2-Factor biometric authentication with body shape and voice

Bachelorarbeit
Malte Paskuda

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Informatik

Fachgebiet Telekooperation
Prof. Dr. Max Mühlhäuser

2-Factor biometric authentication with body shape and voice
Bachelorarbeit
Eingereicht von Malte Paskuda
Tag der Einreichung: 15. September 2011

Gutachter: Prof. Dr. Max Mühlhäuser
Betreuer: Dr. Dirk Schnelle-Walka
Externer Betreuer: Jürgen Coppenhaguen-Bormuth, Deutsche Telekom Laboratories

Technische Universität Darmstadt
Fachbereich Informatik

Fachgebiet Telekooperation (TK)
Prof. Dr. Max Mühlhäuser

Ehrenwörtliche Erklärung

Hiermit versichere ich, die vorliegende Bachelorarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in dieser oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 15. September 2011 Malte Paskuda

Inhaltsverzeichnis

1 Einleitung

Um sich bei modernen Geräten anzumelden, ist häufig ein recht großer Aufwand nötig. Üblicherweise wählt man einen Benutzernamen und gibt das zugehörige Passwort ein, was ein sich wiederholender und aktiver Vorgang ist. Alternativ könnte das Gerät den Menschen erkennen und automatisch anmelden. Dafür gibt es verschiedene Möglichkeiten, beispielsweise die Sprecher- und die Körperformerkennung. Beiden gemein ist, dass sie im Grunde mit modernen Geräten nutzbar sind, sei es durch vorhandene Zusatzhardware wie Microsofts Kinect oder der integrierten Kamera samt Mikrofon in modernen Laptops, Tablets und Handys. Jedoch reicht eines der Erkennungsverfahren alleine nicht aus, um zuverlässig Menschen zu identifizieren und so als Ersatz für passwortbasierte Anmeldeverfahren zu dienen. Normalerweise werden solche biometrischen Verfahren in Kombination mit eindeutigen Verfahren wie einer Passwortsicherung als Teil eines Multi-Faktor-Anmeldesystems genutzt [7].

Ziel der Arbeit ist es, Möglichkeiten zu untersuchen, wie mit dem derzeitigen Stand der Entwicklung in diesen Bereichen ein möglichst einfacher und automatischer Anmeldevorgang durchgeführt werden kann, ohne dass Passwörter eingegeben werden müssten.

1.1 Szenario

Durch den Ansatz dieser Arbeit könnte das Anmelden in einem typischen Szenario wie dem Fernseher im Wohnzimmer angenehmer werden. Bei diesem wäre ein Anmeldevorgang sinnvoll, z.B. um Inhalte für Erwachsene auch nur den Erwachsenen in der Familie zugänglich zu machen. Dafür wird momentan entweder kein Schutz umgesetzt, sondern eine Sendezeitbegrenzung zur Pflicht gemacht, oder bestimmte Sender sind durch eine Pin gesperrt. Gleichzeitig könnte an einem solchen Fernseher eine Xbox mit einer Kinect angeschlossen sein, da viele Haushalte eine solche Spielekonsole besitzen. Es bestünde also die Möglichkeit, über ein oder mehrere Erkennungsverfahren den Menschen vor dem Fernseher zu erkennen und so zu authentifizieren. Dieser müsste sich nicht die richtige Pin merken und trotzdem wäre der Zugang zu dem Gerät geschützt.

1.2 Aufbau

Die Arbeit ist folgendermaßen aufgebaut: In Kapitel 2 werden mögliche Anmeldeverfahren klassifiziert, die technischen Grundlagen der verwendeten Anmeldeverfahren erklärt und die genutzte Hardware beschrieben. In Kapitel 3 wird das Konzept des Anmeldesystems entwickelt. Kapitel 4 beschreibt dann die praktische Umsetzung des Konzepts. Abschließend wird in Kapitel 5 in einer kurzen Evaluation die erreichte Erkennungsrate untersucht.

2 Grundlagen

Anmeldung wird benutzt, um durch Identifikation eines Nutzers diesem Nutzer Rechte zu gewähren. Ohne Identifikation könnten auch Rechte gewährt werden, wenn das Anmeldeverfahren sicher feststellt, dass der Nutzer erwachsen ist.

Durch eine Anmeldung an einem System können Teile des Systems Nutzern zugeordnet werden. Das persönliche Verzeichnis beispielsweise enthält persönliche Daten, die nur einem bestimmten Nutzer gezeigt werden. Ein Spiel speichert unter einem bestimmten Profil die Erfolge eines bestimmten Nutzers. Nicht jedes System braucht eine Anmeldung, gerade bei Spielen kann so etwas unnötig sein. Aber wird eine Anmeldung gebraucht, bleibt immer noch die Frage, welche Form diese Anmeldung hat. Reicht die Kenntnis einer Pin, oder ist ein Fingerabdruck nötig? Banken und auch Amazon verknüpfen mehrere Arten von Anmeldungsverfahren, um nach einer Grundanmeldung kritische Systembereiche gesondert zu sichern. Aber es sind aktive Anmeldungen (oder bei Amazon deren Überbleibsel in From eines Cookies), bei denen eine Aktion des Nutzers nötig ist. Eine passive Anmeldung ist eher eine Zukunftsvision, bei der das Raumschiff den Captain erkennt und nur er es steuern kann. Doch durch biometrische Erkennungsverfahren könnte man durchaus schon in der heutigen Zeit versuche, Systeme Menschen erkennen zu lassen.

Die beiden im Folgenden beschriebenen Erkennungsverfahren, Sprecher- und Körperformerkennung, sind gleichzeitig die Verfahren, die im beschriebenen Szenario mit einer Kinect als Sensor einfach möglich sind. Eine Alternative zu beiden Verfahren ist die Gesichtserkennung. Bei ihr ist jedoch fraglich, ob im beschriebenen Szenario die nötige Annäherung an die Kamera nicht zu störend wäre. Wenn die Kamera auf die Entfernung eine Gesichtserkennung durchführen könnte, wäre sie eine Alternative, die in das Konzept der störungsfreien autoamtischen Anmeldung passen würde.

2.1 Kategorien von Anmeldeverfahren

Ein Anmeldeverfahren ist immer eine Verifikation der eigenen Identität vor dem System. Dafür gibt es verschiedene Möglichkeiten.

Die klassische Sichtweise unterteilt diese Möglichkeiten in drei Bereiche [10]:

- Anmeldung durch Wissen (Passwort).
- Anmeldung durch Besitz (USB-Stick).
- Anmeldung durch Sein (Biometrie).

Eine alternative Sichtweise ist, dass alle diese Bereiche eigentlich der Besitz (im Sinn der freien Verfügung über) von etwas sind. Die Anmeldung geschieht dann durch Besitz, normalerweise durch den Besitz von Wissen (dem Passwort), oder durch den Besitz eines Dinges oder den Besitz einer Eigenschaft. In dieser Arbeit wird Anmeldung nach diesem Prinzip betrachtet. Dieses Prinzip ist unterteilbar in Unterkategorien:

2.1.1 Eindeutige Verfahren

Das klassische Anmelden per Passwort ist ein eindeutiges Verfahren. Das gilt genauso für klassische textbasierte Passwörter wie für grafische Passwörter (z.B. Anmeldung bei Smartphones). Das Passwort stimmt oder stimmt nicht. Genauso ist es beim Anmelden durch Besitz von Hardware, z.B. eines USB-Sticks. Die Anmeldung scheitert auch, wenn nur ein Teil des Anmeldewerkzeugs vorhanden ist, z.B. nur die ersten Stellen eines Passwortes, und unterscheidet sich dadurch von dem etwaigen stufigen Anmeldeverfahren einer Bank [10].

2.1.2 Probabilistische Verfahren

Bei probabilistischen Verfahren ist zwar auch der Besitz ausschlaggebend, z.B. der Besitz einer bestimmten Stimme oder eines bestimmten Gesichts. Das System kann aber nicht simpel feststellen, ob die Verifikation erfolgt ist, sondern bestimmt die Wahrscheinlichkeit einer Übereinstimmung mit der gesuchten oder behaupteten Identität.

Bei probabilistischen Verfahren kann nochmal ein Unterschied zwischen Verifikation und Identifikation gemacht werden. Identifikation meint dann das Bestimmen einer Identität aus einer gegebenen Nutzermenge, während bei Verifikation nur die Übereinstimmung mit jeweils einer Identität geprüft wird, aber zusätzlich unbekannte Nutzer (Imposter) als solche erkannt werden müssen. Die Identifikation bezieht sich also auf eine geschlossene Menge von Nutzern (closed-set), die Verifikation auf eine offene Menge mit unbekannten Nutzern (open-set) [17].

Probabilistische Verfahren haben die Gefahr, dass durch sie Nutzer falsch erkannt werden und so die falschen Rechte bekommen. Kritische Systeme können von ihnen also nur gesichert werden, wenn die Erkennungsrate sehr hoch ist. Aber sie haben den Vorteil, dass durch sie eine passive Erkennung, also ohne Anstrengung seitens des Anzumeldenden, durchgeführt werden könnte. Das wäre bei eindeutigen Verfahren nur durch eine mitgeführte Identifikationsquelle wie einem Sender möglich.

2.1.3 Reine Vermutungen / Externe Datenquelle

Bei Konzepten wie der Prüfung des Terminkalenders der vermeintlichen Person, also ob laut diesem die Person überhaupt anwesend sein kann, liegt nur eine externe Datenquelle vor. Es ist also völlig unerheblich, ob gerade die reale Person versucht sich anzumelden oder eine falsche Identität angegeben wird, da kein konkreter Test gemacht wird. In gewissem Sinne ist dies ein Unterpunkt der probabilistischen Verfahren (Ist es laut Terminplan wahrscheinlich, dass dies die richtige Person ist?), jedoch - da ohne konkreten Test - eine schwächere Absicherung.

In dieser Arbeit werden probabilistische Verfahren genutzt, mit denen bei passender Implementierung eine passive Anmeldung durchgeführt werden könnte.

2.2 Grundlagen der Sprechererkennung

Sprechererkennung basiert auf der Grundannahme, dass Sprache nicht nur eine Nachricht übermittelt, sondern zusätzlich enthält sie inhärent Informationen über den Körperbau, die linguistische Erfahrung und sogar den Geisteszustand des Sprechers. Da diese Charakteristika auf verschiedenen Ebenen gefunden werden können, inklusive dem Spektrum der produzierten Töne, kann eine textunabhängige Sprechererkennung ebenfalls auf verschiedenen Ebenen versucht werden [8].

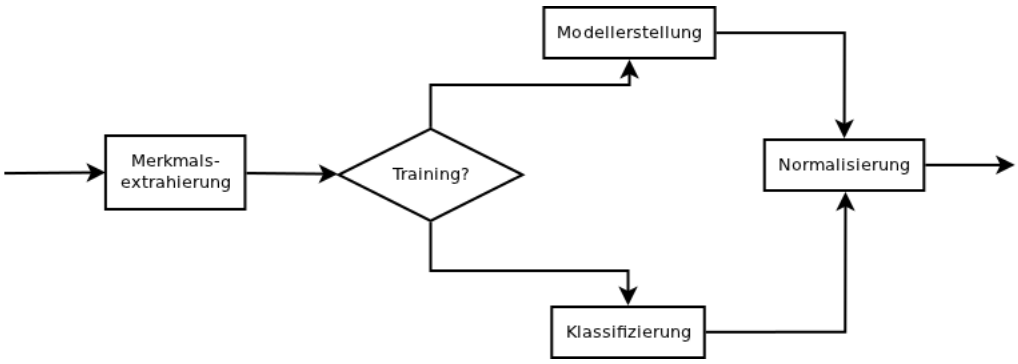


Abbildung 2.1: Vorgehensweise Sprechererkennung, basierend auf [17].

Obwohl es unterschiedliche Vorgehensweisen gibt, ist das grobe Schema der Sprechererkennung universell [17]. Die Merkmale der Sprache werden extrahiert, in der Trainingsphase zusätzlich trainiert, dann klassifiziert, normalisiert und schließlich wird über sie ein Sprecher bestimmt (siehe Abb. 2.1). Im Folgenden werden die Grundlagen der Sprechererkennung beschrieben, die im Rahmen der Arbeit benötigt wurden, da sie genutzt oder in Betracht gezogen wurden.

2.2.1 Merkmale

Welche sprecherabhängigen Merkmale der Sprache gewählt werden, entscheidet über alle folgenden Schritte. Grundsätzlich könnten hier auch High-Level-Informationen wie die Wortwahl gewählt werden [15]. Verbreiteter jedoch sind spektrale Ansätze wie die folgenden [17]. Um diese zu erklären, müssen ein paar Grundbegriffe der Signalverarbeitung genutzt werden:

Sprache kann als Wellenform dargestellt werden, als Zuordnung von Amplitude zu Zeit. Mit einer Fouriertransformation kann aus dieser Wellenform das Spektrum errechnet werden, in diesem Fall die Zuordnung der Frequenz zur Lautstärke (siehe Abb. 2.2).

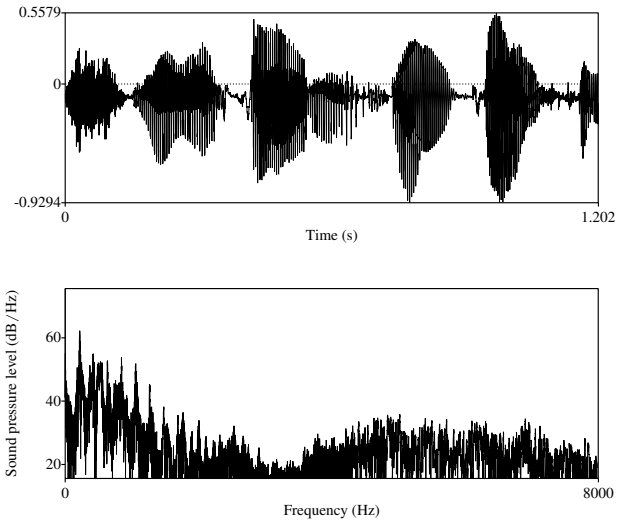


Abbildung 2.2: Von der Wellenform zum Spektrum.

Wird auf den Logarithmus des Spektrums nun nochmals eine inverse Fouriertransformation angewandt, entsteht das sogenannte Cepstrum.

Dieses Cepstrum wird für die folgenden Merkmale genutzt:

Mel Frequency Cepstral Coefficients (MFCC)

MFCCs sind n-dimensionale Vektoren mit reelen Zahlen. In Zeitfenstern von typischerweise 20ms mit jeweils 10ms Überlappung erstellt, beschreiben die Koeffizienten eine Parametrisierung des in Mel umgerechneten Spektrums in Abhängigkeit von den physischen Charakteristika des Sprechers [17]. Dafür werden die Abschnitte per Fouriertransformation in das Spektrum umgewandelt, der Logarithmus genommen, in die Mel-Scala überführt und zusammengefasst und schließlich (mittels einer diskreten Cosinustransformation) dekorreliert [14].

Linear Prediction-based Cepstral Coefficients (LPCC)

Beim LPC-Modell wird Sprache als ein Zusammenspiel von 4 lokalen Filtern beschrieben, wobei der Sprachapparat durch einen globalen Filter (ARMA[1]) repräsentiert werden kann. Die Koeffizienten

[1] Autoregressive moving average model

dieses Filters reichen aus, um einen Sprecher zu charakterisieren. Dafür werden die Koeffizienten der Filter jeweils für ein Fenster des Spektrums (typische Fenstergröße wie bei MFCCs) geschätzt. Aus diesen Koeffizienten können dann die cepstralen Koeffizienten berechnet werden [4].

2.2.2 Modelle

Gaussian Mixture Matrix (GMM)

Liegen die Sprechermerkmale vor, muss noch die Wahrscheinlichkeit bestimmt werden, dass die Merkmale dem Modell eines Nutzers entsprechen. Eine dafür häufig genutzte Formel ist

$$\Lambda(X) = \log p(\vec{X}|\lambda_{hyp}) - \log p(\vec{X}|\lambda_{\overline{hyp}})$$

Dabei ist $\vec{X}$ die Menge der Merkmalsvektoren, $p(\vec{X}|\lambda_{hyp})$ die Wahrscheinlichkeit der Hypothese, dass die Merkmale zum Sprecher gehören, und $p(\vec{X}|\lambda_{\overline{hyp}})$ die Wahrscheinlichkeit, dass sie nicht zum Sprecher gehören [4].

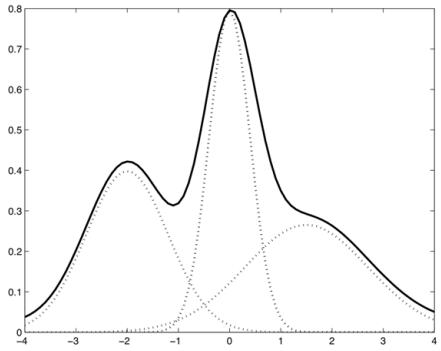


Abbildung 2.3: Symbolbild einer Gaussian Mixture Matrix, einer Annäherung einer Funktion mit Gaußschen Dichtefunktionen. Abb. übernommen aus [11].

Für die Wahrscheinlichkeitsfunktion $p(\vec{X}|\lambda)$ sind bei textunabhängiger Sprecheridentifikation GMMs (siehe Abb. 2.3) die bisher beste Wahl [4]. Sie sind definiert als

$$p(\vec{X}|\lambda) = \sum_{i=1}^{M} w_i p_i(\vec{x})$$

wobei $p_i(\vec{x})$ die angepasste Gaußsche Dichtefunktion ist:

$$p_i(\vec{x}) = \frac{1}{(2\pi)^{n/2}|\Sigma_i|^{1/2}} \times \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu}_i)' \sum_{i=1}^{-1}(\vec{x} - \vec{\mu}_i)\right)$$

Hierbei ist w_i das Gewicht, $\vec{\mu}_i$ der N-Dimensionale Mittelwertvektor und Σ_i die $N \times N$-dimensionale Kovarianzmatrix [17].

Weltmodell (UBM)

Das UBM[2] ist im Grunde nur bei Sprecherverifikation statt reiner Sprecheridentifikation notwendig, also wenn möglicherweise unbekannte Sprecher gerade sprechen könnten [17]. Dann ist das UBM die Alternative zum Sprechermodell, es wird geprüft, ob der Sprecher eher dem gewählten Modell oder dem UBM gleicht, und entspricht damit obigem $p(\vec{X}|\lambda_{\overline{hyp}})$.

[2] von engl.: Universal Background Model

Hidden Markov Modeling (HMM)
HMMs beinhalten, anders als GMMs, Informationen über den zeitlichen Ablauf. Bisher haben sie und andere komplizierte Wahrscheinlichkeitsfunktionen bei der textunabhängigen Sprechererkennung jedoch keinen Vorteil gegenüber GMMs gezeigt [4].

2.2.3 Training

Durch das Training des Modells wird es dahingehend angepasst, dass es bei den gegebenen Daten im System möglichst deutlich den richtigen Sprecher erkennt.

"**Maximum A Posteriori**"**-Training (MAP-Training)**
Bei der Erstellung des Modells des Sprechers wird bei Verwendung des GMM-UBM-Systems das Sprechermodell trainiert. Beim MAP-Training wird das UBM durch die MAP-Methode für jeden einzelnen Sprecher angepasst [17].
MAP-Training ist ein zweistufiger Prozess. Zuerst werden für jedes der M Mixture i des UBM Kennwerte geschätzt, um danach mit diesen die bisherigen Kennwerte des Mixture i ggf. anzupassen.
Für den ersten Schritt wird berechnet:

$$Pr(i|\vec{x}_t) = \frac{w_i p_i(\vec{x}_t)}{\sum_{j=1}^{M} w_j p_j(\vec{x}_t)}$$

Mit diesen Werten werden dann die neuen Parameter für die Kennwerte Gewicht, Mittelwert und Varianz berechnet:

$$n_i = \sum_{t=1}^{T} Pr(i|\vec{x}_t)$$

$$E_i(\vec{x}) = \frac{1}{n_i} \sum_{t=1}^{T} Pr(i|\vec{x}_t)\vec{x}_t$$

$$E_i(\vec{x}^2) = \frac{1}{n_i} \sum_{t=1}^{T} Pr(i|\vec{x}_t)\vec{x}_t^2$$

Im zweiten Schritt werden dann die Kennwerte des UBM für jedes Mixture i angepasst:

$$\hat{w}_i = (\alpha_i n_i/T + (1-\alpha_i)w_i)\gamma$$

$$\hat{\vec{\mu}}_i = \alpha_i E_i(\vec{x}) + (1-\alpha_i)\vec{\mu}_i$$

$$\hat{\vec{\sigma}}_i^2 = \alpha_i E_i(\vec{x}^2) + (1-\alpha_i)(\vec{\sigma}_i^{\,2} + \vec{\mu}_i^{\,2}) - \hat{\vec{\mu}}_i^2$$

γ ist ein Skalierungsfaktor, der Updatefaktor α_i wird bestimmt durch

$$\alpha_i = \frac{n_i}{n_i + r}$$

wobei r ein zu wählender Relevanzfaktor ist [4].

2.2.4 Normalisierung

Durch Normalisierung wird versucht, Unterschiede zwischen den Aufnahmen auszugleichen, damit Aufnahmeunterschiede wie ein anderes Mikrofon die Ergebnisse der Sprechererkennung möglichst gering beeinflussen.

T-Norm

T-Norm ist eine der effektivsten Normalisierungen. Ein Ergebnis s wird angepasst durch

$$\frac{s-\mu}{\sigma}$$

wobei μ und σ der Mittelwert und die Standardabweichung der Referenzmodelle sind [17].

2.2.5 Bewertung

Am Ende steht für einen bestimmten Nutzer für eine gegebene Sprachaufnahme eine Bewertung bereit. Diese Bewertung muss interpretiert werden. Über einen Grenzwert θ, ab dem eine Identifizierung akzeptiert wird, kann zwischen einer Erhöhung der false positives und einer Erhöhung der false negatives gewählt werden. Auf einer Erkennung-Fehler-Trade-Off-Kurve (DET) kann der Konflikt zwischen den Werten gut dargstellt werden (siehe Abb. 2.4).

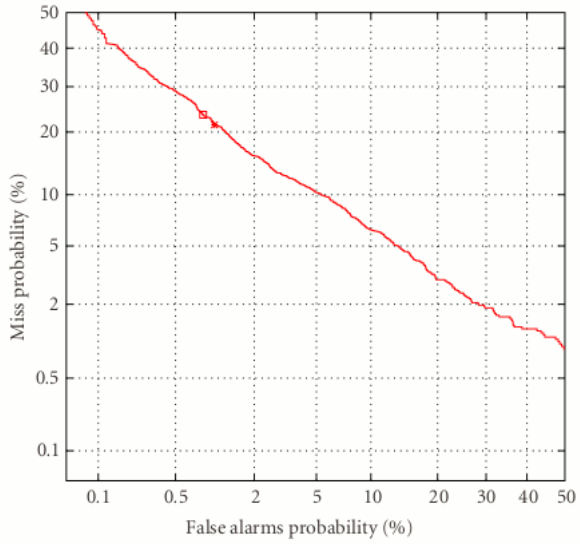


Abbildung 2.4: Beispiel einer DET-Kurve, übernommen aus [4].

Ein optimales θ hat den optimalen Wert, um richtige Erkennungen zu ermöglichen und falsche zu verhinden, also beide Fehler zu minimieren. Das gilt nur unter der Bedingung, dass der Fokus des Systems nicht darauf gelegt wird, dass keinesfalls false positives auftreten dürfen. Dafür wäre ein θ mit mehr false negatives optimal. Genau andersrum könnte es bei unkritischen Systemen sein, bei denen eine falsche Anmeldung verzeihlich ist. Man sieht, dass der optimale Grenzwert immer in einem Konflikt bestimmt wird.

2.3 Grundlagen der Körperformerkennung

Bei einer Körperformerkennung wird durch einen Vergleich zweier Abbilder eines Körpers die Ähnlichkeit bestimmt. Beispielsweise könnte die beliebig repräsentierte Silhouette verglichen werden. Zur Bestimmung dieses Körpers gibt es viele verschiedenen Ansätze, beispielsweise Polygonbeschreibungen in

Abhängigkeit eines Referenzkörpers [6] oder das Erraten von Körperregionen mittels vorher trainierten Entscheidungsbäumen [9].

Eine der Herausforderungen liegt darin, zu erkennen, welcher Teil des Gesamtbildes den Körper enthält, von dem die Körperform bestimmt werden soll. Eine weitere Herausforderung ist das Erkennen der Körperform unter Kleidung [3].

In dieser Arbeit lag der Fokus allerdings nicht auf dem Erstellen des Körpermodells. Vielmehr wurde dafür das proprietäre NITE genutzt, das diese Aufgabe übernahm (siehe Kapitel 4). Aber es wurde ein solches Modell benutzt, um Körper zu vergleichen und zu klassifizieren.

Körperformerkennung ist also analog zu [9] unterteilt in zwei Schritte: Zuerst wird der Körper modelliert, danach kann dieses Modell verglichen werden.

2.3.1 Modellerstellung

Ein Ansatz zur Erstellung eines Körperformmodells ist das Erkennen von Körpersegmenten. Sind diese definiert, können deren Gelenkknoten und Abschnitte als ein Skelett betrachtet werden.

In [9] wird dazu ein Ansatz beschrieben, der aus einem Tiefenbild dieses Skelett erstellen kann. Dieser Ansatz wird deswegen hier vorgestellt, weil er für die auch in dieser Arbeit genutzte Kinect entwickelt wurde - es ist aber nicht das Vorgehen, das von NITE genutzt wird und unbekannt ist.

Der Ansatz ist interessant, weil durch ihn über ein Tiefenbild ohne Bewegungsinformation und ohne Initialisierungspose ein Skelett bestimmt werden kann.

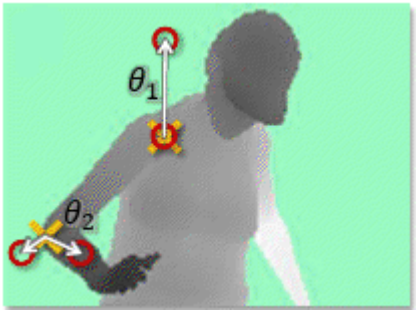


Abbildung 2.5: Ein Körper mit zwei Merkmalen, übernommen aus [9].

Gegeben ist ein Tiefenbild I. Für einen Pixel x des Bildes werden solcherart Merkmale bestimmt:

$$f_\theta(I,x) = d_I(x + \frac{\vec{u}}{d_I(x)}) - d_I(x + \frac{\vec{v}}{d_I(x)})$$

wobei $d_I(x)$ die Tiefe des Pixels x ist und $\theta = (\vec{u}, \vec{v})$ die Offsets $\vec{u}$ und $\vec{v}$ bestimmt, welche die Ausrichtung des Merkmals beschreiben (siehe Abb. 2.5).

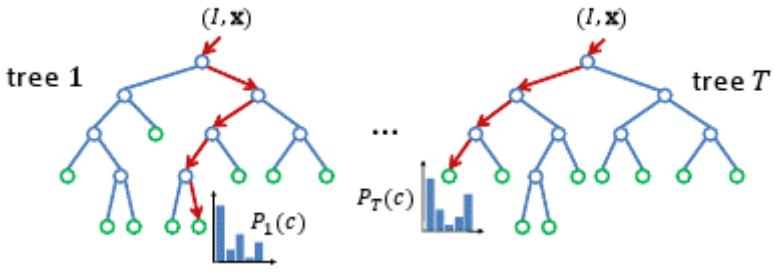


Abbildung 2.6: Ein Wald aus Entscheidungsbäumen, übernommen aus [9].

Diese Merkmale werden nun genutzt, um mithilfe von vorher trainierten Entscheidungsbäumen (siehe Abb. 2.6) das Körpersegment zu bestimmen, zu dem der Pixel gehört. Ein Wald besteht aus T Entscheidungsbäumen. Um einen Pixel zuzuordnen, wird bei jedem Baum bei der Wurzel begonnen und er nach unten traversiert. Dafür wird die obere Gleichung ausgeführt und das Ergebnis mit einem Grenzwert verglichen, der entscheidet, ob nach links oder nach rechts abgebogen wird. Die erreichten Blätter bestimmen das Körpersegment c. Dies entspricht

$$P(c|I,x) = 1/T \sum_{t=1}^{T} P_t(c|I,x)$$

So kann für jeden Pixel des Körpers das zugehörige Körpersegment bestimmt werden. Diese können dann zusammengefasst werden. Das so entstandene Skelett kann dann später einfacher genutzt werden, um z.B. Gestensteuerungen umzusetzen oder vielleicht auch Menschen zu klassifizieren.

2.3.2 Modellvergleich

Für den Vergleich von Körpern gibt es viele verschiedene Genauigkeitsstufen: Vom groben Vergleich der Körpergröße bis zum genauen Vergleich der Relationen aller Körperteile zueinander oder der Ähnlichkeit von Körperregionen [6]. Das Ergebnis kann wiederum unterschiedlich genau gespeichert werden. So könnte man beim Merkmal Körpergröße die gemessenen Menschen in zwei Klassen einteilen, in die Klasse "kleiner als 170 cm" und die Klasse "größer als 170 cm", und so sehr grob kategorisieren. Genauer wäre es selbst bei diesem äußerst ungenauen Merkmal, viele verschiedene Klassen mit unterschiedlichen Größen einzuführen und dadurch im Maximalfall nicht mehr in Oberklassen einzuteilen, sondern direkt mit einelementigen Mengen die Gleichheit der Körpergröße zu prüfen.

Die Körpergröße ist eines der ungenauesten Merkmale. Im Verlauf der Arbeit wurde die Länge von einzelnen Körpersegmenten als Merkmal genommen. Eine frühere Idee war, die Relation der Länge zweier Körpersegmente als Merkmal zu nutzen. Aber es könnte auch versucht werden, eine Körperform vollständig zu modellieren und diese Modelle miteinander zu vergleichen, und so das Verfahren eher der Sprechererkennung anzunähern.

Je kleinschrittiger die Einteilung, desto kleiner die Menge zurückgegebener Nutzer und damit die Gefahr, dass durch eine Messabweichung die Erkennung eine Menge von Nutzern ohne die richtige Person zurückgibt. Dies entspricht dem Konflikt zwischen Precision (möglichst nur die richtigen Ergebnisse) und Recall (möglichst alle richtigen Ergebnisse) in Information-Retrieval-Systemen [16, S. 869].

2.4 Hardware

Abbildung 2.7: Bild einer Kinect.

Als Aufnahmegerät wurde die Kinect von Microsoft genutzt. Die Kinect ist u.a. eine günstige Tiefenbildkamera, die ursprünglich als Zusatzhardware für die XBox 360 Spiele mit Körper/Gestensteuerung

ermöglichen sollte. Vor der Kinect gab es kein System, um mit Unterhaltungshardware Körpertracking robust umzusetzen [9].

In der Kinect (siehe Abb. 2.7) enthalten ist laut Spezifikation eine Tiefenkamera, die mit Infrarotlicht und einem CMOS-Sensor ein Tiefenbild erstellt. Zusätzlich enthalten ist eine normale Farbkamera und ein Mikrofonarray. Die Auflösung der Kameras beträgt 640x480 und die Genauigkeit der Tiefenwerte liegt bei 1 cm [1].

Als Rechner wurde ein EeeBox-PC von Asus eingesetzt. Dieser hat einen Atom-Prozessor von Intel, der alle nötigen Prozessorerweiterungen besitzt (wie die "Supplemental Streaming SIMD Extensions 3"), aber trotzdem nicht viel Rechenleistung hat, sodass auf ihm in Echtzeit laufende Erkennungsverfahren auch auf anderen modernen Systemen laufen sollten.

3 Konzept

Die richtige Mischung der in Kapitel 2 beschriebenen Verfahren zu finden ist kompliziert. Als Beispiel: Ein System sei sowohl durch eine Passworteingabe als auch durch eine sich häufig irrende Gesichtserkennung geschützt. Stimmt nun das Passwort, aber die Gesichtserkennung verneint die Identität, sollte doch trotzdem noch der Besitz des Passworts entscheiden. Dann aber wäre die 2-Faktor-Sicherung sinnlos. Ebenso wirkt bei probabilistischen Verfahren die externe Datenquelle wesentlich unwichtiger als der Test konkret vorliegender Testdaten. Der sinnvolle Anteil der jeweiligen Verfahren am Gesamtergebnis wäre schwer zu bestimmen. Deshalb werden in diesem Konzept nur probabilistische Verfahren mit vorliegenden Testdaten miteinander gekoppelt.

Es sei jedoch erwähnt, dass manche Varianten von Verbindungen von probabilistischen mit eindeutigen Verfahren interessant wären. Eines der Bedenken gegen Anmeldung durch den Besitz eines USB-Sticks könnte sein, dass dieser leichter gestohlen werden kann als ein Passwort verraten wird. Eine passive Körperformerkennung könnte die Anmeldung mit einem gestohlenen USB-Stick in manchen Fällen verhindern. Eine ähnliche Kombination mit einem passiven Sender und einem passiven probabilistischen Verfahren könnte eine sehr angenehme und für viele Systeme ausreichend sichere Anmeldung ermöglichen.

3.1 Verknüpfungen

Es wurden verschiedene Möglichkeiten erarbeitet, mit denen solche Verfahren miteinander kombiniert werden könnten. Im Wesentlichen sind es zwei Varianten, nämlich parallele oder sequentielle Verknüpfung, wobei die Frage des richtigen Konfidenzminimums (KM) weitere Abwandlungen der Konzepte ermöglicht.

Bei allen Varianten ist bewusst undefiniert, was passiert, wenn mehrere Nutzer die Anmeldung passieren, da dies vom jeweiligen System und der Perfomance der verwendeten Erkennungsverfahren abhängt.

Es seien im Folgenden n probabilistische Verfahren und ein Tupel (U, D) aus einer Nutzermenge U und Testdaten D gegeben.

3.1.1 Parallele Verknüpfung

Bei parallelen Verknüpfungen werden die Erkennungsverfahren unabhängig voneinander durchgeführt. Es gibt mindestens zwei verschiedene Verianten:

Globales KM

Für jeden Nutzer $u \in U$ bestimmt jedes Verfahren die Wahrscheinlichkeit X_i einer Übereinstimmung der Testdaten mit dem gespeicherten Profil von u. Die Gesamtwahrscheinlichkeit einer Übereinstimmung ist

$$p(X) = (X_i + ... + X_n)/n$$

Dabei muss jedes X_i natürlich die gleiche Skala haben. Optional könnten die Verfahren bei bekannter Fehlerquote unterschiedlich stark gewichtet werden, um Ergebnisse verlässlicher Verfahren stärker zu beachten (siehe Abb. 3.1).

Eine Anmeldung sollte schließlich dann erfolgen, wenn $p(X) > KM$ ist. Die Wahl des KM würde also bestimmen, wieviele Verfahren mindestens zustimmen müssen.

Lokales KM

Für jeden Nutzer $u \in U$ bestimmt jedes Verfahren, ob $X_i > KM_i$. Wenn ja, wird Zustimmung signalisiert. Wenn jedes Verfahren zugestimmt hat, also $\bigwedge_{i=1}^{n}(X_i > KM_i)$, dann kann der Nutzer angemeldet werden (siehe Abb. 3.2).

Alternativ könnten unterschiedlich große Mehrheiten zur Anmeldung ausreichen, damit das System nicht anfällig für "das Problem der byzantinischen Generäle" ist [12].

3.1.2 Sequentielle Verknüpfung

Die Verfahren werden hintereinander angeordnet. Jedes Verfahren bildet ein neues Tupel (U', D) mit $U' \subseteq U$ und gibt dieses an das folgende Verfahren weiter. Dabei sind nur die Nutzer $u \in U'$, für die $P(X_i) > KM_i$. Die sequentielle Verknüpfung arbeitet also mit zumindest impliziten lokalen KMs (siehe Abb. 3.3).

3.2 Gewählte Verknüpfung

Da die Körperformerkennung in der implementierten Form kein explizites Konfidenzminimum kennt, sondern direkt eine Menge von möglichen Identitäten zurückgibt, wurde die sequentielle Verknüpfung gewählt.

Um die Erkennungsrate des Hauptverfahrens, der Sprechererkennung, zu steigern, wird also die Körperformerkennung zuvor durchgeführt. Diese kann die Person mindestens in eine grobe Kategorie einordnen und so mögliche Identitäten ausschließen. Diese müssen dann von dem zweiten Verfahren gar nicht mehr getestet werden. Das Vorgehen ließe sich auch umkehren, indem erst die Sprechererkennung mögliche passende Identitäten herausfiltert und dann diese von der Körperformerkennung geprüft werden. Die letztendliche Erkennungsrate des Gesamtsystems sollte deutlich steigen können, da $P(Gesamtsystem) = P(B|A) \times P(A)$ statt $P(B) \times P(A)$. Das Verfahren A muss also nur noch unter der Bedingung arbeiten, dass das Verfahren B erfolgreich durchlaufen wurde, was bei einem guten Verfahren B die Auswahl an Möglichkeiten und damit die Möglichkeiten für Fehler reduziert.

3.3 Anforderungen an die Verfahren

Die mögliche Steigerung der Erkennungsrate des zweiten Verfahrens hängt von der Zuverlässigkeit des ersten Verfahrens ab. Filtert es zuverlässig einige falsche Identitäten aus, reduziert es die Auswahl möglicher Ergebnisse für das zweite Verfahren. Es reduziert damit die Wahrscheinlichkeit, dass die falsche Identität ausgewählt wird. Jedoch schadet das erste dem zweiten Verfahren immer dann, wenn es irrt und die echte Identität herausfiltert.

Es ist also wichtig, dass das erste Verfahren möglichst niemals die echte Identität herausfiltert, sondern die Auswahl nur schwach eingrenzt. Es dürfen also eher false positives als false negatives auftreten. Konkret kann dies durch die Wahl eines Verfahrens mit möglichst hohem Recall erreicht werden. Würde z.B. die Sprechererkennung als Vorfilterung genutzt werden, müsste θ (siehe Abschnitt 2.2.5) entsprechend niedrig gesetzt werden, damit der richtige Sprecher möglichst selten aussortiert wird.

3.4 Gewählte Verfahren

Auch bei gegebener Verknüpfung könnten unterschiedliche Erkennungsverfahren gewählt werden, auch ist das Konzept erweiterbar. Die gewählten Verfahren wurden gewählt, weil:

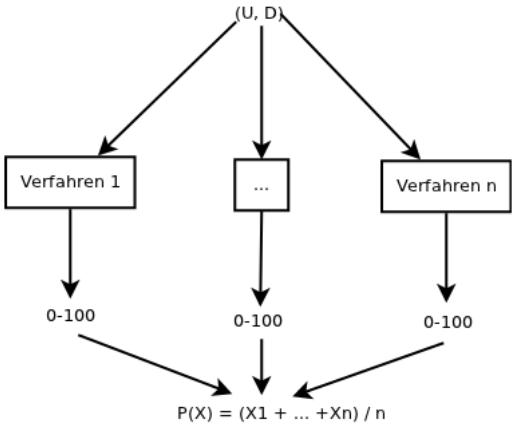


Abbildung 3.1: Parallele Verknüpfung mit globalen Konfidenzminimum.

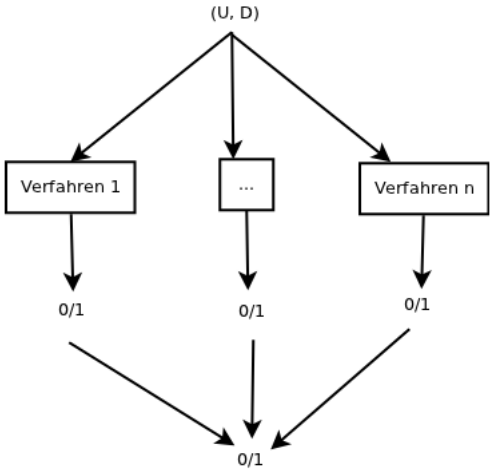


Abbildung 3.2: Parallele Verknüpfung mit lokalen Konfidenzminima.

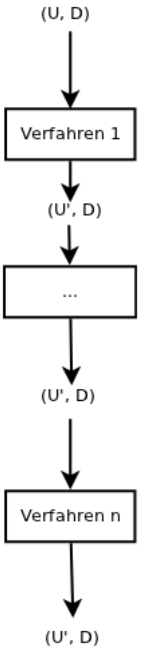


Abbildung 3.3: Sequentielle Verknüpfung.

Körperformerkennung

Die Vorerkennung soll durch eine Körperformerkennung durchgeführt werden. Sie eignet sich dafür, weil sie in der implementierten Form eher einer Körperformklassifizierung entspricht: Mehrere Personen können sich ein Körperprofil teilen. Dadurch wird ein hoher Recall erreicht.
Probabilistische Verfahren sind üblicherweise aus zwei Gründen ungenau:

- Sind die gewählten Merkmale einzigartig?
- Werden die Merkmale zuverlässig gemessen?

Bei der Körperformerkennung über die Länge einzelner Körpersegmente kann durch die Klassifizierung die zweite Ungenauigkeit entschärft werden, wenn die Varianz der einzelnen Längen bekannt ist (siehe Abschnitt 4.4). Die erste Ungenauigkeit ist dagegen hier erstmal irrelevant, weil eine Vorfilterung nicht zuverlässig Imposter aussortieren muss.

Sprechererkennung

Als Hauptverfahren eignet sich jedes Verfahren, das einzigartige Merkmale zur Erkennung nutzt und schließlich eine Identität favorisiert. Sprache als Merkmal könnte einzigartig sein, zumindest legt das die menschliche Wahrnehmung nahe und es ist Grundannahme der Sprechererkennung (siehe Abschnitt 2.2). Andererseits ist die Zuverlässigkeit der Sprechererkennung umstritten [5], sodass gerade sie durch eine Kombination mit anderen Erkennungsverfahren profitieren könnte. Da Sprechererkennung eine Wahrscheinlichkeit zurückgibt, ist der Sprecher mit der besten Bewertung der favorisierte Sprecher. Deswegen eignet sich die Sprechererkennung als Haupterkennungsverfahren.

3.5 Ideale Umsetzung

In einer perfekten Umsetzung würde das System so funktionieren:

Es steht eine Kamera und ein Mikrofon von einem System bereit, an dem noch kein Nutzer angemeldet ist. Ein Nutzer läuft in den Blickwinkel der Kamera, die daraufhin automatisch die Körperformerkennung startet. Von den registrierten Nutzern bleibt eine Untermenge über. Fängt der Nutzer nun an zu reden, prüft die Sprechererkennung für jeden aus der Untergruppe, ob dies die richtige Identität ist.

Verlässt er den Blickwinkel der Kamera oder ändert sich das Ergebnis des ersten Verfahrens (z.B. weil der getragene Mantel ausgezogen wurde) wird die Untergruppe zurückgesetzt bzw. aktualisiert.

Ein solches System wäre passiv, denn: Die Sprechererkennung müsste nicht als aktive Handlung vorgenommen werden, sie könnte Teil der Systembenutzung sein, z.B. wenn das System nicht durch Knopf oder Fernbedienung, sondern durch einen Sprachbefehl angeschaltet wird. Diese Eigenschaft würde verlorengehen, wenn die Sprechererkennung ein explizites Sprachmuster nur für den Zweck der Erkennung erfordern würde, wodurch wieder eine aktive Handlung (wenn auch eine möglichweise bequemere als eine Passworteingabe) gefordert würde.

4 Implementierung

Implementiert wurde ein Javaprototyp mit grafischer Oberfläche, der in einer Demo eine 2-Faktor-Erkennung durchführen soll. Dafür wurden die beiden Erkennungsmodule separat implementiert.

Der Prototyp führt eine Körperformerkennung und eine Sprechererkennung durch. Diese können einzeln ausgeführt oder wie in Abschnitt 3.2 beschrieben hintereinander ausgeführt werden, sodass erst die Körperformerkennung Nutzer aussortiert und dann die Sprechererkennung die Bewertung der verbliebenen Identitäten durchführt.

Als Sensor für beide Module wird eine Kinect genutzt. Insbesondere für die Körperformerkennung war die Kinect die beste Wahl, da die Tiefenkamera die Körperform erst wirklich möglich macht und zusätzlich fertige Skeletterkennungsmodule für die Kinect bereitstehen.

Die Entwicklungsumgebung war GNU/Linux, sodass der Prototyp unter diesem Betriebssystem lauffähig ist. Eine Anpassung an andere Betriebssysteme wie Windows wäre möglich. Die Einschränkung kommt daher, dass die genutzten Binaries für Linux kompiliert und für die Aufnahmen kleine Bashskripte genutzt wurden.

4.1 Architektur

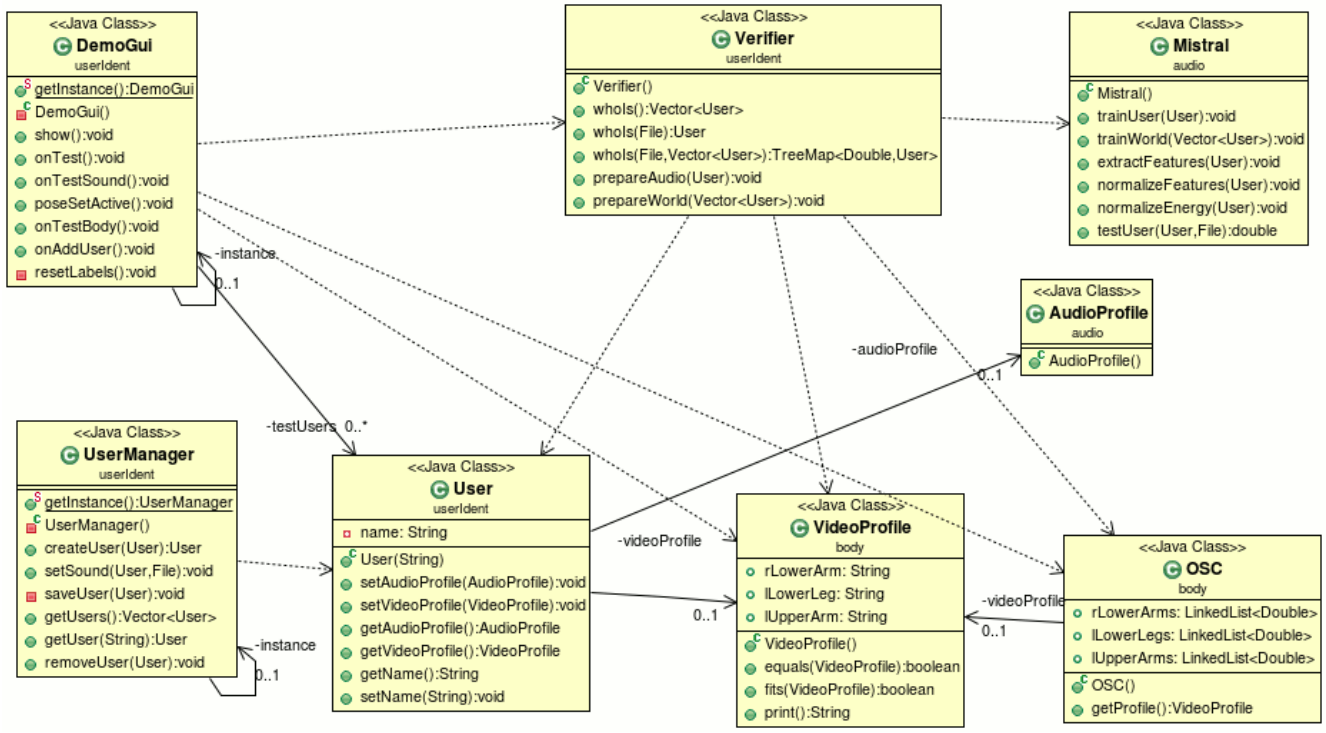


Abbildung 4.1: Das vereinfachte Klassendiagramm des Prototypen.

Verifier
Kern des Programms ist die Verifier-Klasse. Sie stellt Funktionen bereit, mit denen die Erkennungsverfahren gestartet werden können. Weitere Erkennungsverfahren könnten direkt in den whoIs()-Methoden angesprochen werden.

Mistral Die Mistralklasse ist ein loser Wrapper um die ausführbaren Dateien des Mistralframeworks. Alle Funktionen der Sprechererkennung sind hier definiert. Diese wird näher in 4.5 erklärt.

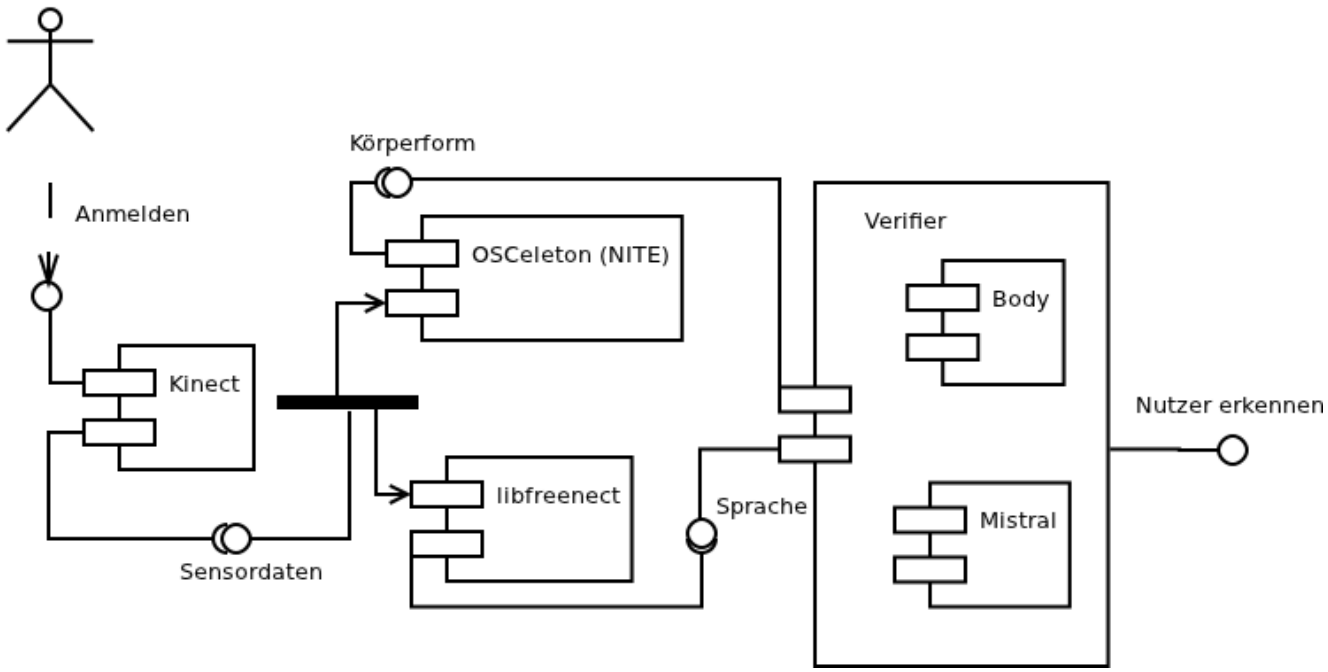


Abbildung 4.2: Das vereinfachte Komponentendiagramm des Prototypen.

OSC
Die Körperformerkennung wird über die OSC-Klasse gesteuert. Hier wird mithilfe der Kinect ein Skelett des Nutzers erstellt, woraus ein sehr grobes Videoprofil errechnet wird. Dies wird näher in 4.4 erklärt.

User
Ein gespeicherter User hat einen Namen, ein Audio- und ein Videoprofil. Das Audioprofil ist nur pro forma eingebaut, da in der Praxis die Sprechererkennung in Mistral auf im Dateisystem korrekt platzierte Dateien vertraut. User werden in einer Objektdatenbank [1] gespeichert. Der UserManager fungiert als Wrapper um diese Datenbank, das heißt er holt bekannte und speichert neue User.

DemoGui
Die DemoGui-Klasse stellt und steuert die Oberfläche.

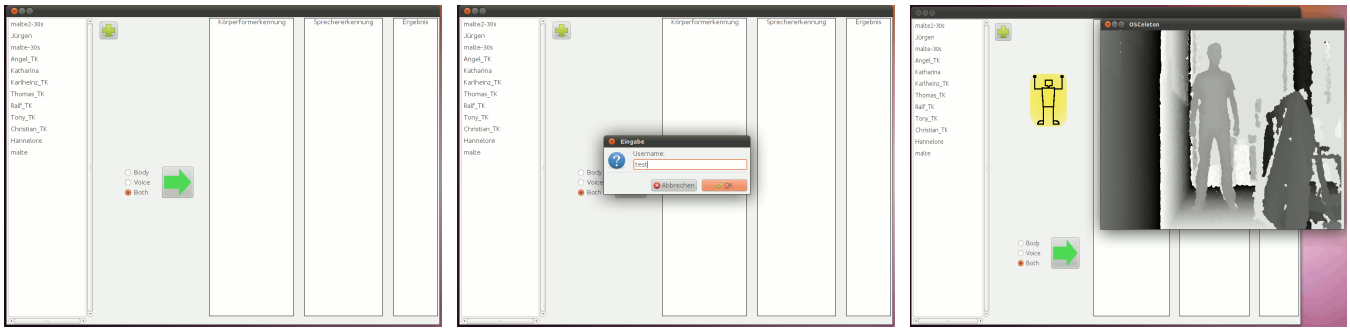
Designpatterns
Erkennbar sind die Singletons bei der DemoGui und bei UserManager. In beiden Fällen werden so Probleme durch mehrfache Instanziierung dieser Klasse verhindert. Bei der DemoGui wird zusätzlich die Manipulation der Oberfläche von anderen Programmteilen aus erleichert.

4.2 Programmablauf

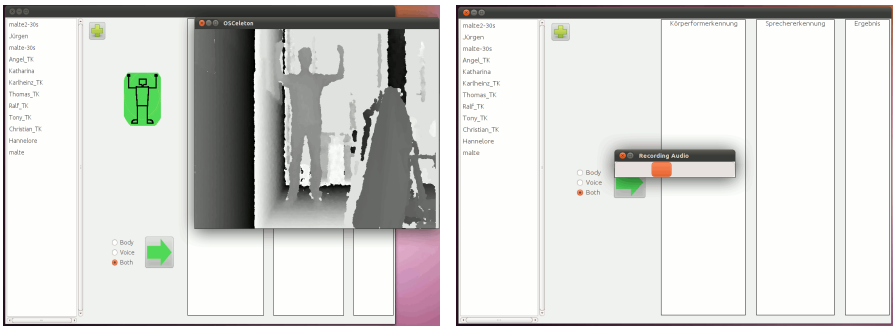
4.2.1 Nutzer hinzufügen

Ein Klick auf das grüne Plus startet den Vorgang (Abb. 4.3a). Zuerst wird der Name eingegeben (Abb. 4.3b). Danach wird das Tiefenbild der Kinect angezeigt, um für die Körperformerkennung den Nutzer einfacher positionieren zu können (Abb. 4.3c). Der Nutzer muss nun die Initialisierungsgeste einnehmen und halten (Abb. 4.3d). Es folgt die Audioaufnahme (Abb. 4.3e).

[1] http://www.db4o.com/



(a) Start **(b)** Namenseingabe **(c)** Körperformerkennung



(d) Körper erkannt **(e)** Audioaufnahme

Abbildung 4.3: Programmablauf beim Hinzufügen eines Nutzers.

Interner Ablauf

Durch den Klick auf den Button mit dem Plus wurde der Trainingsmodus aktiviert. Abgesehen davon startet das gleiche Vorgehen wie bei einer normalen Erkennung.

Von der *Gui*-Klasse aus wird eine Instanz der *Verifier*-Klasse erzeugt. Diese startet das externe Programm OSCeleton, das die Gelenkknoten des Skelettmodells ausliest und an einen Port schickt, den *OSC* abhört.

Nun bekommt *OSC* also Skelettwerte und daraus kann das Körpermodell gebaut werden. Dafür wird die Klasse *Body* genutzt. Sind die nötigen Gelenkknoten gesetzt, berechnen die Getterfunktionen für die einzelnen Längen diese automatisch mithilfe einer *Calc*-Klasse, die auch die Grenzwerte speichert und später die Körperformklassen einteilen kann. Die Klassen des Körpermodells werden gebildet, indem 3000 Gelenkknoten gesetzt werden, alle 15 Knoten werden die Längen gemessen und gespeichert und danach der Median bestimmt.

Dieses Körpermodell kann nun gespeichert werden.

Die *Gui*-Klasse startet nun eine Audioaufnahme, die an die *Mistral*-Klasse geleitet wird. Dieser Wrapper um das Mistral-Framework startet einzelne Programme mit den passenden Konfigurationsdateien, die daraufhin Merkmale extrahieren, normalisieren und das erstellte Modell des Nutzers speichern.

Nun sind die nötigen Modelle für eine Körperform- und eine Sprechererkennung dem neuen Nutzer zugeordnet.

4.2.2 Nutzer erkennen

Es kann ausgewählt werden, ob eines der Erkennungsverfahren oder beide wie in 3.2 beschrieben genutzt werden sollen. Werden beide ausgewählt, werden sie wie bei der Anmeldung hintereinander gestartet. Abschließend wird nach jedem Verfahren das Ergebnis angezeigt und ganz am Ende der Nutzer mit

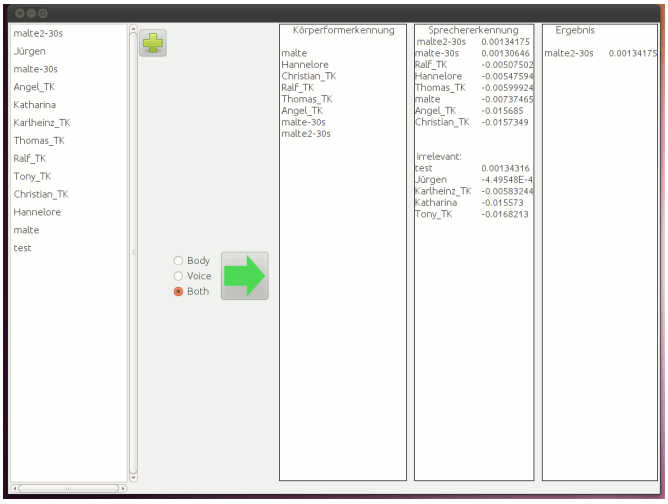


Abbildung 4.4: Ergebnisanzeige nach einem Erkennungsvorgang, vorher wie in Abb 4.3, nur ohne Namenseingabe.

der besten Bewertung besonders hervorgehoben, wenn die beiden Verfahren kombiniert wurden. (siehe Abb. 4.4).

Interner Ablauf

Wieder wird der *Verifier* instanziiert. Dieser bekommt die Menge der im System bekannten Nutzer. Durch den Aufruf der *whoIs(Vector<User> user)*-Funktion im Verifier instanziiert dieser die *OSC*-Klasse.

Diese verhält sich genauso wie beim Hinzufügen von Nutzern.

Also bekommt der Verifier von *OSC* ein Körpermodell, das er mit den Modellen der Nutzer vergleicht und alle Nutzer aus der Menge entfernt, deren Modell nicht mit dem gegebenen übereinstimmt (siehe Abschnitt 4.4).

Die verbliebenen Nutzer werden von der *Gui* zusammen mit einer Audioaufnahme (ohne Leerstellen) an den *Verifier* gegeben. Dieser startet daraufhin die *Mistral*-Klasse. Diesmal wird nicht einfach das Modell des Nutzers gespeichert, sondern *testUser(User user)* aufgerufen. Diese gibt eine Bewertung zurück, die aussagen soll, wie wahrscheinlich laut Mistral eine Übereinstimmung des gespeicherten Nutzers mit dem Sprecher ist.

Diese Bewertung und das Ergebnis der Körperformerkennung werden in der Gui in den Textfeldern angezeigt.

4.3 Kinectanbindung

Um auf die Funktionen der Kinect zuzugreifen, werden die beiden unter Linux verfügbaren Möglichkeiten genutzt:

- libfreenect[2] zur Audioaufnahme
- NITE[3] mit OpenNI[4] zur Skeletterkennung

Unter Windows wäre die Alternative das offizielle SDK von Microsoft[5] für C++, C# und Visual Basic. Dieses wurde für diese Arbeit nicht genutzt, weil es erst im Laufe der Arbeit erschienen ist und es nur unter Windows läuft.

[2] http://openkinect.org/wiki/Main_Page
[3] http://www.primesense.com/?p=515
[4] http://www.openni.org/
[5] http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/

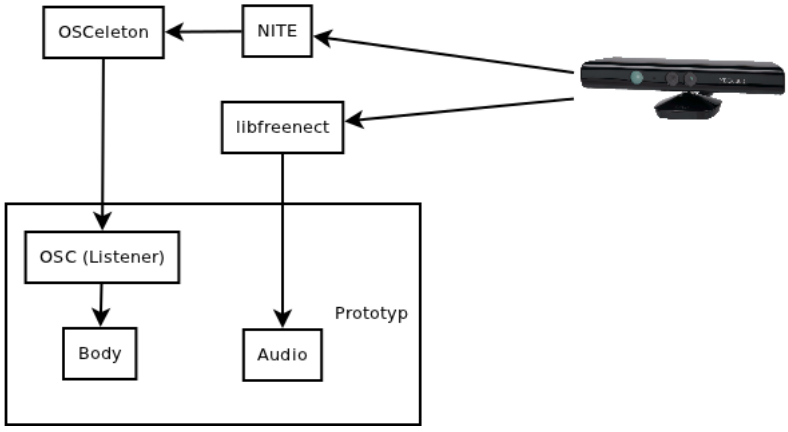


Abbildung 4.5: Anbindung der Kinect an den Prototypen.

Der Protoyp startet OSCeleton. Das von NITE berechnete Skelettmodell wird per OSCeleton[6] zurück an den Protoypen gesendet. Dieser definiert mit den Werten der Gelenkknoten ein internes Körpermodell, das später zur Berechnung der Körpermerkmale genutzt wird. Die eigentliche Körperformerkennung wird also vom proprietären NITE für das restliche System in einer Blackbox durchgeführt (siehe Abb. 4.5).

4.4 Körperformerkennung

Gestartet wird die Körperformerkennung durch das Einnehmen der in Abb. 4.7 schematisch gezeigten Initialisierungshaltung. Nur in dieser Haltung startet NITE damit, ein Skelettmodell zu berechnen. Nachdem ein Nutzer einmal erfasst wurde, könnte er sich frei vor der Kamera bewegen und das Modell würde trotzdem weiterhin berechnet werden.

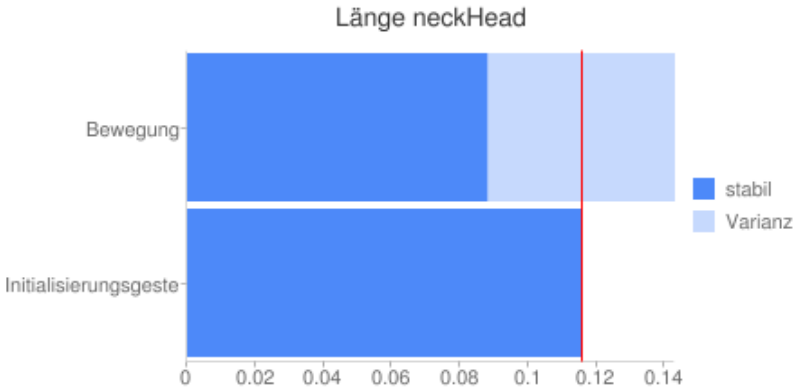


Abbildung 4.6: Mögliche Auswirkung der größeren Varianz bei Bewegung gegenüber der bei Verharren in der Initialisierungsgeste auf die Länge neckHead (siehe 4.7).

In mehreren Tests hat sich jedoch herausgestellt, dass das Skelettmodell von NITE nicht stabil ist. Insbesondere bei Bewegung (siehe auch Tabelle 4.2 und Abb. 4.6) verändert sich die Länge der einzelnen Körperteile, auch relativ zueinander. Dies ist bei der eigentlichen Aufgabe der Kinect, einer Skeletterkennung zum Erkennen von Posen und Gesten, kein ernsthaftes Problem. Das Wiedererkennen von Körpern wird durch diese Verschiebung aber schwierig. Dadurch war es nicht möglich, einfach direkt unter Einbeziehung einer kleinen Messvarianz die Gleichheit der Skelette zu prüfen.

[6] https://github.com/Sensebloom/OSCeleton

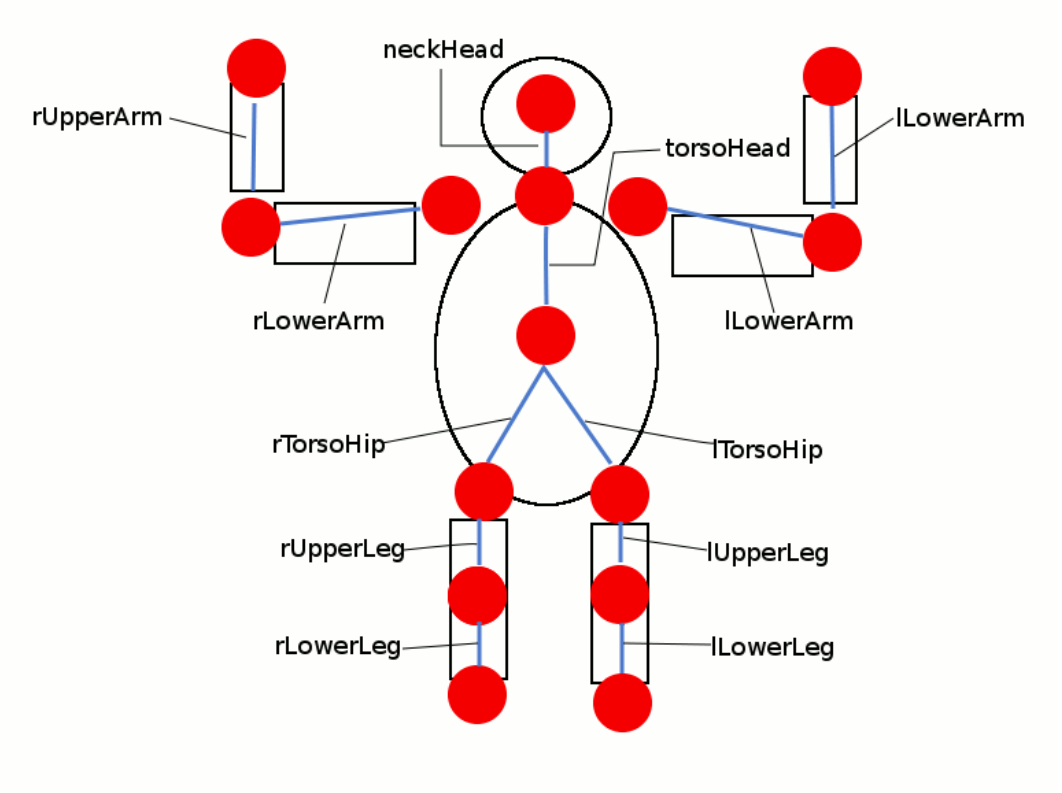


Abbildung 4.7: Schematische Darstellung des Körpermodells und der Initialisierungshaltung. Die Kreise kennzeichnen die Gelenkknoten, die benannten Striche die errechneten Größen.

4.4.1 Attributwahl

Daraufhin wurde mithilfe von waffles[7] eine Attributselektierung durchgeführt, um zu untersuchen, ob das Skelettmodell nicht doch stabile Körpermerkmale beinhaltet, die eine Identifizierung ermöglichen. Als relativ stabil haben sich die Längen des linken Oberarms, des rechten Unterarms und der linken Wade erwiesen, weswegen diese als die bestimmende Attribute des Körpermodells gewählt wurden.

Für diese Attributselektierung wurde dreimal das Körpermodell von fünf verschiedenenen Testpersonen bestimmt, einmal fehlten in der Messung Werte und es ergab sich nur ein komplettes Modell. Die Testgruppe bestand aus 4 Männern und einer Frau mit unterschiedlichen Körpergrößen.

Für jedes Körpermodell wurden etwa 400 mal die 15 Skelettkoordinatenpunkte gemessen. Für jedes Körpermodell lagen also ungefähr 6000 Skelettkoordinatenpunkte vor. Aus denen wurden 400 mal 12 Längen von Körperabschnitten berechnet. Von diesen wurde jeweils der Untermedian bestimmt. Es blieben also 5 Klassen von Körpern mit je dreimal 12 Attributen bzw. einmal nur einmal 12 Attributen (siehe Tabelle 4.3).

Die von waffles durchgeführte Attributselektierung reduzierte dieses Modell auf eines mit den 3 aussagekräftigsten Attributen (siehe Abb. 4.4), indem zuerst alle Werte normalisiert und dann ein Logit-Modell ([2, S. 70 ff]) trainiert wurde, um die Klasse zu erraten. Das Attribut mit dem niedrigsten Gewicht wurde entfernt und der Vorgang wiederholt[8].

[7] http://waffles.sourceforge.net/
[8] http://waffles.sourceforge.net/tutorial/dimred.html

4.4.2 Klassifizierung

Der Median von 200 Messwerten pro Attribut bildet das Körpermodell. Es wird aber, wie in Abschnitt 2.3 beschrieben um den Recall zu steigern, nicht der eigentliche Zahlenwert genommen, sondern dieser in eine Klasse eingeteilt:

- A für Längen über dem jeweiligen Grenzwert
- B für Längen unter dem jeweiligen Grenzwert
- C für Längen innerhalb einer Umgebung δ der Größe der Varianz um den Grenzwert

Als Grenzwert dient der Median der Messwerte des ersten eingelesenen Nutzers.

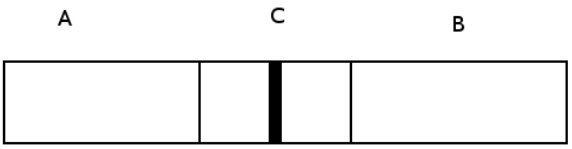


Abbildung 4.8: Das System der Körperklassen.

Die Klasse C dient hierbei als Wildcard, wird also bei einem Vergleich zweier Körpermodelle sowohl A als auch B gleichgestellt. Diese Klasse wurde eingeführt, damit Körpersegmente mit Längen nahe des Grenzwertes nicht mal in A und mal in B eingeordnet werden und dadurch nicht zuverlässig wiederholt erkannt werden können.

Merkmal	Klassen
lUpperArm	{A,B,C}
rLowerArm	{A,B,C}
lLowerLeg	{A,B,C}

Tabelle 4.1: Das Körpermodell.

Es wird also eine Klassifizierung mit drei Merkmalen von Skeletten durchgeführt, nämlich der Länge von drei Körpersegmenten, die jeweils eine von drei Klassen annehmen können (siehe Tabelle 4.1). Da eine der Klassen eine Wildcard ist, entspricht das bzw. sind das im besten Fall 2^3 Kombinationen, mit denen ein Körper klassifiziert werden kann. Durch die Einführung weiterer Klassen oder stabiler Merkmale würde die Klassifizierung sehr schnell stärker zwischen den Körpern differenzieren können.

Attribut	Varianz (unbeweglich)	Varianz bei Bewegung
lLowerArm	.00000189802413601353	.00038251090998282051
rLowerArm	.00000522260986681577	.00029441575930172447
lLowerLeg	.00000683409685323816	.00019825732269323030
rLowerLeg	.00000656352197923367	.00017356448686726196
lTorsoHip	.00000191805083982690	.00001653063711684169
rTorsoHip	.00000212384711803680	.00001359975475265072
lUpperArm	.00001164350666161873	.00011172634766307060
rUpperArm	.00000707820419489244	.00042428532077119166
lUpperLeg	.00000116671928634137	.00012047864802200695
rUpperLeg	.00000071615215435826	.00015064751121774646
torsoNeck	.00000052025019651924	.02495136104634816039
neckHead	.00001214201176281080	.02746950353640956045

Tabelle 4.2: Varianz bei Stillhalten gegenüber der bei Bewegung.

Klasse	lLowerArm	rLowerArm	lLowerLeg	rLowerLeg	lTorsoHip	rTorsoHip	lUpperArm	rUpperArm	lUpperLeg	rUpperLeg	torsoNeck	neckHead
A	0.16206	0.16101	0.22718	0.22191	0.13228	0.12986	0.16103	0.10553	0.24622	0.24744	0.12372	0.11697
A	0.15755	0.15943	0.22535	0.22409	0.13007	0.12719	0.14259	0.11891	0.24353	0.24184	0.12193	0.11567
A	0.15925	0.15669	0.22970	0.22116	0.12944	0.12865	0.14097	0.10849	0.24467	0.24382	0.12199	0.11578
B	0.12852	0.13681	0.16334	0.17130	0.10073	0.09537	0.16806	0.11084	0.27675	0.28762	0.09178	0.09468
B	0.13452	0.13674	0.18666	0.16406	0.11477	0.11308	0.11864	0.11075	0.21392	0.21617	0.10779	0.10584
B	0.13341	0.13753	0.19312	0.20399	0.11574	0.11241	0.12547	0.10524	0.21672	0.21505	0.10755	0.10655
C	0.15150	0.15180	0.24072	0.22582	0.12202	0.12373	0.12104	0.09494	0.23216	0.23206	0.11610	0.11170
C	0.15030	0.15931	0.27519	0.26007	0.11586	0.11749	0.12730	0.10690	0.22021	0.22047	0.11024	0.10780
C	0.15170	0.15252	0.24863	0.23860	0.12097	0.12243	0.11298	0.10229	0.22943	0.22955	0.11486	0.11050
D	0.15424	0.15080	0.19998	0.22184	0.12442	0.12225	0.11249	0.09975	0.23271	0.23324	0.11631	0.11304
E	0.17508	0.17304	0.19381	0.18910	0.13076	0.13149	0.11979	0.10977	0.25072	0.24845	0.12374	0.11738
E	0.16820	0.15642	0.19602	0.18512	0.12943	0.13335	0.12323	0.09792	0.25082	0.25034	0.12409	0.11770
E	0.15701	0.15636	0.20746	0.20799	0.12409	0.12610	0.11777	0.09729	0.23668	0.23695	0.11823	0.11369

Tabelle 4.3: Ausgangspunkt der Attributselektierung (Werte gekürzt).

Klasse	rLowerArm	lUpperArm	lLowerLeg
A	0.16101317386636	0.16103103878325	0.22718785217638
A	0.15943679229718	0.14259887466541	0.2253513415621
A	0.15669097541953	0.14097426899451	0.22970935016877
B	0.13681846657208	0.16806622978017	0.16334726410273
B	0.13674544102256	0.11864546906838	0.1866657149221
B	0.13753252241876	0.12547425324276	0.19312118845388
C	0.15180297266405	0.12104365458921	0.24072980464288
C	0.15931047146701	0.1273097085417	0.27519650562097
C	0.15252021387818	0.11298410803766	0.24863334596404
D	0.15080445657393	0.1124958566721	0.19998974912313
E	0.17304584083474	0.11979320553102	0.19381466777393
E	0.15642079742933	0.12323821010595	0.19602865415658
E	0.15636312824872	0.11777017741409	0.20746466640118

Tabelle 4.4: Ergebnis der Attributselektierung bei gehaltener Initialisierungsgeste.

4.5 Sprechererkennung

Für die Sprechererkennung wird das Mistral-Framework[9] genutzt. Im Wesentlichen wurde dabei das Vorgehen und die Konfiguration des Praat-Plugins[10] übernommen, das Lindh in [13] vorgestellt hat.

4.5.1 Vorgehen der Sprechererkennung

Zuerst schneidet sox[11] Leerstellen aus der Aufnahme heraus. Die folgende Merkmalsextraktion wird durch SPro[12] durchgeführt. Danach normalisiert Mistral die extrahierten Merkmale mithilfe des UBMs und erstellt das Modell des Nutzers (MAP-Training) oder vergleicht es mit den existierenden Modellen (siehe Abb. 4.9).

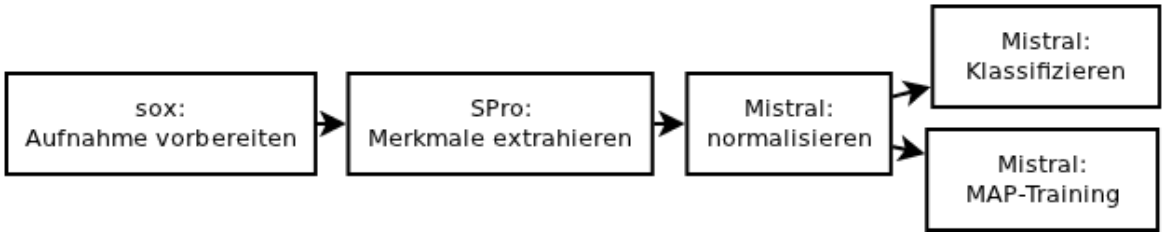


Abbildung 4.9: Die Implementierung der Sprechererkennung.

4.5.2 Erstellung des UBM

Um das UBM zu bauen, wurden Aufnahmen von 220 Menschen mit einer Gesamtlänge von 160 Minuten aus dem voxforge-Projekt[13] eingelernt. Aufnahmen von im Rahmen der Demo hinzugefügten Nutzern werden dagegen nicht automatisch dem UBM hinzugefügt, weil das Lernen des UBMs auf der Testhardware mehrere Stunden dauert.

Die Aufnahmen des UBM wurden automatisiert vom Server des voxforge-Projekts geladen. Der Prototyp wurde erweitert, um über Kommandozeilenparameter Nutzer anlegen und zugehörige Sprecherprofile erstellen zu können. Nachdem die Einzelaufnahmen der Sprecher zusammengefügt wurden, konnten diese dann eingelernt werden. Dabei wurde pro Sprecher nur eine Aufnahme genommen, um überproportional vertretene Sprecher nicht zu sehr die Faktoren des UBM bestimmen zu lassen.

Möglicher Nebeneffekt dieses Vorgehens könnte sein, dass durch die vielen verschiedenen genutzten Mikrofone und die sehr unterschiedliche Qualität der Aufnahmen das UBM geeignet für verschiedene Aufnahmegeräte ist, also nicht speziell an die Mikrofone der Kinect angepasst wurde.

4.5.3 Ergebnis der Sprechererkennung

Am Ende gibt der Prototyp eine Menge von Nutzern und deren Bewertung durch die Sprechererkennung aus, die von der Körperformerkennung selektiert wurden. Der am besten bewertete Nutzer wird selektiert. Es ist aber nicht spezfiziert, ab welcher Bewertung wirklich eine Anmeldung durchgeführt würde. Dafür müsste sehr genau getestet werden, welche Bewertung ein Imposter statt einer korrekten Identität üblicherweise bekommt.

[9] http://mistral.univ-avignon.fr/
[10] http://mistral.univ-avignon.fr/wiki/index.php/Praat_plugin
[11] http://sox.sourceforge.net/
[12] http://www.irisa.fr/metiss/guig/spro/
[13] http://www.voxforge.org/

5 Evaluation

Alle Tests wurden mit einer einzigen Kinect in dem gleichen Büro durchgeführt. Als Testpersonen stellten sich vor allem die Telekom-Mitarbeiter der benachbarten Büros zur Verfügung. Die Kinect wurde auf einem Tisch auf Hüfthöhe platziert. Für die Audioaufnahmen traten die Tester an die Kinect heran.

Von 10 Probanden wurde das Körper- und das Audioprofil gespeichert. Von einer dieser Testpersonen wurden zusätzlich 2 Profile mit längeren Audioaufnahmen angelegt.

5.1 Körperformerkennung

Die Körperformerkennung als Vorfilterung hat sich in den in Abschnitt 2.3 beschriebenen Grenzen bewährt. Der in Tabelle 5.1 dargestellte Test mit einer Testperson zeigte die erhofften 100% Recall, während die Precision wie erwartet nur bei 15.8% lag (siehe Abb 5.1).

Versuch 1	Versuch 2	Versuch 3	Versuch 4	Versuch 5
User 1	User 1	User 1	User 1	User 1
User 2	User 2	User 2	User 2	User 6
User 5	User 5	User 5	User 5	User 7
User 3	User 3	User 3	User 3	User 2
User 4	User 4	User 4	User 4	User 5
Tester	Tester	Tester	Tester	User 3
				User 4
				Tester

Tabelle 5.1: Test der Körperformerkennung mit einer Testperson und einer Testmenge von 10 Körperprofilen.

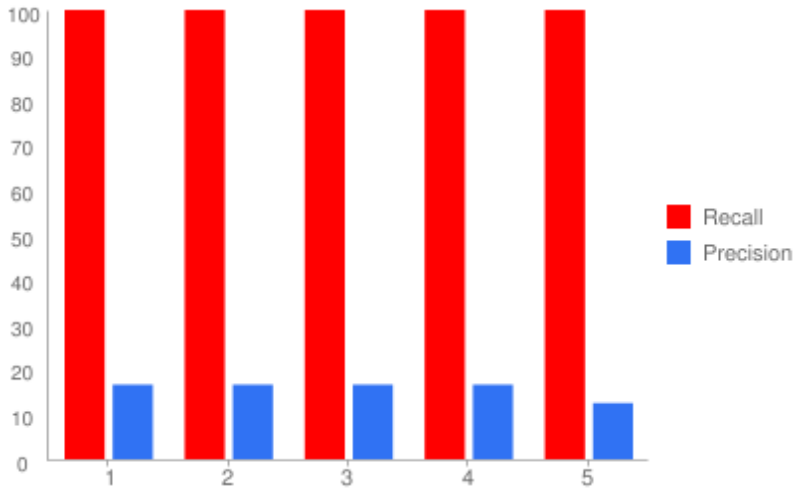


Abbildung 5.1: Recall und Precision der 5 Messungen in Prozent.

5.2 Sprechererkennung

Tabelle 5.2 zeigt die Ergebnisse eines Tests mit einer 10- und zwei 30-sekündigen Aufnahme, letztere lieferten das wesentlich bessere Ergebnis. Um positive Ergebnisse zu erhalten und also die Bewertung, dass die Testaufnahme eher zu einem Sprecherprofil als zum UBM gehört, sind vermutlich längere Aufnahmen beim Training des Sprechermodells besser.

Aufnahmelänge	10s	30s	30s
Bewertung	-0.0061257	7.86813E-4	8.08777E-4

Tabelle 5.2: Test der gleichen Sprachaufnahme mit Profilen aus 3 verschiedenen Trainingsaufnahmen.

Die Performance der Sprechererkennung war in den Tests (siehe Abb. 5.2) nicht zufriedenstellend. Nur in einem von 4 Fällen war das beste Ergebnis auch der gesuchte Sprecher. Außerdem waren die Ergebnisse sehr statisch: Die Reihenfolge der sortierten Bewertung war immer gleich, obwohl sich die Werte in jedem Test unterschieden.

Für den Test sowie beim Erstellen des Sprecherprofils durften alle Sprecher frei reden, allerdings lasen manche beim Training lieber von einem Blatt ab.

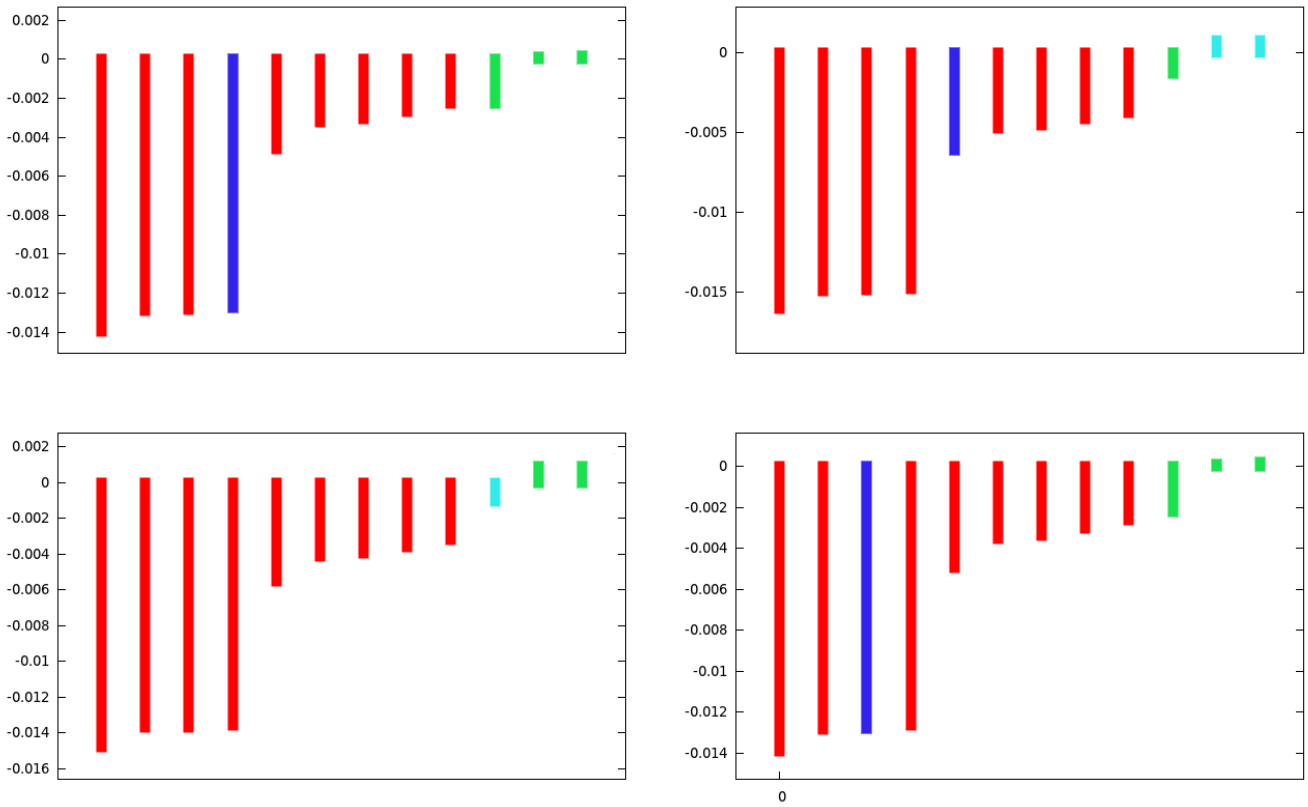


Abbildung 5.2: Ergebnisse der Sprechererkennung. Jeder Balken steht für die Bewertung eines Sprechers. Der gesuchte Sprecher ist blau, Sprecher mit 30-sekündiger Aufnahme beim Training (statt 10s) grün. Türkis markiert sind Sprecher, bei denen beides zutrifft.

5.3 Gesamtsystem

Ein Imposter-Angriff, also der Versuch einer Anmeldung einer dem System unbekannten Person, wird in Tabelle 5.3 gezeigt. Da das ganze System getestet wurde, wurde zuerst die Körperformerkennung und anschließend mit den verbliebenen Nutzern die Sprechererkennung durchgeführt. Die Körperformerkennung konnte direkt 8 der 12 möglichen Übereinstimmungen ausschließen. Bei der folgenden Sprechererkennung war jedoch ein Ergebnis positiv.

Körpererkennung bestanden	
Name	Sprechererkennungsbewertung
User 4-30s	2.38032E-5
User 6	-0.00123187
User 4	-0.00360958
User 5	-0.0121842

Aussortiert durch Körpererkennung	
Name	Sprechererkennungsbewertung
User 4-30s-2	1.14611E-4
User 6	-0.00163442
User 10	-0.00200174
User 7	-0.00218476
User 8-30s	-0.00238606
User 3	-0.0122983
User 2	-0.012351
User 1	-0.0134721

Tabelle 5.3: Imposterangriff.

Die Ergebnisse des Imposterangriffs sind auch ein Hinweis darauf, dass die Sprechererkennung mit besseren Trainingsaufnahmen besser funktionieren könnte: "User 8-30s" ist der gleiche Sprecher, der in den Tests (siehe Abb. 5.2) immer an dritter Stelle stand. Hier jedoch ist er an fünfter Position der aussortierten Sprecher. Die Reihenfolge ist also nicht völlig statisch. Bei einer nächsten Demonstration sollten alle Sprecher für die Modellerstellung mindestens 30 Sekunden sprechen.

Deswegen ist auch an dieser Stelle eine Evaluierung der Erkennungsrate des Gesamtsystems nicht sinnvoll. Weil diese Erkennungsrate von der Erkennungsrate der Sprechererkennung abhängt und diese mit den momentanen Trainingdaten nicht gut ist, würde die Erkennungsrate des Gesamtsystems nicht die potentielle, auch nicht die realistisch erreichbare, wiederspiegeln.

6 Fazit

Es wurde gezeigt, wie man mehrere probabilistische Erkennungsverfahren kombinieren könnte und beschrieben, wie das im konkreten Fall getan wurde. Die Körperformerkennung hat das Potential gezeigt, als Vorfilterung mit akzeptablen Ergebnissen genutzt werden zu können. Im derzeitigen Entwicklungszustand ist ein Praxiseinsatz des Verfahrens aber nicht möglich. Es müssten komplizierte Verbesserungen durchgeführt werden:

- Eine stabile Körperformerkennung wäre zu entwickeln, entweder mit einem komplett anderen Ansatz oder einem stabileren Skelett bei Bewegung.
- Die Körperformerkennung müsste ohne Initialisierungsgeste starten.
- Mehr Klassen für das Körpermodell mit besser bestimmten Grenzwerten sollten eingeführt werden.

Die genutzte Sprechererkennung jedoch zeigte keine guten Ergebnisse. Es ist zu vermuten, dass das an Schwächen im Trainingsprozess wie einer zu kurzen Trainingsaufnahme oder einem qualitativ nicht ausreichend hochwertigen Weltmodell lag, da in [13] bessere Ergebnisse mit der gleichen Technik beschrieben wurden. Es erscheint sinnvoll, dass die Sprechererkennung separat untersucht und verbessert würde, bevor weiter eine Verknüpfung versucht wird. Wünschenswert wäre eine robustere freie Sprechererkennung als einfach einbindbares Modul mit klar dokumentierten Parametern für einen guten Trainingsprozess mit genau diesem Modul und einem vordefinierten UBM.

Eine weitere Möglichkeit zur Erweiterung wäre ein Fokus auf mobile Geräte, wofür die Körperformerkennung mit einer Gesichtserkennung oder einem anderen passenden biometrischen Verfahren ausgewechselt werden könnte.

Mit einem Teil dieser Verbesserungen aber könnte mit Hardware wie der Kinect, die eine Tiefenkamera und ein Mikrofon bereitstellt, im Alltag Funktionen z.B. von Unterhaltungselektronik gesichert werden.

Literaturverzeichnis

[1] *PrimeSense Reference Design*. URL `http://www.primesense.com/?p=514`. 25.08.2011.

[2] Alan Agresti. An introduction to categorical data analysis.

[3] Alexandru O. Balan and Michael J. Blacka. The Naked Truth: Estimating Body Shape Under Clothing.

[4] Frederic Bimbot, Jean-Francois Bonastre, Corinne Fredouille, Guillaume Gravier, Sylvain Meignier, Teva Merlin, Javier Ortega-Garcia, Ivan Magrin-Chagnolleau, Dijana Petrovska-Delacretaz, and Douglas A. Reynolds. A Tutorial on Text-Independent Speaker Verfication. *EURASIP Journal on Applied Signal Processing*, pages 430–451, April 2004.

[5] Joseph P.M. Campbell, Reva Shen, William Schwartz, Jean-Francois Bonastre, and Driss Matrouf. Forensic Speaker Recognition - A need for caution. *IEEE Signal Processing Magazine*, pages 95–103, March 2009.

[6] Isaac Cohne and Hongxia Li. Inference of Human Postures by Classification of 3D Human Body Shape.

[7] Federal Financial Institutions Examination Council. Authentication in an Internet Banking Environment, 2005.

[8] Volker Dwello, Mark Huckvale, and Michael Ashby. How Is Individuality Expressed in Voice? An Introduction to Speech Production and Description for Speaker Classification. *Speaker Classification 1*, pages 1–20, 2007.

[9] Andrew Fitzgibbon, Mat Cook, Toby Sharp, Jamie Shotton, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-Time Human Pose Recognition in Parts from Single Depth Images.

[10] Simson Garfinkel. Design Principles and Patterns for Computer Systems that are Simultaneously Secure and Usable, 2005.

[11] Daniel Jurafsky and James H. Martin. Speech and Language Processing, 2000.

[12] L. Lamport, R. Shostak, and M. Pease. The Byzantine Generals Problem. *ACM Trans. Programming Languages and Systems*, (4):382–401, Juli 1982.

[13] Jonas Lindh. A first step towards a text-independent speaker verification praat plug-in using mistral/alize tools.

[14] Beth Logan. Mel Frequency Cepstral Coefficients for Music Modelling, 2000.

[15] Elizabeth Shriberg. Higher-Level Features in Speaker Recognition. *Speaker Classification 1*, pages 278–297, 2007.

[16] Stuart and Norvig. *Artificial Intelligence - A Modern Approach*. Pearson, 3 edition, 2010. ISBN 9780132071482.

[17] D.E. Sturim, W.M. Campbell, and D.A. Reynolds. Classification Methods for Speaker Recognition. *Speaker Classification 1*, pages 278–297, 2007.

Abbildungsverzeichnis

Tabellenverzeichnis