Personentracking in Kameranetzwerken mittels graphbasierter Bayes'scher Inferenz

Diplomarbeit

Institut für Theoretische Informatik

Fakultät für Informatik
Universität Karlsruhe (TH)

Florian van de Camp

31. März 2008

Betreuer:

Prof. Dr. Alex Waibel
Dr.-Ing. Rainer Stiefelhagen
Dipl.-Inf. Keni Bernardin
Dr. Jie Yang

Erklärung

Hiermit erkläre ich, die vorliegende Arbeit selbständig erstellt und keine anderen als die angegebenen Quellen verwendet zu haben.

Karlsruhe, 31. März 2008 .....

Inhaltsverzeichnis

1 Einleitung

Das Lokalisieren von Personen ist eine grundlegende Aufgabe in einer Vielzahl von Anwendungsszenarien. Ein Wissen über die Position der im Vordergrund stehenden Objekte oder Personen kann für sich allein eine nützliche Information sein, bietet aber auch einen Ausgangspunkt für eine weiterreichende Analyse dieser Daten sowie einen Ansatzpunkt, auf dieser Information aufbauend, weitere Anwendungen zu entwickeln.
Mit der ständig wachsenden Zahl an möglichen Anwendungen und Einsatzgebieten wachsen auch die Anforderungen an die Trackingsysteme. Die Machbarkeit von Anwendungen, deren Ziel es ist sehr große Areale abzudecken und Personen darin zu verfolgen und zu lokalisieren, wird immer realistischer. Dabei spielen neben der reinen Funktion des Trackingsystems allerdings noch viele andere Anforderungen eine Rolle. Außer den Kosten ist auch die Machbarkeit im realen Umfeld eine solche Anforderung, woraus folgt, dass die vollständige Kameraabdeckung eines großen Areals nicht realistisch ist. Obwohl es nun Bereiche gibt die sich den Blickfeldern der Kameras entziehen, wird vom Trackingsystem erwartet, dass es in der Lage ist diese Bereiche zu überbrücken und so, trotz des eingeschränkten Informationsgewinns, brauchbare Ergebnisse liefert.
Ist ein System in der Lage die Ziele auch unter diesen Bedingungen zu verfolgen, eröffnen sich viele neue Einsatzmöglichkeiten. Ein sehr typisches Einsatzgebiet für das Personentracking ist der Bereich der Videoüberwachung. Diese beschränkt sich in der Regel noch auf begrenzte Räumlichkeiten die mit einer einzigen Kamera zu überschauen sind. Oder aber es muss mit großem Aufwand ein Netzwerk aus überlappenden Kameras aufgebaut werden. Dies hätte zwar den Vorteil, dass der Übergang von einer Kamera zur nächsten einfach zu bewerkstelligen wäre, aber eine solche Installation ist aus bereits genannten Gründen in realen Einsatzgebieten wie z.B. Flughäfen oder Bahnhöfen nicht realistisch.
Ein anderes Einsatzgebiet ist die Effizienzanalyse von Bewegungsabläufen. Wenn viele, autonome Aktoren sich in einem großen Areal bewegen, ist es ein komplexes Problem diese Bewegungsabläufe im Hinblick auf das Gesamtsystem zu optimieren. Zu den Problemen dieser Kategorie zählt beispielsweise die Verkehrsführung in Städten, die Planung von Arbeitswegen in Produktions-hallen oder die Optimierung der Struktur von Einkaufszentren und Supermärkten in Hinblick auf das Kaufverhalten der Kunden. Erst ein Tracking der Personen bietet die nötige Datengrundlage um eine Effizienzanalyse dieser komplexen Vorgänge zu ermöglichen.

Oft reicht das Verfolgen von Personen allein nicht zu Realisation einer Anwendung aus. Die Positionsinformation die es liefert ist aber die Grundlage für die weitere Analyse der Trackingziele oder für den Aufbau von Anwendungen, die in Abhängigkeit von Positionen einzelner Personen spezielle Dienste anbieten können. Im Rahmen einer Büroumgebung könnte so eine Anwendung zum Beispiel eine kluge Telefonzentrale sein: Im typischen Büroalltag gibt es Situationen in denen Anrufe erwünscht sind aber nicht entgegengenommen werden, weil die Person sich gerade in einem anderen Raum aufhält oder aber Anrufe unerwünscht sind wenn die Person sich beispielsweise in einem Besprechungszimmer befindet. Ist dem System aufgrund der Position (und ggf. weiteren Analysen) die aktuelle Präferenz bekannt einen Anruf zu erhalten, kann es diesen automatisch in den Raum schalten, in dem sich die Person gerade aufhält oder den Anruf unterdrücken. Hier wird also die durch das Tracking gewonnene, grundlegende Information über den Aufenthaltsort einer Person verwendet, um eine darauf basierende "kluge" Dienstleistung anzubieten.

1.1 Aufgabenstellung

Ziel dieser Arbeit ist die Entwicklung eines Systems, dass in der Lage ist die Positionen verschiedener Personen innerhalb einer weitläufigen Umgebung gleichzeitig zu verfolgen. Dabei kommt ein Netzwerk von nichtüberlappenden Kameras zum Einsatz, zwischen denen deutliche Lücken zu überbrücken sind. Durch diese Bedingung ist ein Informationsverlust unvermeidbar, so dass eine fehlerfreie Verfolgung der Personen deutlich erschwert wird. Um trotzdem ein bestmögliches Trackingergebnis liefern zu können, kommt ein probabilistisches Verfahren zum Einsatz dessen Grundlagen in Kapitel 2 erläutert werden. Konkret muss ein probabilistisches Modell entwickelt werden, das eine Anwendung des erwähnten, grundlegenden Verfahrens auf das hier vorliegende Trackingproblem erlaubt. Auf die damit verbundenen Details wird in Kapitel 4 näher eingegangen.
Die untergeordneten Aufgaben der Detektion, Merkmalsextraktion und des Trackings auf der Ebene der Bildverarbeitung werden in Kapitel 3 vorgestellt. Dabei geht es um die Bestimmung von Merkmalen, die für das gegebene Szenario und die damit verbundenen Probleme geeignet sind Personen so zu beschreiben, dass sie an beliebigen Kameras im Netz wiedererkannt werden können.
Zur Evaluation des Systems ist der Zusammenhang zwischen Sensorgenauigkeit und erzielbaren Trackingergebnissen besonders interessant. Um ein breites Spektrum von Parameterkombinationen für die Genauigkeit von Sensoren und Komponenten evaluieren zu können, wird eine Simulation verwendet, die in Kapitel 5 vorgestellt wird. Die Ergebnisse der Evaluation und eine ausführliche Analyse finden sich schließlich in Kapitel 6.

1.2 Stand der Forschung

Das Lokalisieren von Personen wird schon länger in verschiedensten Anwendungen eingesetzt, weshalb seit Jahren aktive Forschung in diesem Bereich betrieben wird. Ganz prinzipiell lassen sich dabei Einzelkamerasysteme und Multikamerasysteme unterscheiden. Während die Methodik des Personentrackings in Einzelkamerasystemen eine notwendige Grundlage auch für Multikamerasysteme liefert, bringen diese eine große Zahl neuer Aufgaben und Probleme mit sich. Dennoch werden Multikamerasysteme mehr und mehr eingesetzt und die damit Verbundenen Probleme gelöst, da sie auch eine Reihe von Vorteilen mit sich bringen. So ist beispielsweise das Problem von Verdeckungen, das bei dem Einsatz von monokularen Trackern auftaucht, mit mehreren Kameras gut lösbar. Besonders solche Systeme, mit sich überlappenden Kameraansichten auf begrenztem Raum, finden einen vielfältigen Einsatz. Das von Lanz [1] entwickelte System zum Beispiel kann somit, nach manueller Initialisierung, hervorragende Ergebnisse selbst bei zeitweiser, vollständiger Verdeckung erreichen. Das "BraMBLe" System von Isard et al. [2] verfolgt einen probabilistischen Ansatz zur erfolgreichen Lösung des gleichen Problems. Während in Systemen mit sich überlappenden Kameras eine vielzahl von Informationen vorhanden sind, und viele Ansätze um diese zu kombinieren [3, 4, 5, 6], stellen sich bei der Verwendung von weitläufigeren Systemen, bei denen die Kameras kein gemeinsames Blickfeld besitzen, eine Reihe neuer Probleme. Abhängig von der Größe der entstehenden Lücken zwischen den Kameras ist es unter Umständen noch möglich, diese mit herkömmlichen Trackingverfahren zu überbrücken. Denn auch in Kameranetzen mit überlappendem Blickfeld oder auch in Einzelkamerasystemen kann es zu Ausfällen auf der Beobachtungsebene kommen, die das Trackingsystem überbrücken muss. So verwenden Chilgunde et al. [7] einen Kalmanfilter zur Vorhersage der Trajektorie einer Person beim Verlassen des Blickfeldes einer Kamera, da es bei kleinen Lücken auf diese Weise möglich ist zu bestimmen in welcher Kamera die Person zu welchen Zeitpunkt wieder eintritt. Liegen größere Entfernungen zwischen den einzelnen Kameras, ist ein derartiges Vorgehen nicht mehr möglich. Da Personen sich für lange Zeit in unüberwachten Bereichen aufhalten, ist nicht mit ausreichender Präzision vorhersagbar wo eine Person als nächstes erscheinen wird. Daher müssen Personen in jeder Kamera hauptsächlich aufgrund ihres Erscheinungsbildes wiedererkannt werden. Eine Möglichkeit mit den verschiedenen Aufnahmebedingungen umzugehen die aufgrund der grösseren Distanz zwischen den Kameras entstehen, ist eine Analyse der Unterschiede, um diese auszugleichen. So beschreiben Gilbert und Bowden [8, 9] ein System zum inkrementellen Lernen einer Farbkalibrierung, die es erlaubt die Modellierung des Erscheinungbildes einer Person zum Tracken über mehrere Kameras hinweg einzusetzen. Von besonderem Interesse sind die Zusammenhänge der Kameras in weitläufigen Netzwerken, also welche Verbindungswege es prinzipiell gibt. Mit diesem Problem beschäftigen sich Ellis et al. [10, 11]. Obwohl ihr System prinzipiell für das Personentracking in einem Kameranetzwerk

ausgelegt ist, liegt der Fokus auf dem automatischen, unüberwachten Lernen der Netzstruktur. Anstatt also das Wissen über die Verteilung der Kameras vorauszusetzen wird diese Struktur anhand der Beobachtungen der Kameras autonom gelernt. Die maximal möglichen Kameraabstände sind hier insofern beschränkt, als dass die zeitliche Folge von Beobachtungen zur Bestimmung der Identität einer Person beiträgt.

Bayesnetze finden in den verschiedensten Varianten und für diverse Zwecke Verwendung im Bereich des Personentrackings. Das häufig für die Detektion bestimmter Bewegungsabläufe eingesetzte Hidden Markov Modell erlaubt es, das Verhalten von Zielpersonen oder ihrer Bewegungen auf bestimmte Muster hin zu untersuchen [12]. In diesem Zusammenhang ist besonders die Arbeit von Oliver et al. [13, 14] interessant. Auch sie setzen ein Simulationssystem ein, um ihre probabilistischen Modelle mit einer großen Zahl von Testdaten und unter Verwendung verschiedener Parameter evaluieren zu können. Der Fokus liegt dabei allerdings besonders auf der Analyse der für die Detektion von Bewegungsmustern geeignetste Form von Hidden Markov Modellen. Nicht nur im Bereich des Personentrackings in Kameranetzwerken ist es möglich mit Hilfe von Bayesnetzen die Trackingergebnisse zu verbessern. So stellen Buxton und Gong [15, 16] einen monokularen Tracker vor, der Bayesnetze zur Verhaltensanalyse von Objekten einer beobachteten Szene nutzt. Das Wissen um die Verhaltenshistorie einzelner Objekte in der Szene hilft ihnen deren zukünftige Bewegungen einschätzen zu können. Da Bayesnetze eine effektive und parallele Verfolgung verschiedener Erklärungspfade erlauben, nutzen Abrantes et al. [17] sowie Nillius et al. [18] dieses Prinzip um das Zuordnungsproblem von Beobachtungen für mehrere Personen zu lösen. Auf diese Art und Weise können sie mehrere Personen im Blickfeld der Kamera verfolgen und dabei mit Verdeckungen, fehlenden und auch falschen Trackingergebnissen langfristig umgehen. Einen interessanten Einsatz von Bayesnetzen zum Personentracking stellen Madigan et al. [19] vor: Mit Hilfe der Information über die Signalstärken, mit denen im Gebäude verteilte Router eines Wireless-Netzwerks einen Computernutzer erreichen, wird dessen aktuelle Position bestimmt. In diesem Szenario steht zwar der für Systeme des Personentrackings typische Bildverarbeitungsteil außen vor, aber auch wenn sich das Szenario in einigen Aspekten deutlich von dem hier vorgestellten System abgrenzt, zeigt es welches Potential in der probabilistischen Fusion der verfügbaren Beobachtungen steckt um das Tracken von Personen zu verbessern. Mit dem Einsatz probabilistischer Verfahren für das Personentracking in Kameranetzwerken beschäftigen sich Kröse et al. und stellen ein generelles Framework vor [20, 21], das einen besonderen Fokus auf die Reidentifikation einzelner Personen besitzt. In einer praktischen Umsetzung des Frameworks liegen die Übergangzeiten zwischen den Kameras im Schnitt allerdings deutlich unter einer Minute. Außerdem nehmen sie an, dass Detektion und Tracking perfekt gelöst sind - das Kernproblem beschränkt sich also auf die Reidentifikation der Personen. Die anfangs erwähnten Probleme, die in Kameranetzwerken durch die an verschiede-

nen Orten variierenden Aufnahmebedingungen entstehen, finden sich auch in dem System von Zajdel et al. [22] wieder. Hier geht es darum, dass eine mobile Roboterplattform in der Lage sein soll, Personen aufgrund ihres Erscheinungsbildes an beliebigen Orten wiederzuerkennen. Aufgrund der sich bewegenden Plattform stellen sich hier ebenfalls Probleme bezüglich verschiedener Aufnahmebedingungen. Allerdings steht auch hier die Reidentifikation, nicht die Lokalisierung im Vordergrund.

1.3 Beiträge

Die bereits beschriebenen Anforderungen, die an das im Rahmen dieser Arbeit entwickelte System gestellt werden, bringen eine Reihe von Problemen mit sich. Die hier entwickelten Verfahren und Ansätze zur Lösung dieser Probleme sind im Folgenden noch einmal übersichtlich aufgeführt.

- Der Entwurf eines probabilistischen Verfahrens zum simultanen Tracken mehrerer Personen in einem Kameranetzwerk. Besonders hervorzuheben ist an dieser Stelle die sehr dünne Raumabdeckung durch die Kameras und die dadurch entstehenden erheblichen Lücken. Zwischen zwei Beobachtungen vergehen also eher Minuten als nur Sekunden.
- Die Bestimmung eines kompakten Sets intuitiver und einfacher Merkmale für das oben gestellte Trackingproblem. Wie bereits erläutert stellt das Tracking in einem weitläufigen Kameranetzwerk besondere Anforderungen an die eingesetzten Merkmale um eine Vergleichbarkeit über verschiedene Kameras hinweg sicherzustellen. Besonders in Multikamerasystemen spielt die Komplexität der Merkmalsextraktionen die zum Einsatz kommen eine besondere Rolle. Einfache Merkmale erlauben einen effizienten Einsatz auch in komplexen Systemen, die Kombination mehrerer Merkmale bietet trotz einfacher Komponenten eine effektive Identifikation der Personen. Darüber hinaus ist die Untersuchung der Anwendbarkeit der ausgewählten, robusten Merkmale Teil dieser Arbeit.
- Zusätzlich zum Aufbau des Gesamtsystems erfolgt eine detaillierte Analyse der Auswirkungen unterschiedlicher Sensorgenauigkeiten auf die Gesamtperformance. Dies erlaubt zum einen eine bessere Evaluierung der Schätzung des probabilistischen Modells und zum anderen ermöglicht es eine Einschätzung der erreichbaren Systemperformance unter gegebenen Randbedingungen.

2 Grundlagen

In diesem Kapitel sollen die Grundlagen für das in dieser Arbeit entwickelte, probabilistische Modell vorgestellt werden. Im wesentlichen handelt es sich dabei um die so genannten "Hidden Markov Modelle". Der Fokus liegt dabei auf dem Forward Algorithmus als Teil des Evaluierungsproblems und seiner Implementierung, da diese eine zentrale Komponente im Rahmen des vorgestellten Verfahrens darstellt.

2.1 Hidden Markov Modelle

Hidden Markov Modelle bieten ein mächtiges Werkzeug zur Modellierung zeitdiskreter stochastischer Prozesse. Aus der Mathematik stammend, werden Hidden Markov Modelle heute zur Lösung praktischer Probleme in einer großen Zahl von Anwendungsgebieten eingesetzt. Neben verschiedenen Bereichen der Biologie und den Wirtschaftswissenschaften, zählt wohl die Spracherkennung zu den prominentesten Einsatzgebieten [23, 24, 25]. Die grundlegende Idee des Hidden Markov Modells lässt sich anhand der beiden Zufallsprozesse beschreiben aus denen es besteht.
Der erste Prozess modelliert dabei den Teil des Systems, der nicht beobachtbar - also hidden - ist. Dabei handelt es sich um die Zustände und Übergangswahrscheinlichkeiten wie sie in einer Markovkette vorkommen. Eine Markovkette ist ein stochastischer Prozess, dessen Besonderheit darin besteht, dass mit einer begrenzten Vorgeschichte ebensogute Prognosen gemacht werden können, als wäre die gesamte Vorgeschichte bekannt. Der zweite Prozess erzeugt, anhand einer zustandsabhängigen Wahrscheinlichskeitsverteilung, in jedem Zeitschritt ein beobachtbares Ausgabesymbol.

Formal lässt sich ein Hidden Markov Modell als Fünftupel der Form $\lambda = (S, A, B, \pi, V)$ definieren:

- $S = \{s_1, ..., s_n\}$ Die Menge möglicher Zustände
- $A = \{a_{ij}\}$ Die Zustandsübergangsmatrix, wobei a_{ij} die Wahrscheinlichkeit angibt von Zustand s_i in Zustand s_j zu wechseln.

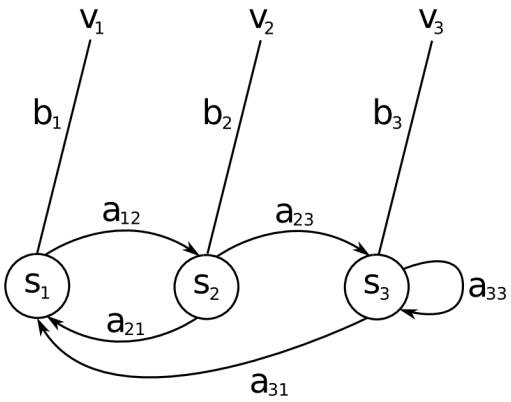


Abbildung 2.1: Illustration des Hidden Markov Modells

- $B = b_1, ..., b_n$ Die Emissionswahrscheinlichkeiten
- $b_i(x)$ Ist dabei die Wahrscheinlichkeit, im Zustand s_i die Beobachtung x zu machen
- π Initiale Wahrscheinlichkeitsverteilung über die Zustände
- V Das Ausgabealphabet, also die Menge der beobachtbaren Symbole

Das Zusammenspiel dieser Modellparameter ist in der Abbildung 2.1 noch einmal illustriert.

Im Zusammenhang mit Hidden Markov Modellen existieren drei übliche Problemstellungen:

Evaluierungsproblem:

Das Evaluierungsproblem beschäftigt sich mit der Fragestellung nach der Wahrscheinlichkeit, dass eine Beobachtung O unter einem gegebenen Hidden Markov Modell λ gemacht wurde. Eine effiziente Lösung dieses Problems stellt der Forward Algorithmus dar, der im Rahmen dieser Arbeit eine entscheidende Rolle spielt und daher weiter unten detailliert besprochen wird.

Dekodierungsproblem:

Innerhalb des Dekodierungsproblems soll geklärt werden, welches die wahrscheinlichste Zustandsfolge durch ein gegebenes Hidden Markov Modell λ war die eine Beobachtung O erzeugt hat. Dieses Problem wird üblicherweise mit dem Viterbi Algorithmus [25] gelöst.

Lernproblem:

Allein anhand von gegebenen Beobachtungssequenzen O sollen die korrekten Parameter des Hidden Markov Modells λ in einem überwachten Lernschritt bestimmt werden. Ein effizienter Algorithmus zur Lösung dieses Problems stellt der Baum-Welch Algorithmus [25] dar.

2.1.1 Forward Algorithmus

Der Forward Algorithmus löst das Evaluierungsproblem eines gegebenen Hidden Markov Modells λ. Er verwendet dazu die Methode der dynamischen Programmierung. Im Rahmen des Evaluierungsproblems steht die Gesamtwahrscheinlichkeit P im Vordergrund. Sie gibt Auskunft darüber wie gut Modell und Beobachtung zusammen passen, also dass das gegebene λ den der Beobachtung zugrundeliegenden Prozess richtig modelliert.
Um P zu berechnen werden die Wahrscheinlichkeiten aller möglichen Wege durch das Modell berechnet. Ziel dabei ist es für jeden möglichen aktuellen Zustand s_j alle theoretisch möglichen Wege zu finden, die eine Erklärung dafür liefern, dass sich das System nach dem i-ten Zeitschritt in Zustand s_j befindet. Aufgrund dieser Vorgehensweise zur Berechnung der Gesamtwahrscheinlichkeit, erhält man zusätzlich in jedem Zeitschritt eine Wahrscheinlichkeitsverteilung über die vorhandenen Zustände - eine Eigenschaft, die bei der Entwicklung des probabilistischen Modells dieser Arbeit ausgenutzt wird.

Die rekursive Berechnung dieser Wahrscheinlichkeitsverteilung (Formeln 2.2 und 2.3) und der sich daraus ergebenden Gesamtwahrscheinlichkeit (Formel 2.4) lässt sich formal wie folgt definieren:

Das Hidden Markov Modell λ ist wie oben definiert als $\lambda = (S, A, B, \pi, V)$.

Die Wahrscheinlichkeit zum Zeitpunkt t bei gegebener Beobachtung $O = (o1, o2, ..., ot)$ im Zustand s_i zu sein, ist:

$$\alpha_t(i) = P(o_1, o_2, ..., o_t, q_t = s_i|\lambda) \tag{2.1}$$

$\alpha_t(i)$ wird dabei als Forward-Variable bezeichnet. Diese (und damit auch die Gesamtwahrscheinlichkeit P) lässt sich rekursiv berechnen:

$$\alpha_1(i) = P(o_1, q_1 = s_i|\lambda) = \pi_i b_i(o_1) \tag{2.2}$$

$$\alpha_t(j) = P(o_1, \ldots, o_t, q_t = s_j|\lambda) = \sum_{i=1}^{|S|} \alpha_{t-1}(i) a_{ij} b_j(o_t) \quad \text{mit } 1 < t \leq T \tag{2.3}$$

$$P(\mathcal{O}|\lambda) = \sum_{i=1}^{|S|} \alpha_T(i) \tag{2.4}$$

Implementierung

Ein typisches Problem bei der Implementierung des Forward Algorithmus sind arithmetische Unterläufe bei der Verwendung von Gleitkommazahlen. Diese entstehen durch die ständig wiederholte Multiplikation von Wahrscheinlichkeiten mit einem Wert < 1. Eine übliche Vorgehensweise zur Lösung dieses Problems [26] ist die Skalierung der Wahrscheinlichkeitswerte $\alpha_t(i)$. Die skalierten Werte $\hat{\alpha}_t$ nach t Beobachtungen berechnen sich dann mittels Formel 2.5.

$$\hat{\alpha}_t(i) = \frac{\alpha_t(i)}{\sum_{i=1}^{|S|} \alpha_t(i)} \tag{2.5}$$

3 Merkmale

Zum Aufbau eines Systems zum Lokalisieren von Personen ist es zunächst notwendig geeignete Features zu bestimmen, die es erlauben die Zielpersonen zu beschreiben und somit von anderen zu unterscheiden. Im allgemeinen Fall von Multikamerasystemen ist es notwendig, dass die verwendeten Merkmale in allen Kameras bestimmt werden können und bei Beobachtung des gleichen Objekts auch zu einer Übereinstimmenden Merkmalsbeschreibung gelangen. Voraussetzung dafür ist, dass alle Beobachtungen unter möglichst identischen Bedingungen zustande kommen. Neben dem wichtigsten Einflußfaktor, der Beleuchtung, spielen aber auch Faktoren wie die Haltung, die Ausrichtung der Person zur jeweiligen Kamera und der Hintergrund bzw. die Umgebung eine wichtige Rolle.
Für Kameras mit sich überlappenden Blickfeldern auf begrenztem Raum gibt es was diese Faktoren betrifft in der Regel keine allzu großen Abweichungen. Hauptsächlich sind Variationen über größere Zeiträume zu beobachten, wie zum Beispiel der Wechsel von natürlichem zu künstlichem Licht im Verlauf eines Tages. Darüber hinaus lassen sich in dem Fall überlappender Blickfelder noch eine Reihe zusätzlicher Informationen nutzen, wenn Kalibrierungsdaten der Kameras vorhanden sind [1].
In dem speziellen Fall von Kameras die kein gemeinsames Blickfeld besitzen und in großer räumlicher Distanz angeordnet sind, fällt ein Großteil dieser Informationen weg und die Bedingungen unter denen Beobachtungen an den einzelnen Kameras gemacht werden, variieren sehr stark.
Aufgrund dieser erschwerenden Faktoren ist es notwendig zunächst Merkmale auszuwählen die in einem solchen Szenario einsetzbar sind oder speziell dafür angepasst werden können.
Im Folgenden soll es nun zunächst um die Auswahl geeigneter Merkmale gehen und die Untersuchung ihrer praktischen Anwendbarkeit. Im Anschluss werden die Merkmale und ihre Gewinnung im einzelnen, sowie die verwendeten Techniken zur Kombination einzelner Beobachtungen sowie die Fusion der verschiedenen Merkmale beschrieben. Da alle vorgestellten Merkmale die Kenntniss über die Präsenz einer Person im Kamerabild voraussetzen werden zum Schluss noch die Detektionsverfahren vorgestellt, die im Rahmen dieser Arbeit zum Einsatz kommen.

3.1 Abstrakte Merkmale

Die bereits beschriebenen Probleme im Zusammenhang mit großen räumlichen Entfernungen zwischen Kameras im Kameranetzwerk reduzieren die Menge der verwendbaren Merkmale im wesentlichen auf solche, die allein das grobe Erscheinungsbild einer Person beschreiben. Gerade dies unterliegt aber aufgrund der an verschiedenen Orten unterschiedlichen Aufnahmebedingungen einer starken Variation. Vergleiche von konkreten Messwerten wie beispielsweise Farbhistogrammen wie sie im Fall überlappender Kameras möglich sind [8], können hier nicht ohne weiteres angestellt werden. Ein verbreitetes Vorgehen, im Bereich der kognitiven Informationsverarbeitung ist die Nachahmung der Problemlösungsprozesse des Menschen. Im Bezug auf das hier geschilderte Problem konkreter Messwerte zum Merkmalsvergleich fällt schnell auf, dass die menschliche Beschreibung einer Person nicht in Zahlen gefasst wird sondern auf einer abstrakteren Ebene ansetzt. Genau dieses abstrakte Beschreibungsniveau ermöglicht es beispielsweise die Farbe „Rot“ auch unter anderen Beleuchtungsverhältnissen als solche wiederzuerkennen.
Es sind also Merkmale notwendig, die unabhängig von lokalen Einflussfaktoren sind. Dies kann zum einen durch die Auswahl von Merkmalen geschehen die von sich aus robust gegenüber lokalen Aufnahmebedingungen sind, oder aber durch eine Vereinfachung und Verallgemeinerung von Merkmalen, so dass Einflüsse dieser Bedingungen minimiert werden. Neben dieser ersten Eigenschaft muss allerdings zusätzlich die praktische Anwendbarkeit der Merkmale gegeben sein. Diese müssen also sehr schnell (nahe Echtzeit) extrahierbar sein und auf realistischen Videodaten nutzbare Ergebnisse liefern.

3.2 Merkmalsextraktion

Im Folgenden sollen nun die einzelnen verwendeten Merkmale vorgestellt werden. Für jedes Merkmal werden dabei die angewandten Techniken und Algorithmen zur Extraktion erläutert. Darüber hinaus wird die Nutzbarkeit anhand von Evaluationen auf jeweils geeigneten Daten untersucht. Alle folgenden Algorithmen setzen entweder eine Gesichtsdetektion, oder zumindest eine Detektion der Person als ganzes vorraus. Details zu den verwendeten Detektionsverfahren finden sich in Abschnitt 3.5

3.2.1 Brille

In diesem Abschnitt geht es um das Merkmal „Brille“, also darum, ob eine Person eine Brille trägt oder nicht. Der Vorteil dieses Merkmals liegt in der Invarianz

gegenüber temporären sowie beleuchtungstechnischen Variationen der Aufnahmebedingungen.
Die Grundidee des verwendeten Algorithmus basiert darauf, dass im Gesicht jedes Brillenträgers in bestimmten Regionen [27] Merkmale zu finden sind die ohne eine Brille nicht im Gesicht vorhanden wären. Hier geht es speziell um die Brücke der Brille (Abbildung 3.1 a) da dies das einzige Merkmal ist, das unabhängig von der großen Variation an Brillenmodellen in jedem Fall vorhanden ist.
Ausgehend von einer Detektion eines frontalen Gesichtes (großes Rechteck in Abbildung 3.1 b) lässt sich aufgrund anatomischer Gegebenheiten die Augenregion grob abschätzen (kleines Rechteck in Abbildung 3.1 b). Im nächsten Schritt wird versucht, die Position beider Augen zu bestimmen. Ausgehend von der Annahme, dass die Pupillen die dunkelsten Bereiche in der Augenregion darstellen wird jedes Pixel zunächst in Abhängigkeit seines Graustufenwertes g gewichtet. Um zu vermeiden, dass dunkle Augenbrauen, Haarsträhnen oder dunkle Brillenrahmen als Pupillen fehlinterpretiert werden, kommt eine zusätzliche Gewichtung aller Pixel in Abhängigkeit ihrer Position zum Einsatz. Dazu werden ausgehend von der ursprünglichen Schätzung der Augenregion auch die ungefähre Lage der linken (Cl_x, Cl_y) und rechten (Cr_x, Cr_y) Pupille berechnet (Zentrum der Kreise mit Durchmesser h in Abbildung 3.1 b) und nahe gelegene Pixel werden höher gewichtet als solche die weit entfernt liegen. Zuvor ist allerdings noch eine Zuordnung der Pixel zur linken oder rechten Pupille nötig. Diese wird zunächst für jedes Pixel durch Bestimmen des kleineren Abstandes festgelegt. Die Berechnung der Pixelgewichte erfolgt dann mittels Formel 3.1 für Pixel, die der linken Pupille zugeordnet wurden (φ_l) und mittels Formel 3.2 für Pixel, die der rechten Pupille zugeordnet wurden (φ_r).

$$\varphi_l(x,y) = \left(1 - \frac{\sqrt{(Cl_x - x)^2 + (Cl_y - y)^2}}{(h/2)}\right) \cdot \left(1 - \frac{g(x,y)}{255}\right) \tag{3.1}$$

$$\varphi_r(x,y) = \left(1 - \frac{\sqrt{(Cr_x - x)^2 + (Cr_y - y)^2}}{(h/2)}\right) \cdot \left(1 - \frac{g(x,y)}{255}\right) \tag{3.2}$$

Das oben beschriebene Verfahren wurde auf der FERET Datenbank [28] evaluiert. Die FERET Datenbank ist ein Standard im Bereich der Gesichtserkennung und ermöglicht aufgrund seiner starken Verbreitung eine gute Vergleichbarkeit von Ergebnissen. Um den Bedingungen eines realen Einsatzes gerecht zu werden, wurden die Aufnahmen aus FERET auf eine Auflösung von 60×120 Pixeln reduziert und dann nur solche Aufnahmen verwendet, bei denen mittels dem in 3.5.1 beschriebenen Verfahren ein Gesicht detektiert wurde. Darüber hinaus sind Brillenträger in der FERET Datenbank unterrepräsentiert, weshalb alle Brillenträger der Datenbank und eine gleiche Anzahl zufällig ausgewählter Personenaufnahmen ohne Brille verwendet wurden.
Auf den verbleibenden 3782 Aufnahmen erreicht das vorgestellte Verfahren eine Fehlerrate von 9%.

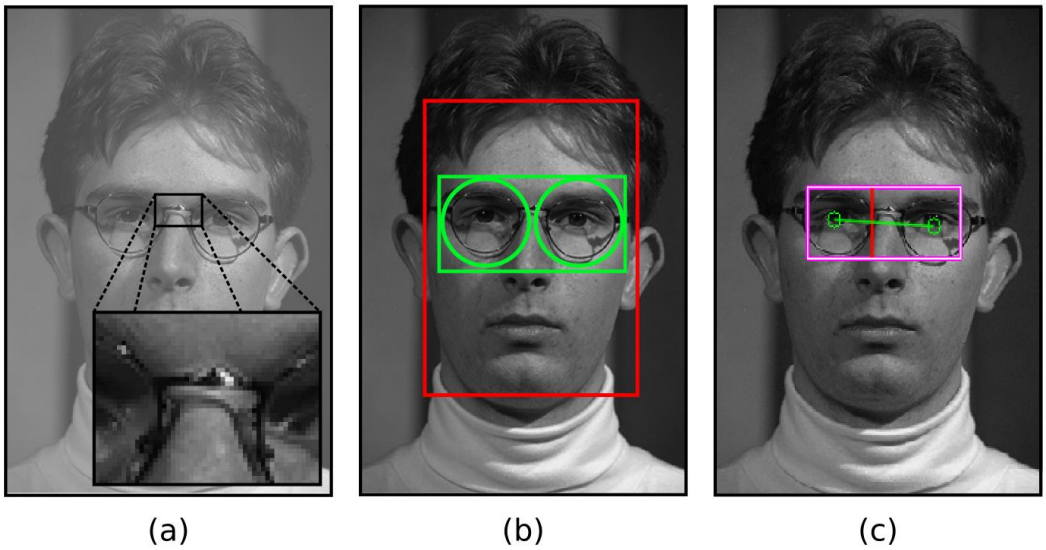


Abbildung 3.1: Illustration der Brillendetektion

3.2.2 Geschlecht

Das Geschlecht einer Person ist ein Merkmal, das völlig unabhängig von Variationen der Aufnahmebedingungen jeglicher Art ist. Natürlich können bestimmte Aufnahmebedingungen die prinzipielle Bestimmung dieses Merkmals erleichtern oder erschweren, aber wenn es erfolgreich extrahiert werden kann ist es ein zuverlässiger Hinweis zur Identifikation der Person.
Selbst Menschen fällt es schwer klare Regeln zu definieren, an denen sich festmachen ließe, ob es sich bei einer Abbildung einer Person um einen Mann oder um eine Frau handelt. Wenn es auch Eigenschaften gibt, die häufig zutreffen (zum Beispiel langes Haar) so sind sie kein zuverlässiger Indikator. Ähnlich wie der Mensch lernt, zwischen Mann und Frau zu unterscheiden ohne dies in klaren Regeln fassen zu können, ist auch hier die Wahl auf einen Ansatz gefallen, der versucht durch eine Methode des maschinellen Lernens anhand einer großen Anzahl von Beispielen zu lernen, diese Unterscheidung zu treffen.
Bei der hier eingesetzten Methode geht es konkret um so genannte Support Vector Machines (SVMs). SVMs sind Klassifikatoren, die sich in vielen praktischen Einsätzen als robust erwiesen haben. Neben ihren in der Regel sehr guten Ergebnissen im Vergleich zu anderen Klassifikatoren haben sie den Vorteil, dass sich das Problem des Overfittings häufig vermeiden lässt [29]. Support Vector Machines sind ein statistisches Lernverfahren und wurden maßgeblich von Vladimir N. Vapnik [29, 30] entwickelt. Die prinzipielle Idee der Support Vector Machines basiert darauf, eine optimale Trennebene (Hyperebene) zwischen den zu trennenden Klassen zu finden. Für linear trennbare Daten ist dies einfach. Andernfalls kommt der so genannten Kernel-Trick zum Einsatz, der auf der Annahme basiert, dass die Daten sich durch Transformation in einen höher dimensionalen Merkmalsraum

doch linear separieren lassen.

Ein wichtiger Faktor beim Einsatz von Support Vector Machines zur Klassifikation visueller Daten ist die Methode mittels derer die Bilddaten in Merkmalsvektoren umgewandelt werden. Die besten Ergebnisse wurden hier mit so genannten Discrete Cosine Transform (DCT) Features erzielt, die zum Vergleich untersuchten Kantenmerkmale führten zu deutlich höheren Fehlerraten. Bei den DCT Features handelt es sich um die Koeffizienten die aus der Transformation des Bildes mittels der diskreten Kosinustransformation hervorgehen [31].

Zur Evaluation wurden wieder die wie oben beschrieben skalierten Aufnahmen aus FERET verwendet. Zum Trainieren wurden die ersten 1000 Aufnahmen verwendet, die Evaluation erfolgte dann auf den restlichen 1934 Aufnahmen. Dabei wurde eine Fehlerrate von 16 % erreicht.

3.2.3 Farben

Obwohl Farben zu den Merkmalen gehören die sehr anfällig auf Veränderungen in der Beleuchtung sind, können sie unter bestimmten Bedingungen auch in Kameranetzwerken eingesetzt werden. Der gewöhnliche Einsatz von Farben als Merkmal basiert auf Vergleichen von aus den Aufnahmen extrahierten Histogrammen. Bereits geringe Variationen der Beleuchtung können Histogramme stark verändern. Ein Einsatz von Farbmerkmalen darf also hier nicht auf dem Vergleich exakter Zahlenwerte beruhen, es geht vielmehr darum einen dominierenden Farbton wiederzuerkennen, als Farbnuancen unterscheiden zu können.
Konkret bedeutet dies, dass die Beschreibung der Farben hier in grobe Kategorien aufgeteilt wird, ohne dass Variationen innerhalb einer Kategorie von Bedeutung wären.
Diese Reduktion der Granularität beschränkt die Unterscheidbarkeit ähnlicher Farben, ermöglicht es aber auch unter variierenden Aufnahmebedingungen divergente Farben zu erkennen.

Haar

Die Haarfarbe ist ein sehr beständiges Merkmal einer Person und kann in wenige grobe Kategorien unterteilt werden. Im folgenden geht geht es also darum die Haarfarbe einer Person in eine der groben Kategorien Blond, Brünett, Schwarz oder Rot einzuordnen und nicht einen bestimmten Farbton möglichst genau zu klassifizieren.

Das größte Problem ist hierbei nicht die Bestimmung der Farbe, sondern die Bestimmung der Bildbereiche in denen Haare abgebildet sind. Ausgehend von einer

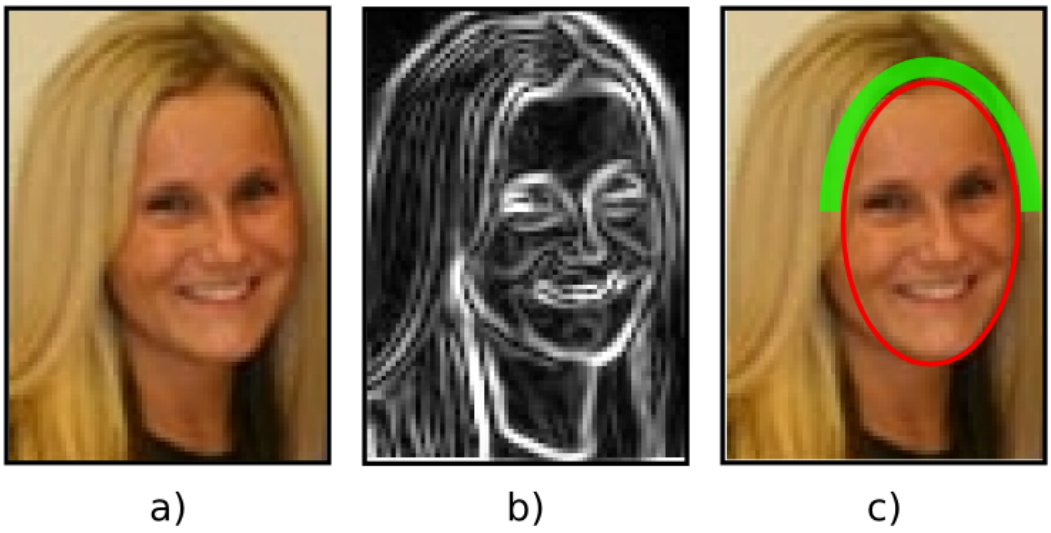


Abbildung 3.2: Extraktion von Haarfarbe

Detektion eines frontalen Gesichtes (Abbildung 3.2 a) wird dazu zunächst ein Gradientenbild berechnet (Abbildung 3.2 b). Mit Hilfe von Ellipsenfitting [32] wird nun eine möglichst gute Annäherung der genauen Umrisse des Gesichtes bestimmt (Ellipse in Abbildung 3.2 c). Dies ist vor allem von Bedeutung um eine möglichst genaue Lokalisierung des Haaransatzes zu ermöglichen. Der Bereich unmittelbar über dem Haaransatz ist der sicherste Ort im Bild um tatsächlich Haare zu finden - andere Regionen hängen zu stark von Variationen in Frisur und Aufnahmewinkel ab.Um Ungenauigkeiten in der Bestimmung des Haaransatzes vorzubeugen werden im nächsten Schritt alle als Hautfarbe zu klassifizierenden Pixel maskiert. Um die Hautfarbe nicht generisch zu beschreiben sondern aufnahmespezifisch definieren zu können, wird diese als ein Histogramm eines Bildausschnittes bestimmt: Ausgehend von dem Rechteck, das die initiale Gesichtsdetektion liefert, wird ein Rechteck von halber Größe aber mit gleichem Zentrum definiert und als Referenz für die Hautfarbe verwendet.
Der Bereich, der nun zum eigentlichen Bestimmen der Haarfarbe verwendet wird, ist die Halbellipse oberhalb des gefundenen Haaransatzes (Abbildung 3.2 c). Zur Bestimmung der Breite dieses Bogens wird der Gradientenverlauf im Bild vom gefundenen Haaransatz aus nach oben untersucht. Ein starker Anstieg oder Abfall des Gradienten deutet auf den Übergang von Haaren zum Hintergrund hin und beschränkt so die Breite des zu untersuchenden Bereichs. Besonders in Fällen von sehr kurzen oder nicht vorhandenen Haaren kann die Breite des Bogens also sehr klein sein, was ab einem Grenzwert dazu führt, dass keine Haarfarbe bestimmt werden kann und das System keinen Farbwert zurückgibt.
Die Evaluation dieses Verfahrens auf der FERET Datenbank analog zu den bisher beschriebenen Merkmalen ist aufgrund der lediglich in Graustufen vorliegenden Aufnahmen nicht möglich. Zur Auswertung wurden daher Videoaufnahmen in

einer Büroumgebung gesammelt. In den Videodaten wurden 3276 Gesichter von 13 verschiedenen Personen unter verschiedenen Aufnahmebedingungen detektiert. Das vorgestellte Verfahren erreicht auf diesen Daten eine Fehlerrate von 22 %.

Ober- und Unterkörper

Die Farben von Ober- und Unterkörper sind besondere Merkmale, da sie als einzige nicht vollständig von einer erfolgreichen Gesichtsdetektion abhängen. Als Oberkörper wird hier der Bereich vom Hals bis zur Gürtellinie bezeichnet, als Unterkörper entsprechend alles abwärts der Gürtellinie. Allein die Position der Person - gewonnen durch Omegashapedetektion (siehe Abschnitt 3.5.2) - ohne frontale Aufnahme des Gesichtes genügt, um diese Merkmale zu extrahieren.
Allerdings stellt sich auch hier das Problem variierender Beleuchtung, weil es sich um Farbmerkmale handelt. Ähnlich wie im Fall der Haarfarbe wird dieses Problem auch hier durch eine gröbere Rasterung gelöst. Das heißt, anstatt mit Histogrammen und exakten Farbwerten zu arbeiten geht es darum, eine Farbe in eine von wenigen vorgegebenen, groben Kategorien einzuordnen. Für Ober- und Unterkörper sind dies eine Reihe typischerweise dominanter Farben: Blau, Schwarz, Braun, Rot, Grün, Weiß, Gelb, Grau. Unabhängig davon, mittels welchem der beiden Detektoren eine Person erkannt wurde, ist in jedem Fall die ungefähre Position und Größe des Kopfes im Bild bekannt. Ausgehend von dieser Information und grundlegenden anatomischen Annahmen können die Positionen und Größen von sowohl Ober- als auch Unterkörper im Bild abgeschätzt werden. Liegen diese im sichtbaren Bildbereich, kann nun das für den jeweiligen Bereich bestimmte Histogramm in die nächstgelegene Farbkategorie eingeordnet werden.
Anders als bei den bisher vorgestellten Merkmalen wurde für die Bestimmung von Ober- und Unterkörperfarben keine eigene Evaluierung durchgeführt. Die Nutzung dieser Art von Merkmalen ist vielfach Beschrieben und evaluiert zum Beispiel in [33]. Der einzige Unterschied zu üblichen Verfahren ist die gut einschätzbare Diskretisierung der Merkmalswerte.

3.3 Zeitliche Akkumulation

Über den gesamten Zeitraum in dem sich eine Person im Blickfeld einer Kamera befindet, werden in der Regel eine ganze Reihe von Detektionen möglich sein. Das heißt also, dass die Bestimmung eines Merkmals nicht anhand einer einzigen Bildaufnahme geschehen muss. Dies bietet zum einen den Vorteil, dass einzelne, schlechte Aufnahmen nicht unbedingt dafür sorgen, dass Merkmale entweder gar nicht oder sogar fehlerhaft extrahiert werden können. Andererseits stellt sich mit einer Reihe von Aufnahmen die Frage nach einer geeigneten Fusion.
Eine Betrachtung der Einzelergebnisse als gleichwertige Informationsquellen würde

Abbildung 3.3: Illustration der Qualität der Aufnahmen mit dem Verlauf der Zeit, beim passieren einer Kamera

nur funktionieren, wenn die Mehrzahl der Ergebnisse korrekt wäre. Andernfalls könnte das Ergebnis sogar schlechter sein als das eines einzelnen Frames. Es stellt sich also die Frage, ob zusätzlich Informationen verfügbar sind, die es erlauben eine Aussage darüber zu treffen, welche Beobachtungen mehr und welche weniger Sicherheit bieten. Neben detaillierten, aber sehr aufwendigen Analysen der Bildqualität einzelner Frames, bietet sich die hier verwendete Größe des detektierten Gesichtes als Indiz an, abzuschätzen wie hilfreich ein Frame für die Gewinnung eines korrekten Gesamtergebnis ist. Umso größer das Gesicht im Bild, umso näher befindet sich die Person an der Kamera, und umso wahrscheinlicher ist eine detailreiche, gut aufgelöste Aufnahme (Abbildung 3.3). Dies ist natürlich kein sicherer Indikator - auch nahe Aufnahmen können eine vielzahl von Störungen enthalten. Diese Methode ermöglicht es jedoch die besonders häufig auftretenden Detektionen aus größerer Entfernung in ihrem Einfluss zu beschränken da sie in den meisten Fällen nicht den nötigen Detaillierungsgrad aufweisen um die verwendeten Merkmale erfolgreich zu extrahieren.
Hier werden alle Aufnahmen, in denen ein Gesicht sowohl in Breite (w) als auch in Höhe (h) größer als eine Obergrenze (w_{max}, h_{max}) ist, voll gewichtet. Aufnahmen werden ausgelassen (Gewichtungsfaktor $s = 0$) falls eine der Untergrenzen w_{min} oder h_{min} unterschritten wird. Für alle Gesichtsgrößen zwischen disen Werten wird die Gewichtung linear adaptiert (Formel 3.3). In der vorliegenden Arbeit wurden die Werte $w_{max} = 120$,$h_{max} = 200$ als Obergrenzen für die Grösse des detektierten Gesichtes verwendet, sowie $w_{min} = 30$ und $h_{min} = 50$ als entsprechende Untergrenzen.

$$s = \begin{cases} 1, & w \geq w_{max} \wedge h \geq h_{max} \\ 0, & w < w_{min} \vee h < h_{min} \\ \frac{1}{2} * (\frac{w}{w_{max} - w_{min}} + \frac{h}{h_{max} - h_{min}}) & \text{sonst} \end{cases} \tag{3.3}$$

3.4 Most distinctive feature vector

Stehen zur Personenbeschreibung mehrere verschiedene Merkmale zur Verfügung, stellt sich die Frage nach der besten Kombination dieser Merkmale. Neben der Möglichkeit gleicher Berücksichtigung gibt es die Möglichkeit, die verschiedenen Merkmale anhand von zusätzlichem Wissen gewichtet zu Kombinieren. Diese Gewichtung kann je nach vorhandenen Informationen auf verschiedenen Annahmen basieren. Wenn beispielsweise bekannt ist, dass bestimmte Merkmale im Durchschnitt deutlich verlässlicher sind als andere, kann es sinnvoll sein, diese entsprechend höher zu gewichten.
In dem hier vorgestellten System geht es in erster Linie darum, die extrahierten Merkmale zu nutzen um zu entscheiden, um welche der dem System bekannten Personen es sich handelt.
Ein Merkmal ist also genau dann besonders nützlich, wenn es eine Person deutlich von möglichst vielen anderen Personen abgrenzt. Ob dies für ein Merkmal gegeben ist hängt allerdings weniger von den Eigenschaften des Merkmals selbst, sondern vielmehr von seiner Verteilung unter den Personen ab. So ist beispielsweise das Merkmal „Brillenträger“ ein sehr nützliches Merkmal wenn es nur einen Brillenträger unter den bekannten Personen gibt. Sind jedoch alle Personen Brillenträger, ist dieses Merkmal wertlos für die Abgrenzung der Personen voneinander.
Aus dieser Überlegung ergibt sich die Idee, aus den Merkmalsvektoren der bekannten Personen einen Skalierungsvektor für jede Person p abzuleiten, dessen Komponenten sich aus der Verteilung des jeweiligen Merkmals unter den restlichen Personen ergeben.
Die Skalierung σ eines Merkmals μ_i für eine Person p_j berechnet sich aus der Häufigkeit h dieses Merkmals unter allen anderen bekannten Personen $\hat{p}$. Für den Fall, dass ein Merkmal absolut eindeutig unter den Personen ist, wird dieses Merkmal mit 1 skaliert. Das andere Extrem, dass ein Merkmal für alle Personen den gleichen Wert besitzt, führt zu einer Skalierung mit 0, das Merkmal wird also nicht verwendet. Für alle anderen Fälle wird eine Skalierung zwischen diesen beiden Extrema bestimmt (Formel 3.4).

$$\sigma(\mu_i, p_j) = 1 - \frac{h(\mu_i, \hat{p})}{|\hat{p}|} \tag{3.4}$$

Aus den Skalierungen aller Merkmale ergibt sich somit der Skalierungsvektor, den wir im Folgenden als „Most distinctive feature vector (MDFV)“ (Formel 3.5) bezeichnen.

$$MDFV = \begin{pmatrix} \sigma_1 \\ \vdots \\ \sigma_n \end{pmatrix} \tag{3.5}$$

Wird nun in Folge einer Personendetektion ein Merkmalsvektor extrahiert, findet zunächst ein Vergleich mit den Merkmalsvektoren der bekannten Personen statt. Dabei werden immer die Elemente, die dem gleichen Merkmal entsprechen verglichen und aufgrund der ausschließlich diskreten Werte erhält man insgesamt einen binären Ergebnisvektor für die Vergleichsoperation mit dem Merkmalsvektor einer Person. Aus dem Skalarprodukt des Ergebnisvektors einer Person und ihres „Most distinctive feature vectors“ erhält man einen Wert für die Sicherheit, mit der die beobachtete Person mit gerader dieser Person übereinstimmt. Um zu verhindern, dass Merkmale die nicht extrahiert werden konnten, weil beispielsweise nur eine Person nicht aber ihr Gesicht detektiert wurde, einen negativen Einfluss auf diesen Wert haben, werden entsprechende Zeilen aus dem Vektor entfernt. In Formel 3.6 ist dieser Ablauf beispielhaft illustriert. Zunächst findet ein Vergleich der einzelnen Komponenten des extrahierten Vektors und dem einer bekannten Person statt. Das Ergebnis ist ein binärer Vektor bei dem bereits der Eintrag entfernt wurde, für den kein Merkmalswert bestimmt werden konnte. Das Skalarprodukt mit dem MDFV (aus dem selbiger Eintrag entfernt und anschließend normiert wurde) liefert dann die Übereinstimmungssicherheit. Durch dieses Skalarprodukt werden also die verschiedenen Übereinstimmungen oder Unterschiede zwischen den einzelnen Merkmalen in ihrer Bedeutung verstärkt oder abgeschwächt in Abhängigkeit davon, wie geeignet ein Merkmal überhaupt ist, eine Person vom Rest der Gruppe zu unterscheiden. Eine einhundertprozentige Übereinstimmung einer Beobachtung mit einer bekannten Person wäre also selbst bei Identität der Merkmalsvektoren nur dann möglich, wenn jedes der Merkmale der betroffenen Person eindeutig unter der Gruppe der bekannten Personen wäre - was aufgrund binärer Merkmale wie „Brillenträger“ oder „Geschlecht“ selten möglich ist.

$$\begin{pmatrix} 1 \\ 0 \\ unknown \\ weiss \\ blau \end{pmatrix} \longleftrightarrow \begin{pmatrix} 0 \\ 0 \\ blond \\ blau \\ blau \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} * \begin{pmatrix} 0.1 \\ 0.5 \\ 0.2 \\ 0.2 \end{pmatrix} = 0.7 \tag{3.6}$$

3.5 Detektion & Tracking

In diesem Abschnitt sollen die beiden verwendeten Detektionsverfahren näher vorgestellt werden. Wie aus den Beschreibungen der verwendeten Merkmale hervorgeht, setzen alle eine Detektion der Person im Bild, einige sogar die genaue Position und Größe des Gesichtes voraus.
Um die für die Merkmalsextraktion notwendigen Bildbereiche zu bestimmen, wird zum einen eine auf Haar-Kaskaden basierende Gesichtsdetektion verwendet und zum anderen ein Omega-Shape-Detektor. Der Grund für den Einsatz verschiedener

Detektoren ist die Bedeutung der Detektionen für das Gesamtsystem: Den Großteil der Zeit befinden sich die Personen außerhalb eines Blickfeldes einer Kamera. Da in dieser Zeit keine Beobachtungen gemacht werden können, ist es besonders wichtig, in den meist kurzen Zeiträumen in denen eine Person eine Kamera passiert dies auch zu beobachten. Die Omega-Shape-Detektion ergänzt die Gesichtsdetektion, da sie auch aus für die Gesichtsdetektion unvorteilhaften Blickwinkeln zumindest die Anwesenheit einer Person erkennen kann.

3.5.1 Gesichtsdetektion

Einige der vorgestellten Merkmale setzen eine Gesichtsdetektion voraus, da sie ausgehend von einer frontalen Gesichtsaufnahme spezifische Bereiche untersuchen. Eine zuverlässige und schnelle Methode zu Gesichtsdetektion ist der Einsatz von Objektdetektoren [34], die auf Gesichtern trainiert wurden. Das Training besteht in der automatischen Analyse von hunderten Ansichten des zu trainierenden Objektes - so genannte positive Beispiele - und einer ebenfalls großen Zahl von Ansichten die das Objekt nicht enthalten - so genannte negative Beispiele. Einfache Objekte werden dabei durch die Kombination vieler grundlegender Muster, so genannter Haar-Features (Abbildung 3.4), repräsentiert. Jedes einzelne dieser Features repräsentiert eine bestimmte Eigenschaft. Beispielsweise sind die Augenpartien dunkler als die dazwischenliegende Nasenpartie. Das nicht Vorhandensein eines Merkmals schließt bereits fast aus, dass es sich bei dem untersuchten Bereich um ein Gesicht handeln könnte. Auch das Vorhandensein eines Merkmals bietet allein keine Auskunft darüber, ob man ein Gesicht gefunden hat. Erst das Auffinden vieler Merkmale liefert Sicherheit. Diese Aneinanderreihung von schwachen Klassifikatoren bezeichnet man als Klassifikatorkaskade, in diesem Fall “Haar-Kaskade”. Haar-Kaskaden können auch für andere Bereiche benutzt werden, wie dem Oberkörper, aber die Verwendung mehrerer Kaskaden treibt die Rechenlast sehr in die Höhe. Der Grund hierfür liegt in der üblichen Arbeitsweise der Kaskaden - es wird jeweils in jedem Bild, zumindest in einem bestimmten Bereich, jede Position und Skalierung auf das Vorhandensein des gesuchten Objektes (hier des Gesichtes) untersucht.

3.5.2 Omega Shape Detektion

Je nach dem wie eine Kamera in den Räumlichkeiten installiert ist, kann es einfach oder sehr schwer sein frontale Aufnahmen des Gesichtes zu erhalten. Der Erfolg hängt darüber hinaus natürlich auch von der Kooperation der jeweiligen Personen ab. Da die Beobachtungen an den einzelnen Kameras die einzigen Informationen sind, die dem System zu Verfügung stehen, ist es wichtig dass eine Person nicht unerkannt bleibt wenn sie das Blickfeld einer Kamera passiert. Um dies auch unter den genannten Einschränkungen für die Gesichtsdetektion zu gewährleisten, wird

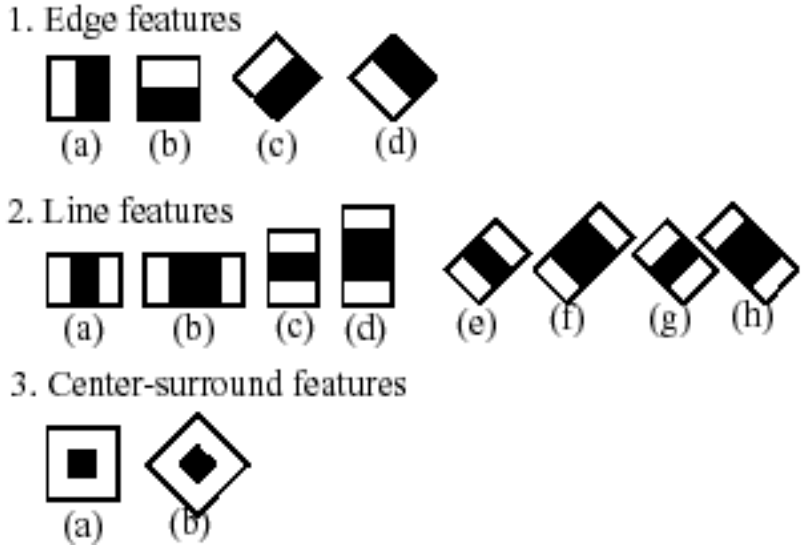


Abbildung 3.4: Beispiele für Haar-Features

zusätzlich ein "Omega Shape Detektor" eingesetzt. Dieser basiert auf der Idee, dass die Silhouette von Kopf und Schultern eine distinktive Form besitzt, die einem Ω ähnlich sieht. Der Vorteil gegenüber der Gesichtsdetektion liegt darin, dass Personen auch erkannt werden können, wenn die Aufnahmequalität nicht sehr gut ist oder die Person nicht in die ungefähre Richtung der Kamera schaut.
Die Personendetektion auf diesem Wege geschieht in einigen Schritten, die im Folgenden Illustriert werden. Stationäre Kameras vorrausgesetzt, gibt es die Möglichkeit anhand von Referenzaufnahmen des "Hintergrundes", also dem Sichtfeld der Kamera ohne Personen, und dessen Subtraktion von der aktuellen Aufnahme (Abbildung 3.5 a) den Vordergrund zu bestimmen (Abbildung 3.5 b). Der so gewonnene "Vordergrund" besteht zunächsteinmal aus allen Bildbereichen, die sich von der Referenzaufnahme unterscheiden. Da dies beliebige Objekte - nicht nur Personen - sein können, ist eine weitere Analyse des Vordergrundes notwendig. Im ersten Schritt werden durch Clustern der Vordergrundpixel größere, zusammenhängende Bereiche im Vordergrund ausgemacht. Da für die weitere Analyse lediglich die Umrandung der Objekte von Interesse ist, können diese als Polygon dargestellt werden (Abbildung 3.5 c). Die Darstellung als Polygon erlaubt auch eine Untersuchung der Lage seiner Eckpunkte darauf hin, ob eine Folge dieser Punkte eine Omegaform beschreibt. Da man in aller Regel davon ausgehen kann, das der oberste Punkt eines Vordergrundobjektes dem höchsten Punkt einer abgebildeten Person entspricht, bietet es sich an, an diesem Punkt (*Tip*) anzusetzen, um die Form der Silhouette zu Untersuchen. Wenn ein Teil der Eckpunkte eine Omegaform beschreibt, ist zu erwarten, dass sich von dem obersten Eckpunkt ausgehend zu beiden Seiten an etwa gleichen, relativen Positionen weitere Eckpunkte befinden. Um diese Punktpaare zu bestimmen, wird wie folgt vorgegangen: Von dem Tip ausgehend wird der nächstgelegene Eckpunkt gesucht. In gleicher Entfernung in y

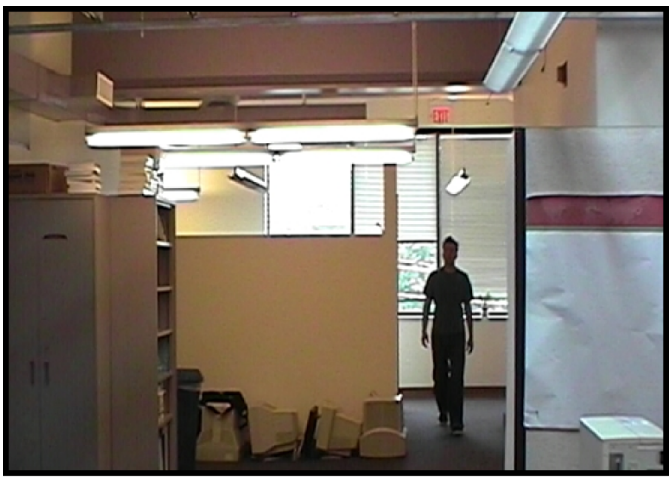
Richtung und gleicher Entfernung in *negativer* x Richtung ist der korrespondierende Eckpunkt zu erwarten. Kann dieser innerhalb eines gewissen Toleranzbereichs (Abbildung 3.6 a) bestimmt werden ist das erste Punktepaar gefunden und die Suche setzt sich mit dem nächsten Eckpunkt fort.
Um zum einen ein Abbruchkriterium festzulegen und zum anderen zu vermeiden, dass beliebige symmetrische Formen erkannt werden ist eine weitere Auswertung der gefundenen Punktepaare (Abbildung 3.6 b) nötig. Dazu werden die Abstände der Punktepaare untersucht. Handelt es sich um einen Kopf und daran anschließenden Oberkörper sollten - im Rahmen eines Toleranzbereiches - die Abstände zunächst leicht zunehmen, dann leicht abnehmen und dann in einem sehr kleinen y Achsenabschnitt stark zunehmen. Ist diese Charakteristik zu einem ausreichenden Grad erfüllt, wird das erkannte Vordergrundobjekt als Person angenommen.

3.5.3 Tracking

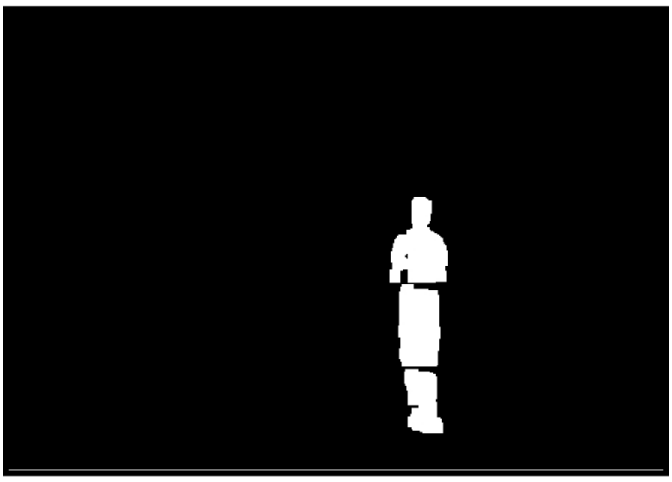
Obwohl das eingesetzte Verfahren zur Gesichtsdetektion verhältnismäßig performant ist, bietet es sich an, nach einer ersten Detektion nicht mehr das gesamte Bild zu analysieren, um das bereits detektierte Gesicht in den folgenden Frames wiederzufinden. Im Vordergrund steht hierbei aber nicht eine Beschleunigung der Bildanalyse, sondern das lokale Tracking im Blickfeld der Kamera soll helfen, Detektionen verschiedenen Personen zuzuordnen wenn sich mehrere im Bild befinden.
Das Wissen darüber, dass sich in dem von der Detektion gelieferten Rechteck das Gesicht, also im wesentlichen Hautfarbe befindet, wird genutzt um ein Histogramm aufzubauen, das Hautfarbe repräsentiert. Der verwendete Farbraum ist der HSV-Raum (Abbildung 3.7), in dem die Farbe eines Pixels durch den Farbton (**H**ue), die Sättigung (**S**aturation) und die Helligkeit (**V**alue) repräsentiert wird, was eine bessere Anpassung an unterschiedliche Farbintensitäten ermöglicht.

Zur Optimierung des Histogramms werden noch zwei weitere Schritte durchgeführt: Zunächst wird der Bereich aus dem das Histogramm erstellt wird auf die Hälfte des ermittelten Rechtecks beschränkt, um sicherzustellen, dass der Bereich keinen Hintergrund enthält. Weiterhin gilt für das Histogramm H und einen gegebenen Farbwert x: $H(x) = P(x|Person)$. Durch Anwendung von Bayes' Regel erhält man $\frac{P(Person|x)}{P(\neg Person|x)} \sim \frac{P(x|Person)}{P(x|\neg Person)} = \frac{H(x)}{H_{neg}(x)}$ wobei H_{neg} ein Histogramm ist, das gerade nicht die Farben der Person, sondern die des Hintergrundes modelliert. Daher wird das Vordergrundhistogramm durch ein weiteres Histogramm bin-weise dividiert ($H_{res} = \frac{H}{H_{neg}}$), welches aus dem Gesamtbild erstellt wird. So werden Histogrammwerte, die im Hintergrund stark vertreten sind, abgeschwächt und das Histogramm wird so stark wie möglich vom Hintergrund abgegrenzt.

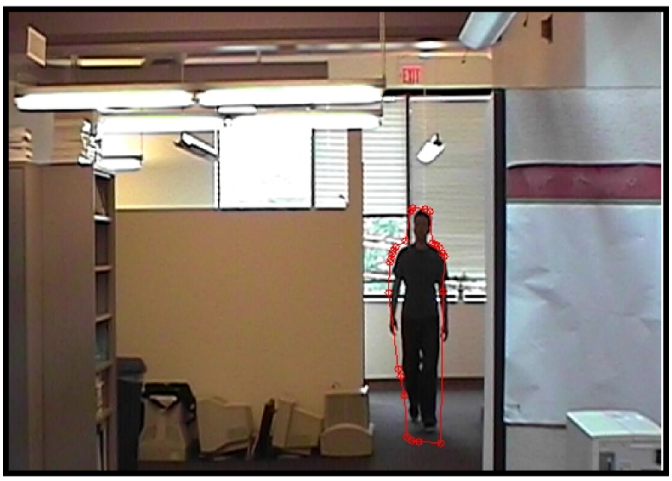
Das resultierende Histogramm wird im folgenden Trackingschritt verwendet, um



(a)



(b)



(c)

Abbildung 3.5: Extraktion der Silhouette einer Person

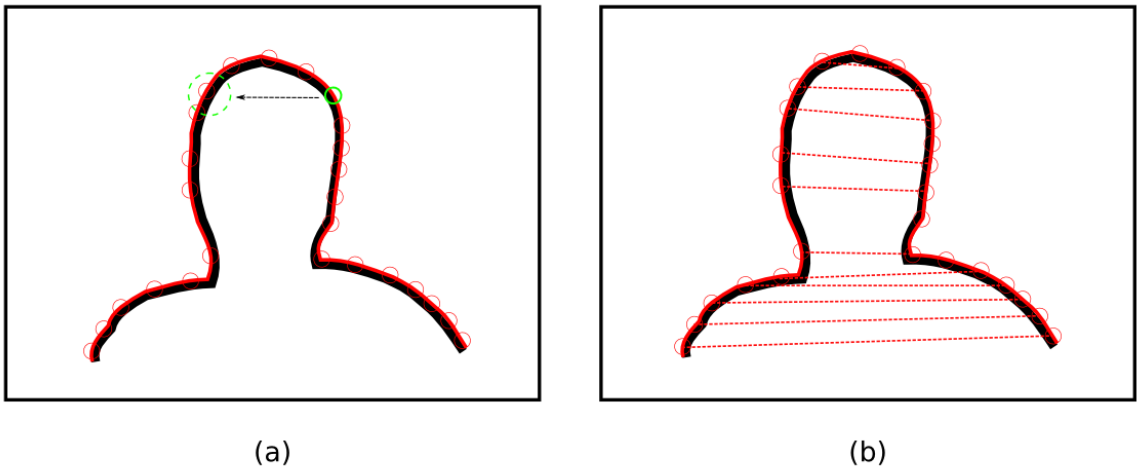


Abbildung 3.6: Schematische Darstellung der Silhouettenanalyse

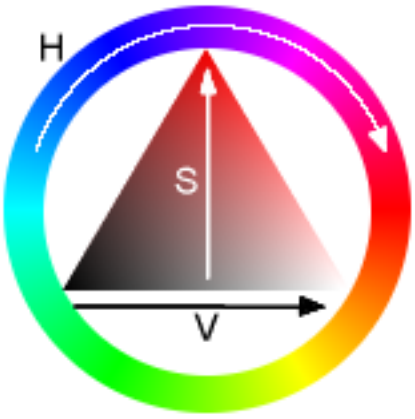


Abbildung 3.7: Komposition einer Farbe im HSV-Farbraum

festzustellen wo im nächsten Bild das Gesicht der Person zu finden ist. Dazu wird für jedes Pixel im Bild anhand seiner Farbe bestimmt, wie wahrscheinlich es ist, dass es Haut darstellt. Um nun die wahrscheinlichste Position des Gesichtes zu bestimmen, wird der Mean-Shift Algorithmus [35] eingesetzt um die Ansammlung von hautfarbenen Pixeln zu finden, die ihrer Größe nach und unter Berücksichtigung der vorherigen Position am ehesten das Gesicht darstellt.

Wie in Trackingsystemen üblich, gilt es die richtige Balance zwischen Fehldetektionen ("misses") und falschen Treffern ("false positives") zu finden indem die Parameter des Trackingverfahrens angepasst werden. Um die Fehlerquelle später im probabilistischen Modell möglichst genau definieren zu können, wurde der hier entwickelte Tracker daraufhin optimiert, möglichst keine false positives zu generieren. Trotz dieser restriktiven Ausrichtung erreicht der Tracker insgesamt gute Detektionsergebnisse. Lediglich für den Fall, dass mehrere Personen gleichzeitig im Bildbereich zu sehen sind, kann es dazu kommen, dass alle Detektionen einer Person zugeordnet werden, und eine zweite Person daher nicht detektiert wird. Genau dieser Fall der Nicht-Detektion wird daher in der folgenden Beschreibung des probabilistischen Modells, sowie in der Evaluierung des Gesamtsystems besonders berücksichtigt.

4 Probabilistisches Trackingmodell

Die bisher beschriebenen Verfahren können zwar verwendet werden, um Personen anhand einer Kombination von Merkmalen wiederzuerkennen, aber das allein genügt nicht um Aussagen über die Aufenthaltswahrscheinlichkeiten aller Personen zu einem gegebenen Zeitpunkt zu machen. Die Identifikation der Personen an einzelnen Kameras bietet lediglich lokale und unsichere Informationen, die in unregelmäßigen zeitlichen Abständen verfügbar sind. Um aus diesen Einzelbeobachtungen eine globale Aussage über die wahrscheinlichsten Aufenthaltsorte einzelner Personen abzuleiten, ist eine Analyse der Information auf einer übergeordneten Ebene notwendig.
Weder die Detektion noch die Identifikation einer Person ist mit absoluter Sicherheit möglich. Jede Beobachtung, die als Hinweis auf den aktuellen Aufenthaltsort einer Person dient, ist also fehlerbehaftet. Es ist daher sinnvoll im Falle einer neuen Beobachtung die bisherige Annahme über den Aufenthaltsort nicht gleich zu verwerfen, sondern beide Möglichkeiten weiterhin in Betracht zu ziehen.
Über die Zeit betrachtet, sollte idealerweise also jeder mögliche nächste Schritt mit zugehöriger Wahrscheinlichkeit vorgehalten werden. Die Berechnung der Wahrscheinlichkeiten aller so entstehenden, möglichen Wege von Personen im Kameranetz besitzt allerdings für $|S|$ Zustände und t Zeitschritte eine Komplexität von $\mathcal{O}(|S|^t)$ und muss in jedem Zeitschritt erneut durchgeführt werden. Ein sehr ähnliches Problem wurde bereits in Kapitel 2 im Zusammenhang mit dem Evaluierungsproblem der Hidden Markov Modelle (HMMs) beschrieben. Wenn sich das hier vorliegende Problem zur Berechnung der möglichen Wege mit Hilfe eines Hidden Markov Modell beschreiben ließe, wäre es möglich den bereits vorgestellten Forward Algorithmus zur Lösung des Problems einzusetzen und somit die Komplexität auf $\mathcal{O}(t * |S|^2)$ zu reduzieren.
Im Folgenden soll es nun darum gehen den Aufbau eines HMMs zu erläutern, dass die probabilistisch relevanten Parameter des Trackingsystems modelliert um so eine effiziente Lösung des eingangs beschriebenen Problems zu ermöglichen. Während sich Abschnitt 4.1 mit der grundlegenden Modellierung beschäftigt, wird in den Abschnitten 4.2 bis 4.5 detailliert auf die Bestimmung der einzelnen Parameter eingegangen.

4.1 Aufbau

Die Zustandsmenge bildet zusammen mit den Übergangswahrscheinlichkeiten eine wichtige Grundlage zum Aufbau eines HMMs. Von den Übergangswahrscheinlichkeiten sind zunächst lediglich solche relevant die größer als 0 sind, denn durch diese wird zusammen mit den Zuständen die Netztopologie eines HMMs definiert. Anders als in vielen anderen Anwendungsgebieten in denen HMMs eingesetzt werden, bietet es sich hier weder an Standardtopologien zu verwenden, da es sich um kein Standardproblem für HMMs handelt, noch ist es möglich eine Topologie zu lernen, da keine Trainingsdaten vorhanden sind. Dennoch lässt sich aus den verfügbaren Informationen über das Kameranetzwerk eine Topologie für ein HMM ableiten: Modelliert man die die Kameras, so wie die Wege zwischen ihnen jeweils als Zustände im HMM, repräsentiert die gewonnene Zustandsmenge die möglichen Aufenthaltsorte einer Person im Kameranetzwerk. Die konkreten Werte für die Übergangswahrscheinlichkeiten werden in Abschnitt 4.5 erläutert, zunächst sei lediglich festgelegt, dass nur direkte Verbindungen zwischen einer Kamera und einem Weg (also Übergänge zwischen einem Zustand der eine Kamera repräsentiert und einen Zustand der einen Weg repräsentiert) eine Übergangswahrscheinlichkeit $t > 0$ besitzen. Das so gewonnene Netz - also die Topologie des HMM - spiegelt die räumlichen Gegebenheiten des realen Kameranetzwerks wieder und bildet so eine gute Voraussetzung zu seiner Modellierung.
Die nächste Komponente die für Berechnungen mittels eines HMMs und des Forward Algorithmus notwendig ist, sind die Ausgaben. Diese stellen den beobachtbaren Teil eines HMMs dar, entsprechen also den Informationen die zur Lösung des Problems konkret zur Verfügung stehen. Im Falle des realen Kameranetzwerk können Personen an den einzelnen Kameras beobachtet werden. Diese Einzelbeobachtungen sind also der Teil des Systems, der bekannt ist. Für HMMs ist es üblich die Menge aller theoretisch möglichen Beobachtungen oder Ausgaben in einem Ausgabealphabet V zusammenzufassen. Um dieses Ausgabealpabet für das hier zu modellierende System angeben zu können gilt es zunächst die Beobachtungen zu untersuchen um festzustellen welche Beobachtungen prinzipiell möglich sind.
Eine Beobachtung - also ein Element des Ausgabealphabets - setzt sich aus dem Beobachtungstyp, dem der Kamera, an dem die Beobachtung gemacht wurde, entsprechende Zustand und dem Personenvektor zusammen, der wie folgt definiert ist: Wenn eine Person im Blickfeld einer Kamera detektiert wird, findet eine Identifikation der Person statt. Da diese selten eindeutig ist, wird ein Personenvektor erstellt, dessen Einträge jeweils aus der Identität einer Person und der zugehörigen Konfidenz, dass gerade diese Person beobachtet wurde, bestehen. Die Beschreibung der Notwendigkeit für verschiedene Beobachtungstypen und eine Erklärung dieser findet sich in Abschnitt 4.3.
Nun da eine Menge von möglichen Ausgaben bestimmt ist, stellt sich die Frage nach der Wahrscheinlichkeit mit der die einzelnen Ausgaben beobachtet werden können. Diese Emissionswahrscheinlichkeiten werden in den allermeisten Fällen

gelernt. Da hier wie bereits erwähnt keine Trainingsdaten zur Verfügung stehen müssen die Wahrscheinlichkeiten anders bestimmt werden. Konkret muss für jeden möglichen Zustand ($s_1 \in S$) bestimmt werden, wie Wahrscheinlich es dort ist, die Beobachtung $v \in V$ zu machen. Überlegt man sich woher die Beobachtungen stammen und welche Information sie enthalten, wird klar, dass aufgrund dieser Information bereits gewisse Aussagen über die Emissionswahrscheinlichkeiten möglich sind. Eine Beobachtung die an Kamera $C1$ gemacht wurde, wird sich im Modell sehr viel wahrscheinlicher in dem Zustand auswirken, der diese Kamera modelliert als in jedem anderen. Dies ist nur ein einfaches Beispiel für die Ableitung der Emissionswahrscheinlichkeit aus den vorhandenen Informationen. Für die endgültige Bestimmung aller Emissionswahrscheinlicheiten gibt es eine große Zahl von verschiedenen Berechnungswegen, die in Abschnitt 4.4 ausführlich vorgestellt werden. Die Berechnung der Emissionswahrscheinlichkeiten ist genauso wie die Bestimmung mittels Trainingsdaten eine Schätzung der idealen Werte, bietet jedoch neben der Unabhängigkeit von Trainingsdaten den Vorteil, dass Emissionswahrscheinlichkeiten noch zur Laufzeit angepasst werden können. Diese Eigenschaft wird besonders für den Fall mehrerer Trackingziele ausgenutzt und wird im Folgenden sowie in Abschnitt 4.4 näher erläutert.

Auch wenn bis hierher alle notwendigen Komponenten vorgestellt wurden, die notwendig sind, um das Trackingproblem in einem Kameranetzwerk mittels eines HMM zu lösen, ist ein Teilaspekt bisher außen vorgelassen worden: Das gleichzeitige Tracken mehrerer Personen. Ein HMM so wie es bisher beschrieben wurde und unter Anwendung des Forward Algorithmus, liefert eine Wahrscheinlichkeitsverteilung über die Menge der Zustände und damit den wahrscheinlichsten Aufenthaltsort für *eine* Person. Würde man eine Modellierung der Aufenthaltsorte *aller* in nur einem HMM erreichen wollen, müsste die Struktur darauf angepasst werden. Die Modellierung der Aufenthaltsorte alle Personen in einem einzigen HMM steigert aber nicht nur deutlich die Komplexität, sondern führt auch Abhängigkeiten zwischen den Personen ein was nicht ihrem realen Verhalten entspricht.
Die gleichzeitige Lokalisierung aller Personen wird daher durch die folgende Modellierung gelöst: Zunächst wird für jede Person (P_n) ein eigenes HMM (M_n) initialisiert ganz nach der oben beschriebenen Vorgehensweise. Da die Personen sich in der Realität voneinander unabhängig bewegen, ist diese Trennung legitim. In jedem HMM werden also zunächst unabhängig voneinander die Positionen der Personen bestimmt. Wie bereits beschrieben sind dazu die beobachtbaren Ausgaben des Systems notwendig. An dieser Stelle ist es daher wichtig zu verstehen wie diese für die einzelnen Modelle zustandekommen. Wenn eine Person im realen Kameranetzwerk beobachtet wird, werden Merkmale extrahiert und damit die Identität der beobachteten Person bestimmt. Da diese Identifikation selten eindeutig ist, weiß man zunächst lediglich das die gemachte Beobachtung mit jeweils einer bestimmten Konfidenz einer bestimmten Person zuzuordnen ist. Für die HMMs, die den Aufenthaltsort einer Person modellieren, deren Übereinstimmungskonfidenz

bei der Identifikation größer Null war, wird die aktuelle Beobachtung o_t der Folge von Beobachtungen O hinzugefügt. Das Hauptproblem mit dem die einzelnen HMMs also zu kämpfen haben, ist die Unterscheidung zwischen Beobachtungen, die von der Person ausgelöst wurden, deren Position sie modellieren und denen, die von einer anderen augelöst wurden.
Bei der Lösung dieses Problems kann es sehr Hilfreich sein, wenn etwas über die wahrscheinlichen Aufenthaltsorte anderer Personen bekannt ist. Wenn es beispielsweise sehr unwahrscheinlich ist, dass eine andere Person als Person P_1 sich in dem Zustand (also an der Kamera oder dem Weg) befindet an dem die Beobachtung gemacht wurde, dann ist es sehr wahrscheinlich das gerade Person P_1 beobachtet wurde. Um diese Informationen ausnutzen zu können, müssen die HMMs ihre Informationen über den Aufenthalt der jeweiligen Personen untereinander austauschen. Dazu werden nach jeder neuen Beobachtung die vorhergesagten Wahrscheinlichkeitsverteilungen über die Zustände aller HMMs zur Auswertung der Beobachtung genutzt.

4.1.1 Prädiktion und Aktualisierung

Wie durch den Forward Algorithmus vorgegeben, beinhaltet das Trackingverfahren einen Prädiktions- und einen Aktualiersierungsschritt. Im Aktualisierungsschritt wird dafür gesorgt, dass die Modellierung des Systemzustandes an die sich stetig vergrößernde Liste von Beobachtungen angepasst wird. Vom Prädiktionsschritt hingegen wird gefordert auf Grundlage der aktuellen Modellierung des Systems eine Aussage über den wahrscheinlichsten Aufenthaltsort einer Person im nächsten Zeitschritt zu machen. Zunächst soll der Aktualisierungsschritt im Hinblick auf die Reaktion auf Veränderungen im realen System beschrieben werden: Wenn eine Person das Blickfeld einer Kamera betritt wird sie mit einer gewissen Wahrscheinlichkeit detektiert. Wenn sie detektiert wird, beginnt als nächstes die Extraktion der Merkmale aus denen dann ein Merkmalsvektor aufgebaut wird. Nun gilt es den extrahierten Merkmalsvektor zur Identifikation der beobachteten Person einzusetzen. Dazu findet ein Vergleich des extrahierten Merkmalsvektors mit denen der bekannten Personen statt. Die Ergebnisvektoren dieses Vergleichs werden im Anschluss mit Hilfe des “most distinctive feature vector” gewichtet. Abhängig von der Qualität der Merkmalsextraktion erhält man somit verschiedene Übereinstimmungen für jede der Personen. Im Idealfall erhält man die größte Übereinstimmung für die Person die tatsächlich beobachtet wurde und geringe oder gar keine Übereinstimmungen mit allen anderen Personen.
Nun wird in allen HMMs M_n die Liste der Beobachtungen O um die aktuelle Beobachtung o_t erweitert, vorausgesetzt die Korrespondierende Person P_n erhielt bei der Identifikation eine Übereinstimmungskonfidenz größer Null. Anhand der Beobachtung können nun in allen Modellen die Emissionswahrscheinlichkeiten wie in 4.4 beschrieben berechnet werden.

Unter Verwendung der Emissionswahrscheinlichkeiten und den bisherigen Aufenthaltswahrscheinlichkeiten werden diese mittels des Forward Algorithmus nun an den neuen Informationsstand - also die Erweiterung des Wissens (O) um die neue Beobachtung (o_t) - angepasst.

Der Prädiktionsschritt unterteilt sich in zwei Aufgaben. Zum einen soll hier eine Ausgabe des wahrscheinlichsten Aufenthaltsorts für jede Person stattfinden, zum anderen soll das Wissen über die Aufenthaltsorte aller Personen unter den Modellen ausgetauscht werden. Das direkt ablesbare Ergebnis zu jedem Zeitpunkt in einem HMM ist die Wahrscheinlichkeitsverteilung über die Menge der Zustände. Genau diese Wahrscheinlichkeitsverteilung jedes HMMs wird auch beim Aktualisierungsschritt aller personenbezogenen HMMs verwendet, da sie die durch das Modell bestmögliche Beschreibung des Systemzustandes repräsentieren. Für eine spätere Evaluation sowie für viele Anwendungen ist es dennoch wünschenswert eine klare Aussage bezüglich des Aufenthaltsorts einer Person zu machen. Im Rahmen dieser Arbeit wird diese "harte" Entscheidung getroffen, indem der Zustand mit der höchsten Wahrscheinlichkeit ausgewählt wird. Es muss klar sein, das eine solche "harte" Entscheidung im Falle eines Fehlers auch "hart" bestraft wird. Sind zum Beispiel zwei Zustände nahezu gleichwahrscheinlich, beschränkt sich die finale Aussage vollständig auf den geringfügig wahrscheinlicheren Zustand, was offensichtlich eine schlechte Repräsentation des modellierten Systemzustands zur Folge hat.

4.2 Topologie

Für den Aufbau einer HMM Netztopologie gibt es eine vielzahl von Möglichkeiten. Neben zahlreichen etablierten Modellen für diverse spezielle Anwendungen ist es oftmals üblich die Topologie anhand von Trainingsdaten zu bestimmen [25]. In dem vorliegenden Problem des Personentrackings in Kameranetzwerken bietet es sich allerdings an, sich an den Gegebenheiten und Beschränkungen der realen Einsatzumgebung zu orientieren. Die zwei wesentlichen Komponenten aus denen ein Netz besteht, sind zum einen die Knoten und zum anderen die Kanten um diese miteinander zu Verbinden. Hier wird eine Parallele zu dem realen Kameranetzwerk deutlich: Das reale Kameranetzwerk besteht aus Kameras die an verschiedenen Orten positioniert sind und es gibt unterschiedliche Wege, die diese Orte miteinander verbinden. Da gerade diese Wege im Fokus des Personentrackers liegen bietet es sich an die physikalischen Gegebenheiten des realen Kameranetzes auf die Modellierung zu übertragen.
Auf diesem Weg wird erreicht, dass allen möglichen Aufenthaltsorten und vor allem ihren Beziehungen zueinander in der Modellierung Rechnung getragen wird. Die Abbildung 4.2 zeigt die Struktur eines HMMs, also die Modellierung eines Kameranetzwerkes innerhalb einer Büroumgebung mit sechs Kameras zwischen

denen sich die Personen auf bestimmten Wegen bewegen können. Betrachtet man die Konnektivität des Netzes kann man sehen, dass nur manche Kameras direkt miteinander Verbunden sind, ganz wie es durch architektonische Gegebenheiten für die reale Umgebung des Kameranetzes der Fall ist.

In Abbildung 4.2 werden Zustände die einen Weg zwischen zwei Kameras repräsentieren mit W_{ij} bezeichnet. Eine Besonderheit stellen dabei die Zustände W_{11} und W_{66} dar: Aufgrund der räumlichen Gegebenheiten ist es an diesen Stellen möglich, dass die Person sich in einem Bereich außerhalb des Blickfeldes der Kamera aufhält ohne das dieser Weg zwangsläufig zu einer anderen Kamera führt, wie es für andere Wege der Fall ist (vergleiche Abbildung 6.1). Für die Repräsentation der Kameras im realen Kameranetzwerk, wird in der Modellierung unterschieden ob eine Person an der korrespondierenden Kamera detektiert wurde. Hat eine Person eine Kamera betreten und wurde detektiert, so wird dies über einen Zustand mit der Bezeichnung C_i modelliert. Ist die Person nicht detektiert worden, obwohl sie sich an der Kamera befindet, wird dies durch einen „NichtDetektions“ - Zustand mit der Bezeichnung N_i repräsentiert. Der Grund für diese Unterscheidung ergibt sich nicht aus Überlegungen zur Topologie, sondern ist notwendig um eine von dem vorherigen Geschehen abhängige Fallunterscheidung zu ermöglichen. Um zu verstehen, warum eine Fallunterscheidung überhaupt notwendig ist, ist eine nähere Betrachtung der Detektion im realen Kameranetzwerk und den damit zusammenhängenden Beobachtungen nötig.

4.3 Beobachtungstypen

Um die gewünschten Berechnungen mittels des Forward-Algorithmus innerhalb des Netzes anstellen zu können, ist neben der Topologie auch noch die Definition eines Ausgabealphabets nötig, also die Festlegung aller Ausgaben die das Modell erzeugen kann. In den meisten Fällen sind diese Ausgaben das, was man von einem unbekannten Systemzustand beobachten kann. Im Fall des Kameranetzwerks bieten sich hierfür die Beobachtungen an den einzelnen Kameras an, denn sie sind der einzige Hinweis auf den Gesamtzustand des Systems, also die Positionen der Personen.
Es gilt also zu definieren, welche Beobachtungen an den Kameras möglich sind. Dazu ist es zunächst notwendig, den Informationsgewinn der Detektionen zu untersuchen, die mittels der in Abschnitt 3 beschriebenen Verfahren ermittelt werden. In den meisten Trackingsystemen ist vor allem die initiale Detektion interessant. Auch hier hat diese eine zentrale Bedeutung, denn aus ihr wird der Eintritt einer Person in das Blickfeld einer Kamera abgeleitet. Beobachtungen, die durch diese initiale Detektion zustandekommen werden als Typ 1 Beobachtungen bezeichnet und stellen damit die erste von drei Kategorien von Beobachtungen dar.

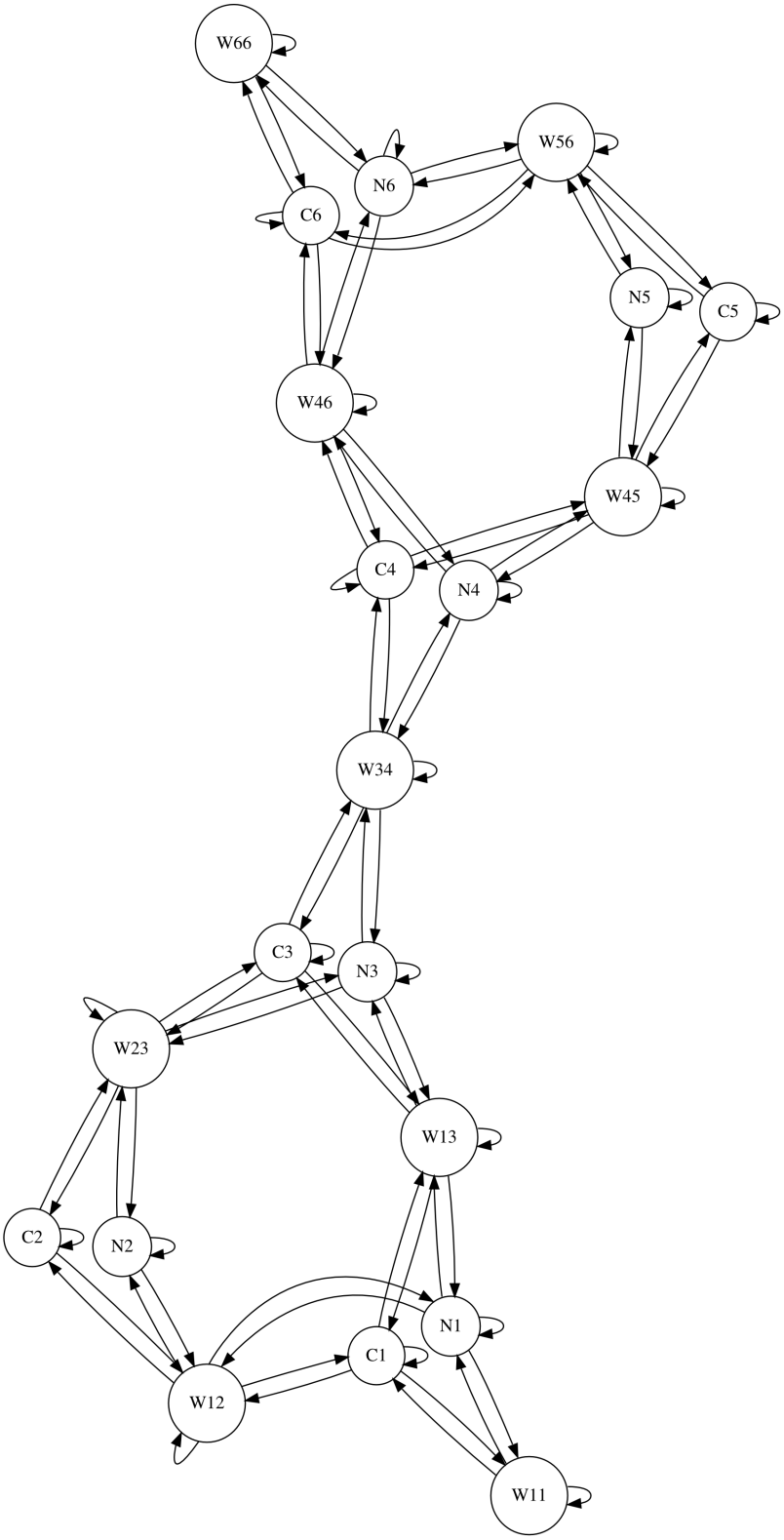


Abbildung 4.1: Beispiel für eine Netztopologie mit 6 Kameras. C1-C6 sind Zustände, die Kameras repräsentieren, W11-W66 repräsentieren Wege zwischen den Kameras und N1-N6 sind „NichtDetektions“ - Zustände

Eine weitere wichtige Beobachtung ist der Verlust eines Tracks. In der Regel passiert dies, wenn eine Person das Blickfeld einer Kamera verlässt und ist somit ein Hinweis darauf, dass sich der Aufenthaltsort der Person geändert hat. Beobachtungen dieser Art werden als Typ 2 Beobachtungen bezeichnet. Wenn man sich klar macht, wie also diese beiden Beobachtungstypen zustandekommen, wird deutlich dass eine Beobachtung vom Typ 2 von einer vorhergehenden Typ 1 Beobachtung abhängig ist: Wird eine Person nicht Detektiert, so kann auch kein Track dieser Person verloren gehen. Im Zusammenhang damit, lässt sich nach der Einführung des letzten Beobachtungstyps, Typ 0, dann auch die Notwendigkeit der in Abschnitt 4.2 erwähnten Fallunterscheidung erklären:
Beobachtungen vom Typ 0 stellen eine Besonderheit dar, da sie nicht direkt durch Tracking-Komponenten im realen Kameranetzwerk ausgelöst werden, sondern künstlich generiert werden. In einem HMM finden nur dann Änderungen statt, wenn sich die Ausgaben ändern bzw. neue Beobachtungen hinzu kommen. Dies würde bedeuten, dass eine aktuelle Prädiktion des wahrscheinlichsten Systemzustandes nur gemacht werden kann wenn eine Person real detektiert wurde. Da aber die Detektion nicht immer fehlerfrei arbeitet, kann eine Person unter Umständen eine Kamera passieren ohne dass dies detektiert wird. Um diesen Umstand modellieren zu können werden also mit einer gewissen "Taktung" Beobachtungen vom Typ 0 generiert die Aussagen, dass sich im System zwar seit der letzten Beobachtung nichts geändert hat aber unter gewissen Voraussetzungen dennoch eine Zustandsänderung stattgefunden haben kann. Als eine geeignete Taktung bietet sich zunächst die Framerate an, dies führt jedoch zu einem deutlich höherem Rechenaufwand, ohne dass die erzielten Ergebnisse diesen Aufwand rechtfertigen würden. Der geeignete Takt wurde daher empirisch bestimmt und auf 5 Sekunden festgelegt, er lässt zwischen zwei Beobachtungen von Typ 1 oder 2 genügend Spielraum, Zustände die durch eventuelle Fehldetektionen verursacht wurden, zu durchlaufen.

Es gibt also drei grundlegende Beobachtungstypen:

- Typ 0 : Seit der letzten gemachten Beobachtung hat sich nichts geändert
- Typ 1 : Eine Person betritt das Blickfeld einer Kamera
- Typ 2 : Eine Person verlässt das Blickfeld einer Kamera

Diese Zustandsänderungen durch Beobachtungen vom Typ 0 erklären die Notwendigkeit der eingeführten Fallunterscheidung zwischen Zuständen die eine Kamera mit Detektion modellieren (C_i) und den „NichtDetektions“ - Zuständen (N_i) die die gleiche Kamera ohne Detektion modellieren. Ist eine Person beim betreten des Blickfeldes einer Kamera detektiert worden, so ist aufgrund der erwähnten Abhängigkeit zwischen Eintritt und Austritt ein Verlassen dieses Zustandes nur durch eine Beobachtung vom Typ 2 möglich, nicht aber durch eine Beobachtung

vom Typ 0. Genau umgekehrt verhält es sich für den Fall dass eine Person beim Eintritt nicht detektiert wurde, denn in diesem Fall kann auch kein Austritt detektiert werden. Die Modellierung dieser Fallunterscheidung erfordert ein Zwischenspeichern eines Teil der Historie und wird erreicht indem eine Kamera im realen Netz durch zwei Zustände (Typ C und Typ N, siehe auch Abbildung 4.2) im Modell repräsentiert wird.

Neben der Kategorisierung in Beobachtungstypen ist es noch wichtig festzustellen, wo im Kameranetz eine Beobachtung gemacht wurde. Prinzipiell können Beobachtungen an Zuständen gemacht werden die Kameras modellieren. Das konkrete Ausgabealphabet setzt sich also aus allen möglichen Kombinationen der Menge der Beobachtungstypen und der Menge der Zustände zusammen. Beispiele für Beobachtungen nach diesem Schema (Beobachtungstyp, Beobachtungsort, Übereinstimmung mit bekannten Personen) könnten also wie folgt aussehen:

- T1_C1_[(P1,80%),(P3,10%)(P5,40%)]
- T1_C1_[(P2,20%),(P3,70%)(P5,20%)]
- T2_C2_[(P1,90%),(P3,10%)]
- ...

4.4 Emissionswahrscheinlichkeiten

Nachdem nun das Ausgabealphabet definiert ist, ist es notwendig festzulegen welche Ausgaben an welchen Zuständen Auftreten können und wie wahrscheinlich diese sind. Zu diesem Zweck werden für jede mögliche Beobachtung in jedem Zustand Emissionswahrscheinlichkeiten durch eine Ausgabefunktion f festgelegt. Oft wird diese bestimmt, indem mit Hilfe einer großen Zahl von Trainingsbeispielen statistische Aussagen über das Auftreten der Ausgabesymbole an den jeweiligen Zuständen gemacht werden.
Hier ist eine solche Festlegung der Emissionswahrscheinlichkeiten jedoch nicht möglich, vielmehr werden sie in Abhängigkeit der verschiedenen Parametern des jeweiligen Systemzustands berechnet. Da sich dieser ständig ändert, werden auch die Emissionswahrscheinlichkeiten dynamisch angepasst.
Wie bereits erwähnt, ist jedem HMM M_n eine Person P_n zugeordnet, deren Position es modelliert. Diese Person wird im Folgenden als "Fokusperson" bezeichnet. Aufgrund der fehlerbehafteten Identifikation der Personen ist eine Beobachtung selten eindeutig zuzuordnen. Vielmehr wird eine Beobachtung in der Regel mit mehreren Personen übereinstimmen, wenn auch mit unterschiedlicher Konfidenz. Ein HMM wird also eine große Zahl von Informationen erhalten die nur aufgrund

fehlerhafter Identifikation der Fokusperson zugeordnet werden können. Die wichtigste Aufgabe des probabilistischen Modells besteht darin zu entscheiden, welche Bedeutung einer Beobachtung beizumessen ist - ob sie also tatsächlich ein Indiz für die Position der Fokusperson ist oder aber eine Fehlinformation, ausgelöst durch eine Fehlidentifikation einer beliebigen anderen Person. Für Beobachtungen von Typ 1 und Typ 2 geht es also zunächst darum, zwischen Beobachtungen zu unterscheiden die von der Fokusperson oder von einer beliebigen anderen Person ausgelöst wurden. Für Beobachtungen von Typ 0 geht es hingegen darum zu entscheiden, wie wahrscheinlich es unter verschiedenen Bedingungen ist, dass eine Person trotz fehlender Detektion einen Zustand betreten, verlassen oder passiert hat. Es muss also möglich sein, auch zu berücksichtigen, dass eine Person an manchen Kameras schlicht überhaupt nicht bemerkt werden könnte.
Da die Menge der möglichen Beobachtungen in Kategorien eingeteilt werden kann, wird im Folgenden die Berechnung der Emissionwahrscheinlichkeiten für die Kategorien beschrieben anstatt auf jede Kombinationsmöglichkeit einzeln einzugehen.

4.4.1 Beobachtungen vom Typ 1

Beobachtungen vom Typ 1 werden durch das Betreten des Kamerablickfeldes durch eine Person ausgelöst. Eine wichtige, in der Beobachtung enthaltene Information ist die des Ortes an dem sie gemacht wurde. In Abhängigkeit von dem angenommenen Standort der Fokusperson ist dies ein wichtiges Indiz um zu entscheiden, ob tatsächlich die Fokusperson beobachtet wurde oder aber eine andere Person, bei der aufgrund der Unschärfe der Identifikation eine Übereinstimmung mit der Fokusperson erkannt wurde.

Der aktuelle Zustand repräsentiert eine Kamera

Für Zustände die eine Kamera modellieren, wird angenommen, dass eine Person im realen Kameranetzwerk das Blickfeld einer der Kameras betreten hat. Wie bereits erwähnt ist der Ort, an dem die Beobachtung gemacht wurde, Teil des Beobachtungsvektors. Die Emissionswahrscheinlichkeit der aktuellen Beobachtung im aktuellen Zustand hängt davon ab, ob der aktuelle Zustand den Ort der Beobachtung modelliert. Ist die Beobachtung am aktuellen Ort gemacht worden und der aktuelle Zustand vom Typ C, so ist davon auszugehen dass die Fokusperson beobachtet wurde. Die Voraussetzungen hierfür sind zum einen eine erfolgreiche Detektion und zum anderen die Konfidenz der Identifikation der Fokusperson. Die Wahrscheinlichkeit für eine erfolgreiche Detektion wird im Folgenden als Detektionswahrscheinlichkeit δ bezeichnet und beschreibt die Wahrscheinlichkeit, dass eine Person erkannt wird, wenn sie das Blickfeld einer Kamera betritt (vergleiche Kapitel 3). Die Emissionswahrscheinlichkeit ε kann somit in Abhängigkeit der

Detektionswahrscheinlichkeit δ und der Eindeutigkeit des Merkmalsmatchings M_i mittels Formel 4.1 berechnet werden.

$$\varepsilon = \delta * M_i \tag{4.1}$$

Die "Eindeutigkeit des Merkmalsmatchings" für eine Person i wird dabei mit Hilfe von Formel 4.2 berechnet und erklärt sich wie folgt:
Der Vorgang des Matchings ist eine einfache Berechnung der Korrelation zwischen dem aus der Beobachtung extrahierten Merkmalsvektor und dem einer bekannten Person. Das Ergebnis dieses Matchings ist die Übereinstimmung der beiden Vektoren und wird für Person i als m_i bezeichnet (siehe Abschnitt 3.4). Da Personen sich im Allgemeinen nicht in allen Merkmalen unterscheiden und die Merkmalsbestimmung dazu noch fehlerbehaftet ist, ist eine Identifikation meist nicht absolut eindeutig. Wie stark eine Beobachtung nun für eine Person spricht hängt aber nicht allein davon ab wie gut die extrahierten Merkmale mit denen der Person übereinstimmen. Es spielt ebenfalls eine Rolle, wie wahrscheinlich es ist, dass selbige Beobachtung von anderen Personen hätte erzeugt werden können. Eindeutigkeit unter der Menge der Personen P bedeutet hier also Abgrenzung einer Person i von den $|P| - 1$ anderen Personen und kann somit wie in Formel 4.2 berechnet werden.

$$M_i = m_i * \prod_{k=0, k \neq i}^{|P|} (1 - m_k) \tag{4.2}$$

Für den Fall, dass der aktuelle Zustand s_j nicht den Ort der Beobachtung modelliert oder es sich um einen Zustand handelt, der eine Kamera ohne Detektion modelliert ($TypN$), ist die aktuelle Beobachtung mit einer gewissen Wahrscheinlichkeit von einer anderen als der Fokusperson ausgelöst worden.
Um zu verhindern, dass ein passieren einer Kamera in der Modellierung auch ohne Detektion an dieser Kamera möglich ist, muss hier unterschieden werden welcher Zustand Vorgänger des aktuellen ist. Modelliert der vorhergehende Zustand einen angrenzenden Weg, ist die Emissionswahrscheinlichkeit 0 um das Betreten ohne Detektion zu unterbinden. Ist der vorherige Zustand aber identisch mit dem aktuellen, wird berechnet wie wahrscheinlich es ist, dass eine andere Person die aktuelle Beobachtung verursacht hat. An dieser Stelle kommen die anderen Personen anderer Modelle neben der Fokusperson des aktuellen Modells ins Spiel. Wie bereits beschrieben, sind in jedem Schritt, für jede Person neben der eigenen auch die vorausgesagten Aufenthaltswahrscheinlichkeiten aller übrigen Personen bekannt. Sei also die Aufenthaltswahrscheinlichkeit für Person i im Zustand n im vorherigen Schritt mit α_{in}^{t-1} bezeichnet. Multipliziert man diese Aufenthaltswahrscheinlichkeit α_{in}^{t-1} mit den jeweiligen Übergangswahrscheinlichkeiten $a(j, k)$

zu den direkt angrenzenden Zuständen N, so kann man aus der Aufenthaltswahrscheinlichkeit für eine Person i im vorherigen Schritt auf die im aktuellen Schritt $\hat{\alpha}^t_{is_j} = \sum_{n=0}^{|N|} \alpha_{in} * a(n, s_j)$ schließen.
Insgesamt ergibt sich die Emissionswahrscheinlichkeit aus Formel 4.3. Für eine Beobachtung vom Typ 1 muss eine Person zunächst detektiert werden, somit ergibt sich (δ). Da wie bereits erklärt, hier davon auszugehen ist, dass eine andere Person als die deren Position vom aktuellen HMM modelliert wird die Beobachtung verursacht hat, wird für alle übrigen Personen P berechnet wie Wahrscheinlich es sich um sie handelt. Dazu wird zunächst ihre jeweilige Übereinstimmung bei der Identifikation M_i berücksichtigt und zusätzlich die Wahrscheinlichkeit, dass sie sich im aktuellen Schritt im aktuellen Zustand s_j befinden ($\hat{\alpha}^t_{is_j}$). Um zu unterbinden, dass durch Beobachtungen, die durch andere Personen verursacht wurden, Zustände passiert werden können die eine Kamera mit Detektion modellieren (C), wird für den Fall eines Zustandswechsel ($s_j(t) \neq s_j(t-1)$) die Emsissionswahrscheinlichkeit auf 0 festgelegt.

$$\varepsilon = \begin{cases} \delta * \sum_{i=0}^{|P|} M_i * \hat{\alpha}^t_{is_j}, & \text{falls } s_j(t) = s_j(t-1) \\ 0, & \text{sonst} \end{cases} \tag{4.3}$$

Der aktuelle Zustand repräsentiert einen Weg

Beobachtungen von Typ 1, also das Auftauchen einer Person können natürlich nur in solchen Zuständen ausgegeben werden, die Kameras repräsentieren. Wenn wir davon ausgehen, dass die Fokusperson sich auf einem Weg befindet (also dass der aktuelle Zustand einen Weg modelliert), muss die Beobachtung von einer anderen Person ausgelöst worden sein und die Emissionswahrscheinlichkeit berechnet sich somit aus der Formel 4.3.

4.4.2 Beobachtungen vom Typ 2

Beobachtungen vom Typ 2 treten auf wenn eine Person das Blickfeld einer Kamera verlässt, also beim Übergang von einer Kamera zu einem Weg.

Der aktuelle Zustand repräsentiert eine Kamera

Wenn die Fokusperson sich im Blickfeld einer Kamera befindet, kann sie nicht gleichzeitig eine Beobachtung ausgelöst haben die besagt, dass sie eine Kamera verlassen hat. Dabei ist es nicht relevant, ob die Person beim Eintritt in das Blickfeld dieser Kamera detektiert wurde, also ob sie sich in einem Zustand vom Typ C oder N befindet. Die Beobachtung muss also durch eine der anderen Personen

und durch fehlerhafte Identifikation ausgelöst worden sein. Die Emissionswahrscheinlichkeit wird für diesen Fall also mit Hilfe der bereits bekannten Formel 4.3 berechnet.

Der aktuelle Zustand repräsentiert einen Weg

Wie bereits erwähnt ist in diesem Fall prinzipiell anzunehmen, dass die Beobachtung von der Fokusperson ausgelöst wurde. Es ist allerdings noch zu beachten, wo im Netz die Beobachtung gemacht wurde. Da Beobachtungen an Kameras gemacht werden, in diesem Fall aber für den Aufenthalt auf einem Weg sprechen - die Kamera ist ja gerade verlassen worden, lässt sich der Beobachtungsort nicht einfach mit dem Ort vergleichen den der aktuelle Zustand modelliert, so wie es für Beobachtungen von Typ 1 der Fall war. Nach der Beobachtung, dass eine Person den Beobachtungsort (also das Blickfeld einer Kamera) verlassen hat, kann sie sich auf jedem der angrenzenden Wege befinden. Grenzt also der Ort der Beobachtung an den aktuellen Weg an, gilt für die Emissionswahrscheinlichkeit: $\varepsilon = M_i$ wobei M_i wieder mittels Formel 4.2 berechnet wird. Das Problem mit Beobachtungen vom Typ 2 ist, dass sie vollständig von einer vorhergehenden Typ 1 Beobachtung abhängen. Es wird also keine erneute Detektion benötigt, aber es besteht die gleiche Unsicherheit über die Identität der Person. Das führt genau dann zu Problemen wenn sich mehr als eine Person im Blickfeld der Kamera aufhält. Um trotzdem eine möglichst gute Zuordnung der Typ 2 Beobachtung zu gewährleisten, wird der Grad der Eindeutigkeit bei der Personenidentifikation (M_i) hier als Emissionswahrscheinlichkeit verwendet.

Wurde die Beobachtung jedoch an einem anderen, weiter entfernten Weg gemacht, so ist wiederum anzunehmen, dass die Beobachtung von einer anderen Person ausgelöst wurde und es kommt Formel 4.3 zum Einsatz.

4.4.3 Typ 0

Beobachtungen vom Typ 0 stellen eine Besonderheit dar: Anders als Typ 1 und Typ 2 Beobachtungen beschreiben sie nicht die Veränderung eines Systemzustandes, sondern genau das Gegenteil, nämlich dass sich nichts geändert hat. Sie sind notwendig um mögliche Wege zu berücksichtigen, die eine Person zurückgelegt hat, ohne detektiert zu werden. Da es an verschiedenen Stellen im Netz auch mehr oder weniger wahrscheinlich ist sie zu passieren ohne eine Beobachtung von Typ 1 oder 2 auszulösen, besteht für die Emissionswahrscheinlichkeiten eine Abhängigkeit vom vorherigen Zustand. Darüber hinaus ist eine Fallunterscheidung bezüglich des Betretens von Kameras nötig. Die Notwendigkeit dieser Fallunterscheidung rührt von der Funktionsweise der Detektion auf unterster Ebene her. Das Verlassen einer Kamera entspricht im realen Kameranetz dem verschwinden eines Tracks einer Person

Wechsel von einem Weg in eine Kamera ohne Detektion:	$\varepsilon = 1 - \delta$
Verbleib in einer Kamera ohne beim Eintritt detektiert worden zu sein:	$\varepsilon = 1$
Verlassen einer einer Kamera ohne beim vorherigen Eintritt detektiert worden zu sein:	$\varepsilon = 1$
Verbleib auf einem Weg: mit $\|N\|$ = Anzahl angrenzender Kameras	$\varepsilon = 1 - \sum_{i=0}^{\|N\|}(1 - \delta)$
Wechsel von Weg in eine Kamera unter Detektion:	$\varepsilon = 0$
Verbleib in einer Kamera nach Detektion des Eintritts:	$\varepsilon = 1$
Verlassen einer Kamera wobei der vorherige Eintritt detektiert wurde:	$\varepsilon = 0$

Tabelle 4.1: Übersicht über Emissionswahrscheinlichkeiten für Beobachtungen von Typ 0

aus dem Blickfeld einer Kamera. Damit ein solcher Track entsteht muss die Person (also ihr Eintritt) aber überhaupt erst detektiert worden sein. Das bedeutet, dass es ohne Detektion des Eintritts auch keine Detektion des Austritts geben wird und dass auf einen detektierten Eintritt früher oder später stets ein Austritt erfolgt. Um diese Abhängigkeit von Eintrittsdetektion und Austrittsdetektion modellieren zu können ist eine Fallunterscheidung und die damit verbundene Einführung der bereits erwähnten "NichtDetektion"- Zustände notwendig.

In Tabelle 4.1 sind die Emissionswahrscheinlichkeiten für die Beobachtungen von Typ 0 zusammengefasst. Der Wechsel von einem Zustand der einen Weg repräsentiert in einen der eine Kamera repräsentiert die betreten wurde ohne das eine Detektion stattgefunden hat bedeutet, dass eine Person das Blickfeld einer Kamera im realen Kameranetzwerk zwar betreten hat, aber nicht detektiert wurde. Die Wahrscheinlichkeit dafür ist also gerade $1 - \delta$, mit δ der wie oben definierten Detektionswahrscheinlichkeit. Für den Verbleib in einem solchen Zustand gilt genauso wie für das Verlassen, dass die Wahrscheinlichkeit hierfür 1 ist. Die einzige Möglichkeit in einen solchen Zustand zu gelangen ist ja gerade nicht detektiert zu werden - infolgedessen kann auch ein weiterer Aufenthalt oder das Verlassen nicht detektiert werden und es wird keine Veränderung beobachtet werden, also eine Typ 0 Beobachtung.
Die Wahrscheinlichkeit keine Veränderung zu beobachten, wenn die Person auf einem Weg verbleibt ist etwas aufwendiger zu berechnen: Die einzige Möglichkeit, dass die Person sich nicht mehr, wie im vorherigen Schritt, auf dem Weg befindet, wäre das Eintreten in eine Kamera ohne dabei beobachtet zu werden. Berechnet man dies für jede an den Weg angrenzende Kamera ($\sum_{i=0}^{|N|}(1 - \delta)$), ergibt sich aus

der Umkehrung $(1 - \sum_{i=0}^{|N|}(1 - \delta))$ gerade die Wahrscheinlichkeit für den Verbleib auf dem Weg.
Das Betreten oder Verlassen des Blickfeldes einer Kamera unter Annahme der Detektion hingegen, sind nicht möglich wenn keine Veränderung - also die Detektion an sich - beobachtet wurde. Hier macht sich die eingangs besprochene Fallunterscheidung bemerkbar, die hier durch ein aufsplitten der physikalischen Kameras in zwei Zustände des Modells erreicht wird. Die Wahrscheinlichkeit für den Verbleib im Blickfeld einer Kamera für den Fall das keine Veränderung beobachtet wird ist 1, da die Beobachtung keiner Veränderung ja gerade für das beibehalten des aktuellen Systemzustandes spricht und ein Verlassen wie bereits bemerkt, ausgeschlossen ist.

4.5 Übergangswahrscheinlichkeiten

Übergangswahrscheinlichkeiten repräsentieren, wie wahrscheinlich der Übergang von einem Zustand zu einem anderen ist. Dabei ist hier durch die Netzstruktur bereits festgelegt welche Übergänge überhaupt möglich sind, also $a(i, j) > 0$. Häufig werden diese Wahrscheinlichkeiten anhand von Trainingsdaten, durch eine einfache statistische Analyse der Häufigkeit verschiedener Übergänge bestimmt. Da hier, aufgrund fehlender Trainingsdaten oder anderer Anhaltspunkte, keine Aussagen über mehr oder weniger wahrscheinliche Übergänge gemacht werden können, sind alle Übergänge gleich wahrscheinlich. Sie sind lediglich abhängig von der Anzahl der von einem Zustand abgehenden Transitionen, so dass für jeden Zustand i die Bedingung in 4.4 erfüllt ist. N ist dabei wie oben die Menge der direkt angrenzenden Zustände.

$$\sum_{j=0}^{|N|} a(i, j) = 1 \tag{4.4}$$

5 Praktische Umsetzung

Bei der konkreten Umsetzung der beschriebenen Verfahren und Methoden gibt es eine Vielzahl von Möglichkeiten. Je nach dem welche Ausrichtung im Vordergrund steht bieten sich verschiedene Lösungsansätze an. Bei der Implementierung der Verfahren zur Merkmalsextraktion geht es besonders darum, auf realistischen Daten arbeiten zu können und dabei ausreichend gute Ergebnisse zu erzielen um die Nutzbarkeit der Verfahren zu untersuchen. Außerdem muss auch eine Performance erreicht werden die eine Verwendung der Verfahren im live Einsatz ermöglicht. Bei der Umsetzung des probabilistischen Trackingverfahrens ist der Anspruch an die Echtzeitfähigkeit natürlich genauso gegeben, stellt aber in diesem Fall keine wesentliche Hürde dar. Entscheidender ist die Flexibilität des Systems und die Möglichkeit eines breiten Spektrums von Anwendungen, aber auch Evaluationen. Besonders Interessant, speziell weil es sich um ein probabilistisches Modell handelt, ist der Einfluss verschiedener äußerer Einflußfaktoren auf die erzielbare Trackingleistung. So ließe eine detaillierte Untersuchung der Systemeigenschaften unter variierender Detektionswahrscheinlichkeit δ der Personen einen Rückschluss auf notwendige Sensorgenauigkeiten zu, wenn eine Untergrenze der Trackingleistung vorgeschrieben wird. Um diesem Anspruch gerecht zu werden, wurde eine Simulationsumgebung für Netzwerke von verteilten Kameras aufgebaut, die es ermöglicht das Systemverhalten mit beliebigen Parametern und vollkommen reproduzierbarem Verhalten der Akteure zu analysieren.

5.1 Simulation

Eine Simulation bringt ganz generell eine Reihe von Vorteilen aber auch Nachteile mit sich. Ein bereits erwähnter Vorteil ist zum einen die Reproduzierbarkeit von Systemverhalten aber auch die Variation von Parametern ohne einen nennenswert größeren Aufwand. Auf der anderen Seite stellt eine Simulation immer auch eine Vereinfachung dar und kann daher die Realität nur begrenzt wiedergeben. Wenn also mit einer Simulation gearbeitet wird um ihre Vorteile zu nutzen, ist es wichtig das simulierte System eingehend zu untersuchen um die wirklich relevanten Eigenschaften möglichst gut in der Simulation darstellen zu können.
Im vorliegenden System gibt es zwei Schwerpunkte. Zum einen die Detektion und Identifikation und zum anderen das globale Tracking der Personen. Um Detektion und Identifikation berücksichtigen zu können, muss für beide Komponenten die

Abbildung 5.1: Beispiele für die mittels HalfLife2 gerenderten Szenen

Möglichkeit bestehen, ihre Fehlerrate in der Simulation abzubilden.
Das globale Tracking der Personen ist stark abhängig von den räumlichen Gegebenheiten in denen das Kameranetzwerk aufgebaut ist, sowie den Bewegungsmustern der einzelnen Personen. Diese Bewegungsmuster hängen natürlich stark von der Umgebung aber auch von der Zahl der Personen und der Interaktion untereinander ab. Ein Simulationswerkzeug das eine sehr gute Annäherung an reale Interaktionen und räumliche Gegebenheiten bietet ist das so genannte "Object Video Virtual Video Tool" [36].

5.1.1 Half Life 2

Eines der Hauptziele vieler aktueller Computerspiele ist die Erschaffung möglichst realistischer Welten um Grenzen zwischen Realität und Spiel immer weiter zu reduzieren. Um diesem Ziel möglichst nah zu kommen wird die reale Welt mit einem enormen Aufwand und Detail im Hinblick auf die Optik aber auch die Interaktion von Akteuren und physikalischen Eigenschaften von Objekten nachgebildet. Das in diesen Aufgabenbereich schon sehr viel Geld und Arbeit geflossen ist nutzt das "Object Video Virtual Video Tool" (OVVV), eine Modifikation des Spiels "Half Life2", aus. Durch die Verwendung der Engine[1] von HalfLife2 ist das OVVV in der Lage virtuelle Welten zu generieren die nicht nur im Hinblick auf das Aussehen (Abbildung 5.1) sehr realitätsnah sind, sondern auch im Hinblick auf das Verhalten der Akteure. Die ebenfalls künstlich generierten Personen in einer solchen virtuellen Welt bewegen sich unter Berücksichtigung der Umgebung sowie der anderen Akteure. So müssen Türen geöffnet und geschlossen werden, Wege zurückgelegt werden und auch anderen Personen durch Ausweichen oder Warten begegnet werden.

Die virtuellen Welten selbst sind nicht vorgegeben, sondern können mit Hilfe des "Hammer Editors" sehr frei gestaltet werden.

[1] Die Engine eines Spiels stellt ein Grundgerüst dar. Sie umfasst vor allem die Grafik-Engine zur Berechnung aller optischen Elemente sowie das Physiksystem und Basisfunktionen wie Eingabe, Sound und Datenmanagement

5.1.2 Hammer Editor

Der Hammer-Editor (Abbildung 5.2) ähnelt vom Aufbau her typischen 3D Modellierungs und Animationsprogrammen. Neben einem großen Katalog vorgefertigter Objekte bietet Hammer sehr viel Freiheit bei der Gestaltung eigener Objekte. Unter Zuhilfenahme der Szenenansichten aus verschiedenen Perspektiven, können Boden-, Decken- und Wandelemente zu beliebigen Gebäudekomplexen aufgebaut werden. Die freie Texturierung aller Elemente erlaubt einen großen Einfluss auf das Erscheinungbild aller Objekte. Neben statischen Objekten lassen sich auch dynamische Objekte, wie Personen, im Hammer Editor erstellen. Diese können entweder sehr starre Verhaltens- und Bewegungsmuster aufgeprägt bekommen oder im Rahmen allgemeiner Verhaltensregeln frei agieren.

5.2 Implementierung

Für die Implementierung aller Merkmalsextraktionen sowie des Detektionsverfahrens und des lokalen Trackings wurde C++ als Programmiersprache unter Einsatz der Bildverarbeitungsbilbliothek OpenCV [37] verwendet. Für die Geschlechtsdetektion mittels SVMs kam darüber hinaus die Bibliothek SVM^{light} [38] zum Einsatz. Das globale Tracking im Zusammenspiel mit den Simulationsdaten wurde in Python realisiert. Besonderheiten zur Implementierung des probabilistischen Modells finden sich in Abschnitt 2.1.1. Zur Verwendung der Simulationsdaten in der Python Implementierung war es notwendig eine geeignete Schnittstelle zu schaffen, da eine solche nicht Teil der Simulationsumgebung ist.

5.2.1 Einbinden der Simulationsdaten

Das Einbinden der Simulationsdaten in die Implementierung bringt zwei Aufgaben mit sich. Zum einen muss eine Schnittstelle für das Datenformat der Simulationdaten geschaffen werden und zum anderen müssen die - da künstlich erzeugt - perfekten Simulationsdaten mit den aus Detektion und Identifikation entstehenden Fehlern versehen werden.
Die Simulation liefert für alle Personen Grundwahrheiten (“Groundtruth” Daten) in einem dokumentierten Binärformat. Dieses wurde in XML konvertiert, womit ein einfacher Zugriff auf die Daten aus Python heraus gegeben ist. Um einen Fehler in der Detektion zu Simulieren genügt es für jede durch die Simulation gegebene Beobachtung mit einer vorgegebenen Wahrscheinlichkeit zu entscheiden ob sie an das System weitergereicht wird oder nicht. Einen Fehler in der Identifikation zu simulieren ist komplexer da diese aus mehreren Komponenten besteht. Genaugenommen hängt die Identifikationsqualität davon ab, wie sicher die einzelnen Merkmale bestimmt werden können. Anders als bei der Detektion gibt es hier

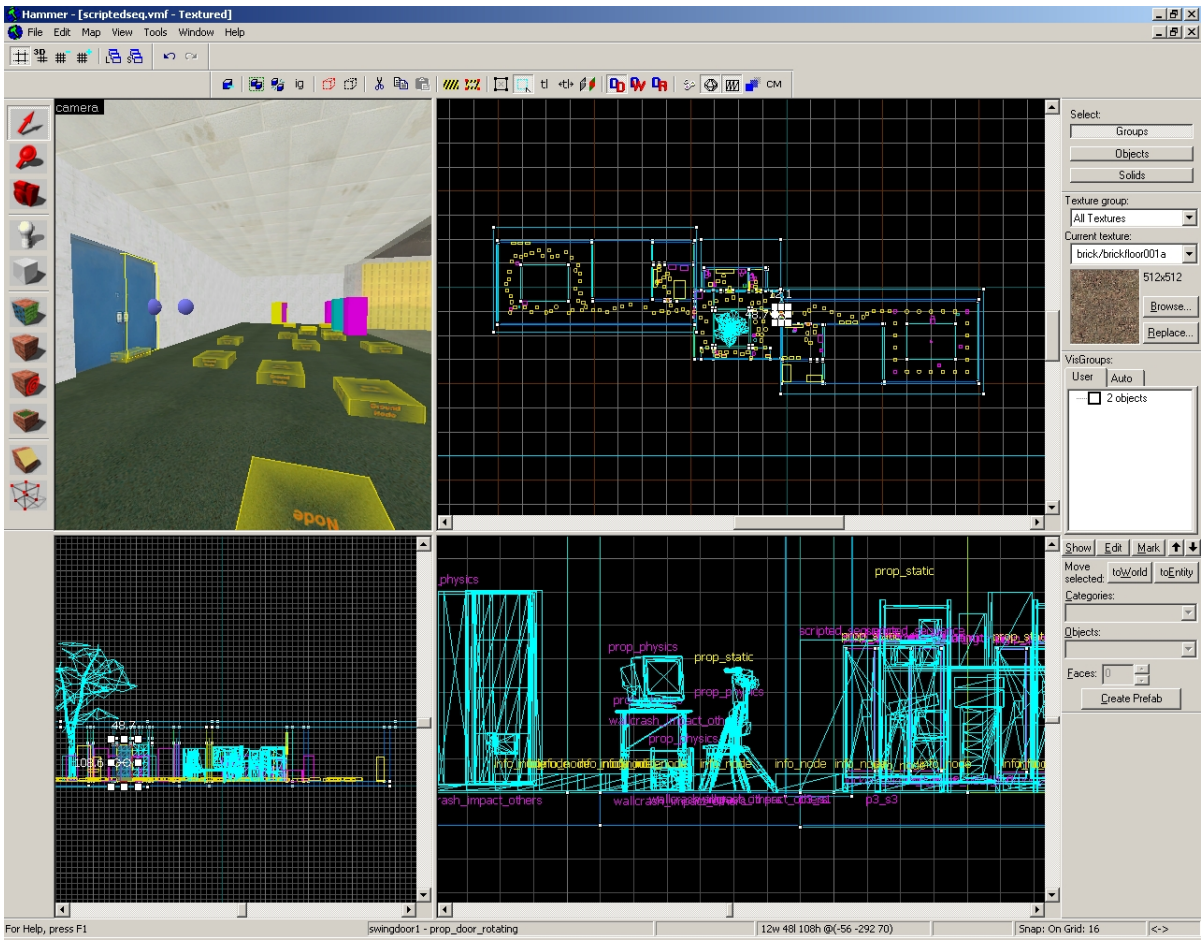


Abbildung 5.2: Screenshot des Hammer Editors

allerdings mehr Möglichkeiten für einen Fehler. Außer dass ein Merkmal überhaupt nicht bestimmt werden konnte, kann vor allem ein falscher Wert für ein Merkmal erkannt worden sein. Um dieser Konstellation von Möglichkeiten gerecht zu werden, wird zunächst mit einer gewissen, vorgegebenen Wahrscheinlichkeit entschieden ob der richtige Wert für ein Merkmal verwendet wird. Ist dies nicht der Fall wird zufällig einer der anderen möglichen Merkmalswerte einschließlich dem Merkmalswert “unbekannt” in der weiteren Berechnung der Identität verwendet.

6 Ergebnisse

In dem folgenden Kapitel sollen die Ergebnisse des Gesamtsystems vorgestellt werden. Zunächst werden dazu die Rahmenbedingungen erläutert unter denen die Experimente zustandegekommen sind und einige der Einflussfaktoren näher untersucht. Dazu gehört zum einen die Beobachtungskonfidenz, die den Einfluss der Merkmalsextraktion verdeutlichen soll. Aufgrund der probabilistischen Eigenschaften des Systems ist desweiteren eine Untersuchung der hier gewählten Evaluierungsmethoden in diesem Zusammenhang zu analysieren.
Den zentralen Punkt bilden die ausführlichen Ergebnisse unter Variation sowohl der Detektionsqualität als auch der Qualität der Merkmalsextraktion (im Folgenden "Extraktionsqualität"). Um das System auch in Hinblick auf die in Kapitel 3 erreichten Genauigkeiten der Merkmalsextraktion auf realistischen Daten hin zu untersuchen werden zum Schluss die damit erreichten Ergebnisse in 6.5 vorgestellt.

6.1 Aufbau

Alle Evaluationen wurde auf Daten durchgeführt, die mittels der in Kapitel 5 vorgestellten Simulationsumgebung gewonnen wurden. Insgesamt kamen dabei sechs Kameras zum Einsatz, die wie in Abbildung 6.1 angeordnet sind. Der Grundriss ist dabei an den real vorhandenen Räumlichkeiten des Instituts für Theoretische Informatik an der Universität Karlsruhe angelehnt. Innerhalb dieses Areals wurden die Wege von fünf Personen über den Zeitraum einer halben Stunde simuliert. Die Personen bewegen sich, dank zufälliger Zielwahl innerhalb der Simulation, selbstständig durch das gesamte Areal. Für alle Simulationsdurchläufe wurde eine auf diese Weise einmal generierte Sequenz wiederholt eingesetzt, um eine Vergleichbarkeit der Ergebnisse zu ermöglichen. Die zur Auswertung notwendigen Referenzdaten werden von der Simulationsumgebung synchron zu den Videodaten erzeugt, wodurch kein Annotieren der gewonnenen Daten nötig war.
Alle Ergebnisse wurden gewonnen, indem über die Einzelergebnisse von zehn Durchläufen gemittelt wurde. Mehrere Durchläufe sind hier sinnvoll, da zwar immer das gleiche Bewegungsmuster verwendet wird, aber die Ergebnisse wegen simulierter, probabilistischer Detektionsqualität sowie simulierter Extraktionsqualität variieren.

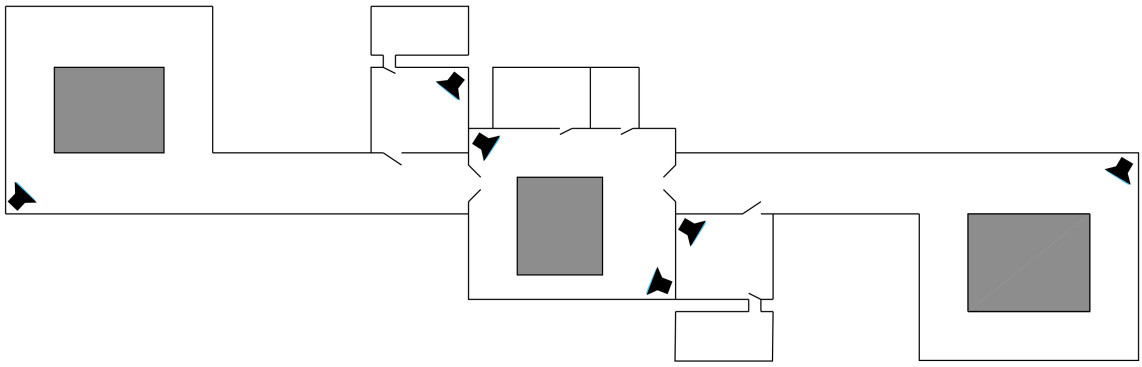


Abbildung 6.1: Kameraverteilung in der simulierten Welt

6.1.1 Auswertung des probabilistischen Ergebnisses

Die Schätzung des Systemzustands, den man mittels des probabilistischen Modells gewinnt, entspricht einer Wahrscheinlichkeitsverteilung über die im Netz vorhandenen Zustände. Die Aussage des Modells ist also, wie wahrscheinlich es für jeden einzelnen Zustand ist, dass eine bestimmte Person sich dort aufhält. Die Referenzdaten hingegen geben an diskreten Zeitpunkten (genau dann wenn eine Person eine Kamera passiert) die korrekte Position fest vor. Um einen direkten Vergleich zu ermöglichen ist es wünschenswert, dass das Modell ebenfalls einen einzigen Zustand als Ergebnis ausgibt. Aus diesem Grund gibt das System zur Evaluierung in jedem Zeitschritt jeweils den Zustand mit der größten Aufenthaltswahrscheinlichkeit einer bestimmten Person aus. Untersucht man die Differenz zwischen der größten und der zweitgrößten Aufenthaltswahrscheinlichkeit, stellt man fest, dass dieser im Schnitt bei nur 2,7% liegt. Dies zeigt deutlich, dass die Verwendung des Maximums der Wahrscheinlichkeitsverteilung nicht immer die genaue Qualität der Schätzung des Systemzustands wiedergeben kann.

6.2 Beobachtungskonfidenz

Die Beobachtungskonfidenz beschreibt für eine Beobachtung, wie eindeutig sie einer einzigen Person zugeordnet werden kann und Berechnet sich wie in Abschnitt 4.4 erläutert. Es geht also darum, wie stark eine Person von allen anderen abgegrenzt werden kann und somit um den Nutzen einer Beobachtung. Stimmen die bei einer Beobachtung bestimmten Merkmale mit allen Personen gleichermaßen überein, ist der Wert der Beobachtung sehr gering weil nicht klar ist, welcher Person diese Information zuzuordnen ist. In Abbildung 6.2 ist die Entwicklung der Beobachtungskonfidenz unter Variation der “Extraktionsqualität” veranschaulicht. Dabei ist die Extraktionsqualität, die sich in ihrer simulierten Form wie in 5.2.1

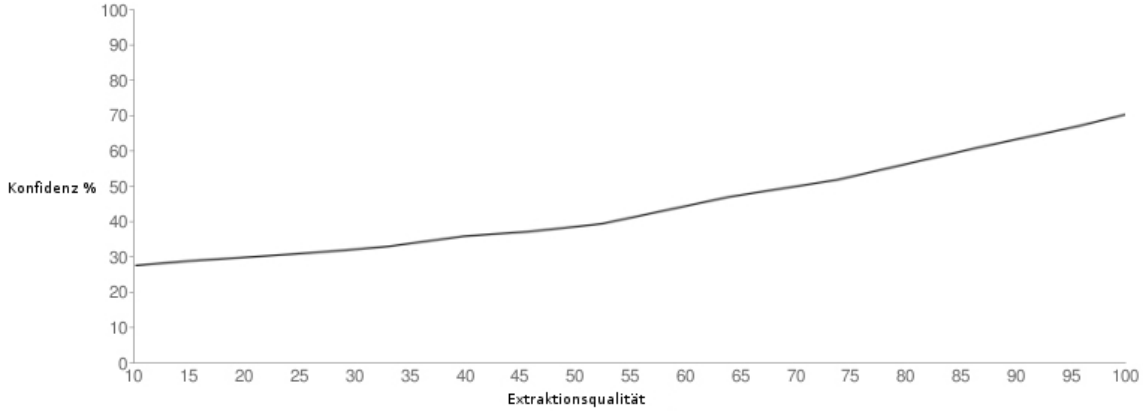


Abbildung 6.2: Durchschnittliche Beobachtungskonfidenz

ergibt, der x-Achse zugeordnet. Auf der y-Achse ist die durchschnittliche Beobachtungskonfidenz zu sehen, also die durchschnittliche Sicherheit, mit der bei den Durchläufen mit dieser Parameterkombination, Beobachtungen einer bestimmten Person zugeordnet werden konnten.
Die Detektionsqualiät ist hier auf 100% festgelegt, da sie hier kaum einen Einfluss hat. Dies erklärt ebenfalls, warum die Konfidenz mit sinkender Extraktionsqualität abnimmt: Je mehr Unsicherheit es bei der Bestimmung der Merkmale gibt, umso größer die Wahrscheinlichkeit dass diese Merkmale mit denen anderer Personen übereinstimmen.

Es sollte beachtet werden, dass auch mit bester Detektionsqualität (100%) und bester Extraktionsqualität (100%) keine einhundertprozentige Beobachtungskonfidenz erreicht wird. Der Grund dafür liegt darin, dass eine solche Konfidenz nur dann erreicht werden könnte wenn eine Person sich in allen Merkmalen von allen anderen Personen unterscheidet. Dies ist aufgrund binärer Merkmale wie dem tragen einer Brille jedoch nicht immer möglich. Hier wird also auch deutlich, dass die Merkmale der Personen für die Simulation nicht künstlich völlig verschieden gewählt wurden, um die Unterscheidung zu erleichtern.

6.3 Gesamtergebnisse unter Variation der Sensorgenauigkeiten

Wie bereits erwähnt geht die Evaluation über eine einfache Auswertung hinaus. Es soll vielmehr eine Analyse des Zusammenspiels der relevanten Parameter stattfinden. Wie für die Abbildung 6.2 zur Beobachtungskonfidenz ist auch in den Abbildungen 6.3 und 6.4 die Extraktionsqualität der x-Achse zugeordnet, während

die einzelnen Kurven die Durchläufe mit jeweils verschiedener Detektionsqualität darstellen. Zur Evaluierung wurden zwei Auswertungskriterien verwendet. Eine diskrete (Abbildung 6.3) und eine kontinuierliche Auswertung (Abbildung 6.4).
Für die diskrete Auswertung wurde an den Zeitpunkten, an denen eine reale Beobachtung (Typ 1 oder Typ 2) gemacht wurde, überprüft ob die aktuelle Schätzung des Systems korrekt ist. Auf der y-Achse des Diagramms findet sich demnach der prozentuale Anteil korrekt geschätzter Zustände. Die kontinuierliche Auswertung hingegen überprüft zu jedem wie in Abschnitt 4.3 beschriebenen Systemtakt, ob die Schätzung des Modells mit der aktuellen Position der Person übereinstimmt. Die y-Achse dieses Diagramms stellt also den prozentualen Zeitanteil dar, in dem das System die Position der Person korrekt geschätzt hat. Für die Auswertung des kontinuierlichen Falls gibt es eine Besonderheit. Verlässt eine Person eine Kamera, weiß das System im Idealfall zwar, dass die Person sich auf einem der angrenzenden Wege befinden muss, nicht aber auf welchem, da alle Wege zunächsteinmal gleichwahrscheinlich angenommen werden. Solange also das System einen der angrenzenden Wege als Position ausgibt, wird dies nicht als Fehler gewertet. Bei der kontinuierlichen Auswertung muss darüber hinaus noch beachtet werden, dass identische Fehler sich in der zeitlichen Betrachtung sehr verschieden auswirken können. Wenn die Schätzung nicht mehr mit der Position der Person übereinstimmt, kann diese in aller Regel korrigiert werden sobald neue Informationen vorliegen. Wie lange es allerdings dauert bis neue Informationen verfügbar sind, hängt davon ab, wann die Person das nächste Mal eine Kamera passiert. Im Anschluss an einen Fehler wirkt sich hier also das Personenverhalten und die aktuelle Umgebung auf die Auswertung verschieden aus.
In der kontinuierlichen Auswertung wird darüber hinaus deutlich, welchen Fehler die harte Auswertung des Ergebnisses des probabilistischen Modells verursacht. Da hier auch an den Zeitpunkten zwischen den "realen" Beobachtungen evaluiert wird, macht sich der Fehlertyp stärker bemerkbar, der durch die zeitliche Diffusion der Wahrscheinlichkeitsverteilung durch die erzeugten Beobachtungen vom Typ 0 zustande kommt.

In Abbildung 6.3 ist zunächst zu bemerken, dass mit 100% Extraktionsqualität und Detektionsqualität keine Fehler gemacht werden. Obwohl, wie in Abschnitt 6.2 erläutert, 100% Extraktionsqualität nicht gleichbedeutend mit absoluter Sicherheit über die Identität einer Person ist, wird in diesem Fall die Konfidenz für die beobachtete Person stets die größte sein. Daher ist dieses Ergebnis für die diskrete Auswertung zu erwarten. Im kontinuierlichen Fall ist der Durchlauf für dieser Parameterkombination allerdings auch fehlerfrei. Im kontinuierlichen Fall wäre, anders als im diskreten Fall, auch unter diesen Idealbedingungen ein Fehler möglich, der durch die zeitliche Diffusion der Aufenthaltswahrscheinlichkeiten zwischen zwei realen Beobachtungen (Typ 1 oder Typ 2) auftritt.
Betrachtet man die Ergebnisse, die mit verschiedenen Detektionsqualitäten (die Kurven in Abbildung 6.3) für 100 prozentige Extraktionsqualität erreicht werden können, fällt auf, dass selbst für Detektionsraten bis hinunter zu 10% Ergebnisse

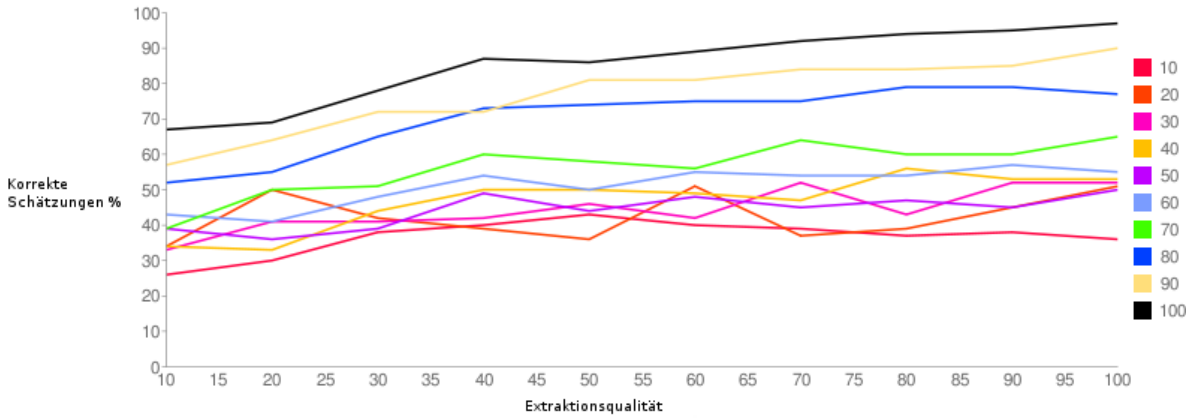


Abbildung 6.3: Auswertung des Trackingergebnisses an diskreten Zeitpunkten

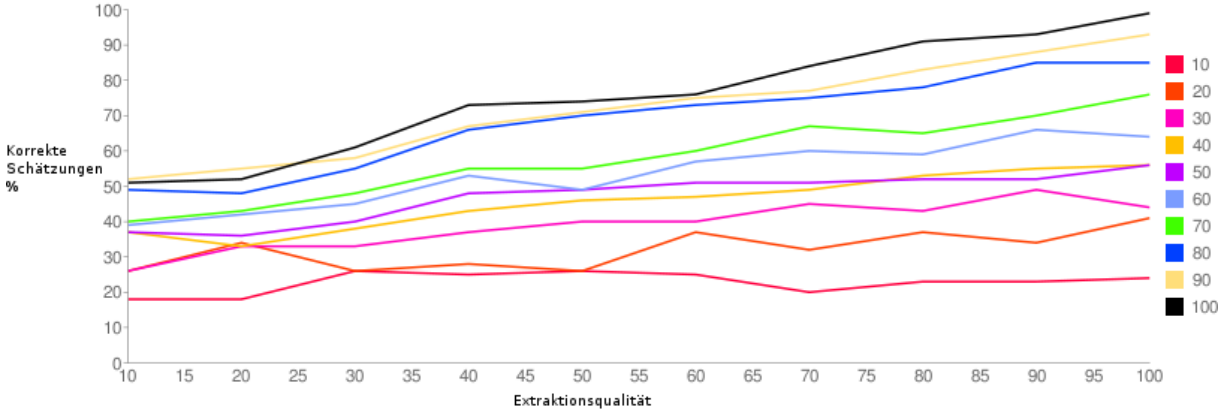


Abbildung 6.4: Auswertung des Trackingergebnisses anhand des Zeitanteils korrekter Vorhersagen

über 40% erreicht werden. Solange also eine gewisse Sicherheit über die Identität einer beobachteten Person besteht, ist das System in der Lage, mit den fehlenden Detektionen umzugehen. Hier machen sich die in Kapitel 4 eingeführten Zustände N bemerkbar, da sie ein Verfolgen von Wegen erlauben auf denen Personen nicht detektiert wurden.
Betrachtet man das andere Extrem - eine Extraktionsrate von nur 10% - zeigt sich, dass große Unsicherheit über die Identität von beobachteten Personen durch das probabilistische Modell soweit kompensiert werden kann, dass bei 100 prozentiger Detektionsqualität noch ein Ergebnis von über 60% erreicht werden kann.
Insgesamt fällt auf, dass die Kurven mit sinkender Extraktionsqualität nur langsam abfallen. Dies deutet darauf hin, dass das probabilistische Modell aufgrund der Berücksichtigung der Gesamtsituation in der Lage ist, Beobachtungen korrekt zuzuordnen - also zu entscheiden welche Person eine Beobachtung tatsächlich verursacht hat, ohne zu stark durch die sich verschlechternde Identifikation beeinflusst zu werden.

6.4 Baseline Experiment

Um die vorgestellten Ergebnisse in Relation setzen zu können, wurde ein weiteres Experiment durchgeführt. Hierbei kam das gleiche System zum Einsatz, allerdings wurde alle Zustände vom Typ N entfernt und keine Beobachtungen vom Typ 0 generiert. Dies hat zur Folge, dass ein passieren von Zuständen nicht mehr ohne reale Detektion möglich ist. Dies kann ein Vorteil sein weil es die zeitliche Diffusion der Aufenthaltswahrscheinlichkeit verhindert, führt aber vor allem zu Problemen für niedrige Detektionsraten, da das System so nicht in der Lage ist mögliche Wege, auf denen keine Detektion stattgefunden hat, zu modellieren. Vergleicht man Abbildungen 6.3 und 6.5 im Hinblick auf niedrige Detektionsraten, wird dieser Unterschied deutlich. Für eine Detektionsqualität von 10% und eine Extraktionsqualität von 100% ist sogar eine Verdoppelung des erreichten Ergebnisses zu beobachten.
Sehr schlechte Extraktionsqualität ($< 30\%$) kann dazu führen, dass gar keine Übereinstimmung mit der tatsächlich beobachteten Person vorhanden ist. Dies entspricht einer fehlenden Detektion und führt dazu, dass die Ergebnisse auch bei guter Detektionsqualität im Bereich schlechter Extraktionsraten stark abfallen.

6.5 Bezug zur Merkmalsextraktion

In Tabelle 6.1 wurde eine Evaluation unter Verwendung der auf realen Daten erzielten Ergebnisse zur Merkmalsextraktion aus Kapitel 3 durchgeführt. Anders als

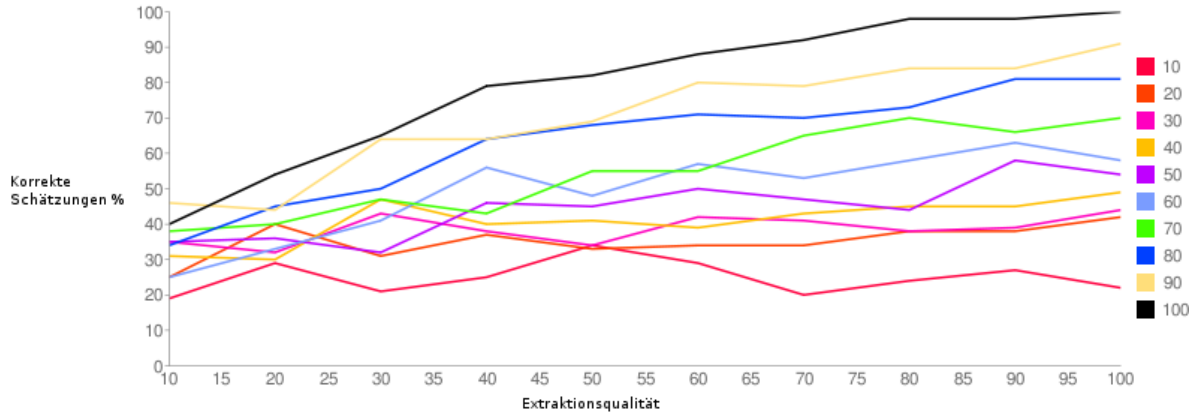


Abbildung 6.5: Auswertung des baseline Trackingergebnisses an diskreten Zeitpunkten

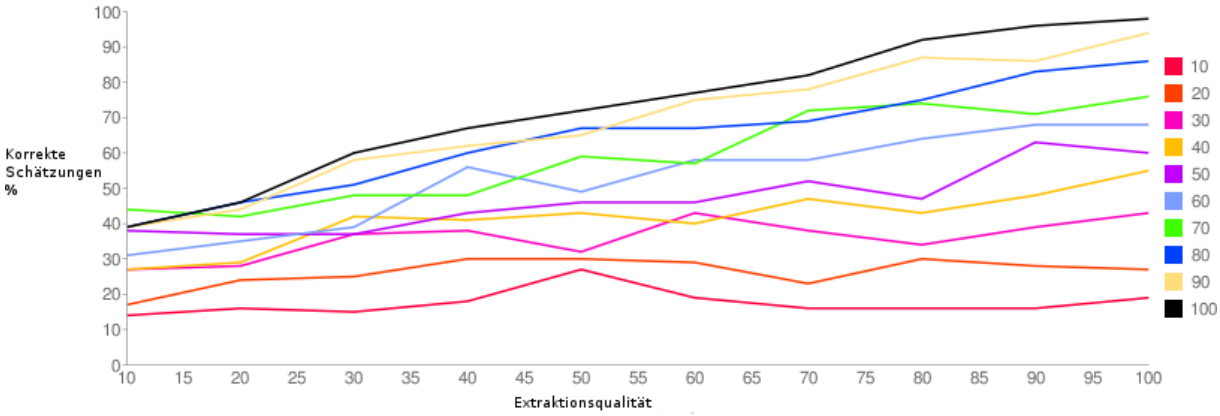


Abbildung 6.6: Auswertung des baseline Trackingergebnisses anhand des Zeitanteils korrekter Vorhersagen

Detektionsqualität (in %)	Diskret (in %)	Kontinuierlich (in %)
100	96	91
90	87	84
80	79	77
70	62	63
60	57	53
50	53	50
40	51	45
30	47	38
20	42	32
10	36	24

Tabelle 6.1: Trackingergebnisse bei feststehender Extraktionsqualität und variierender Detektionsqualität

in den bisherigen Experimenten, in denen die Extraktionsqualität für alle Merkmale gleichgesetzt wurde, wurden hier die für jedes Merkmal erreichten Werte verwendet. Für die Bestimmung des Merkmals „Brillenträger“ wurde eine Extraktionsqualität von 91% verwendet, für das Geschlecht einer Person 84% und für die Haarfarbe 78%. Wie bereits in Kapitel 3 erläutert wurden im Rahmen dieser Arbeit keine eigenen Evaluationen für die Bestimmung der Farben von Ober- und Unterkörper gemacht. Für diese Auswertung wurden diese daher mit 70% angenommen. Um zusätzlich die Auswirkungen verschiedener Detektionsqualitäten zu demonstrieren, wurden diese von 10% bis 100% in 10% Schritten variiert. Die jeweiligen Ergebnisse finden sich in den Spalten “Diskret” und “Kontinuierlich”, die unter den oben beschriebenen Auswertungsverfahren zustande gekommen sind. Die Ergebnisse zeigen, dass das Gesamtsystem - eine gewisse Detektionsqualität vorausgesetzt - gute Trackingergebnisse liefern kann.

7 Zusammenfassung und Ausblick

Das Tracken von Personen und das daraus resultierende Wissen über ihre Aufenthaltsorte bietet eine Vielzahl von Anwendungsmöglichkeiten. In vielen Fällen ist allein durch die Positionsbestimmung schon das Ziel erfüllt, in anderen Fällen bieten die erkannten Positionen die Basis für weitergehende Analysen. Die Komplexität der Trackingaufgabe und die damit verbundenen Probleme sind stark von dem jeweiligen Szenario abhängig. Das Tracken von Personen in einer weitläufigen Umgebung stellt dabei eine besondere Schwierigkeit dar, da für einen praktikablen Einsatz eine vollständige Abdeckung durch Kameras nicht gegeben sein kann. Somit ist der Anspruch an das Trackingsystem gegeben, diesen Ausfall an Information zu kompensieren und ein unter den gegebenen Bedingungen bestmögliches Ergebnis zu liefern.
Die vorliegende Arbeit setzt zwei Schwerpunkte um diesen Anforderungen gerecht zu werden. Um dem Problem stark abweichender Aufnahmebedingungen an verschiedenen Kameras entgegenzuwirken, wurden Merkmale ausgewählt die von solchen lokalen Variationen weitestgehend unabhängig sind und somit einen Vergleich über Kameras hinweg erlauben. Teil dieser Merkmalsselektion ist auch eine Analyse der praktischen Anwendung auf realen Daten um die Eignung eines Merkmals nicht nur auf theoretischer sondern auch auf praktischer Ebene zu klären.
Da aber selbst mit speziellen Merkmalen eine Identifikation von Personen nicht immer völlig fehlerfrei ist, sind die über den Systemzustand vorhanden Beobachtungen mit einer Unschärfe versehen. Um trotz diesen nur teilweise verlässlichen Informationen und den fehlenden Informationen über Bereiche zwischen den Kameras verlässliche Aussagen treffen zu können, wurde ein probabilistisches Verfahren entwickelt, dass durch die parallele Verfolgung aller möglichen Wege stets die bestmögliche Aussage über die Lage von Personen unter den bisher vorhandenen Informationen liefert. Unter Verwendung des “Forward Algorithmus”, einer Methode der Bayes'schen Inferenz, ist es möglich, diese komplexe Aufgabe zur Laufzeit zu lösen.
Mit dem Ziel eine umfassende Evaluation des Trackingsystems mit besonderem Fokus auf dem probabilistischen Modell zu ermöglichen, wurde eine Simulation verwendet und eine Schnittstelle zum entwickelten System entworfen. Die Auswertung einer umfassenden Menge an Parameterkombinationen zeigt die Zusammenhänge zwischen der Genauigkeit eingesetzter Sensoren und der erzielbaren Trackingleistung auf. Neben dem Erreichen fehlerfreier Ergebnisse unter Idealbedingungen von 100% Detektionsqualität und 100% Extraktionsqualität, ist es besonders interessant zu sehen, welche Ergebnisse mit fallender Extraktionsqualität

noch erreicht werden können. Selbst bei minimaler Extraktionsqualität von 10% kann bei 100 prozentiger Detektionsqualität noch ein Trackingerfolg von über 60% erreicht werden. Hier macht sich insbesondere die Leistung des probabilistischen Modells bemerkbar. Der nichtlineare Abfall der Trackingleistung mit sinkender Sensorgenauigkeit macht deutlich, dass durch das Modell Fehler auf Sensorebene zu einem Teil kompensiert werden können. Die Komplexität des Gesamtsystems sowie die große Menge einflussreicher Parameter bieten Ansatzpunkte zur weiteren Verbesserung und Erweiterung:

Open set

Die wichtigste Einschränkung der vorliegenden Arbeit besteht in der Beschränkung auf eine feste Gruppe bekannter Personen ("closed set"). Eine Erweiterung könnte darin bestehen das Tracking für eine unbekannte Anzahl von Personen, sowie den Umgang mit unbekannten Personen ("open set") zu ermöglichen. Neben den dadurch hinzukommenden Unsicherheitsfaktoren über die Anzahl der Personen, bestünde das Hauptproblem in der Unterscheidung zwischen bekannten und neuen, unbekannten Personen und der damit verbundenen Frage nach einer verlässlichen Initialisierung.

Erlernen der Netztopologie

Im vorliegenden System ist die Netzstruktur manuell festgelegt worden. Auch wenn diese initiale Strukturierung in vielen Anwendungsfällen machbar ist, wäre es trotzdem eine interessante Erweiterung, das System dahingehend zu modifizieren, dass ein autonomes erlernen der Netzstruktur allein anhand der Beobachtungen im Kameranetzwerk möglich ist. Zum diesem Thema verwandte Arbeiten [10, 11] könnten einen guten Ausgangspunkt darstellen.

Übergänge lernen

Eine Eigenschaft von Bayes Netzen, die im vorliegenden System weitesgehend ungenutzt bleibt ist die der Modellierung von Übergangswahrscheinlichkeiten. Diese stellen häufig eine mächtige Menge von Parametern dar und können das probabilistische Modell stark beeinflussen. Sind, anders als im vorliegenden System, reale Daten der Bewegungsabläufe von Personen verfügbar, lassen sich diese Parameter gewinnbringend nutzen. Dazu wird anhand einer großen Menge von Trainingsdaten bestimmt, welche Übergänge besonders häufig und welche weniger häufig verwendet werden. An dieser Stelle kann es - abhängig

von der Indentifikationssicherheit - auch sinnvoll sein, Personenspezifische Parameter zu verwenden, da einzelne Personen häufig jeweils eigenen, von anderen verschiedenen Bewegungsmustern folgen.

Bewegungsmuster vorhersagen

Ziel des vorgestellten Systems ist es, eine möglichst akkurate Aussage über den *aktuellen* Systemzustand, also über die Positionen aller Personen zu machen. Ausgehend von dem Wissen über die sich oftmals wiederholenden Bewegungsabläufe im Büroalltag oder ähnlichen Szenarien, ist es denkbar einen Schritt weiter zu gehen. Wenn das System in der Lage wäre aus der Masse an beobachteten Zustandsfolgen typische oder häufige Muster herauszufiltern, wäre es möglich Vorhersagen zu machen, die zeitlich über den aktuellen Systemzustand hinausgehen. Neben den denkbaren Anwendungsszenarien die sich aus dieser zusätzlichen Information ergeben, könnte sie gleichfalls zur Verbesserung der Trackingergebnisse beitragen weil mit ihr eine weitere Informationsquelle zur Bestimmung des Bewegungsverhaltens der Personen gegeben wäre.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] Oswald Lanz. Approximate bayesian multibody tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1436–1449, 2006.

[2] Michael Isard and John MacCormick. Bramble: A bayesian multiple-blob tracker. In *ICCV 2001*, volume 2, pages 34–41, 2001.

[3] N.S.V. Rao. On fusers that perform better than best sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):904–909, August 2001.

[4] J. Yang, J.Y. Yang, D. Zhang, and J.F. Lu. Feature fusion: parallel strategy vs. serial strategy. *Pattern Recognition*, 36(6):1369–1381, June 2003.

[5] Hanzi Wang and David Suter. Efficient visual tracking by probabilistic fusion of multiple cues. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 892–895, Washington, DC, USA, 2006. IEEE Computer Society.

[6] G. Kamberova and M. Mintz. Robust multi-sensor fusion: A decision-theoretic approach. In *DARPA90*, pages 867–873, 1990.

[7] A. Chilgunde, P. Kumar, S. Ranganath, and W.M. Huang. Multi-camera target tracking in blind regions of cameras with non-overlapping fields of view. In *BMVC04*, 2004.

[8] A. Gilbert and R. Bowden. Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity. In *ECCV06*, pages II: 125–136, 2006.

[9] A. Ilie and G. Welch. Ensuring color consistency across multiple cameras. In *ICCV05*, pages II: 1268–1275, 2005.

[10] D. Makris, T. Ellis, and J. Black. Bridging the gaps between cameras. In *CVPR04*, pages II: 205–210, 2004.

[11] J. Black, D. Makris, and T. Ellis. Validation of blind region learning and tracking. In *ICCCN '05: Proceedings of the 14th International Conference on Computer Communications and Networks*, pages 9–16, Washington, DC, USA, 2005. IEEE Computer Society.

[12] Hung H. Bui, Svetha Venkatesh, and Geoff West. *Hidden Markov models: applications in computer vision*, chapter Tracking and surveillance in wide-area spatial environments using the abstract hidden Markov model, pages 177–196. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2002.

[13] Nuria M. Oliver, Barbara Rosario, and Alex Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.

[14] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *CVPR 1997*, pages 994–999, 1997.

[15] Hilary Buxton and Shaogang Gong. Advanced visual surveillance using bayesian networks. In *International Conference on Computer Vision*, Cambridge, Massachusetts, June 1995.

[16] S. Gong and H. Buxton. Bayesian nets for mapping contextual knowledge to computational constraints in motion segmentation and tracking. In *Proceedings of 1993 British Machine Vision Conference*, September 1993.

[17] A.J. Abrantes, J.S. Marques, and J.M. Lemos. Long term tracking using bayesian networks. In *ICIP02*, pages III: 609–612, 2002.

[18] Peter Nillius, Josephine Sullivan, and Stefan Carlsson. Multi-target tracking - linking identities using bayesian network inference. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2187–2194, Washington, DC, USA, 2006. IEEE Computer Society.

[19] David Madigan, Wen-Hua Ju, P. Krishnan, A.S. Krishnakumar, and Ivan Zorych. Location estimation in wireless networks: A bayesian approach. *Statistica Sinica*, 16(2):495–522, 2006.

[20] W. Zajdel, B. Kröse, and N. Vlassis. *Intelligent Algorithms in Ambient and Biomedical Computing*, chapter Bayesian Methods for Tracking and Localization, pages 243–258. Springer Netherlands, 2006.

[21] W.P. Zajdel and B.J.A. Kröse. A sequential bayesian algorithm for surveillance with non-overlapping cameras. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 19(8):977–996, December 2005.

[22] W. Zajdel, Z. Zivkovic, and B. Kröse. Keeping track of humans: have i seen this person before? In *Proceedings of the IEEE Int. Conf. on Robotics and Automation 2005, Spain*, 2005.

[23] Kevin Karplus, Kimmen Sjolander, Christian Barrett, Melissa Cline, David Haussler, Richard Hughey, and Liisa Holm. Predicting protein structure using hidden markov models. Technical Report UCSC-CRL-97-13, 1997.

[24] S E Levinson. Continuously variable duration hidden markov models for automatic speech recognition. *Comput. Speech Lang.*, 1(1):29–45, 1986.

[25] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[26] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001. Foreword By-Raj Reddy.

[27] X. Jiang, M. Binkert, B. Achermann, and H. Bunke. Towards detection of glasses in facial images. In *ICPR '98: Proceedings of the 14th International Conference on Pattern Recognition-Volume 2*, page 1071, Washington, DC, USA, 1998. IEEE Computer Society.

[28] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[29] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[30] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[31] Gregory K. Wallace. The jpeg still picture compression standard. *Commun. ACM*, 34(4):30–44, 1991.

[32] M. Fitzgibbon, A. W.and Pilu and R. B. Fisher. Direct least-squares fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, May 1999.

[33] K. Bernardin, F. van de Camp, and R. Stiefelhagen. Automatic person detection and tracking using fuzzy controlled active cameras. In *VS07*, pages 1–8, 2007.

[34] Michael Jones Paul Viola. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518, 2001.

[35] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.

[36] G.R. Taylor, A.J. Chosak, and P.C. Brewer. Ovvv: Using virtual worlds to design and evaluate surveillance systems. In *VS07*, pages 1–8, 2007.

[37] Opencv library, `http://sourceforge.net/projects/opencvlibrary/`, 2001.

[38] T. Joachims. *Advances in Kernel Methods - Support Vector Learning*, chapter Making large-Scale SVM Learning Practical. MIT-Press, 1999.