Universität Karlsruhe
Fakultät für Informatik
Institut für Logik, Komplexität und Deduktionssysteme
Prof. A. Waibel

SS 2002

Studienarbeit

3D-Tracking von Gesicht und Händen mittels Farb- und Tiefeninformation

Kai Nickel

31. Mai 2002

Betreuer: Rainer Stiefelhagen

Inhaltsverzeichnis

1 Einleitung

Menschen verfügen über eine Vielzahl von Möglichkeiten, ihren Willen auszudrücken und ihre Vorstellungen verständlich zu machen. Sprache, Schrift, Mimik und Gestik werden im alltäglichen Leben situationsabhängig und oft kombiniert angewandt. Jede dieser Modalitäten hat ihre spezifischen Stärken und Schwächen. Die Festlegung auf eine einzige – wenn auch bisweilen gute Gründe dafür sprechen – wird gemeinhin als beschränkend und ineffizient empfunden.

Aktuelle Projekte im Bereich der multimodalen Mensch-Maschine-Interaktion sind folglich darauf ausgerichtet, vom Menschen als Benutzer ein umfassenderes Bild zu gewinnen, als dies durch die Interpretation von Tastatureingaben und Mausbewegungen alleine möglich ist. Erste Anwendungen aus der Sprach- und Handschrifterkennung haben bereits Einzug in unseren Alltag gehalten: Diktiersysteme, telefonische Fahrplanauskunft, maschinell verarbeitete Formulare, stiftbedienbare Kleinstcomputer, etc. Weiter gehende Systeme (automatische Übersetzung[1], Sprachverstehen) sind Gegenstand der Forschung.

Die Erfassung von Personen und ihrer Umgebung mittels immer kleinerer und leistungsfähigerer Videokameras ist die Grundlage für Systeme, die Personen und Objekte im Raum lokalisieren und in Beziehung setzen. Durch die Erkennung menschlicher Gesten (Hand-, Arm- oder Kopfbewegungen) und die Interpretation von Mimik bietet sich die Möglichkeit einer natürlicher wirkenden und effizienteren Kommunikation zwischen Mensch und Maschine.

Für Roboter, die sich selbstständig im menschlichen Lebensraum bewegen sollen[2], ist die Beobachtung von Personen in ihrer Umgebung unerlässlich. Auch sogenannte intelligente Räume sowie Anwendungen im Bereich der virtuellen Realität zählen zu den Einsatzgebieten des Personen-Trackings. Beispiele hierfür sind die Steuerung von Beleuchtung und multimedialen Installationen in Gebäuden, selbstständig protokollierende Besprechungsräume, Navigation in virtuellen Welten, etc.

Die vorliegende Arbeit beschäftigt sich mit dem Aufbau eines Systems zum Auffinden und Verfolgen von Gesicht und Händen: In der von einer Stereofarbkamera

[1] z. B. die Projekte Verbmobil (http://verbmobil.dfki.de/) und Nespole! (http://nespole.itc.it/)
[2] siehe auch: Sonderforschungsbereich „Humanoide Roboter“ (http://wwwiaim.ira.uka.de/sfb/)

gelieferten Bildsequenz soll nach dem Vorkommen einer Person gesucht werden, woraufhin die Positionen von Gesicht und Händen im dreidimensionalen Raum bestimmt und von Bild zu Bild verfolgt werden. Ein solches System kann beispielsweise Teil des Wahrnehmungsapparates eines Roboters und Grundlage zur Erkennung von menschlichen Gesten sein.

Die Arbeit gliedert sich in folgende Teile:

Kapitel 2 stellt Hautfarbe als Merkmal zur Lokalisierung von Gesicht und Händen in Farbbildern vor.

Kapitel 3 beschreibt die Verarbeitung von Stereobildern zur Gewinnung von 3D-Koordinaten und Tiefenbildern.

Kapitel 4 beschäftigt sich mit der Verknüpfung von Farb- und Tiefeninformation. Das Ergebnis ist eine Liste von Regionen, die Kandidaten für Gesicht und Hände darstellen.

Kapitel 5 zeigt, wie mithilfe eines Körpermodells aus der Kandidatenliste Kopf und Hände ausgewählt, und ihre Positionen von Bild zu Bild verfolgt werden können.

Kapitel 6 gibt eine kurze Zusammenfassung über das Erreichte.

In den Anhängen werden einzelne Methoden der Bildverarbeitung näher beschrieben, und es wird ein Überblick über Struktur und Funktionen der entwickelten Software gegeben.

2 Segmentierung durch Hautfarbe

Eine einfache und schnelle Möglichkeit, in einem Farbbild Gesichter, Hände und andere unbedeckte Körperteile zu lokalisieren, besteht darin, das Bild auf das Vorkommen von Hautfarbe hin zu untersuchen. Im Folgenden wird gezeigt, wie in einem gegebenen Bild Regionen gefunden werden können, die mit hoher Wahrscheinlichkeit menschliche Haut darstellen.

2.1 Farbraumtransformation

Nach [Yang et al. 97] ist bekannt, dass die zur Hautfarbe gehörenden Farbvektoren im RGB-Farbraum nicht gleichmäßig verteilt auftreten, sondern sich in einem begrenzten Bereich ballen. Ebenfalls wurde dort gezeigt, dass sich die Streuung der Farbvektoren noch deutlich verringern lässt, indem man dem Umstand Rechnung trägt, dass zur Identifikation von Hautfarbe die Helligkeit der Farbe keine Rolle spielt, sondern nur ihr reiner Farbwert:

Zwei Farbvektoren $[r_1, g_1, b_1]$ und $[r_2, g_2, b_2]$ haben trotz unterschiedlicher Helligkeit dieselbe Farbe, wenn gilt

$$\frac{r_1}{r_2} = \frac{g_1}{g_2} = \frac{b_1}{b_2}. \tag{2.1}$$

Im so genannten *chromatischen Farbraum* (auch: *rg-Farbraum*) werden diese beiden Farbvektoren auf denselben Punkt abgebildet, wodurch die Helligkeitsinformation eliminiert wird. Der chromatische Farbraum entsteht aus dem RGB-Farbraum durch Farbnormalisierung der Form

$$\begin{aligned} r &= R/(R+G+B), \\ g &= G/(R+G+B). \end{aligned} \tag{2.2}$$

Durch diese Transformation ergibt sich neben der Verringerung der Streuung der Hautfarbwerte auch eine erfreuliche Reduzierung der Dimension des Merkmalsraums $\mathbf{R}^3 \rightarrow \mathbf{R}^2$.

2.2 Modell der Hautfarbe

Um die Anordnung der Hautfarbwerte im Farbraum zu charakterisieren, ist es erforderlich, ein *Modell* der Hautfarbe zu erstellen. Prinzipiell ist die Repräsentation der Hautfarbverteilung durch ein parametrisches Modell, wie z. B. eine Gaußmischverteilung, oder durch ein nicht-parametrisches Modell, z. B. ein Histogramm, möglich. In der vorliegenden Arbeit kommt letzteres zur Anwendung.

Um die Verteilung der Farbwerte im zweidimensionalen rg-Farbraum zu modellieren, ist entsprechend auch ein zweidimensionales Histogramm erforderlich. Bedingt durch die hier verwendete Farbtiefe von 8 Bit hat das Histogramm in beiden Dimensionen eine maximale Auflösung von je 256 Werten. Die tatsächliche Auflösung kann auch geringer sein, was dazu führt, dass mehrere nebeneinander liegende Farbwerte einem einzigen Histogrammeintrag zugeordnet werden.

Initialisiert wird das Histogramm mithilfe eines oder mehrerer Beispielbilder, in denen die Regionen der Hautfarbe markiert wurden. Für jeden Pixel aus diesen Regionen wird der mit seiner Farbe (r, g) assoziierte Histogrammeintrag $H(r, g)$ um 1 erhöht. Teilt man $H(r, g)$ durch die Anzahl n der zum Histogramm beitragenden Pixel, berechnet sich die Wahrscheinlichkeit, mit der die Farbe (r, g) zu dem durch H gegebenen Hautfarbmodell gehört, wie folgt:

$$P\left((r, g)|Haut\right) = \frac{H(r, g)}{n} \tag{2.3}$$

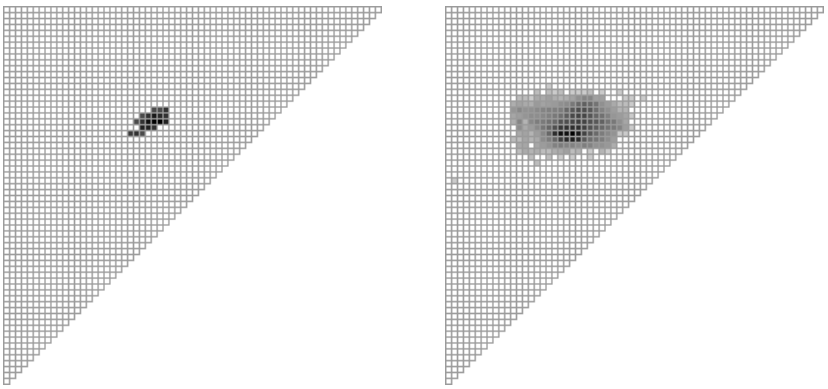
Normiert man H so, dass die Summe aller Histogrammeinträge den Wert 1 ergibt, so lässt sich oben genannte Wahrscheinlichkeit für eine Farbe $x = (r, g)$ direkt aus dem Histogramm ablesen. Sei H_+ dieses normierte Histogramm, so gilt:

$$P(x|Haut) = H_+(x) \tag{2.4}$$

Analog zum Histogramm H_+ als Modell der Hautfarbe lässt sich auch ein Histogramm H_- erstellen, zu dessen Initialisierung all die Pixel herangezogen werden, die *nicht* innerhalb der markierten Hautfarbregionen liegen. H_- ist somit ein Modell für die Klasse *Nicht-Hautfarbe*:

$$P(x|\neg Haut) = H_-(x) \tag{2.5}$$

Abbildung 2.1 ist ein Beispiel für die Histogramme der Klassen Hautfarbe und Nicht-Hautfarbe. Die dreieckige Form der Histogramme ergibt sich aus der Form des rg-Farbraums. Je dunkler die Farbe eines Histogrammfelds ist, desto höher ist sein Wert.



H_+ H_-

Abbildung 2.1: Histogramme

2.3 Klassifikation

Nach der Regel von Bayes errechnet sich die Wahrscheinlichkeit, dass es sich um Hautfarbe handelt, wenn eine Farbe x beobachtet wird, wie folgt:

$$P(Haut|x) = \frac{P(x|Haut) \cdot P(Haut)}{P(x)} \tag{2.6}$$

Analog ergibt sich die Wahrscheinlichkeit, dass es sich bei x *nicht* um Hautfarbe handelt:

$$P(\neg Haut|x) = \frac{P(x|\neg Haut) \cdot P(\neg Haut)}{P(x)} \tag{2.7}$$

Eine Farbe x kann als Hautfarbe („positiv“) klassifiziert werden, wenn gilt:

$$P(Haut|x) > P(\neg Haut|x) \tag{2.8}$$

Aus 2.6 bis 2.8 sowie $P(Haut) + P(\neg Haut) = 1$ folgt für die positive Klassifikation:

$$\frac{P(x|Haut)}{P(x|\neg Haut)} > \frac{1 - P(Haut)}{P(Haut)} \tag{2.9}$$

$P(x|Haut)$ und $P(x|\neg Haut)$ lassen sich gemäß 2.4 und 2.5 direkt aus den normierten Histogrammen H_+ und H_- ablesen. Die a-priori Wahrscheinlichkeit $P(x)$ entfällt. Die a-priori Wahrscheinlichkeit $P(Haut)$ des generellen Vorkommens von Hautfarbe in den Beispielbildern ist eine Konstante und lässt sich z. B. anhand der Größenverhältnisse Kopf-Bild grob mit 0,1 abschätzen, wodurch sich für die gesamte rechte Seite von 2.9, im folgenden mit S bezeichnet, ein Wert von 9 ergeben würde. Wie man sieht, lässt sich S auch als Schwellwert für die Strenge der

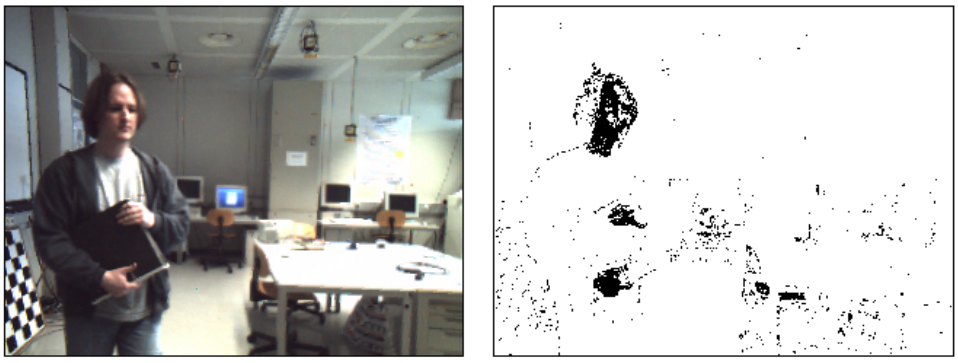


Abbildung 2.2: Klassifikation nach Hautfarbe

Klassifikation ansehen: je höher S gewählt wird, desto weniger Pixel werden als Hautfarbe akzeptiert.

Zusammenfassend gilt: Ein Farbwert x wird positiv klassifiziert, wenn

$$\frac{H_+(x)}{H_-(x)} > S. \tag{2.10}$$

Abbildung 2.2 zeigt das Ergebnis dieses Vorgangs: Jeder Pixel des Farbbildes auf der linken Seite wurde gemäß 2.10 klassifiziert und im rechten Bild schwarz (für Hautfarbe) oder weiß (für Nicht-Hautfarbe) eingezeichnet.

2.4 Hautfarbmaske

Ein Problem bei der weiteren Verwendung des Hautfarbenklassifikationsbildes besteht darin, dass es eine „pixelige“ und keine flächige Strukur aufweist: Selbst innerhalb geschlossener Hautfarbflächen werden einzelne Pixel nicht als Hautfarbe klassifiziert, wodurch „Löcher“ entstehen. Umgekehrt tauchen auch in Bereichen ohne Hautfarbe immer wieder einzelne Pixel auf, die fälschlicherweise auf Hautfarbe hinweisen.

Um diese störenden Effekte zu eliminieren, wird das Bild mit morphologischen Operatoren (siehe Anhang A.1) behandelt: Zuerst werden durch morphologisches Schließen (Dilatation mit anschließender Erosion) mit einem 3x3-Strukturelement kleinere Lücken in den positiv klassifizierten Bereichen geschlossen. Danach wird das Bild durch morphologisches Öffnen (Erosion + Dilatation, ebenfalls 3x3) von vereinzelten schwarzen[1] Pixeln gereinigt. Abbildung 2.3 zeigt die Wirkung der Operationen an einem Beispiel.

[1] In der hier verwendeten Darstellung sind gesetzte Pixel (Wert=255) *schwarz* eingezeichnet.

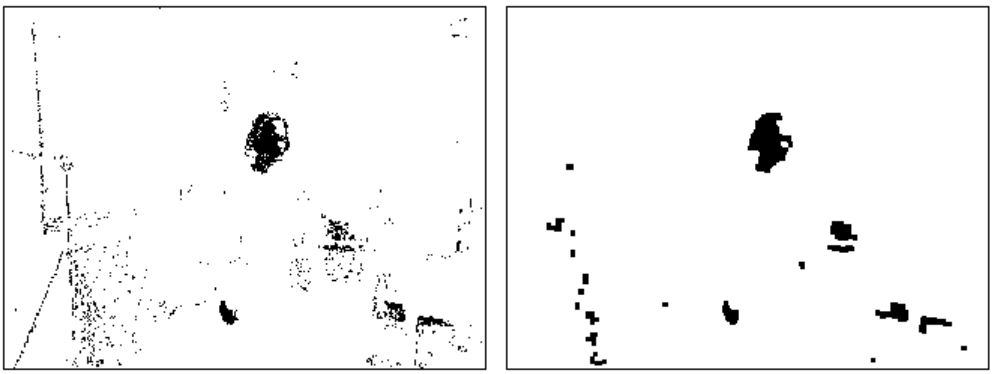


Abbildung 2.3: Anwendung morphologischer Operatoren

Das Ergebnis ist die *Hautfarbmaske*: ein Schwarzweißbild, welches über das Ursprungsbild gelegt nur noch hautfarbene Bereiche erscheinen lässt.

2.5 Initialisierung

Von zentraler Bedeutung für ein gutes Klassifikationsergebnis ist die Initialisierung des Modells. Ist die Markierung der Hautregionen in den Beispielbildern schlecht, befinden sich also „falsche“ Farbwerte unter den Hautfarbwerten, so führt dies dazu, dass die Klassifikation an Trennschärfe verliert und im Extremfall sogar „umkippt“. Dann werden weite Teile des Bildes fälschlicherweise positiv klassifiziert, und die eigentliche Hautfarbe wird zurückgewiesen.

Das Problem der fehlerhaften Markierung der Hautregionen tritt insbesondere dann auf, wenn die Markierung nicht manuell erfolgt, und statt dessen die Position des Kopfes anhand anderer Merkmale (siehe Kapitel 5) automatische bestimmt und die entsprechende Bildfläche zur Initialisierung herangezogen wird. Für den Fall, dass eine solche automatische Auswahl misslingt, und anstatt des Kopfes ein ganz anderes Objekt präsentiert wird[2], wäre es wünschenswert, wenn das so entstandene Modell der Hautfarbe als „schlecht“ zurückgewiesen werden könnte.

Initialisiert man ein Modell mit Farbwerten aus einem markierten Beispielbild und klassifiziert dann mit dem neuen Modell eben jenes Bild, so stellt man folgendes fest: Das Modell ist sicherlich nur dann brauchbar, wenn es innerhalb der Markierung im Schnitt eine wesentlich höhere Anzahl von Pixeln positiv klassifiziert als außerhalb der Markierung.

Abbildung 2.4 zeigt ein Beispielbild, das einmal mit einer schlechten Markierung (rot) und einmal mit einer guten Markierung (grün) zur Initialisierung eines Mo-

[2]Problematisch ist auch, wenn das Gesicht der Kamera abgewandt ist, so dass anstatt Haut nur Haare zu sehen sind.

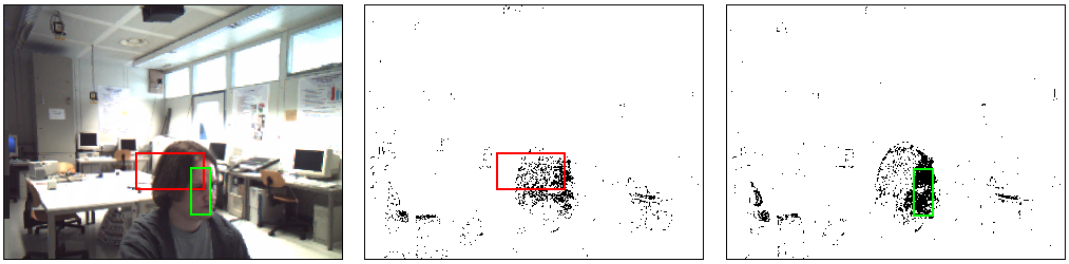


Abbildung 2.4: Initialisierung mit schlechter (rot) und guter (grün) Markierung.

dells verwendet wurde. Am Klassifikationsergebnis kann man erkennen, dass im Falle der schlechten Markierung aufgrund ihrer farblichen Inhomogenität die Anzahl der positiv klassifizierten Pixel innerhalb der Markierung deutlich geringer ist als bei der guten Markierung.

Im später vorgestellten Programm wird ein Modell daher nur dann akzeptiert, wenn es innerhalb der Markierung mindestens 50% aller Pixel, und außerhalb der Markierung maximal ein Zehntel des inneren Prozentsatzes positiv klassifiziert. Schon durch diesen einfachen Vergleich gelingt es, einen Großteil der fehlerhaften Markierungen auszuschließen.

2.6 Probleme

Bei der Verwendung des Merkmals Hautfarbe zur Lokalisierung von Gesicht und Händen treten prinzipbedingt einige Probleme auf. Eines besteht darin, dass es Oberflächen gibt, die farblich nicht von Haut zu unterscheiden sind. In diesem Zusammenhang sind vor allem Holz (wie auch in Abbildung 2.2 zu erkennen ist) sowie einige Textilien zu nennen. Bei letzteren besteht besondere Verwechslungsgefahr, da sie direkt am Körper getragen werden.

Eine andere Schwierigkeit stellen durch Überbelichtung weiß gewordene Bildbereiche dar, die durch ungünstige Voreinstellung der Kamerablende, reflektierende Oberflächen oder generell durch sich schnell ändernde Lichtverhältnisse entstehen können[3]. Auch ist es bei Motiven mit großem Helligkeitsumfang schwer möglich, in den hellen Gebieten Überbelichtung zu vermeiden und gleichzeitig die dunklen Bildteile noch farblich aufzulösen.

Generell ist es so, dass ein Modell nur für die Zustände Gültigkeit besitzt, die zum Zeitpunkt seiner Initialisierung herrschten. Die Klassifikation versagt, wenn sich, z. B. durch Verwendung einer anderen Kamera oder Veränderungen in der

[3]Diese Änderungen treten z. B. ein bei Kamerabewegungen, Abschattung, Anschalten einer Lampe, ...

spektralen Zusammensetzung des Umgebungslichts[4], die von der Kamera gelieferten Farbwerte verschieben. Um das Modell aufrecht zu erhalten, muss es also in regelmäßigen Abständen an die geänderten Verhältnisse angepasst werden.

Insbesondere ein mobiler Roboter wird mit sich ständig ändernden Lichtverhältnissen konfrontiert sein und ist folglich darauf angewiesen, ständig neue Beispiele für Hautfarbwerte geliefert zu bekommen. Eine Möglichkeit, diese Daten zu beziehen, ohne sich dabei auf ein bestehendes Modell der Hautfarbe stützen zu müssen, ergibt sich aus der Stereobildverarbeitung, wie sie im Kapitel 3 vorgestellt wird.

[4]Die spektrale Zusammensetzung von Sonnenlicht ist tageszeitabhängig, die von künstlicher Beleuchtung ist abhängig vom verwendeten Leuchtmittel (Halogenbirne, Glühbirne, Leuchtstoffröhre, etc.).

3 Stereobildverarbeitung

Durch die Kombination der Bilder zweier Kameras ist es möglich, die Position eines Objektes im dreidimensionalen Raum zu bestimmen, wenn es von beiden Kameras gleichzeitig erfasst wird. Die so gewonnene Information wird im Rahmen der vorliegenden Arbeit genutzt, um das Auffinden von Gesicht und Händen zu erleichtern und deren Bewegungen entlang einer Trajektorie im Raum zu beschreiben.

3.1 Stereogeometrie

Abbildung 3.1 zeigt zwei parallel ausgerichtete Kameras mit den Brennpunkten L bzw. R, deren Bildebenen in einer gemeinsamen Ebene eingebettet liegen. b bezeichnet den horizontalen Abstand der Kameras (Grundlinie) und f die Brennweite. Beide Kameras beobachten ein Objekt im Punkt P, das sich in der Entfernung z zur Basislinie befindet. Als horizontale Bildkoordinate[1] von P ergibt sich für die linke Kamera der Wert x_L und für die rechte Kamera der Wert x_R. Der Abstand $d := x_L - x_R$ wird als *(horizontale) Disparität* (siehe [Jähne 97]) bezeichnet. Mithilfe des Strahlensatzes ergibt sich für die Entfernung z die Beziehung

$$z = \frac{f \cdot b}{d} \tag{3.1}$$

Ist der Abstand z bekannt, lassen sich auch die x- und y-Koordinaten von P im Raum berechnen. In einem Koordinatensystem, dessen Ursprung im Brennpunkt der linken Kamera liegt, gilt:

$$x = \frac{x_L \cdot z}{f}, \quad y = \frac{y_L \cdot z}{f} \tag{3.2}$$

Die Entfernung ist umgekehrt proportional zur Disparität, eine Disparität von 0 steht für unendliche Entfernung. Je näher ein Objekt den Kameras kommt, desto stärker verschieben sich seine Abbilder im linken und im rechten Kamerabild

[1] Der Nullpunkt befindet sich in der Bildmitte.

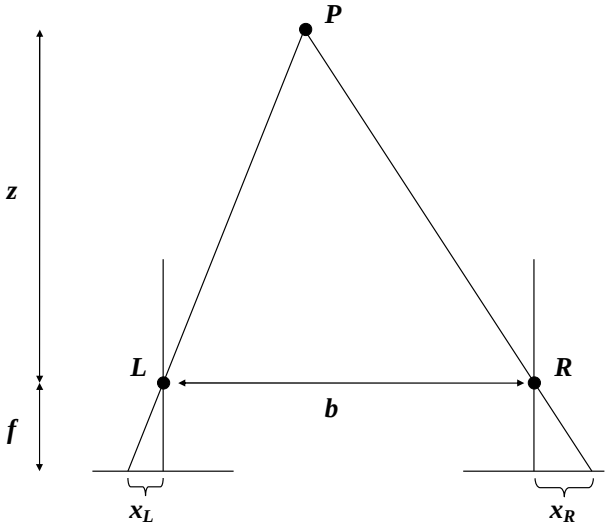


Abbildung 3.1: Stereogeometrie

gegeneinander; die Disparität steigt. In der Praxis gibt es bei parallel ausgerichteten Kameras einen maximalen Wert für die Disparität, so dass der Bereich des dreidimensionalen Sehens eingeschränkt wird. Um die Tiefeninformation nicht zu verlieren, muss also ein Mindestabstand von den Kameras eingehalten werden. Die Ebene, die den nutzbaren Bereich begrenzt, wird als *Horopter* bezeichnet.

3.2 Disparitätenbild

Ein Disparitätenbild (dense disparity map, „Tiefenbild“) ist ein Graustufenbild, in dem der Helligkeitswert eines Pixels für die Disparität steht, welche die beiden korrespondierenden Punkte aus dem linken und rechten Bild zueinander aufweisen. Zur Erstellung des Disparitätenbildes ist es also notwendig, in beiden Kamerabildern jene Teile zu identifizieren, die Abbildungen desselben Objekts sind (*Korrespondenzproblem*).

Es gibt verschiedene Möglichkeiten, Korrespondenzen zu finden, die sich hinsichtlich des Berechnungsaufwands und der Genauigkeit unterscheiden. Die hier zur Disparitätenberechnung eingesetzte SVS-Bibliothek (siehe Abschnitt B.1.1) geht nach [Konolige 97][2] folgendermaßen vor:

Die Graustufenbilder beider Kameras werden zuerst mit einem Laplacian-of-Gaussian-Filter (siehe [Horn 86]) behandelt, um die Kanten hervorzuheben.

[2]Zusätzliche Informationen über den SVS-Stereoalgorithmus finden sich im SVS User's Manual sowie unter http://www.ai.sri.com/∼konolige/

Abbildung 3.2: Linkes und rechtes Kamerabild, Disparitätenbild

Um festzustellen, ob es sich bei einem Punkt im linken und einem Punkt im rechten Bild um Abbildungen desselben Objekts handelt, wird ein kleines quadratisches Suchfenster (area correlation window) mit 5-13 Punkten Breite um jeden der Punkte gelegt. Von den Helligkeitswerten der entsprechenden Punkte unter den Fenstern wird die (absolute) Differenz gebildet. Die Summe über alle Differenzen ist ein Maß für die Korrelation der Bildteile unter den Fenstern: je geringer die Summe der Differenzen, desto größer die Korrelation.

Für jedes Fenster im linken Bild wird das maximal korrelierende Fenster im rechten Bild gesucht. Dazu werden die Suchfenster über alle Punkte des linken und des rechten Bildes geschoben. Da die Kameras nur in horizontaler Richtung voneinander versetzt angebracht sind, können beide Suchfenster immer auf gleicher vertikaler Position bleiben und müssen nur horizontal gegeneinander verschoben werden.

Hat man die maximale Korrelation für ein Fenster um einen Punkt im linken Bild gefunden worden, ergibt sich die Disparität dieses Punktes aus dem horizontalen Abstand der korrelierenden Fenster. Zur Begrenzung der Rechenzeit kann der horizontale Suchbereich und somit die maximal mögliche Disparität z. B. auf 64 Pixel beschränkt werden.

In einem Nachbearbeitungsschritt werden die Disparitätenwerte an jenen Stellen im Bild wieder entfernt, an denen die Stärke der sichtbaren Textur ein gegebenes Konfidenzmaß nicht erfüllt (confidence filter).

Die Kameras sehen das Bild aus leicht unterschiedlichen Blickwinkeln. Dadurch entstehen an Objekträndern Bereiche, die nur von einer Kamera gesehen werden, was zu fehlerhaften Disparitätenwerten führt. Diese Bereiche können gefunden werden, indem zunächst das Bild der linken Kamera als fester Ausgangspunkt bei der Suche nach Korrelation verwendet wird, und dann der Vorgang noch einmal mit dem Bild der rechten Kamera wiederholt wird. Unterscheiden sich an einem Punkt die beiden so gefundenen Disparitätenwerte, werden sie entfernt (left-/right filter).

Abbildung 3.2 zeigt einen Frame aus einer Videosequenz (linke und rechte Kamera) und das daraus errechnete Disparitätenbild. Ein schwarzer Punkt steht für

maximale Disparität. Punkte, für die keine Disparitätenwerte errechnet werden konnte, erscheinen weiß. Die nutzbare Bildfläche des Disparitätenbildes ist geringer als die Größe der Kamerabilder, da sich das Sichtfeld der beiden Kameras an den Bildrändern nicht überlappt. Es wird deutlich, dass die Disparitätenberechnung auf schwach texturierten Oberflächen (wie beispielsweise der Wand im Hintergrund) nicht gelingt. Flächen außerhalb des Horopters (im Beispielbild die linke untere Ecke) zerfallen in zufällige Disparitätenwerte.

3.3 Kalibrierung

Die bisherigen Ausführungen gingen von vollkommen parallel ausgerichteten Lochkameras aus, wie es sie in der Realität nicht gibt. Durch Linsenverzerrungen, kleine Unterschiede in der Brennweite und Abweichungen von der idealen Ausrichtung entstehen fehlerhafte Bilder, die die korrekte Berechnung von Disparitäten verhindern. Daher ist es notwendig, die Kameraanordnung vor Gebrauch zu *kalibrieren.*

Durch die Kalibrierung werden zwei Arten von Parametern gewonnen, so genannte *innere* und *äußere* Parameter. Deren Kenntnis ermöglicht es, die Bilder so zu transformieren, dass sie den Bildern idealer Kameras entsprechen.

Zu den inneren Parametern (intrinsics) einer Kamera gehören Brennweite, Linsendezentrierung, radialsymmetrische Verzerrung und das Pixel-Seitenverhältnis des Bildsensors. Die äußeren Parameter (extrinsics) beschreiben die räumliche Anordnung beider Kameras – insbesondere den Grundlinienabstand und die Abweichung der optischen Achsen von der Parallele.

Die verwendete SVS-Bibliothek unterstützt die Kalibrierung mithilfe eines Schachbrettmusters, das in fünf verschiedenen Ansichten aufgenommen wird (siehe Abbildung 3.3). Aus diesen Bildern werden die oben genannten Parameter extrahiert, um sie später im Stereo-Algorithmus zur Korrektur der Kamerabilder zu verwenden. Eine kurze Beschreibung des Verfahrens gibt [Konolige et al. 01].

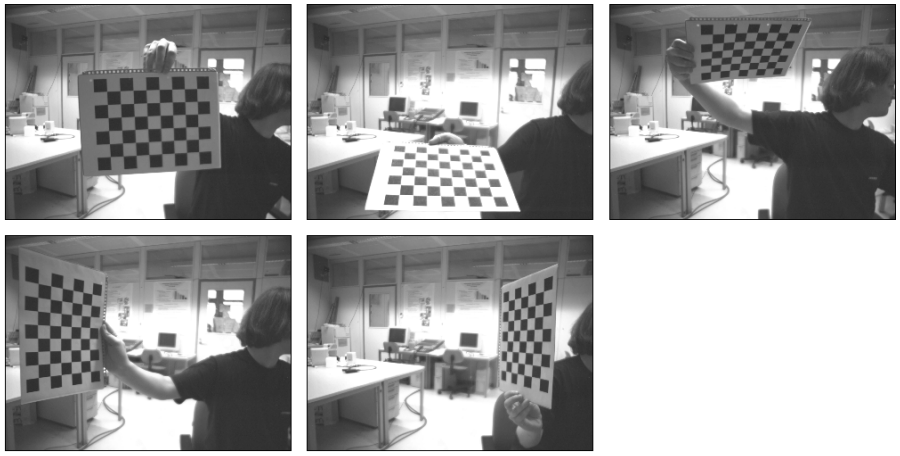


Abbildung 3.3: Kalibrierung mit einem Schachbrettmuster

4 Kombination von Farbe und Tiefe

Ein Ergebnis aus Kapitel 2 ist die Hautfarbmaske, deren Nutzen bei der Suche nach Gesicht und Händen dadurch eingeschränkt wird, dass sie auch andere im Bild vorkommende Oberflächen mit einschließt, die farblich nicht von menschlicher Haut unterschieden werden können (z. B. Holz). In ungünstigen, aber häufig vorkommenden Fällen verschmilzt zudem eine Region, die tatsächlich Haut zeigt, mit einer anderen Region, die nur eine hautähnliche Oberfläche zeigt.

Im Folgenden wird eine Methode beschrieben, das aus Kapitel 3 stammende Disparitätenbild so mit der Hautfarbmaske zu verknüpfen, dass eine neue Maske entsteht, die besser als die Hautfarbmaske alleine Gesicht und Hände freizustellen vermag. Damit soll dann eine Liste von isolierten Bildregionen (Komponenten, Blobs) erstellt werden, die jeweils möglicherweise ein Gesicht oder eine Hand zeigen.

4.1 Tiefenbereichsmaske

Das durch die Stereobildverarbeitung gewonnene Disparitätenbild kann man sich als Segmentierung des Kamerabildes nach Tiefenebenen vorstellen. Alle Pixel im Disparitätenbild, die den gleichen Wert haben, liegen auf einer gemeinsamen Ebene im Raum, die parallel zur Bildebene ist[1].

Wenn bekannt ist, in welcher Entfernung zur Kamera sich die zu beobachtende Person befindet, lassen sich aus dem Disparitätenbild genau jene Pixel auswählen, die sich in ähnlicher Entfernung befinden. Die so ausgewählten Pixel bilden eine Maske, die im Folgenden als *Tiefenbereichsmaske* bezeichnet wird. Der Umfang des durch die Maske abgedeckten Tiefenbereiches ist das sogenannte *Tiefenfenster*.

Die Entfernung der zu beobachtenden Person kann anhand ihrer Entfernung im vorangegangenen Bild der Videosequenz geschätzt werden. War die Person im

[1] Ausnahme sind die Pixel mit dem Wert 0, der für unendliche bzw. unbekannte Entfernung steht.

Abbildung 4.1: Kamerabild, Disparitätenbild, Tiefenbereichsmaske

vorangegangenen Bild beispielsweise 2,50m von der Kamera entfernt, so muss die Tiefenbereichsmaske für das aktuelle Bild all jene Pixel umfassen, die Entfernungen zwischen etwa 1,50m und 3,00m aufweisen[2]. Diese Werte ergeben sich aus dem Körpermodell, das in Kapitel 5 vorgestellt wird. Abbildung 4.1 zeigt eine solche Tiefenbereichsmaske.

Wenn sich noch keine Person im Bild befindet, so kann das Tiefenfenster zumindest noch auf den Entfernungsbereich eingeschränkt werden, außerhalb dessen – z. B. bedingt durch die Kameraauflösung oder den Horopter – ohnehin keine sinnvolle Beobachtung mehr möglich ist. Die oft zahlreichen Pixel, die unendliche oder unbekannte Entfernung haben (z. B. die weißen Pixel des Disparitätenbildes in Abbildung 4.1), fallen zusätzlich weg.

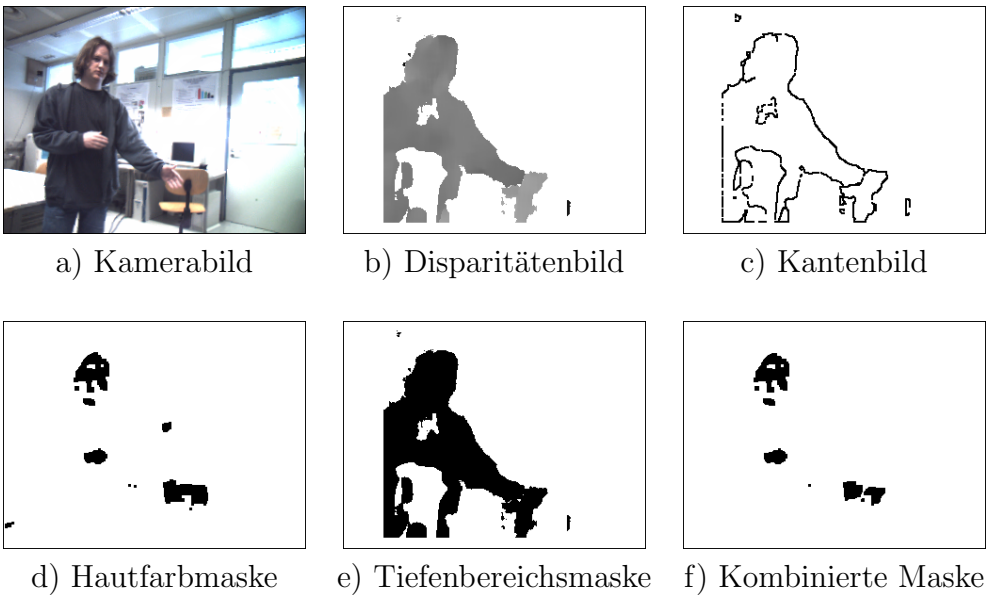
Auf diese Weise kann die zu verfolgende Person unter günstigen Umständen – wenn sie frei vor der Kamera steht und keine anderen Objekte sich auf gleicher Tiefenebene befinden – vollständig vom Hintergrund getrennt werden. Ein ähnlicher Effekt kann durch das Verfahren der *Background-Subtraction* (siehe [Wren et al. 97]) erzielt werden. Dazu wird ein Modell des statischen Bildhintergrundes aufgebaut. Plötzliche Veränderungen durch neu hinzugekommene, sich bewegende Gegenstände oder Personen können so erfasst werden.

Anders als die Background-Subtraction ist das hier beschriebene Verfahren auch für sich frei bewegende Kameras und dynamische Hintergründe geeignet, was insbesondere für den Einsatz in mobilen Robotern notwendig ist.

4.2 Fusion der Masken

Hautfarbmaske und Tiefenbereichsmaske lassen sich beispielsweise mit einem binären UND-Operator verknüpfen. Übrig bleiben nur die Punkte, die in beiden

[2]Das Tiefenfenster muss so großzügig gewählt sein, dass ein in Richtung Kamera ausgestreckter Arm noch erfasst werden kann.



a) Kamerabild b) Disparitätenbild c) Kantenbild

d) Hautfarbmaske e) Tiefenbereichsmaske f) Kombinierte Maske

Abbildung 4.2: Kombination von Farb- und Tiefenbereichsmaske

Masken gesetzt sind – die also sowohl Hautfarbe zeigen, als auch in einem Entfernungsbereich vor der Kamera liegen, in dem die zu beobachtende Person vermutet wird.

Auch im reduzieren Tiefenbereich kann es vorkommen, dass Regionen miteinander verschmelzen, die zwar beide Hautfarbe zeigen, aber nicht zum selben Objekt gehören. Befinden sich zwei derartige, im Bild aneinander grenzende Regionen auf deutlich unterschiedlichen Tiefenebenen, dann ist es – wie in [Darrel et al. 98] beschrieben – möglich, sie folgendermaßen zu trennen: Auf dem Disparitätenbild wird eine *Kantendetektion* (siehe Anhang A.2) durchgeführt. Das Ergebnis ist ein Bild, in dem all jene Pixel gesetzt sind, in deren Umgebung sich der Wert der Disparität stark ändert. Subtrahiert man dieses Kantenbild von der im letzten Absatz beschriebenen kombinierten Maske, so entstehen die gewünschten Lücken zwischen Bereichen unterschiedlicher Tiefe[3].

Abbildung 4.2 zeigt an einem Beispiel das hier beschriebene Verfahren: Hautfarbmaske und Tiefenbereichsmaske werden mit einem UND-Operator zur kombinierten Maske verknüpft. Ein Hintergrundbereich in der Bildmitte, der starke Hautfarbcharakteristik aufweist aber nicht mit der Person auf einer Tiefenebene liegt, wird dadurch eliminiert. Hand und Stuhllehne bilden in der Hautfarbmaske eine Einheit, können aber durch Subtraktion des Kantenbildes voneinander getrennt werden, da sie sich in ihrer Entfernung zur Kamera unterscheiden.

[3]Die Trennwirkung des Kantenbildes wird noch verstärkt, indem man es vor der Subtraktion per Dilatation mit einem 2x2 Strukturelement „verdickt“.

4.3 Komponentenanalyse

Die kombinierte Maske stellt im Prinzip nichts anderes dar als eine Menge von Pixeln, von denen vermutet wird, dass sie Teil des Abbildes eines Gesichts oder einer Hand sind.

Führt man eine *Komponentenanalyse* (siehe Anhang A.3) auf der kombinierten Maske durch, dann erhält man eine Liste voneinander getrennter, aber in sich zusammenhängender Regionen (*Komponenten*), die charakterisiert werden durch:

- Die Position und das Ausmaß ihrer bounding box[4],
- den Umriss (Kontur) in Form einer Aufreihung aller Randpixel,
- die Fläche,
- den Mittelpunkt (centroid, Massenschwerpunkt).

Mithilfe der Stereogeometrie (Kapitel 3) lassen sich für jede Komponente zusätzlich berechnen:

- Die Position des Mittelpunktes im Raum (x, y und z Entfernung vom Brennpunkt der linken Kamera in Metern),
- die reale Fläche in m^2.

Betrachtet man die resultierenden Komponenten, so stellt man fest, dass offensichtlich zusammengehörende Bereiche gelegentlich durch feine Risse unterteilt sind und dadurch als unterschiedliche Komponenten angesehen werden. In einem Nachbearbeitungsschritt werden daher alle Komponenten miteinander vereinigt, deren bounding boxes sich überlappen[5], die sich aber in ihrer Entfernung zur Kamera nur geringfügig unterscheiden[6].

Jede der übrig gebliebenen Komponenten ist ein Kandidat für ein Gesicht oder eine Hand.

[4] Die bounding box ist das kleinste Rechteck, das eine Region vollständig umschließt.

[5] Für das Vereinigen von Komponenten sind auch andere Kriterien denkbar wie z. B. ein geringer Mittelpunktsabstand.

[6] Würde diese Bedingung nicht eingehalten, gingen die positiven Effekte der Kantendetektion wieder verloren.

5 Körpermodell

Im Folgenden wird beschrieben, wie aus der Liste der Kandidaten aus Kapitel 4 Kopf und Hände ausgewählt, und deren Positionen fortlaufend von Bild zu Bild verfolgt werden.

5.1 Klassifikation

Um von einer gegebenen Komponente sagen zu können, ob es sich um eines – und wenn ja, um welches – der gesuchten Körperteile handelt, muss man sich zunächst Informationen über Ausmaß und Form dieser Körperteile beschaffen.

Von einem aufrecht gehaltenen Kopf könnte man beispielsweise annehmen, dass er durchschnittlich eine Höhe von $25cm$, eine Breite von $18cm$ und eine Fläche von $400cm^2$ hat[1]. Im so genannten *Körpermodell* werden alle Annahmen dieser Art zusammengefasst. Die Verwendung der *realen* Größe im Unterschied zur Pixelgröße ist hier wesentlich – und wird erst durch die aus der Stereobildverarbeitung gewonnene Tiefeninformation möglich. Tabelle 5.1 zeigt das in dieser Arbeit verwendete Körpermodell.

Die beobachteten Merkmale werden als normalverteilt angesehen, weshalb im Köpermodell neben dem Mittelwert μ auch die Standardabweichung σ angegeben ist. Neben dem Ausmaß, der Fläche und dem Seitenverhältnis der Körperteile befindet sich auch der maximale Hand-Kopf-Abstand und die maximal mögliche Geschwindigkeit der Bewegung von Kopf und Händen im Körpermodell.

Gewonnen wurden die Werte durch Auswertung mehrerer realer Videoaufnahmen von drei verschiedenen Personen über einen Gesamtzeitraum von ungefähr drei Minuten. Die Maximalgeschwindigkeiten und der maximale Kopf-Hand-Abstand wurden geschätzt.

[1] Streng genommen handelt es sich in dieser Arbeit immer nur um einen Teil des Kopfes, nämlich den, der Hautfarbe trägt (Gesicht und sichtbarer Teil des Halses). In der Praxis spielt diese Unterscheidung aber keine große Rolle, zumal auch Haare in aller Regel eine starke Hautfarbcharakteristik aufweisen.

	μ	σ
Kopf, Höhe	0,275 $[m]$	0,040
Kopf, Breite	0,200 $[m]$	0,040
Kopf, Fläche	0,0300 $[m^2]$	0,0130
Kopf, Verhältnis[2]	1,41	0,35
Hand, Höhe	0,100 $[m]$	0,045
Hand, Breite	0,100 $[m]$	0,045
Hand, Fläche	0,0045 $[m^2]$	0,0040
Person, Höhe	1,60 $[m]$	0,15
Person, Breite	0,90 $[m]$	0,20
Person, Fläche	0,50 $[m^2]$	0,10
Person, Verhältnis	2,1	0,3

	Maximalwert
Entfernung Kopf-Hand	1,1 $[m]$
Geschwindigkeit Kopf	1,5 $[m/sec]$
Geschwindigkeit Hand	2,5 $[m/sec]$

Tabelle 5.1: Körpermodell

Es ist auffällig, wie hoch die Standardabweichung bei der Größe der Hände ist[3]. Der Grund dafür ist, dass das zweidimensionale Abbild einer Hand je nach Position, Abschattung und Verdeckung durch den Ärmel sehr stark variiert. Die hohe Standardabweichung bei der Höhe einer Person liegt darin begründet, dass Personen, die einen gewissen Abstand zur Kamera unterschreiten, nicht mehr in ganzer Höhe von der Kamera erfasst werden können.

Für die Klassifikation der Komponenten wurde folgender Ansatz gewählt: Für eine gegebene Komponente c lässt sich ein Maß S („Score“) für die Zugehörigkeit zur Klasse K ($K = Kopf, Hand, Person$) angeben, indem man jedes gemessene Merkmal $x_i(c)$ ($i = Hoehe, Breite, \dots$) in die Dichtefunktion der Normalverteilung N einsetzt, und hieraus das Produkt bildet:

$$S_K(c) = \prod_i N(x_i(c)|\mu_{K,i}, \sigma_{K,i}) \tag{5.1}$$

Je höher $S_K(c)$ ist, desto stärker stimmt die Komponente c in ihren Eigenschaften $x_i(c)$ mit dem imaginären Prototypen aus der Klasse K überein. $S_K(c)$ wird auf 0 gesetzt, wenn mindestens einer der Messwerte so stark aus dem Rahmen fällt,

[3]Durch die Anwendung der morphologischen Operatoren wachsen die Flächen leicht an, was die allgemein recht hohen Mittelwerte erklärt.

dass er mehr als 2,5 Standardabweichungen vom Mittelwert entfernt liegt[4]. Die Komponente c wird in diesem Fall als nicht zur Klasse K gehörig gewertet.

Die hier verwendeten Merkmale wie Breite und Höhe stellen zusammen mit dem beschriebenen Verfahren einen sehr einfachen Klassifikationsansatz dar. Es sind auch aufwändigere Verfahren denkbar, wie z. B. der Vergleich der Form des Kopf-Kandidaten mit einer Ellipse oder der Einsatz eines neuronalen Netzes (siehe [Rowley et al. 96]) zur Identifikation von Gesichtern.

5.2 Tracking

Beim Start des Systems ist noch keine Kopfposition bekannt, und das Tiefenfenster ist vollständig geöffnet. In jedem neuen Bild (*Frame*) der Videosequenz wird jede Komponente aus der Kandidatenliste mithilfe des Körpermodells daraufhin überprüft, ob es sich bei ihr um einen Kopf handeln kann ($S_{Kopf}(c) > 0$). Ist das der Fall, wird diese Komponente fortan als Kopf angesehen. Gibt es in einem Frame mehrere mögliche Kandidaten für einen Kopf, wird der mit dem größten Kopf-Score ausgewählt.

Ist ein Kopf vorhanden, kann in seiner Nachbarschaft, die durch die physisch maximal mögliche Kopf-Hand-Distanz begrenzt wird, nach Händen gesucht werden. Die beiden Kandidaten mit dem größten Hand-Score werden ausgewählt.

Die Auswahl der Hände erfolgt erst nach der Auswahl des Kopfes, weil ein Kopf relativ zuverlässig zu identifizieren ist, während kleine Regionen mit den Eigenschaften von Händen sehr viel häufiger im Bild anzutreffen sind. Die Einschränkung auf den Kopfradius hilft wesentlich dabei, aus der Reihe der Hand-Kandidaten die richtigen auszuwählen.

Abbildung 5.1 zeigt ein Beispiel für die Auswahl von Kopf und Händen aus einer Reihe von Kandidaten. Die Komponenten werden durch ihre bounding box und ihren Mittelpunkt repräsentiert.

Ein Mensch kann sich nicht mit beliebig hoher Geschwindigkeit bewegen. Die maximale Distanz, die ein Körperteil von Bild zu Bild zurücklegen kann, ergibt sich aus seiner Höchstgeschwindigkeit und der Framerate (Anzahl der Bilder pro Sekunde) des Bildverarbeitungssystems. Ist die Kopf- bzw. Handposition in einem Bild bekannt, muss man sie im folgenden Bild nur noch innerhalb dieser maximalen Distanz suchen. Für jeden Kandidaten im Suchradius werden drei Werte berechnet:

- Der Kopf- bzw. Handscore,

[4]Nach den Regeln der Normalverteilung ist er in diesem Fall schlechter als der entsprechende Wert von etwa 99% aller Klassenmitglieder.

Abbildung 5.1: Kandidaten (links), ausgewählter Kopf mit Händen (rechts)

- der Quotient aus der Entfernung zum Vorgänger und der maximal möglichen Entfernung (Entfernungsmaß),
- der Quotient aus der Fläche des Vorgängers und der eigenen Fläche (Ähnlichkeitsmaß)[5].

Diese drei Werte werden multipliziert, woraufhin der Kandidat mit dem höchsten Ergebnis als Nachfolger ausgewählt wird.

Kann überhaupt kein Nachfolger für Kopf oder Hand gefunden werden, so ist es zweckmäßig, die alte Position über einige wenige Frames festzuhalten, um kurzzeitige „Aussetzer“ zu überbrücken. Dabei ist zu beachten, dass der Suchradius (und das Tiefenfenster) mit jedem Frame, der seit der letzten Sichtung der Komponente vergangen ist, vergrößert werden muss. Kann endgültig kein Nachfolger mehr gefunden werden, beginnt erneut die Initialisierung.

5.3 Automatische Initialisierung des Hautfarbmodells

Zum Zweck der automatischen Initialisierung des Hautfarbmodells (siehe Kapitel 2) kann auch im Disparitätenbild allein nach einer Kopfregion gesucht werden. Das Disparitätenbild wird hierzu den gleichen Vorverarbeitungsschritten unterzogen wie in Kapitel 4 beschrieben. Danach wird die Komponentenanalyse direkt auf der Tiefenmaske durchgeführt.

Mithilfe des Körpermodells wird eine Komponente gesucht, bei der es sich um die Silhouette einer Person handeln könnte. Dazu wird für alle Komponenten

[5]Es sind auch weiter gehende Ähnlichkeitsmaße denkbar, die z. B. auf einem Vergleich der Form beider Komponenten beruhen.

Disparitätenbild Personenkomponente Kopfkomponente

Abbildung 5.2: Auswahl der Kopfkomponente

analog zum Hand- bzw. Kopf-Score nun der Personen-Score berechnet. Falls eine Komponente c mit $S_{Person}(c) > 0$ existiert, wird ihr oberer Teil (30cm) – also der Teil, in dem im Falle einer Person der Kopf zu vermuten wäre – einer erneuten Komponentenanalyse unterzogen. Resultiert daraus eine Komponente k, bei der es sich um einen Kopf handeln kann ($S_{Kopf}(k) > 0$), wird diese zum Initialisieren des Hautfarbmodells aus Kapitel 2 verwendet.

Abbildung 5.2 zeigt ein Beispiel für die Auswahl einer Person aus dem Disparitätenbild. Die Kopfkomponente wurde nach der Auswahl morphologisch reduziert, um sicherzustellen, dass ausschließlich hautfarbene Bereiche zur Initialisierung verwendet werden.

6 Zusammenfassung

In der vorliegenden Arbeit wurde ein System zum Auffinden und Verfolgen von Gesicht und Händen in Stereo-Videobildsequenzen entworfen. Dazu wurden die Bilder anhand des Merkmals *Hautfarbe* (vorgestellt in [Yang et al. 97]) segmentiert und mit einem an [Darrel et al. 98] angelehnten Verfahren mit dem Disparitätenbild verknüpft. Unter den möglichen Kandidatenregionen für Gesicht und Hände wurden mithilfe eines Körpermodells basierend auf Größe und Flächeninhalt die Wahrscheinlichsten ausgewählt.

Das System baut kein Modell des Bildhintergrundes auf und ist somit auch für bewegte Kameras und dynamische Umgebungen geeignet. Durch die automatische Initialisierung des Hautfarbmodells ist es in der Lage, sich selbstständig an veränderte Lichtverhältnisse anzupassen. Diese beiden Eigenschaften lassen es insbesondere für den Einsatz in mobilen Robotern geeignet erscheinen.

Mit der im Rahmen dieser Arbeit entwickelten Software gelingt die Identifikation und Verfolgung des Kopfes zuverlässig für den Fall, dass sich eine einzelne Person größenteils unverdeckt vor der Kamera befindet. Aufgrund der geringeren Größe[1], der höheren Geschwindigkeit und der höheren Varianz in Form und Größe ist das Verfolgen der Hände deutlich schwieriger und somit nicht frei von Aussetzern und falschen Zuordnungen.

Abhängig von der Art der späteren Anwendung kann es sinnvoll sein, eine spezielle Behandlung von Fällen wie Vereinigung und Aufspaltung von Kopf- und Handregionen zu implementieren, und das Tracking auf mehrere Personen auszudehnen.

Auf einem PC mit einem 1GHz Pentium 3 Prozessor wurde bei einer Auflösung von 320x240 Punkten typischerweise eine Framerate von 12 Bildern pro Sekunde erreicht.

[1] Bei der hier verwendeten Bildgröße von 320x240 Pixeln hat das Abbild einer Hand oft nur eine Fläche von wenigen Pixeln.

A Methoden der Bildverarbeitung

In diesem Anhang werden einzelne Verfahren der Bildverarbeitung näher beschrieben, die im Rahmen der vorliegenden Arbeit Verwendung fanden.

A.1 Morphologische Operatoren

Die morphologische Bildbearbeitung (siehe [Jähne 97]) baut auf zwei Grundoperationen auf: der *Dilatation* und der *Erosion*. Bei der Dilatation wird jeder Bildpunkt eines Graustufen- oder Binärbildes auf den maximalen Wert der Punkte in seiner Nachbarschaft gesetzt, bei der Erosion auf den minimalen. Die Entscheidung, welche Punkte zur Nachbarschaft gehören, wird mithilfe einer binären Maske getroffen – dem sogenannten *Strukturelement*. Das Einheitsstrukturelement ist eine 3x3 Matrix, in der alle Punkte gesetzt sind, was einer sogenannten *8-Nachbarschaft* gleichkommt. Ein Feld des Strukturelements (z. B. der Mittelpunkt) muss als Bezugspunkt ausgezeichnet sein.

Anschaulich wird bei der Dilatation eines Binärbildes das Strukturelement – durch seinen Bezugspunkt zentriert – über jeden gesetzten Punkt gelegt, wodurch in seiner Nachbarschaft neue Punkte entstehen. Umgekehrt bleiben bei der Erosion nur noch jene Bildpunkte übrig, in deren Nachbarschaft das Strukturelement komplett „hineinpasst“.

Oft werden diese beiden Grundoperationen paarweise angewandt. Man spricht dann von morphologischem Öffnen (Erosion mit anschließender Dilatation) bzw. morphologischem Schließen (Dilatation mit anschließender Erosion). Anschaulich entstehen durch das Öffnen Spalten zwischen schwach zusammenhängenden Regionen, umgekehrt werden durch das Schließen solche Regionen verbunden. Durch die Kombination von Dilatation und Erosion bleibt die Fläche der Regionen im Wesentlichen unverändert.

Abbildung A.1 zeigt an einem Beispiel das morphologische Schließen mit einem 2x1 Strukturelement: Ein Loch in der Mitte des Objekts verschwindet.

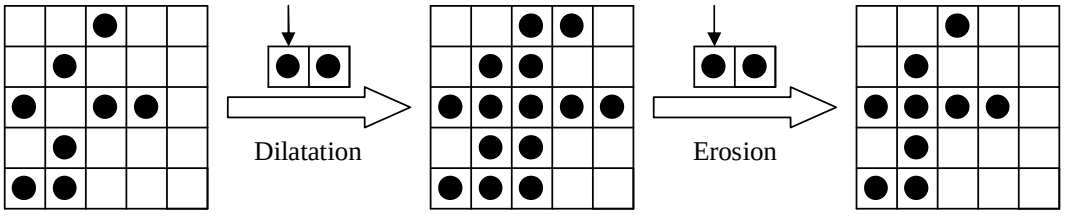


Abbildung A.1: Morphologisches Schließen

In der vorliegenden Arbeit werden morphologische Operatoren an verschiedenen Stellen eingesetzt, so z. B. zum Eliminieren von Störungen in der Hautfarbmaske (Kapitel 2). Es wurde die Implementierung der Open CV Bibliothek verwendet.

A.2 Kantendetektion

Mathematisch gesehen ist eine Kante in einem Graustufenbild eine Unstetigkeit der räumlichen Grauwertfunktion. Zum Finden solcher Kanten wurde in der vorliegenden Arbeit die OpenCV-Bibliothek herangezogen, die eine Implementierung des Verfahrens aus [Canny 86] zur Verfügung stellt. Der Prozess der Kantendetektion nach Canny besteht aus mehreren Schritten:

Zunächst wird das Bild mithilfe einer zweidimensionalen Gaussfunktion geglättet. Die Größe der Standardabweichung ist ein Maß für die gewünschte Feinheit der zu findenden Strukturen. Daraufhin wird ein einfacher zweidimensionaler Operator angewendet, der in jedem Punkt den Gradienten in X- und Y-Richtung berechnet.

Kanten bestehen aus Punkten, in denen der Betrag des Gradienten maximal ist, und die sich entlang von Konturlinien befinden. Es werden daher alle Punkte gelöscht, die in ihrer Umgebung nicht maximal sind. Überprüft wird dies, indem der Wert jedes Punktes mit den Werten seiner Nachbarpunkte in 4 verschiedenen Richtungen verglichen wird.

Um „Aussetzer“ in den so gefundenen Konturlinien zu reduzieren, werden zwei verschiedene Schwellwerte zum Löschen eines Punktes betrachtet, die einen Hysteresebereich bilden. Befindet sich der Wert unterhalb des unteren Wertes, wird der Pixel auf jeden Fall gelöscht. Ebenso wird er auf jeden Fall gesetzt, wenn er sich oberhalb des oberen Wertes befindet. Innerhalb des Hysteresebereiches bleibt ein Punkt gesetzt, wenn er mit einem anderen gesetzten Punkt verbunden ist.

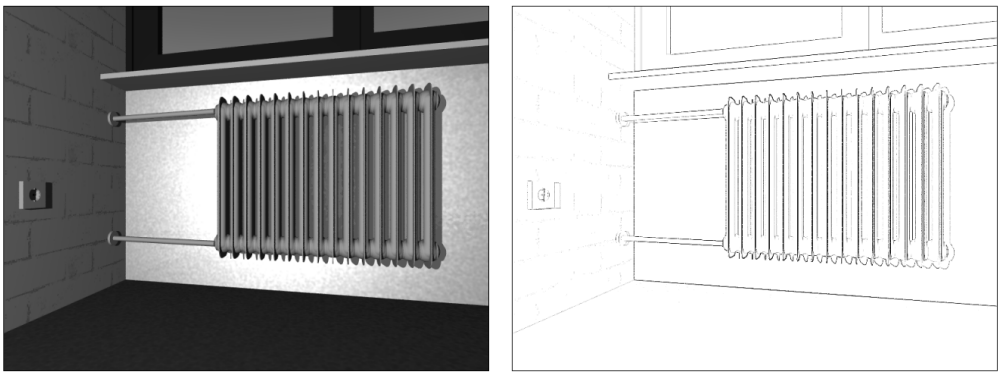


Abbildung A.2: Kantendetektion

A.3 Komponentenanalyse

Aufgabe der Komponentenanalyse ist es, zusammenhängende Regionen in einem Binärbild zu ermitteln und aus diesen Merkmale wie Fläche oder Mittelpunkt zu extrahieren. Zwei Pixel gelten hier als zusammenhängend (8-Zusammenhang), wenn sie so aneinander grenzen, dass sich jeweils einer in einer 3x3-Umgebung um den anderen befindet[1].

Die OpenCV-Bibliothek implementiert einen Algorithmus aus [Suzuki et al. 85], der die äußersten Konturen, die in einem Bild vorkommen, als verkettete Liste von Polygonen liefert. Von der Suche nach leeren Flächen innerhalb geschlossener Flächen wird bewusst abgesehen, weil es sich bei den gesuchten Regionen ausschließlich um geschlossene Flächen handelt, und somit das Schließen von Löchern eine erwünschte Nebenwirkung ist.

Zur Bestimmung der Fläche und des Masseschwerpunkts einer Region werden ihre *Momente* betrachtet. Die Zugehörigkeitsfunktion f einer Region R gibt an, welche Punkte (x, y) Teil von R sind:

$$f(x,y) = \begin{cases} 1, & \text{für} (x,y) \in R \\ 0, & sonst \end{cases} \tag{A.1}$$

Die Definition der Momente $m_{p,q}$ von f lautet

$$m_{p,q} = \int \int x^p y^q f(x,y) dx dy. \tag{A.2}$$

[1] Analog existiert der Begriff des 4-Zusammenhangs, der nur dann erfüllt ist, wenn beide Pixel direkt neben- bzw. übereinander liegen.

Die Fläche A einer der Region, sowie die Koordinaten des Massenschwerpunkts (x_c, y_c) lassen sich wie folgt berechnen:

$$A = m_{0,0} \tag{A.3}$$

$$x_c = \frac{m_{1,0}}{m_{0,0}}, \qquad y_c = \frac{m_{0,1}}{m_{0,0}} \tag{A.4}$$

Zu weiteren Informationen bezüglich der Bildanalyse durch Momente siehe [Jähne 97].

B Aufbau und Funktionsumfang der Software

Das im Rahmen dieser Arbeit entwickelte 3D-Tracking-System besteht aus einer Stereokamera und einem PC, der die Kamerabilder verarbeitet und die Position von Kopf und Händen auf dem Bildschirm visualisiert.

Als Bildquelle kam ein „Mega-D Digital Stereo Head“ der Firma *Videre Design*[1] zum Einsatz, dessen zwei Farbvideokameras über eine gemeinsame IEEE-1394-Schnittstelle (FireWire) mit dem PC verbunden wurden. Der Kameraabstand betrug 9cm. Die Videobilder wurden synchron aufgenommen und haben eine maximale Auflösung von 1288x1032 Pixeln, wurden hier aber aus Geschwindigkeitsgründen auf eine Größe von 320x240 Pixeln reduziert. Bei dieser Auflösung wurde auf einem PC mit 1GHz Pentium 3 Prozessor insgesamt eine Framerate von 12 Bildern pro Sekunde erreicht.

Die Tracking-Software wurde in C++ programmiert und mit Microsoft Visual C++ compiliert. Das Programm wird unter Microsoft Windows ausgeführt. Dank der Verwendung plattformübergreifender Bibliotheken sind bei einer Portierung – insbesondere auf Linux – keine prinzipiellen Schwierigkeiten zu erwarten.

B.1 Externe Bibliotheken

Zur Ansteuerung der Kamera, zur Durchführung von Bildverarbeitungsoperationen und zur Erzeugung der graphischen Benutzeroberfläche werden externe Bibliotheken verwendet.

B.1.1 Small Vision System

Das *SRI Small Vision System (SVS)* [2] ist ein Softwarepaket mit Funktionen zur Ansteuerung von Videokameras (insbesondere auch des oben genannten Mega-D

[1] Videre Design, 865 College Ave, Menlo Park, CA 94025 (http://www.videredesign.com/)
[2] Dieses System wird ebenfalls von Videre Design vertrieben.

Stereokopfes), zur Berechnung von Disparitäten und zur Kalibrierung von Stereokameras.

Die Funktionen zum Einlesen von Videobildern und zur Berechnung von Disparitäten können mittels einer dynamischen Bibliothek (DLL) in eigene Anwendungen eingebunden werden. Für die Kalibrierung der Kamera ist ein separates Programm notwendig.

SVS ist erhältlich für Windows und für Linux.

B.1.2 Open Source Computer Vision Library

Die *Open Source Computer Vision Library (Open CV)*[3] stellt eine große Anzahl von Funktionen zur Verfügung, die bei der Programmierung von Anwendungen aus dem Bereich des Maschinensehens bzw. der visuellen Mensch-Maschine-Interaktion hilfreich sind.

Im vorliegenden Programm finden die folgenden Funktionsgruppen aus Open CV Verwendung: elementare Bildverarbeitungsoperationen, statistische Funktionen, morphologische Operatoren, Kantendetektion, Komponentenanalyse, einfache Zeichenfunktionen.

Open CV basiert auf Intels Image Processing Library (IPL). Beide Bibliotheken sind für Windows und für Linux erhältlich.

B.1.3 Fast Light Toolkit

Das *Fast Light Toolkit (FLTK)*[4] ist eine objektorientierte Bibliothek zur Erstellung graphischer Benutzeroberflächen. Es bietet eine Reihe von Dialogbausteinen wie Fenster, Schaltflächen oder Schieberegler, Operationen zum Darstellen von Rasterbildern und dreidimensionalen OpenGL-Szenen, sowie einen einfachen Callback-Mechanismus zur Ereignisbehandlung.

FLTK ist erhältlich für Windows, Unix und MacOS.

B.2 Module

Abbildung B.1 zeigt schematisch den Aufbau des Tracking-Systems. Es zerfällt in 5 Module, die nacheinander die Einzelbilder der Videosequenz bearbeiten. Gesteuert

[3] Open CV wird bereitgestellt von der Firma Intel (http://www.intel.com/research/mrl/-research/opencv/).
[4] http://www.fltk.org/

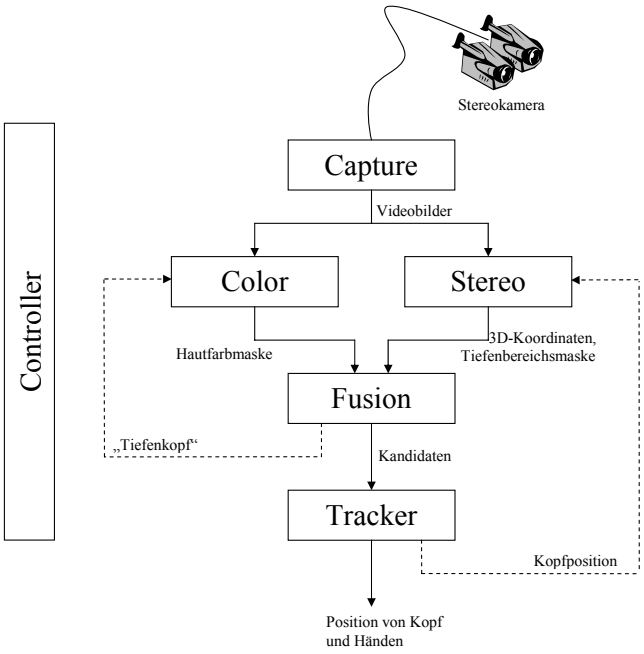


Abbildung B.1: Module

wird der Ablauf vom `Controller`-Modul. Die Module sind jeweils als einzelnes Objekt realisiert und können über eine globale Referenz angesprochen werden.

Jedes Modul verfügt über einen eigenen Bereich innerhalb der graphischen Benutzeroberfläche (GUI), in der seine Parameter verändert und Zwischenergebnisse betrachtet werden können. Alle Module werden von einer gemeinsamen Basisklasse `Module` abgeleitet, die Funktionen bereitstellt, mit der ein Modul auf einfache Weise die für seine Oberfläche erforderlichen Dialogelemente erzeugen kann. Hierdurch wird erreicht, dass der Code der abgeleiteten Module weitgehend frei von GUI-Aufrufen bleibt. Abbildung B.2 zeigt die Oberfläche aller Module.

Im Folgenden werden die einzelnen Module näher beschrieben:

- Der `Controller` übernimmt die globale Ablaufsteuerung des Trackings. Er beinhaltet die Hauptschleife, die auch angehalten werden kann, um manuell in Einzelbildschritten fortzufahren.
- Das Modul `Capture` greift über die SVS-Bibliothek auf die Stereokamera zu und liefert mit jedem Schritt ein neues Bildpaar. Kameraparameter wie Helligkeit und Farbverteilung können hier eingestellt werden. Es ist möglich, Bildsequenzen abzuspeichern, und diese Dateien später als Bildquelle zu benutzen.
- Im Modul `Color` werden das Modell der Hautfarbe aufgebaut und damit die einkommenden Farbbilder klassifiziert. Der Schwellwert der Klassifikation

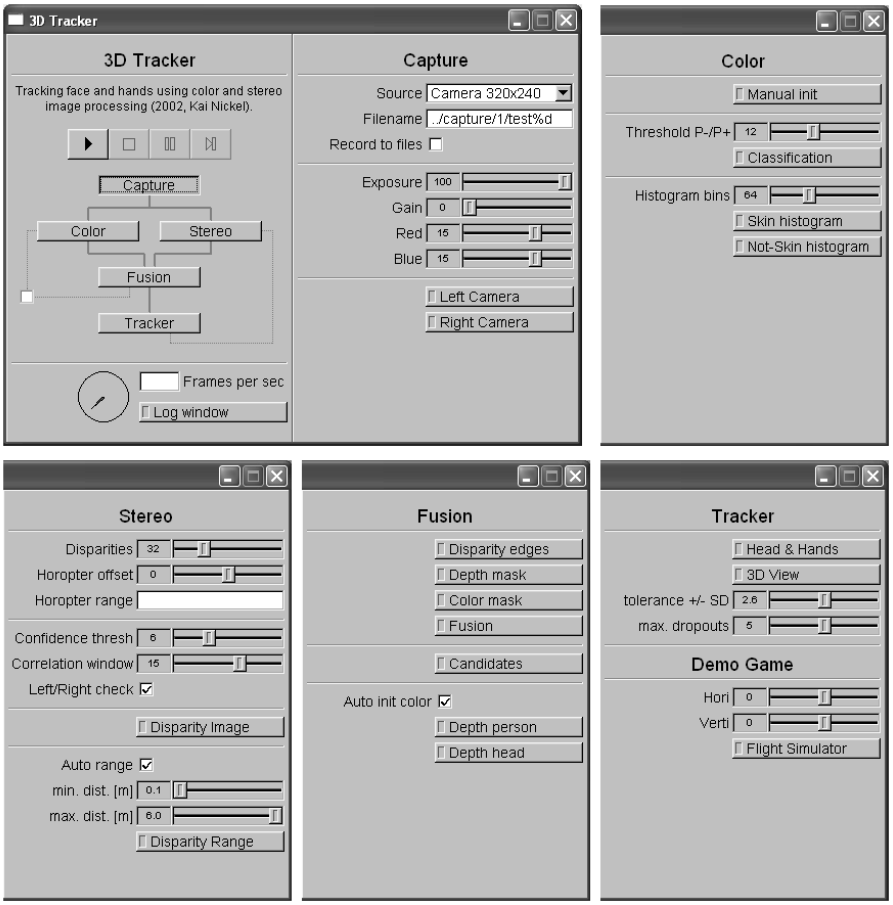


Abbildung B.2: Graphische Benutzeroberfläche

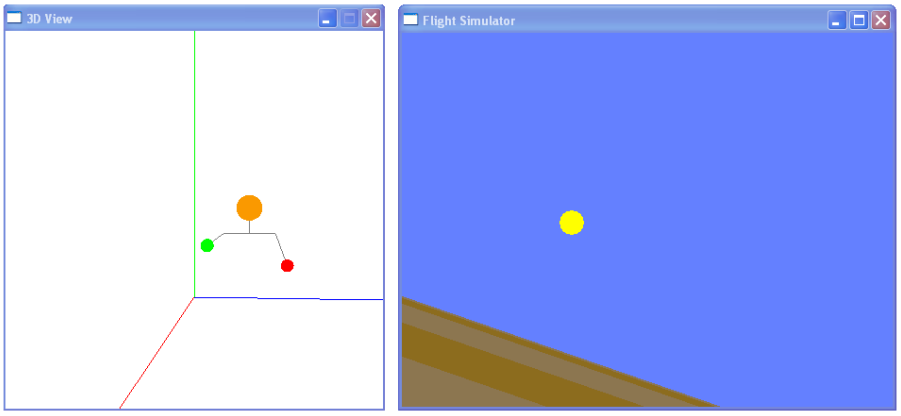


Abbildung B.3: 3D-Anzeige und Flugsimulator-Demo

und die Auflösung der Histogramme können eingestellt, und das Ergebnis der Klassifikation kann in einem separaten Fenster angezeigt werden. Zur Initialisierung des Modells kann von Hand ein rechteckiger Bildbereich markiert werden.

- Im `Stereo`-Modul wird mithilfe der SVS-Bibliothek das Disparitätenbild berechnet. Alle Parameter dieses Vorgangs können eingestellt werden. Ist die Option „Auto Range“ aktiv, wird der Tiefenbereich in Abhängigkeit von der Kopfposition automatisch eingestellt.

- Das Modul `Fusion` vereinigt Tiefenbereichs- und Farbmaske. Mithilfe der Open CV Bibliothek wird die Komponentenanalyse durchgeführt, deren Ergebnis eine Reihe von Kopf-/Hand-Kandidaten ist. Diese werden zur Visualisierung mit ihrer bounding box und ihrem Mittelpunkt über das Videobild gelegt. Ist die Option „Auto init color“ aktiviert, wird kontinuierlich ein Kopfkandidat im Tiefenbild gesucht, um das Hautfarbmodell zu initialisieren[5].

- Der `Tracker` findet in der Liste der Kandidaten Kopf und Hände und verfolgt deren Position von Bild zu Bild. Die Toleranz des Körpermodells und die Anzahl der aufeinander folgenden Frames, in denen Aussetzer erlaubt sind, können eingestellt werden. Neben der Visualisierung von Kopf und Händen als bounding box über dem Videobild gibt es noch die Möglichkeit, die Kopf- und Handpositionen in einem dreidimensionalen OpenGL-Fenster auszugeben.

  Zu Demonstrationszwecken wurde eine einfache Flugsimulation implementiert, die per Handbewegung gesteuert wird: beide Hände werden (wie um einen virtuellen Steuerknüppel oder ein Lenkrad gelegt) nebeneinander zwischen Kamera und Körper gehalten. Durch Drehbewegungen wird das Flugzeug in Schräglage gebracht, durch Entfernen (bzw. Heranziehen) der Hände vom Körper hebt oder senkt sich der Bug des Flugzeuges (Abbildung B.3).

[5] Die automatische Initialisierung verbraucht etwa 25% der Gesamtrechenzeit!

Literaturverzeichnis

[Canny 86] J. Canny (1986). *A Computational Approach to Edge Detection*. IEEE Trans. on Pattern Analysis and Machine Intelligence, S. 679-698.

[Darrel et al. 98] T. Darrell, G. Gordon, M. Harville, J. Woodfill (1998). *Integrated person tracking using stereo, color, and pattern detection*. IEEE Conference on Computer Vision and Pattern Recognition (Santa Barbara, CA), S. 601-608.

[Horn 86] B. Horn (1986). *Robot Vision*. The MIT Press, Cambridge, Massachusetts.

[Jähne 97] B. Jähne (1997). *Digitale Bildverarbeitung*. Springer-Verlag, Berlin-Heidelberg, 4. Ausgabe.

[Konolige 97] K. Konolige (1997). *Small Vision Systems: Hardware and Implementation*. Eighth International Symposium on Robotics Research, Hayama, Japan.

[Konolige et al. 01] K. Konolige, D. Beymer (2001). *SRI Small Vision System – Calibration Supplement to the User's Manual*.

[Rowley et al. 96] H. Rowley, S. Baluja, T. Kanade (1996). *Neural Network-Based Face Detection*. Proc. IEEE Conf. Computer Vision and Pattern Recognition, CVPR-96, S. 203-207, IEEE Computer Society Press.

[Suzuki et al. 85] S. Suzuki, K. Abe (1985). *Topological Structural Analysis of Digital Binary Images by Border Following*. CVGIP, S. 32-46.

[Wren et al. 97] C. Wren, A. Azarbayejani, T.Darrell, A. Pentland (1997). *Pfinder: Real-Time Tracking of the Human Body*. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7.

[Yang et al. 97] J. Yang, W. Lu, A. Waibel (1997). *Skin-color modeling and adaption*. Technical Report of School of Computer Science, CMU, CMU-CS-97-146.