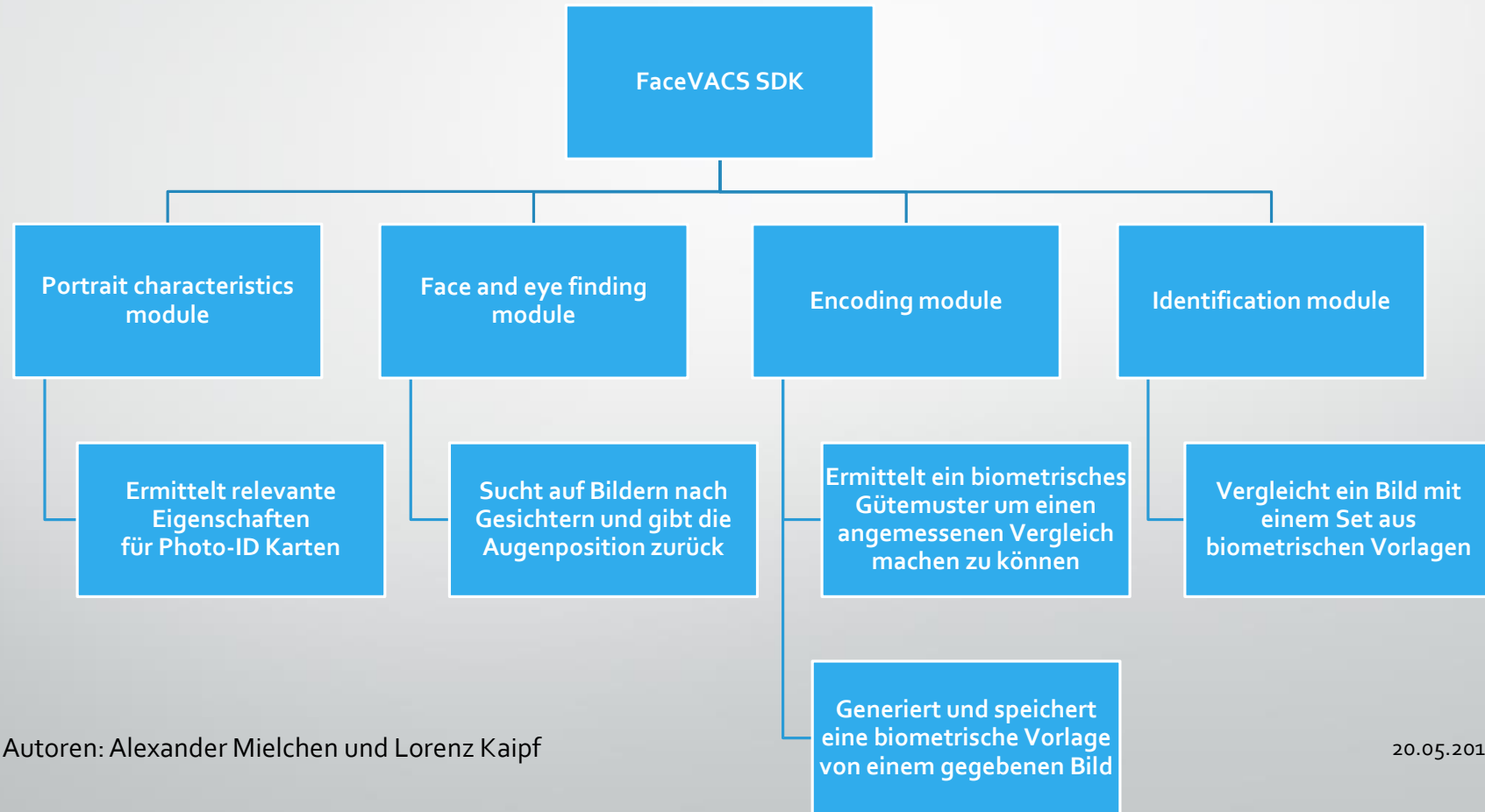


Bildquelle: Cognitec

Gliederung

- Aufgabenstellung und Randbedingungen
- Ablauf und Verfahren der Gesichtsidentifikation
- Anwendungen für die Gesichtserkennung bzw. -identifikation
- **Vorstellung des FaceVACS SDK**
- Aktueller Stand des Projektes
- Further Work

Vorstellung des FaceVACS SDK (1)



Vorstellung des FaceVACS SDK (2)

Beispiel: Face and eye finding module

```
cd <application path>  
./facefind -cfg <configfile path>/frsdk.cfg -imgs <imagefile path>/*.jpg
```

Gliederung

- Aufgabenstellung und Randbedingungen
- Ablauf und Verfahren der Gesichtsidentifikation
- Anwendungen für die Gesichtserkennung bzw. -identifikation
- Vorstellung des FaceVACS SDK
- **Aktueller Stand des Projektes**
- Further Work

Aktueller Stand

- Einarbeitung / Sichtung FaceVACS SDK und dessen Möglichkeiten ✓
 - Ausführung der Beispiele durch Bash-File und Ausgabe der Ergebnisse auf Terminal möglich, ABER kein direkter Zugriff auf die einzelnen Ergebnis-Parameter
BESSER Eclipse als Entwicklungsumgebung einrichten
- Sichtung Vorgängerprojekt SCITOS „Can I help you?“ ✓
 - Aufruf der FaceFind-Funktion auch nur aus Bash-File heraus!
- Inbetriebnahme / Integration der uEye Kamera (Tec.Rätsch6) -> bessere Bildqualität ✓
 - Installation von OpenCV und IDS-Bibliothek inkl. uEyeDemo (Kameraviewer)
 - Test der Kamera
- Literaturrecherche ✓

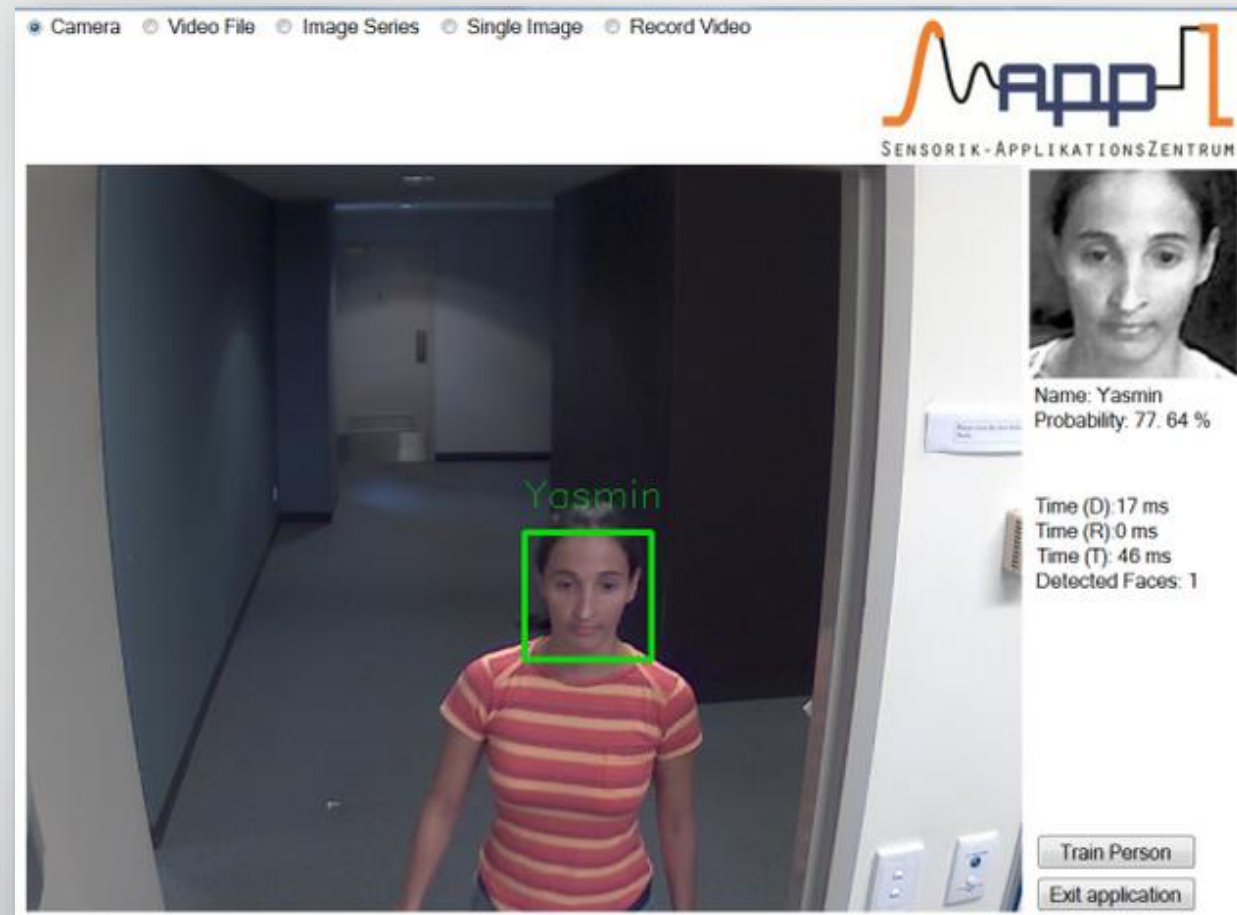
Gliederung

- Aufgabenstellung und Randbedingungen
- Ablauf und Verfahren der Gesichtsidendifikation
- Anwendungen für die Gesichtserkennung bzw. -identifikation
- Vorstellung des FaceVACS SDK
- Aktueller Stand des Projektes
- **Further Work**

Further work

- Eclipse als Entwicklungsumgebung einrichten
 - Kameraansteuerung über C++ Datei ermöglichen (aktuell über Python Skript realisiert)
- Einfache Test Applikation entwickeln (Fahrbefehle aufgrund von Rückgabeparameter des FaceVACS möglich?)
- Galerie mit Lerndaten für zu erkennende Personen anlegen, z.B. 3-5 Personen
 - Anzahl, Aufnahmepositionen, Auflösung usw. der Bilder definieren
 - Erzeugung einer Galerie manuell oder durch eine Applikation
- Applikation die ein live Bild mit allen vorhandenen Galerien vergleicht und eine identifizierte Person mit ihrem Namen anspricht
 - Funktion zum Vergleich eines live Bildes mit allen vorhandenen Galerien implementieren
 - Funktion zur sprachlichen Interaktion mit einer identifizierten Person implementieren
- Test und Validierung der entwickelten Software


Denkbare Wunschlösung



Bildquelle: [3]

Quellenangaben

- [1] http://www.medien.ifi.lmu.de/fileadmin/mimuc/hs_ws0506/papers/Automatische_Gesichtserkennung.pdf
- [2] http://patrec.cs.tu-dortmund.de/pubs/theses/ba_pavelko.pdf
- [3] <http://sappz.de/index.php/en/projekte/bildverarbeitung>



Vielen Dank für Eure Aufmerksamkeit!

Fragen?