



Bachelorarbeit zum Thema

Audiovisuelle Datenfusion mittels Partikelfilter für HCI-orientierte Anwendungen

von Tobias Hanke

Prüfer:

Prof. Dr.-Ing. Klaus-Dietz Tönnies
Otto-von-Guericke Universität Magdeburg
Lehrstuhl für Bildverarbeitung / Bildverstehen
Institut für Simulation und Graphik (ISG)

Betreuer:

Jun.-Prof. Dr.-Ing. Ayoub Al-Hamadi
Otto-von-Guericke Universität Magdeburg
Institut für Elektronik, Signalverarbeitung und Kommunikationstechnik (IESK)

Hanke, Tobias

Studiengang: Computervisualistik

Otto-von-Guericke-Universität Magdeburg

Bachelorarbeit 2011

Audiovisuelle Datenfusion mittels Partikelfilter
für HCI-orientierte Anwendungen

Danksagung

An dieser Stelle möchte ich mich bei all jenen Menschen bedanken, welche mich über die Zeit meines Praktikums und der Ausarbeitung dieser Arbeit unterstützt haben.

Vorweg ein Danke an meinen Prüfer, Professor Tönnies, für die Betreuung meiner Arbeit und für die Geduld.

Weiterhin danke ich Junior-Professor Al-Hamadi, welcher mich bei meinem Praktikum betreute und mir viel Freiraum bei der Bearbeitung des Themas gab.

Zusätzlich möchte ich mich bei den anderen Mitarbeitern des Instituts, welche mir ein angenehmes Arbeitsklima bei meinem Praktikum vermittelten, bedanken. Besonderer Dank geht dabei an Michael Heuer für die Gespräche zu bestimmten Problematiken.

Natürlich möchte ich mich auch bei meiner Familie und meinen Freunden bedanken, welche mich moralisch unterstützten, besonders bedanke ich mich bei Viktor für das ausführliche Korrekturlesen der Arbeit.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Bachelorarbeit selbstständig, ohne unzulässige Hilfe und nur unter Verwendung der angegebenen Hilfsmittel angefertigt habe. Sämtliche genutzten Quellen und deren Einwirkung auf diese Arbeit sind im Literaturverzeichnis einzusehen.

Magdeburg, den 28.04.2011

Tobias Hanke

Kurzfassung

Heutzutage halten Computer immer mehr Einzug in das alltägliche Leben - in vielen Bereichen sind sie sogar nicht mehr wegzudenken. Umso wichtiger ist daher eine geeignete Kommunikationsmöglichkeit von Mensch zu Maschine. Die Entwicklung solcher Systeme, welche eine intuitive Human-Computer-Interaktion als Grundlage verwenden ist jedoch aufgrund vieler zu beachtender Faktoren nicht einfach.

Während meiner zwanzigwöchigen Praktikumszeit am Institut für Elektronik, Signalverarbeitung und Kommunikationstechnik war ich unter der Aufsicht von Junior.-Professor Ayoub Al-Hamadi damit beschäftigt, an der Entwicklung der Grundlage eines solchen HCI-orientierten Systems mitzuwirken. Mir oblag dabei die Aufgabe die Sensordaten einzulesen, zu verarbeiten und mittels audiovisueller Datenfusion die Person zu identifizieren, welche derzeit mit dem System interagiert.

Dabei sollten die folgenden Bedingungen erfüllt werden. Das System sollte in Echtzeit funktionieren, zunächst sollte nur auf die akustische Interaktion geachtet werden und die verwendeten Geräte sollten einfach gehalten bleiben.

Die vorliegende Bachelorarbeit beschreibt daher verschiedene Methoden, meine Vorgehensweise, die Ergebnisse, vergleichbare Projekte und liefert einen Ausblick auf weitere Möglichkeiten.

Inhaltsverzeichnis

<i>Kapitel</i>	<i>Seite</i>
Allgemeines	3
Inhaltsverzeichnis	6
Abbildungsverzeichnis	7
1. Einleitung	8
1.1 Motivation	8
1.2 Problemstellung	9
1.3 verwandte Arbeiten	10
2. Grundlagen	14
2.1 Datenfusion	14
2.2 Partikelfilter	16
2.3 Mensch-Computer-Interaktion	18
2.4 Geräte.	19
2.5 Programm-Bibliotheken	19
3. Konzept	21
3.1 Programmablauf	21
3.2 Video	22
3.3 Audio	31
3.4 Fusion	33
4. Implementierung	34
5. Evaluation	38
6. Ausblick und Zusammenfassung	41
7. Literaturverzeichnis	42

Abbildungsverzeichnis

1-1	Fusions-Grundskizze	9
1-2	Gesichtserkennung mit der 360-Grad-Kamera	10
1-3	Aufbau der Testumgebung	11
1-4	Aufbau des Raumes	12
1-5	Bildfolge	13
2-1	Datenfusions-Funktionalitäten	15
2-2	Fusionsebenen	15
2-3	Condensation Ablauf	17
3-1	schematischer Programmablauf	21
3-2	RGB-Farbraum	23
3-3	HSV-Farbraum	24
3-4	YCbCr-Farbraum	25
3-5	Hautfarbenverteilung in Cb und Cr	26
3-6	Pyramidenansatz für den LK-Algorithmus	27
3-7	Blink-Detektion	28
3-8	Haar-like-Features	29
3-9	Trainingsbeispiel	30
3-10	CamShift	31
3-11	Kreuzkorrelation	32
3-12	mögliche Position einer Geräuschquelle durch TDOA	32
4-1	Audio-Frames	34
4-2	Programmablauf	35
5-1	Kamerabild	38
5-2	Ada-Boost	39
5-3	Lucas-Kanade	39
5-4	Hautfarbenerkennung	40
5-5	Audioframes und Kreuzkorrelation	40

1. Einleitung

In diesem Kapitel beschreibe ich die Aufgabe, welche ich erfüllen sollte, die Motivation dahinter und gehe auf die Problemstellung und meine genauen Aufgaben ein.

Überblick:

- 1.1 Motivation
- 1.2 Problemstellung
- 1.3 verwandte Arbeiten

1.1 Motivation

Die Technologisierung schreitet immer mehr voran. In sämtlichen Bereichen des täglichen Lebens haben bereits Computer Einzug gefunden und dieser Trend setzt sich immer weiter fort. Die Kommunikation zwischen Mensch und Maschine gestaltet sich dabei jedoch nicht immer leicht, weswegen ein benutzerfreundlicher Umgang mit den allgegenwärtigen Computern sehr wünschenswert ist. Die Nutzung von Eingabegeräten, wie etwa Maus, Tastatur, Trackball oder Touchpad, erfordern oftmals Eingewöhnungszeit und verfügen nicht immer über die gewünschte Präzision. Des Weiteren verfügen solche Systeme auch nicht über Modifikationen, welche auf Emotionen des Nutzers zugreifen könnten.

Von Vorteil wäre es also, wenn man ohne Übertragungsmedium direkt mit dem Computer interagieren könnte. Ein wichtiger Bestandteil dieser Interaktion ist die Identifizierung der Person, welche dabei für den Computer relevant ist. In weiteren Schritten kann dann von dieser Person die Gestik, Mimik oder auch die Sprache genutzt werden um eine intuitivere Interaktion zwischen Mensch und Maschine zu realisieren.

Mögliche Anwendungsgebiete gibt es viele, etwa Steuerung des Kamerafokus für beispielsweise Videokonferenzen oder in Computern, welche auf Emotionen des Nutzers achten sollen, wie etwa Ratlosigkeit bei Servicerechnern oder Schmerz in medizinischen Anwendungsbereichen. Andere damit verwandte, mögliche Anwendungsgebiete sind etwa Überwachung und Personenidentifizierung.

Die für diese Thematik relevante Herangehensweise beschäftigt sich mit den visuellen und akustischen Merkmalen und deren Fusion. Probleme, die dabei zu bewältigen sind, wären Beleuchtungsaspekte, verschiedene Hautfarben, Störobjekte, Störgeräusche, Komplexität der Technik und Rechenaufwand.

Wenn diese Probleme erst einmal gelöst sind, könnte man dann ein System entwickeln, welches die Kommunikation zwischen Mensch und Maschine auf eine andere Ebene, als die übliche Steuerung per Maus oder Tastatur bringt und es so ermöglicht mittels Gestik und Mimik mit dem System zu interagieren.

1.2 Problemstellung

Mein Praktikum und diese Arbeit beschäftigen sich mit dem ersten Teil der Entwicklung eines HCI-basierten Systems, der Identifikation der interagierenden Person.

Zu diesem Zweck sollen die Daten von einer Webcam und zwei Mikrofone eingelesen werden. Die auf diese Weise gewonnenen visuellen und akustischen Daten sollen, nach der Verarbeitung, dann mittels Partikelfilter fusioniert werden, um - auch bei einer größeren Anzahl von Personen - die aktuell kommunizierende Person zu lokalisieren und zu tracken. Die Lokalisierung soll dabei möglichst nah an der Echtzeit erfolgen, sodass der Rechenaufwand minimal bleiben sollte. Dabei ist darauf zu achten, dass die verwendeten Verfahren robust sind und, dass die Möglichkeit gegeben ist, später weitere Modifikationen zu adaptieren, wie etwa Gestik-Erkennung oder die Verwendung einer 360-Grad-Kamera. Die Idee hinter dem Ganzen ist nicht neu. Es gibt zahlreicher Projekte, welche sich mit der Datenfusion von Audio- und Videodaten beschäftigen. Einige davon werde ich im nächsten Abschnitt kurz vorstellen.

Das Hauptproblem dieser Arbeit ist es jedoch mittels einfacher technischer Geräte (eine Webcam und 2 Mikrofone) die Daten zu gewinnen, diese sinnvoll zu verarbeiten und dann mittels Partikelfilter geeignet zu fusionieren.



Abbildung 1-1 - Fusions-Grundskizze

1.3 verwandte Arbeiten

Hier möchte ich kurz drei verwandte Projekte vorstellen, welche sich auch mit der Fusion von Audio- und Videodaten befassen.

Überblick:

1.3-A Smart Room: participant and speaker localization and identification

1.3-B multiple person and speaker activity tracking with a particle filter

1.3-C audio-visual speaker tracking with importance particle filters

1.3-A Smart Room: participant and speaker localization and identification

In diesem Paper geht es um die echtzeitfähige Lokalisation und Tracking der Personen und deren Sprachaktivitäten und einer ID-Zuweisung der einzelnen Personen um so über die Zeit deren Aktionen aufzeichnen zu können. Die Auslegung der Entwicklung ist dabei speziell für eine 360-Grad-Kamera also für einen vollen Rundum-Blick und ist für die Anwendung in Videokonferenzen oder Audio-Video-Indexzuweisungen gedacht. Das System nennt sich "Smart Room", womit die Raumdaten bei den Berechnungen zur Verfügung stehen. Der multimodale Aufbau beinhaltet dabei vier CCD-Kameras in den Ecken des Raumes, die besagte 360-Grad-Kamera in der Mitte und ein Mikrofon-Feld mit 16 Mikrofonen. Die Position der einzelnen Personen wird dabei über ein dreidimensionales Polygon-Model über die vier synchronisierten Kameras und eine Gesichtsdetektion über die Rundum-Blick-Kamera bestimmt. Die Daten werden dann mittels eines dynamischen Modells mithilfe einer Gaußverteilung fusioniert. Hinzu kommen die über das Mikrofon-Feld gewonnenen akustischen Merkmale, welche eine ID-Zuweisung der einzelnen Personen ermöglichen. Über die Fusion dieser ganzen Daten werden schließlich Position und Identität der Sprecher ermittelt.



Abbildung 1-2 - Gesichtserkennung mit der 360-Grad-Kamera - Quelle: [Bus05]

Die akustischen Merkmale werden dabei mittels Time Difference of Arrival-Technik (TDOA) und eine Phasen-Transformation analysiert. Um Vorder- und Hintergrund zu trennen, wird ein selbstständig lernendes Gauß-Modell verwendet. Die Personen-Lokalisierung wird dabei über die Änderungen im Gauß-Modell und eine Kantendetektion realisiert um so mittels der vier Kameras eine dreidimensionale Zuordnung vornehmen zu können. Die Gesichtserkennung der 360-Grad-Kamera wird dann mittels Haar-like-Features vorgenommen.

Das System erreichte bei ersten Tests in zwei Konferenzen eine Performanz von rund 70 Prozent, trotz der bekannten Umgebung und der Menge an multimodaler Technik. An diesem Beispiel kann man erkennen, dass es nicht so einfach ist die Daten geeignet zusammenzuführen um ein gutes Fusionsergebnis zu erhalten.

1.3-B multiple person and speaker activity tracking with a particle filter

Dieses Paper stellt ein weiteres System vor um Audio- und Videodaten zu kombinieren um verschiedene Personen in einer Umgebung mit mehreren Objekten zu verfolgen und deren Sprachaktivitäten zu analysieren.

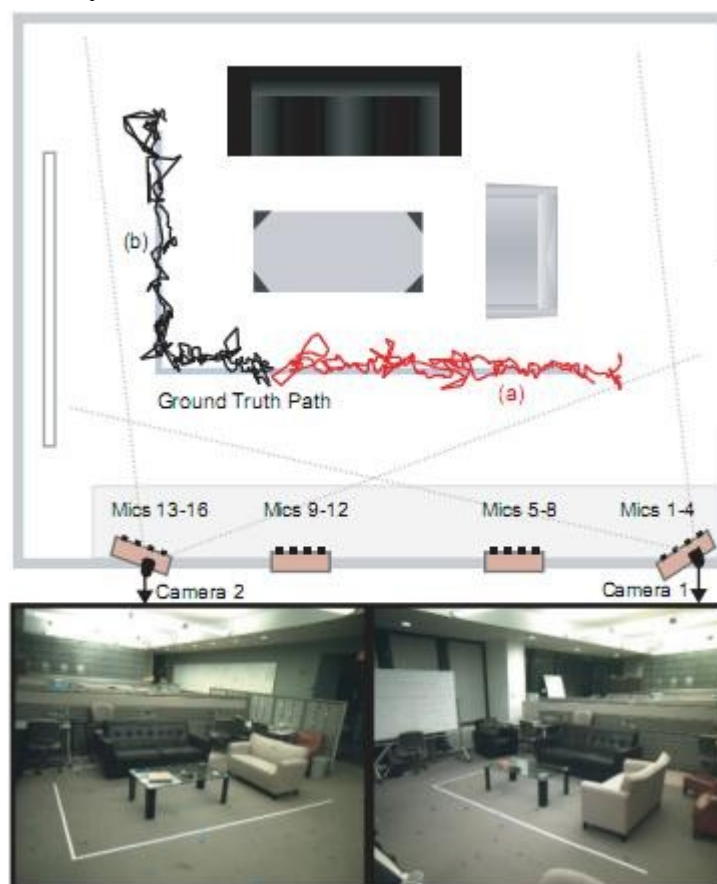


Abbildung 1-3 - Aufbau der Testumgebung - Quelle: [Che04]

Um die dabei entstehende nicht-gaußsche und nicht lineare Verteilung zu berücksichtigen, wurde ein Partikelfilter für die Fusion der Audio- und Videokomponenten verwendet. Als Ergebnis erhält man die Anzahl an Personen, ihre Position und deren Sprachaktivität über die Zeit.

Der multimodale Aufbau ist in Abbildung 1-3 ersichtlich und beinhaltet zwei Kameras in den Ecken und 16 Mikrofone. Die Daten jedes Objekts werden dabei zu jedem Zeitpunkt in einem Zustands-Vektor gespeichert und umfassen X- und Y-Position, sowie die Höhe und die Sprachaktivität zu diesem Zeitpunkt. Um Sprachaktivität zwischen den einzelnen Objekten zu erkennen, also Konversationen zu finden, wird der Gesamt-Sprach-Aktivitäts-Zustand für jedes Partikel berechnet und über eine Transitions-Matrix aktualisiert. Die Mikrofone sind zur Bestimmung der Soundquellen in lenkbaren Feldern angelegt. Die Audio-Signale werden dabei Fourier-transformiert und mittels Bandpass gefiltert. Für die Video-Beobachtung wird ein Hintergrundmodell und ein Bewegungsmodell verwendet, um so die Objekte zu lokalisieren.

Der in dem Paper beschriebene Lösungsansatz verwendet also Wissen über die Raumdaten, Informationen über Vorder- und Hintergrund, bewegliche Mikrofon-Felder und einen festen Bereich in denen sich Personen aufhalten können. Mit diesen Eigenschaften werden Vordergrunddetektion, Bewegungsanalyse über die Unterschiede der aufeinanderfolgenden Bilder und Analyse der Sound-Daten genutzt, um so Anzahl, Position und Sprachaktivität der Personen zu ermitteln. Im Testverlauf zeigte sich dabei eine Übereinstimmung der Daten von 80 Prozent.

1.3-C audio-visual speaker tracking with importance particle filters

Diese Arbeit befasst sich, wie die anderen beiden Beispiele auch, mit der Fusion von Audio- und Video-Daten, mit dem Ziel die sprechenden Personen zu lokalisieren und zu tracken. Im

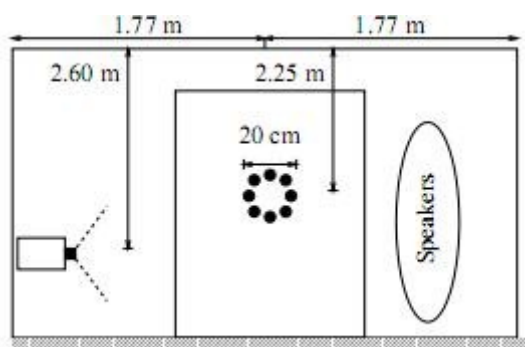


Abbildung 1-4 - Aufbau des Raumes - Quelle: [Gat03]

Gegensatz zu den anderen beiden Arbeiten wird hierbei jedoch nur eine Weitwinkel-Kamera verwendet. Für die Audio-Modalität wird ein Mikrofon-Feld mit 8 Mikrofonen verwendet. Dabei sind auch in dieser Arbeit wieder Entfernungen und Maße bekannt.

Der verwendete Algorithmus nutzt dabei einen Importance-Partikelfilter um zweidimensionale Objektformen und die erlangten Audio-Informationen zu fusionieren. Der Ansatz ist hierbei basierend auf den Bilddaten, dies heißt, dass zunächst über die Video-Informationen die entsprechenden Kandidaten gesucht werden und dann mittels Fusion mit den Audio-Daten zusätzliche Informationen extrahiert werden, um die sprechende Person zu finden. Im Gegensatz zu den anderen Beispielen ist die hier vorgestellte Lösung nicht in der Lage mehrere Sprecher gleichzeitig zu verfolgen, kann jedoch zwischen mehreren Sprechern wechseln.

Die Video-Merkmale werden dabei aus der Kontur gewonnen, um so der Kopfform zu folgen. Die dreidimensionalen Audio-Merkmale werden über den Zeitabstand der Signalkunft bei den Mikrofon-Paaren gebildet, wobei eine Kreuzkorrelation gebildet und das Ergebnis schließlich phasentransformiert wird, um ein robusteres Ergebnis zu erhalten. Anschließend werden diese Daten in den zweidimensionalen Raum gemappt und dort die informativsten Regionen ausgewählt.



Abbildung 1-5 - Bildfolge - Quelle: [Gat03]

2. Grundlagen

In diesem Kapitel gehe ich auf die Grundlagen, welche für das Verständnis der Arbeit wichtig sind, ein. Dies umfasst neben den Grundbegriffen auch die verwendeten Geräte und Programm-Bibliotheken.

Überblick:

- 2.1 Datenfusion
- 2.2 Partikelfilter
- 2.3 Human-Computer-Interaktion
- 2.4 Geräte
- 2.5 Programm-Bibliotheken

2.1 Datenfusion

Als Datenfusion bezeichnet man die Zusammenführung verschiedener Informationen und deren Übertragung in ein einheitliches, repräsentatives Format. Das Ziel der Datenfusion ist es Mehrdeutigkeiten zu beseitigen und damit genauerer Informationen zu erlangen.

Die in unserem Beispiel verwendeten Modalitäten sind Audiodaten (von zwei Mikrofone) und Videodaten (von einer Webcam).

Die Fusion hat drei verschiedene mögliche Funktionalitäten, eine komplementäre, bei der von den Sensoren eine umfangreichere Datensammlung erstellt wird, indem verschiedene Bereiche mit den Sensoren erfasst werden, sowie eine konkurrierende Funktionalität, mit welcher man sich genauer an reale Werte eines Sensorbereichs annähert. In der dritten Funktionalität, welche als kooperativ bezeichnet wird, wird eine vollständigere Datensammlung durch die gleichzeitige Nutzung der Eigenschaften der verschiedenen Modalitäten erstellt. Der Unterschied zwischen konkurrierender und kooperativer Arbeitsweise, liegt also in der Nutzung der Daten, dies bedeutet, ein konkurrierendes System nutzt die am besten geeigneten Daten oder eine Mittlung davon um Unsicherheiten zu beseitigen, während bei einem kooperativen System diese erst zusammenfasst werden müssen, um an den vollständigen Informationsgehalt zu gelangen. In der folgenden Beispielskizze wird dies nochmal anschaulich verdeutlicht.

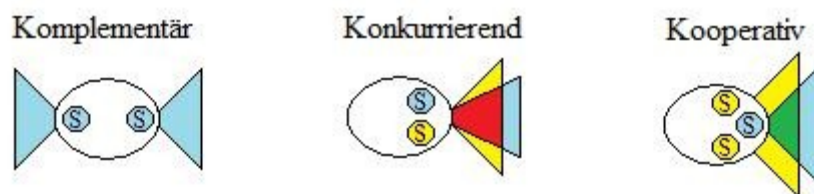


Abbildung 2-1 - DF-Funktionalitäten

Eine weitere Unterscheidung bei der Fusion liegt in der zeitlichen Abstimmung der Fusion, dabei unterscheidet man bei den Verfahrensweisen zwischen synchronisiertem und temporär registriertem Datenfluss. Der Unterschied liegt einfach darin, ob die Verarbeitung der Daten parallel erfolgt oder ob die Daten erst gesammelt werden und dann nach und nach verarbeitet werden.

Zuletzt unterscheidet man noch die Fusionsmethoden nach ihrer Koppelung, schwach gekoppelt bedeutet dabei, dass die Fusion die Sensoreinstellungen nicht beeinflusst und stark gekoppelt bedeutet eine Beeinflussung der Sensoreinstellungen, wie etwa eine Neusetzung des Kamerafokus.

Die Datenfusion findet auf verschiedenen Ebenen statt, dies sind die Signalebene, die Merkmalsebene und die Objektebene, dabei erfolgt die Fusion aber auch zwischen den Ebenen.

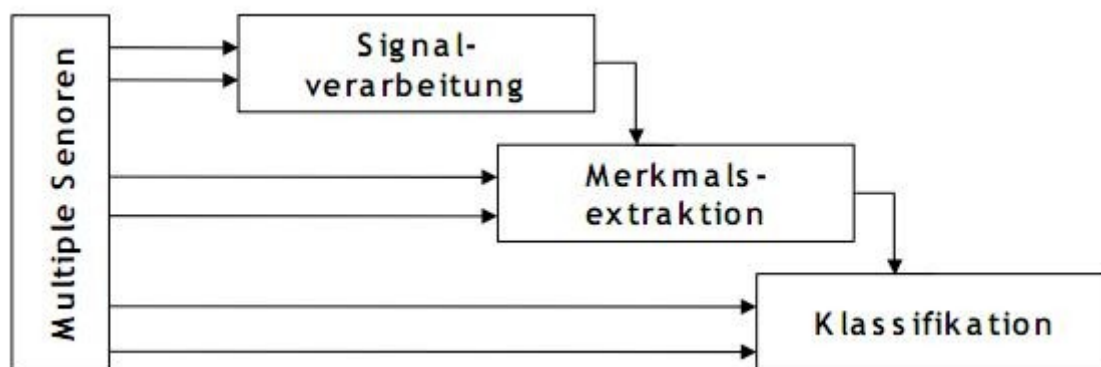


Abbildung 2-2 - Fusionsebenen

Die Datenfusion entsteht aus der Summe aller einzelnen Sensordaten mit der entsprechenden Gewichtung, welche noch normiert werden kann.

$$DF = w_1 * SD_1 + w_2 * SD_2 + \dots + w_x * SD_x = \sum w_a * SD_a$$

In dieser Arbeit wird die kooperative Funktionalität genutzt, um zunächst mittels Datenfusion der einzelnen Sensoren die Personen zu identifizieren und schließlich über die akustischen Merkmale zu selektieren. Weiterhin ist die Fusion schwach gekoppelt, da keine Anpassung der Sensorik vorgenommen werden und die gewählte Verfahrensweise entspricht im Gesamten betrachtet dem temporär registriertem Datenfluss, während die Fusion der Videodaten synchronisiert erfolgt.

2.2 Partikelfilter

Um bei der Datenfusion letztendlich ein Ergebnis zu bekommen, gibt es verschiedene Möglichkeiten die einzelnen Sensordaten zu gewichten. Man kann zum Beispiel eine einfache Mittlung vornehmen oder aber einen Filter verwenden. Ein Filter ist dabei eine Funktion, welche eine Eingangssignal mittels einer mathematischen Abbildung in ein Ausgabesignal überführt. In der Datenfusion hat dies das Ziel eine Zustandsschätzung von Objekten über die Zeit zu ermöglichen. Dabei folgt jeder Filter dem Ablauf:

Prädiktion → Messung → Korrektur

Prädiktion bezeichnet dabei die Grundannahme, welche der Filter trifft, auf welche der über die Messung berechnete Korrekturfaktor angewandt wird um so eine Schätzung über den eigentlichen Objektzustand machen zu können.

Die bekanntesten Filter sind dabei wohl der Kalman- und der Partikelfilter. Da der Kalman-Filter durch die zugrundeliegende Gauß-Verteilung für nicht-lineare Anwendungsfälle ungeeignet ist und er auch nicht in der Lage ist mehrere Objekte zu berücksichtigen, werde ich in dieser Arbeit nur auf den Partikelfilter näher eingehen.

Der Partikelfilter schätzt den Zustand eines Objekts mit Hilfe von einem Objektmodell und einem Bewegungsmodell. Durch diese beiden Modelle ergibt sich für die Schätzung eine Menge von möglichen Zuständen. Diese Zustände werden über Partikel repräsentiert und mithilfe des Beobachtungsmodells auf ihren Wahrheitsgehalt überprüft. Aus der Relevanz der Zustände zur Beobachtung ergibt sich dann die Gewichtung der einzelnen Partikel, über welche dann eine multimodale Wahrscheinlichkeitsverteilung aufgebaut wird mit deren Hilfe der Vorgang iterativ fortgesetzt werden kann. Dieses Verfahren ist nötig, da der über das Beobachtungsmodell abgelesene Zustand des Objekts zum Beispiel durch Rauschen, Verdeckung oder Messungenauigkeiten gestört sein kann.

Die drei verwendeten Modelle haben dabei die im folgenden erklärten Funktionen. Das Objektmodell beschreibt die spezifischen Eigenschaften des betrachteten Objekts, gibt also zum Beispiel Aufschluss über Farbe, Größe oder Form. Das Bewegungsmodell gibt Auskunft

über die mögliche Art und Stärke der Änderungen der Eigenschaften des Objekts, zum Beispiel also die Position oder andere Werte wie die Skalierung und Rotation. Diese beiden Modelle beschreiben also den geschätzten Zustand des Objekts, während das Beobachtungsmodell beschreibt, wie die Sensordaten interpretiert und somit nutzbar gemacht werden sollen, also eine Repräsentation des tatsächlichen Zustands ist.

Abbildung 2-3 zeigt den Condensation-Algorithmus, welcher eine spezielle Form des Partikelfilters implementiert. Dieser Partikelfilter arbeitet iterativ und mit einer Gewichtung der Partikel und wurde von Blake und Isard entwickelt. Anhand dieser Abbildung werde ich den groben Ablauf dieses Partikelfilters erläutern.

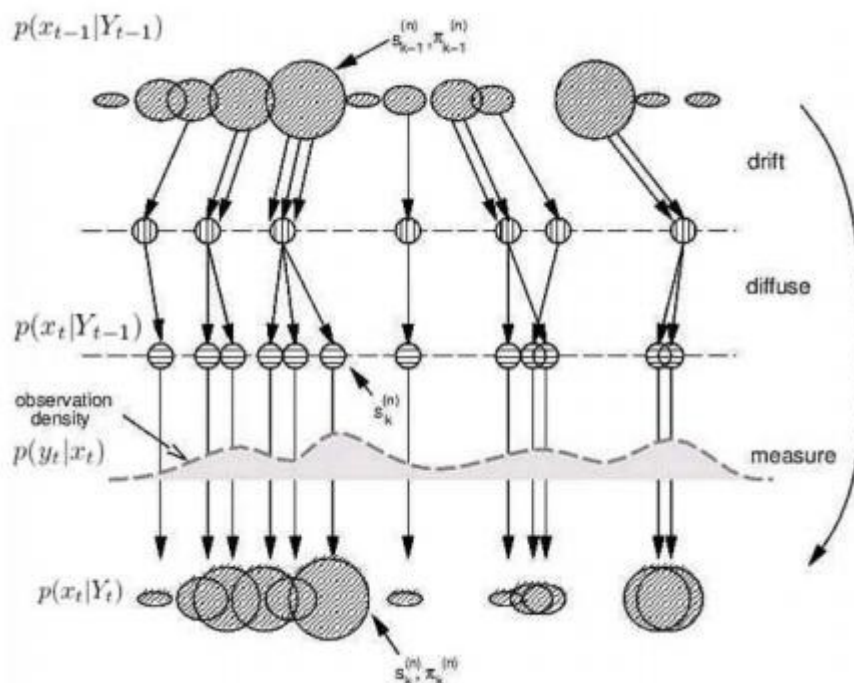


Abbildung 2-3 - Condensation Ablauf - Quelle: [Ost05]

Die Abbildung stellt einen Ablauf der drei Schritte des iterativen Prozesses dar, x steht hierbei für den Objektzustand, während Y für die Messung zum jeweiligen Zeitpunkt t steht. Im ersten Schritt werden die Partikel aus der alten Wahrscheinlichkeitsdichte $p(x_{t-1}|Y_{t-1})$ entsprechend ihres Gewichts ausgewählt, wobei Partikel mit hohen Gewichten bevorzugt und Partikel mit niedrigen Gewichten dabei eher vernachlässigt werden. Danach werden die ausgewählten Partikel entsprechend des Bewegungsmodells verschoben und mit Rauschen überlagert. Dieser Schritt dient also der Korrektur und Anpassung der Partikel. Im zweiten Schritt wird dann aus dieser Partikelmenge die Prädiktion $p(x_t|Y_{t-1})$ über den nächsten Zustand gewonnen. Der dritte Schritt nutzt dann die Daten aus den Messungen des Beobachtungsmodells und gewichtet damit die Partikel neu. Somit entsteht eine neue

Wahrscheinlichkeitsdichte $p(x_t|Y_t)$ für den nächsten Ablauf der Iteration. Da diese im Nachhinein gebildet wird, spricht man auch von einer a-posteriori Wahrscheinlichkeitsdichte.

2.3 Mensch-Computer-Interaktion

Die Mensch-Computer-Interaktion oder auch Human-Computer-Interaction (HCI) ist Bestandteil der Informatik und beschäftigt sich mit der Schnittstelle zwischen Mensch und Computer, dies umfasst sowohl die Informationswiedergabe vom Computer in einer gut nutzbaren Form, als auch die Interaktionsmöglichkeiten des Menschen, sowie eine Anpassung des Systems an den Nutzer. Es ist also ein umfangreiches Teilgebiet der Informatik und beschäftigt sich sowohl mit Hardware als auch mit der Software. Dabei werden Erkenntnisse aus der Psychologie (Wahrnehmung, Kognition), Ergonomie und Design (Gestaltungsprinzipien) verwendet.

Die Forschung in diesem Bereich ist wichtig, da die Computer immer mehr in unser Leben integriert werden, sei es nun an Arbeitsplätzen, daheim oder im Alltag an Servicerechnern. Eine gut umgesetzte HCI-Schnittstelle erlaubt dem Nutzer dabei einen schnell erlernbaren intuitiven Umgang, eine präzise Steuerung oder einfach eine schnelle Informationsbeschaffung. Durch eine gute Mensch-Computer-Schnittstelle ist es also möglich den komplexen menschlichen Verstand mit der Rechenleistung eines Computers effektiv zu verbinden.

Dazu kann es nötig sein, die Eingabemöglichkeiten zu überarbeiten. Maus und ähnliche Eingabegeräte benötigen beispielsweise erst eine gewisse Einarbeitungszeit - ein sprachlicher Befehl an den Computer wäre in manchen Fällen deutlich intuitiver, bringt aber viele Probleme mit sich, wie etwa das Filtern von Störsignalen oder unterschiedliche Interpretationsmöglichkeiten. Auch die Anpassung an den Nutzer wäre eine gute Möglichkeit um die Kommunikation zwischen Mensch und Maschine zu gewährleisten - etwa eine Unterscheidung zwischen einem versiertem und einem unerfahrenem Nutzer.

Für alle Interaktionsmöglichkeiten ohne direktes Eingabegerät muss dabei jedoch erst einmal der Nutzer erkannt werden und zu diesem Zweck bietet sich eine Fusion von Audio- und Videodaten an, um so ein System zu ermöglichen, welches über Tracking den Nutzer verfolgen kann oder auch um einfach ein multimodales Kommando zu erzeugen, wie etwa ein akustischer Befehl und ein Zeigen mit den Fingern.

In dieser Bachelorarbeit soll es aber in erster Linie um die Erkennung der interagierenden Person gehen für eine Grundlage eines Systems, welches in weiteren Schritten um beispielsweise Gestik- oder Mimikerkennung erweitert werden kann.

2.4 Geräte

Zur Durchführung der gestellten Aufgabe standen mir die folgenden Geräte zur Verfügung:

Computer

Der genutzte Computer war mit einem Intel Core 2 Quad CPU Q9550-Prozessor ausgerüstet, die 4 Prozessorkerne hatten dabei jeweils 2,83GHz und der PC verfügte über 3,25 GB RAM.

Webcam Pro 9000

Webcam von Logitech mit einer maximalen Auflösung von 1600x1200 und bis zu 30 Bildern pro Sekunde.



USB Desktop Microphone

2 Standmikrofone ebenfalls von Logitech mit einem Frequenzbereich von 100Hz-16kHz und einer Eingangsempfindlichkeit von -67 dBV/μBar.



2.5 Programm-Bibliotheken

An dieser Stelle möchte ich kurz die verwendeten Bibliotheken vorstellen und erklären was diese leisten können. In diesem Projekt wurden zwei Bibliotheken eingebunden, zum einen OpenCV für die Verarbeitung der Videodaten und zum anderen die Bass-Bibliothek für die Sound-Verarbeitung. Beide Bibliotheken sind kostenlos für den nicht-kommerziellen Gebrauch und OpenCV sogar für kommerziellen Nutzen.

Überblick:

2.5-A OpenCV

2.5-B Bass

2.5-A OpenCV



OpenCV steht für Open Source Computer Vision und ist eine freie Bibliothek für Real-Time Computer Vision-Anwendungen, welche von Willow Garage betreut wird. Mit dieser Bibliothek sind Bilddatenmanipulationen, das Auslesen und Wiedergeben von Bildern und Videos, Manipulation von Matrizen und Vektoren, Bildverarbeitungsalgorithmen, Struktur- und Bewegungsanalyse sowie die Erstellung eines einfachen grafischen Benutzerinterfaces möglich.

Anwendung findet diese Bibliothek bei vielen Projekten, was wohl vor allem an der guten Implementierbarkeit und den schnellen Algorithmen, sowie der Kostenfreiheit liegt. Spezielle Anwendungsgebiete sind dabei die Mensch-Maschinen Interaktion, die Segmentierung, die Objekterkennung und die Bewegungserkennung.

Da es zu lange dauern würde hier sämtliche Funktionen durchzugehen, soll hier nur kurz der grundlegende Aufbau der Funktionen in OpenCV erklärt werden um dies bei einer späteren Nennung besser nachvollziehen zu können. Eine Funktion sieht vom Grundaufbau so aus:

`cvAktionZielZusatz`

Das "cv" kennzeichnet die Funktion dabei als Bestandteil der OpenCV-Bibliothek. Die Aktion beschreibt was gemacht wird, wie beispielsweise create oder get. Der nächste Teil beschreibt dabei womit die Aktion ausgeführt wird, wie beispielsweise image oder contour. Der letzte Teil schließlich ist eine Angabe über zusätzliche Modifikatoren.

2.5-B Bass



Bass ist eine von Un4seen betreute Audio-Bibliothek, welche viele Funktionen beinhaltet, darunter abspielen und verwalten von verschiedenen Sound-Formaten. Für uns ist hierbei jedoch die Aufnahme-Funktion am wichtigsten um die Daten der beiden Mikrofone speichern und später anderweitig verarbeiten zu können.

3. Konzept

Das dritte Kapitel befasst sich mit dem zugrundeliegendem Konzept für die Programmierung, dies umfasst eine Erklärung für den allgemeinen Programmablauf, sowie Erläuterungen zu den möglichen Verfahren für die Videoauswertung, die Audiolokalisation und die Fusion dieser Daten.

Überblick:

- 3.1 Programmablauf
- 3.2 Video
- 3.3 Audio
- 3.4 Fusion

3.1 Programmablauf

Für das Programm sind fünf grobe Teilbereiche notwendig. Der erste befasst sich mit dem Einlesen und der Vorverarbeitung der Daten aus Kamera und den beiden Mikrofonen. Der zweite und dritte Schritt umfasst dann die Analyse dieser Daten, einmal für die Video- und einmal für die Audio-Daten. Darauf folgt die Fusion dieser beiden Daten, welche schließlich im fünften Schritt ausgegeben werden muss.

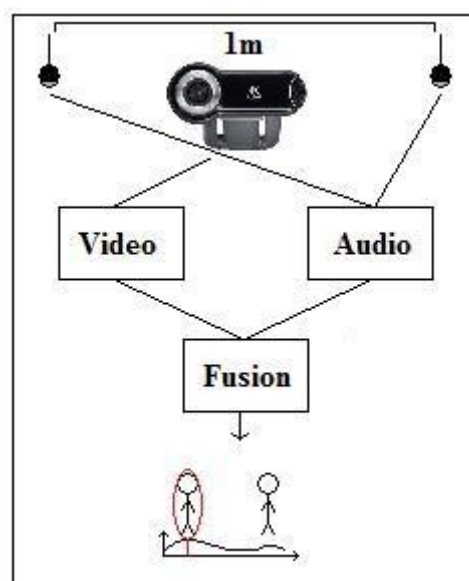


Abbildung 3-1 - schematischer Programmablauf

Diese Schritte sind in der Abbildung 3-1 noch einmal schematisch dargestellt. In den nächsten Unterkapiteln werde ich einige mögliche Verfahren vorstellen, mit welchen man das gewünschte Ergebnis im entsprechenden Schritt erreichen kann und werde diese auch genauer erläutern und auf ein paar von deren Vor- und Nachteile eingehen. Im nächsten Kapitel erkläre ich dann, welche Verfahren tatsächlich Anwendung gefunden haben und die Gründe dafür. Ein Problem bei der Aufgabenstellung ist es, dass keine genaueren Angaben spezifiziert wurden. Dies lässt auf der einen Seite viel Freiraum für verschiedene Methoden, setzt auf der anderen Seite jedoch auch eine gewisse Allgemeingültigkeit voraus. Wir wissen also im Vorfeld nichts über Abstände, Maße, Licht, Position, Umgebungslautstärke oder die Anzahl an Probanden. Auch ist dabei zu beachten, dass diese Arbeit ein Grundgerüst beschreibt, welches später um weitere Funktionalitäten erweitert werden kann.

3.2 Video

Videodaten haben einige Vorteile gegenüber der Audio-Modalität, so sind räumliche Auflösung und die Menge an Eigenschaften größer, dafür gibt es Probleme mit Überdeckungen und anderen Störfaktoren wie etwa die Beleuchtung. Im Zuge der Videoverarbeitung muss das Programm die Kameradaten einlesen, Personen, beziehungsweise Gesichter finden und diese verfolgen. Im Folgendem möchte ich auf ein paar der verschiedenen Methoden für die Gesichtsdetektion und für das Tracken von Objekten eingehen.

Überblick:

3.2.1 Personen-/Gesichtsdetektion

- 3.2.1-A Farbe

- 3.2.1-B Bewegung

- 3.2.1-C Kontur

- 3.2.1-D Merkmale

3.2.2 Objekttracking

3.2.3 Zusammenfassung

3.2.1 Personen-/Gesichtsdetektion

An dieser Stelle werde ich vier Grundformen vorstellen, mit welchen man über die Video-Auswertung die Position einer Person, beziehungsweise deren Gesicht bestimmen kann.

3.2.1-A Farbe

Ein sehr wichtiges Merkmal bei der Informationsgewinnung aus Bildern ist die Farbe. Eine Möglichkeit Personen in einem Bild zu lokalisieren ist also eine Segmentierung über die Hautfarbe. Im Idealfall ist dies die einfachste Möglichkeit um auf diese Weise Gesichter zu finden, leider gibt es dabei einige Probleme. Zunächst einmal gibt es eine Unmenge an Hautfarben, wodurch das Spektrum an Farben die segmentiert werden müssten gewaltig groß ist. Durch dieses große Spektrum fallen dann auch viele Objekte mit hautfarbenähnlicher Färbung in die Segmentierung, weiterhin wird kann ohne weitere Merkmale damit auch nicht zwischen dem Gesicht und sonstigen Körperregionen unterschieden werden. Weitere Probleme sind Verdeckung durch Kleidung, sowie Licht- und Schatteneinwirkung. Diese Probleme können zum Teil mit den verschiedenen Farbräumen abgemildert werden. Da wir sie später verwenden werden, möchte ich kurz die drei Farbräume RGB, HSV und YCbCr anhand von [Web04] vorstellen.

Der bekannteste Farbraum ist wohl der RGB-Farbraum. Er wird gerne genutzt, da er recht gut mit der menschlichen Farbwahrnehmung harmonisiert, leider hat dieser Farbraum auch ein paar Schwächen bei Bildern mit Inhalt aus der realen Welt, sodass andere Farbräume hinzugenommen werden müssen.

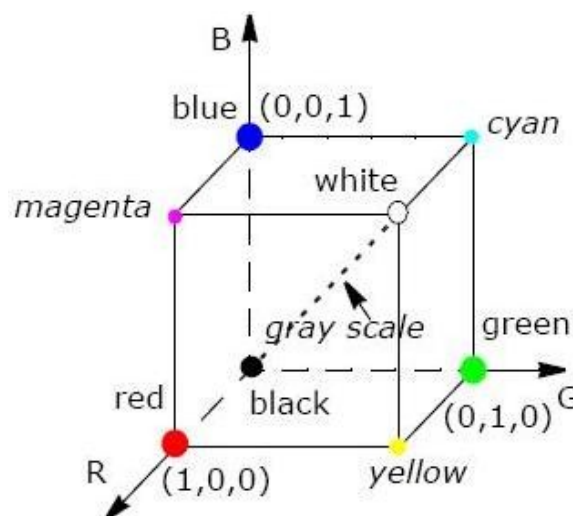


Abbildung 3-2 - RGB-Farbraum - Quelle: [Web04]

Abbildung 3-2 zeigt noch einmal den RGB-Farbraum. Die einzelnen Farben entstehen dabei aus der Mischung der drei Komponenten rot, grün und blau, man spricht daher auch von einem additivem Farbraum.

Ein weiterer bekannter Farbraum ist der HSV-Farbraum, dieser ist nah an der menschlichen Farbinterpretation und ermöglicht so einen intuitiven Umgang mit diesem Farbraum. Der H-Wert gibt dabei den Farbton (englisch hue) an, S steht für die Sättigung (englisch saturation) und V ist die Helligkeit (englisch value). In Abbildung 3-3 sieht man die Farbverteilung für diesen Farbraum.

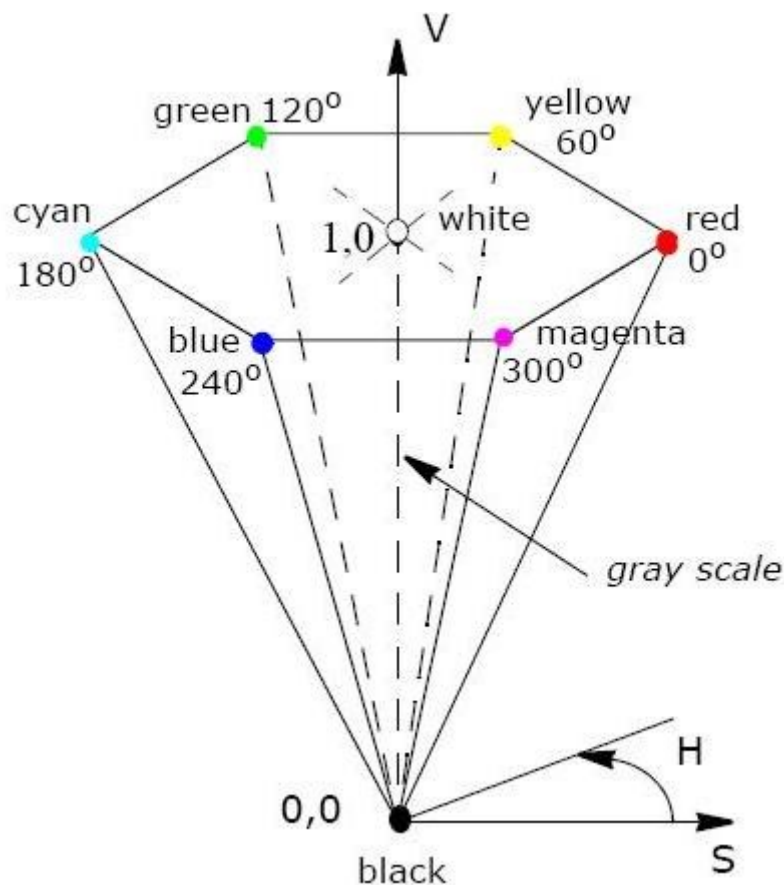


Abbildung 3-3 - HSV-Farbraum - Quelle: [Web04]

Der dritte Farbraum den ich vorstellen möchte, ist der YCbCr-Farbraum. Dieser ist ein technischer Farbraum und verwendet dabei die Helligkeit als ausschlaggebendes Farbmerkmal und wird im Y-Wert gespeichert. Die beiden anderen Werte stehen dabei für die Chrominanz und somit für die Verschiebung in Richtung blau (Cb), beziehungsweise rot (Cr). Die für unsere Versuche verwendete Umrechnung sieht wie folgt aus:

$$\begin{aligned}
 Y &= 0.299 * R + 0.587 * G + 0.114 * B \\
 Cb &= -0.16874 * R - 0.33126 * G + 0.5 * B + 128 \\
 Cr &= 0.5 * R - 0.41869 * G - 0.08131 * B + 128
 \end{aligned}$$

In der nachfolgenden Abbildung 3-4 kann man die Farbverteilung im YCbCr-Farbraum erkennen.

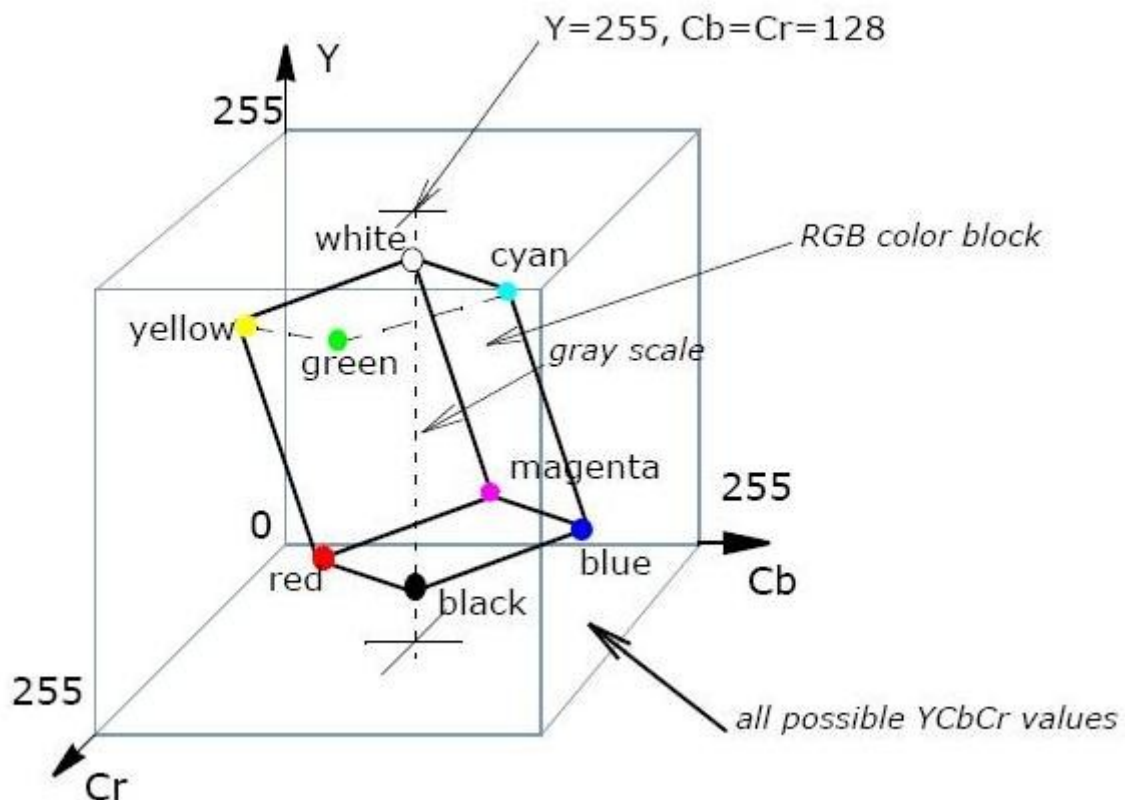


Abbildung 3-4 - YCbCr-Farbraum - Quelle: [Web04]

Diese drei Farbräume wurden in [Rah06] genutzt um eine gute Annäherung an die Hautfarbe zu erreichen. Die Bereiche in den verschiedenen Farbräumen, welche als Hautfarbe erkannt werden sollten, wurden dabei aus zehn Bildern ausgewertet, welche über eine große Variation an verschiedenen Völkern und Hautfarben-Typen. Die Ergebnisse sollen hier einmal kurz zusammengefasst werden. Im RGB-Farbraum wurde zwischen normalem Licht und starker Beleuchtung unterschieden, sodass eine der beiden folgenden Bedingungen zutreffen muss:

$$\begin{aligned}
 & (R > 95) \text{ AND } (G > 40) \text{ AND } (B > 20) \quad \text{AND} \\
 & (\max\{R, G, B\} - \min\{R, G, B\} > 15) \quad \text{AND} \\
 & (|R - G| > 15) \text{ AND } (R > G) \text{ AND } (R > B)
 \end{aligned}$$

oder unter Beleuchtung:

$$\begin{aligned}
 & (R > 220) \text{ AND } (G > 210) \text{ AND } (B > 170) \quad \text{AND} \\
 & (|R - G| \leq 15) \text{ AND } (R > B) \text{ AND } (G > B)
 \end{aligned}$$

Im HSV-Farbraum muss ebenfalls eine der beiden Bedingungen zutreffen, entweder:

$$H < 25$$

oder:

$$H > 230$$

Diese Aufsplittung kommt durch den Farbkreis zustande bei dem der Farbton nach Winkel eingeteilt wird und somit 360 der gleichen Farbe wie 0 entspricht.

Der YCbCr-Farbraum schließlich umfasst eine Reihe von Bedingungen um einen entsprechenden Bereich als Hautfarbe zu bestimmen.

$$\begin{aligned} Cr &\leq 1.5862 \times Cb + 20 && \text{AND} \\ Cr &\geq 0.3448 \times Cb + 76.2069 && \text{AND} \\ Cr &\geq -4.5652 \times Cb + 234.5652 && \text{AND} \\ Cr &\leq -1.15 \times Cb + 301.75 && \text{AND} \\ Cr &\leq -2.2857 \times Cb + 432.85 && \text{AND} \end{aligned}$$

In Abbildung 3-5 ist die Verteilung der Hautfarben im Cb- und Cr-Bereich, sowie die Abgrenzungen durch die Regelungen, gut ersichtlich.

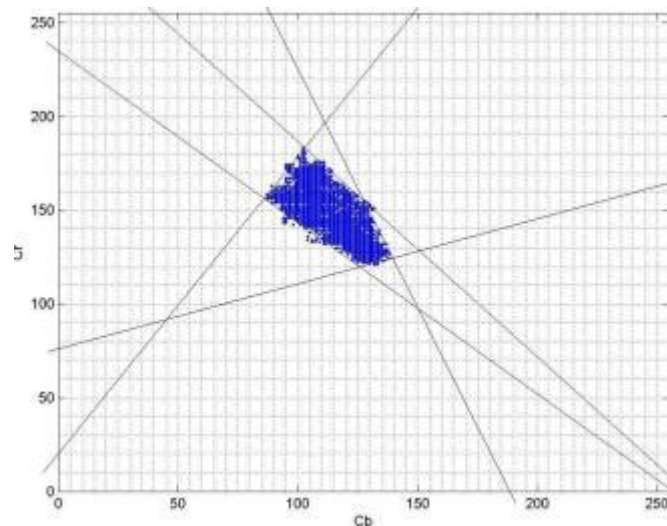


Abbildung 3-5 - Hautfarbenverteilung in Cb und Cr - Quelle: [Rah06]

In meinen Tests erwies sich diese Verteilung jedoch an manchen Stellen als lückenhaft und wurde daher auf der eben geschilderten Grundlage angepasst.

3.2.1-B Bewegung

Die nächste Möglichkeit, Personen oder Objekte in Videodaten zu finden, ist die Bewegungsanalyse, um so den statischen Hintergrund vom bewegten Vordergrund zu trennen. Die Bedingung hierfür ist eine statische Kamera und die Annahme, dass sich die gesuchten Objekte auch bewegen und möglichst keine Störbewegungen von anderen Objekten vorhanden sind. Da unter Bewegungslosigkeit dieses Verfahren nicht wirklich von Nutzen ist, verwendet man es eher in Kombination mit anderen Verfahren. Speziell in der Gesichtserkennung kommt bei der Bewegungsanalyse noch die Detektion der Augen über das

Blinzeln hinzu. Im Folgenden stelle ich kurz eines der sogenannten "Optical Flow"-Verfahren, die "Eye-Blink-Detektion" und die Hintergrund-Subtraktion vor.

Die Erklärungen zu dem Verfahren des optischen Flusses orientiert sich dabei an [BraKa] und umfasst die Erklärung zum Pyramiden-Lucas-Kanade-Algorithmus, welcher eine Erweiterung des Lucas-Kanade-Algorithmus ist. Weitere Verfahren sind die Horn-Schunck-Methode und das Block-Matching.

Der Lucas-Kanade-Algorithmus ist eine lokale Methodik, welche nur einen kleinen Bereich um eine Auswahl von Punkten absucht. Dabei werden Ecken- oder Kantenpunkte, welche sich gut wiederfinden lassen, genutzt, um die Position des Objekts im nächsten Bild zu bestimmen. Dieses Verfahren macht den Algorithmus zu einem schnellen und effizienten "Optical Flow"-Verfahren, führt jedoch dazu, dass große Bewegungen nicht erfasst werden. Um diesen Fehler zu beheben wird als Erweiterung das Pyramiden-System eingeführt, welches auf der einen Seite größere Bewegungen wahrnehmen kann und auf der anderen Seite immer noch schnell arbeitet. Für Lucas-Kanade sind drei Annahmen wichtig, erstens eine relativ konstante Helligkeit zwischen zwei Frames, zweitens zwischen den einzelnen Frames bewegen sich die Objekte nicht zu schnell um aus dem Suchfenster zu rutschen. Die dritte Bedingung ist räumliche Verwandtschaft, dies bedeutet, dass benachbarte Punkte einer Fläche eine ähnliche Bewegung haben. Um die zweite Bedingung zu umgehen, wird im pyramidischen Ansatz vom höchsten Pyramiden-Level mit den geringsten Details zu den niedrigeren Levels gewandert um so mehr Details zu haben.

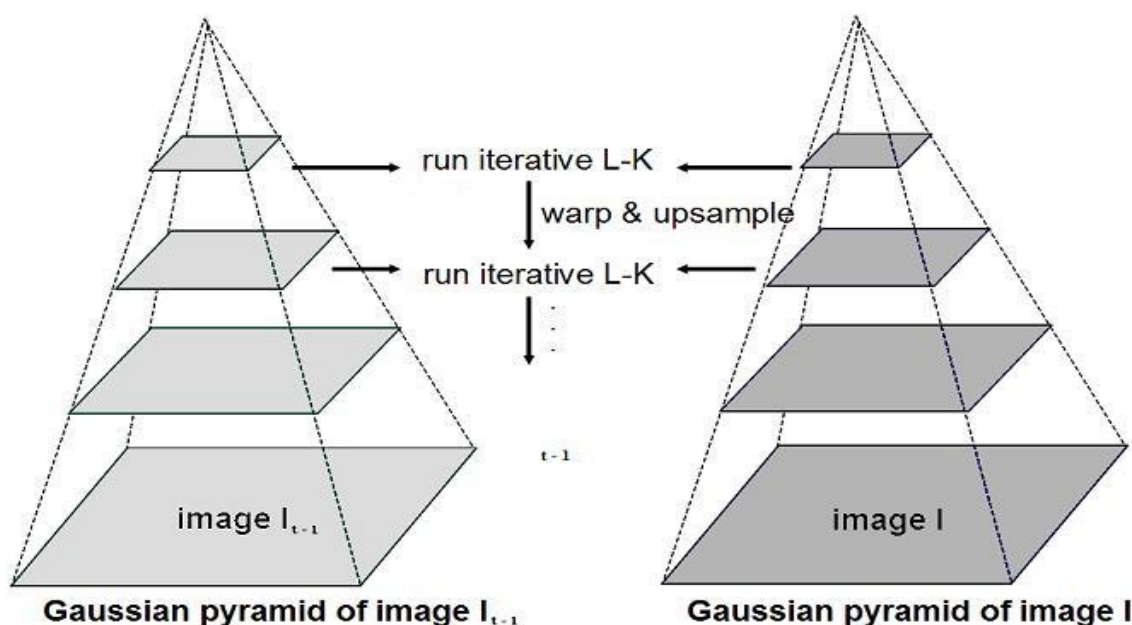


Abbildung 3-6 - Pyramidenansatz für den LK-Algorithmus - Quelle: [BraKa]

Eine einfachere Technik, welche jedoch auch störanfälliger ist, ist die Hintergrund-Subtraktion. Dieses Verfahren wird in [For03] beschrieben und basiert einfach auf der Subtraktion des Hintergrundbildes vom aktuell interessanten Bild. Das Hintergrundbild kann dabei zum einen einfach im Vorfeld aufgenommen werden, was jedoch keine sich über die Zeit leicht verändernden Hintergründe berücksichtigt, oder aber zum anderen die Subtraktion von 2 aufeinanderfolgenden Bildern, wodurch der Hintergrund in beiden Bildern identisch ist und somit verschwindet.

Dieses Verfahren kann man bei schneller Framerate auch verwenden um die in [Cha05] beschriebene Blink-Detektion zu realisieren. Mittels zweier aufeinanderfolgender Bilder werden bei einer geeigneten Bildwiederholungsrate jeweils die Differenzen gebildet. Dabei soll der Zeitabstand klein genug sein, dass sich in erster Linie nur das Blinzeln der Augen bemerkbar macht, oder es wird mit der ovalen Form ein Matching betrieben, um so die blinzelnenden Augen zu finden.

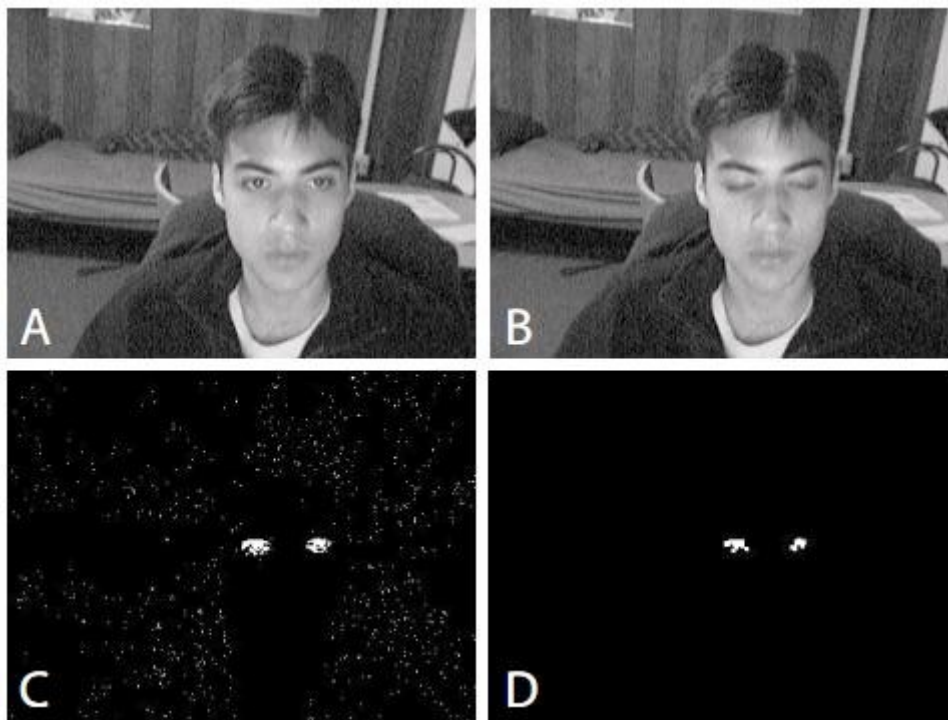


Abbildung 3-7 - Blink-Detektion - Quelle: [Cha05]

In Abbildung 3-7 ist der Vorgang gut dargestellt. Bei A und B sind die beiden aufeinanderfolgenden Bilder zu sehen. In Abbildung C erkennt man die Segmentierung über die Bewegung mit den Störungen und in Abbildung D wurden diese Störungen größtenteils durch eine Filteroperation entfernt. Nach der Detektion des Blinzeln, kann man die Augen dann tracken und somit die Position des Gesichtes bestimmen.

3.2.1-C Kontur

Die Kontur eines Objektes ist eine weitere Möglichkeit die Video-Daten zu analysieren. Dabei bietet es sich an, dies mit anderen Verfahren zu kombinieren. Ein Beispiel wäre etwa die Kombination von Farbinformationen und der Suche nach einer ovalen, passenden Form, um so die wahrscheinliche Ausrichtung des Gesichts zu erhalten, wie in [Aya03]. Auch kann eine Kontur über verschiedene Kantenfilter, wie etwa Sobel-Filter oder Canny-Edge-Filter gefunden werden, als Beispiel sei hier [Phu03] aufgeführt. Auch Active-Shape-Models wie in [Kosch] können leicht angepasst verwendet werden um die Form des Objektes zu finden. All diese Methoden eignen sich auch gut um die entsprechenden Objekte zu verfolgen und so findet man die Kontur-Eigenschaft auch oft in Tracking-Algorithmen wieder.

3.2.1-D Merkmale

Neben der Kontur kann man auch mittels der Merkmale die Objekte im Bild lokalisieren. Ein Beispiel für ein solches Verfahren ist die Principal Components Analysis (PCA), welche die Hauptkomponenten analysiert, vergleiche auch [Men99]. Eine andere Möglichkeit ist das Ada-Boost-Verfahren [Vio04]. Da ich dieses Verfahren verwendet habe, möchte ich es kurz näher erklären.

Bei dem Ada-Boost-Verfahren (Adaptive Boosting) handelt es sich um ein Lernverfahren, welches mittels kleiner Klassifikatoren durch deren Zusammenschluss einen großen Klassifikator erzeugt. Die kleinen verwendeten Klassifikatoren heißen Haar-like-Features und sind kleine Kanten-Merkmale, wie in Abbildung 3-8 zu sehen.

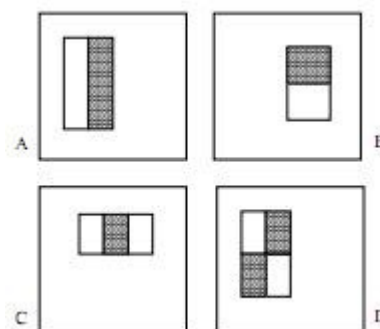


Abbildung 3-8 - Haar-like-Features - Quelle: [Vio04]

Dabei werden zunächst starke Indikatoren verwendet, um schnell einen Großteil der falschen Bereiche auszuschließen. Je nach Trainingsmenge variiert das Ergebnis der Identifikation.

Bei einer ausreichend großen Trainingsmenge können so auch Objekte in einer gewissen Rotation und sogar deren Seitenprofile erkannt werden.

In Abbildung 3-9 erkennt man die Anwendung der kleinen Klassifikatoren auf dem Bild. Die beiden ausgewählten Merkmale sind in der oberen Zeile zu erkennen und überlagern dann in der Zeile darunter das Bild des Trainings-Gesicht. Das erste Merkmal findet die Kante an der die Intensität sich ändert zwischen Augen und etwas über den Wangen. Das zweite Merkmal wird zwischen den beiden Augen in der Region der Nasenbrücke fündig. Durch die Kaskade dieser und weiterer Features und der Verwendung eines Integralbilds wird eine schnelle Lokalisation in späteren Anwendungsfällen ermöglicht, welche zudem skalierungsunabhängig ist.

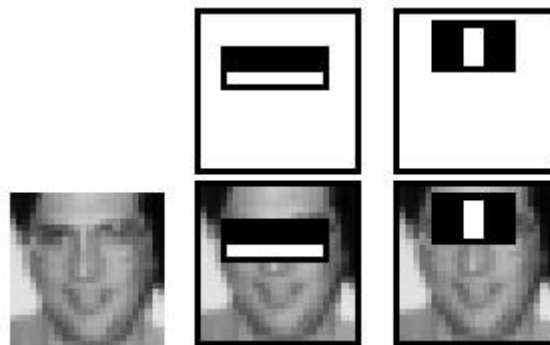


Abbildung 3-9 - Trainingsbeispiel - Quelle: [Vio04]

3.2.2 Objekttracking

In diesem Teil der Bachelorarbeit möchte ich auf die Methoden der Verfolgung von gefundenen Objekten eingehen. Zunächst erläutere ich die Unterscheidungsmöglichkeiten zwischen den Tracking-Verfahren. Es gibt vier verschiedene Kategorien, diese sind: regionenbasierte, merkmalsbasierte, modellbasierte und konturbasierte Verfahren, siehe [Hu04]. Zu den konturbasierten Trackingverfahren wurde bereits im Vorfeld schon etwas geschrieben, auch die regionsbasierten Ansätze wurden über die Bewegungsanalyse schon im Grundsatz erklärt. Merkmalsbasierte Verfahren verfolgen nicht das Objekt, sondern einzelne Merkmale, welche erst in folgenden Schritten wieder zu dem Objekt zusammengefasst werden. Sobald über ein mittels Vorwissen geschaffenes Modell versucht wird einem Objekt zu folgen, spricht man von einem modellbasierten Verfahren. Ein mögliches Beispiel dafür ist die Verfolgung von Farbinformationen mit Hilfe des Partikelfilters, wobei die Partikel wieder über ein Bewegungsmodell gesteuert werden und somit das Bild absuchen.

Ein weiteres Verfahren, welches ich kurz erläutern möchte, ist das CamShift-Verfahren, siehe [Web05]. CamShift besteht aus vier Schritten: als erstes wird ein Farb-Histogramm erstellt, welches das Gesicht repräsentiert. Im nächsten Schritt wird für jeden Pixel der folgenden Videobilder die Wahrscheinlichkeit berechnet, dass dieser Pixel zu dem Gesicht gehört. Das Lokalisierungs-Rechteck wird entsprechend in jedem Video-Bild bewegt und im vierten Schritt wird Größe und Winkel berechnet.

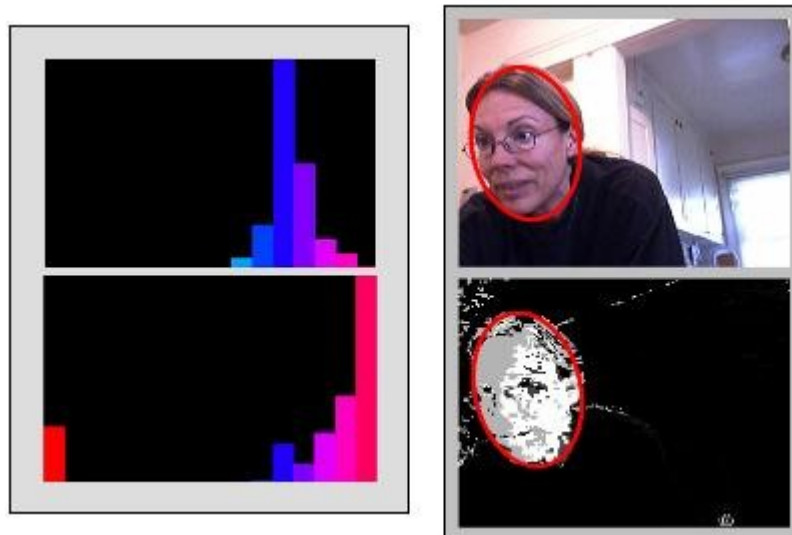


Abbildung 3-10 - CamShift - Quelle: [Web05]

3.2.3 Zusammenfassung

Ich habe nun einige Verfahren zur Personen-/Gesichtsdetektion und zur Verfolgung dieser ermittelten Objekte vorgestellt. Wobei mir die Koppelung von verschiedenen Herangehensweisen am sinnvollsten erschien, um so die Vorteile der verschiedenen Verfahren nutzen zu können. Allgemein bieten sich viele der Methoden zur Kombination untereinander an und auch Tracking und Lokalisation geht recht gut ineinander über, zum Beispiel bei den formbasierten Verfahren. Welche Verfahren ich bei der Programmierung verwendet habe und wie diese implementiert wurden, wird in Kapitel 4 erläutert.

Zusätzlich zur Lokalisation möchte ich an dieser Stelle noch auf die Möglichkeiten der Rauschbeseitigung in den segmentierten Bildern eingehen, also das Entfernen von kleinen Fehldetektionen. Zwei Methoden bieten sich hierfür an, zum einen die Blob-Detektion (Binary Large Object) an, welche über verbundene Komponenten arbeitet und deren Größe misst, zum anderen eine einfache Morphologie, mit welcher die segmentierten Bereiche ausgedehnt oder abgetragen werden.

3.3 Audio

Für die Audioverarbeitung müssen die Mikrofon-Daten getrennt eingelesen werden, weiterhin müssen Störungen beseitigt, die Richtung der Geräuschquelle lokalisiert und danach auf die Kameradaten gemappt werden.

Um das Rauschen zu verringern, kann zunächst das signal-to-noise Verhältnis verbessert werden, um so schwache Audiosignale, welche kaum Daten, dafür aber mit hoher Wahrscheinlichkeit Störungen beinhalten, zu löschen. Dann im Folgeschritt könnte man die verbliebenen Werte normalisieren und anschließend über eine Kreuzkorrelation auswerten.

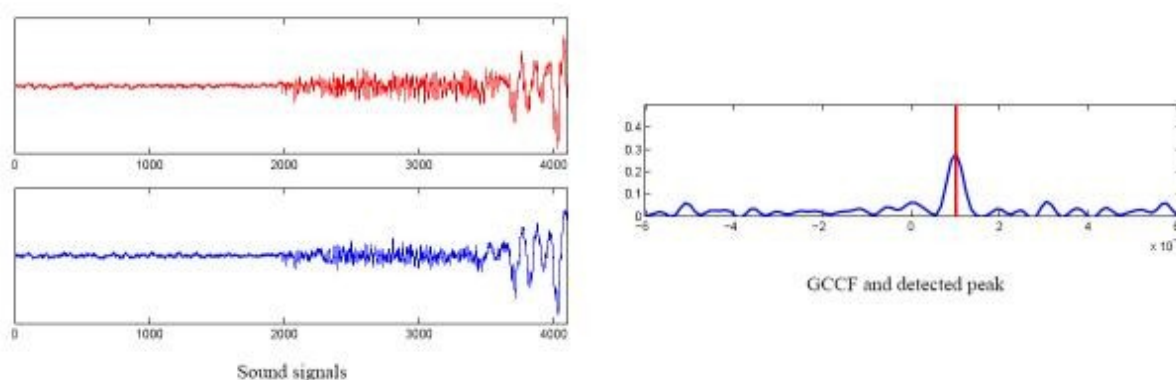


Abbildung 3-11 - Kreuzkorrelation - Quelle: [Gen05]

Abbildung 3-11 zeigt die beiden Eingangssignale und die daraus resultierende Kreuzkorrelation. Da sowohl die Daten von Mikrofon eins, als auch die Daten von Mikrofon zwei in etwa die gleiche Form haben, entsteht an ihrem Treffpunkt ein Maximum.

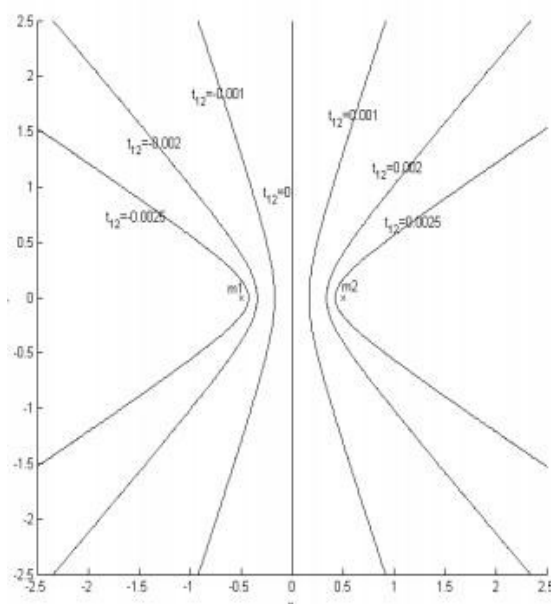


Abbildung 3-12 - mögliche Position einer Geräuschquelle durch TDOA - Quelle: [Gen05]

Um das Time-difference of arrival (TDOA) zu berechnen, muss nun einfach anhand des Maximums die Verzögerung aus dem Graphen abgelesen werden. Mit Hilfe einer Phasen-Transformation kann das Signal dabei robuster gestaltet werden. Leider kann aus den TDOA-Daten nur der X-Wert ausgelesen werden, weder die Höhe noch die Entfernung. In Abbildung 3-12 ist gut zu erkennen, welche möglichen Orte eine Geräuschquelle haben kann, wenn es an der entsprechenden Stelle von zwei Mikrofonen geortet wird, unter [Gus03] wird der ganze Vorgang nochmal ausführlich erklärt.

Eine weitere Möglichkeit die Richtung abzuschätzen ist über die Lautstärke, beziehungsweise über das Verhältnis der Lautstärken an beiden Mikrofonen.

3.4 Fusion

Im Fusionsschritt sollen schließlich die Video- und Audio-Daten zusammengeführt werden. Die Videodaten sollen dabei zur Bestimmung interessanter Orte dienen, während mittels der Audiobeobachtungen danach bestimmt werden soll, welche der beobachteten Regionen die interessanteste ist, also die Region mit dem höchsten Geräuschaufkommen.

Die Regulierung erfolgt dabei über die Vergabe der Gewichte, hinzu kommt eine Filterung der Daten um Rauschen zu verringern. Weiterhin müssen die Daten aneinander angepasst werden um auch gemeinsam verwendet werden zu können.

4. Implementierung

In diesem Teil der Arbeit erkläre ich bestimmte Parametereinstellungen, gehe darauf ein welche Methoden ich verwendet habe, wie ich diese kombiniert habe und welche Probleme dabei aufgetreten sind.

In der Abbildung 4-2 kann man den Gesamtaufbau des Programms erkennen. Da mir ein Computer mit vier Prozessorkernen zur Verfügung stand, habe ich das Programm auf einem Multi-Thread-Ansatz aufgebaut. Multi-Threading bedeutet, dass jeder Prozessorkern für sich versucht eine Aufgabe zu bewältigen und ermöglicht auf diese Weise paralleles Arbeiten, wodurch zum einen die Laufzeit verbessert wird und zum anderen wird dadurch in dieser speziellen Aufgabenstellung eine kontinuierliche Datensammlung von Audio- und Videodaten gewährleistet, welche für die Funktionalität der Audio-Lokalisation unerlässlich ist. Die Realisierung der Threads wurde mit Event-Handlern vorgenommen.

In der Initialisierungsphase werden die notwendigen Variablen deklariert und sämtliche Vorbereitungen für die folgenden Funktionsaufrufe getroffen. Daraufhin werden die Hauptthreads aufgerufen. Diese Threads verwalten die Audio- und Videoverarbeitung. Dabei ist ein Thread für die Audio-Verarbeitung, ein Thread für die Videoverarbeitung und ein Thread für die Videodaten-Gewinnung, sowie die spätere Ausgabe verantwortlich.

Der Audio-Thread ruft die Audio-Funktionen auf und gibt die Ergebnisse an den Partikelfilter weiter. Die aufgerufenen Audio-Funktionen lesen zum einen den Audio-Datenstrom ein, filtern diesen um das SNR zu verbessern, führen eine Lokalisierung der Audiodaten über eine normierte Kreuzkorrelation durch und wenden darauf den TDOA-Algorithmus an. Zusätzlich wird noch über die Lautstärke die Intensitätsverteilung als zusätzliches Ortungs-Merkmal verwendet.

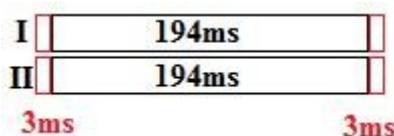


Abbildung 4-1 - Audio-Frames

Abbildung 4-1 zeigt den Aufbau der Audio-Frames. Jedes Audio-Frame ist 200ms lang, verfügt über eine Sample-Rate von 50 je ms, also über 10000 Samples. Dabei gibt es jeweils ein Audio-Frame für jedes Mikrofon. Für die Kreuzkorrelation wird dann das 194ms lange Audio-Frame über das komplette andere Audio-Frame geschoben und danach nochmal gewechselt um so keine Audioinformationen zu verpassen. Die jeweils drei Millisekunden auf jeder Seite, welche als Überschneidung dienen, sind für die Verschiebung da. Die beiden Mikrofone befinden sich in einem Abstand von etwas weniger als einem Meter. Die Geschwindigkeit von Schall bei normalen Bedingungen beträgt 0,343m/ms, sodass für die gesamte Strecke drei Millisekunden gebraucht werden, was den Seitenstreifen entspricht.

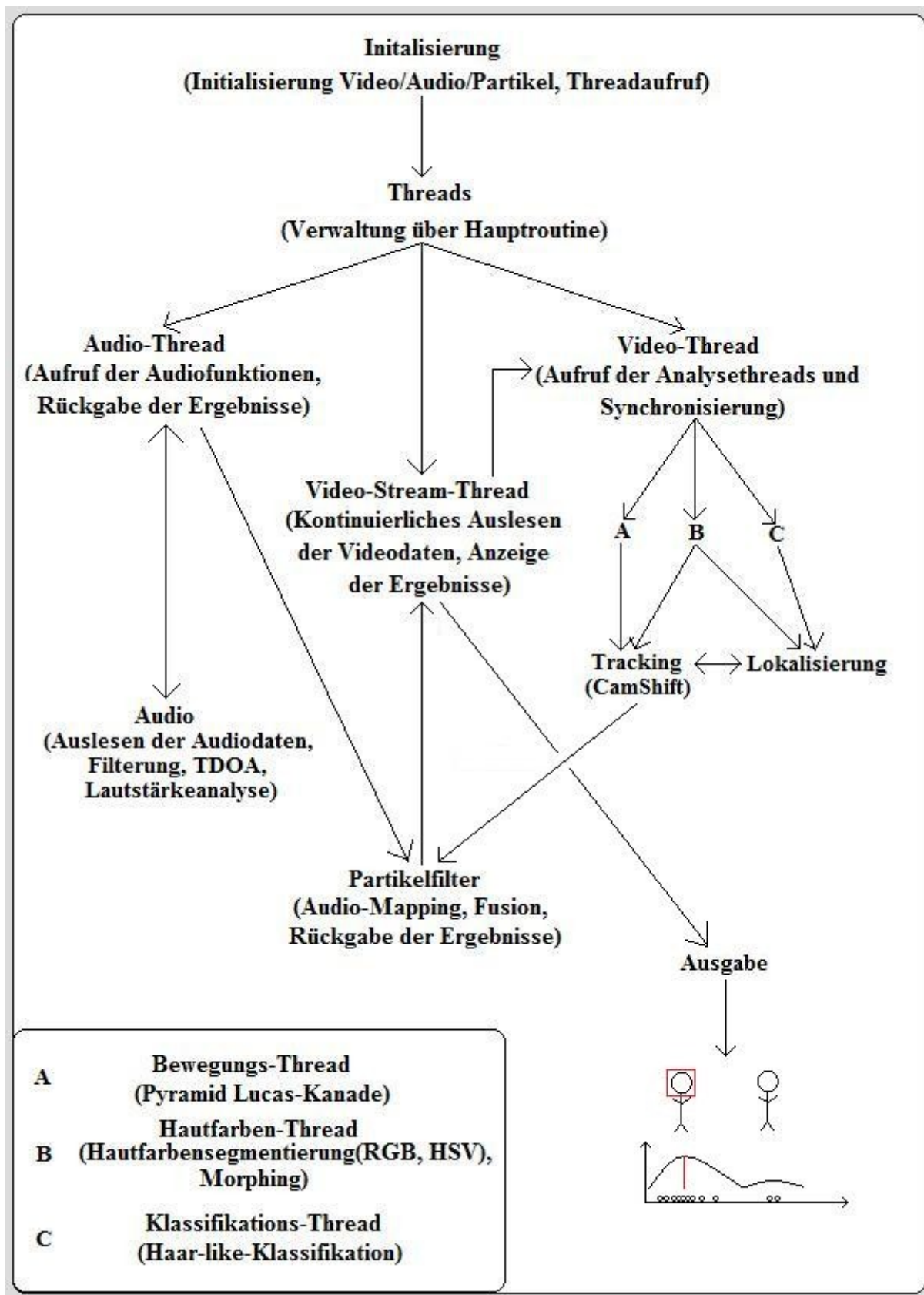


Abbildung 4-2 - Programmablauf

Nach der Korrelation wird der gewonnene Zeitunterschied zwischen den beiden Mikrofonen auf die Kamerakoordinaten gemappt und als Information an den Partikelfilter weitergegeben. Zusätzlich zu dieser Information erhält der Partikelfilter noch die Relation der Lautstärke bei beiden Mikrofonen als weiteres Gewichtung-Merkmal.

Die Videoanalyse erfolgt auf der Grundlage von Farbe, Bewegung und Haar-like-Features. Diese drei Funktionen werden jeweils auch in separaten Threads aufgerufen und synchronisiert. Die Videobilder liegen dabei in der Auflösung 640x480 vor. Für die Ada-Boost-Klassifikation mittels Haar-like-Features wurde der Klassifizierer

haarcascade_frontalface_default.xml

verwendet und ist in der Lage zuverlässig auch leicht zur Seite gedrehte Gesichter zu erkennen. Als zusätzliches Merkmal für den Partikelfilter wurde die Hautfarbe gewählt. Diese wird hierbei über den RGB- und HSV-Farbraum definiert. Der ursprünglich geplante YCrCb-Farbraum erwies sich nach einer Neujustierung der Werte als überflüssig und wurde daher entfernt, sodass eine einfache Und-Verknüpfung zwischen der RGB und HSV-Segmentierung ausgeführt werden muss.

Die von mir dabei mittels Tests über verschiedene Bilder verwendeten Werte sind für den RGB-Farbraum:

normale Bedingungen:

$(r > 90) \& (g > 40) \& (b > 20) \& ((\max(r, b) - (\min(b, g))) > 10) \& ((r - g) > 10)$ OR

starke Beleuchtung:

$(r > 220) \& (g > 210) \& (b > 170) \& (\text{abs}(r - g) \leq 15) \& (r > b) \& (g > b)$ OR

Schatten oder dunkle Hautfarben:

$(r > 25) \& (r < 90) \& (g > 20) \& (g < 60) \& (b > 10) \& (b < 60) \& (r > g) \& (r > b)$

und für den HSV-Farbraum:

normale Bedingungen:

$(h < 32)$ OR

$(h > 280) \& \& (v > 25)$ OR

Schatten oder dunkle Hautfarbe:

$(h \geq 32) \& \& (h < 40) \& \& ((s - v) < 10) \& \& (s < 70)$

dem ganzen geht noch ein Filter voran, um die Entscheidung schneller zu machen und falsche Kandidaten auszuschließen, folgende Bedingungen führen also zu einer Negativ-Segmentierung:

$(s < 1)$ OR

$(s < 30) \& \& (v < 30)$ OR

$(s > 70) \& \& (v > 70)$

Die Hautfarbe wird damit in den meisten Fällen gut erkannt, nur wenige Hautfarben verursachen Probleme bei der Segmentierung. Ein großes Problem ist jedoch sehr starker Tageslichteinfall.

Um die Objekte nicht nur zu Lokalisieren sondern auch verfolgen zu können, soll der

CamShift-Algorithmus verwendet werden. Dafür wird für die Farbhistogramm-Erstellung die Hautfarbensegmentierung verwendet, welche durch eine Morphologie angepasst ist um zum einen Störfaktoren zu beseitigen und zum anderen um geschlossene Gesichtsflächen zu erhalten, und als zusätzliche Hilfe die über den Pyramiden Lucas-Kanade-Algorithmus gewonnenen Bewegungsinformationen.

Dies alles wird dann im Partikelfilter gewichtet und ausgewertet. Dabei erhalten Video-Beobachtungen das fünffache Gewicht, während Audio-Beobachtungen mit dem Gewicht von eins arbeiten. Auf diese Weise sind die Video-Informationen wichtiger und der Partikelfilter orientiert sich eher an den gefundenen Personen und richtet sich dann erst nach der Audio-Information aus um die letztendliche Entscheidung zu treffen.

Nachdem der Partikelfilter die Entscheidung gefunden hat, gibt er sein Ergebnis an die Ausgabe weiter.

Probleme bei der Ausführung macht vor allem die Audio-Verarbeitung. Störgeräusche und eine zu ungenaue Bestimmung der Position durch nur die beiden Mikrofone, sorgen oft für Fehldetektionen.

5. Evaluation

Dieser Teil der Bachelorarbeit ist dafür da, die Ergebnisse auszuwerten, ein paar Beispielbilder zu zeigen und die Performanz zu bewerten und den Vergleich zwischen den Zielen und dem Erreichten, sowie den Problemen, welche bei der Bearbeitung der Aufgabenstellung aufgetreten sind.

Die Aufgabenstellung umfasste, ein Grundgerüst für weitere Bemühungen in die Richtung von HCI-Systemen zu schaffen und dies mit einfachen Modalitäten ohne Vorkenntnisse oder besondere Voraussetzungen, was diese Aufgabe sehr anspruchsvoll machte. Schon in den einführenden Beispielen hat man gesehen, dass auch mit deutlich mehr Modalitäten nur eine Performanz von maximal 80 Prozent erreicht wurde. Aufgrund der geringen Modalität-Anzahl mussten zahlreiche Funktionen für die Genauigkeit verwendet werden. Somit kommt das System beim derzeitigen Stand nur auf eine Performanz von 5 Auswertungen/Sekunde, ohne Multithreading wären es sogar nur 2 Auswertungen/Sekunde. Der Vorsatz die Umsetzung in Echtzeit zu schaffen ist damit nicht erfüllt, wobei es wahrscheinlich sowieso schwer ist, diesen mit dieser einfachen Technik zu erfüllen ohne dabei auf Genauigkeit zu verzichten. Denn schon in diesem Ansatz produziert die Audio-Auswertung genug Fehler. Hinzu kommt noch das Problem mit starkem Lichteinfall. Alles in allem zeigt das System aber gute Ansätze und ist durchaus ausbaufähig und mit mehr Vorbedingungen sicherlich auch noch besser realisierbar. Ein allgemein gültiges System wäre mit dieser Technik sowieso nicht möglich. Durch die interessanten Ansätze ist es als Grundgerüst für weitere Entwicklungen jedoch zu gebrauchen und erfüllt somit die gestellte Aufgabenstellung, soweit dies möglich ist.

Auf den folgenden Seiten sind für ein Beispiel die Ergebnisbilder zu sehen.



Abbildung 5-1 - Kamerabild



Abbildung 5-2 - Ada-Boost



Abbildung 5-3 - Lucas-Kanade



Abbildung 5-4 - Hautfarbenerkennung -
 A: RGB-Segmentierung, B: HSV-Segmentierung, C: Segmentierungsbild, D: Regionen nach Morphologie

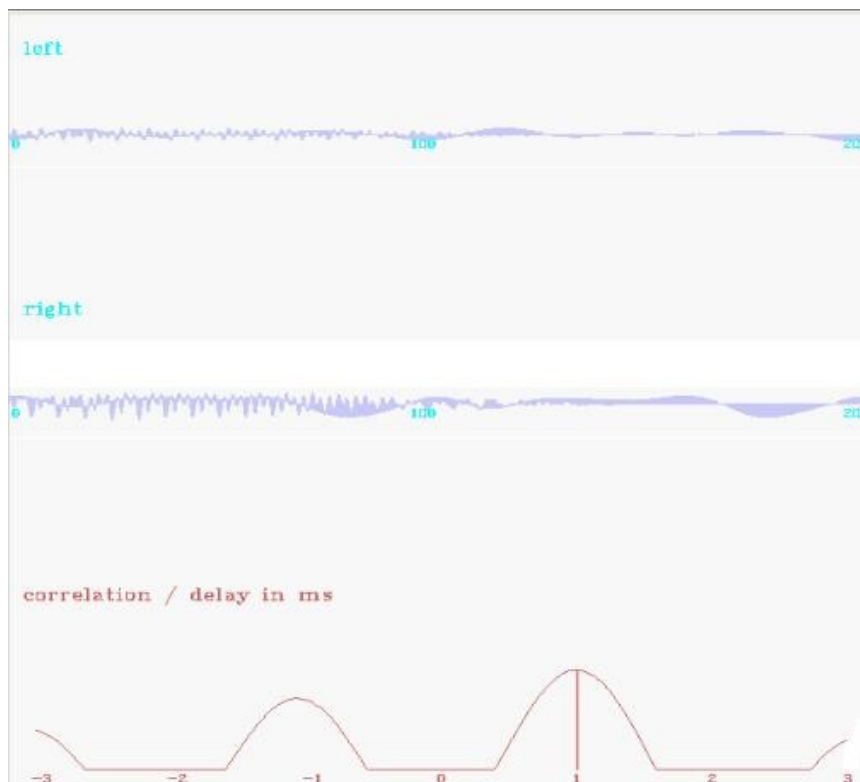


Abbildung 5-5 - Audioframes und Kreuzkorrelation

6. Ausblick und Zusammenfassung

Als abschließender Teil folgt ein Ausblick auf den weiteren Werdegang des Projekts, ein allgemeiner Ausblick und eine kurze Zusammenfassung.

In den nächsten Wochen werde ich weiter an dem System arbeiten und zunächst erst einmal versuchen das gesamte Verfahren robuster zu gestalten, sowohl in der Farbwahrnehmung, als auch in der Audio-Lokalisierung. Dabei werde ich mich näher mit Superpixeln, PCA, konturbasierten Verfahren und auch mit Verbesserungen des TDOA-Algorithmus befassen, auch wenn ich nicht mehr so viel Zeit dafür aufbringen werde, wie zu meinem Praktikum.

Nach dieser intensiven Beschäftigung mit den verschiedenen Techniken bin ich mir sicher, dass es auch in Zukunft viele interessante Ansätze für andere HCI-basierte Anwendungen geben wird, welche das Leben in dieser technologiebasierten Zeit für alle erleichtert. Beispielsweise Hologramm-Bedienfelder und Produkte aus der Augmented Reality, da die Entwicklung immer weiter geht. Bei mir hat mein Praktikum großes Interesse geweckt mich weiter in diese Richtung zu informieren und auch das ein oder andere Projekt in diesem Bereich zu unternehmen.

Zusammenfassend habe ich in dieser Bachelorarbeit zum einen viele Methoden vorgestellt die für die Datenfusion von Audio- und Videodaten für HCI-orientierte Anwendungen von Bedeutung sind, und zum anderen meine 20-wöchige Praktikumsarbeit vorgestellt und bin dabei auf die Besonderheiten und Probleme eingegangen.

7. Literaturverzeichnis

Hier stelle ich die verwendete Literatur vor. Dieses Verzeichnis unterteilt sich dabei in drei Kategorien. Die erste Kategorie sind Literatureinträge, welche für mein Praktikum besonders relevant waren und daher auch für diese Bachelorarbeit von Interesse sind. In der zweiten Kategorie befinden sich Einträge, welche für die entsprechenden Themen von Bedeutung sind und die für die Erstellung des entsprechenden Abschnitts meiner Bachelorarbeit einen Anteil haben, wie etwa Abbildungen, Ideen oder spezielle Erklärungen. Zum Abschluss folgt die dritte Kategorie, welche weiterführende Einträge enthält, welche ich zwar gelesen und für interessant befunden, jedoch nicht speziell genutzt habe.

Überblick:

- 7-I Allgemeine Literatur
- 7-II Spezielle Literatur
- 7-III Weiterführende Literatur

7-I Allgemeine Literatur:

- [Pér04] Pérez, Vermaak, Blake: *Data Fusion for Visual Tracking with Particles*, 2004
- [Ver01] Vermaak, Gangnet, Blake, Pérez: *Sequential Monte Carlo Fusion of Sound and Vision for Speaker Tracking*, 2001
- [Zho08] Zhou, Taj, Cavallaro: *Target detection and tracking with heterogeneous sensors*, 2008
- [Gen05] Oya Gencol: *Datenfusion bei Objektverfolgung*, 2005
- [BraKa] Bradski, Kaehler: *Learning OpenCV*, 2008 First Edition
- [Web00] Goodridge: *Multimedia Sensor Fusion for Intelligent Camera Control and Human-Computer Interaction*, <http://www.ise.ncsu.edu/kay/msf/contents.htm>

7-II Spezielle Literatur:

verwandte Arbeiten (1.3):

- [Bus05] Busso, Hernanz, Chu, Kwon, Lee, Georgiou, Cohen, Narayanan: *Smart Room: participant and speaker localization and identification*, 2005
- [Che04] Checka, Wilson, Siracusa, Darrell: *multiple person and speaker activity tracking with a particle filter*, 2004
- [Gat03] Gatica-Perez, Lathoud, McCowan, Odobez, Moore: *audio-visual speaker tracking with importance particle filters*, 2003

Datenfusion (2.1):

- [Ngu08] Thien-Nghia Nguyen: *Informationsfusion in Fahrerassistenzsysteme*, 2008
- [Rus06] Ruser, Puente: *Informationsvision - Eine Übersicht*, 2006
- [Gen05] Oya Gencol: *Datenfusion bei Objektverfolgung*, 2005

Partikelfilter (2.2):

- [Blake] Blake, Isard: *The Condensation algorithm*
- [Ost05] Sebastian Oster: *Objektverfolgung mit dem Partikel Filter*, 2005

Mensch-Computer-Interaktion (2.3):

- [Kar08] Karray, Alemzadeh, Saleh, Arab: *Human-Computer Interaction: Overview on State of the Art*, 2008
- [V-HCI] Klaus Lepsky: *Vorlesungs-Folien Mensch-Computer-Interaktion FH Köln*

Geräte (2.4):

- [Web01] Logitech-Seite: <http://www.logitech.com/>

Programm-Bibliotheken (2.5):

- [Aga06] Gady Agam: *Introduction to programming with OpenCV*, 2006
- [BraKa] Bradski, Kaehler: *Learning OpenCV*, 2008 First Edition
- [Web02] Entwicklerseite von OpenCV: <http://opencv.willowgarage.com/>
- [Web03] Entwicklerseite von Bass: <http://www.un4seen.com/>

Farbe (3.2.1-A):

- [Web04] Farbmodelle: http://software.intel.com/sites/products/documentation/hpc/ipp/ippi/ippi_ch6/ch6_color_models.html
- [Rah06] Rahman, See: *RGB-H-CbCr Skin Colour Model for Human Face Detection*, 2006
- [Sig03] Sigal, Sclaroff, Athitsos: *Skin Color-Based Video Segmentation under Time-Varying Illumination*, 2003
- [Ald10] Aldasouqi, Hassan: *Human Face Detection System Using HSV*, 2010

Bewegung (3.2.1-B):

- [BraKa] Bradski, Kaehler: *Learning OpenCV*, 2008 First Edition
- [Cha05] Chau, Betke: *Real Time Eye Tracking and Blink Detection with USB Cameras*, 2005
- [For03] Forsyth, Ponce: *Computer Vision: A Modern Approach*, 2003

Kontur (3.2.1-C):

- [Aya03] Ayaz, Adah, Üstün: *Tracking Human Faces Using Motion, Skin Color Segmentation and Ellipse Fitting*, 2003
- [Phu03] Phung, Bouzerdoum, Chai: *skin segmentation using color and edge information*, 2003

[Kosch] Koschan, Curio: *"Active Shape Models" für die Verfolgung nicht-formfester Objekte in Farbbildfolgen*

Merkmale (3.2.1-D):

[Men99] Menser, Müller: *face detection in color images using principal components analysis*, 1999

[Vio04] Viola, Jones: *Rapid Object Detection Using a Boosted Cascade of Simple Features*, 2004

Objekttracking (3.2.2):

[Hu04] Hu, Tan: *A Survey on Visual Surveillance of Object Motion and Behaviors*, 2004

[Hom07] Homberg: *Multi-Objektverfolgung in Farbbildsequenzen auf der Basis von Partikelfiltern*, 2007

[Web05] CamShift:
http://www.cognitics.com/opencv/servo_2007_series/part_3/sidebar.html

[Kri08] Kristan: *Tracking people in video data using probabilistic models*, 2008

Audio (3.3):

[Gen05] Oya Gencol: *Datenfusion bei Objektverfolgung*, 2005

[Gus03] Gustafsson, Gunnarsson: *positioning using time-difference of arrival measurements*, 2003

[Voo04] Voordouw, Yang, Rothkrantz, Mast: *a comparison of the ILD and TDOA sound source localization algorithms in a train environment*, 2004

[Bac10] Bachu, Kopparthi, Adapa, Barkana: *Separation of Voiced and Unvoiced using Zero crossing rate and Energy of the Speech Signal*, 2010

7-III Weiterführende Literatur:

- [Brego] Bregonzio, Taj, Cavallaro: *multi-modal particle filtering tracking using appearance, motion and likelihoods*
- [Iro09] Badali, Valin, Michaud, Aarabi: *Evaluating Real-time Audio Localization Algorithms for Artificial Audition in Robotics*, 2009
- [Ngu03] Nguyen, Aarabi, Sheikholeslami: *Real-time sound localization using field-programmable gate arrays*, 2003
- [Sun06] Sun, Chen, Shi, Chung: *A Novel Method for Multi-sensory Data Fusion in Multimodal Human Computer Interaction*, 2006