Technische Universität München
Fakultät für Informatik

Diplomarbeit

Entwurf und Integration eines kamerabasierten Trackingsystems für ein Flugzeugcockpit zur Darstellung fortschrittlicher Flugführungsinformationen in einem Head-Mounted Display

Franz Mader

Technische Universität München
Fakultät für Informatik

Diplomarbeit

Entwurf und Integration eines kamerabasierten Trackingsystems für ein Flugzeugcockpit zur Darstellung fortschrittlicher Flugführungsinformationen in einem Head-Mounted Display

Franz Mader

Aufgabenstellerin:	Prof. Gudrun Klinker, Ph.D.
Betreuer:	Dipl.-Inf. Martin Wagner,
	Dipl.-Inf. Hesam Najafi,
	Dipl.-Ing. Florian Holzapfel
Abgabedatum:	15. Juli 2004

Ich versichere, dass ich diese Diplomarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 15. Juli 2004 Franz Mader

Zusammenfassung

Mit der stetigen Zunahme des globalen Luftverkehrsaufkommens in den letzten Jahrzehnten werden pilotenunterstützende Maßnahmen bei der Flugführung immer wichtiger, um einen reibungslosen und sicheren Betrieb zu gewährleisten. Diese Unterstützung erfährt der Pilot primär durch grafische Aufbereitung von Sensordaten zur Darstellung der gegenwärtigen Flugsituation (situational awareness). Hierbei spielen Blickfelddarstellungsgeräte bei der Informationspräsentation in zunehmendem Maße eine Rolle. Der zu beobachtende Trend führt vom Display auf dem Instrumentenbrett über das Head Up Display hin zum Head-Mounted Display, was dem Piloten zusätzliche Freiheiten einräumt. Die zum Einsatz kommenden Geräte zum Tracking der Kopfposition sind hierbei meist proprietäre Lösungen, die auf der Grundlage des elektromagnetischen Trackings basieren.
Im Bereich der Augmented Reality treten sehr ähnliche Fragestellungen in Bezug auf Positionstracking und Augmentationsmöglichkeiten auf, hier jedoch meist in einem viel allgemeineren Kontext. Besonders das optische Tracking findet dabei, aufgrund einiger Vorteile gegenüber anderen Verfahren, häufig Anwendung. Die bekannten Nachteile optischer Systeme, wie z.B. rechenintensive Bildverarbeitung, können jedoch durch die rapide Entwicklung im Hardwaresektor, insbesondere leistungsfähiger Prozessoren und Grafikhardware, sowie durch die Einschränkung eines allgemeinen Ansatzes zugunsten einer „maßgeschneiderten“ Lösung weitgehend kompensiert werden.
So ist es Ziel dieser Arbeit, aus bestehenden Lösungsansätzen des Fachbereiches der Augmented Reality ein Trackingsystem für den Flugsimulator des Lehrstuhls für Flugmechanik und Flugregelung[1] (LFM) der Technischen Universität München zu entwerfen. Der in dieser Arbeit präsentierte Ansatz passt bestehende Techniken allgemeiner Ansätze auf die Trackingumgebung Flugzeugcockpit an, was durch die gegebenen Einschränkungen teilweise zu Vereinfachungen führt, welche bewusst ausgenutzt werden. Andererseits verhindern an mancher Stelle die im Cockpit vorhandenen Restriktionen auch die Verwendung existenter Lösungen. Realisiert wurde ein kamerabasiertes optisches Trackingverfahren, welches auf aktiven Markern im Infrarotbereich beruht. Die Vorgaben der Plattformunabhängigkeit sowie der Adaptionsmöglichkeit auf andere Cockpittypen wurden weitgehend eingehalten. Obwohl Kriterien, wie die Anforderungen zur behördlichen Zulassung in einem Luftfahrzeug, im Rahmen dieser Arbeit keine Berücksichtigung finden konnten, wurde das Systemdesign dennoch bewusst über den reinen Simulatorbetrieb

[1] Fakultät für Maschinenwesen, Institut für Luft-& Raumfahrttechnik, Technischen Universität München

hinaus angelegt, was sich z.B. in der Berücksichtigung stark schwankender Lichtverhältnisse bei der Wahl der Marker widerspiegelt. Die Ergebnisse dieser Diplomarbeit mögen den Leser bei der Wahl eines problemspezifischen Trackingverfahrens unterstützen und mögliche Schwierigkeiten und Lösungsansätze bei der Umsetzung aufzeigen.

Abstract

In light of the continuously increasing air traffic during the last decades, technical aids needed to support the pilot in navigation and flying has become more critical than ever to assure a flawless and safe flight. Primarily, this support is given to the pilot by a graphical presentation of sensor data to depict the actual flight condition (situational awareness). In this concern, visual displays have a crucial role to play nowadays. But the trend leads away from monitors on the centerpannel and head up displays to integrated head-mounted devices, which increase the pilot's freedom of movement. But most of the devices used today are proprietary solutions, with many of them being based on electromagnetical tracking. In the application field of augmented reality, we encounter many of the same questions, but most of them in a more general context. Camera-based optical tracking is frequently applied in this field, due to some major advantages over other tracking methods. The obvious drawbacks of optical tracking, such as highly complex and time-consuming image processing, may be compensated soon with the ongoing development of faster CPUs[2] and GPUs[3]. Moreover, by reducing a „one-fits-all“ solution to a problem-customized approach, several known problems of optical tracking can be avoided or at least reduced. The goal of this thesis is to design a tracking system for the LFM's[4] flight research simulator by using existing techniques from the research field of augmented reality. The approach described here adapts solutions from existing systems to the special requirements of the cockpit environment, which leads in several aspects to constraints that need to be intentionally taken advantage of whenever possible. Otherwise, the cockpit environment introduces several limitations which inhibit the application of existing solutions without modification. Hence, a camera-based optical tracking approach was chosen, based on acitvely emitting fiducials in the infrared spectrum. The requirements of a platform-independent implementation as well as the future option to migrate the system to other cockpit types were complied with as much as possible throughout. Even though the requirements for an approval by federal flight authorities could not be taken into account, the system was nevertheless intentionally designed beyond the exclusive use in the flight simulator, which is reflected particularly in the fiducial design, by assuming a wide range of

[2]Central Processing Unit: Hauptprozessor eines Computers

[3]Graphics Processing Unit: Grafikprozessor, nimmt der CPU Rechenarbeit ab und erhöht somit die Gesamtperformanz des Systems bei grafikintensiven Anwendungen erheblich

[4]Lehrstuhl für Flugmechanik und Flugregelung

lighting conditions in the cockpit. The results presented in this thesis may serve the reader as support in choosing a tracking system tailored to a specific problem, and showing some difficulties and possible approaches to their solution.

Danksagung

Ich möchte mich an dieser Stelle ganz herzlich bei all denjenigen bedanken, die mich bei meiner Arbeit während der letzten sechs Monate in so vielfältiger Weise unterstützt haben. Besonderer Dank gilt Frau Prof. Gudrun Klinker dafür, dass sie mit ihrer Vorlesung „Einführung in die Erweiterte Realität“ mein Interesse an Augmented Reality geweckt hat und sowohl bei meinem Systementwicklungsprojekt als auch bei der Diplomarbeit immer ein offenes Ohr für mich hatte, desweiteren bei meinen Betreuern Martin Wagner, welcher mir besonders in der Endphase mit Rat und Tat zur Seite stand, und Hesam Najafi, der mich in Fragen der Bildverarbeitung und projektiver Geometrie unterstützt hat. Ganz besonderer Dank gebührt Florian Holzapfel, welcher das Projekt von Seiten des Lehrstuhls für Flugmechanik und Flugregelung betreut hat und ohne dessen schnelle und unbürokratische Unterstützung vieles gar nicht erst möglich gewesen wäre. Darüber hinaus möchte ich mich noch bei folgenden Personen bedanken: Prof. Sachs für die Bereitschaft zur Kooperation zwischen seinem Lehrstuhl und dem Fachbereich Augmented Reality, Peter Wanschura für seinen handwerklichen Einsatz beim Bau der Cockpit-Verkleidungen sowie Christian Alt für seine technischen Ratschläge und die Stunden in der Werkstatt, in denen wir die Dioden gelötet haben. Desweiteren möchte ich mich noch bei Georg Mumelter, Sven Kuhlmann, Stefan Myschik, Marius Heise und Navabalachandran Jayabalan für ihre Kollegialität während meiner Zeit am Lehrstuhl bedanken.

Einen besonderen Dank möchte ich zum Abschluss meiner Familie und meinen Freunden für ihre Geduld und die Unterstützung während der gesamten Zeit aussprechen, insbesondere Bianca Künnecke, Alexander Hammer, Daniel deBruycker sowie meiner Schwester Barbara und meinen Eltern Franz und Monika.

Inhaltsverzeichnis

Kapitel 1

Augmented Reality und ihre Anwendungen

Unter dem Begriff Augmented Reality[1] (Erweiterte oder Augmentierte Realität, kurz AR) versteht man die Überlagerung (=Erweiterung) von Information mit der Realität in Echtzeit. Obwohl diese Überlagerung nicht zwingend optisch erfolgen muss, sondern beispielsweise auch akustischer oder haptischer Natur sein kann (z.B. bei Anwendungen für Blinde [27]), bildet die visuelle Überlagerung aufgrund der herausragenden Bedeutung des Sehsinnes für den Menschen das Hauptanwendungsfeld der Augmented Reality. Hierbei soll die Information möglichst am richtigen geometrischen Ort platziert werden. Unter einem AR-System versteht man das System der technischen Bestandteile die nötig sind, um eine AR-Anwendung aufzubauen: Kamera, Trackinggeräte usw. Die Literatur verwendet meist die Definition der Erweiterten Realität von Azuma [11], die

- virtuelle Realität und die Realität kombiniert,
- in Echtzeit interaktiv ist,
- die virtuelle Welt an der Realität ausrichtet.

Eine weitere anschauliche Definition geben Milgram und Kishino [10], indem sie ein Spektrum von der virtuellen Realität (VR, völlig synthetische Sicht) über Mixed Reality (wobei sowohl echte als auch virtuelle Objekte wahrgenommen werden) bis hin zur normalen, unaugmentierten Realität aufspannen.

Beispiele, wo Augmented Reality bereits im Alltag ihre Anwendung findet, sind bei Fussballübertragungen das Einblenden des Abstands zum Tor im Falle eines Freistoßes oder bei Schwimmmeisterschaften das Mitlaufen einer Linie, welche die Weltrekordzeit angibt [43]. Hierbei sei jedoch erwähnt, dass nach der Definition von AR durch Azuma die interaktive Komponente fehlt. Weitere Anwendungsfelder sind:

- Medizinische AR: Einblendung von inneren Organen/Tumoren/Knochen oder Führungshilfen für die Operationsinstrumente [24] [29]

[1] (engl.) to augment: erweitern, vergrößern, steigern

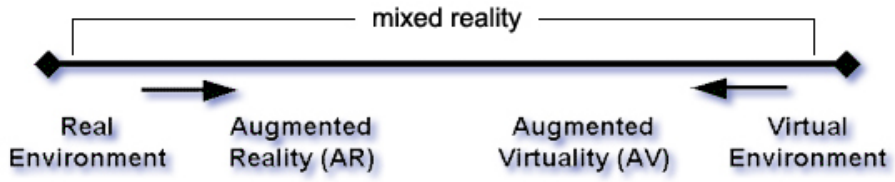


Abbildung 1.1: Reality-Virtuality Kontinuum nach Milgram und Kishino

- Wartungsarbeiten an Maschinen/Flugzeugen/Autos: Darstellung von Arbeitsschritten, markieren von Bauteilen etc. [30]
- Militärische Anwendungen: Navigation in unbekanntem Terrain, interaktive Karten- und Situationsdarstellung [20]
- Architektur/Innenraumgestaltung: Begehen von virtuell eingerichteten Räumen [29]

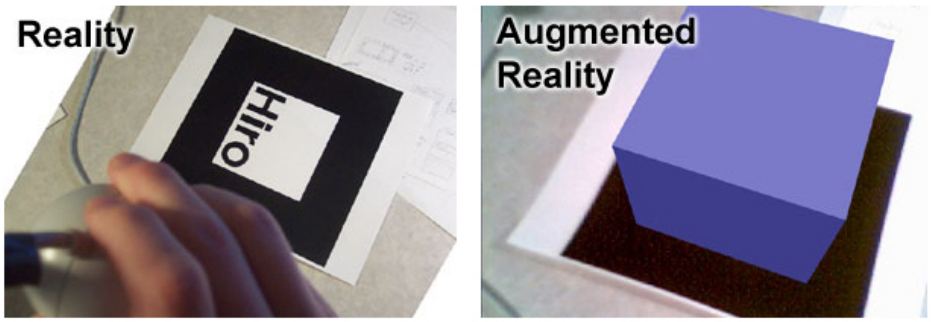


Abbildung 1.2: Reale und augmentierte Szene [40]

So vielfältig die Palette möglicher Anwendungsgebiete auch scheint, so treten doch immer gleichartige Probleme zur technischen Umsetzung auf. Um die Realität auf irgendeine Weise zu „registrieren“, benötigt man stets Information über den gegenwärtigen Standort bzw. die Orientierung/Blickrichtung. Hierfür gibt es verschiedenartige Ansätze, auf welche später in Kapitel 4 im einzelnen eingegangen wird. Desweiteren ist ein je nach Anwendung zum Teil beträchtlich großer Datensatz von Nöten, welcher die Informationen/Objekte enthält, die situationsabhängig dargestellt werden sollen. Hieraus ergeben sich zwei grundlegende Probleme, denen jedes AR-System unterworfen ist.

1. Um der Forderung nach Echtzeit gerecht zu werden, müssen im Idealfall 30 oder mehr Frames[2] pro Sekunde erzeugt werden, um eine flüssige

[2]Ein Frame bezeichnet ein vollständiges Einzelbild in einer Sequenz von Bildern. Ab einer Rate von 25 Bildern pro Sekunde kann das menschliche Auge diese nicht mehr als Einzelbilder wahrnehmen und abgefilmte Bewegungen erscheinen flüssig.

Darstellung der virtuellen Objekte zu gewährleisten. Dies bedeutet im Umkehrschluss, dass zur Berechnung der Lage/Orientierung sowie der Generierung der Augmentierung und deren Ausgabe nicht mehr als 34 ms zur Verfügung stehen.

2. Die Verzögerung, welche von der Signalaufnahme des Sensors bis zur tatsächlichen Darstellung auf dem Ausgabegerät entsteht, sollte möglichst gering gehalten werden, um dadurch auftretende Phänomene wie Schwindel und Übelkeit bis hin zur Desorientierung [24] (unter dem Begriff Cyber-Sickness zusammengefasst) auf ein Minimum zu reduzieren.

1.1 Visualisierung bei (tragbaren) Augmented Reality Systemen

Abhängig vom Kontext gibt es in der Augmented/Virtual Reality eine Vielzahl an Möglichkeiten, wie die Sicht des Anwenders augmentiert werden soll.

- Darstellung auf einer Leinwand bzw. einem Monitor
- Dreidimensionale Darstellung in einer CAVE [43] mit einer „Shutter"-Brille
- Einspiegelung in einen halbdurchlässigen Spiegel [31]
- Darstellung in einem tragbaren Display [24][44][25]

Für unseren Anwendungszweck beschränken wir uns jedoch auf die tragbaren Geräte. Die zwei gebräuchlichsten sollen hier kurz vorgestellt werden.

1.1.1 Video See-Through

Bei der Video See-Through Technik befinden sich unmittelbar vor den Augen des Benutzers zwei kleine Displays, auf welchen ein Videobild dargestellt wird. Eine Kamera, die am Kopf des Trägers befestigt ist und dessen Blickrichtung filmt, liefert das Videosignal, welches mitsamt den augmentierenden Daten in die Brille weitergeleitet wird. Vorteilhaft hierbei ist, dass sowohl virtuelle als auch real (abgefilmte) Objekte die gleiche Auflösung besitzen [29]. Darüber hinaus entsteht bei Kopfbewegungen keine zeitabhängige Verzögerung der virtuellen Objekte, da sie ja für das gerade dargestellte Bild berechnet wurden. Es ist zwar möglich, dass das Gesamtbild mit einer Verzögerung behaftet ist, aber nicht, dass die virtuellen Objekte ihrer wahren Position zeitlich „hinterher" sind. Ein gravierender Nachteil (für die hier angestrebte Anwendung) ist jedoch, dass man bei Ausfall des Systems nur noch zwei schwarze Monitore sieht, d.h. man verliert völlig die Sicht, was insbesondere bei sicherheitskritischen Anwendungen nicht hinnehmbar ist. Ein weiterer Nachteil ist das relativ hohe Gewicht der Brille, was sich insbesondere bei möglicherweise auftretenden Fliehkräften nachteilig auswirkt. Zudem besteht eine geometrische Diskrepanz zwischen dem eigentlichen Augpunkt des Benutzers und der Kamera, welche meist etwas oberhalb der Brille montiert ist [26][29].

Abbildung 1.3: ProView XL50 Video See-Through Head-Mounted Display

1.1.2 Optical See-Through

Bei der Optical See-Through Technik wird das Bild in der Brille des Benutzers transparent dargestellt, bei Ausfall des Geräts bleibt in diesem Fall die unaugmentierte Sicht erhalten. Daher fiel im Rahmen dieser Diplomarbeit die Wahl auf ein Optical See-Through HMD vom Typ SONY Glasstron, welches sich zudem durch sein relativ geringes Gewicht auszeichnet. Nachteilig wirken jedoch die beschränkten Kontrastmöglichkeiten bzw. die Ablesbarkeit in sehr heller Umgebung, was es beispielsweise unmöglich macht, reale Objekte durch virtuelle vollständig zu überzeichen/abzudecken. Zudem kann die heutige Technik noch keine Auflösungen bei der Darstellung liefern, welche an die des menschlichen Auges heranreichen - mit der Konsequenz, dass die augmentierten Objekte stets „pixelig“ wirken und vom Auge klar als künstlich erkannt werden [29].

1.2 Augmentation in der Luftfahrt

Ist bereits im normalen Leben der Sehsinn der wichtigste aller Sinne, so kann dies im Bereich der Luftfahrt nur noch mehr herausgestellt werden. Das Auflösungsvermögen der menschlichen Augen liegt bei ca. 1000 Megapixel. 80% aller Informationen, so haben Studienergebnisse gezeigt, nimmt ein Pilot mit den Augen auf [5]. Hierbei wechselt die Sicht, und damit der Fokus des Auges, stets zwischen dem Blick aus dem Fenster (auf unendlich fokussiert) und dem Blick ins Cockpit auf die (ca. 50-80 cm nahen) Instrumente. Aus dieser Tatsache resultiert das Bestreben, wichtige Informationen direkt (auf unendlich fokussiert) in die (Fern-)Sicht des Piloten zu integrieren. Eine der

Abbildung 1.4: HMD der SONY Glasstron Familie

ersten Anwendungen war es bereits in den 40er Jahren, das Fadenkreuz der Bordgeschütze dem Piloten in ein so genanntes Reflex-Visier, eine schräg montierte Glasscheibe direkt über dem Instrumentenbrett, einzuspiegeln [30].

Abbildung 1.5: Reflex-Visier einer Messerschmitt Bf-109 G2

1.2.1 Head Up Display (HUD)

Die Weiterentwicklung des Reflex-Visiers stellt das Head Up Display dar, in welches nun, wie bei einem Display, dynamisch Information eingespiegelt werden kann [31]. Heutige Kampfflugzeuge sind mit Weitwinkel-HUDs ausgestattet, die bis zu 25° x 30° des Blickfelds abdecken [5]. Sie sind in der Lage, eine Vielzahl von relevanten Informationen gleichzeitig darzustellen, wie z.B.:

- Fluggeschwindigkeit
- Flughöhe
- Kurs
- Anstellwinkel
- Steig-/Rollwinkel
- Funkfrequenzen
- Waffensystemspezifische Informationen
- Wegpunktinformationen (Richtung, Entfernung, geschätzte Ankunftszeit etc.)

Abbildung 1.6: Head Up Display einer Lockheed Martin F-16 C

Zusätzlich variiert die dargestellte Information abhängig vom gewählten Modus, in dem sich das System gerade befindet [32]. So gibt es auf verschiedene Situationen zugeschnittene Modi, welche neben den Grundanzeigen, die in jedem Modus vorhanden sind, die gerade benötigten Daten darstellen. Solche Modi sind zum Beispiel:

- Navigationsmodus
- Luft-Luft Modus (für den Entfernungsluftkampf ausgelegt)
- Dogfight Modus (für den Kurvennahkampf ausgelegt)
- Luft-Boden Modus (für die Bekämpfung von Bodenzielen)
- ILS Modus (für den Landeanflug)

Nach langjähriger Weiterentwicklung und erfolgreichem Einsatz von Head Up Displays bei Kampfflugzeugen kann man nun zunehmend deren Einzug in die Cockpits von Transportflugzeugen und zivilen Strahlflugzeugen beobachten. So bieten heutzutage die meisten Hersteller von Airlinern und Business-Jets optional die Ausstattung ihrer Flugzeuge mit Head Up Displays an.

Abbildung 1.7: Ziviles Head Up Display einer Boeing 737-832; Foto: (c) Michael Ciasullo

1.2.2 Helmet-Mounted Display

Die Weiterentwicklung des Head Up Displays stellt das Head-Mounted Display dar. Dieses ermöglicht es, dem Piloten auch beim Blick zur Seite bzw. nach oben Informationen in sein Sichtfeld einzuspiegeln. Wie bereits beim Head Up Display kommt auch hier der militärischen Anwendung eine Vorreiterrolle zuteil. Insbesondere im Luftkampf spielt die Erfassung und Verfolgung von Zielen mit einem großen „off-boresight"-Winkel (der Winkel zwischen der Längsachse des Flugzeugs und dem Ziel) eine entscheidende Rolle. So ist das Cockpit von Amerikas neuestem Kampfflugzeug, der F-35 (Joint Strike Fighter), bereits ausschließlich auf ein HMD ausgelegt und verfügt zudem über kein HUD mehr.

Im zivilen Bereich ist die Motivation für den Übergang vom HUD auf ein Head-Mounted Display natürlich durch andere Problemstellungen begründet. Die aufmerksamkeitsintensiven Phasen des Fluges spielen sich hier fast ausschließlich beim Abflug sowie beim Landeanflug und der Landung ab.

Sind heutzutage noch viele Endanflüge auf Flughäfen sogenannte „Straight-in approaches“, also Anflüge, die bereits in einiger Entfernung zur Landebahn auf diese ausgerichtet sind, so werden zukünftig komplexere An- und Abflugverfahren in ihrer Anzahl zunehmen. Dies liegt zum einen in dem stetig wachsenden Luftverkehrsaufkommen begründet, zum anderen aber in den zunehmend strengeren Lärmschutzanforderungen, welche zu Änderungen der An- und Abflugschneisen führen. So wird im Bereich der synthetischen Sichtaufbereitung aktuell ein starker Forschungsaufwand betrieben, unter anderem durch die NASA[3][22], das DLR[4][23] sowie universitäre Forschung wie hier am Lehrstuhl für Flugmechanik und Flugregelung.

Aufgrund dieses Trends ist mit zunehmender Miniaturisierung und sinkenden Hardwarekosten die Übernahme der HMD-Technologie in angepasster Form in den zivilen Flugzeugmarkt absehbar.

Abbildung 1.8: Joint Helmet-Mounted Cueing System, US Air Force

[3]National Aeronautics and Space Administration
[4]Deutsches Zentrum für Luft- und Raumfahrt

Kapitel 2

Der Flugsimulator am LFM

Der Lehrstuhl für Flugmechanik und Flugregelung (LFM) der Fakultät Maschinenwesen an der Technischen Universität München betreibt seit 2002 einen neuen Forschungsflugsimulator. Dieser besteht aus einem Cockpit, welches den Stand zukünftiger Airliner widerspiegelt und aufgrund seiner generischen Auslegung verschiedene Flugzeugtypen simulieren kann. Es verfügt über 4 große LCD Multifunktionsdisplays sowie einen Overhead-Touchscreen und ist in dem Mock-Up einer Fairchild Dornier 728 Jet untergebracht. Die Außensicht wird über 3 CRT-Projektoren auf eine gekrümmte Leinwand projiziert, die ein Blickfeld von 150° horizontal und 45° vertikal abdeckt. Als primäre Steuerorgane dienen zwei FAA[1] zugelassene Sidesticks, die Schubregelungseinheit eines Airbus A320 sowie die Ruderpedale des Eurofighter-Simulators [28].

Abbildung 2.1: Flugsimulator des LFM

[1] FAA: Federal Aviation Administration, US-amerikanische Luftfahrtzulassungsbehörde

2.1 Fortschrittliche Flugführungsinformationsdarstellung (Tunnel, Predictor)

Gegenstand gegenwärtiger Forschungsarbeit ist unter anderem das Projekt „Highway in the Sky“. Hierbei handelt es sich um die Darstellung eines Tunnels im Sichtfeld des Piloten, welcher den geplanten idealen Flugpfad (insbesondere beim Landeanflug) projiziert und mit dem Landschaftsbild überblendet. Dadurch erhält der Pilot eine anschauliche Darstellung des Flugwegs sowie seiner relativen Position dazu, insbesondere bei unübersichtlichem Terrain oder schlechten Sichtverhältnissen [28]. Hält der Pilot nun seine Maschine innerhalb des Tunnels, so befindet er sich auf dem vorhergesehenen Kurs. Unterstützung erhält er dabei durch den sogenannten Predictor, ein Symbol, welches die Position des Flugzeugs in 5 Sekunden vorhersagt. Diese Vorhersage wird aus den gegenwärtigen Flugdaten (Lage, Beschleunigung) sowie den Steuereingaben des Piloten getroffen. Befindet sich die Maschine nun im Landeanflug, so läuft innerhalb des Tunnels ebenfalls ein Rahmen mit, welcher die optimale Flugposition in 5 Sekunden repräsentiert. Sollte sich das Flugzeug nun außerhalb des Tunnels befinden, so muss der Pilot das Flugzeug so steuern, dass der Predictor mit dem Vorhersagerahmen zur Deckung kommt, und die Maschine „findet“ ihren Weg auf den Anflugpfad zurück. Durch die Darstellung von Tunnel und Predictor wird das präzise Abfliegen komplizierter Verfahren deutlich vereinfacht und auch bei Nullsicht ermöglicht. Hierbei reduziert sich die Abweichung von der Sollflugbahn von einigen hundert Metern, was den Toleranzen bei herkömmlichen Verfahren entspricht, auf einige wenige Meter.

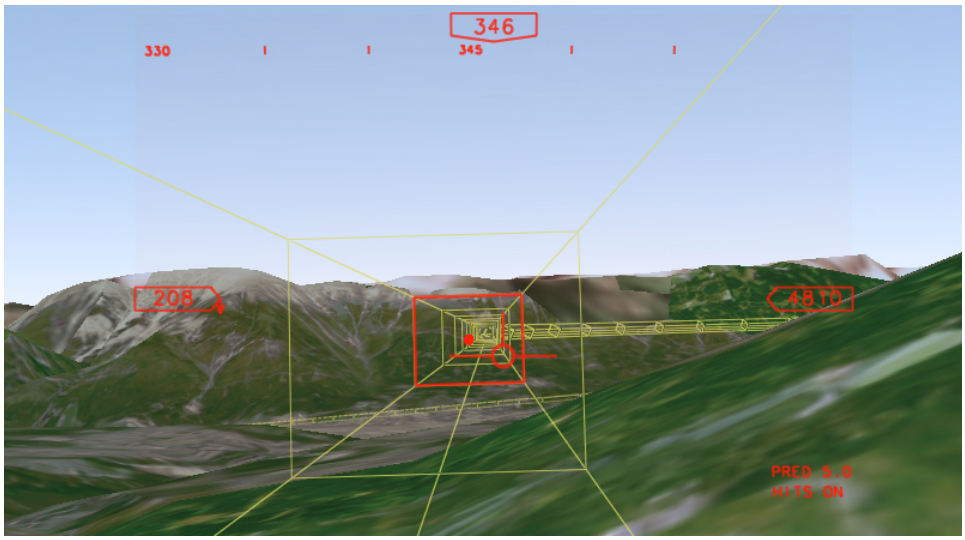


Abbildung 2.2: Flugführungsdarstellung der Zukunft: Tunnel und Predictor

2.2 Integration eines Headtracking Systems zum Übergang auf ein HMD

Bisher erfolgt die Anzeige des Tunnels und der HUD-ähnlichen Darstellung weiterer Flugführungsinformationen durch schlichte Überlagerung über die Landschaftsprojektion der Außensicht des Flugsimulators auf die Leinwand. Der Lehrstuhl für Flugmechanik und Flugregelung besitzt jedoch auch ein eigenes Forschungsflugzeug, in welchem die Tunneldarstellung bereits mit echten Flugdaten verifiziert werden konnte. Hier war die Anzeige jedoch auf ein im Cockpit montiertes LCD Display beschränkt. Um in der realen Anwendung die Vorteile der Darstellung von Tunnel und Predictor voll auszunutzen, bedarf es einer Möglichkeit, direkt in das Sichtfeld des Piloten zu augmentieren. Da in der zivilen Luftfahrt - im Gegensatz zum Militär - jedoch davon ausgegangen werden kann, dass der Pilot keinen Helm trägt, bieten sich kommerziell erhältliche Head-Mounted Displays als kostengünstige Alternative an.

Abbildung 2.3: LFM Flugsimulator: Projektion des Tunnels mitsamt der Außensicht

Als Integrationsumgebung wird vorerst der lehrstuhleigene Simulator benutzt, wobei die Möglichkeit der späteren Adaption auf andere Cockpittypen stets im Auge behalten wird. Dieser Tatsache wird in der Analyse des Cockpits als Trackingumgebung im folgenden Kapitel Rechnung getragen. Ziel ist es, das System nach Validierung im Simulator in das lehrstuhleigene Ultraleicht-Forschungsflugzeug zu übertragen.

Kapitel 3

Das Flugzeugcockpit als Trackingumgebung

Bei der Betrachtung von Flugzeugcockpits lassen sich vom kleinen Sportflugzeug bis hin zur Boeing 747 einige Gemeinsamkeiten feststellen. Die meisten Flugzeugcockpits verfügen über zwei nebeneinander angeordnete Sitze. Vor ihnen (manchmal auch seitlich angeordnet), befindet sich das Steuerhorn bzw. der Steuerknüppel (bzw. Joystick oder Sidestick), auf dem Boden die Ruderpedale für die Seitenruder. In Armreichweite vor dem Piloten ist das Instrumentenbrett angebracht, das je nach Alter, Ausstattung und Typ der Maschine über analoge Anzeigegeräte bzw. Displays verfügt. Oberhalb davon befindet sich die Frontscheibe, welche eine Sicht nach draußen ermöglicht.

Abbildung 3.1: Leichtes, 2-sitziges Motorflugzeug Diamond Aircraft DA-20 Katana

Meist zentral zwischen den beiden Sitzen, bei kleineren Maschinen auch

Abbildung 3.2: Vierstrahliges Langstreckenverkehrsflugzeug Airbus A340

direkt am Instrumentenbrett, befinden sich die Schubregler. An den Außenseiten der Sitze findet man, insbesondere bei größeren Maschinen, weitere Konsolen, auf denen zusätzliche Instrumente bzw. Anzeigen angebracht sind. Schräg oberhalb des Piloten befindet sich je nach Flugzeugtyp eine Overheadkonsole (z.B. bei Linienmaschinen), der Tragflügel (bei Hochdeckern) oder freie Sicht durch die Cockpitkanzel (bei Tiefdeckern und Segelflugzeugen). Die meisten Piloten sind i.A. während des gesamten Fluges angeschnallt, was ihren Bewegungsspielraum relativ stark eingrenzt. So kann man klare Vorhersagen über die Kopfposition des Piloten machen, die sich auf einen Bereich von ca. 80cm x 50cm x 20cm (Breite x Tiefe x Höhe) einschränken lässt. Beobachtungen am Lehrstuhl haben gezeigt, dass diese Werte bei aufmerksamkeitsintensiven Flugphasen wie dem Landeanflug noch deutlich unterschritten werden.

3.1 Anforderungen an ein Trackingsystem im Flugzeugcockpit

Im Folgenden soll beschrieben werden, welche allgemeinen Rahmenbedingungen für eine Trackinganwendung in einem Flugzeugcockpit erforderlich sind. Es sei hiermit ausdrücklich darauf hingewiesen, dass diese Gesichtspunkte keinesfalls den Erfordernissen zur behördlichen Zulassung für den Luftverkehr genügen, dies würde auch beiweitem den Rahmen dieser Arbeit sprengen. Vielmehr geht es darum, von einem allgemeinen Ansatz hin zu einer für diese spezielle Anwendung maßgeschneiderten Lösung zu kom-

men. Ungeachtet dessen sollen für den Betrieb im Flugsimulator realistische Bedingungen als grundlegende Annahmen für die folgenden Ausführungen dienen.

3.1.1 Allgemeine Anforderungen

Das Gesamtsystem sollte, besonders im Hinblick auf einen möglichen Einsatz im Ultraleicht-Flugzeug, über ein geringes Gewicht verfügen. Alle zum Tracking notwendigen Installationen müssen unempfindlich gegenüber Vibrationen und allen im Cockpit auftretenden Lasten sein. Das System sollte außerdem immun gegenüber schwankenden Licht- und Schattenverhältnissen sein. Beim Einsatz irgendeiner Form von Markern ist darauf zu achten, dass diese weder Instrumente oder die Außensicht verdecken, noch dass sie durch ihre Gestalt den Piloten ablenken oder blenden. Schließlich dürfen keinerlei elektromagnetische Interferenzen mit der Flugzeugavionik auftreten oder sonstige Strahlungen, welche eine Gefahr für Mensch und Maschine darstellen oder den sicheren Betrieb des Luftfahrzeugs beeinträchtigen würden.

3.1.2 Beeinträchtigung der Sichtfreiheit

Eine der wichtigsten Anforderungen an das gesamte Trackingsystem ist die Aufrechterhaltung der uneingeschränkten Sichtfreiheit des Piloten. Dieses stellt eine absolut sicherheitsrelevante Forderung dar. Die Gewährleistung der Sicht, auch im Falle eines Systemausfalls, ist eines der stärksten Argumente gegen die Benutzung eines Video See-Through HMDs. Würde im Endanflug das Sichtsystem ausfallen, hätte der Pilot keine Möglichkeit mehr, in der verbleibenden Zeit das HMD abzunehmen, beziehungsweise wäre es völlig unmöglich, mit inoperablem HMD zu landen. Daher fällt die Wahl bei unserem System auf ein Optical See-Through Display, da hier lediglich die Augmentierung verloren ginge, die Sicht aus dem Fenster und auf die Instrumente jedoch erhalten bliebe. Die Forderung nach Aufrechterhaltung der Sichtfreiheit beschränkt sich jedoch nicht nur auf die Wahl der HMD-Technik, auch der Blick auf die Fluginstrumente darf durch keinerlei Installation für das Headtracking verdeckt oder beeinträchtigt werden. Dies wird insbesondere bei der Wahl der Marker noch eine Rolle spielen.

3.1.3 Platzierung von künstlichen Markern

Beschränkt man sich bei der Betrachtung der in Frage kommenden Trackingmethoden auf markerbasiertes Tracking, da markerlose Ansätze beim heutigen Stand der Technik noch nicht die erforderliche Performanz erreichen [13], so benötigt man zur Positionsbestimmung irgendeine Form künstlicher Marker, welche durch ihre wahrgenommene Lage Aufschluss über die aktuelle Position geben. Aufgrund der Forderung nach uneingeschränkter Sichtfreiheit reduziert sich die Wahl der Platzierungsmöglichkeiten vollständig um den Bereich der Fenster. Mögliche Orte zur Anbringung der Markierungen sind demzufolge das Instrumentenbrett, der Bereich über und hinter dem Kopf des Piloten - soweit verfügbar -, sowie die Verstrebungen zwischen den

Scheiben. An dieser Stelle wollen wir uns jedoch an die verschiedenen Cockpitkonfigurationen (siehe Kap. 3) erinnern, um einen möglichst portablen Ansatz zu wählen. Da bei Tiefdeckern z.T. eine strebenfreie Kanzel den Piloten umgibt, fällt der Bereich oberhalb des Kopfes als Platzierungsort für Marker weg. Auch der Freiraum hinter dem Pilotensitz, welcher in Airliner Cockpits recht großzügig bemessen ist, beschränkt sich bei anderen Flugzeugtypen auf ein Minimum, so dass auch hier keine gemeinsame Grundlage für ähnliche Rahmenbedingungen zur Markerplatzierung besteht. So bleibt das Instrumentenbrett als einzige Konstante über alle Flugzeugtypen, wobei dieses sich immer durch seine Planarität und dem rechten Winkel zur Blickrichtung des Piloten auszeichnet. Die erste Wahl zur Platzierung der Marker fällt folglich, um den obigen Forderungen genüge zu leisten, auf die Freiräume des Instrumentenbretts. Dies wiederum setzt eine relativ kleine Markergröße voraus.

Abbildung 3.3: Das Instrumentenbrett des LFM-Flugsimulators

3.1.4 Robustheit gegenüber Verdeckung

Auf dem Instrumentenbrett eines Flugzeugs befinden sich neben reinen Anzeigegeräten auch Instrumente, an welchen der Pilot während des Flugs Einstellungen vornehmen muss (Autopilot, Funk, Fahrwerkshebel) oder bestimmte Tasten bedient, um Zugriff auf die benötigte Information zu erlangen (Multifunktions-Bildschirme). Daher muss bei der Auslegung des Trackingsystems davon ausgegangen werden, dass zeitweise einige Bereiche des Instrumentenbretts durch die Hände/Arme der Piloten verdeckt werden. Es ist also notwendig, dass auch unter solchen Umständen das System noch einwandfrei funktioniert.

3.1.5 Anforderungen an die Genauigkeit

Um eine ortsgenaue Überblendung der im HMD dargestellten Flugführungsinformationen mit der Umwelt zu erreichen, müssen die vom Trackingsystem gelieferten Daten gewissen Genauigkeitsanforderungen entsprechen. Hierbei spielt die translatorische Genauigkeit eine eher untergeordnete Rolle, da die virtuelle Entfernung der projizierten Daten (10m - 1000m+) wie der Tunnel oder Predictor um ein Vielfaches größer ist als der mögliche Offset bei translatorischer Bewegung des Kopfes ($<$ 100cm). Von größter Bedeutung ist demgegenüber die Winkelgenauigkeit, da diese starken Einfluß auf die Deckungsgenauigkeit von virtuellen und realen Objekten hat[1]. Bei einer Projektion des Flugvektors in einer Entfernung von einem Kilometer entspricht eine Winkelabweichung von einem Grad radial einer Missweisung des Vektors von ca. 17m.

$$\text{Abweichung von } 1^\circ = \frac{(2 * 1000m * \pi)}{360^\circ}$$

Der Flugführungstunnel hat eine Ausdehnung von 50m in der Breite. Der Predictor und der voranlaufende Rahmen basieren auf einer Vorhersage der Flugzeugposition in fünf Sekunden. Der Abstand ist somit von der Fluggeschwindigkeit abhängig. Setzten wir eine realistische Anflugsgeschwindigkeit von 160kts[2] voraus, so resultiert daraus ein Prediktionsabstand von 411,2m. Auf diese Entfernung entspricht eine Missweisung von 1° einem Offset von ca. 7,1m. Bei einer Tunnelbreite von 50m liegt dieser Fehler im Toleranzbereich. Somit postulieren wir eine Trackinggenauigkeit der Rotationswinkel von unter einem Grad als Mindestanforderung.

3.2 Annahmen über Einschränkungen eines Trackingsystems im Flugzeugcockpit

Wie bereits in der Einleitung erwähnt, soll sich der in dieser Arbeit verfolgte Ansatz die Vorteile der Trackingumgebung Cockpit zu Nutze machen. Zeichnen sich viele Augmented-Reality Projekte durch die Mobilität des Benutzers aus, so können in unserem Fall ganz klare Aussagen über die (begrenzte) Bewegungsfreiheit des Piloten getroffen werden. So wird im weiteren davon ausgegangen, dass der Pilot seinen Kopf nur innerhalb eines spezifizierten Bereiches bewegt (Kap. 3). Diese Annahme wird durch Beobachtungen von Piloten des Simulators am LFM gerechtfertigt. Zudem ist nur dann eine Augmentierung der Sicht erforderlich, wenn die Kopfposition darauf schließen lässt, dass der Pilot aus dem Fenster blickt. Diese Einschränkung wird hauptsächlich die Platzierung der Marker beeinflussen. Die Begründung dieser Einschränkung liegt darin, dass der Pilot trotz HMD dennoch seine

[1] Folgendes Experiment möge diese Tatsache verdeutlichen. Betrachtet man die Sonne (sehr weit entferntes Objekt) durch einen Dia-Rahmen unter der gleichen Winkelausrichtung (z.B. mit einer Blickrichtung genau nach Süden (180°)), so spielt es keine Rolle, ob man sich 100m weiter links oder rechts befindet. Dreht man sich jedoch um nur 1° nach links oder rechts, so wandert die Sonne durch den gesamten Rahmen.

[2] 160 Knoten entsprechen ca. 296km/h respektive 82,24m/s.

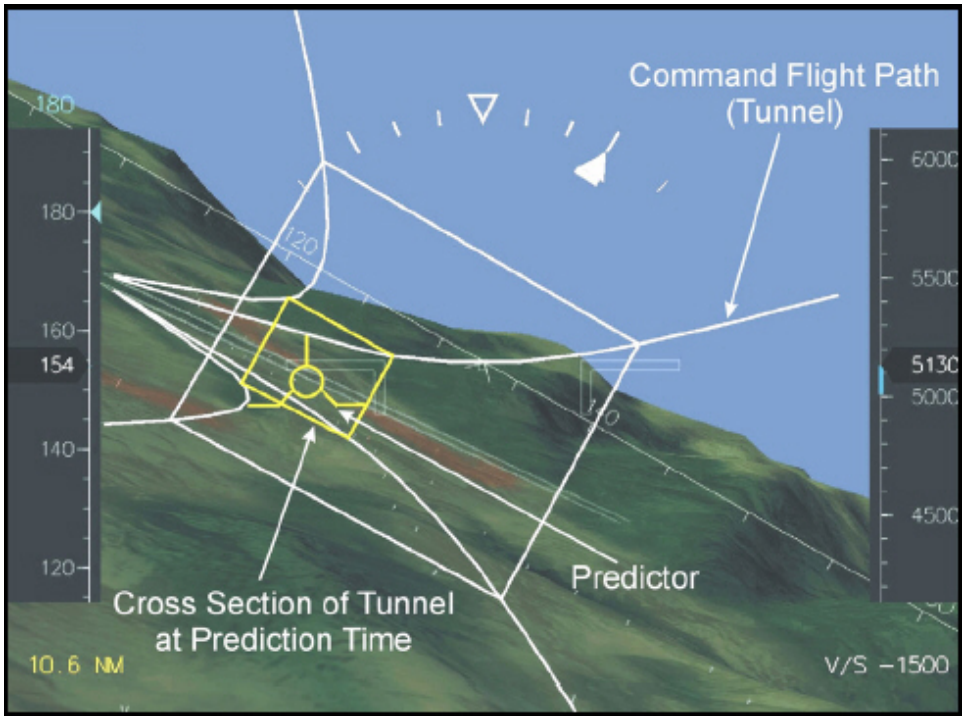


Abbildung 3.4: Predictor im Anflugstunnel

Instrumente und Anzeigen auf dem Instrumentenbrett sowie der Overhead-Konsole ablesen können muss. Daher soll eine Augmentierung bewußt nur im Bereich der Fenster erfolgen.

Eine weitere Vereinfachung ergibt sich aus der Tatsache, dass es sich bei einem Cockpit um eine weitestgehend statische Umgebung handelt. Demzufolge sind außer der zeitweisen Verdeckung von Teilen des Instrumentenbretts durch den Pilot/Copilot mit keinen weiteren Verschattungen innerhalb der Trackingumgebung zu rechnen. Da es sich zudem nur um ein zu trackendes Objekt - den Kopf des Piloten - handelt, kann es auch nicht zu einer gegenseitigen Verdeckung mehrerer zu trackender Objekte kommen.

Kapitel 4

Tracking

Für jede Augmented Reality ist es unerlässlich, Informationen über die Position und Orientierung der Objekte sowie des Betrachters in der Szene zu haben. Diese gewinnt man durch Tracking. Da es sich, wie wir in Abschnitt 3.2 gesehen haben, beim Flugzeugcockpit um eine statische Umgebung handelt, ist für unsere Anwendung lediglich die Position und Orientierung des Pilotenkopfes von Interesse. Da viele Augmented Reality Anwendungen mit mobilen Benutzern zu tun haben, und dadurch auch „globale“ Aufenthaltsinformationen eine Rolle spielen, sind in solchen Anwendungen auch grobe Ortsbestimmungen (in ihrer Positionsgenauigkeit im 10m-1m Bereich) erforderlich. Solche können beispielsweise über satellitengestützte Navigationssysteme wie das U.S.-amerikanische Global Positioning System (GPS/DGPS)[1] oder das europäische GALILEO[2] gewonnen werden. Da sich unsere Anwendung jedoch auf einen sehr begrenzten Raum beschränkt, soll die Betrachtung solcher Trackingmethoden in der folgenden Kurzübersicht ausgeklammert werden. Generell spielen natürlich bei der Auswahl eines jedes Trackingsystems neben den technischen Spezifikationen wie

- Präzision,
- Reichweite/erfasster Bewegungsraum,
- Störanfälligkeit,
- Trägheit und
- Drift

auch weitere, nichtfunktionale Parameter eine Rolle. Solche sind u.a.:

- Gewicht

[1] GPS: Globale Positioning System, ein vom US-Militär betriebenes Satellitennavigationssystem mit 24 aktiven Satelliten in 4 Umlaufbahnen, dessen Signal auch zivilen Anwendern bereitgestellt wird.
DGPS: Differential GPS: Durch Zuhilfenahme einer geodätisch exakt vermessenen Bodenstation in der Nähe des Aufenthaltsortes des Empfängers kann die Positionsbestimmung weiter verbessert werden.

[2] GALILEO: europäisches Gegenstück zum GPS, modernere Technik, ausschließlich für zivile Anwendungen

- Energieverbrauch
- Verträglichkeit für die Umwelt (z.B. elektromagnetische Strahlungsintensität)
- Kosten

4.1 Elektromagnetisches Tracking

Das Prinzip des (aktiven) elektromagnetischen Trackings ist folgendes [26]: In dem Raum, in dem getrackt werden soll, wird ein elektromagnetisches Feld erzeugt. Das getrackte Objekt ist mit elektromagnetischen Sensoren versehen, welche aus der Messung der lokalen Magnetfeldeigenschaften auf die Position zurückschließen können. Zu den Vorteilen elektromagnetischer Tracker gehören ihre Genauigkeit, welche im mm-Bereich bezüglich der Position und im zehntel°-Bereich bezüglich der Orientierung liegen, ihre Unempfindlichkeit gegenüber optischen Einflüssen (vorherrschende Helligkeits-/Kontrastverhältnisse) sowie die relativ geringen Systemkosten[42] [46]. Elektromagnetische Trackingsysteme des abgebildeten Typs werden beispielsweise von der U.S. Air Force für Flugsimulatoren eingesetzt[41]. Nachteile sind die Ortsgebundenheit solcher Systeme (aufgrund der Reichweite, welche i.A. unter 10m liegt), die nicht homogen verteilte Präzision (abhängig von der Entfernung zum Sensor) sowie die extreme Störanfälligkeit in Bezug auf ferromagnetische Stoffe.

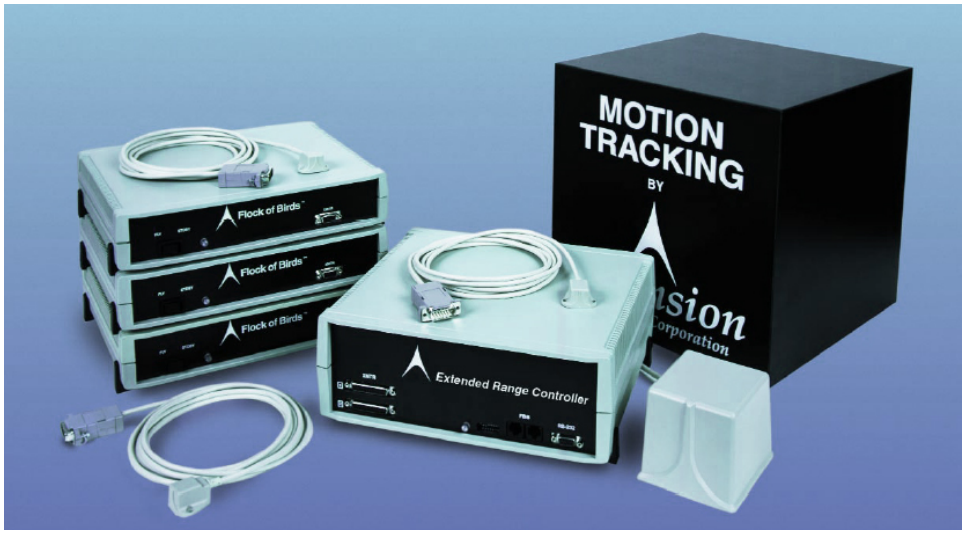


Abbildung 4.1: Ascension Technology Corp. - Flock of Birds, magnetisches Trackingsystem

4.2 Mechanisches Tracking

Eine andere Möglichkeit der Ortsbestimmung ist das mechanische Tracking. Hierbei besteht, wie der Name bereits andeutet, eine mechanische Verbindung zwischen dem zu trackenden Objekt und einem Referenzpunkt. Diese Art der Positionsbestimmung findet hauptsächlich ihre Anwendung in Verbindung mit robotergestützter Arbeit, wo bereits ab inito eine mechanische Verbindung besteht. Sie liefert eine hohe Präzision der Positions- und Orientierungsdaten. In fast allen anderen Bereichen, in denen keine direkte Kopplung zwischen den beweglichen Objekten und den Fixpunkten besteht, ist diese Art des Trackings jedoch ungeeignet.

Abbildung 4.2: Handgesteuertes mechanisches Tracking

4.3 Inertiales Tracking

Inertiales Tracking macht sich das Prinzip der Trägheitsnavigation zunutze, dass durch Messung der Beschleunigungen in einem orthogonalen System und zweifache Integration darüber den zurückgelegten Weg in Richtung der Referenzachsen bestimmt. Somit kann nach vorhergehender Initialisierung des Systems eine Aussage über den gegenwärtigen Standort und die Orientierung getroffen werden. Inertiale Tracker sind nach der Initialisierung völlig autonom von der Außenwelt, was beispielsweise ihren jahrzehntelangen Einsatz in Flugzeugen (insbesondere bei Transatlantikflügen weitab von Funkfeuern und erdgebundenen Referenzstationen) begründet. Der größte

Abbildung 4.3: Inertiales Trackingsystem PCTracker von Intersense

Nachteil des Inertialen Trackings besteht trotz seiner Genauigkeit darin, dass es mit fortschreitender Zeit zu driften beginnt, was die Ergebnisse ohne Rekalibrierung zunehmend verschlechtert [26].

4.4 Laserbasiertes optisches Tracking

Einen sehr interessanten Ansatz verfolgt das Trackingsystem LaserBird2 [33]. Es ist für den Betrieb im Flugsimulator/Flugzeugcockpit prädestiniert (Abstand Sensor-Emitter laut technischer Spezifikation: 0,15m-1,83m) und arbeitet nach folgendem Prinzip: Ein fest installierter Emitter strahlt vier zu Fächern aufgespreizte Laserstrahlen ab, welche ein pyramidenförmiges Volumen abscannen, worin das Trackingobjekt lokalisiert ist. Trifft nun das Laserlicht, welches hochfrequenzmoduliert im nahen Infrarotbereich strahlt ($\tilde{7}85$ nm), auf die drei photooptischen Sensoren, die am Objekt befestigt sind, so kann aus deren Empfangssignal die Position des Detektors in den 6 Freiheitsgraden bestimmt werden. Trotz der Genauigkeit des Systems (Herstellerangaben: Position: 0.7 mm Orientierung: 0,5°) liegt wohl der größte Nachteil in den hohen Anschaffungskosten.

4.5 Optisches (kamerabasiertes) Tracking

Beim optischen Tracking wird das Bild der Umgebung mit einer Videokamera erfasst und als Pixelmatrix verarbeitet. Üblicherweise arbeiten optische Systeme im sichtbaren Bereich oder aber im Infrarotbereich. Nun müssen bestimmte Punkte im Bildausschnitt identifiziert werden, deren Lage im Raum bekannt ist. Dann kann aus deren Konstellation die Position und Orientie-

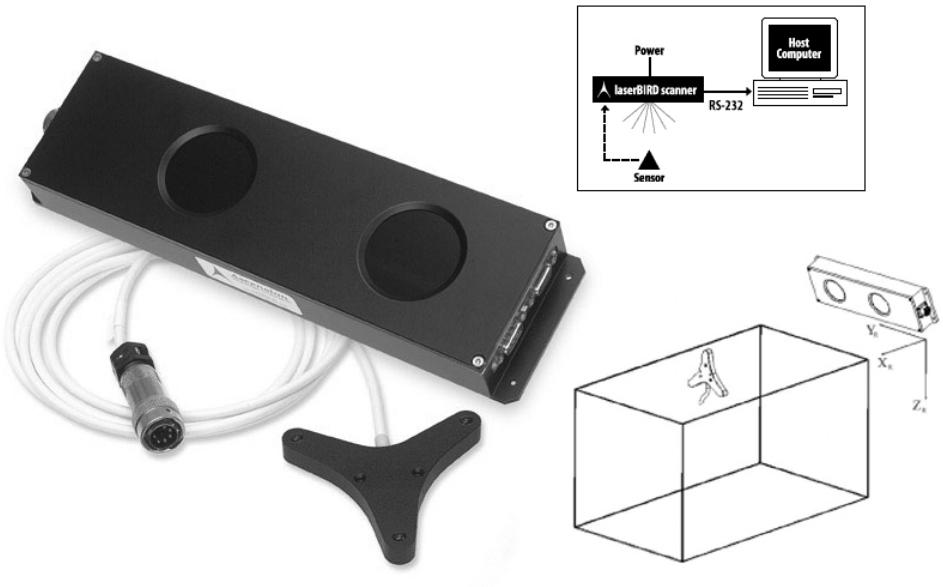


Abbildung 4.4: Ascension Technology Corp. - laserBIRD 2

rung der Kamera ermittelt werden. Ist die Position der Kamera erst bekannt, so gibt es zwei verschiedene Strategien: Entweder man berechnet aus dem Vergleich des Folgebildes mit dem aktuellen die in der Zwischenzeit erfolgte Rotation/Translation und akkumuliert diese auf die aktuellen Werte [13], oder man berechnet für jedes Bild die Lage der Kamera neu. Letztere Methode hat den Vorteil, dass keine Drift-Effekte auftreten können, da keine Fehlerakkumulation über die Zeit stattfindet. Nachteilig wirkt sich jedoch aus, dass die vollständige Positionsberechnung für jedes Einzelbild erfolgen muss, was bei rechenaufwändiger Bildanalyse zu Verlangsamung des Systems führen kann.

4.5.1 Natürliche/künstliche Marker

Eine grundlegende Frage beim Entwurf eines optischen Trackingsystems ist die Benutzung von natürlichen oder künstlichen Markern. Bei Verzicht auf künstliche Marker, also beim markerlosem Tracking [13] [14], müssen andere Bildmerkmale wie Kanten, Ebenen oder der Horizont als Referenzpunkte herangezogen werden. Derartige Ansätze leiden jedoch noch unter mangelnder Präzision bzw. unzureichender Geschwindigkeit. Demgegenüber stehen klare Vorteile von künstlichen Markern. Zuerst lassen sich durch ihre Verwendung Objekte eindeutig identifizieren. Desweiteren kann ihre genaue Position durch einmalige Vermessung exakt ermittelt werden. Einzige Einschränkung für die Verwendung künstlicher Marker ist, dass die Anwendung eine freie Markerplatzierung bzw. Markergröße nicht zulässt.

4.5.2 Aktive/passive Marker

Ist die Wahl auf die Benutzung künstlicher Marker getroffen, so stellt sich die Frage nach der Benutzung von aktiven oder passiven Markern. Aktive Marker emittieren Strahlung (farbiges Licht, Infrarotlicht), Schall oder sonstige Signale, durch welche sie identifiziert werden können. Von großem Vorteil ist hierbei, dass durch die Erhöhung der Emissionsintensität ein hoher Kontrast zum Hintergrund hergestellt werden kann, was das Auffinden der Marker stark vereinfacht. Zu beachten sind jedoch mögliche Wechselwirkungen oder Störeinflüsse auf die Umwelt. So stellt beispielsweise bei medizinischen Anwendungen in Verbindung mit Kernspintomographen die Benutzung ferromagnetischer Marker ein Problem dar [24]. Darüber hinaus benötigen aktiv emittierende Marker eine eigene Energieversorgung in Form von Batterien oder kabelgebundener Stromversorgung, was die Installation und Platzierung der Marker erschwert und zudem die Kosten, das Gewicht und den Strombedarf des Gesamtsystems erhöht. Passive Marker sind demgegenüber einfach herzustellen, doch sind sie in viel stärkerem Maße Störeinflüssen wie mangelndem Kontrast oder ungünstigen Beleuchtungsverhältnissen (wie z.B. starkem Schattenwurf durch grelles Licht) ausgesetzt. Einige dieser Probleme lassen sich jedoch durch die Verwendung eines mit der Kamera synchronisierten Blitzes und Marker mit einer reflektiven Oberfläche kompensieren [24].

4.5.3 Inside-Out vs. Outside-In Tracking

Eine weitere konzeptionelle Frage beschäftigt sich mit der Anbringung der Kamera. Befestigt man diese ortsfest und versieht das zu trackende Objekt mit Markern, so spricht man von Outside-In Tracking. Wird die Kamera jedoch direkt am zu trackenden Objekt (wie z.B. dem Pilotenhelm) angebracht und die Umgebung mit Markern versehen, so spricht man von Inside-Out Tracking. Wie bei den vorangegangenen Entscheidungen ist auch hier die konkrete Anwendung ausschlaggebend. Ein Parameter, welcher starken Einfluss auf die Entscheidung hat, ist die wahrgenommene Größe der Marker im Kamerabild. Betrachtet man Trackingsysteme mit nur einer Kamera, so muss diese beim Outside-In Tracking den gesamten zulässigen Bewegungsraum des zu trackenden Objekts abdecken. Nun wird ersichtlich, dass bei größerer Entfernung dessen von der Kamera die Anzahl der Pixel, welche das Objekt (mit den daran angebrachten Markern) repräsentieren, abnimmt und unter Umständen nur noch einen kleinen Bruchteil der zur Verfügung stehenden Auflösung ausschöpft. Kann man demgegenüber jedoch gewährleisten, dass bei Anbringung der Kamera am zu trackenden Objekt im Bewegungsraum bei jeglichem Aufenthaltsort des Objekts genügend Marker im Bild sind, so ist die Aufrechterhaltung einer gleichbleibenden Präzision möglich. Beim Inside-Out Tracking kann man daher - über den gesamten Bewegungsraum betrachtet - von einer höheren Winkelgenauigkeit als beim Outside-In Tracking ausgehen.

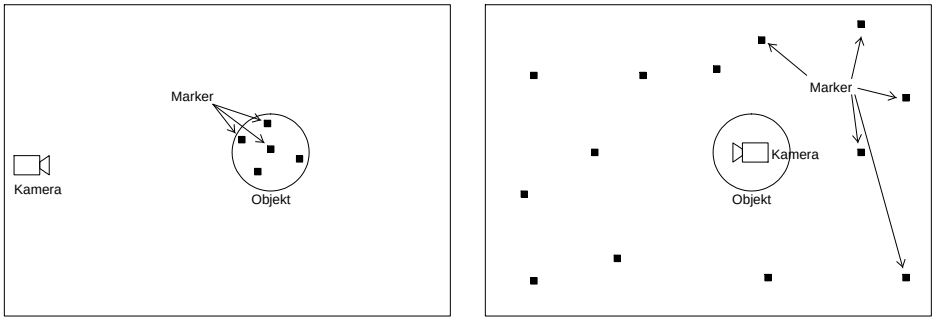


Abbildung 4.5: a) Outside-In Tracking b) Inside-Out Tracking

4.5.4 Markerdesign

Das eigentliche Design der künstlichen Marker hängt stark von der verwendeten Registrationstechnik ab. So steht im Infrarotspektrum nur monochromatisches Licht zur Verfügung, d.h. Marker können sich lediglich durch Form und Gestalt, nicht jedoch durch ihre Farbe unterscheiden. Aktive Marker, wie Infrarot-Dioden, sind per se oft gleichförmiger Gestalt und somit ununterscheidbar. Durch die Gruppierung zu einem eindeutigen Muster kann jedoch eine eindeutige Zuordnung erreicht werden. Aus der Gesamtkonstellation, bzw. einem Ausschnitt derer, wird dann auf die aktuelle Lage im Raum geschlossen. Andere Ansätze werden im sichtbaren Lichtspektrum verfolgt [18] [19]. Dort greift man oftmals auf größerdimensionierte Marker zurück, wie z.B. beim AR-Toolkit. Hierbei genügt es dann jedoch, einzelne Marker im Bildausschnitt zu erfassen. Durch kontraststarke Markergrenzen (schwarze Box auf weißem Hintergrund) kann der Marker im Bild gefunden werden. Die Unterscheidung der Marker geschieht dann über die Analyse des Musters innerhalb des Rahmens mittels Patternmatching [19]. Eine weitere Möglichkeit zur Indentifizierung des Markers ist die Verwendung von 2d-Barcodes, wie in [16] referenziert.

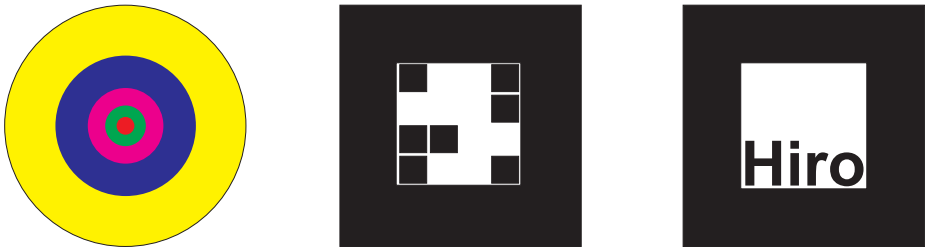


Abbildung 4.6: a) Multi-Ring multi-color Marker b) 2d-barcodierter Marker c) AR Toolkit Marker

Schließlich gibt es auch Ansätze, welche durch Farbkodierung die Marker unterscheiden [17]. Es ist jedoch zu beachten, dass sich durch die Verwendung

von farbigen Kamerabildern (z.B. RGB[3]) im Vergleich zu Graustufenbildern die Datenmenge pro Bild verdreifacht, da statt eines Grauwertes pro Pixel nun für 3 Farbkanäle Information gespeichert werden muss. Bei einer Bildgröße von 640x480 Pixel bedeutet dies eine Datenmenge von 901 KB[4] (RGB) im Vergleich zu 300 KB (Graustufen). Legt man eine Bildwiederholrate von 30 Frames zugrunde, so steht eine Datenmenge von 27,03 Megabyte pro Sekunde einer von 9,01 MB/sec gegenüber.

[3]Pro Bildpunkt (Pixel) werden 3 Farbkanäle gespeichert, d.h. jeder Pixel hat einen Rot-, Grün- und Blauanteil.

[4]KB: 1 Kilobyte entspricht 1024 Byte, 1024 KB entsprechen 1 MB (Megabyte)

Kapitel 5

Anforderungsanalyse

Das Ziel dieser Diplomarbeit ist es, ein optisches Headtrackingsystem mit den zur Verfügung stehenden Mitteln an Hardware und bekannten Augmented Reality Techniken an die Anforderungen und Einschränkungen eines Flugzeugcockpits anzupassen. Hierzu sollen im Folgenden die Ziele definiert sowie weitere Einschränkungen bzw. Vorgaben aufgezeigt werden. Die Haupteinschränkung liegt, wie es für viele Projekte typisch ist, in den begrenzten Resourcen, was sowohl Zeit und verfügbare Hardware als auch das zur Verfügung stehende Budget anbelangt.

5.1 Funktionelle Anforderungen

Die funktionellen Anforderungen bestehen darin, dem Piloten beim Blick aus dem Fenster die in Abschnitt 2.1 beschriebenen Flugführungsinformationen in ein Head-Mounted Display einzublenden. Hierzu muss das Trackingsystem mit akzeptabler Geschwindigkeit und Wiederholrate die Kopfposition und Orientierung bestimmen und dementsprechend das Rendern der Informationsdarstellung veranlassen. Diese wird daraufhin im Head-Mounted Display dargestellt.

5.2 Nichtfunktionelle Anforderungen

Zu den nichtfunktionellen Anforderungen gehört primär die Minimierung der Latenzzeit des Systems. Die bedeutet, dass die von der Sensorerfassung bis zur Darstellung der Information im HMD verstrichene Zeit auf ein Minimum reduziert werden muss, da bei Systemen der Flugnavigation Echtzeitanforderungen gelten. Darüber hinaus wird eine hohe Updaterate für die Positionsbestimmung angestrebt, um eine möglichst flüssige Darstellung zu gewährleisten. Limitierender Faktor ist hier die Bildwiederholfrequenz der Kamera, welche in unserem Fall als Sensor zur Positionsbestimmung dient. Diese liegt bei einer Auflösung von 640x480 Pixel im Graustufenmodus bei 30Hz, im Farbmodus der Kamera (RGB) fällt diese bei gleicher Auflösung auf 15Hz ab. Was die Präzision der berechneten Kopfposition anbelangt, wird eine Winkelgenauigkeit von unter 1° angestrebt. Die translatorische Präzision der Rechenergebnisse spielt bei dieser Anwendung eine zweitrangige Rolle,

da die projizierten Objekte in relativ weiter Entfernung zum Cockpit liegen und somit Positionsungenauigkeiten im Bereich von ca. 10cm einen vernachlässigbaren Einfluss auf die Ausgabe darstellen.

5.3 Pseudo Anforderungen

Die zu entwickelnde Software wird in C/C++ geschrieben. Hierbei wird, soweit dies möglich ist, auf betriebssystemspezifische Softwarekomponenten verzichtet, um die Portabilität auf andere Systeme zu erhalten. Die Bildverarbeitung erfolgt mit Hilfe der openCV Bibliothek der Intel Corporation [38].

5.4 Zur Verfügung stehende Hardware

Folgende Hardware stand für das Projekt zur Verfügung:

- Standard-PC mit Intel Pentium 4 2,40 GHz, 512MB RAM, nVidia GeForce 4 Ti 4200
- PointGrey Research Firefly 2 Progressive Scan Webcam, Firewire 1394, 640x480 schwarz/weiß bei 30Hz
- Infrarotfilter (Tageslichtsperrfilter) Linos RG 830
- Sony Glasstron Head-Mounted (Optical See-Through) Display

Zusätzlich wurden zur Installation der Infrarot-Dioden sowie der Cockpitverschalung folgende Komponenten benutzt:

- Instrumentenbrett-Verkleidung aus Kunststoff, 5mm
- Autopilot-Pannel-Verkleidung aus Aluminium
- Infrarotdioden SFH 421
- LEGO Bauplatte 628 380mm x 380mm
- LEGO Baustein 1x2 3725 8mm x 16mm x 11mm
- Elektrischer Leiter, Doppellitze FL124
- Elektrischer Widerstand WID.0207 33 Ohm, 0,6 Watt
- Gleichstromtransformator 12V Siemens Arnold ASS

Kapitel 6

Architektur und Design-Entscheidungen

Im folgenden Abschnitt wird die Architektur des entwickelten Systems vorgestellt und die einzelnen Module erläutert. Aufgrund der programmatischen Vorgaben und der verfügbaren Hardware ist der verfolgte Ansatz ein optisches Trackingsystem, das sich durch sein geringes Gewicht und seine geringe Größe auszeichnet. Da es sich beim vorliegenden Konzept um ein System mit nur einer Kamera handelt, der Trackingraum über sehr begrenzte Dimensionen verfügt und der Schwerpunkt bei der Positionsgenauigkeit auf den Winkeldaten liegt, also mehr der Orientierung als der translatorischen Position, fiel die Entscheidung auf einen Inside-Out Trackingansatz. Dies bedeutet, dass am HMD (also am zu trackenden Objekt) eine Kamera befestigt wird, welche die im Cockpit angebrachten Marker erfasst. Aufgrund der sehr beschränkten Platzverhältnisse zwischen den Instrumenten (wie in Abb. 3.1 zu sehen) müssen die Marker sehr klein sein. Diese Tatsache spricht gegen die Verwendung von Markertypen, wie sie beispielsweise in ARToolkit [19] verwendet werden. Darüber hinaus eignen sich derartige Marker besser zur Augmentierung direkt auf die markierte Fläche, weniger jedoch zur Extraktion genauer Positions- und Orientierungsdaten.

6.1 Überblick

Schaubild 6.1 soll das Zusammenwirken der einzelnen Komponenten verdeutlichen.

Die Kamera erfasst die im Cockpit installierten, aktiven Marker. Sie ist über eine Firewire Schnittstelle nach IEEE 1394 Norm mit dem Rechner verbunden. Hier wird das Videosignal verarbeitet und die Positionsberechnung durchgeführt. Als weitere Eingabe erhält der Rechner die Positions- und Orientierungsdaten des Flugzeugs. Müsste im realen Flugzeug dieser Datentransfer über eine CAN-Aerospace Datenbus Karte erfolgen, so ist im Simulationsbetrieb ein Datenaustauch über das Simulationsnetzwerk möglich. Aus der Verrechnung der Flugzeugdaten mit der aktuellen Blickrichtung des Piloten kann daraufhin die für den Tunnel zu rendernde Perspektive bestimmt werden. Diese Daten werden an die Rendering-Engine in Form einer Trans-

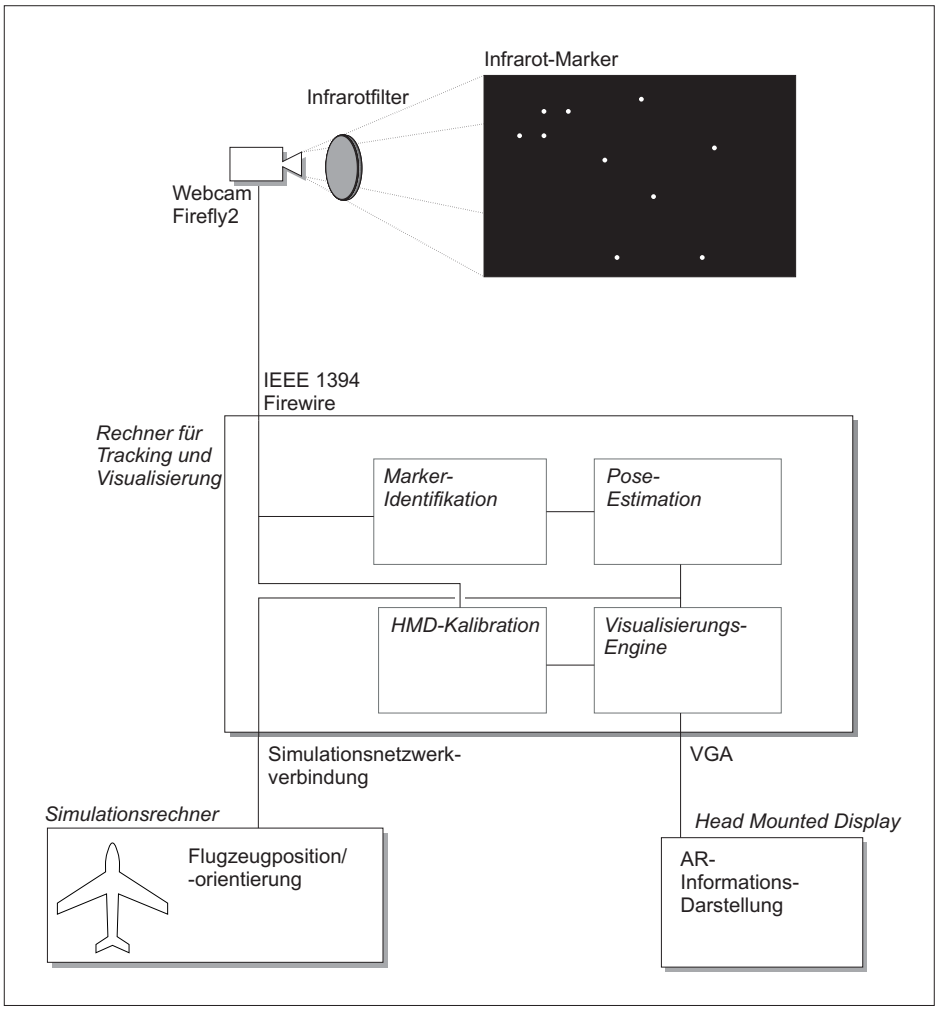


Abbildung 6.1: Systemarchitektur - Überblick

formationsmatrix übergeben, welche letztendlich die Darstellung im Head-Mounted Display ausführt. Dieses wird über den Standard-Monitorausgang mit dem Rechner verbunden. Die Software setzt sich aus folgenden Modulen zusammen:

- Framegrabber+Kameratreiber
- Bildverarbeitung (Filterung und Markeridentifikation)
- Pose Estimation (3d-Positionsberechnung aus den 2d-Bilddaten)
- Ausgabe der Visualisierung

Zusätzlich sind folgende Kalibrationsroutinen zu implementieren:

- Einmalige Kalibration der Kameraparameter
- Benutzerspezifische Kalibration des Head-Mounted Displays

6.2 Marker und deren Anbringung im Cockpit

Als Marker wurden Leuchtdioden gewählt, welche im Infrarotspektrum emittieren. Diese erleichtern in Kombination mit einem Tageslichtsperrfilter die nachfolgende Bildverarbeitung, was die Gesamtperformanz des Systems steigert. Durch die Wahl aktiver Marker im IR-Spektrum ergeben sich jedoch weitere Vorteile. So basieren die Methoden beim optischen Tracking zum Auffinden der Marker im Videobild oftmals auf der Suche nach Kanten oder kontraststarken Regionen [19]. Fallen jedoch bei grellem Tageslicht Licht-Schatten Grenzen auf optische Marker, so macht dies deren Identifikation unter Umständen unmöglich. Demgegenüber bietet das gewählte Verfahren zwei Vorteile. Zum einen könnte durch einen weiteren Filter, welcher den Frequenzbereich oberhalb der Emissionswellenlänge der IR-Dioden absorbiert, das Spektrum des erfassten Lichtes scharf auf den Emissionsbereich der Dioden eingegrenzt werden. Zum zweiten können aktive Marker in ihrer Helligkeit angepasst werden. So könnte ein trotz Filterung verbleibendes Rest-Hintergrundbild einfach durch Erhöhung der Helligkeit der Marker überstrahlt werden.

Die ausgewählten Dioden vom Typ Siemens SFH-421 verfügen über einen großen Abstrahlwinkel (bei 60° noch 50% Intensität). Dies ist notwendig, um auch unter schrägem Blickwinkel ein ausreichendes Signal zu erfassen. Darüber hinaus haben sie sehr kleine Einbaumaße, was insbesondere im Hinblick auf die beschränkten Platzverhältnisse zwischen den Instrumenten von Vorteil ist.

Sie haben einen Spannungsabfall von 1,5V bei einer benötigten Stromstärke von 100mA pro Diode. Als Versorgungsspannung wurden 12V gewählt, da dies eine übliche und an Bord verfügbare Größe darstellt. Zur Vereinfachung der Installation wurden jeweils sechs Leutdioden in Reihe geschaltet und mit einem Vorwiderstand von 33 Ohm versehen. Jeweils 4 dieser Reihen wurden parallel an einen 12V Gleichstrom-Transformator angeschlossen (siehe Anhang B.1).

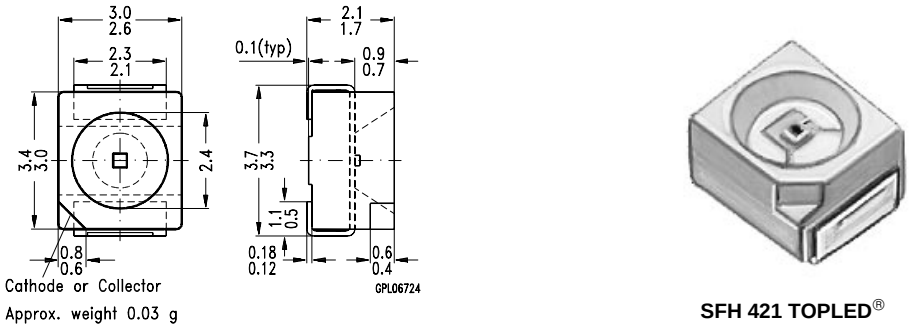


Abbildung 6.2: Siemens GaAIAs-IR Lumineszensdiode SFH 421

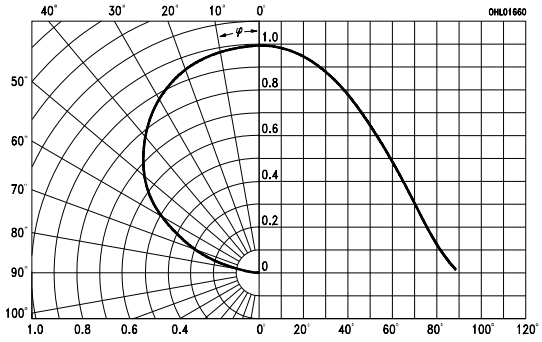


Abbildung 6.3: SFH 421: Leuchtintensität über den Abstrahlwinkel aufgetragen

Um in der Testphase möglichst flexibel in der Anordnung der Marker zu bleiben, wurden die Hauptbereiche, die für eine Ausstattung mit Markern in Frage kamen, mit LEGO Platten ausgekleidet. Diese sind auf Aluminium bzw. Kunststoffplatten aufgebracht, welche über den Konsolen im Cockpit montiert wurden. Diese Konstruktion ist offensichtlich nur für Testzwecke geeignet, da dadurch z.T. Bedienelemente des Simulators verdeckt wurden. Dennoch schien sie unter den gegebenen Rahmenbedingungen und dem parallel dazu weiterlaufenden Forschungsbetrieb im Simulator als die beste Lösung.

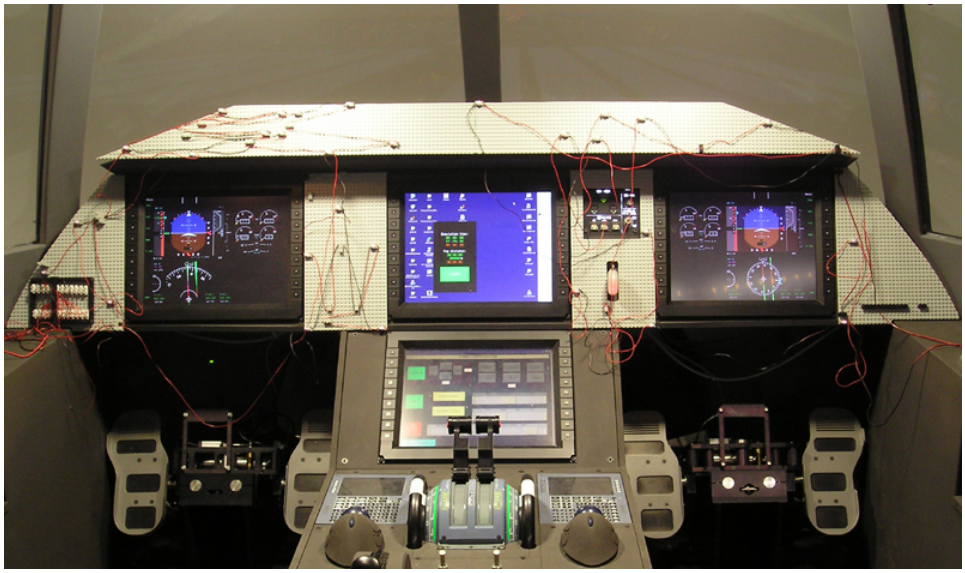


Abbildung 6.4: LEGO-Auskleidung des Instrumentenbretts

Neben dem raschen Ein- und Ausbau der LEGO-Verkleidungen in den Simulator erwies sich das einfache Anbringen/Umsetzen der LEDs sowie die schnelle Vermessungsmöglichkeit aufgrund der diskreten Steinabstände als weiterer Vorteil.

6.3 Bilderfassung

Zur Bilderfassung dient uns eine Firefly2 Progressive Scan Webcam von Point Grey Research, welche über eine Firewire Schnittstelle (IEEE 1394) mit dem PC verbunden ist. Sie ist in der Lage, bei einer Auflösung von 640x480 Pixel non-interlaced Graustufenbilder bei einer Rate von 30 Herz zu liefern.

Der auf der Kamera verwendete SONY ICX098AX CCD Sensor [36] ist in der Lage, Lichtwellen jenseits des sichtbaren Bereiches zu erfassen. Insbesondere im Infrarotbereich hat er zudem ein sehr gutes Ansprechverhalten. Als Objektiv kommt ein Weitwinkelobjektiv mit einer Brennweite von 4mm zum Einsatz. Da im erfassten Bildauschnitt genügend Marker erfasst werden müssen, um eine Positionsbestimmung errechnen zu können, fiel die Wahl auf eben dieses Objektiv. Bei einem Schwenkbereich der Blickrichtung von ca. 150° vertikal kann so der gesamte Cockpitbereich mit etwa 4 Vollbildern

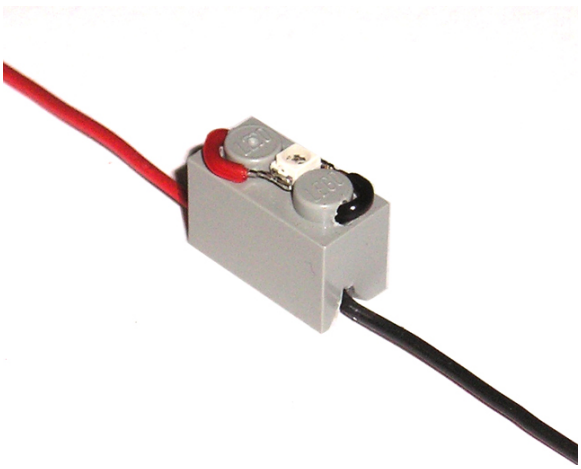


Abbildung 6.5: SFH-421 mit Stromversorgung auf LEGO-Stein montiert

Abbildung 6.6: Point Grey Research - Firefly 2

abgedeckt werden (Abb. 6.7). Ein Objektiv mit noch größerer Brennweite, d.h. mit geringerem Sichtwinkel, würde die Anzahl der benötigten Marker deutlich steigern. Demgegenüber würde eine noch geringere Brennweite und damit ein noch größerer Sichtbereich die radiale Linsenverzerrung weiter erhöhen, was zu zunehmender Ungenauigkeit am Bildrand führen würde. Somit bildet die gewählte Lösung einen vertretbaren Kompromiss zwischen der Anzahl der Marker, welche im Größenbereich von ca. 10 - 15 Markern pro Bild liegt, und der angesprochenen „Fischaugen"-Verzerrung des Bildes. Angesichts der Tatsache, dass die Pilotensicht (wie in Abschnitt 3.2 beschrieben) nur beim Blick aus dem Fenster mit dem Tunnel augmentiert werden soll, muss die am HMD besestigte Kamera mit einem Neigungswinkel von ca.25° nach unter montiert werden, um sicherzustellen, dass stets genügend der auf dem Instrumentenbrett angebrachten Marker von der Kamera erfasst werden können.

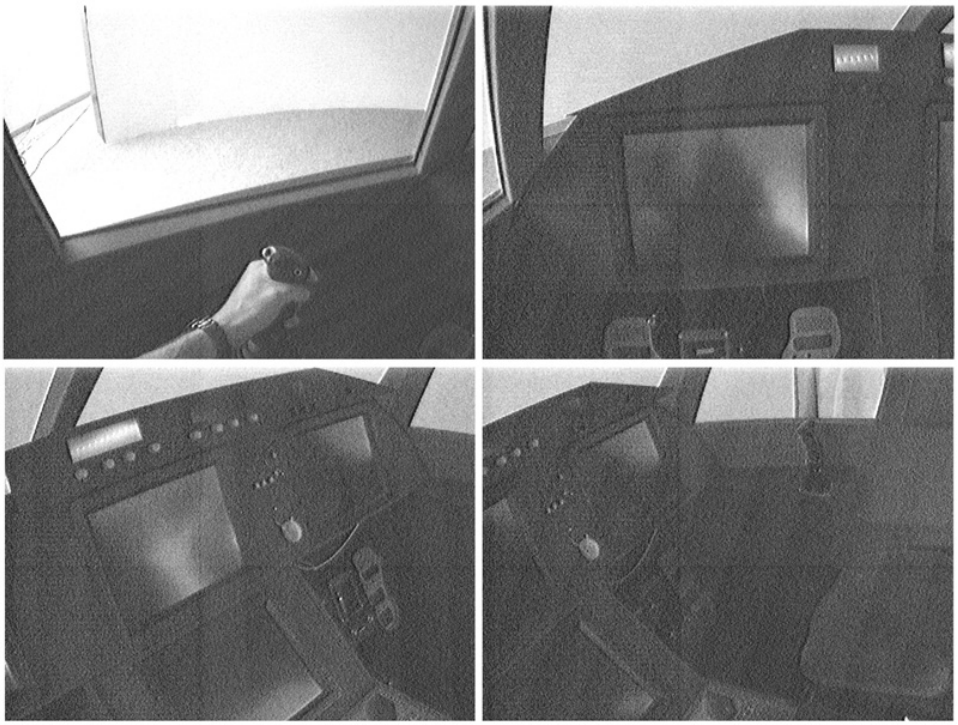


Abbildung 6.7: Blickfeldabdeckung der Kamera mit 4mm-Objektiv

Um nun die im vorangegangenen Abschnitt beschriebene Verlagerung der Kalibrationspunkterfassung ins Infrarotspektrum zu erlangen, wird vor der Linse der Kamera ein Tageslichtsperrfilter angebracht, welcher elektromagnetische Wellen bis 730nm vollständig herausfiltert. Hierzu wurde ein Planoptik-Filter der Firma Linos Photonics vom Typ RG 830 verwendet.

Dessen Durchlässigkeit beginnt bei 720nm und erreicht einen Transmissionsgrad von 90% bei 850nm (siehe Anhang B.4). Die verwendeten Infrarot-Dioden haben ihre maximale Emissionswellenlänge bei 880nm +/- 20nm und werden damit hervorragend vom Kamera/Filtersystem erfasst (siehe Anhang

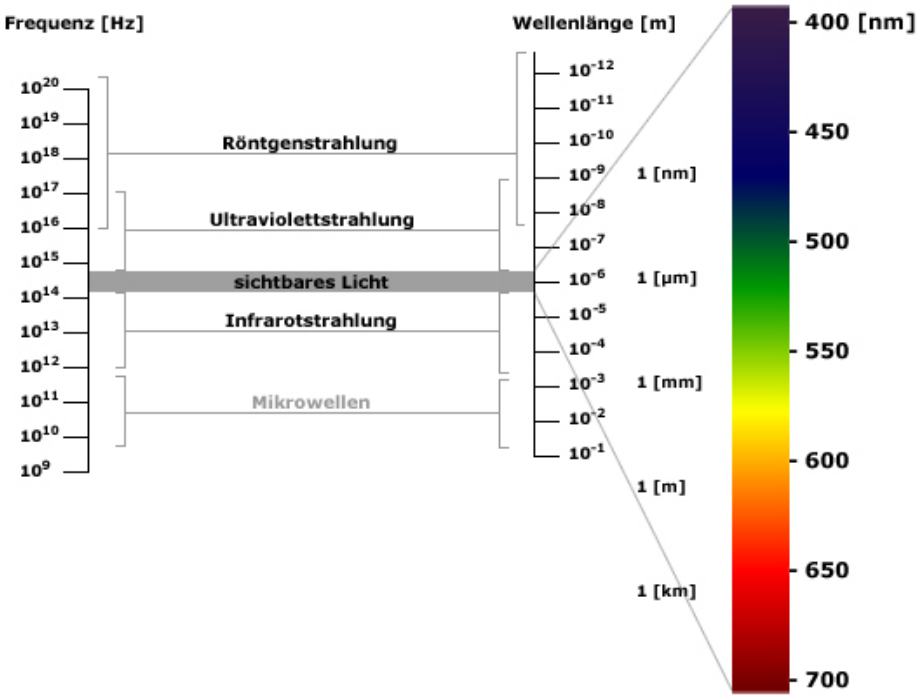


Abbildung 6.8: Das sichtbare Licht im elektromagnetischen Spektrum

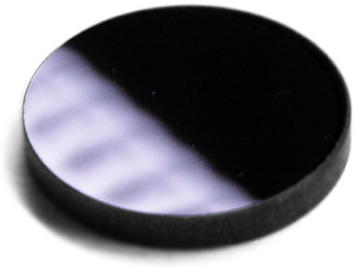


Abbildung 6.9: Linos Photonics - IR-Filter RG 830

B.5).

Im Diagramm angedeutet ist der Bereich sichtbaren Lichts zwischen 380nm und 750 nm. Rechts violett unterlegt ist der vom Filter durchgelassene Spektralbereich zu erkennen sowie das Emissionsband der IR-Leuchtdiode von 860nm - 900nm.

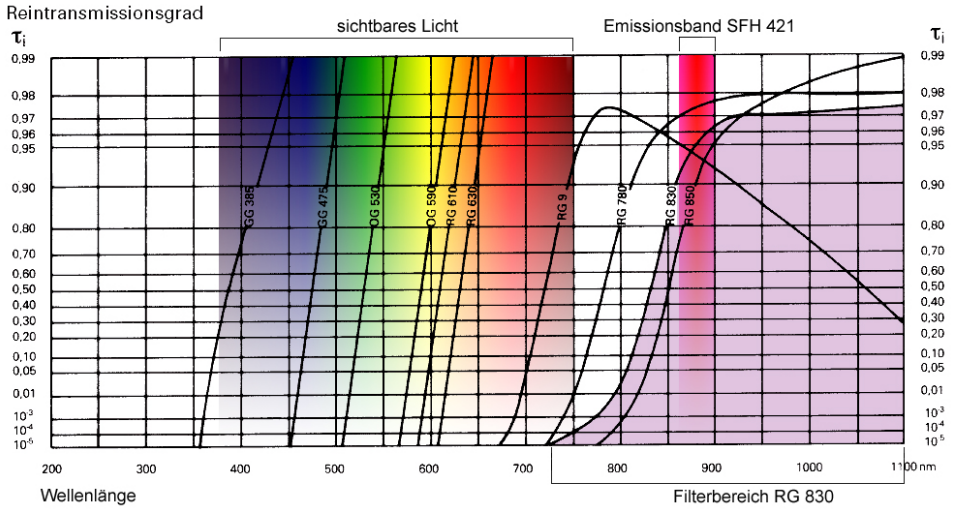


Abbildung 6.10: Filterbereich RG 830 und Emissionsbereich SFR 421

Abb. 6.11 soll einen Eindruck davon vermitteln, in wie weit das Auffinden der Marker durch die Filterung vereinfacht wird.

6.4 Bildverarbeitung

Die nun folgende Bildverarbeitung besteht aus zwei Hauptaufgaben:

1. Berechnung der Schwerpunkte der einzelnen IR-Marker
2. Identifikation jedes (bis dato ununterscheidbaren) Markers

Durch die optische Vorfilterung des Kamerabildes wird der Aufwand, die Marker im Videoframe ausfindig zu machen, bereits extrem vereinfacht. Dennoch unterliegt jedes einzelne Bild noch Störeinflüssen, welche sich durch Rauschen bemerkbar machen. Bild 6.12 macht das im Originalausschnitt (links) kaum bemerkbare Rauschen durch Verstärkung des Bildkontrasts und der Bildhelligkeit sichtbar.

Um jedoch die Schwerpunkte der einzelnen Lichtpunkte berechnen zu können, wird das Bild mittels eines Schwellenwertes binärisiert (Thresholding). Jeder Pixel, dessen Graustufenwert unter dem Grenzwert liegt, wird auf schwarz (0) gesetzt, jeder Pixel der heller ist, auf weiß (255).

Auf den binären Bilddaten (Abb. 6.13) kann nun eine Schwerpunktsberechnung erfolgen, deren Ergebnisse dann als Datensatz für die 2d-3d Korellation an den Positionsbestimmungsalgorithmus übergeben werden. Dazu muss jedoch noch die einzelne Identität jedes Markers vorliegen. Deren Bestimmung wird im nachfolgenden Kapitel ausführlich dargestellt.

Abbildung 6.11: Cockpitausschnitt a) ungefiltert b) bei Dunkelheit c) im Infrarot-Bereich

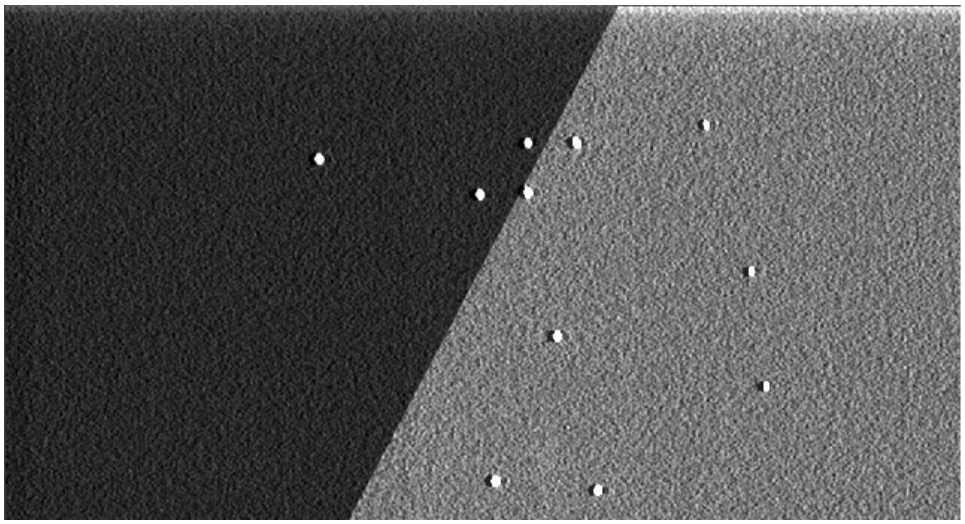


Abbildung 6.12: Vorgefiltertes IR-Videobild - Hinergrundrauschen

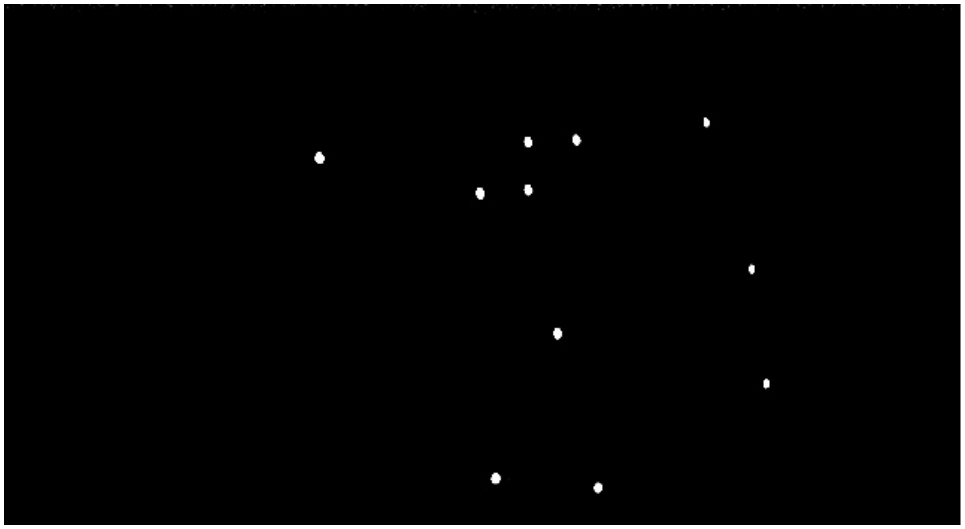


Abbildung 6.13: Binärisiertes IR-Bild

6.5 Positionsberechnung

Liegt letztendlich vollständiges Wissen über die Marker vor, d.h. sowohl ihre Position im Kamerabild (X-/Y-Koordinate), als auch ihre 3d-Position im globalen Referenzsystem (in unserem Fall einem im Cockpit aufgespannten System in X-/Y-/Z-Richtung), so kann bei genügend großer Anzahl von Korellationspunkten die Kameraposition und -orientierung errechnet werden. Hierzu wird der Standardalgorithmus von Roger Tsai verwendet.

6.5.1 Tsai's Kameramodell und reale Kamera

Der Algorithmus von Tsai verwendet ein Lochkameramodell zur Modellierung von Kamera und Objektiv. Wichtig ist hierbei, dass im Algorithmus von einer konstanten Brennweite der Linse (Focal lenght f) ausgegangen wird. So darf diese nach erfolgtem Kalibrationsprozess nicht mehr verändert werden. Bei der Anwendung im Cockpit muss gewährleistet sein, dass alle Marker scharf dargestellt werden. Dazu wurden die Minimal- bzw. Maximalabstände zwischen Augpunkt (Kameraposition) und dem nähsten bzw. am weitest entfernten Marker vermessen, welche bei 55cm bzw. 160cm lagen. Durch die Benutzung des 4mm-Objektivs sowie einmaliger Justage des Schärfebereichs konnten die obigen Forderungen erfüllt werden. Desweiteren modelliert der Algorithmus von Tsai eine radiale Linsenverzerrung, welche bei der Verwendung weitwinkliger Objektive eine zunehmende Rolle spielt. Diese wird mathematisch durch eine unendliche Reihe modelliert [6], defacto wird jedoch nur der erste Koeffizient *k1* zur Berechnung herangezogen [7]. Nach Tsai spielt der Einfluß der weiteren Reihenelemente eine zu vernachlässigende Rolle und deren Berücksichtigung würde eher zu einer numerischen Instabilität führen.

6.5.2 Tsai's Algorithmus - Überblick

Wie bereits im vorangegangenen Abschnitt erwähnt, modelliert Tsai die optische Abbildung der Szene auf den Kamerasensor durch ein Lochkameramodell. Hierbei ergibt sich der Bildpunkt auf der Projektionsebene durch folgenden geometrischen Zusammenhang:

1. Der Strahl vom optischen Zentrum O (Brennpunkt) zum Weltkoordinatenpunkt P schneidet die Bildebene in P_u, dem unverzerrten Bildpunkt von P (P undistorted).

2. Da jedoch eine Linse dazwischen liegt, verzerrt sich der tatsächliche Bildpunkt radial vom Zentrum der Bildebene O_f abhängig vom Abstand zu diesem (Radius Bildmitte - P_u). Dieser die Verzerrung berücksichtigende Punkt wird P_d genannt (P distorted).

Die vollständige Berechnung der Kameraposition und Orientierung läuft nun folgendermaßen ab. Der im karthesischen Weltkoordinatensystem registrierte Punkt P_w wird durch Rotation R und Translation T in ein 3d-Koordinatensystem überführt, in welchem die durch die X- und Y-Achse

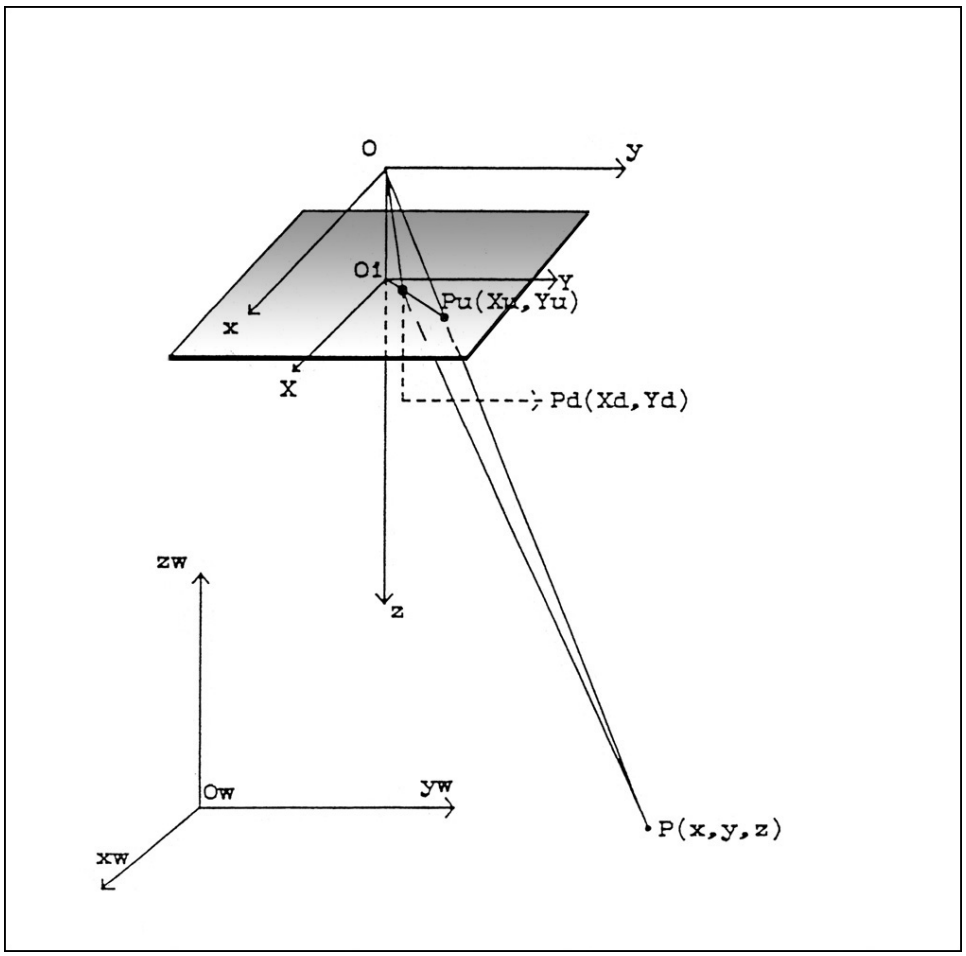


Abbildung 6.14: radiale Linsenverzerrung im Lochkameramodell bei Tsai's Algorithmus

aufgespannte Ebene koplanar zur Bildebene liegt, die Z-Achse die Bildebene in ihrem Mittelpunkt (C_x,C_y) durchstößt und folglich auch durch das optische Zentrum verläuft. Nun wird auf den Bildpunkt $P_u$die radiale Linsenverzerrung angewandt und es ergibt sich P_d. Dieser wird schießlich in konkrete Bildkoordinaten transformiert. Diese ergeben sich aus der Anzahl der Sensorelemente der Kamera, der Auflösung des Videobildes (Anzahl der Pixel in X- und Y-Richtung), weiterer technischer Parameter sowie der zusätzlich eingeführten Größe s_x, welche mögliche hardwareseitig auftretende Zeitverzögerungen bei der Bilderfassung modelliert. Somit kann man die gesamte Transformation von 3d-Weltkoordinate zur 2d-Bildkoordinate in geschlossener Form darstellen. Dies soll in Abb.6.15 verdeutlicht werden.

Bestehen nun genügend solcher 3d-2d Korellationen, so entsteht ein überbestimmtes Gleichungssystem, durch dessen Lösung die gesuchten Parameter ermittelt werden können. Die verwendete Implementation bietet zwei verschiedene Routinen zur Lösung an. Erstere setzt voraus, dass alle Registrationspunkte im Weltsystem in einer Ebene liegen. Zur vollständigen Lösung dieses (coplanaren) Falles sind mindestens 5 solcher 3d-2d Korellationen nötig. Sind die Weltpunkte in der abgebildeten Szene jedoch räumlich angeordnet, so erhält man mit der non-coplanaren Variante des Algorithmus ab einer Anzahl von mindestens 7 korrespondierenden Welt-/Bildpunkten eine Lösung. Aufgrund dieser Minimalanforderung und den im Abschnitt 6.2 beschriebenen Abdeckungsbereich der Kamera wurde eine Gesamtanzahl von ca. 40-50 Marker über die gesamte Cockpitblickfeldbreite angesetzt. Dadurch sollten zu jedem Zeitpunkt ca. 10 bis 15 Marker gleichzeitig erfasst werden, wodurch die algorithmischen Anforderungen auf jeden Fall erfüllt werden. Der Algorithmus von Tsai errechnet nun die Position und Orientierung der Kamera und gibt diese in Form eines Translationsvektors und einer 3x3 Rotationsmatrix bzw. 3 Eulerwinkeln aus. Ursprung dieses Referenzsystems ist der Ursprung der 3d-Weltkoordinatenangaben.

6.6 Datenausgabe

Ist die Orientierung und Position der Kamera relativ zum Referenzsystem Cockpit bekannt, so sind zur perspektivisch korrekten Darstellung des Tunnels im HMH noch zwei Transformationen vonnöten. Zum einen muss der (konstante) Offset zwischen der Kamera und dem HMD berechnet werden, da die Position des Augpunktes des Benutzers natürlich nicht mit der berechneten Kameraposition übereinstimmt. Zusätzlich ist eine benutzerspezifische Kalibration des HMDs notwendig, da sowohl der Augabstand von Benutzer zu Benutzer variiert, als auch nicht davon ausgegangen werden kann, dass das HMD jedesmal exakt gleich aufgesetzt wird. Zur Kalibration des HMDs wird der sogenannte SPAAM-Algorithmus [21] verwendet. Dabei wird dem Träger des HMDs an einer definierten Stelle im Display eine Marke eingeblendet, welche er durch Kopfbewegung mit einem fixen Referenzpunkt zur Deckung bringen muss. Ist dies geschehen, erfolgt eine Bestätigung des Benutzers und die Marke wird an einer anderen Stelle eingeblendet, welche er wiederum zur Deckung bringen muss. Dieser Vorgang wird einige Male wiederholt, bis

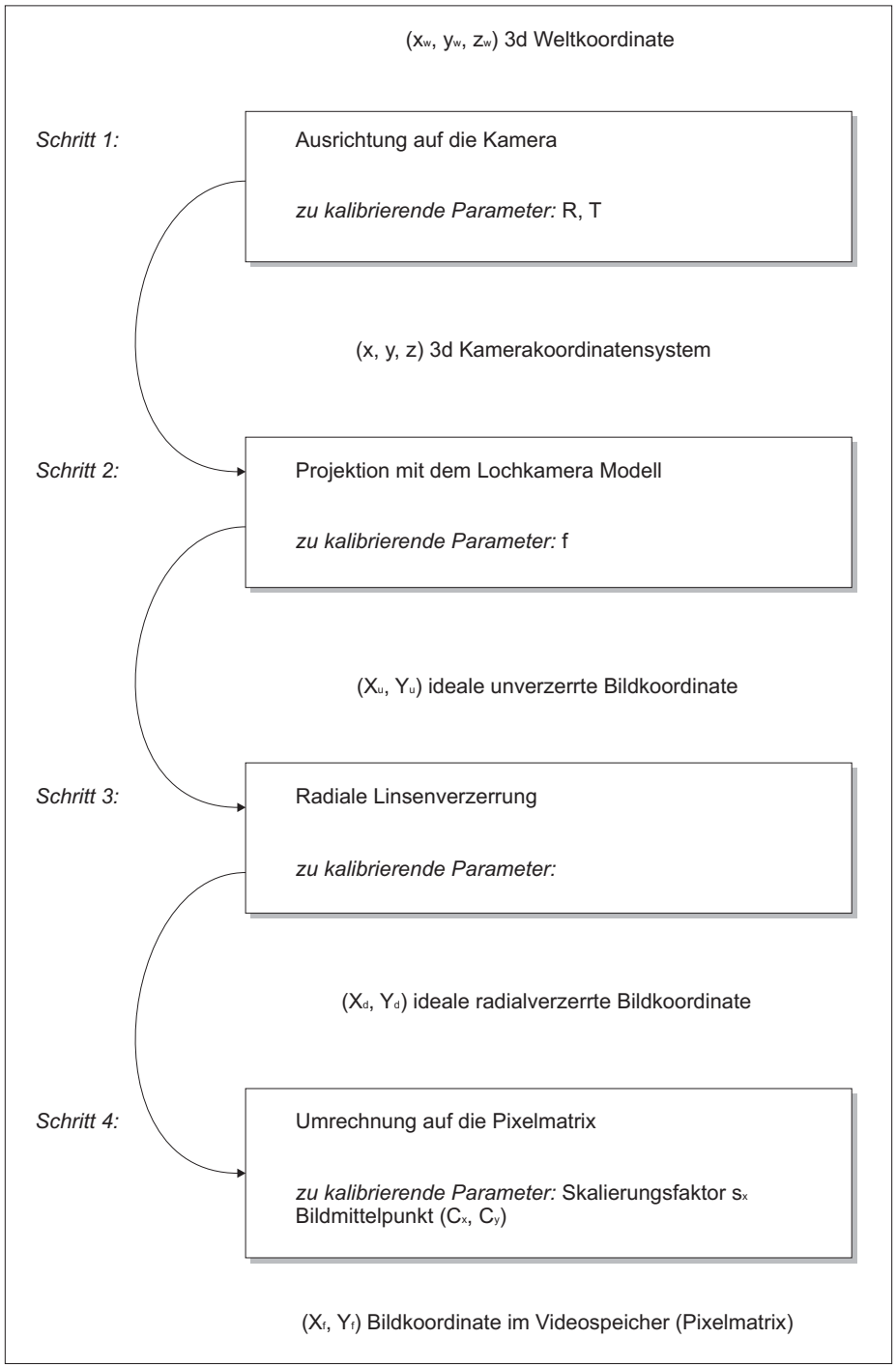


Abbildung 6.15: Tsai's Algorithmus: von der Weltposition zum Bildpunkt

die Kalibration des HMDs erfolgreich ist. Die zweite Transformation, welche zur Tunneldarstellung benötigt wird, ist die aktuelle Flugzeugposition und -orientierung. Kann man diese im Simulationsbetrieb über das Ethernet-Netzwerk erhalten, so müsste der Datentransfer im echten Flugbetrieb über eine Anbindung an den flugzeugeigenen CAN-Aerospace Echtzeit-Datenbus erfolgen. Keine der beiden Möglichkeiten konnte jedoch im Rahmen dieser Arbeit mehr realisiert werden. So wurde die Korrektheit der Trackingergebnisse dadurch verifiziert, dass die 3d-Daten der erfassten Marker in das Kamerabild rückprojiziert wurden (siehe Abschnitt 9.2.1).

Kapitel 7

Markeridentifikation

Ein Schwerpunkt dieser Diplomarbeit ist es, aus den durch die Kamera wahrgenommenen Markern ihre eindeutige Zuordnung zu ermitteln. Hierzu wird die Tatsache ausgenutzt, dass sich alle Marker auf nur wenigen Ebenen befinden. Die Idee bei der Markeridentifikation sei an dieser Stelle in ihrer Konzeption vorgestellt, die genaue Umsetzung ist in den jeweiligen Unterkapiteln beschrieben.

7.1 Prinzip der Markeridentifikation

Da alle Marker in ihrer Erscheinung ununterscheidbar sind, beruht die Identifikation auf der Suche nach einer bestimmten Konstellation[1], welche als Master bezeichnet wird. Ist diese gefunden, können die dazugehörigen Marker identifiziert werden. Nun wird, basierend auf dem Wissen über ihre Lage im Raum und ihre perspektivische Wahrnehmung durch die Kamera, eine Transformationsmatrix errechnet, welche eben diese Perpektive zum Ausdruck bringt. Daraufhin kann für die noch unidentifizierten Marker mittels der berechneten Matrix eine Positionsvorhersage getroffen werden. Von diesen projizierten Punkten wird nun nach dem Markern mit dem geringsten Abstand im Videobild gesucht und dieser bei Einhaltung von festgelegten Toleranzgrenzen der ID des projezierten Punktes zugewiesen.

7.1.1 Finden des Masters

Ausgangspunkt für die Zuweisung der Marker ist das Auffinden der Master-Konstellation. Diese zeichnet sich dadurch aus, dass sie aufgrund ihrer Form stets erkannt wird. Um ebendies zu gewährleisten, sind die Abstände zwischen den Master-Markern deutlich geringer als jegliche Abstände zweier beliebiger anderer Marker. Der Master besteht aus vier Markern, da dies die Mindestanzahl notwendiger Korrespondenzpunkte zur Berechnung der perspektivischen Transformation ist. Diese vier Marker sind folgendermaßen angeordnet:

Eine genaue Begründung der gewählten Anordnung sowie die Methode zur Identifikation wird in Kapitel 8.3.1 dargestellt. So gehen wir an dieser

[1] Das genaue Vorgehen wird in Kapitel 8.3.1 beschrieben

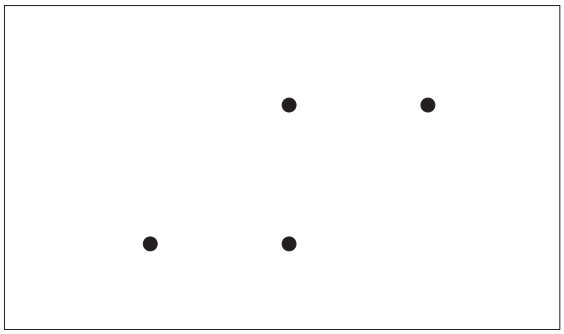


Abbildung 7.1: Anordnung der 4 Marker zum Master

Stelle davon aus, dass die vier dem Master zugehörigen Marker eindeutig identifiziert sind.

7.1.2 Projektion der Positionsvorhersage

Aus den aktuellen, subpixelgenauen Schwerpunktkoordinaten der vier Master-Marker und ihren Raumkoordinaten wird mithilfe des DLT-Algorithmus die Transformationsmatrix H berechnet, welche nun dazu dient, die Positionskoordinaten aller anderen Marker dieser Projektion zu unterziehen, um eine Positionsvorhersage im Videobild zu erlangen. Hierbei ist jedoch sicherzustellen, dass keine drei der vier Punkte kollinear sind. Eine genaue Beschreibung der Berechnung erfolgt in Kapitel 7.2 und Kapitel 8.3.2.

Bild 7.2 stellt eine typische Situation dar: Die im Bild vorhandenen Marker werden als solche erkannt und ihre Schwerpunkte werden berechnet (rote Kreuze). Die Master-Marker werden aufgrund ihrer Abstände als solche erkannt und identifiziert, dargestellt durch die gelben kreisförmigen Markierungen. Der Toleranzbereich für die Zugehörigkeit eines Markers zum Master wird durch die roten Abstandskreise visualisiert. Die erwartete Position der anderen Marker, welche durch die Projektion bestimmt wurden, sind durch die magentafarbenen Kreuze markiert.

7.1.3 Bewegungsvorhersage (Trend)

Sind die sichtbaren Marker eines Videoframes identifiziert, so werden ihre Schwerpunktkoordinaten in einer Tabelle festgehalten. Kann ein Marker über eine Sequenz von aufeinanderfolgenden Bildern hinweg identifiziert werden, so wird aus seinen Schwepunktkoordinaten eine Projektion für die erwartete Position im nächsten Frame berechnet. Diese erwartete Position kann wiederum zur Identifizierung des Markers dienen, da die einzelnen Marker über relativ große Abstände im Vergleich zu der Positionsdifferenz eines Markers zwischen zwei aufeinanerfolgenden Frames verfügen. Selbst bei raschen Kopfbewegungen bleibt dieser Abstand angesichts einer Bildwiederholrate von 30Hz - was einer Zeitdifferenz von 33ms zwischen zwei erfassten Bildern entspricht - deutlich unter dem Mindestabstand zweier Marker. Abb. 7.3 zeigt die Projektion der jeweils erfassten Marker aufgrund ihrer Bewegung über

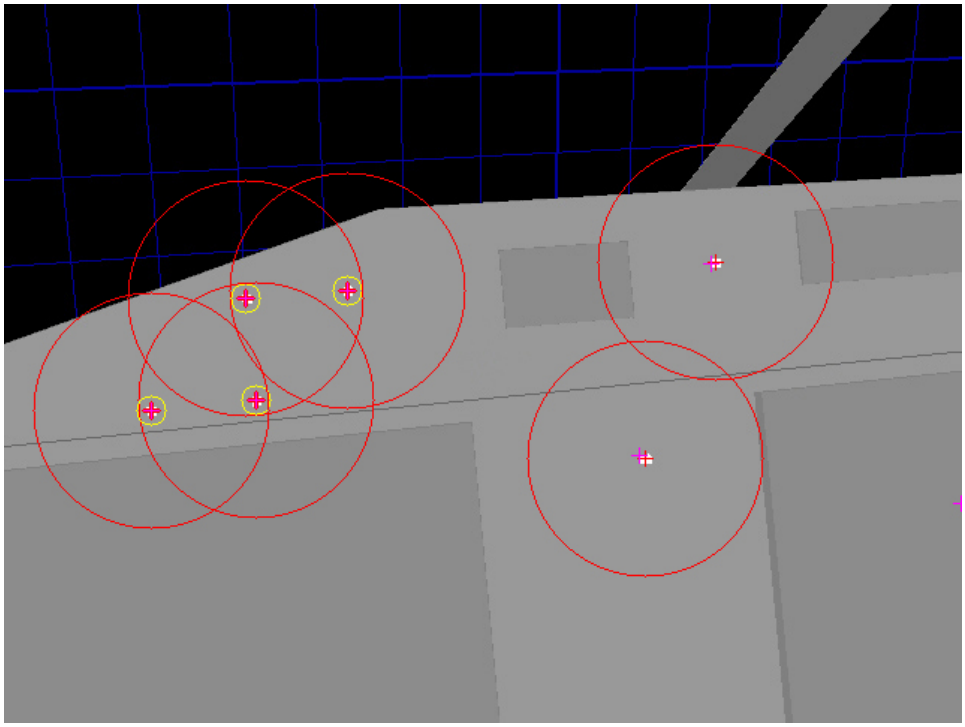


Abbildung 7.2: Master-Marker als Berechnungsgrundlage für die perspektivische Projektion

die letzten Frames. Man beachte, dass über den rechtesten Marker im Bild noch keine Bewegungsvorhersage verfügbar sind, da dieser im aktuellen Frame erstmals sichtbar geworden ist. Abb. 7.4 zeigt die selbe Szene ein Frame später. Die Blickrichtung ist weiter nach rechts geschwenkt, wodurch nun auch beim rechtesten Marker eine Trend-Projektion erfolgen kann. Unterschreitet im nächsten Frame der Abstand vom tatsächlich aufgenommenen Marker und seiner vorhergesagten Position einen festgelegten Grenzwert, so kann für diesen Marker eine Identifikation über die Bewegungsvorhersage getroffen werden, unabhängig von der Homographieberechnung aus Abschnitt 7.1.2.

7.1.4 Verdeckung von Markern

Wie in Kap. 3.1.4 gefordert, muss der Fall in Betracht gezogen werden, dass einzelne Marker zeitweise verdeckt werden. Dies kann beispielsweise beim Griff zum Fahrwerkshebel, bei der Bedienung des Autopiloten oder beim Drücken von Tasten der Multifunktions-Bildschirme der Fall sein. Sind also ein oder mehrere Marker verdeckt, so müssen zwei Fälle unterschieden werden, nämlich ob der Marker Bestandteil des Masters ist oder nicht.

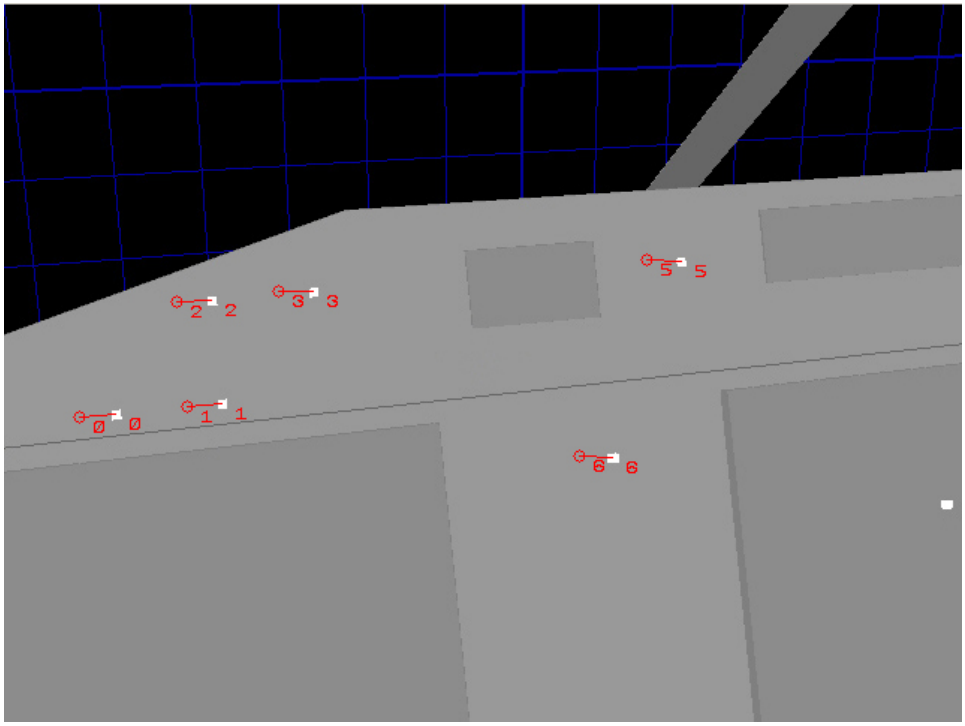


Abbildung 7.3: Trendbestimmung zum Zeitpunkt t

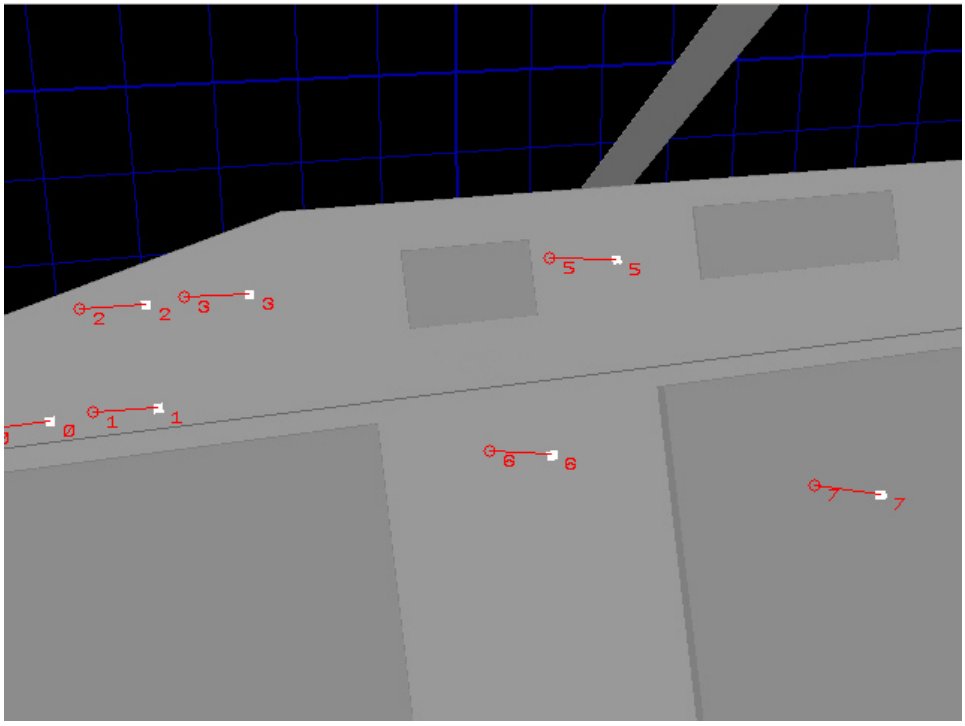


Abbildung 7.4: Trendbestimmung zum Zeitpunkt (t+1)

Verdeckung des Masters

Der Masters wurde im Cockpit am Autopilotenpannel platziert, um ihn weitgehend vom Verdeckungsbereich typischer Handbewegungen fernzuhalten. Dennoch muss der Fall der Verdeckung eines oder mehrerer Marker des Masters behandelt werden. Liegt eine Verdeckung des Masters vor, so kann dennoch davon ausgegangen werden, dass eine Teilmenge der verbleibenden sichtbaren Marker, egal ob Teil des Masters oder nicht, über die Trendverfolgung identifiziert werden können. Ist deren Anzahl größer oder gleich vier, so werden sie nun zur Berechnung der Homographie herangezogen. Kann dieses Kriterium nicht erfüllt werden, so ist eine Positionsberechnung erst wieder bei vollständiger Erfassung des Masters möglich. Konnten jedoch vier oder mehr Marker durch die Trendverfolgung eindeutig zugeordnet werden, so werden vier davon ausgewählt, um mit ihnen die Berechnung der Homographiematrix durchzuführen. Auch hier sei für eine genaue Beschreibung der Auswahl auf Kap. 8.3.2 verwiesen. Abb. 7.5 zeigt die Projektion der Marker, diesmal jedoch basierend auf vier bekannten Markern (gelbe Kreise), drei davon aber nicht dem Master zugehörig. Man beachte die hohe Genauigkeit der Vorhersage (magentafarbenen Kreuze) mit den tatsächlichen Schwerpunkten (rote Kreuze). Diese ist auf die großflächigere Verteilung der zur Homographieberechnung herangezogenen Korellationspunkte zurückzuführen.

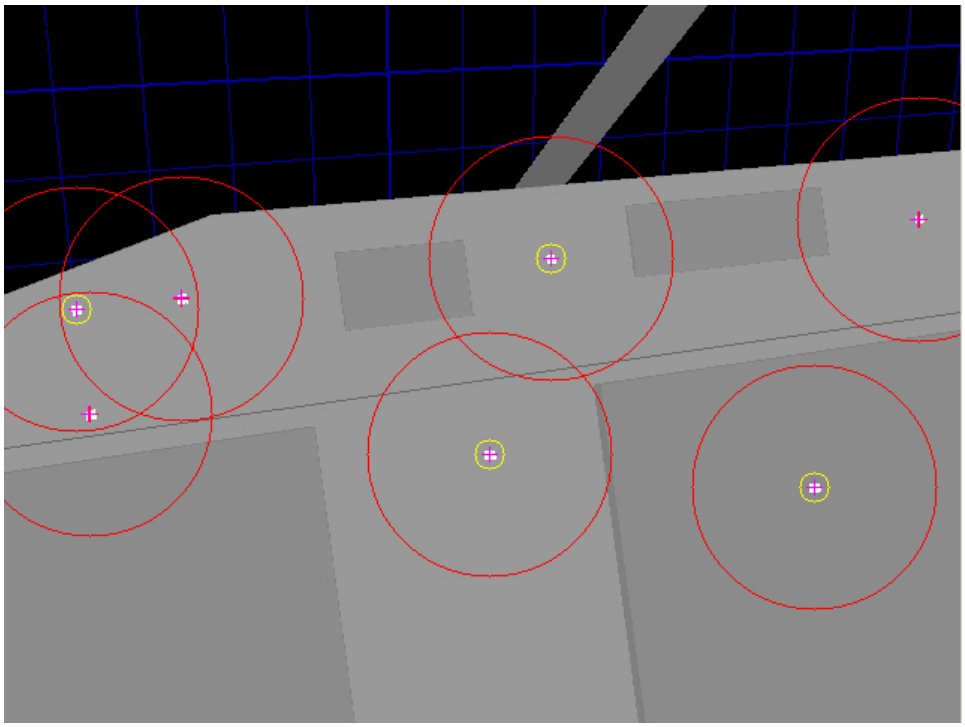


Abbildung 7.5: Alternative Berechnung der Homographie bei Verdeckung des Masters

Verdeckung eines Markers

Wird ein (oder mehrere) Marker verdeckt, welcher nicht Teil des Masters ist, so kann die Homographieberechnung ungehindert durchgeführt werden. Folglich werden die erwarteten Positionen aller nicht dem Master zugehörigen Marker projiziert. Da bei der Identifikation der tatsächlichen Marker jedoch stets von den projizierten Positionen ausgegangen wird, geschieht im Falle einer Verdeckung folgendes:

1. Der Algorithmus erwartet an einer bestimmten Position innerhalb eines definierten Radius einen Marker.
2. Durch die Verdeckung ist dieser jedoch nicht sichtbar.
3. Es wird keinem Marker die ID des verdeckten Marker zugewiesen.

Faktisch geschieht also bei der Verdeckung eines nicht-Mastermarkers dasselbe wie bei einem Marker, welcher sich außerhalb des Bildes befindet. Dieser Fall benötigt daher keine Sonderbehandlung.

7.2 Mathematischer Hintergrund zur Berechnung der Homographie

Das folgende Kapitel möge dem Leser die mathematischen Grundlagen nahebringen, welche zur Bestimmung der perspektivischen Projektion und somit zur Markeridentifikation notwendig sind. Schlüsselkonzepte hierbei sind homogende Koordinaten, projektive Transformationen sowie letztendlich die konkrete Berechnung mittels des Direkt Linear Transformation (DLT) Algorithmus. Da die folgende Betrachtung jedoch gezielt auf die Lösung des gegebenen Problems zielt, sei der interessierten Leser auf [1] für eine ausführliche Darstellung des Themengebietes verwiesen.

7.2.1 Homogene Koordinaten

In der projektiven Geometrie finden homogene Koordinaten oftmals Anwendung, da sich durch sie Berechnungen deutlich vereinfachen lassen. Neben vielen weiteren Vorteilen lässt sich beispielsweise eine euklidsche Transformation, welche klassischerweise aus einer Translation und einer Rotation besteht, nunmehr in homogener Darstellung lediglich durch eine Matrixmultiplikation ausdrücken.

$$\begin{pmatrix} x_{neu} \\ y_{neu} \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} \begin{pmatrix} x_{alt} \\ y_{alt} \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Um einen euklidschen 2d-Punkt (x/y) in homogene Darstellung zu transformieren, wird ihm als dritte Vektorkomponente eine 1 angehängt (x/y/1). Nun lässt sich obige Transformation mit einer einzigen Matrixmultiplikation ausführen:

$$\begin{pmatrix} x_{neu} \\ y_{neu} \\ w_{neu} \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{alt} \\ y_{alt} \\ w_{alt} \end{pmatrix}$$

Da der projektive Raum, im Gegensatz zum euklidschen, indifferent gegenüber Skalierung ist, kann es für ein und denselben Punkt mehrere Repräsentationen geben. Zur Rückkonvertierung eines homogenen Punktes in einen euklidschen ist sein Vektor so zu skalieren, dass seine w-Komponente wieder 1 wird.

7.2.2 Projektive geometrische Abbildungen

Mit Hilfe der homogenen Darstellung lassen sich nun projektive Abbildungen im zweidimensionalen Raum folgendermaßen darstellen:

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix}$$

Hierbei unterscheiden man verschiedene Klassen von Transformationen. In jeder Klasse gelten bestimmte Invarianten, d.h. Eigenschaften, welche unter den gegebenen Transformationen erhalten bleiben. Ein Beispiel hierfür wäre, dass bei euklidschen Transformationen, welche ja nur aus Verschiebungen und Rotationen bestehen können, sowohl Längen- als auch Flächenmaße invariant gegenüber der Projektion sind, was spätestens bei Betrachtung von Abb. 7.6 offensichtlich wird. Allgemein lässt sich feststellen, dass sich mit zunehmendem Freiheitsgrad der Transformation die Menge der verbleibenden Invarianten verkleinert.

Euklidsche Transformation, Isometrie

Wie bereits in der Einleitung angedeutet, beschreiben Isometrien[2] Transformationen, welche aus einer Translation und einer Rotation bestehen. Folgende Matrix repräsentiert eine euklidsche Transformation in der Ebene (zweidimensionaler Raum) in homogener Darstellung:

$$\begin{pmatrix} cos\theta & -sin\theta & t_x \\ sin\theta & cos\theta & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

oder etwas allgemeiner ausgedrückt:

$$\begin{pmatrix} r_{11} & r_{12} & t_{13} \\ r_{21} & r_{22} & t_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

wobei r für den Rotationsanteil und t für den translatorischen Anteil der Matrixelemente steht. Eine euklidsche Transformation hat drei Freiheitsgrade, einen für die Rotation und zwei für die Translation (x- und y-Richtung).

[2](griech.)iso: gleichförmig; (griech.)metrum: das Maß

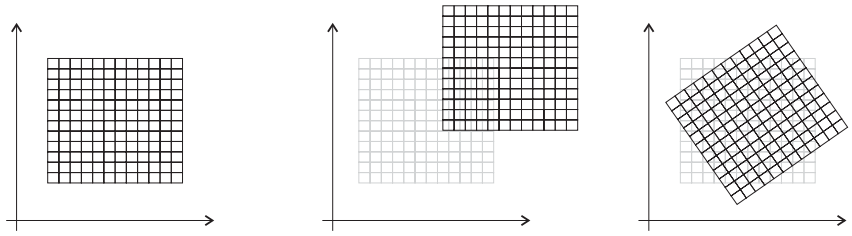


Abbildung 7.6: Euklidsche Transformation

Ähnlichkeitstransformation (Similatity)

Die Klasse der Ähnlichkeitstransformation erweitert die der Isometrie um eine isotrope[3] Skalierung. Mathematisch bedeutet dies, dass der Rotationsanteil in der Matrix um einen Skalarfaktor s erweitert wird. Somit bekommt diese Klasse einen weiteren Freiheitsgrad. Die homogene Darstellung lautet:

$$\begin{pmatrix} s \cdot cos\theta & -s \cdot sin\theta & t_x \\ s \cdot sin\theta & s \cdot cos\theta & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

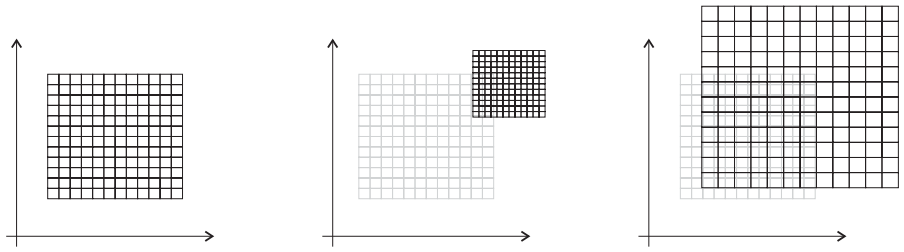


Abbildung 7.7: Similarity

Affine Transformation

Die nächsthöhere Klasse in der Hierarchie der projektiven Transformationen ist die der affinen Transformationen. Sie beschränkt sich nun nicht mehr auf isotrope Skalierung, sondern lässt eine freie Skalierung in beide Richtungen zu.

$$\begin{pmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

[3] isotrop: gleichförmig

Jede affine Transformationsmatrix kann als Konkatenation folgender Transformationen angesehen werden:

1. Rotation um ϕ
2. Skalierung um λ_1
3. Skalierung um λ_2
4. Rückrotation um $-\phi$
5. Rotation um θ

Die Besonderheit der affinen Transformation ist die Skalierung in orthogonale Richtungen unter einem bestimmten Drehwinkel. Durch die anisotrope (ungleichförmige) Skalierung gehen Invarianten wie Längenverhältnisse oder Winkeltreue verloren.

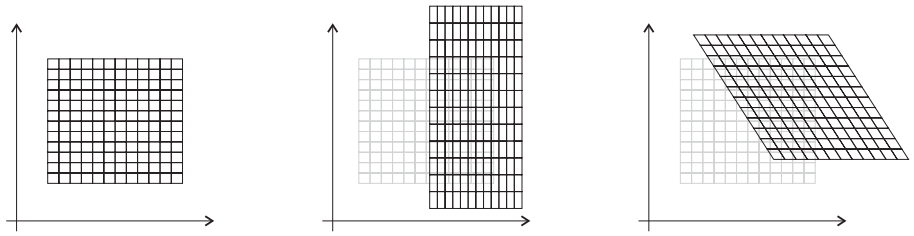


Abbildung 7.8: Affine Transformation

Parallelität beispielsweise bleibt jedoch auch unter affinen Transformation erhalten. Affine Transformationen haben sechs Freiheitsgrade entsprechend den besetzten Einträgen in der Transformationsmatrix.

Projektive Transformation

Die allgemeine, nicht singuläre, lineare Transformation homogener Koordinaten bietet die größte Freiheit und erlaubt es, eine perspektivische Darstellung mathematisch zu beschreiben. Da in der homogenen Darstellung, wie bereits in 7.2.1 erwähnt, nicht die Skalierung, wohl aber die Verhältnisse eine Rolle spielen, wird die Transformationsmatrix effektiv durch acht Parameter bestimmt, was folglich zu einem Freiheitsgrad von acht führt.

$$\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}$$

Eine projektive Transformation zwischen zwei Ebenen kann durch eine Korrespondenz von jeweils vier Punkten berechnet werden, von denen sichergestellt sein muss, dass jeweils keine drei colinear liegen. Diese Eigenschaften können also herangezogen werden, um ein perspektivisch verzerrtes Bild, durch Markierung mit 4 (nicht-colinearen) Punkten und dem Wissen über ihre wahre Lage, in ein verzerrungsfreies Bild zurück zu transformieren.

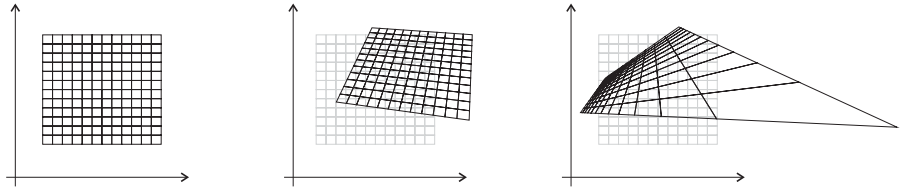


Abbildung 7.9: Projektive Transformation

7.3 Der Direct Linear Transformation (DLT-) Algorithmus

Der Direct Linear Transformation Algorithmus erlaubt es uns, aus einer Korrespondenz von jeweils 4 Punkten im perspektivisch verzerrten und unverzerrten 2d-Bild die Transformationsmatrix H zu berechnen. Diese, von der Matrix H beschriebene, projektive Transformation nennt man Homographie.

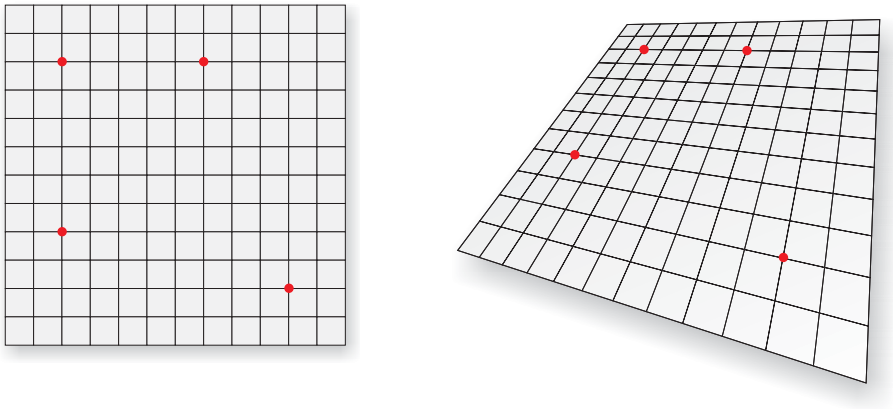


Abbildung 7.10: Berechnung der Transformationsmatrix aus 4 Korellationspunkten

Ausgangspunkt für die Berechnung sind jeweils 4 Punkte in homogener Darstellung:

$$x_i \leftrightarrow x_i' \text{ bzw. } \begin{pmatrix} x_i \\ y_i \\ w_i \end{pmatrix} \leftrightarrow \begin{pmatrix} x_i' \\ y_i' \\ w_i' \end{pmatrix} \text{ für } 0 \leq i \leq 3$$

Die Transformation wird durch folgende Gleichung beschrieben:

$$x_i' = H \cdot x_i$$

Da es sich bei der Gleichung um homogene Vektoren handelt, wird durch sie lediglich die Gleichheit der Richtung zum Außdruck gebracht, die Vektoren selbst können sich jedoch durch einen nichtnegativen Skalierungsfaktor in ihrem Betrag unterscheiden. Dennoch gilt Aufgrund der Übereinstimmung in ihrer Richtung, dass ihr Kreuzprodukt null ist.

$$x_i' \times Hx_i = 0$$

Die Matrix H kann folgendermaßen ausgedrückt werden:

$$Hx_i = \begin{pmatrix} h^{1\top} x_i \\ h^{2\top} x_i \\ h^{3\top} x_i \end{pmatrix}$$

$h^{j\top}$ bezeichnet hierbei den j-ten Zeilenvektor der Matrix H. Schreibt man den Vektor x_i' in transponierter Form als (x_i', y_i', w_i'), so lässt sich das Kreuzprodukt explizit angeben.

$$x_i' \times Hx_i = \begin{pmatrix} y_i' h^{3\top} x_i - w_i' h^{2\top} x_i \\ w_i' h^{1\top} x_i - x_i' h^{3\top} x_i \\ x_i' h^{2\top} x_i - y_i' h^{1\top} x_i \end{pmatrix}$$

Da aber $h^{j\top} x_i$ gleich $x_i^\top h^j$ für $j = 1 \ldots 3$ ist, gilt:

$$\begin{pmatrix} 0^\top & -w_i' x_i^\top & y_i' x_i^\top \\ w_i' x_i^\top & 0^\top & -x_i' x_i^\top \\ -y_i' x_i^\top & x_i' x_i^\top & 0^\top \end{pmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0$$

Diese Gleichungen sind der Form $A_i h = 0$, wobei A eine 3x9 Matrix ist und h ein 9-Vektor, bestehend aus den Matrixeinträgen der Homographiematrix. Von diesen Gleichungen sind jedoch nur 2 linear unabhängig, die dritte Zeile ist bis auf einen Skalierungsfaktor eine Linearkombination der ersten beiden. Daher erhält man für jede Korrespondenz von Punkten zwei Gleichungen als Einträge von H. Durch die vier Korrespondenzpunkte erhält man folglich vier 2x9 Matrizen A_i welche zu einer 8x9 Matrix zusammengesetzt werden.

7.3.1 Eigenwertdekomposition zur Berechnung der Transformationsmatrix

Nun wird auf der 8x9 Matrix A eine Eigenwertzerlegung[4] durchgeführt.

$$A = UDV^\top$$

Hierbei hat U die Dimension 8x8, D ist eine Diagonalmatrix mit den Eigenwerten in absteigender Reihenfolge auf der Diagonalen und V hat die Dimension 9x9. Der Einheits-Eigenvektor, welcher dem kleinsten Eigenwert zugehört, bildet die Lösung von h und kann aus dem letzten Spaltenvektor

[4]SVD: Singular Value Decomposition

von V abgelesen werden. Entsprechen der obigen Definition von H werden diese Werte zur Matrix H zusammemgefügt.

$$h = \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{pmatrix}, H = \begin{pmatrix} h^{1\top} \\ h^{2\top} \\ h^{3\top} \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}$$

Nun können für die berechnete Transformation zwischen Urbild und Bild beliebige Vektoren durch Multiplikation mit der Homographie-Matrix H berechnet werden. Da die Bildkoordinaten jedoch euklidsche Koordinaten sind, muss nach ausgeführter Transformation der Ergebnisvektor noch normiert werden. Dies erreicht man, indem er mit einem Skalar mit dem Wert 1/w multiplitziert wird. Da nun die w-Komponente gleich eins ist, entsprechen die x-/y-Werte den euklidschen Koordinaten des Bildpunktes.

Kapitel 8

Komponenten-Design und deren Umsetzung

War das Ziel des vorangegangenen Kapitels, das theoretische Vorgehen zur Markeridentifikation zu erläutern, so soll in diesem Kapitel die konkrete Umsetzung und Lösung der Einzelprobleme beschrieben werden. Zur Einleitung jedes Abschnitts wird die spezielle Problemstellung erläutert. Daraufhin wird die implementierte Lösung und ihre Funktionsweise vorgestellt. Zum Ende werden die aufgetauchten Probleme beschrieben und soweit möglich Ansätze für deren Lösung angegeben.

8.1 Bilderfassung (Image Acquisition)

Nach Installation der mitgelieferten Kameratreiber von Point Grey (Unified PGR Camera 1394 Driver) in der Version 1.4.3.27 wird die Firefly2 Kamera vom Betriebssystem korrekt erkannt. Der PGR FlyCapture SDK[1] stellt eine Bibliothek bereit, mit welcher die Kamera direkt unter C/C++ angesprochen werden kann. Hierzu wird zuerst überprüft, ob eine Kamera angeschlossen ist, welche daraufhin initialisiert wird. Dabei wird das Bildformat spezifiziert, mit welchem die Bilderfassung erfolgen soll. Diese ist in in unserem Fall eine Auflösung von 640x480 Pixel, ein Farbkanal (Graustufen) bei einer Wiederholrate von 30Hz.

```
flycaptureStart( context, FLYCAPTURE_VIDEOMODE_640x480Y8,
FLYCAPTURE_FRAMERATE_30 );
```

Der Datenstrom, welcher nach der Initialisierung fortlaufend von der Kamera übermittelt wird, wird zyklisch in vier Speicherpuffer geschrieben. Folgender Befehl kopiert das aktuellste vollständige Frame in das in der API[2] spezifizierte Dateiformat `FlyCaptureImage`:

```
flycaptureGrabImage2( context, &image );
```

[1] Software Development Kit
[2] Application Program Interface

Da zur weiteren Bildverarbeitung auf die Funktionalität der OpenCV Bibliothek zurückgegriffen werden soll, ist es notwendig, die Bilddaten in das OpenCV-bibliothekseigene IPL-Image Format zu konvertieren. Zu diesem Zweck wird ein neues IPL-Bild mit entsprechenden Parametern und Headerangaben erzeugt (640x480 Pixel, 1 Kanal) und der Speicherbereich des Rohdatenblocks des `FlyCaptureImage`-Formats kopiert.

8.2 Finden der Marker

Die Aufgabe ist nun, die Lichtpunkte im IR-gefilterten Kameraframe zu finden und deren Schwerpunkte zu berechnen. Hierzu wird zuerst eine Schwellenwertberechnung ausgeführt (Thresholding), deren Resultat ein Binärbild ist.

8.2.1 Filterung

Aufgrund der bereits erfolgten Infrarotfilterung unterliegt der Helligkeitswert des Hintergrunds praktisch keinen Schwankungen, ob nun bei Helligkeit oder Dunkelheit gefilmt wird. Deshalb ist eine dynamische Anpassung des Schwellenwertes nicht notwendig. Tests haben ergeben, dass bei einem Schwellenwert von 170[3] alle Artefakte des Rauschens beseitigt werden.

Abbildung 8.1: Histogramm eines Infrarot-Kameraframes

8.2.2 Schwerpunktberechnung

Wir betrachten nun eine Pixelmatrix, deren Hintergrund schwarz (0) und der Marker weiß (255) ist. Der Koordinatenursprung bei allen nun folgenden Berechnungen auf der Pixelmatrix liegt in der oberen linken Ecke. Die X-Achse verläuft horizontal, die Y-Achse vertikal. Der Schwerpunkt einer Bildregion (auch erstes Moment der Region genannt) kann durch folgende Formel berechnet werden.

$$X = \frac{\sum\limits_{all(x,y)\in P} P(x,y) \cdot x}{\sum\limits_{all(x,y)\in P} P(x,y)}$$

[3]Ein Helligkeits-/Graustufenwert von 0 entspricht schwarz, 255 entspricht weiß.

Folgende Annahmen werden hierbei vorausgesetzt:

- P bezeichnet das gesamte Bild.
- P(x,y) bezeichnet den Farbwert des Pixels mit den Koordinaten (x,y).
- Da es sich bereits um ein binärisiertes Bild handelt, gilt: $P(x, y) \in [0, 1]$.
- Ist der Pixel P(x,y) Element der Region, so ist $P(x, y) = 1$.
- Ist der Pixel P(x,y) nicht Element der Region, so ist $P(x, y) = 0$.

Als Resultat solch einer Berechnung erhält man einen subpixelgenauen X-/Y-Wert des Schwerpunktes. Dies ist zudem eine Anforderung des Algorithmus von Tsai zur Pose Estimation. Das Problem besteht nun darin, die einzelnen Marker im Binärbild als solche zu finden, zu unterscheiden und für jeden einzelnen den Schwerpunkt zu ermitteln. Aufgrund der Installation im Cockpit darf von der Annahme ausgegangen werden, dass keine zwei Marker so nahe beieinander liegen, dass die sie umschließenden rechtwinkligen Boxen sich überschneiden würden. Zusätzlich wird jeder gefilmte und gefilterte Marker auf genau eine Zusammenhangskomponente abgebildet. Da die Kamerabilder non-interlaced sind, besteht nicht die Gefahr, dass bei schnellen Bewegungen längs der X-Achse des Bildes durch Interlacing-Effekte der Marker in mehrere Zusammenhangskomponenten „zerrissen“ wird.

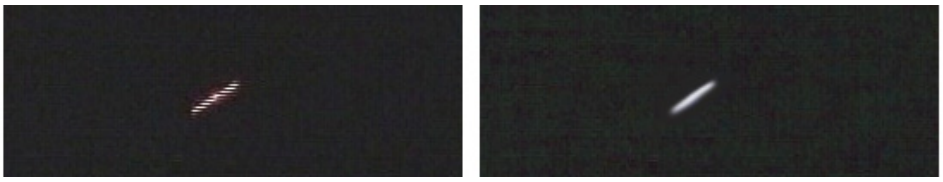


Abbildung 8.2: Verwischen der Marker bei sehr schneller Kamerabewegung a) interlaced b) non-interlaced

Folgendes Vorgehen ermittelt nun eine Liste aller im Bild vorhandener Marker und deren X-/Y-Koordinate des jeweiligen Schwerpunktes. Das Bild wird zeilenweise durchlaufen; wird ein weißer Pixel gefunden, so wird in ihm der von der OpenCV Bibliothek implementierter Befehl `floodfill`[39] zum Finden der 4-fach Zusammenhangskomponente gestartet. Das Floodfilling betrachtet nun jeweils vom aktuellen Pixel den nördlichen, östlichen, südlichen und westlichen Nachbarpixel. Ist dieser ebenfalls weiß, so gehört er zur selben Komponente und wird dementsprechend markiert. Dies wird für alle markierten Pixel wiederholt.

Der `floodfill`-Befehl ermittelt darüber hinaus eine „Bounding-Box“, d.h. eine die gefundenen Zusammenhangskomponente vollständig, aber minimal umschließende rechtwinklige Box. Diese wird nun verwendet, um in ihr den Algorithmus zur Schwerpunktsberechnung zu starten. Da keine anderen Marker außer dem betrachteten aufgrund der oben getroffenen Annahme in dem Bereich der Box liegt, liefert der Algorithmus das korrekte Ergebnis.

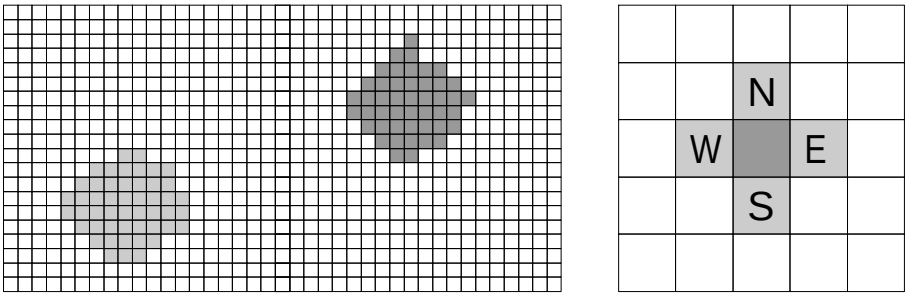


Abbildung 8.3: a) Markierung von 4-fach-Zusammenhangskomponenten b) Betrachtete Nachbarpixel bei der Suche nach 4fachen Zusammenhangskomponenten (Nord, Ost, Süd, West)

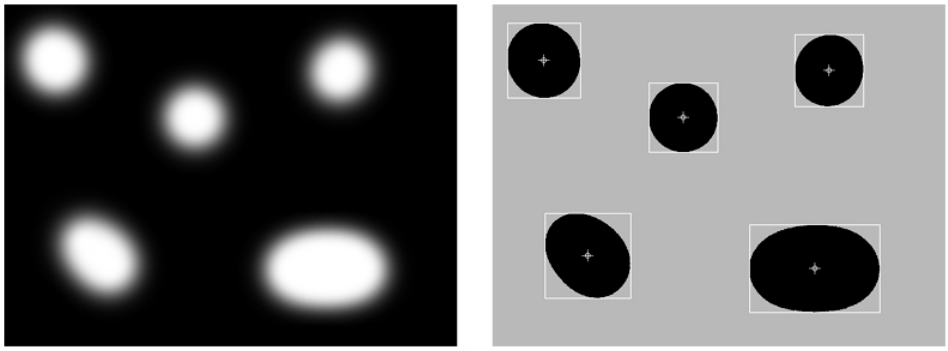


Abbildung 8.4: a) Vor der Schwellenwertberechnung b) Nach der Schwerpunktberechnung mit Bounding-Boxen

Trotz der getroffenen Einschränkungen gibt es dennoch einen Problemfall, welcher zu einem falschen Ergebnis des Schwerpunktes führt. Ist ein Marker so nahe am Bildrand, dass ein Teil nicht mehr erfasst wird, so bezieht sich der berechnete Schwerpunkt nur auf die noch im Bild vorhandenen Pixel und entspricht so also nicht mehr dem wahren Schwerpunkt des Markers. Dieser Fall kann jedoch vermieden werden, indem man einen Mindestabstand zwischen der Boundary-Box und dem Bildrand von 1 Pixel fordert. Dadurch würde der Erfassungsraum von 640x480 Pixel entsprechend um ca. einen halben Markerdurchmesser an jeder Seite beschnitten werden.

8.3 Identifikation der Marker

Nach den bisher durchgeführten Operationen verfügen wir also über eine Liste von Schwerpunktkoordinaten. Bekannt ist jedoch die 3d-Position aller Marker im Cockpit. Das Problem besteht darin, herauszufinden, welchen Ausschnitt von der Gesamtmenge der Marker die Kamera gerade erfasst und die bis jetzt ununterscheidbaren IR-Dioden eindeutig zu unterscheiden. Das hierzu angewandte Verfahren macht sich einige der Vorteile zunutze, welche die Einschränkungen durch das Cockpit mit sich bringen. Wie bereits in Kap 3.1.3 beschrieben, befinden sich alle Marker auf dem Instrumentenbrett bzw. auf den seitlichen unteren Fensterrahmen. Sie sind also weitgehend coplanar angeordnet. Zusätzlich befindet sich der Pilot (im Rahmen seiner Bewegungsfreiheit) stets am selben Ort. Es müssen also keine Sonderfälle berücksichtigt werden, wie z.B. dass die gesamte Szene von hinten gesehen wird oder auf dem Kopf steht. Vielmehr sollen gerade solche Einschränkungen, z.B. dass oben und unten bzw. links und rechts feste Größen sind, ausgenutzt werden.

8.3.1 Master-Marker

Wie in Kapitel 7.1.1 beschrieben, basiert die Identifikation der Marker auf dem Auffinden des Masters und der Berechnung einer projektiven Transformationsmatrix. Doch wie wird der Marker als solcher identifiziert? Gesucht ist nun ein möglichst einfaches und robustes Verfahren, den Master schnell aufzufinden. Zwei Kriterien spielen hierbei eine entscheidende Rolle. Hierzu werden zwei Annahmen getroffen, welche unter normalen Bedingungen als realistisch angesehen werden können:

1. Der Kopf des Piloten befindet sich innerhalb eines wohldefinierten Bereiches (siehe Kap. 3).

2. Der Kopf wird keinen Rollwinkel von über 30° einnehmen[4].

Geht man nun von einer Positionierung des Master-Markers unmittelbar vor dem Piloten am Autopilotenpannel aus (die Begründung dieser Position erfolgte bereits in Abschnitt 7.1.4), so kann unter Berücksichtigung obiger Annahmen davon ausgegangen werden, dass die Sichtlinie der Kamera den

[4]Rollwinkel beteutet in diesem Zusammenhang anschaulich, das Ohr auf der Schulter ablegen

Master-Marker stets in einem Winkel nahe 90° erfasst. Das bedeutet anschaulich, dass man niemals sehr schräg auf den Master blickt, wodurch die kürzesten Abstände zwischen den Master-Markern auch bei leichter perspektivischer Betrachtung bis auf eine Toleranz von unter 15% als äquidistant angesehen werden können.

Mit Berücksichtigung der zweiten Einschränkung und dem Wissen, dass der Abstand zweier zum Master gehöriger Marker viel kleiner ist als derjenige jeglicher anderer Marker, lassen sich mit folgendem algorithmischen Vorgehen die vier Master-Marker eindeutig finden. Diese sind aufgrund ihrer Position N(orth), E(ast), S(outh) und W(est) benannt.

- Finde den kürzesten Abstand zweier Marker im Bild und nenne diesen d_{min}.
- Markiere alle Marker, die in einem Radius von $d_{min} + 15\%$ einen anderen Marker haben.
- Ist die Anzahl markierter Marker gleich 4, so liegt ein Master vor.
- Ist die Anzahl kleiner vier, so ist anzunehmen, dass der Master nur teilweise im Bild liegt oder partiell verdeckt wird.

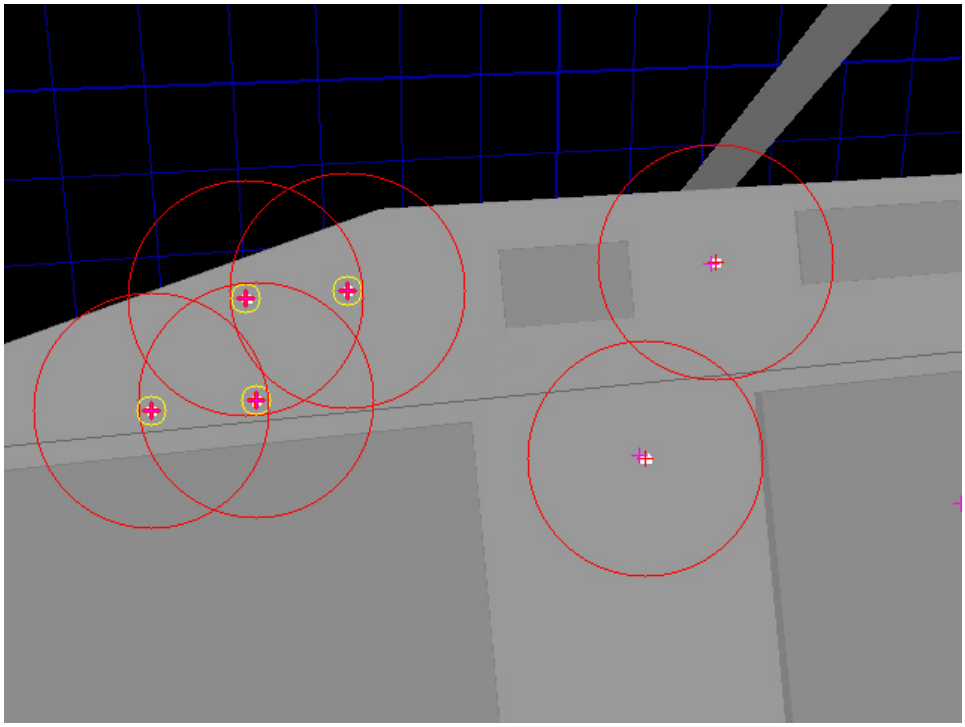


Abbildung 8.5: Auffinden der Master-Marker durch Suche nach minimalen Abstand (+ Toleranz)

Obwohl diese Methode der Master-Marker Markierung in der Praxis sehr zuverlässig funktioniert hat, sei dem aufmerksamen Leser dennoch nicht die

Tatsache vorenthalten, dass unter ungünstigen und sehr seltenen Bedingungen (kein Master auch nur teilweise sichtbar) vier zufällig äquidistante Marker fälschlicherweise als Master erkannt werden. Dieser Fehler könnte jedoch spätestens durch Rückprojektion der (korrumpierten) Daten der Positionsbestimmung aufgedeckt werden.

Ein zweiter Ansatz zur Bestimmung von d_{min}, welcher dieses Fehlerpotential nicht in sich birgt, ist die Berechnung von d_{min} aus einem vielfachen eines Markerdurchmessers. Dieser wächst bei Annäherung der Kamera proportional zum Abstand zwischen den Master-Markern. Hierbei treten jedoch andersartige Probleme auf. So deformiert sich ein binärisierter Marker bei schneller Bewegung der Kamera zu einer ellipsenähnlichen Form (Abb. 8.2), welche keine Rückschlüsse mehr auf den unbewegten Durchmesser zulässt.

Sind nun die vier dem Master zugehörigen Marker als solche erkannt, wird aufgrund ihrer X-Koordinate der Linkeste und Rechteste als West bzw. East identifiziert. Das dies korrekt ist, belegt Abb. 8.6, gestützt auf die Annahme des maximalen Kopf-Rollwinkels. Der verbleibende Marker mit der kleineren Y-Komponente wird als North, der mit der größeren Y-Komponente als South identifiziert.

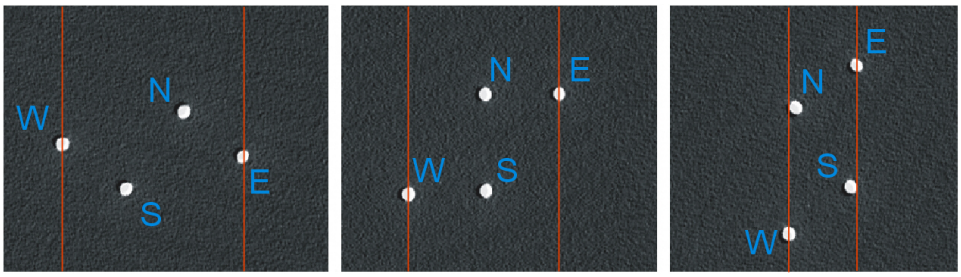


Abbildung 8.6: X-Koordinaten des E- und W-Markers unter zulässiger Verdrehung

Die gewählte Form des Masters liegt jedoch nicht nur in der Anwendbarkeit dieses einfachen Verfahrens, welches auf eine quadratische Formgebung aufgrund von Doppeldeutigkeiten nicht anwendbar ist, begründet. Um eine numerisch gute Homographiebestimmung zu erreichen, sollten die Korellationspunkte entsprechend gut über das Bild verteilt sein. Dies ist jedoch kontradiktorisch zum Masterdesign mit minimalen Abständen, da dieser in seiner Ausdehnung nur einen Bruchteil des Gesamtbildes abdeckt. Gerade deshalb wurde seine Dimensionierung im Verhältniss 2:1 (Breite : Höhe) gewählt, um der des Videobildes (640:480) ansatzweise zu entsprechen.

Dennoch ergibt sich hier ein Interessenkonflikt, dessen Lösung es in einem ausgewogenen Kompromiss zu suchen galt.

Folgende Punkte mögen dem Leser diese Schwierigkeiten vermitteln:

- Großflächiger Master → wenige Marker pro Bild
- Wenige Marker pro Bild → schlechte Positionsdaten (mind. 7 Marker erforderlich)

- Kleinflächiger Marker → ungenaue Homographie → Fehler bei der Markeridentifikation
- Verwendung eines größeren Weitwinkelobjektivs → Erfassung mehrerer Marker, aber → stärkere Linsenverzerrung (größere Ungenauigkeit am Bildrand)

In der Praxis besteht dieser Kompromiss nun aus einer Markerzahl von 10-15 pro Bild.

Wie in Abb. 6.7 gezeigt wurde, kann der gesamte Blickbereich des Piloten mit vier Kamera-Frames abgedeckt werden. In dieser Arbeit konnte jedoch aus Zeitgründen nur das Hauptpannel mit Markern ausgestattet werden.

8.3.2 Homographiebestimmung und Projektion

Das Prinzip, auf welchem die Projektion der Marker beruht, wurde bereits in Kap. 7 vorgestellt. An dieser Stelle soll die Wahl der zur Homographie herangezogenen Marker beschrieben werden sowie einige Probleme aufgezeigt werden, welche bei der Implementierung aufgetreten sind. Ist ein Master vollständig im Bild erfasst und identifiziert, so werden seine Marker (N, E, S, W) zur Berechnung der Homographie-Matrix verwendet. Wie in Abschnitt 8.3.1 beschrieben, wäre jedoch eine größere Streuung der Korellationspunkte über das Bild wünschenswert. Tritt also der Fall ein, dass der Master nur unvollständig im Bild oder verdeckt ist, so muss - wie in Abschnitt 7.1.4 beschrieben - eine Auswahl aus den über die Trendverfolgung (Abschnitt 7.1.3) identifizierten Markern getroffen werden, um diese als Korellationspunkte zu benutzen. Hierzu wird jeweils der nördlichste, östlichste, südlichste und westlichste Marker mit bekannter ID ausgewählt. Das Verfahren hierzu ähnelt dem aus Abschnitt 8.3.1. Durch die nun erlangte größer Streuung reduziert sich die numerische Instabilität bei der Projektion der Punkte deutlich.

Markerprojektion auf eine Ebene

Alle bisher angestellten Betrachtungen bezogen sich ausschließlich auf die Verteilung der Marker in einer Ebene. Mag dies für viele Cockpittypen (wie in Abb. 3.1) zutreffend sein, so sind im Falle des LFM Simulators die Marker auf dem Hauptpannel auf zwei Ebenen verteilt, die eine bildet das Autopilotenpannel, die andere das Hauptpannel mit dem Bildschirmen.

Aufgrund des relativ geringen Abstandes der zwei Ebenen (10cm) und ihrer Parallelität wurde die Autopilotenebene als Hauptebene definiert (auf ihr liegt der Master), und alle Punkte der Ebene des Hauptpannels auf sie zu projizieren.

Projektionszentrum hierbei ist die ideale Kopfposition des Piloten; die maximale Verschiebung des projizierten Punktes bei Translation des Kopfes wird durch den begrenzten Bewegungsspielraum limitiert. Dieser Unsicherheitsbereich hängt maßgeblich von folgenden Größen ab:

1. Der Größe des Bewegungsspielraums des Projektionszentrums

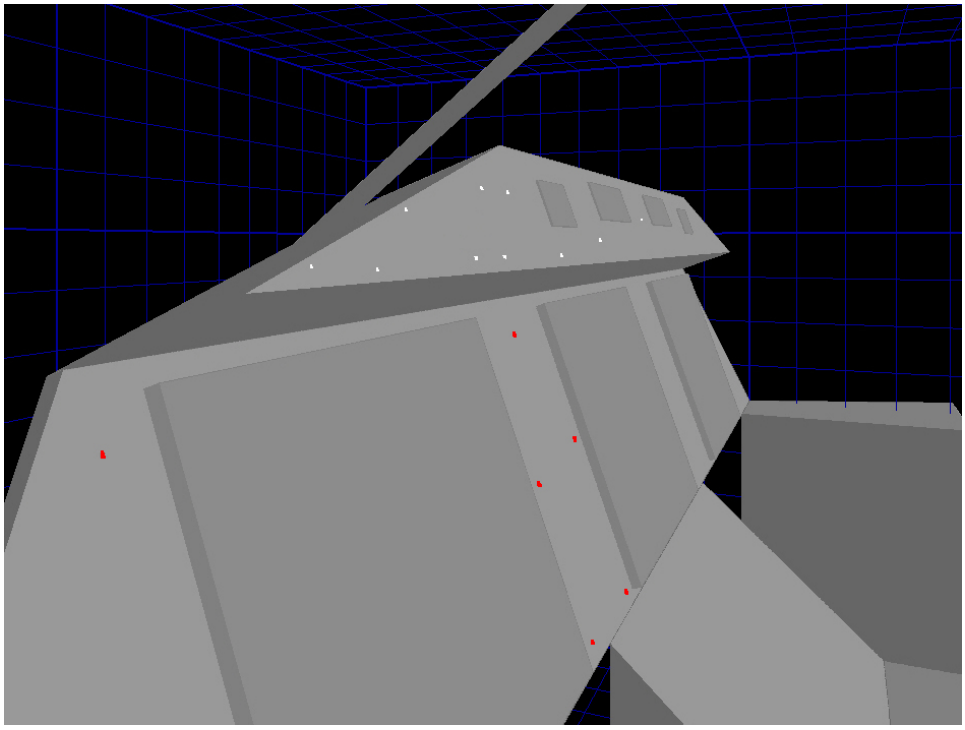


Abbildung 8.7: Verteilung der Marker auf 2 parallelen Ebenen (weiß: Autopilotenpannel, rot Hauptpannel)

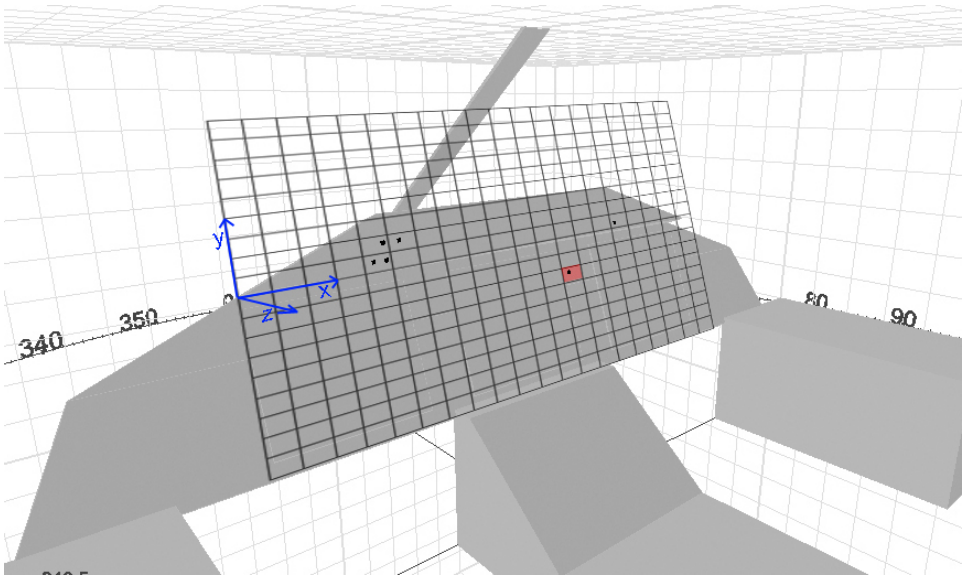


Abbildung 8.8: Hauptebene und Koordinaten-Referenzsystem im Cockpit

2. Dem Abstand der zwei parallelen Ebenen
3. Dem Verhältnis der Abstände Projektionszentrum - Projektionsebene und Projektionsebene - Ebene der Marker

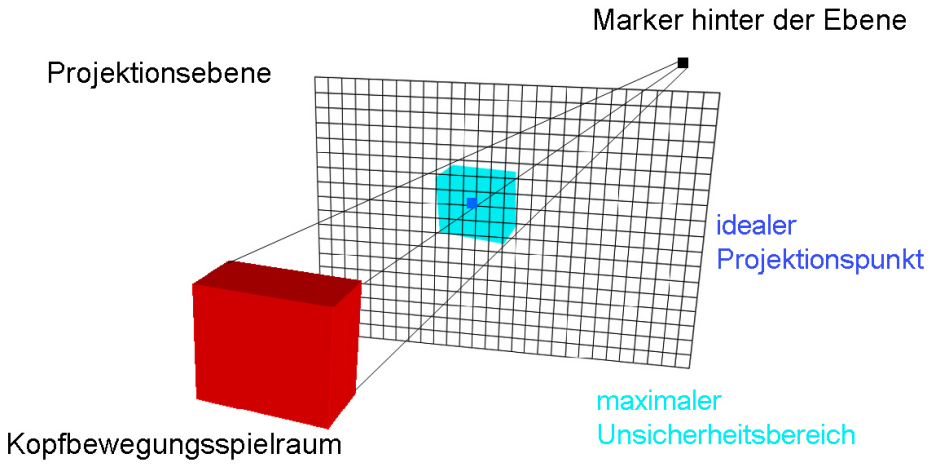


Abbildung 8.9: Projektion eines Markers auf dem Hauptpannel in die Ebene des Autopilotenpannels

4. Dem Blickwinkel

Unter realistischen Annahmen ergeben sich im LFM-Cockpit Unsicherheitsbereiche, welche in ihren Ausmaßen unter den Minimalabständen des Master-Markers liegen und somit vertretbar sind.

Globale Markertafel

Wurde bisher stets von „Wissen über die 3d-Position der Marker“ gesprochen, so soll dies nun konkretisiert werden. So wurde jeder Marker in einem global festgelegten rechtsorientierten Koordinatensystem erfasst und ihm eine eindeutige Identifikation zugewiesen. Der Ursprung des Koordinatensystems wurde auf die äußerste linke Ecke des Autopilotenpannels gelegt, die X-Achse verläuft längs der Unterkante nach rechts, die Y-Achse parallel zur Oberfläche nach oben und die Z-Achse bildet eine Normale auf die Frontfläche des Pannels. Zusätzlich zur wahren 3d-Position jedes Markers wurde für diejenigen, welche nicht in der Ebene des Autopilotenpannels liegen, ihr Projektionspunkt in der Hauptebene (mit der idealen Kopfposition des Piloten als Projektionszentrum) registriert. Diese Daten werden für die Vorhersage der erwarteten Position im Bild aufgrund der Homographie benötigt. Abb. 8.10 gibt einen Überblick über die Marker mit ihren Identifikationskennungen und ihren Koordinaten.

8.3.3 History-Table / Positionsverfolgung

Für die Positionsverfolgung jedes sichtbaren Markers im Bild wurde folgende Methode implementiert: Ist ein Marker im Bild sichtbar, so werden seine Bildkoordinaten aufgezeichnet. Wird derselbe Marker im nächsten Frame wiederum erfasst und identifiziert, so wird seine erwartete Position für das nächste Bild durch lineare Interpolation gewonnen. Weicht nun die tatsächliche Position im nächsten Bild von dieser ab, so fließt diese Korrektur in

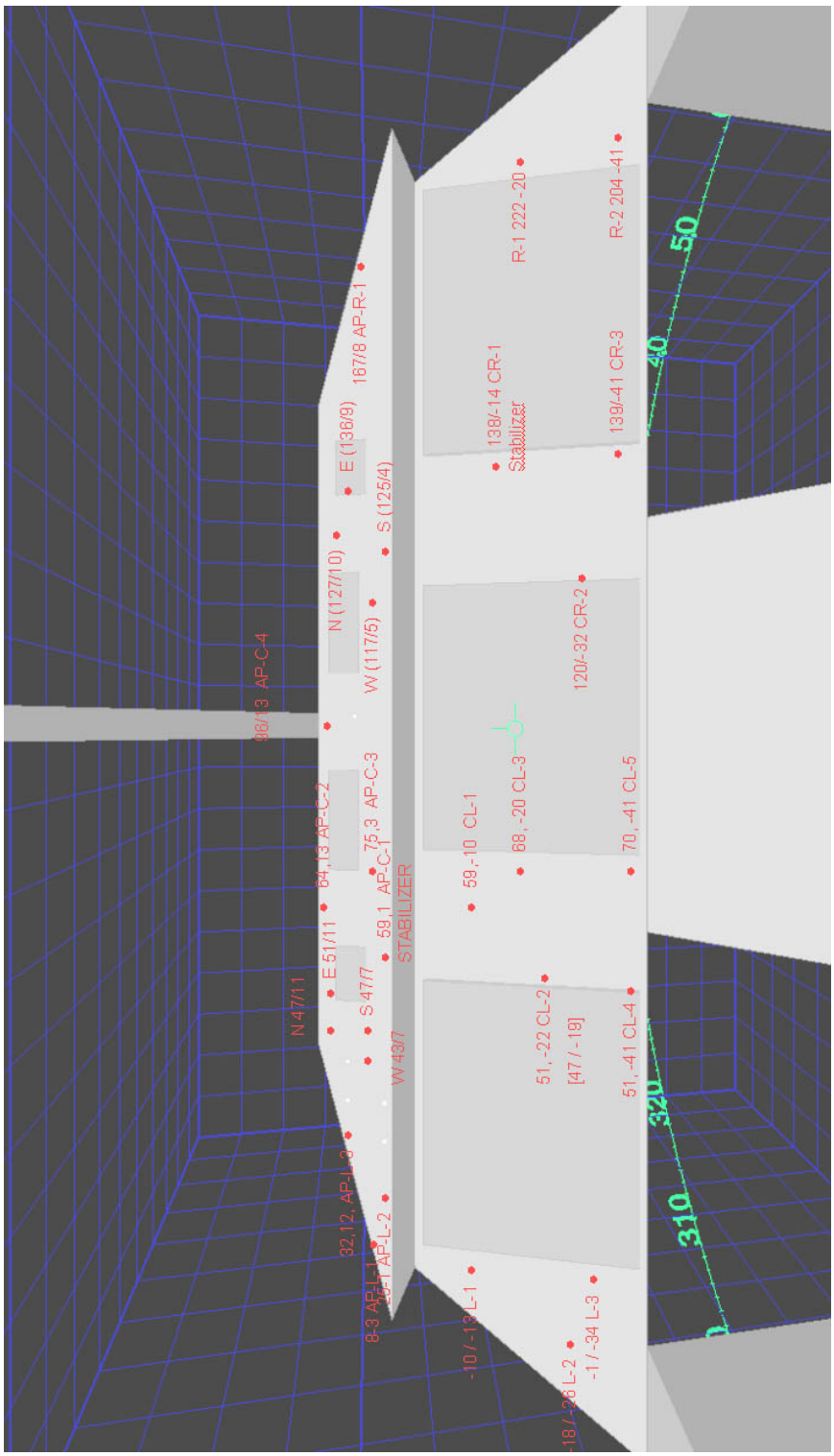


Abbildung 8.10: Registrierte Positionen der Marker

die Vorhersage für das nächste Frame mit ein. Wenngleich dieses Verfahren sehr simpel ist, funktioniert es insbesondere in Fällen, in denen die Kopfbewegungen minimal sind, sowie bei gleichmäßigen Bewegungen sehr gut. Wie in Kap. 3 beschrieben beschränken sich Kopfbewegungen beim Landeanflug im Tunnel auf ein Minimum.

Probleme mit der Positionsverfolgung existieren hingegen bei ruckartigen Bewegungen. Gerade im Bereich des Masters, wo die Marker die geringsten Abstände zueinander aufweisen, können abrupte Bewegungen zu Fehlern führen, insbesondere wenn die Bewegungsrichtung parallel der Verbindungslinie zweier eng benachbarter Marker ist.

Das eben beschriebene Verfahren zeichnet sich besonders durch seine geringe Rechenintensität aus, was es sehr schnell macht. Es existieren im Bereich der Bildverarbeitung jedoch eine Reihe anderer, komplexere Algorithmen, welche die Problemstellung der Punktverfolgung über eine Sequenz von Bildern hinweg lösen. Ein solcher wäre beispielsweise der Lukas-Kanade Optical-flow Algorithmus [38],[39].

8.4 Pose Estimation

Sind alle Marker eines Bildes identifiziert, so können die Schwerpunktdaten der Bildkoordinaten mit den Positionskoordinaten der Marker im 3d-Referenzsystem an den Algorithmus von Tsai übergeben werden. Dieser ermittelt daraufhin die Position der Kamera sowie die Rotationsmatrix ihrer Orientierung, beide in Bezug auf das Cockpitkoordinatensystem, in welchem auch die Marker mit ihren Positionsdaten erfasst sind.

8.5 Algorithmischer Ablauf

Der nun folgende Pseudo-Code beschreibt die Hauptarbeitsschritte des Programms auf einem Einzelbild.

```
Erfasse neues Bild von der Kamera
Binärisiere das Bild
Berechne die Schwerpunkte aller sichtbaren Marker
Finde den Master

if (Master gefunden)
   Identifiziere Master
   Berechne Homographiematrix H aus den Master-Markern
   Projiziere die erwartete Position aller anderen Marker
   Assoziiere die Marker im Bild mit den IDs der projizierten Marker
   Berechne die Bewegungsvorhersage (Trend) für alle identifi-
   zierten Marker
   if(Anzahl identifizierter Marker  >= 7)
      berechne Position und Orientierung mit Tsai
   endif
endif
```

```
else \\ Master nicht oder nicht vollständig gefunden
   Identifiziere Marker über Bewegungsvorhersage (Trend)
   if (Anzahl identifizierter Marker >= 4)
      Wähle 4 der identifizierten Marker zur Homographieberechnung
      Berechne Homographiematrix H aus den gewählten Markern
      Projiziere die erwartete Position aller anderen Marker
      Assoziiere die Marker im Bild mit den IDs der proji-
      zierten Marker
      Berechne die Bewegungsvorhersage (Trend) für alle
      identifizierten Marker
      if (Anzahl identifizierter Marker  >= 7)
         berechne Position und Orientierung mit Tsai
      endif
   else \\ weniger als 4 Marker konnten identifiziert werden
      Keine Positionsberechnung möglich
endelse

if (Positionsberechnung erfolgreich)
   Überprüfe Ergebnis durch Rückprojektion der 3d-Daten in
   das Videobild
   Ausgabe der Position und Orientierung an die OpenGL
   Visualisierung
endif
```

Kapitel 9

Ergebnisse / Auswertung

Folgendes Kapitel stellt die Ergebnisse der implementierten Systemkomponenten vor. Da der zeitliche Rahmen der Arbeit nicht zur vollständigen Integration des Gesamtsystems ausreichte, werden an dieser Stelle die bis zum Abschluss der Diplomarbeit erreichten Ergebnisse vorgestellt. Die zur Gesamtintegration verbleibenden Aufgaben werden im Folgekapitel zusammengefasst.

9.1 Stabilisierte Homographieberechnung

Die Berechnung der Homographie funktionierte in der OpenGL Cockpitsimulation mit sehr geringen Fehlern. Durch die zusätzlichen Störfaktoren der Linsenverzerrung und des Sensorrauschens bei der Arbeit auf echten Bilddaten konnte eine vergleichbar hohe Genauigkeit der Markerpositions-Vorhersage jedoch nicht mehr erreicht werden. Gründe hierfür liegen in der Problematik der Masterdimensionierung, welche einen Kompromiss aus Mastergröße und Anzahl der Marker in einem Frame darstellt. Zudem führt das Rauschen zu einer Instabilität bei der Berechnung der Markerschwerpunkte, was sich auf die Projektionsmatrix und dadurch auf die Markervorhersage auswirkt. Zu Testzwecken wurde relativ zum Master-Marker ein Stabilisationspunkt eingeführt, welcher zur Homographieberechnung einen der Markerpunkte ersetzte und dadurch die Streuung der Korellationspunkte erhöhte, was zu stabileren Ergebnissen führte.

Mit dieser Verbesserung konnten für Kamerapositionen mit relativ konstanter Positionierung eine stabile Markeridentifikation erzielt werden.

Abb. 9.1 zeigt die erfassten Marker sowie ihre projizierten Pendants (dunkelgraue Kreuze). Die Berechnung der Homographie erfolgte mit den Markern N, E, W und AP-C-1 STAB L.

Eine Verbesserung der numerischen Stabilität bei der Homographieberechnung könnte durch eine verbesserte Schwerpunktberechnung erreicht werden. So bleibt zu überprüfen, inwieweit der Übergang von einer binären Schwerpunktberechnung zu einer graustufenbasierten die durch das Signalrauschen induzierten Abweichungen vom wahren Schwerpunkt minimieren würde.

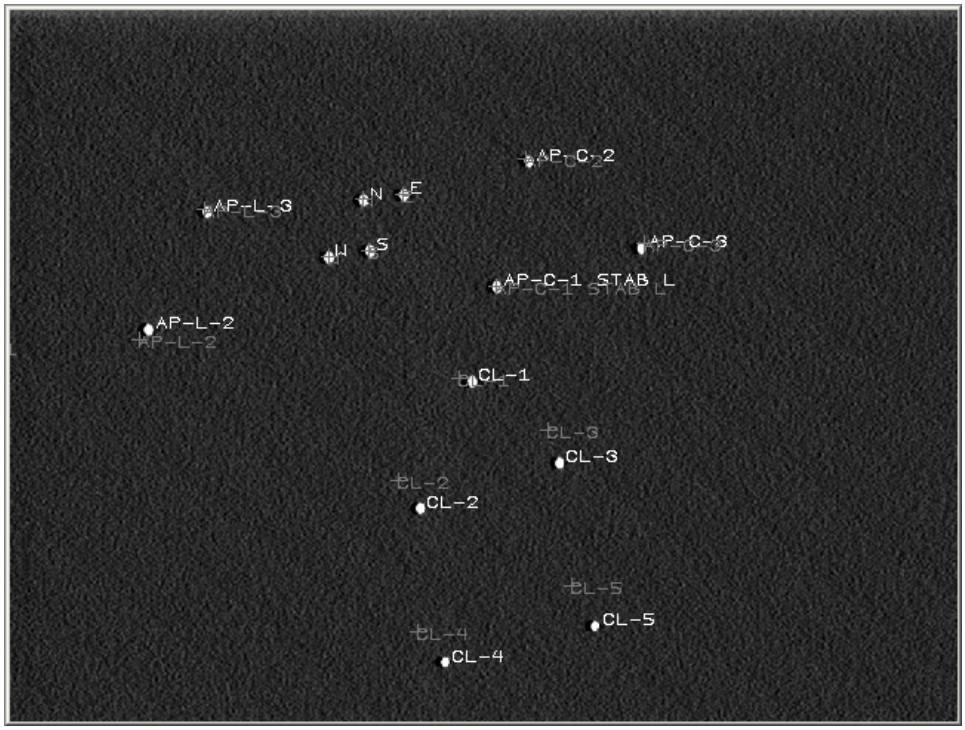


Abbildung 9.1: Markerprojektion auf Basis des stabilisierten Masters

9.2 Erreichte Trackinggenauigkeit

Aufgrund von Zeitmangel konnte die Visualisierung nicht mehr für das Head-Mounted Display umgesetzt werden. Um dennoch einen Eindruck von den erzielten Trackingergebnissen zu erlangen, wurden zwei Methoden angewandt.

9.2.1 Rückprojektion

Die im Tsai-Paket enthaltenen Routinen erlauben es, nach erfolgter Positions- und Orientierungsberechnung eine Rückprojektion von 3d-Weltdaten vorzunehmen. Dies bedeutet, dass nach Durchführung der Pose Estimation die Positionskoordinaten der sichtbaren Marker im Cockpit durch Anwendung der diversen Tsai-Transformationen (Abb. 6.15) wieder in Bildpunkte nach dem Lochkameramodell verwandelt werden. Deren Deckung mit den tatsächlich von der Kamera erfassten Koordinaten gibt Aufschluss über die Güte der errechneten Position. Die Koordinaten der rückprojizierten Marker sind in Abb. 9.2 durch die weißen Kreise dargestellt.

Bei fixierter Kameraposition bewegten sich die numerischen Schwankungen der drei Eulerwinkel im Bereich von 0,2°. Obgleich die Rahmenbedingungen zum Erlangen dieser Werte noch nicht die geforderte Allgemeinheit erreichen, unterstreichen diese Ergebnisse dennoch die Funktionstüchtigkeit des angewandten Verfahrens.

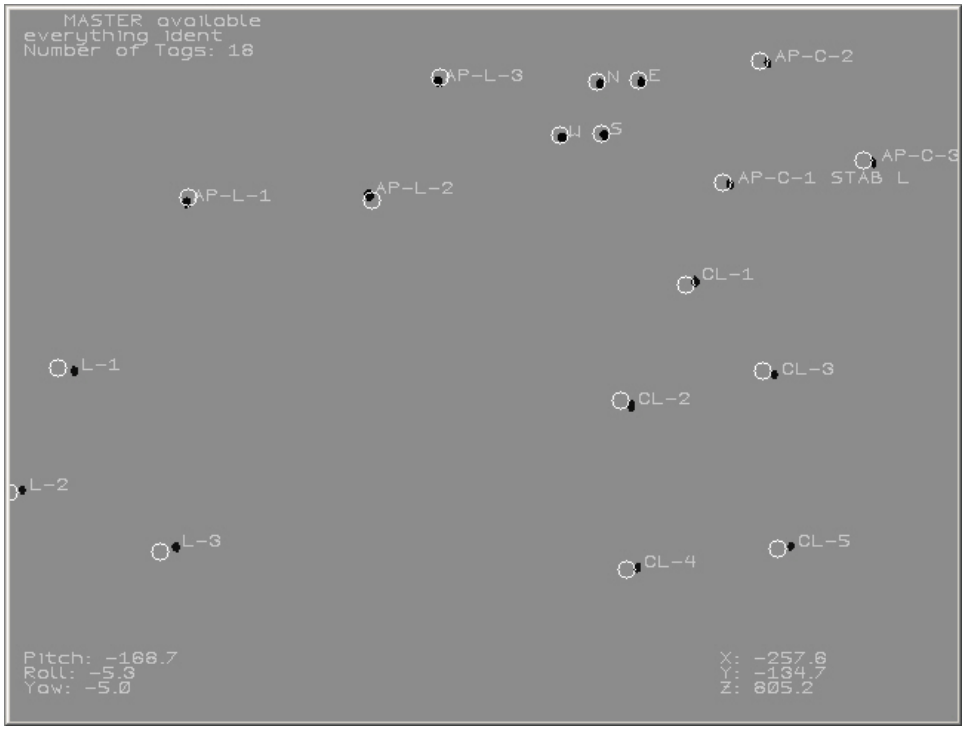


Abbildung 9.2: Rückprojektion der 3d-Markerpositionen in das Kamerabild

9.2.2 Rendern der 3d-Position

Als zweiter Validierungsschritt sollte die berechnete Position in einer OpenGL Szene gerendert werden. Hierzu wird der Camera-Output in eine OpenGL-Textur transformiert, welche als Hintergrund eingeblendet wird. Darüber wird ein dreidimensionales Modell der Marker gelegt, welches unter der berechneten Perspektive betrachtet wird. Nun sollte bei Bewegung der Kamera das 3d-Modell mit dem Videohintergrund korellieren. Abb. 9.3 zeigt die 3d-modellierten Marker (magenta), ein Referenzgittermodell (blau) sowie den tatsächlich gefilmten Hintergrund.

Trotz der scheinbar guten Überlagerung lag bei Abschluss dieser Ausarbeitung noch ein Fehler vor, welcher bis dato nicht korrigiert werden konnte. So ähneln sich beim Schwenken der Kamera die Bewegungen in beiden Repräsentationen (Tsai und OpenGL), stimmten aber noch nicht vollständig überein. Der Fehler ist in den Positionstransformationen bzw. den unterschiedlichen Kameramodellen zwischen dem Tsai-Modell und dem OpenGL-Modell zu suchen.

9.2.3 Performance

Neben der Forderung nach hoher Winkelgenauigkeit war eine zweite wichtige Anforderung eine hohe Updaterate. Dieses Kriterium wurde vollständig erfüllt. So läuft das System mitsamt der OpenGL-Ausgabe auf der in Abschnitt 5.4 spezifizierten Hardware mit einer Framerate von 30 Bildern pro

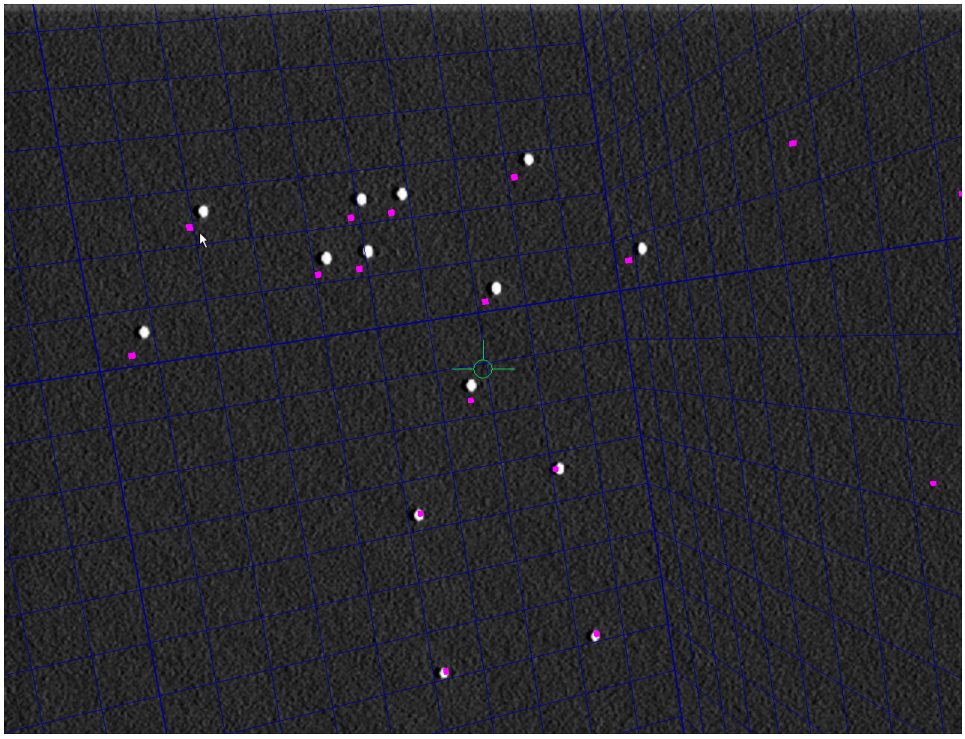


Abbildung 9.3: OpenGL-Szene der Marker über Videobild, das zur Berechnung der Perspektive diente

Sekunde. Folglich wird die maximale Berechnungszeit für ein Frame von 33ms unterschritten bzw. erreicht.

9.3 Aufgetretene Probleme

In diesem Kapitel werden noch einige Probleme adressiert, welche während der Entwicklung aufgetreten sind. Soweit möglich werden Lösungsansätze skizziert.

9.3.1 Sprünge in der Positionsberechnung

Die gesamte für das Tracking notwendige Pose Estimation erfolgt durch die Menge von Markern, welche in einem Kamerabild sichtbar sind. Geht man von einer Anzahl von beispielsweise 15 Markern aus, so hat jeder einzelne Schwerpunkt eine Anteil von ca. 7% am gesamten Datensatz, auf welchen die Berechnung zurückgreift. Hier liegt ein Problem im Verborgenen, denn verschwindet bei einem Schwenk der Kamera ein Marker aus dem Bildrand, so geht sprunghaft ein nicht zu vernachlässigender Teil der Daten verloren. Dies kann, wenn gerade dieser Marker entscheidend zur Feststellung der Perspektive beigetragen hat, zu einem Sprung in den errechneten Winkel- und Positionsdaten führen.

9.3.2 Reflexionen

Ein weiteres Problem ergab sich durch die Verwendung aktiv emittierender IR-Dioden. Diese produzierten nämlich Reflexionen in den stark geneigten Frontscheiben des Cockpits. Die Vermeidung dieser wurde dadurch erreicht, dass die entsprechenden Dioden (nicht alle produzierten Reflexionen) mit einer kleinen Kappe versehen wurden, welche sie in Richtung der entsprechenden Scheibe abschatteten.

9.3.3 Stromversorgung

Ein anderes unerwartetes Problem trat bei der Trendverfolgung im Falle eines Kameraschwenks auf. So wurden teilweise mehr Dioden im Bild erkannt als überhaupt im Cockpit installiert waren. Die Analyse von Einzelbildern während des Schwenkens der Kamera ergab eine überraschende Erklärung. Obwohl der Transformator laut Bezeichnung 12V Gleichstrom liefern sollte, lag an den Dioden eine pulsierende Spannung an. Da diese jedoch bei einer Frequenz von ca. 100Hz lag, konnte sie weder mit bloßem Auge noch in normalen Kamerabildern erkannt werden. Erst bei sehr schnellen Schwenkbewegungen trat die beschriebene Tatsache ans Licht. Abb. 9.4 zeigt ein Einzelbild einer solch schnellen Bewegung. Die Behebung dieses Problems durch den Einbau eines Kondensators konnte in der zur Verfügung stehenden Zeit nicht mehr realisiert werden.

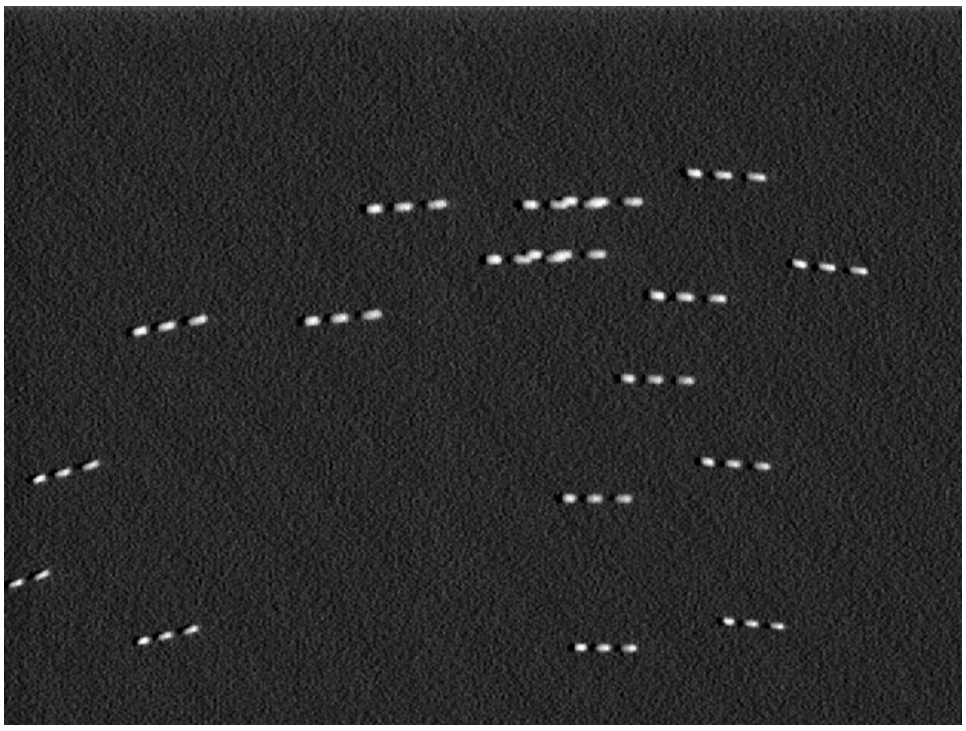


Abbildung 9.4: Hochfrequentes Flackern der Dioden (ca. 100 Hz)

Kapitel 10

Weiterentwicklung und Ausblick

10.1 Abdeckung des restlichen Blickfeldes mit Markern

Der bisher beschriebene, mit IR-Dioden versehene Trackingbereich beschränkte sich auf das Instrumentenbrett vor dem Piloten, wie in Abschnitt 8.3.1 erwähnt. Für die volle Funktionalität des Systems sollen jedoch die gesamten 150° vom Blick aus dem linken Fenster bis zum Blick aus dem rechten Fenster abgedeckt werden (Abb. 6.7).

Trotz der Tatsache, dass beim Flug im Tunnel der Blick zu über 95% geradeaus nach vorne gerichtet ist, müssen die restlichen 5% berücksichtigt werden, da gerade in diesen Fällen die Vorteile des HMD gegenüber einem HUD zur Geltung kommen. Dies gilt besonders dann, wenn das Flugzeug außerhalb des Tunnels fliegt und der Pilot versucht, in diesen zurückzukehren.

Für das Tracking bei einer Blickrichtung nach links oder rechts gibt es nun zwei mögliche Verfahren:

Das Erste besteht darin, beim Blick gerade aus die Markeridentifikation mittels des Masters zu initialisieren und bei darauffolgenden Kopfbewegungen nach links oder rechts mittels der in Kapitel 7 beschriebenen Verfahren ein Tracking der Marker und deren Identifikation aufrecht zu erhalten. Geht jedoch das Tracking verloren, müsste eine Reinitialisierung durch einen Blick auf den Master erfolgen. Wegen dieser Einschränkung und den in den Abschnitten 8.3.3 und 9.3.3 erläuterten Problemen bei der Verfolgung der Marker bei raschen Kopfbewegungen scheint eine andere Herangehensweise erfolgversprechender.

Hierbei stattet man jede der vier Hauptblickrichtungen (Abb. 6.7) mit einem eigenen Master aus (Master #0 links bis Master #3 rechts). Dieser ist räumlich jeweils so konfiguriert, dass er von der idealen Kopfposition aus im Kamerabild exakt gleich erscheint. Je weiter dieser Master vom der idealen Kopfposition entfernt ist, desto geringer fällt seine Verzerrung im Falle einer Kopfbewegung des Piloten innerhalb des Toleranzspielraums aus.

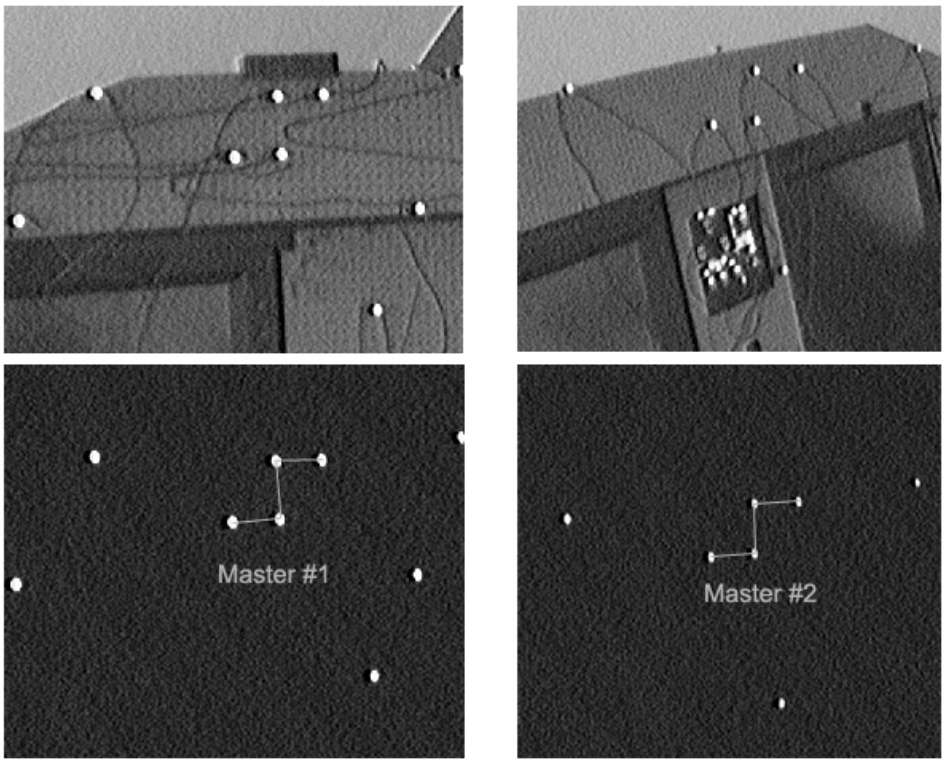


Abbildung 10.1: Rechter Master (#2) erscheint unter der Perspektive des Blickwinkels im Videobild mit der gleichen Geometrie wie Master #1

Die Unterscheidung dieser vier Master kann nach einem ähnlichen Prinzip erfolgen wie die Stabilisierung durch einen fünften Marker (Abschnitt 9.1). Dessen ausgewiesene Position relativ zum Marker könnte diesen eindeutig identifizieren. Eine andere Möglichkeit wäre die Identifikation basierend auf der errechneten Kamerarotation. Da bei einem Schwenkbereich von 150° ca. 4 Frames zur Abdeckung des gesamten Bereiches ausreichen, könnte die Entscheidung, welcher Master der gerade erfasste ist, über den Rotationswinkel φ der Kamera erfolgen.

$\varphi \approx -30°$	$\rightarrow$	Blick nach links:	Master #0
$\varphi \approx 0°$	$\rightarrow$	Blick geradeaus:	Master #1
$\varphi \approx 50°$	$\rightarrow$	Blick nach halbrechts:	Master #2
$\varphi \approx 80°$	$\rightarrow$	Blick nach rechts:	Master #3

Abb. 10.2 zeigt die wahre Anordnung der Marker des Master #2. Das beschriebene Prinzip, alle Master so zu gestalten, dass sie von der idealen Kopfposition aus zur gleichen geometrischen Abbildung im Videobild führen, wurde am Master #2 bereits erfolgreich im Simulator getestet. Offene Fragen sind hierbei jedoch noch die Gestaltung der Übergänge bei einem Kameraschwenk, wenn eventuell kurzzeitig zwei Master sichtbar sind.

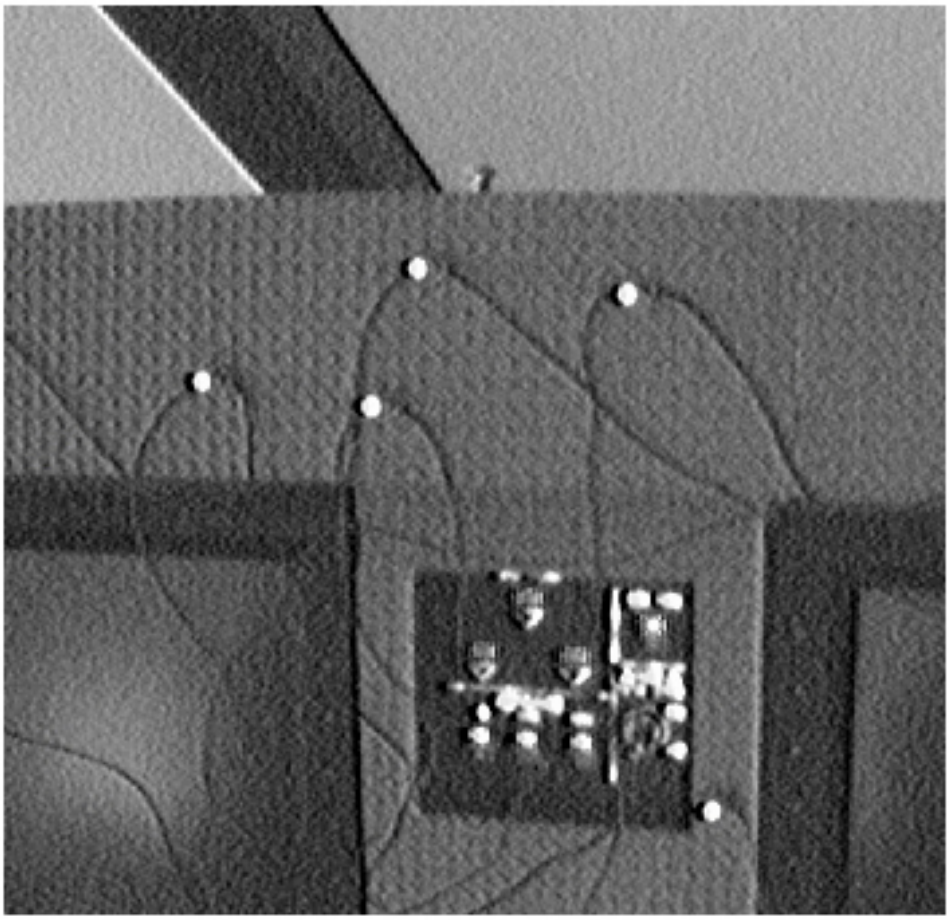


Abbildung 10.2: Anordnung der Marker zum Master #2

Auch das in Abschnitt 7.1.4 beschriebene Verfahren, bei Nichtvorhandenseins eines Masters auf andere Marker zur Homographieberechnung auszuweichen, welche besser über das Bild gestreut sind, birgt noch Ausbaupotential in sich, da durch die bessere Verteilung der Korellationspunkte die numerische Stabilität bei der Positionsvorhersage deutlich verbessert wird.

10.2 Aufbereitung der Positionsdaten für das HMD

Zur Realisierung der Tunneldarstellung im HMD, welche im Rahmen dieser Arbeit nicht mehr erfolgt ist, sind noch folgende Aufgaben zu lösen: Es ist eine benutzerspezifische Kalibration des HMDs durchzuführen, welche das Sichtfeld des Displays auf die individuelle Trageweise und den persönlichen Augenabstand des Benutzers anpasst. Diese Daten, welche mittels des SPAAM-Algorithmus [21] und der in Abschnitt 6.6 beschriebenen Kalibration gewonnen werden, müssen der Rendering-Engine, die letztendlich den Tunnel darstellt, übergeben werden, damit dort die passenden perspektivischen OpenGL Einstellungen gesetzt werden können. Zusätzlich benötigt die Rendering-Engine jedoch noch die Positions- und Orientierungsdaten

vom Piloten relativ zum Cockpit sowie vom Flugzeug relativ zur Außenwelt. Wichtig ist hierbei, dass sich die Positionsdaten des Flugzeugs, welche in der Simulationsumgebung über das Simulationsnetzwerk abgefragt werden können, auf dessen Schwerpunkt beziehen. Es ist also bei der Berechnung der Position des Piloten noch der Offset zwischen Flugzeugschwerpunkt und Cockpit zu berücksichtigen. Ebenso ist zu beachten, dass es sich bei der errechneten Kopfposition in Wahrheit um die Position der Kamera handelt. Daher ist auch noch der (statische) Offset zwischen der Kamera, welche auf dem HMD montiert ist, und dem Augpunkt des Benutzers zu berechnen. Schließlich sollte noch beim Rendern der Tunneldarstellung eine Schwarz-Maske über den Tunnel gelegt werden, welche diejenigen Bereiche ausspart, wo sich im Cockpit die Fenster befinden. So wird sichergestellt, dass keine Instrumente innerhalb des Cockpits vom Tunnel überzeichnet und dadurch schlechter ablesbar werden.

10.3 Bewertung und Ausblick

Bei der Entwicklung des vorgestellten Systems wurde konsequent versucht, Einschränkungen dort zu treffen wo sie sinnvoll sind, andererseits jedoch im Rahmen der Themenstellung eine gewisse Anpassungsfähigkeit an variierende Rahmenbedingungen zu bewahren. Fällt die Entscheidung auf ein kamerabasiertes Trackingsystem, was insbesondere durch die geringen Anschaffungskosten gerechtfertigt werden kann, so stellt die hier beschriebene Konfiguration wohl einen erfolgversprechenden Ansatz dar. Obgleich viele Komponenten des Gesamtsystems deutliches Optimierungspotential in sich tragen, bleibt letztendlich die Frage, ob die existierenden Limitationen von kamerabasierten Trackingsystemen durch Einschränkungen in der Systemauslegung kompensiert werden können. Zu diesen Limitationen zählt beispielsweise die Update-Rate von 30Hz, welche im Vergleich mit anderen Verfahren [33] (bis zu 240Hz) deutlich geringer ausfällt. Auch bleibt abzuwarten, ob markerlose Trackingverfahren [13],[14],[15] in der Zukunft die notwendige Performanz erreichen, um hohe Winkelgenauigkeiten bei hohen Updateraten zu produzieren.

Anhang A

Benutzte Software

A.1 OpenCV Bibliothek

OpenCV ist eine von Intel zur Verfügung gestellte, sowohl für die Forschung als auch für kommerzielle Anwender freie Bibliothek zur Bildverarbeitung. Sie besteht aus einer Ansammlung von C-Funktionen und einigen C++ Klassen, in welchen effiziente Algorithmen zur Bildmanipulation implementiert sind. OpenCV besteht aus über 300 Funktionen, welche vom Intel Research Lab, Nizhny Novgorod, Russland entwicklet wurden. Darunter befinden sich unter anderem folgende Funktionen:

- Schwellenwertberechnung
- Histogrammfunktionen
- Finden von Zusammenhangskomponenten
- Farbformat-Konvertierung
- Rudimentäre Zeichenfunktionen (line, ellipse, rectangle) und Textausgabe
- find good features to track
- Graphenoperationen
- Objekterkennung

Darüber hinaus bietet OpenCV ein rudimentäres Graphical User Interface (GUI) zur Darstellung gängiger Bildformate (.jpg, .bmp), was die rasche Überprüfung von Ergebnissen deutlich vereinfacht. Folgendes kurzes Beispiel soll die einfache Handhabung verdeutlichen:

```
#include "cv.h"       // include core library interface
#include "highgui.h"  // include GUI library interface
IplImage* imgc  = 0;  // Declare IPL/OpenCV image pointers
IplImage* imgsw = 0;

// filename of the image to load
char* filename[] = { "../color640.jpg"};

int main()
{
CvSize imgSize;

// create HighGUI windows with its title
cvvNamedWindow( "Original", 1 );
cvvNamedWindow( "Greyscale", 1 );

imgc = cvvLoadImage( filename[0] );   // load  image

// define image size for new image
imgSize.width = 640;
imgSize.height= 480;

// create new image (640x480, 1 channel (=greyscale))
imgsw  = cvCreateImage (imgSize, 8 ,1);

// convert original image to greyscale
cvCvtColor( imgc, imgsw , CV_RGB2GRAY );

// show image in window with its respective titles
cvvShowImage( "Original", imgc );
cvvShowImage( "Greyscale", imgsw );

cvvWaitKey(0);                // wait till user presses key

cvReleaseImage( &imgc );      // release  images
cvReleaseImage( &imgsw);

cvDestroyWindow("Original");  // destroy window
cvDestroyWindow("Greyscale");

return 0;
}
```

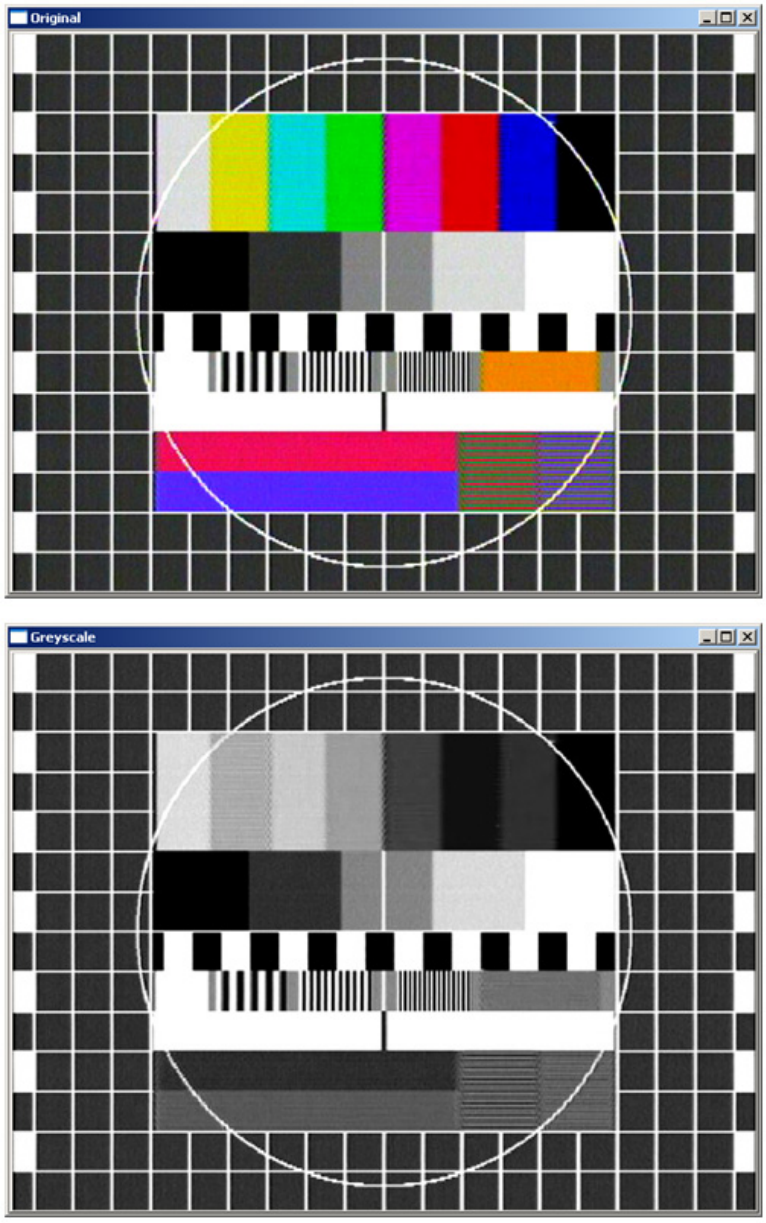


Abbildung A.1: OpenCV: Konvertierung eines 3-Kanal RGB Bildes in (1-Kanal) Graustufen und Darstellung mit dem GUI

A.2 PointGrey Research Capture Software

Im Umfang der mit der Firefly2 Kamera gelieferten Software ist auch der PGR FlyCapture Software Development Kit enthalten, dessen einfache Handhabung anhand einiger mitgelieferter Beispielprogramme illustriert wird. Eines dieser Programme, welches in angepasster Form auch in der Systemimplementation zum Einsatz kommt, sei an dieser Stelle kurz vorgestellt. Die Gesamtübersicht der API ist in [37] enthalten. Folgendes Beispielprogramm initialisiert die Kamera, erfasst eine Sequenz von Bildern und speichert das letzte Bild auf der Festplatte ab.

Line #	**Description**
19	Creates a FlyCaptureError variable. In order to reliably debug your application, define a variable of this type to capture meaningful errors returned by API functions.
26	Enumerates, or lists, all PGR cameras sitting on the bus and their bus index, starting at zero. This function does not enumerate non-PGR cameras.
37	Creates a FlyCaptureContext. A context acts as a handle to the camera, and is required to initialize and start the camera.
41	Initializes the camera located at bus index zero and associates it with the camera context. Multiple cameras connected to the FireWire bus are enumerated at bus indices (FireWire nodes) that start at 0.
45	Starts an initialized camera at a defined resolution (640x480 pixels), image format (Y8 8 bits per pixel) and frame rate (15 frames captured per second). This function also allocates four buffers in main memory that are used to hold the images that are streamed in from the camera. Once a camera has been started, it immediately begins capturing and streaming images via DMA to these memory buffers. Once these buffers are full, they will be overwritten with consecutive images unless they are locked by the user.
49	Creates a FlyCaptureImage variable. The FlyCaptureImage structure contains the image data, as well as video mode, whether the image is stippled (color) and timestamp information. See the section *Timestamping Mechanisms* in the *Advanced Features* chapter for more information on the types of timestamps available.
58	When a call to flycaptureGrabImage2 is made, a pointer to the image buffer (&image) with the newest (latest) complete image is returned. The call to grabImage2 does not involve copying, so it is quite fast. The user will never be given an image that is older than one that has already been seen. Once the pointer to the buffer is returned to the user, this buffer remains locked until grabImage2 is called again. If no buffer contains an image newer than the last returned, then the flycaptureGrabImage2 call will block until a new image is available.
65	Converts the last image grabbed to a 24-bit per pixel image that can be displayed by Microsoft Windows (which uses the BGR format). If the camera is not a color camera, Y8 and Y16 images are converted to BGR24 greyscale.
68	Writes the image out to a PPM format file.
72	Destroys the camera context. In order to prevent memory leaks from occurring, this function must be called when the user is finished with the FlyCaptureContext.

```
1  //=============================================================================
2  // PGRFlyCaptureTest.cpp
3  //   - Grabs a few images and saves the last to disk.
4  //=============================================================================
5
6  #include "pgrflycapture.h"
7
8  // The maximum number of cameras on the bus
9  #define _MAX_CAMS       32
10
11 // The number of images to grab.
12 #define _IMAGES_TO_GRAB 5
13
14 // The index of the camera to grab from.
15 #define _CAMERA_INDEX   0
16
17 int main( int /* argc */, char* /* argv[] */ )
18 {
19    FlyCaptureError   error;
20    FlyCaptureContext context;
21
22    // Enumerate the cameras on the bus
23    FlyCaptureInfo arInfo[ _MAX_CAMS ];
24    unsigned int  uiSize = _MAX_CAMS;
25
26    error = flycaptureBusEnumerateCameras( arInfo, &uiSize );
27
28    for( unsigned int uiBusIndex = 0; uiBusIndex < uiSize; uiBusIndex++ )
29    {
30       printf( "Bus index %u: %s (%u)\n",
31               uiBusIndex,
32               arInfo[ uiBusIndex ].pszModelString,
33               arInfo[ uiBusIndex ].SerialNumber );
34    }
35
36    // create the flycapture context.
37    error = flycaptureCreateContext( &context );
38
39    // Initialize the camera.
40    printf( "Initializing camera %u.\n", _CAMERA_INDEX );
41    error = flycaptureInitialize( context, _CAMERA_INDEX );
42
43    // Start grabbing images in 8-bit greyscale (or stippled, if this is a
44    // colour camera) 640x480 mode with a frame rate  of 15 fps.
45    error = flycaptureStart(
46         context, FLYCAPTURE_VIDEOMODE_640x480Y8, FLYCAPTURE_FRAMERATE_15 );
47
48    // Time the grabbing of 30 images.
49    FlyCaptureImage image;
50
51    // Initialize the image structure to sane values
52    image.iCols = 0;
53    image.iRows = 0;
54
55    printf( "Grabbing images" );
56    for ( int iImage = 0; iImage < _IMAGES_TO_GRAB; iImage++ )
57    {
58       error = flycaptureGrabImage2( context, &image );
59    }
60
61    // Convert the last image to BGR24 for flycaptureWritePPM().
62    unsigned char* pimageBGR24 =
63       new unsigned char[ image.iCols * image.iRows * 3 ];
64
65    error = flycaptureConvertImage(
66       context, &image, FLYCAPTURE_OUTPUT_BGR, pimageBGR24 );
67
68    error = flycaptureWritePPM(
69       pimageBGR24, image.iRows, image.iCols, "image.ppm" );
70
71    // Destroy the context.
72    error = flycaptureDestroyContext( context );
73
74    delete [] pimageBGR24;
75
76    return 0;
77 }
```

A.3 OpenGL Cockpit Simulator

Zur Überprüfung der Bildbearbeitugskomponenten wurden die wichtigsten Bauteile des LFM-Cockpit mit einer Genauigkeit im Millimeterbereich als OpenGL-Modell nachgebildet. Diese Cockpit-Simulation erlaubte es, die angewandten Verfahren in einer kontrollierten und störungsfreien Umgebung zu testen. So unterliegen die aus der Cockpitsimulation extrahierten Testdaten weder Rauscheinflüssen noch radialer Linsenverzerrung. Die Simulation erlaubt eine exakte Positionsbestimmung des Augpunktes des Betrachters sowie einen frei skalierbaren Öffnungswinkel der Kamera (Field of View). Sowohl Position als auch Blickrichtung können frei gewählt werden.

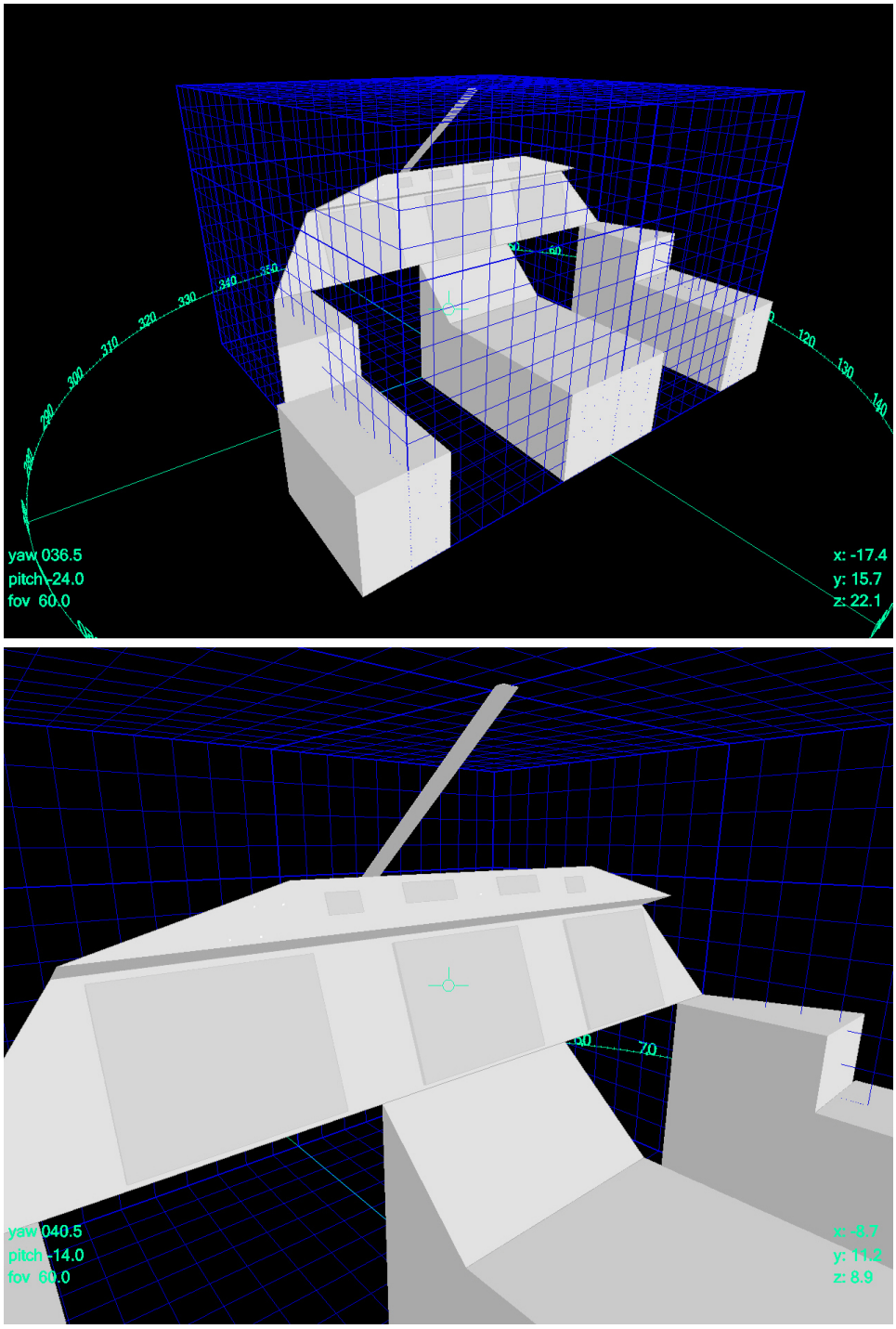


Abbildung A.2: OpenGL Cockpit Simulation

A.4 Kamera-Kalibration für den Tsai-Algorithmus

Zur Errechnung der Kameraposition (Translation in X, Y und Z) sowie ihrer Orientierung im Raum (Yaw, Pitch und Roll-Winkel der Kamera) sind im Tsai Modell drei Parameter-Gruppen vorhanden [6]. Diese sind:

```
camera parameters cp
calibration constants cc
calibration data cd
```

Im folgenden wird beschrieben aus welchen Variablen sich die jeweilige Gruppe zusammen setzt und wie man die einzelnen Werte ermittelt.

A.4.1 Camera parameters

Die Kameraparameter, welche Tsai verwendet, sind kameraspezifische Daten die vom Hersteller zur Verfügung gestellt werden. Sie beinhalten folgende Werte:

- `Ncx:` beschreibt die Anzahl horizontal auf dem Chip angeordneter Sensorelemente [sel]
- `Nfx:` beschreibt die Anzahl der vom Framegrabber erfassten Pixel in einer Scanline (in x-Richtung) [pix]
- `dx:` Zentrum-zu-Zentrum Abstand zweier benachbarter Sensorelemente in x-Richtung (entspricht der Breite eines Sensorelements plus dem Abstand zwischen zwei Elementen)
- `dy:` gibt den Zentrum-zu-Zentrum Abstand zweier vertikal benachbarter Sensorelemente in y-Richtung an. Dieser ist bei quadratischen Sensorelementen i.A. gleich `dx`.
- `dpx:`
- `dpy:`
- `Cx:` bestimmt den Durchstosspunkt der Kamera-Z-Achse durch die Bildebene, also den Bildursprung in x-Richtung
- `Cy:` bestimmt den Bildursprung in y-Richtung

Moderne Digitalkameras, wie die verwendete FireFly2, haben einen CCD-Bildsensor. Die verwendeten Werte lassen sich auf dem Datenblatt des CCD Chips nachlesen [36]. Folgende Werte wurden für die FireFly2-Cam ermittelt:

```
camera parameters cp:

  Ncx = 659                    //[sel]
  Nfx = 640                    //[pix]

  dx = 0.066474                //[mm/sel]
  dy = 0.066474                //[mm/sel]
```

```
dpx = cp.dx * cp.Ncx / cp.Nfx //[mm/pix]
dpy = cp.dy                   //[mm/pix]

Cx = 299.629804               //[pix]
Cy = 279.245491               //[pix]

sx = 1.032665                 //[-]
```

Erwähnenswert hierbei ist, dass die Kalibrationsdaten in einem offline Prozess a-priori ermittelt werden können. Diese müssen dann später nur noch einmalig vor Ablauf des Programms geladen werden.

Calibration constants

Die zweite Gruppe von kameraabhängigen Variablen sind die Kalibrationskonstanten. Diese beinhalten neben den gewünschten Positionsdaten (Translation/Rotation) noch weitere Kameraparameter. Es folgt eine vollständige Aufzählung der bei Tsai verwendeten Kalibrationskonstanten:

```
calibration constants cc:

f          //[mm]
kappa1     //[1/mm^2]
p1         //[1/mm]
p2         //[1/mm]

Tx         //[mm]
Ty         //[mm]
Tz         //[mm]

Rx         //[rad]
Ry         //[rad]
Rz         //[rad]

r1         //[-]
r2         //[-]
r3         //[-]
r4         //[-]
r5         //[-]
r6         //[-]
r7         //[-]
r8         //[-]
r9         //[-]
```

- `f`: Brennweite der Lochkamera (focal length) in mm.
- `kappa1`: Koeffizient zur Bestimmung der radialen Linsenverzerrung

- `Tx, Ty, Tz`: Translationsvektor vom Ursprung des Weltkoordinatensystems zum Projektionspunkt der Kamera.
- `Rx, Ry, Rz`: Rotationswinkel der Kamera um die Lateral-, Longitudinal- und Vertikalachse
- `r1-r9`: Einträge derRotationsmatrix `R`

Calibration data

Die letzte Gruppe von Variablen bilden den eigentlichen Datensatz der 3d-2d Registration. Sie besteht aus einer Liste von 3d Weltkoordinaten in einem orthonormalen rechts-orientierten Koordinatensystem in Millimeter und den korrespondierenden 2d-Bildkoordinaten in Pixel. Beispiel:

```
180.0  180.0  0.0  129.0  321.5
212.0  180.0  0.0  188.0  316.5
212.0  212.0  0.0  193.0  264.5
244.0  212.0  0.0  252.5  259.5
308.0  132.0  0.0  389.0  392.5
308.0  284.0  0.0  367.0  145.0
412.0  284.0  0.0  544.0  130.5
412.0  148.0  0.0  601.5  336.5
```

A.4.2 Vorgehensweise

Zur Durchführung der Kamerakalibration wurden 24 der auch im Cockpit verwendeten Infrarot-Dioden in einer schachbrettartigen 6x4 Anordnung auf einer Platte montiert. Diese wurde aus verschiedenen Höhen abgefilmt, um mehrere Datensätze zur Verfügung zu haben. Abb. A.3 zeigt den Kalibrationsaufbau zu zwei verschiedenen Zeitpunkten.

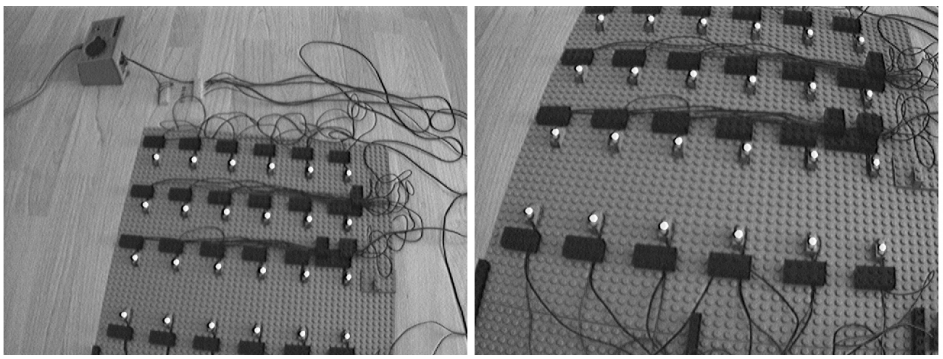


Abbildung A.3: Kamerakalibration nach Tsai

Non-Coplanar Pose Estimation

Um alle für das Lochkameramodell benötigten Parameter berechnen zu können ist es obligatorisch, dass eine räumliche Verteilung der Kalibrationspunkte vorliegt. Die benutzte Implementation des Tsai-Algorithmus bietet eine langsame, aber optimierte Routine für diesen nicht-coplanaren Fall an. Um über eine ausreichende Zahl von Messpunkten zu verfügen wurden die planaren Datensätze zu einem dreidimensionalen Datensatz kombiniert. Abb. A.4 zeigt die ermittelten Kalibrationsparameter sowie eine Rückprojektion der 3d-Kalibrationspunkte in das ursprüngliche Bild.

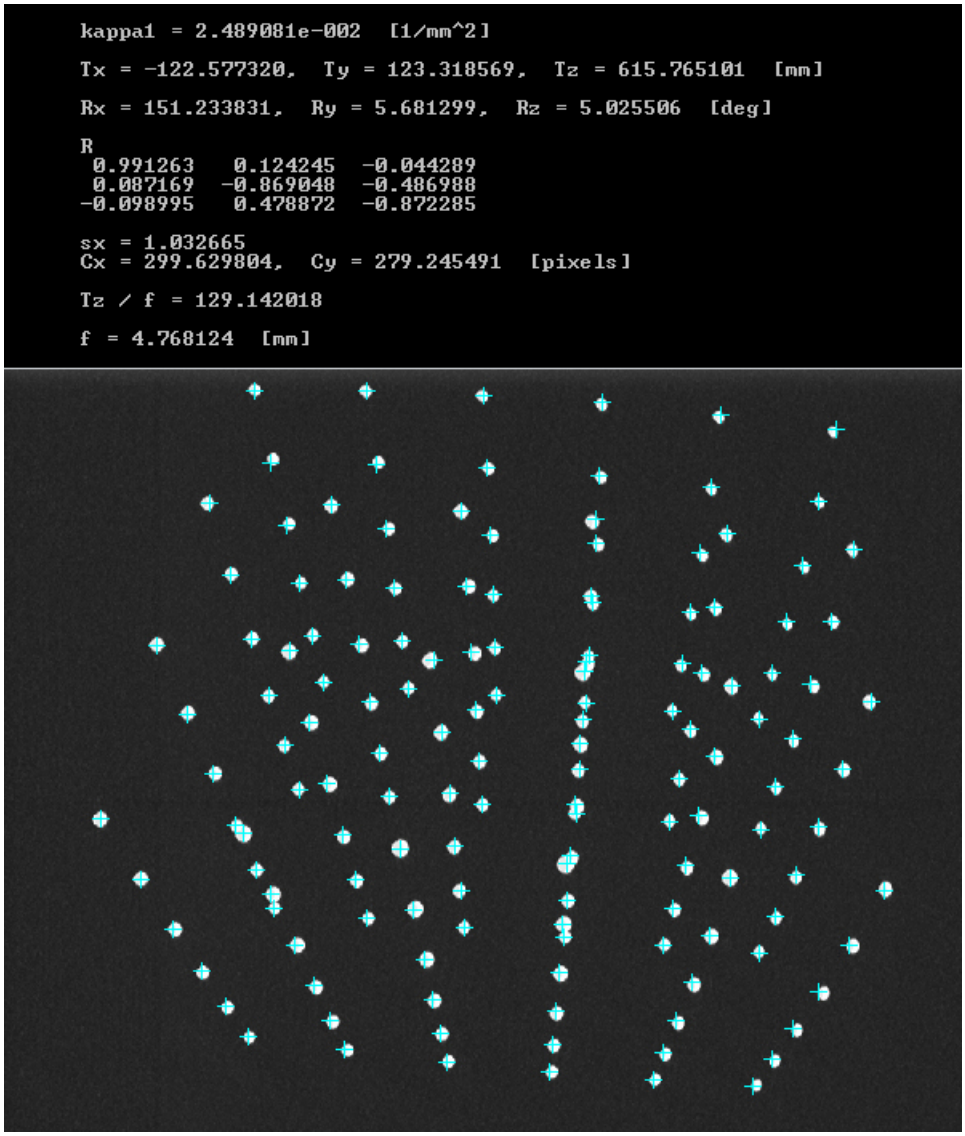


Abbildung A.4: Ergebnisse der Kamerakalibration

Anhang B

Benutzte Hardware

B.1 Stromversorgung/Schaltplan der LEDs

Als Versorgungsspannung dient ein Gleichstromtransformator, welcher eine Spannung von 12V liefert. Der Spannungsabfall am Vorwiderstand berechnet sich aus der Gesamtspannung minus der Summe der Einzelspannungsabfälle der Dioden. Dieser beträgt im vorliegenden Fall 3V. Bei einer erforderlichen Stromstärke von 100 mA für die LEDs ergibt sich ein theoretischer Widerstand von 30 Ohm. Tatsächlich wurde ein 33 Ohm Widerstand verwendet.

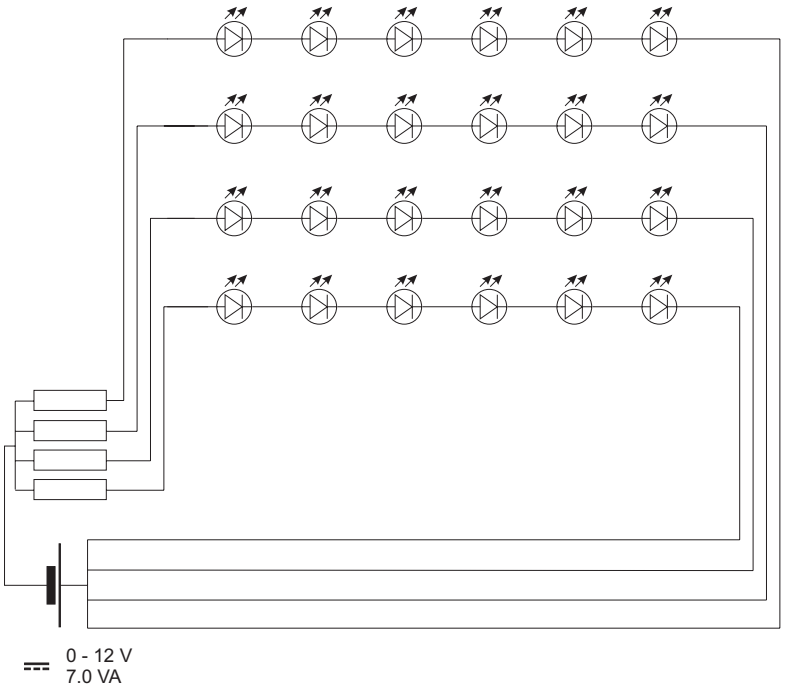


Abbildung B.1: Schaltplan der LEDs mit 12V Spannungsquelle und 33Ω Vorwiderstand

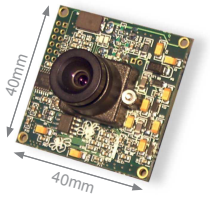
B.2 PGR FireFly2 Progressive Scan Camera

compact, low cost IEEE-1394 digital camera

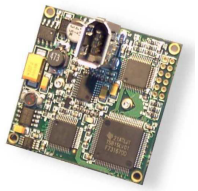
- 6 Pin IEEE-1394 interface
- 1/4" progressive scan Sony CCD
- Compact size - 40x40mm
- 640x480 uncompressed color images
- Low cost OEM board camera soulution

firefly2

POINT GREY RESEARCH



front view



back view

Firefly2 is a compact board level IEEE-1394 Digital Video Camera. Firefly2 uses a 1/4'' progressive scan CCD in order to stream VGA quality color images at 30 FPS without compression. The camera is provided as a complete development kit with an IEEE-1394 interface card, cable, and image acquisition software.

Package includes:

- Firefly2 board level camera
- 4.5 meter, 6-pin, IEEE-1394 cable
- 4, 6 and 8mm focal length M12 micro lenses
- IEEE-1394 OHCI PCI Interface card
- PGRFlyCapture image acquisition and camera control C/C++ SDK

System requirements:

- Intel Pentium II or better
- Windows 2000 or XP

Camera specifications:

Imaging Device	1/4'' Sony CCD (ICX098AK) Color VGA 640x480 format HAD image sensor with square pixels Progressive scan
Supported frame rates:	3.75, 7.5, 15 & 30 FPS
Supported formats:	YUV 4:1:1, YUV 4:2:2, YUV 4:4:4, and RGB 24-bit
Digital camera specification:	Version 1.04
Signal to noise ratio:	>40dB
Connector:	6-pin IEEE-1394, vertical
Power:	Through IEEE-1394, 625mW standby, 1.25 W active
Brightness:	Auto/Manual (-3dB to 33dB)
Exposure:	Auto/Manual (1/25s to 1/15000s)
Saturation:	Manual
White Balance:	Auto/Manual
Lens focal length:	4mm, 6mm or 8mm (included in the kit)
Footprint:	40 x 40mm
Weight:	12g with a micro lens

www.ptgrey.com

305-1847 West Broadway, Vancouver, B.C.
Canada, V6J 1Y6
T: 604-730-9937 F: 604-732-8231

Point Grey Research is a product engineering and technology company founded in January 1997. The company designs and develops computer vision technologies for commercial applications worldwide. Point Grey Research technology has been successfully used in people tracking, object tracking, modeling and dimensioning, mobile robotics, mining and many other computer vision applications.

B.3 SONY ICX098AK CCD Image Sensor

SONY **ICX098AK**

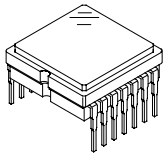
Diagonal 4.5mm (Type 1/4) Progressive Scan CCD Image Sensor with Square Pixel for Color Cameras

Description

The ICX098AK is a diagonal 4.5mm (Type 1/4) interline CCD solid-state image sensor with a square pixel array which supports VGA format. Progressive scan allows all pixels signals to be output independently within approximately 1/30 second. Also, the adoption of monitoring mode allows output to an NTSC monitor without passing through the memory. This chip features an electronic shutter with variable charge-storage time which makes it possible to realize full-frame still image without a mechanical shutter. High resolution and high color reproductivity are achieved through the use of R, G, B primary color mosaic filters. Further, high sensitivity and low dark current are achieved through the adoption of HAD (Hole-Accumulation Diode) sensors.

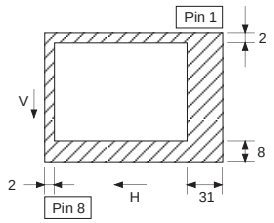
This chip is suitable for applications such as electronic still cameras, PC input cameras, etc.

14 pin DIP (Plastic)



Features

- Progressive scan allows individual readout of the image signals from all pixels.
- High horizontal and vertical resolution (both approx. 400TV-lines) still image without a mechanical shutter.
- Supports monitoring mode
- Square pixel
- Supports VGA format
- Horizontal drive frequency: 12.27MHz
- No voltage adjustments (reset gate and substrate bias are not adjusted.)
- R, G, B primary color mosaic filters on chip
- High resolution, high color reproductivity, high sensitivity, low dark current
- Continuous variable-speed shutter
- Low smear
- Excellent antiblooming characteristics
- Horizontal register: 3.3V drive
- 14-pin high precision plastic package (enables dual-surface standard)



Optical black position (Top View)

Device Structure

- Interline CCD image sensor
- Image size: Diagonal 4.5mm (Type 1/4)
- Number of effective pixels: 659 (H) × 494 (V) approx. 330K pixels
- Total number of pixels: 692 (H) × 504 (V) approx. 350K pixels
- Chip size: 4.60mm (H) × 3.97mm (V)
- Unit cell size: 5.6µm (H) × 5.6µm (V)
- Optical black: Horizontal (H) direction: Front 2 pixels, rear 31 pixels
  Vertical (V) direction: Front 8 pixels, rear 2 pixels
- Number of dummy bits: Horizontal 16
  Vertical 5
- Substrate material: Silicon

Wfine CCD®

* Wfine CCD is a registered trademark of Sony Corporation.
Represents a CCD adopting progressive scan, primary color filter and square pixel.

B.4 Linos RG 830 Tageslichtsperrfilter

Farbglasfilter

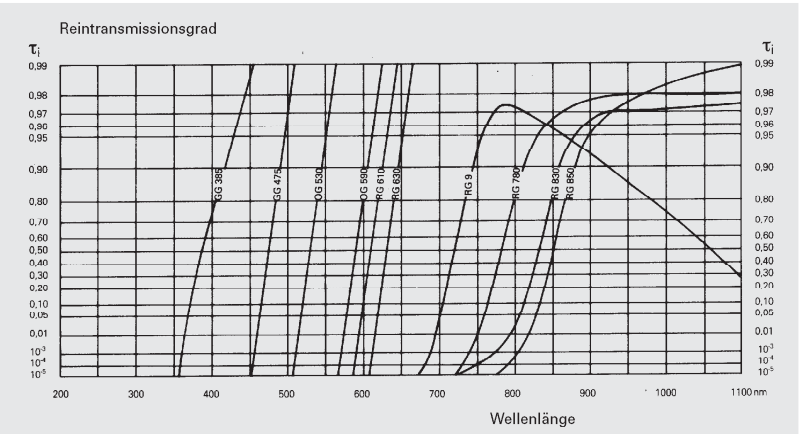
		ungefasst		gefasst*)				
				Fassung CL freier Ø 21.4 mm	Fassung 50 freier Ø 48 mm			
	Typ	Ø 22.2 mm Best.-Nr.	Ø 50 mm Best.-Nr.	Best.-Nr.	Best.-Nr.	Kanten-lage bei •• (nm)	Re-flexions-faktor P	Dicke (mm)
UV-Absorptionsfilter, farblos	**GG 385**	**37 0106**	**37 0014**	**06 3438**	**03 1875**	385	0.905	2
Gelbfilter	**GG 475**	**37 0088**	**37 0013**	**06 3452**	**03 1869**	475	0.915	3
Orangefilter	**OG 530**	**37 0009**	**37 0012**	**06 3453**	**03 1870**	530	0.915	3
Rotfilter, hell	**OG 590**	**37 0090**	**37 0058**	**06 3439**	**03 1880**	590	0.915	2
Rotfilter, mittel	**RG 610**	**37 0107**	**37 0041**	**06 3440**	**03 1871**	610	0.915	2
Rotfilter, dunkel	**RG 630**	**37 0081**	**37 0011**	**06 3081**	**03 1872**	630	0.915	3
IR-Filter	**RG 780**	**37 0091**	**37 0019**	**06 3454**	**03 1874**	780	0.915	3
IR-Filter	**RG 830**	**37 0092**	**37 0063**	**06 3455**	**03 1881**	830	0.915	3
IR-Filter	**RG 850**	**37 0108**	**37 0109**	**06 3441**	**03 1884**	850	0.910	3
IR-Selektivfilter für nahes IR	**RG 9**	**37 0093**	**37 0042**	**06 3456**	**03 1873**	735	0.915	2

*) Abmessungen der Fassungen im Kapitel Optikfassungen

in Fassung zum direkten Einbau in die Mikrobank

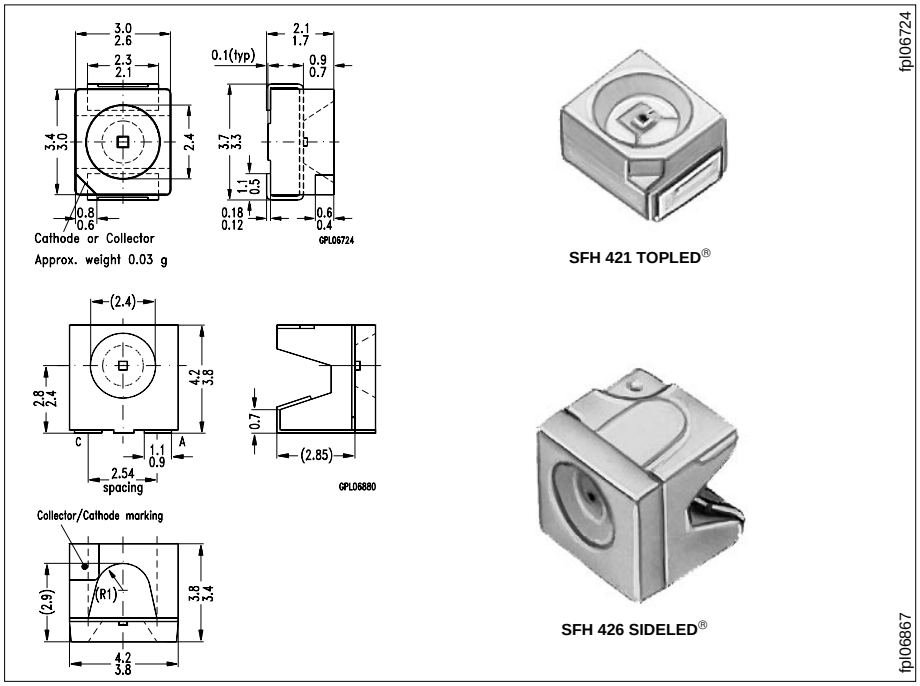
Toleranzen

Ø 22,2 mm	-0,3 mm
Ø 50 mm	-0,6 mm
Dicke	±0,2 mm
Blasen	1/5x0,25
Schlieren	2/03
Oberflächenqualität	5/3x0,4
Parallelität	15'



B.5 SFH-421 High Intensity IR-LEDs

GaAlAs-IR-Lumineszenzdiode in SMT-Gehäuse **SFH 421**
GaAlAs Infrared Emitter in SMT Package **SFH 426**



Maβe in mm, wenn nicht anders angegeben/Dimensions in mm, unless otherwise specified.

Wesentliche Merkmale

- GaAlAs-LED mit sehr hohem Wirkungsgrad
- Gute Linearität ($I_e = f\,[I_F]$) bei hohen Strömen
- Gleichstrom- (mit Modulation) oder Impulsbetrieb möglich
- Hohe Zuverlässigkeit
- Hohe Impulsbelastbarkeit
- Oberflächenmontage geeignet
- Gegurtet lieferbar
- SFH 421 Gehäusegleich mit SFH 320/420
  SFH 426 Gehäusegleich mit SFH 325/425
- SFH 426: Nur für IR-Reflow-Lötung geeignet. Bei Schwallötung wenden Sie sich bitte an uns.

Features

- Very highly efficient GaAlAs-LED
- Good Linearity ($I_e = f\,[I_F]$) at high currents
- DC (with modulation) or pulsed operations are possible
- High reliability
- High pulse handling capability
- Suitable for surface mounting (SMT)
- Available on tape and reel
- SFH 421 same package as SFH 320/420
  SFH 426 same package as SFH 325/425
- SFH 426: Suitable only for IR-reflow soldering. In case of dip soldering, please contact us first.

Anwendungen

- Miniaturlichtschranken für Gleich- und Wechsellichtbetrieb, Lochstreifenlaser
- Industrieelektronik
- “Messen/Steuern/Regeln”

Applications

- Miniature photointerrupters
- Industrial electronics
- For drive and control circuits

Typ **Type**	**Bestellnummer** **Ordering Code**	**Gehäuse** **Package**
		Kathodenkennzeichnung: abgesetzte Ecke cathode marking: bevelled edge
SFH 421	Q62703-P2407	TOPLED
SFH 426	Q62703-P0331	SIDELED

Grenzwerte (T_A = 25 °C)
Maximum Ratings

Bezeichnung **Description**	**Symbol** **Symbol**	**Wert** **Value**	**Einheit** **Unit**
Betriebs- und Lagertemperatur Operating and storage temperature range	T_{op}; T_{stg}	– 55 ... + 100	°C
Sperrschichttemperatur Junction temperature	T_j	100	°C
Sperrspannung Reverse voltage	V_R	5	V
Durchlaßstrom Forward current	I_F	100	mA
Stoßstrom, τ = 10 µs, D = 0 Surge current	I_{FSM}	2.5	A
Verlustleistung Power dissipation	P_{tot}	180	mW
Wärmewiderstand bei Montage auf FR4-Platine, Padgröße je 16 mm² Thermal resistance mounted on PC-board (FR4), padsize 16 mm² each	R_{thJA}	450	K/W
Chip zu Lötstelle chip to solder point	R_{thJS}	≈ 200	K/W

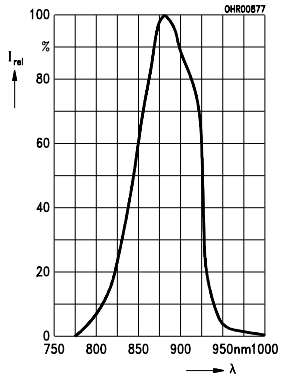
Kennwerte (T_A = 25 °C)
Characteristics

Bezeichnung **Description**	**Symbol** **Symbol**	**Wert** **Value**	**Einheit** **Unit**
Wellenlänge der Strahlung Wavelength at peak emission I_F = 100 mA, t_p = 20 ms	λ_{peak}	880	nm
Spektrale Bandbreite bei 50% von I_{max} Spectral bandwidth at 50% of I_{max} I_F = 100 mA	$\Delta\lambda$	80	nm
Abstrahlwinkel Half angle	φ	± 60	Grad deg.
Aktive Chipfläche Active chip area	A	0.16	mm^2
Abmessungen der aktiven Chipfläche Dimension of the active chip area	$L \times B$ $L \times W$	0.4 × 0.4	mm
Schaltzeiten, I_e von 10 % auf 90 % und von 90 % auf 10 %, bei I_F = 100 mA, R_L = 50 Ω Switching times, I_e from 10 % to 90 % and from 90 % to 10 %, I_F = 100 mA, R_L = 50 Ω	t_r, t_f	0.5	µs
Kapazität Capacitance V_R = 0 V, f = 1 MHz	C_o	25	pF
Durchlaßspannung Forward voltage I_F = 100 mA, t_p = 20 ms I_F = 1 A, t_p = 100 µs	 V_F V_F	 1.5 (≤ 1.8) 3.0 (≤ 3.8)	 V V
Sperrstrom Reverse current V_R = 5 V	I_R	0.01 (≤ 1)	µA
Gesamtstrahlungsfluß Total radiant flux I_F = 100 mA, t_p = 20 ms	Φ_e	23	mW
Temperaturkoeffizient von I_e bzw. Φ_e, I_F = 100 mA Temperature coefficient of I_e or Φ_e, I_F = 100 mA	TC_I	– 0.5	%/K
Temperaturkoeffizient von V_F, I_F = 100 mA Temperature coefficient of V_F, I_F = 100 mA	TC_V	– 2	mV/K
Temperaturkoeffizient von λ, I_F = 100 mA Temperature coefficient of λ, I_F = 100 mA	TC_λ	+ 0.25	nm/K

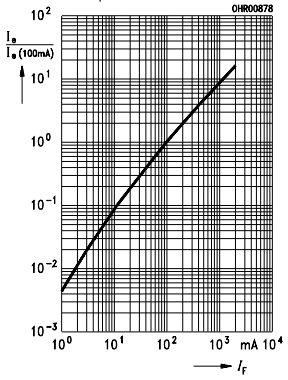
Gruppierung der Strahlstärke I_e in Achsrichtung
gemessen bei einem Raumwinkel Ω = 0.01 sr
Grouping at radiant intensity I_e in axial direction
at a steradian of Ω = 0.01 sr

Bezeichnung **Description**	**Symbol**	**Werte** **Values**	**Einheit** **Unit**
Strahlstärke Radiant intensity I_F = 100 mA, t_p = 20 ms	I_e	> 4	mW/sr
Strahlstärke Radiant intensity I_F = 1 A, t_p = 100 µs	$I_{e\ typ.}$	48	mW/sr

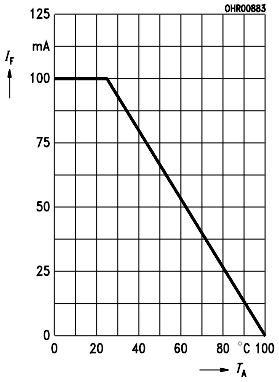
Relative spectral emission
$I_{rel} = f\ (\lambda)$



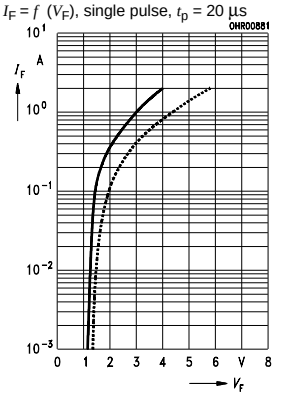
Radiant intensity $\frac{I_e}{I_{e100mA}} = f(I_F)$
Single pulse, t_p = 20 µs



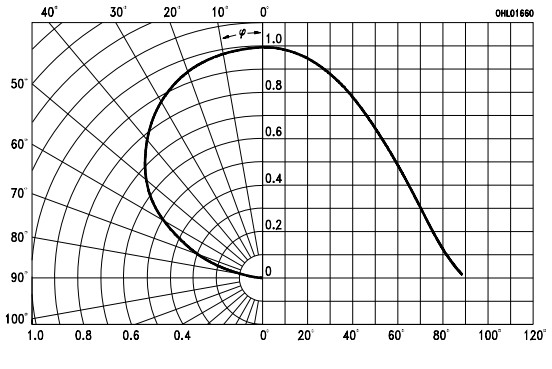
Max. permissible forward current
$I_F = f\ (T_A)$



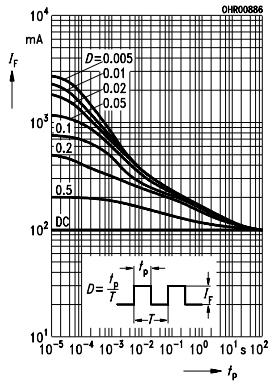
Forward current
$I_F = f\ (V_F)$, single pulse, t_p = 20 µs



Radiation characteristics $S_{rel} = f\ (\varphi)$



Permissible pulse handling capability
$I_F = f\ (t_p)$
duty cycle D = Parameter, T_A = 25 °C



Löthinweise
Soldering conditions

Bauform **Types**	**Tauch-, Schwall- und Schlepplötung** **Dip, wave and drag soldering**			**Reflowlötung** **Reflow soldering**	
	Lötbad-temperatur **Temperature of the soldering bath**	**Maximal zulässige Lötzeit** **Max. perm. soldering time**	**Abstand Lötstelle – Gehäuse** **Distance between solder joint and case**	**Lötzonen-temperatur** **Temperature of soldering zone**	**Maximale Durchlaufzeit** **Max. transit time**
TOPLED	260 °C	8 s	–	260 °C :	10 s :
SIDELED	–	–	–	215 °C Vorheizung Preheating: 150 °C	40 s approx. 1 min.

Literaturverzeichnis

[1] Richard Hartley, Andrew Zisserman, *Multiple View Geometry in computer vision*Cambridge University Press, 2000

[2] Bernd Bruegge, Allen H. Dutoit, *Object Oriented Software Engineering - Conquering Complex and Changing Systems*, Prentice Hall, New Jersey (2000)

[3] Emanuele Trucco, Alessandro Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, 1998

[4] Dave Shreiner, Mason Woo, Jackie Neider, Tom Davis *OpenGL Programming Guide, Fourth Edition*, Addison-Wesley

[5] Karl Schwarz, *Neue Display Technologie im Fighter Cockpit*, Flug Revue Juni 2004, pp.78-83

[6] Roger Y. Tsai *a Versatile Camera Calibration Technique for High Accuracy 3D Machine Vision Metrology using Off-the-Shelf TV Cameras and Lenses,* IEEE Journal of Robotics and Automation, RA-3(4): pp. 323-344, August 1987

[7] Michael Tapper, Phillip J. McKerrow, Jo Abrantes *Problems Encountered in the Implementation of Tsai's Algorithm for Camera Calibration*, Proc. 2002 Australasian Conference on Robotics and Automation, Auckland, 27.-29. November 2002, pp. 66-70

[8] Berthold K. P. Horn *Tsai's camera calibration method revisited*, 2002, MIT Artificial Intelligence Laboratory, Cambridge www.ai.mit.edu/people/bkph/papers/tsaiexplain.pdf

[9] Chinmoy B. Bose and Israel Amir, *Design of Fiducials for Accurate Registration Using Machine Vision*, IEEE Transactions on pattern analysis and machine intelligence, Vol. 12, No. 12, December 1990

[10] Milgram P., Kishino F., *A taxonomy of Mixed Reality and Visual Displays,* IEICE Trans. Information Systems, Vol. E77-D, No. 12, pp. 1321-1329

[11] Azuma R. T., *A Survey of Augmented Reality*, Presence: Teleoperators and Virtual Environments 6 (1997), pp. 355-385

[12] Charles B. Owen, Fan Xiao, Paul Middlin, *What is the best fiducial?*, Media and Entertainment Technologie Labs, Media Interface and Network Design Lab, Michigan State University

[13] Gilles Simon, Andrew W. Fitzgibbon, Andrew Zisserman *Markerless tracking using Planar Structures in the Scene*, Robotics Research Group, Department of Engineering Science, University of Oxford

[14] Gilles Simon, Marie-Odile Berger *Reconstructing while registering: a novel approach for markerless augmented reality*

[15] David Claus, Andrew W. Fitzgibbon *Reliable Fiducial Detection in Natural Scenes*, Department of Engineering Science, University of Oxford

[16] Peiran Liu, Nicolas D. Georganas, Pierre Boulanger, *Designing Real-Time Vision Based Augmented Reality Environments for 3D Collaborative Applications*, University of Ottawa and University of Alberta, IEEE Canadian Conference on Electrical & Computer Engineering, 2002

[17] Youngkwan Cho, Jongweon Lee, Ulrich Neumann *A Multi-ring Color Fiducial System and A Rule-Based Detection Method for Scalable Fiducial-tracking Augmented Reality*, Computer Science Department, Integrated Media Systems Center, University of Southern California, USA

[18] Friedrich, Wolfgang *ARVIKA Augmented Reality für Entwicklung, Produktion und Service* Publicis Verlag, 2004

[19] Hirokazu Kato, Mark Billinghurst, Ivan Poupyrev *ARToolKit Version 2.33 Technical Manual* Hiroshima City University, Human Interface Technology Laboratory, University of Washington, November 2000

[20] Mark A. Livingston, Lawrence J. Rosenblum, Simon J. Julier, Dennis Brown, Yohan Baillot, J. Edward Swan II, Joseph L. Gabbard, Deborah Hix *An Augmented Reality System for Military Operations in Urban Terrain*,Proceedings of Interservice/Industry Training, Simulation & Education Conference (I/ITSEC) 2.-5. Dezember 2002, Orlando, Florida

[21] Tyceryan, M., Navab, N.: *Single Point Active Alignment Method (SPAAM) for Optical See-Through HMD Calibration*, ISAR 2000

[22] Lynda J. Kramer.a, Lawrence J. Prinzel IIIa, Jarvis J. Arthur IIIa, Randall E. Baileya *Pathway design effects on synthetic vision head-up displays* NASA Langley Research Center, 24 West Taylor Street, Hampton, VA, USA

[23] Rüdiger Rodloff, Peter Hecker, *Landeanflug: Klare Sicht bei schlechtem Wetter* http://www2.dlr.de/oeffentlichkeit/nachrichten/97/40-43.pdf

[24] Sielhorst, Tobias *High Accuracy Tracking for Medical Augmented Reality (Hochgenaues Tracking für Erweiterte Realität in der Medizin)*, Technische Universität München, 24. Oktober 2003

[25] Wagner, Martin *Design, Prototypical Implementation and Testing of a Real-Time Optical Feature Tracker*, Technische Universität München, 15. Februar 2001

[26] Bauer, Martin *Design and Implementation of a Module for the Dynamic Combination of Different Position Trackers*, Technische Universität München, 15. Februar 2001

[27] Harrasser, Günther, Reitmeir, Florian *Navigation Aid for Visually Impaired*, Technische Universität München, 25.9.2003 http://wwwbruegge.in.tum.de/pub//DWARF/ProjectNavi/demo-1.pdf

[28] Mumelter, Georg *Implementation of a Highway In The Sky - flight guidance display in a research simulator*, Technische Universität München, 1.12.2003

[29] Klinker, Gudrun *Introduction to Augmented Reality*, Vorlesungsbegleitende Folien WS 2001/2002, Technische Universität München http://atbruegge27.informatik.tu-muenchen.de/teaching/ws01/AR-Vorlesung/index.html

[30] Paelke, Volker *Vorlesung Augmented Reality*, Vorlesungsbegleitende Folien, Institut für Kartographie und Geoinformatik, Universität Hannover

[31] Tom Frey, *Next Generation Controls & Displays*, Lockheed Martin Tactical Aircraft Systems, 30 Juli 1999

[32] *F-16 C/D Flight Manual*, US Air Force, 23. Juli 1984

[33] Ascension Technology Corporation, *laserBIRD2 product brochure*, http://www.ascension-tech.com/products/laserbird2.pdf

[34] Ascension Technology Corporation, *laserBIRD2 Installation and operations guide*, ftp://ftp.ascension-tech.com/manuals

[35] Ascension Technology Corporation, *Flock of Birds product brochure*, http://www.ascension-tech.com/products/flockofbirds.pdf

[36] SONY Corporation *ICX098AK Technical Specification*

[37] PGR FlyCapture Single Lens Digital Video Camera System *User Manual and API Reference Version 1.4*, Point Grey Research, www.ptgrey.com

[38] Intel Corporation, *Open Source Computer Vision Library - Reference Manual*

[39] Intel Corporation, *openCV - Online Reference Manual*

[40] ARPE - Augmented Reality Prototyping for Entertainment Carnegie Mellon University, Entertainment Technology Center http://www.etc.cmu.edu/projects/ar/

[41] Ascension Technology Corporation, *Press Release: Ascension Flock Trackers Selected for F/A-22 Air Combat Simulation*, 17.Mai 2004, Burlington, Vermont http://www.abstech.com/press_room/novidades_rv/FA-22_ACS_release14.pdf

[42] Polhemus Tracking Devices
http://www.polhemus.com/

[43] Wikipedia - Die freie Enzyklopädie
http://de.wikipedia.org/wiki/Augmented_reality
http://de.wikipedia.org/wiki/CAVE

[44] http://www.vrealities.com/hmd.html

[45] Intersense Motion Tracking
http://www.intersense.com

[46] Ascension Technology Corporation
http://www.ascension-tech.com

[47] Neon Helium Productions - OpenGL Tutorial
http://nehe.gamedev.net/