



Fachhochschule Köln
University of Applied Sciences Cologne



Fraunhofer Gesellschaft

Analyse und Entwicklung optischer Trackingverfahren für das dreidimensionale Skizzieren in der virtuellen Produktentwicklung

Masterarbeit
Vorgelegt an der Fachhochschule Köln
Campus Gummersbach

Im Studiengang Medieninformatik

Ausgearbeitet von:

Dipl.-Ing.(FH) Patrick Tobias Fischer

Erster Prüfer: Prof. Dr. rer. nat. Wolfgang Konen
Zweiter Prüfer: Dipl.-Inform. Johann Habakuk Israel

Berlin, 17. April 2008

Inhaltsverzeichnis

Inhaltsverzeichnis	1
Kurzdarstellung der Arbeit	4
Abbildungsverzeichnis.....	5
Tabellenverzeichnis.....	9
Akronyme und Glossar	10
1 Einleitung und Aufbau der Arbeit	13
2 Einführung in das Thema.....	16
3 Produktentwicklungsprozesse	20
3.1 Der schöpferisch kreative Prozess in der Entwicklung.....	20
3.2 Kreativität und Innovation	28
3.3 Die 3D-Skizze und ihre weiteren Rollen im Entwicklungsprozess.....	31
4 Stand der Wissenschaft und Technik	33
4.1 Virtuelle Realität.....	33
4.2 Interaktion in immersiven Umgebungen	36
4.2.1 3D Interfaces	39
4.2.2 3D Displays	40
4.3 Tangible User Interfaces und ihre Vorteile gegenüber herkömmlichen Benutzerschnittstellen ...	41
5 Methodik zum praktischen Teil der Arbeit	46
6 Dreidimensionales Skizzieren in der Cave mit SketchApp	48
6.1 Allgemeine Anforderungen zum 3D-Skizzieren.....	48
6.2 Funktionale Anforderungen an SketchApp.....	49
6.3 Interaktionsraum CAVE	51
6.4 3D-Skizzierwerkzeuge (TUIs)	52

6.5	Schwächen des aktuellen Skizziersystems	54
7	Experimente (Analyse)	56
7.1	Maximale Handgeschwindigkeiten	56
7.2	Mittlere Handgeschwindigkeiten beim 3D-Skizzieren	59
7.3	Betrachtung verschiedener Räume der Cave.....	64
7.3.1	Wahrnehmungsraum des Anwenders in Bezug zur CAVE	65
7.3.2	Analyse des tatsächlichen Interaktionsraum der CAVE	66
8	Ergänzung der 3D-Skizzieranwendung durch optisches Tracking.....	69
8.1	Bestehende Systeme	69
8.2	Hardware.....	71
8.2.1	Kamera, Objektiv und IR Setup	73
8.2.2	Der Marker (Target, Fiducial)	78
9	Allgemeine Schritte des Trackings	86
9.1	Kamerakalibrierung	89
9.2	Entzerrung der Bildebene	90
9.3	Kamerakalibrierung in Bezug auf Opti- und StereoTracker	92
9.4	Ermittlung und Tracking von distinkten Bildmerkmalen.....	92
9.4.1	Color Blob Tracking via Meanshift/Camshift.....	93
9.4.2	Blobdetection	95
9.4.3	Ergebnis.....	96
9.5	Pose Estimation.....	98
9.5.1	Stereoskopische Pose Estimation.....	99
9.5.2	Monoskopische Pose Estimation	100
9.5.3	POSIT als mögliche Lösung zur Pose Estimation	101
9.5.4	SoftPOSIT	102
9.6	Identifikation der Marker	103
9.6.1	3D-2D-3D Korrespondenz	103

9.6.2	Identifikation eines Multipunktmarkers	103
10	Testimplementierungen für die Skizzieranwendung	105
10.1	Bild-Akquisition	106
10.2	Stereoskopisches Tracking – StereoTracker	107
10.2.1	Testimplementierung zur Epipolargeometrie	107
10.2.2	Triangulation	108
10.3	Monoskopisches Tracking – Optitrack	109
10.3.1	POSIT	109
10.3.2	SoftPOSIT	110
11	Ergebnisse und Diskussion	111
12	Zukünftige Arbeit und Ausblick	112
13	Schlusswort	113

Kurzdarstellung der Arbeit

Das Entwerfen mit Stift und Papier ist nach wie vor eine weit verbreitete Methode in den frühen Phasen des Designs in den Domänen der Produktentwicklung, Konstruktion und Architektur; und dies, obwohl die digitale Technologie längst in allen anderen Arbeitsprozessen Einzug erhalten hat. Auch die frühen Phasen der Entwicklung sollen durch Informationstechnologien profitieren können. Neue Benutzerschnittstellen werden derzeit entwickelt, als eine Möglichkeit schnelle, intuitive Skizzen mit den Vorteilen der digitalen Technologie zu verbinden. Dazu gehört die Idee, das zweidimensionale Zeichenpapier um eine weitere räumliche Dimension zu erweitern. Das Skizzieren in dreidimensionalen, immersiven Umgebungen schafft somit neuen Raum für kreatives Handeln und innovative Lösungen.

Diese interdisziplinäre Arbeit beschäftigt sich in umfassender Weise mit der Unterstützung kreativer Arbeit durch dreidimensionales Skizzieren im immersiven Raum und den inhärenten technischen Problemen der Schnittstelle zwischen Mensch und Computer. Der sonst so einfache Akt des Skizzierens mit dem Stift kann nur schwer in den virtuellen Raum transferiert werden. Das Papier als Material verschwindet und Striche entstehen im virtuellen Raum. Die digitale Verbindung zwischen realem Stift und virtuellem Strich, also dem Arbeitsraum des Menschen und dem Rechenraum des Computers, sollte keine Einschränkungen für die Handlung des Skizzierens bewirken.

Optisches Tracking ist in der Lage, kabellose Verbindungen zwischen Mensch und Computer bereit zu stellen. Dabei kann größtmögliche Freiheit des Anwenders erreicht werden, aber technische Probleme entstehen. Eine Analyse der Gesamtsituation und Möglichkeiten des optischen Trackings in immersiven Umgebungen, wie der CAVE, ist Gegenstand dieser Arbeit und öffnet in der virtuellen Produktentwicklung neue Wege für das kreative Entwerfen.

Abbildungsverzeichnis

Abbildung 1 Innovatives Produkt - Senseo Kaffeemaschine.....	17
Abbildung 2 Vision in Product (ViP) Modell	18
Abbildung 3 Sam Hechts Design einer Muji Kaffeemaschine	19
Abbildung 4 Weg zum innovativen Produkt.....	20
Abbildung 5 Konvergenz- und Divergenztrichter	23
Abbildung 6 Der Konstruktionsprozess nach Pahl und Beitz	24
Abbildung 7 Der Reinterpretationskreislauf	29
Abbildung 8 Unterschied Skizze – Prototyp (Buxton 2007)	31
Abbildung 9 Informationsumsatz nach (Pahl & Beitz 1986)	32
Abbildung 10 VR-Dreieck (Grigore Burdea 1994).....	35
Abbildung 11 Transitionen von Informationen zwischen Mensch und Computer	37
Abbildung 12 Abstrahiertes HCI Modell.....	38
Abbildung 13 The Bat von Ware und Jessome.....	39
Abbildung 14 6DOF Maus von Dan Vanolia	39
Abbildung 15 The GyroPoint von Gyration Inc.....	39
Abbildung 16 Nintendo Wii Controller.....	39
Abbildung 17 Haptisches Display	40
Abbildung 18 3D Audio Display	40
Abbildung 19 Propriozeptorisches Display	40
Abbildung 20 3D Visuelles Display	40
Abbildung 21 Semantischer Vergleich zwischen TUI- und GUI-Artefakten (Underkoffler & Ishii 1998)	42
Abbildung 22 GUI versus TUI (Hiroshi Ishii 1997)	42
Abbildung 24 TUI Framework (Brygg Ullmer 2000)	43
Abbildung 23 Vom User Interface zum Tangible User Interface (Hiroshi Ishii 1997)	43
Abbildung 25 Kopplung der Räume	44
Abbildung 26 Enge Kopplung der Räume durch TUI	44
Abbildung 27 Exploratives Vorgehen	47
Abbildung 28 CAVE des Fraunhofer IPK	51
Abbildung 29 Koordinatensystem der CAVE	51
Abbildung 30 Stift Prototyp.....	52

Abbildung 31 Shaper Werkzeug	52
Abbildung 32 3D-Modell des Snapper	52
Abbildung 33 Bezier Werkzeug Prototyp	53
Abbildung 34 Verzerrung des magnetischen Feldes	55
Abbildung 35 Verzerrung in XY-Ebene	55
Abbildung 36 Verzerrung in YZ-Ebene	55
Abbildung 37 Verzerrung in XZ-Ebene	55
Abbildung 38 Bewegung entlang der Y-Achse	56
Abbildung 39 Bewegung entlang der X-Achse	56
Abbildung 40 Bewegung entlang der Z-Achse	57
Abbildung 41 Bilder der Highspeed-Kamera	57
Abbildung 42 Geschwindigkeiten entlang der X-Achse	60
Abbildung 43 Geschwindigkeiten entlang der Y-Achse.....	60
Abbildung 44 Geschwindigkeiten entlang der Z-Achse.....	60
Abbildung 45 Gesamtgeschwindigkeit	61
Abbildung 46 Quader - isometrische Ansicht.....	62
Abbildung 47 Quader - XY-Ebene	62
Abbildung 48 Quader - ZY-Ebene	62
Abbildung 49 3D-Plot der Geschwindigkeiten	62
Abbildung 50 Schriftzug geprägt durch anatomische Gegebenheiten	63
Abbildung 51 Theoretisches räumliches 3D Interaktionsmodell (Previc 1998)	65
Abbildung 52 Mapping des räumlichen Interaktionsmodell auf die CAVE	65
Abbildung 53 Alle Daten von Aufg. 5 (XY-Ansicht).....	66
Abbildung 54 Alle Daten von Aufg. 5 (ZY-Ansicht).....	66
Abbildung 55 Alle Daten von Aufg. 5 (XZ-Ansicht).....	66
Abbildung 56 Alle Daten von Aufg. 1 (XZ-Ansicht).....	67
Abbildung 57 Histogramm aller Koordinaten	68
Abbildung 58 Sichtbereich der Kameras innerhalb der CAVE.....	71
Abbildung 59 Kamerasetup Nr. 1 - Beschränkter Interaktionsraum.....	71
Abbildung 60 Optisches Tracking durch die CAVE Wand.....	72
Abbildung 61 Kamerasetup Nr. 2 - Seitlich ausgerichtetes Kamerasetup	72
Abbildung 62 Kamerasetup Nr. 1 – Seitenansicht.....	72
Abbildung 63 Kamerasetup Nr. 2 - Seitenansicht	72

Abbildung 64 Relative Empfindlichkeit des Sony ICX-204AK	74
Abbildung 65 Relative Empfindlichkeit des Sony ICX-204AL.....	74
Abbildung 67 Low-Cost Visible Light Filter	75
Abbildung 66 Frequenzgang verschiedener Visible Light Filter	75
Abbildung 68 2D-Marker aus ARToolkit.....	78
Abbildung 69 Line-Pattern Marker von Smit (Smit, Rhijn et al. 2006).....	78
Abbildung 70 3D Multipunktmarker	78
Abbildung 71 Projektive Invarianz	79
Abbildung 72 Tracking mit planaren und kollinearen Marker Pattern	79
Abbildung 74 Selbstgebauter aktiver 4-Punkt-Marker	80
Abbildung 75 Aktiver 4-Punkt-Marker mit Lenseflare-Effekt	80
Abbildung 73 Planarer aktiver Multipunktmarker (William Hoff 2000).....	80
Abbildung 76 LED Refraktor	81
Abbildung 77 Probleme beim Blob-Tracking durch direkten Lichteinfall	81
Abbildung 78 Zusammenhang der Koordinatensysteme.....	86
Abbildung 79 Ausschnitt möglicher Lösungswege.....	87
Abbildung 80 Flussdiagramm Optitrack.....	88
Abbildung 81 Projektion in die Bildebene einer Kamera	90
Abbildung 82 Radiale Verzerrung der Bildkoordinaten (Joshua A. Staples 2006)	90
Abbildung 83 Blobtracking mittels Meanshift.....	93
Abbildung 84 Meanshift Suchfensterproblem	94
Abbildung 85 Meanshift Initialisierungsproblem.....	94
Abbildung 86 Beeinflussung des Lichtes auf den Blob.....	94
Abbildung 87 Flussdiagramm des Blobdetector	95
Abbildung 88 Blobdetector Ergebnis	96
Abbildung 89 Blob-Tracking Pipeline	97
Abbildung 90 Modulansicht der einzelnen Trackingschritte	105
Abbildung 91 Testimplementierung zur Epipolargeometrie.....	107
Abbildung 92 Kamerakalibrierung durch Schachbrettmuster	108
Abbildung 93 OptiTrack Screenshot.....	109
Abbildung 94 SoftPosit - Initiale Pose	110
Abbildung 95 SoftPOSIT - 6. Iteration	110
Abbildung 96 SoftPOSIT Test.....	110

Abbildung 97 Skizze im Stehen	114
Abbildung 98 Skizze im Stehen - XZ-Ebene	114
Abbildung 99 Skizze im Stehen - XY-Ebene	114
Abbildung 100 Skizze im Stehen - YZ-Ebene	114

Tabellenverzeichnis

Tabelle 1 Vorgehensweise beim Problemlösen nach VDI Richtlinie 2221	25
Tabelle 2 Vorgehensweise zum Problemlösen nach Cougar	26
Tabelle 3 Ideen zu Interaktionstechniken des 3D-Skizzierens	53
Tabelle 4 Maximale Translationsgeschwindigkeit der Hand	57
Tabelle 5 Maximale Rotationsgeschwindigkeit der Hand	58
Tabelle 6 Mittlere Skizziergeschwindigkeiten	63
Tabelle 7 Kommerzielle Tracking-Systeme	70
Tabelle 8 Vergleich der verwendeten Kameras	73
Tabelle 9 Beispiele für verschiedene Anwendungsgebiete (ImagingSource 2006)	76
Tabelle 10 projektive Invarianz nach (Liere & Mulder 2003)	79
Tabelle 11 Theoretische Blobgrößen	83
Tabelle 12 Vergleich zweier Methoden zur Bildmerkmalsextraktion	97
Tabelle 13 Vergleich iterativer Pose Estimation Algorithmen	102

Akronyme und Glossar

AR	<p>Augmented Reality, dt. verstärkte Realität. Die AR ist ein Raum, der sowohl virtuelle als auch reale Elemente beinhaltet. In der wissenschaftlichen Literatur wird häufig die Definition von Ronald Azuma (Azuma 1995) vorgezogen. Danach ist AR definiert als:</p> <ul style="list-style-type: none">• Kombination von Realem und Virtuellem.• Ein in Echtzeit interaktiver Raum.• Raum mit dreidimensionalen Ausprägungen.
CAVE	<p>Cave Automatic Virtual Environment. Die CAVE ist eine Umgebung der immersiven virtuellen Realität. Meist ein Raum mit mehreren Projektionswänden auf denen Computergrafiken so dargestellt werden können, dass sie den Eindruck von Dreidimensionalität erwecken. Der Name wurde als Referenz zur Allegorie des platonischen Höhlelebens gewählt, in dem ebenso Perzeption, Realität und Illusion thematisiert wurde.</p>
CV	<p>Computer Vision, auch Machine Vision, dt. Maschinelles Sehen bzw. Bildverstehen</p>
DOF	<p>Degrees of Freedom, dt. Anzahl der Freiheitsgrade. DOF sind ein Satz voneinander unabhängiger Größen die eine Verschiebung oder Verformung einer Position eines Körpers oder eines Systems bewirken. Im dreidimensionalen Raum sind normalerweise 6DOF möglich. X,Y und Z Translation (3DOF) und die Rotationen um diese Achsen (3DOF).</p>
extrinsische Kamera Parameter	<p>Extrinsische Kamera Parameter definieren Verhältnisse die außerhalb des Kamerasystems liegen und wichtige Informationen über die Relation der Kamera zu anderen Systemen beinhalten. Das Objektiv ist dabei Teil der Kamera.</p>
Fiducial	<p>Engl. Fachbegriff für eine Marke jeglicher Art. Point- oder Planar-Fiducials werden genutzt, um einen Punkt im Realen Raum zu markieren.</p>
fps	<p>Frames per second. Engl. für die Anzahl der Bilder die innerhalb einer Sekunde aufgenommen werden können.</p>
FOV	<p>Field of Vision, dt. Sichtbereich</p>

HCI	Human Computer Interaction, dt. Mensch-Computer-Interaktion
intrinsische Kamera Parameter	Parameter die eine Kamera betreffen. Hier, horizontale und vertikale Brennweite, optischer Mittelpunkt der Linse und Skalierungsparameter
Immersion	Grad der möglichen kognitiven Eintauchtiefe in einen Sachverhalt.
Marker	Ein Objekt, welches meist an einem anderen Objekt angebracht wird, mit dem Zweck dieses andere Objekt per Kamera zu identifizieren oder andere Berechnungen damit durchführen zu können, wie z.B. Pose Estimation
object model	Dt. Objektmodell. Eine Ansammlung von Positionskoordinaten, die ein bestimmtes geometrisches Objekt parametrisieren und im Speicher des Computers vorliegt. Z.B. in der Architektur ein 3D-Hausmodell.
Pose Estimation	Dt. Lageerkennung bzw. –bestimmung.
px	Maßeinheit für Pixel.
RR	In Anlehnung an den Begriff VR bezeichnet RR die reale uns umgebende physikalische Welt.
Sensorik	Sammelbegriff für die Gesamtheit der Sinnesorgane von Lebewesen.
Skizze	Die "Skizze ist eine nicht unbedingt maßstäbliche, vorwiegend freihändig erstellte Zeichnung (DIN 199-1, 1996,3).
Target	Ähnlich Fiducial. Mit dem Unterschied, dass ein Target nicht zwangsläufig eine zu trackende Marke ist, sondern evtl. das zu trackende Objekt selbst.
Transformation	Der Begriff Transformation dient als Überbegriff der Operationen der Translation und Rotation.
Translation	Positionierung, Verschiebung
TUI	Der Begriff TUI ist zweideutig. Innerhalb dieser Arbeit wird er in TU-Interface und TU-Interaction unterschieden.

TU- Interface	Bezeichnet eine greifbare Benutzerschnittstelle, die sowohl für Eingaben als auch für Ausgaben von Informationen in/aus ein Computersystem dient.
VE	Virtual Environment, dt. virtuelle Umgebung. Ähnlich dem Begriff virtuelle Realität, allerdings mit stärkerem Bezug zum Raum. Als Beispiel eines VE kann die CAVE genannt werden.
Virtualität	Virtualität spezifiziert eine gedachte Entität, die zwar nicht physisch, aber doch in ihrer Funktionalität oder Wirkung vorhanden ist.
VR	Virtual Reality, dt. erdachte Realität.

1 Einleitung und Aufbau der Arbeit

Die vorliegende Arbeit ist im Rahmen eines externen Hochschulaufenthaltes am Fraunhofer-Institut für Produktionsanlagen und Konstruktionstechnik (IPK), Geschäftsfeld virtuelle Produktentwicklung, entstanden. Das Institut macht es sich zur Aufgabe, die Wettbewerbsfähigkeit seiner Kunden und Partner aus produzierenden und dienstleistenden Unternehmen zu unterstützen und zu verbessern. Hierbei spielt die virtuelle Produktentwicklung eine wesentliche Rolle, denn der verschärfte Wettbewerb durch globale Märkte und individualisierte Kundenanforderungen erfordern, den Produktentwicklungsprozess als Gegenstand von Optimierungsmaßnahmen zu betrachten. Die Leistungsfähigkeit eines Unternehmens wird nun einmal maßgeblich durch seine Produktentwicklung bestimmt. Durch immer schnellere Entwicklung innovativer Technologien nimmt daher die Unterstützung von Produktentwicklung eine herausragende Rolle ein. Ziel der Forschungs- und Entwicklungsarbeit ist die Optimierung von Produktentwicklungsprozessen hinsichtlich von Kriterien wie Zeit, Kosten, Innovationsfähigkeit und Ergebnisfähigkeit.

Besonders die beiden letztgenannten Ziele, mit Hinblick auf die frühen Phasen der Produktentwicklung, motivieren diese Arbeit. Oft spielen in diesen Phasen Kreativität und die einhergehende Erschöpfung des Lösungsraumes eine erhebliche Rolle. Die von Johann Habakuk Israel entwickelte dreidimensionale Skizzieranwendung (SketchApp) für immersive Umgebungen bildet hierzu den Grundstein. Aufbauend auf seinen Forschungen und Entwicklungen, bildet diese Arbeit eine Ergänzung seiner Ergebnisse und Weiterführung seiner Arbeit.

Weiterhin wird versucht dem interdisziplinären Charakter des Studienfaches gerecht zu werden, indem nicht nur technische Lösungen präsentiert werden, sondern das Gesamtproblem entfaltet und näheren Betrachtungen unterzogen wird. Dies geschieht etwa durch umfassende Literaturrecherchen zu den Themen Produktentstehung, Kreativität, Human-Computer-Interaction sowie eigens durchgeführten Experimenten, Analysen und Testimplementierungen.

Die Arbeit untergliedert sich in zwei Teile. Nachdem im ersten Teil die theoretischen Überlegungen eingehend dargestellt werden, widmet sich der zweite Teil der Analyse und erkundet mögliche praktische Lösungen zur Umsetzung des dreidimensionalen Skizzierens mittels optischem Tracking.

Teil Eins beginnt mit einer Problembeschreibung (Kapitel 2), indem ein beispielhafter Designprozess in der Produktentwicklung beschrieben wird. Das Kapitel dient zur Sensibilisierung des Themas 3D-Skizzieren im Zusammenhang mit dem schöpferisch kreativen Prozess. Dabei werden besonders die

Prozesse in der Konstruktion und Entwicklung betrachtet um fest zu stellen, in welcher Phase Kreativität erfordert ist und wo Potentiale für innovative Lösungen liegen, die evtl. Unterstützung einer 3D-Skizzieranwendung fordern. Da das Skizzieren eine Art „Denken mit der Hand“ ist, werden auch Kognitionspsychologische Betrachtungen mit einbezogen (Kapitel 3) bei denen deutlich wird, dass denken und skizzieren sehr eng miteinander verknüpft sind und Skizzen ebenso als Kommunikationsmittel eine hervorstechende Rolle spielen.

Im darauf folgenden Kapitel 4 wird der Stand der Wissenschaft dargestellt und Definitionen der Begriffe Virtualität und immersiver Raum gegeben. Zusätzlich werden verschiedene Modelle der Mensch-Computer-Interaktion betrachtet da sie gewissermaßen die Verbindung zwischen Mensch und Computer, hier also Skizzieren und immersive Umgebung erklären. Dabei wird ein eigenes Modell entwickelt, welches besonders die Unterschiede zwischen graphischen Benutzerschnittstellen, die mit Tastatur und Maus bedient werden müssen und greifbaren Schnittstellen, die sowohl Anzeige- als auch Eingabegerät miteinander verschmelzen lassen (sog. TUI), verdeutlicht. Der Begriff der Kopplung, der den Grad der Verbindung zwischen Mensch und Computer bzw. Realität und Virtualität darstellt, wird hier thematisiert.

Nach dem ersten Teil erläutert ein Zwischenkapitel die Vorgehensweise für den Entwurf eines möglichen optischen Tracking-Systems.

Teil Zwei dieser Arbeit analysiert die konkrete Situation des dreidimensionalen Skizziersystems und beginnt mit einer Beschreibung des derzeitigen Systems (Kapitel 6). Es werden Anforderungen, Aufbau und derzeitig vorhandene Probleme, die vor allem durch das verwendete magnetische Tracking System entstehen, erläutert. Die verschiedenen Zeichenwerkzeuge und Interaktionstechniken werden beschrieben.

Es folgt Kapitel 7 mit einer Beschreibung der Ergebnisse verschiedener Experimente, die dazu dienen Grenz- als auch Normalsituationen beim dreidimensionalen Skizzieren zu verstehen. Auch die Beschränkungen des immersiven Raums werden erkundet, in dem genau analysiert wird welche Bewegungsmuster bestehen. So werden Daten von 24 Probanden, welche je fünf Skizzieraufgaben bewältigt haben, dazu herangezogen, um herauszufinden, wie die CAVE Handlungen beschränkt oder auch motiviert.

Kapitel 8 thematisiert die Ergänzung der bestehenden Anwendung SketchApp und präsentiert gewonnene Erkenntnisse bezüglich zu verwendender Hardware, die ein optisches Tracking ermöglichen sollen. Zusätzlich werden konkrete Empfehlungen zu Kamera, Objektiv und Marker gegeben.

In Kapitel 9 werden Erkenntnisse zu den einzeln notwendigen Schritten des optischen Trackings zusammengefasst und ein möglicher, neuer Weg des monoskopischen Trackings, unter Verwendung des sog. POSIT-Algorithmus, entwickelt. Dabei wird das stereoskopische Tracking ebenso berücksichtigt, da einige Methoden zum Tracking sowohl bei der Verwendung von einer, als auch von zwei Kameras genutzt werden kann.

Begleitend zu dieser Arbeit wurde ein eigenes Trackingsystem entwickelt, welches vor allem die verschiedenen Möglichkeiten der 3D-Pose-Estimation implementiert und testet. Besonders der Gesamtvorgang des Tracking war während der Entwicklungen von Interesse. Kapitel 10 erläutert einige Testimplementierungen, die im Zuge der Exploration möglicher Wege des optischen Trackings entstanden sind.

Die Arbeit schließt mit einer Zusammenfassung der wichtigsten Ergebnisse und gibt eine Empfehlung über die Richtung weiterer Entwicklungen bezüglich optischer Tracking-Systeme für eine CAVE.

2 Einführung in das Thema

In diesem Kapitel soll zunächst der Entstehungsprozess von Produkten erläutert werden. Dies dient vor allem dazu, dem Leser ein Verständnis über die potentiellen Einsatzmöglichkeiten eines dreidimensionalen Skizziersystems im Produktentstehungsprozess zu geben sowie dem praktischen zweiten Teil der Arbeit den nötigen Kontext zu gewähren. Es wird versucht die Begriffe Innovation (lat. novus, für neu bzw. lat. innovatio, etwas neu Geschaffenes) und Kreativität (lat. creare, für etwas neu schöpfen, etwas erfinden, etwas erzeugen, herstellen, aber auch auswählen) zu definieren bzw. voneinander ab zu grenzen und diese in den Prozess der Produktentstehung einzuordnen, da sie potentiell über Erfolg und Misserfolg eines Produktes auf dem Markt entscheiden können. Ziel ist hierbei die Sensibilisierung des Lesers gegenüber dem Prozess des Problemlösens und das Erkennen etwaiger Chancen, involvierte schöpferische Prozesse mittels Technologie bzw. VR-Technologie zu unterstützen.

Ein Entwicklungsprozess gliedert sich im Allgemeinen in mehrere Phasen. Auch bei der Produktentwicklung ist dies nicht anders. Zu Beginn steht hierbei meist eine vage Idee über das zu entwickelnde Produkt im Vordergrund. Von dieser Idee ausgehend, folgen eine Analyse-, eine Konzept-, eine Strategieentwicklungsphase, eine Phase der Realisierung (in der das physische Produkt entsteht), eine Testphase und die Produkteinführungsphase. Bereits in dieser kurzen Auflistung der Phasen wird klar, dass ein Produkt aus dem immateriellen Raum – nämlich der Gedankenwelt – heraus entsteht und mit zunehmender Reife der stofflichen Verwirklichung immer näher kommt.

Von besonderem Interesse in dieser Arbeit, sind die frühen Phasen der Produktentstehung. Also der Punkt, an dem die vagen Ideen über ein Produkt an Gestalt gewinnen. Der Punkt, an dem sich aus den vielen Ideen für ein einzelnes Produkt ein Ganzes herausbildet. Als Beispiel für diese Phase der Produktentstehung sei hier die Entwicklung einer neuen Kaffemaschine genannt. Nehmen wir an, wir stünden davor eine solche Maschine neu zu entwickeln. Das Endprodukt wird die Senseo Kaffeemaschine von Philips sein. Wir wären also im Begriff einige schöpferisch-kreative Arbeitsschritte zu durchlaufen. Z.Z. stehen wir aber noch am Anfang und wissen noch nicht, wie dieses fertige Produkt aussehen wird, da es schließlich gerade entwickelt werden soll. Dennoch haben wir einige vage Vorstellung davon, welche Abhängigkeiten sich bezüglich des „Kaffee-Machens“ ergeben.

Z.B. wissen wir, dass man Kaffepulver braucht und heißes Wasser, welches zum brühen den Kaffes wichtig ist. Daraus ergeben sich weitere Faktoren die zu berücksichtigen wären, wie der Wasserbehälter, der für das Wasser gebraucht wird ebenso wie ein Erhitzungssystem, welches unser Wasser auf Brühtemperatur bringt. Vielleicht besteht sogar schon die Idee der sog. Kaffepads, durch die der Kaffee laufen muss. Auch eine Tasse mit einer gewissen Höhe spielt eine Rolle, in die der Kaffee letztendlich hinein muss.



Abbildung 1 Innovatives Produkt - Senseo Kaffeemaschine

Abgesehen von diesen eher technisch-physikalischen Voraussetzungen für das Brühen von Kaffee, gibt es noch weitere Faktoren die auf den Designprozess der neuen Kaffeemaschine wirken. Diese sind etwa Geschäftsführung mit ihrer Idee der Firmenidentität (Corporate Identity) und -maximen (Mission Statement), das Marketing mit den Interessen von Innovativen und attraktiven Produkten, die Kalkulation mit ihrem Wunsch nach kostengünstiger Produktion, etc. Trotz aller dieser Stakeholder, die jeder für sich bestimmte Interessen verfolgen und somit die Lösungsfindung in ein mehrdimensionales Problem transferieren bzw. in mehrere „Räume des Problems“ aufteilen, geht es darum ein wirklich *innovatives* Produkt zu entwickeln, welches besonderen Anklang beim Nutzer findet. Um eben diese Besonderheit und wie man diese erreicht geht es. Terry Winograd (Stanford, CA) beispielsweise, betont dies immer wieder. Laut ihm entstünden viele Entwicklungen unter dem Aspekt der Realisierbarkeit und der Durchführbarkeit, folgen also technischen und geschäftlichen Kriterien. Der Nutzer und seine Bedürfnisse hingegen würden oft zu wenig berücksichtigt. Dabei sei es die Schnittmenge aus allen drei Bereichen die interessantes Design ausmachen (wp 2008).

Wie dem auch sei, während des Designprozesses sollten die Anforderungen der Stakeholder zwar bekannt sein, aber nicht zwangsläufig in die Entwürfe mit einfließen. Es ist wichtig das kreative Potential der Designer nicht vollends durch irgendwelche Vorgaben ein zu schränken. Aber andererseits muss dem Designer bewusst sein, dass das Produkt mehr ist, als die Einzelteile des Produktes. Somit obliegt es seiner Aufgabe die Einzelteile innerhalb eines Produktes integrierbar zu halten. Das sich evtl. ein Problem aus einem anderen Problemraum (z.B. Recyclefähigkeit des Produktes) auf seine Lösungsmöglichkeiten (auch Lösungsraum genannt) auswirkt, stellt die besondere Schwierigkeit beim

Design dar. Und schließlich ist das Ganze etwas ganz Spezielles, was zu kreieren ist. Um es mit Aristoteles Worten zu sagen: „Das Ganze ist mehr als die Summe seiner Teile.“

Einige Forscher aus der Design Domäne versuchen dem Besonderen des Ganzen, welches zuweilen als Emergenz- oder Synergieeffekt bezeichnet wird und im fertigen Produkt immanent vorliegt, systematisch auf die Spur zu kommen und es zu entschlüsseln. Sie entwickeln Modelle, um solche positiven Emergenzeffekte in Produkten zu maximieren. Hierbei werden weitreichendere Betrachtungen, als die auf die bloßen Produkteigenschaften bezogenen, durchgeführt. D.h., am Beispiel der Kaffeemaschine verdeutlicht, dass es keinesfalls hinreichend ist, nur ein Gehäuse als Abdeckung des Gerätes zu entwerfen, die alle technischen Einzelteile einfach nur verbirgt. Gemäß dem altbekannten Gestaltungsgrundsatz aus Design und Architektur – „Form follows function“ – könnte man dies wohl fälschlicherweise annehmen, aber die Form muss noch weiteren Funktionen folgen, als die der oben genannten. Gemeint sind hier z.B. die *Interaktion* zwischen Mensch und Maschine (z.B. unterstützt durch geeignete Affordances) sowie der implizit gegebene *Kontext* der Kaffeemaschine zu ihrer Umwelt. Paul Hekkert geht sogar soweit und sagt, dass ein Design des Kontextes am Anfang eines jeden Designprozesses stehen sollte (Paul Hekkert 2003). Hierzu wählt der Designer ein Set von Faktoren, wie Status quo, Trends, prinzipielle Funktionsweisen, Einsatzort etc. die den, für ihn geltenden Kontext bilden. Davon ausgehend, kann er verschiedene Interaktions-Szenarien zwischen Maschine und Mensch entwickeln, und anschließend weitere Eigenschaften des Produktes designen. Diese Vorgehensweise nennt Hekkert *Vision in Product (ViP)* (siehe Abbildung 2).

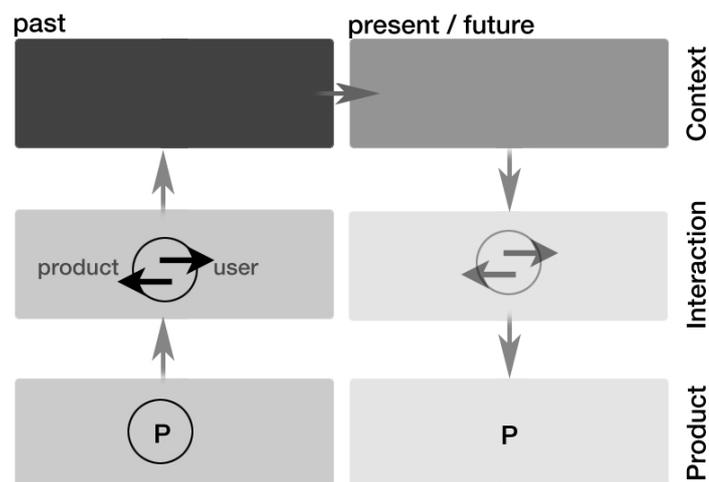


Abbildung 2 Vision in Product (ViP) Modell

Gerade der Kontext für ein Produkt scheint zunehmend in das Bewusstsein der Designer zu treten und prospektiv im Designprozess begegnet zu werden, wie das ViP-Modell von Hekkert zeigt. Bestätigen tut

dies auch eine von Sam Hecht entworfene Kaffeemaschine für die Firma Muji, welche deutliche Anpassung an den *räumlichen Kontext* aufweist. Seine Kaffeemaschine passt perfekt in die Ecke einer Küche, wo sie nach empirischen Untersuchungen auch tatsächlich oft ihren Wirkungsort findet, kann aber ebenso gut mitten auf dem Tisch platziert werden und macht dabei weiterhin „eine gute Figur“. Andere Kaffemaschinen haben dagegen eine eher minder sehenswerte Rückseite, da sie sowieso meist an einer



Abbildung 3 Sam Hechts Design einer Muji Kaffeemaschine

Wand stehen oder wie gesagt in einer Ecke ihren Platz finden. Zudem hat sich Hecht auch mit der Interaktion näher beschäftigt. So kann die Kanne der Maschine sehr leicht von unterschiedlichen Richtungen entnommen werden und hält den Kaffee dank Doppelwandtechnik für viele Stunden warm. Hechts Kaffeemaschine bietet also über den Weg der persönlichen Auseinandersetzung mit dem Kontext und der Interaktion eine neue Qualität gegenüber anderen Kaffemaschinen, was als *Innovation* gedeutet werden kann.

Hier wird klar, dass Vorgehensmodelle, auch wenn ein Designer solche üblicherweise für einschränkend empfindet, Anregungen für innovative Produkte geben können. Auch die Senseo Kaffeemaschine bietet einige unübersehbare Innovationen, sowohl in technischer, funktionaler, interaktiver, kontextueller, als auch in ästhetischer, emotionaler Hinsicht. Ob dies im Einzelnen der ViP-Methode zu verdanken ist, kann und soll hier nicht geklärt werden (zumindest versucht Hekkert dies glaubhaft zu machen). Viel mehr jedoch interessiert es, wie „wir“ zu einem solch formidablen Ergebnis gekommen sind und wie man diesen Weg zum Ergebnis evtl. durch kreativitätsfördernde, digitale Systeme unterstützen kann. Um dies zu verstehen, ist es zunächst nötig, sich einen Entwicklungsprozess näher an zu schauen.

3 Produktentwicklungsprozesse

Wie aus der Einführung hervorgeht, gilt es für Unternehmen als erstrebenswert, innovative Produkte zu entwickeln. Es wurde anhand eines Beispiels gezeigt, dass Produktdesigner bestimmte Methoden besitzen, die ihnen dabei helfen innovative Produkte designen zu können, wie beispielsweise durch Einsatz des ViP-Modells. Dabei ist das ViP-Modell im eigentlichen Kern ein Vorgehensmodell zum lösen eines essentiellen Problems. Nämlich, wie kann ein innovatives Produkt geschaffen werden? In der Konstruktion und Produktentstehung ist Innovation ein Resultat mehrerer Schritte für die zuweilen ein kreatives Vorgehen notwendig ist.

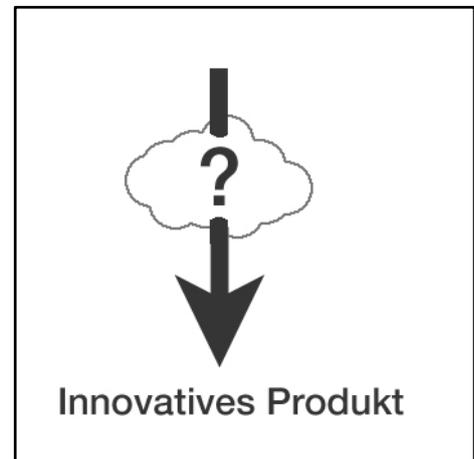


Abbildung 4 Weg zum innovativen Produkt

Pahl und Beitz bieten in ihrem Buch „Konstruktionslehre“ einen guten Überblick über die basalen Schritte methodischen Vorgehens zur Problemlösung. Und da es sich bei der Produktentstehung um ein solches, ständiges Problemlösen handelt, dürfte jeder Ingenieur, Konstrukteur und Designer schon am eigenen Leibe erfahren haben. Im Folgenden werden also kurz Vorgehensweisen zum Problemlösen beschrieben, um eine klarere Vorstellung über die Situation der Entwickler, Designer, Ingenieure oder Architekten bei der Entstehung eines Produktes zu bekommen. Es kann somit genau lokalisiert werden, wo Innovation entsteht und wo Kreativität der Innovation zuträglich ist (Kapitel 3.1). Was genau unter den Begriffen Kreativität und Innovation zu verstehen ist (Kapitel 3.2) und wie diese beiden Begriffe zusammenhängen soll ebenfalls versucht werden zu klären (Kapitel 3.3). Anschließend können daraus Anforderungen gegenüber kreativitäts- und Innovationsfördernder Systeme, wie etwa die 3D-Skizzieranwendung für die Cave, abgeleitet und formuliert werden (siehe Kapitel 6.1).

3.1 Der schöpferisch kreative Prozess in der Entwicklung

Definitionen des Phänomens bzw. Konzeptes der Kreativität wurden in vielen Bereichen der Wissenschaft versucht und noch immer ist der Begriff nicht vollends verstanden. Beschäftigt man sich jedoch etwas näher mit diesem Begriff, so ist es bemerkenswert, dass immer wieder genetische Anlagen wie die *intuitive Begabung* und *Widerstandsfähigkeit* dafür verantwortlich gemacht werden. Somit spielen die individuellen Möglichkeiten einer einzelnen Person eine besondere Rolle. Nach Holm-Hadulla sind die Motivationsfaktoren einer kreativen Person das intrinsische Interesse an der Sache selbst, Neugier und Selbstwertsteigerung. Hierbei kann sich der Schöpferisch-Aktive zeitweilig sogar in einem

besonderen Bewusstseinszustand des sog. Floatings oder auch Flow befinden, in dem er sogar vorübergehend den Verlust des Zeitbewusstseins erleidet. Dem schöpferischen Akt wird weiterhin ein intuitiver Denkprozess zwischen Unterbewusstsein und Bewusstsein, der stark einfallsbetont ist, zugeschrieben. D.h., eine Erkenntnis tritt plötzlich in das Bewusstsein. Pahl und Beitz beschreiben diese Kreativität, als die *primäre Kreativität*.

Die Nachteile der *primären Kreativität* liegen allerdings auf der Hand:

- Der richtige Einfall kommt selten zum gewünschten Zeitpunkt, da er nicht erzwungen oder erarbeitet werden kann.
- Das Ergebnis ist stark von der Veranlagung und Erfahrung der Person abhängig.
- Die Gefahr besteht, dass sich die Lösungen nur innerhalb eines fachlichen Horizonts des Bearbeiters vor allem durch dessen Vorfixierung einstellen.

Glücklicherweise erlauben formale Kreativitätstechniken diese Nachteile zu übergehen, denn die „Alltägliche Kreativität“ (Holm-Hadulla 2005) kann durch sie geweckt werden, oder sogar erlernt werden (Knieß 2005). Pahl und Beitz nennen dieses strukturierte, diskursive Denken *sekundäre Kreativität*.

In wie weit Aussagen wie „Das menschliche Denken ist nicht intuitiv, sondern diskursiv.“, von Immanuel Kant gegen Intuition und somit gegen die *primäre Kreativität* sprechen, sein hier nicht weiter verfolgt, da der Begriff der Intuition von Kant als eine erkenntnisfähig, *fühlende Emotion* verstanden wird und somit nicht dem hier propagierten Verständnis von intuitivem Denken zuträglich ist. Zudem gibt es einige Forschungsergebnisse, die darauf deuten, dass intuitiv getroffene Entscheidungen manchmal zu besseren Ergebnissen führen, als mit bewusstem Verstand. Die Theorie dahinter: Das Unbewusste ist in der Lage, weitaus mehr Informationen zu berücksichtigen als das Bewusstsein, das zwar sehr präzise ist, jedoch nur wenige Informationen prozessieren kann.

Aus den oben angesprochenen Nachteilen der *primären Kreativität* die stark einfallsbetont (intuitiv) abläuft, dürfte ersichtlich sein, dass nur ein methodisches Vorgehen mit diskursivem Anteil und entsprechenden unterstützenden und unterschiedlichen Lösungsmethoden, Kreativität, als *sekundäre Kreativität* generieren kann.

Der wesentliche Anteil der Arbeitsweise beim Lösen von Problemen besteht nach Pahl und Beitz in einem Vorgang der *Analyse* und in einem Vorgang der *Synthese*, welche zirkulär durchlaufen werden. Beide Schritte laufen in Arbeits- und Entscheidungsschritten ab. Zu analysieren bedeutet Wesentliches von Unwesentlichem zu trennen und das Problem in einzelne Teilprobleme zu zerlegen und somit für diskursive Lösungssuche vorzubereiten, wie z.B. Strukturanalyse, Schwachstellenanalyse, etc. Die

Synthese ist das (denkerische) Verbinden und Verknüpfen von (Lösungs-)Elementen mit insgesamt neuen Wirkungen und das aufzeigen einer zusammenfassenden Ordnung in etwa einem System. Es werden also einzelne Erkenntnisse oder Teillösungen zu einem funktionierenden Gesamtsystem zusammengefasst. Die Eigenschaft des funktionierenden Gesamten steht hierbei im Vordergrund, wobei natürlich Informationen aus der Analyse mit einfließen.

Neben der Analyse und der Synthese gibt es im Denkprozess weiterhin die Möglichkeit der *Abstraktion*. Sie wirkt vor allem komplexitätsreduzierend und ermöglicht dem Denkenden von bestimmten Teilproblemen, die sich aus der Analyse ergeben haben, zwar ein mentales Modell zu formen und somit gedanklich zu verarbeiten, aber gleichzeitig auch die Einzelheiten des Teilproblem zu „ignorieren“ und nur die Schnittstellen zu kennen. Dennoch bleibt das Problem auf abstrakte Weise formuliert und ist alleine deswegen nicht dem Zufall im Lösungs- bzw. Entwicklungsprozess überlassen, was vor allem positiv zu bewerten ist.

Neben der Analyse, Synthese und Abstraktion bestehen noch weitere anwendbare Methoden des diskursiven Problemlösens (Pahl & Beitz 1986, S. 41):

- Methode des gezielten Fragens
- Methode der bewussten Negation und Neukonzeption
- Methode des Vorwärtsschreitens (divergentes Denken)
- Methode des Rückwärtsschreitens (konvergentes Denken)
- Methode der Faktorisierung
- Methode der systematischen Variation

Leider kann in diesem Rahmen nicht auf alle dieser Methoden eingegangen werden. Situationen oder Systemfunktionen in denen diese Methoden aktiv unterstützt werden, können aber leicht abgeleitet werden. Dennoch soll an dieser Stelle kurz auf das *divergente* und das *konvergente* Denken eingegangen werden. Der Grund hierfür ist die Erwähnung dessen, von einem führenden Forscher und Designer im Bereich der Mensch-Computer-Interaktion (HCI) - Bill Buxton - welcher sich langjährig mit Eingabemöglichkeiten bzw. -schnittstellen und Interaktionstechniken für Computersysteme im Bereich der Konstruktion, beschäftigt hat. Er beschreibt in (Buxton 2007), wie nach dem Modell von Laseau (Laseau 1980) diese beiden diametralen Denkmethoden für den Entwicklungsprozess zu betrachten sind (siehe Abbildung 5) und wo er genau den Einsatz von Kreativität im Design- bzw. Entwicklungsprozess sieht.

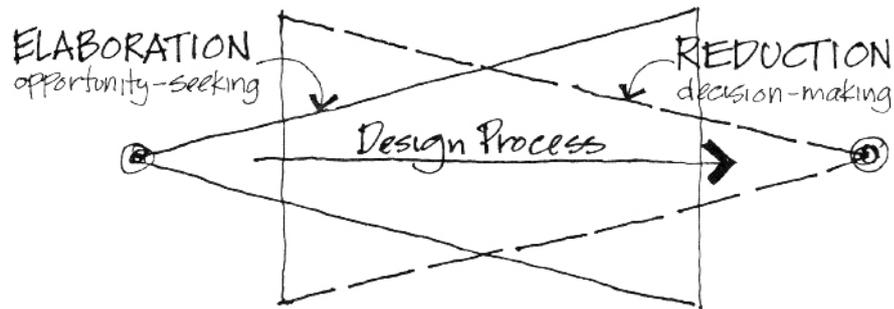


Abbildung 5 Konvergenz- und Divergenztrichter

Das Modell zeigt den divergenten und konvergenten Denkprozess jeweils als eine Art Trichter. Buxton beschreibt diese wie folgt:

„Design is choice, and there are two places where there is room for creativity: (1) the creativity that you bring to enumerating [generating] meaningfully distinct options from which to choose and (2) the creativity that you bring to choosing the criteria, or heuristics, according to which you make your choices.“ (Buxton 2007)

Aus dieser Definition gehen zwei fundamentale Aspekte für den Designprozess hervor. Zum einen wirkt die Kreativität als generatives Mittel (ähnlich dem divergenten Denken), auf der anderen Seite aber als reduzierendes Mittel (ähnlich dem konvergenten Denken). Der kreative Entstehungsprozess ist somit also ein Erarbeiten oder Suchen von Möglichkeiten, als auch ein Reduzieren dieser Möglichkeiten durch Treffen von Entscheidungen. Über (Pugh 1990) kommt auch Buxton in seinem Buch auf die Begriffe *Analyse* und *Synthese*, welche dem konvergenten und divergenten zugeordnet werden und damit die Relevanz der Kreativität im Entwicklungsprozess bestätigt.

In der Literatur sind weiterhin noch detailliertere Schritte bezüglich des Problemlösens zu finden. Polya (Polya 1945) sieht drei wesentliche Phasen,

1. Verstehen bzw. Formulieren eines Problems
2. Generieren von Lösungshypothesen und Auswahl der alternativen Hypothesen
3. Testen der Lösung und deren Bewertung

welche bei näherer Betrachtung sechs Phasen integriert in drei Punkten sind. Für Klauer sind die wesentlichen Phasen (Klauer 1995),

1. Orientierungsphase als planlose Suche bzw. „chaotisches Ausprobieren“
2. Einer Phase des lokalen Planens, in der die Problembearbeiter zumindest kleine Sequenzen von Maßnahmen planen können
3. Einer Phase des globalen Planens, die ein zielgerichtetes Vorgehen ermöglichen

wobei uns diese Phasen eher als ein Problemlöseprozess im Team vorgeschlagen werden und somit mehr an Vorgehensweisen zur Teamarbeit erinnern, wie z.B. das von Tuckman eingeführte Modell des „*Forming, Storming, Norming, Performing, (Adjourning)*“ (Tuckman 1965) und somit weniger Aufschluss über die einzelnen individuell zu durchlaufenden Phasen gibt. Dennoch sei es hier erwähnt, da Indizien darauf hinweisen, dass es evtl. auch eine Art *Gruppenkreativität* in gruppendynamischen Prozessen gibt, die evtl. als neue Möglichkeiten der kreativen Unterstützung im Immersiven Raum interessant sind.

Das klarste und deutlichste Modell zur Veranschaulichung des Problemlöseprozesses bieten wiederum Pahl und Beitz in (Pahl & Beitz 1986). Sie bieten ein Grundschema mit sechs aufeinander folgenden Stufen, die in iterativer bzw. rückgekoppelter Weise durchlaufen werden (siehe Abbildung 6). Es beginnt mit der Konfrontation, einer Gegenüberstellung von Problemen und bekannten oder (noch nicht) bekannten Realisierungsmöglichkeiten. Wie stark eine solche *Konfrontation* ist, hängt sehr stark vom Wissen, Können und der Erfahrung des Konstrukteurs (*Information*) und dessen Domain, in der er tätig ist, ab. In jedem Falle ist aber eine Information über nähere Aufgabenstellung, Bedingungen, mögliche Lösungsprinzipien und bekannte ähnliche Lösungen nützlich, so Pahl und Beitz. Zum wiederholten Male wird klar, dass der Konstruktionsprozess dem Designprozess sehr ähnelt. Denn auch für das ViP-Modell von Hekkert (vgl. Abbildung 2) spielt Konfrontation bzw. Information – besonders über vormalige

Lösungen (linke Seite des Modells) – eine wichtige Rolle.
 Die Definition der wesentlichen Probleme (Wesenskern der Aufgabe) auf abstrakter Ebene ermöglicht es, die Zielsetzung festzulegen und die wesentlichen Bedingungen zu beschreiben. Die Formulierung in abstrakter Weise ist in sofern notwendig, als dass eine zu konkrete Formulierung bestimmte Lösungswege vordefinieren würde und ein Durchbruch zu außergewöhnlichen Lösungen verhindert.

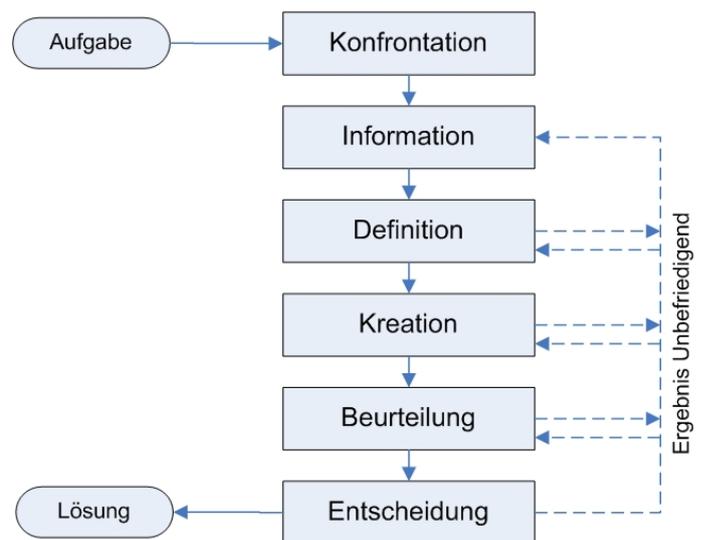


Abbildung 6 Der Konstruktionsprozess nach Pahl und Beitz

Auf den Designprozess bezogen, verifiziert sich hier die in Kapitel 2 „

Einführung in das Thema“ getroffene Aussage, dass dem Designer zwar die Intensionen der verschiedenen Stakeholder bekannt sein sollten, er deren Lösungsvorgaben nicht zwangsläufig berücksichtigen sollte, um sich nicht in seinem möglichen Lösungsraum künstlich beschränken zu lassen. Die Methode der Abstrahierung hilft ihm hierbei.

Die eigentliche schöpferische Phase in Abbildung 6 ist die *Kreation*, in der Lösungsideen nach verschiedenen Lösungsmethoden entwickelt und (methodisch) variiert oder kombiniert werden. Der hier verwendete Begriff der Lösungsmethoden, hinterlässt unter Umständen den Eindruck, dass auch die Phase der Kreation nur wenig mit Kreativität zu tun hat, da Kreativität als Methodisch zu planend verstanden wird. Viel mehr geht es aber darum, Lösungsmethoden als kreativitätsfördernde Hilfsmethoden zu sehen, welche Kreativität im Sinne der sekundären Kreativität entstehen und in diesem Schritt anwenden lassen. Die Stufe der Kreation ist Übrigends auch der Punkt, an dem auch Buxton den Einsatz von Kreativität sieht, wie aus dem obigen Zitat hervorgeht.

Die Vielzahl der Variationen, die wir nach der Kreation erhalten, erfordert eine *Beurteilung*; die Grundlage zur Entscheidung für die Anscheinend bessere Variante. Die *scheinbar* bessere Variante ist genau diese, die hinsichtlich der Zielsetzung befriedigend oder die als „am Besten“ erscheint. Ob sie dies wirklich ist, kann letztendlich nicht mit Bestimmtheit gesagt werden. Es spielt also immer noch die erwähnte Erfahrung bzw. wenn diese fehlt die Intuition eine tragende Rolle.

Liefert eine der Stufen aus Abbildung 6 ein unbefriedigendes Ergebnis, so kann zu einer oder mehrerer vorangegangener Stufen zurückgesprungen werden, wie es bei modernen, zirkulären Entwicklungsprozessen nun einmal der Fall ist.

- Problemanalyse
- Problemformulierung
- Systemsynthese
- Systemanalyse
- Beurteilung
- Entscheidung

Eine weitaus förmlichere Beschreibung zur Vorgehensweise beim Problemlösen, als die bereits vorliegende, bietet die Richtlinie 2221 des Verbands Deutscher Ingenieure (VDI). Auch sie postuliert zyklische Schritte als eine allgemeine Handlungsstrategie zum Problemlösen. Tabelle 1 zeigt die einzelnen Schritte.

Tabelle 1 Vorgehensweise beim Problemlösen nach VDI Richtlinie 2221

Bei Betrachtung der Schritte der VDI Richtlinie 2221 fällt auf, dass dem kreativen Aspekt der Arbeit keinen Platz im Entstehungsprozess eingeräumt wird. Kreativität wird hinter Begriffen wie Synthese und Analyse versteckt, um somit nicht den Eindruck zu erwecken, hier seien kreative Designer am Werk. Man könnte der Richtlinie des VDI sogar bewusstes leugnen der kreativen Tätigkeit beim Konstruieren vorwerfen.

Aber wie bereits durch Buxton verdeutlicht wurde, ist Kreativität in Analyse und Synthese immanent. Somit beinhaltet dieses Modell viel mehr Stufen der Kreativität als bereits Genannte.

Cougers Ergebnis, nachdem er etwa 22 verschiedene Vorgehensmodelle zum kreativen Problemlösen verglichen hatte, beinhaltet folgende fünf Schritte aus Tabelle 2 (Couger 1996):

- Opportunity, delineation, Problem definition
- Compiling relevant information
- Generating Ideas
- Evaluating, prioritizing Ideas
- Developing an implementation plan

Tabelle 2 Vorgehensweise zum Problemlösen nach Cougar

Auch sein Prozess ähnelt dem von Pahl und Beitz sehr stark, misst aber leider die explizite Stufe der Entscheidung. Es wird lediglich eine Priorisierung vorgeschlagen, die natürlich nicht sofort in einem Implementierungsplan enden kann, ohne vorherige Entscheidung. Es ist wirklich schade, dass er diesen Punkt nicht explizit nennt, denn die Entscheidung ist von so enormer Wichtigkeit, dass sie gar nicht vergessen werden darf. Denn ohne sie ist kein Weiterkommen. Wird z.B. eine Entscheidung immer wieder vertagt, so ist nicht klar, in welche Richtung weiter gearbeitet werden soll.

Die Lösungsmöglichkeiten werden immer mehr und es würde dazu führen, dass man nach einiger Zeit plötzlich über Gott und die Welt diskutiert und somit alle potentiell mögliche Lösungen exploriert und sich in ihnen verliert. Zum wiederholten Mal wird klar, dass einfache Mechanismen wie Elaboration und Reduktion (vgl. Abbildung 5) notwendig sind. Genauso wie Analyse und Synthese, welche wiederum zugrundeliegenden Denkmöglichkeiten wie divergentes und konvergentes Denken erfordern.

Es fällt auf, dass gerade Modelle zum kreativen Problemlösen die immense Wichtigkeit der Entscheidung und der *Problemdefinition* nicht erkennen. So beschreibt z.B. Dewy and Wallace (1926) die notwendigen Schritte des „Creative Problem Solving“ als:

- Preparation
- Incubation
- Inspiration
- Verification

Osborn und Barnes (1972) Modell beinhaltet dagegen schon das „problem finding“, allerdings noch nicht die Stufe der Entscheidung.

- Fact finding
- Problem finding
- Idea Finding
- Solution finding
- Acceptance finding

Eine gänzlich anderen Erklärungsansatz, in dem wir von der Aufgabe zum Ziel gelangen beschreibt Pierre Sachse in (Sachse 2002). Er definiert nicht die notwendigen Stufen, die für das Lösen von Aufgaben erforderlich sind, sondern beschreibt, was uns als Problemlösenden davon abhält überhaupt Probleme lösen zu können. Seine Problemklassifikationen (nach Dörner und Fricke) lautet:

- Interpolationsbarriere (einfache Variantenreduktion)
- Synthesebarriere (Konstruktion bei sich widersprechenden Zielen, enge Lösungsräume)
- Dialektische Barriere (Ziel unbekannt, Anforderungen können nur nach und nach ermittelt werden)
- Synthetische und dialektische Barriere (Neukonstruktion, z.B. Anforderung und Lösungen sind unklar)

Aus den obigen Ausführungen wird deutlich, dass Designer und Konstrukteure ähnlichen Prozessen zur Problemlösung unterliegen. Beide haben das Problem, Probleme lösen zu müssen. Eben dieser Problemlösungsprozess verbindet beide Disziplinen. Zudem ist das Problemlösen eine so menscheigene Fähigkeit, dass der Prozess im Grunde ein Ergebnis unserer eingepflichten Denkmöglichkeit selbst ist. Dabei stehen uns, wie kurz angedeutet wurde, verschiedene Möglichkeiten (wie zu denken ist) offen.

Um nicht gänzlich in Erkenntnistheoretischen Philosophien unter zu gehen, sei noch eine Auffälligkeit zwischen dem Design-Lösungsprozess und dem Konstruktions-Lösungsprozess genannt. Lösungen basieren auf Information. Wie schwer eine Aufgabe sein mag, hängt stark von der zu Verfügung stehenden Information ab. Somit können Probleme „knowledge rich“, „knowledge poor“, oder sogar „knowledge clean“ sein. Es scheint, dass besonders in Technik nahen Bereichen Probleme ein hohes Maß an Wissen aus den Naturwissenschaften, z.B. der Mathematik aufweisen und deshalb oft als „knowledge rich problems“ auftreten. In dem Bereich des Produktdesign mag das benötigte Wissen zuweilen unterschätzt werden und diese Probleme eher als „knowledge poor“ deklariert werden. In der Tat treten Probleme mit hohen Wissensvoraussetzungen in beiden Bereichen auf. Unterstützungen in den kreativen Leistungen durch digitale Techniken, werden dem Designer jedoch selten geboten. Dabei sind die am wenigsten formalisierbaren frühen Phasen der Problemfindung und Problemanalyse und das Auswählen und Konzipieren eines Lösungsprinzips bestimmend für nachhaltiges Innovationspotential (Sachse 2002). Song bestätigt diese Aussage in Bezug auf das Erstellen von Skizzen in dieser frühen Phase durch entsprechende empirische Untersuchungen (Shuang Song 2004).

3.2 Kreativität und Innovation

Kommen wir nach dieser Betrachtung zurück zu unserem Ausgangsproblem: Das Entwickeln einer innovativen Kaffeemaschine. Bisher haben wir den Entwurfsprozess als einen vom kreativen Akt getriebenen, eingebetteten Prozess verstanden. Dabei wurde versucht, dar zu stellen, dass Kreativität und Intuition keineswegs etwas ist, was einfach so vom Himmel fällt. Auch wenn es uns bisweilen durch ein archimedisches „Heureka!“ so vorkommt. Dennoch trifft der Ausspruch von Lois Pasteur „Der Zufall trifft nur einen vorbereiteten Geist“, das Phänomen der gefühlten, plötzlich einfallenden Erkenntnis viel eher.

Der Gesamtprozess, in allen seinen oben dargestellten Facetten, ist es, der es uns ermöglicht, ein innovatives Produkt herstellen zu können. Dabei wurde der Bezug zwischen den Begriffen Innovation und Kreativität noch nicht diskutiert. Daniel Couger (Couger 1995) bietet uns hier eine hilfreiche Trennung der beiden Konzepte. Er sieht Kreativität als ein weit breitbandigeres Konzept an, als es Innovation ist. Innovation ist ein beschränkterer Prozess um eine Idee in ein Produkt oder einen Service umzusetzen. Weiterhin bemerkt er, dass Kreativität nicht nur in den ersten Phasen, Teil der Produktentstehung ist, sondern von der „Geburt der Idee“ bis hin zum Reifen Produkt involviert ist.

Heaps Definition von Kreativität ist dagegen eher enger gefasst (Heap 1989):

„Creativity is the synthesis of new ideas and concepts by the radical restructuring and reassociation of existing ones’, whereas innovation is the implementation of the results of creativity.“

Die einfachste und prägnanteste Definition von Innovation bietet Amabile (Amabile 1998):

„Innovation is the successful implementation of creative ideas in any domain“

Eine weitere eher Breite Definition von Kreativität ist von Rhodes (Rhodes 1961). Er unterteilt diesen Begriff in vier verschiedene Grundelemente, die so genannten vier Ps der Kreativität:

1. Die kreative Person
2. Den kreativen Prozess
3. Das kreative Produkt
4. Das kreative Umfeld (engl. press, environment)

Seine Beschreibungen werden vor allem mit dem Begriff der „absichtlichen Kreativität“ in Verbindung gebracht welche nach Pahl und Beitz der sekundären Kreativität gleich kommt. Eine detaillierte Beschreibung dieses Modells würde an dieser Stelle zu weit führen. Dennoch ist es interessant, dass neben der individuellen Person der Prozess in Bezug auf Kreativität thematisiert wird.

Weshalb wurde *Kreativität* und *Innovation* in der Literatur in der Vergangenheit so sehr voneinander getrennt betrachtet? Während sich Betriebswissenschaften auf *Innovation* konzentrieren (Prozess und Produkt), haben sich Humanwissenschaftler auf die Erforschung der *Kreativität* (Personen und Umgebung) fokussiert. Die Human-Computer-Interaction Wissenschaft vereint Personen und Umgebung in Computer-Software (Produkte) und sollte sich somit aus beiden Forschungsfeldern bedienen. Aber das Problem ist auch nicht ganz neu. Der unverständlich herrschende Dualismus wurde auch von Kreativitätsforschern in der Betriebswirtschaft festgehalten. Hier heißt es z.B. von Cameron Ford (Ford 1995), dass Kreativitätsforscher und Innovationsforscher es verpasst haben, gegenseitig potentielle Synergien aus ihren Gebieten zu nutzen. Forscher aus dem einen Gebiet waren nur peripher mit dem vertraut, was im anderen Forschungsgebiet bereits erforscht wurde. Ähnliches gilt meiner Meinung auch für die Domäne des Designs und den Ingenieursdisziplinen. Beide Disziplinen haben die gleichen Probleme, nämlich Probleme lösen zu müssen. Dennoch haben beide Wissenschaften ihre eigenen Methoden gefunden, die sich aber auch wiederum gleichen, da ihnen die Basis der Kreativität, bzw. Generierung von neuen Ideen, gemein sind. Vielleicht sollten sich beide Forschungsgebiete mehr dem grundlegenden Erkenntnis- und Systemtheoretischen widmen, um auf gemeinsamen Fundamenten auf zu bauen.

Bevor wir die mögliche Unterstützung der Cave in Bezug auf den Problemlösungsprozess näher betrachten und im Speziellen, Potentiale des dreidimensionalen Skizzierens beleuchten, soll zunächst der Skizzierprozess als praktischer Weg zum Problemlösen kurz erörtert werden, denn wie Song in (Shuang Song 2004) dringlichst betont:

“In all engineering design domains – from software to hardware, from architecture design to mechanical design, graphical communication plays a key role throughout the design process, especially during the conceptual, formative phase.”

In Bezug auf das Skizzieren, ist anzunehmen, dass einige der oben beschriebenen Prozesse automatisch ablaufen, ohne dass man besonders auf sie achtet. Dennoch dürfte ebenso klar sein, dass nicht alle Schritte Anwendung finden können, da der Skizzierprozess etwas anderes ist, als das Problemlösen selbst. Das Skizzieren kann als eine Reduktion der obigen Schritte verstanden werden, und auf die beiden Schritte Analyse und Synthese zurückgeführt werden. Dieser Prozess ist auch als der

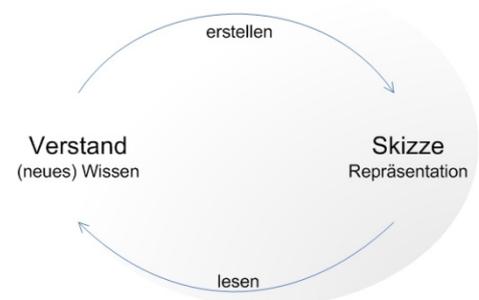


Abbildung 7 Der Reinterpretationskreislauf

Reinterpretationskreislauf bekannt (siehe Abbildung 7).

Das Skizzieren dient der Externalisierung von Gedanken und der Exploration eines Lösungsraumes mit dem Ziel, diesen bestmöglich erschöpfen zu können. Eine große Anzahl von Skizzen vermag z.B. dem Designer die Gewissheit zuteil werden lassen, dass er viele Alternativen in Betracht gezogen und das Problem ausreichend evaluiert hat. Die Exploration des Lösungsraumes in den frühen Produktphasen hat zudem den positiven Effekt, die Wahrscheinlichkeit über eine bessere Lösung zu späterem Zeitpunkt zu stolpern, zu verringern. Weiterhin verringert sich die Möglichkeit, dass ein Konkurrent ein weit überlegeneres Produkt mit dramatisch besseren Eigenschaften konzipiert, als das Eigene in der Entwicklung befindliche (K. Ullrich 1995).

Zudem ist das Skizzieren gegenüber dem Zeichnen abzugrenzen. Die Skizze ist eine vage Aussage, die noch keine Detaillierungen über Maße, Material, Funktion, etc. beinhaltet und nicht eindeutig ist. Die Mehrdeutigkeit der Skizze ist als Vorteil zu verstehen, da sie somit weitere Lösungswege eröffnet.

Die Skizze dient als ein Entwurf, ein Konzept oder einem ersten Überblick, die keineswegs detailliert oder ausgearbeitet sein muss. Sie dient dem Ersteller als schnelle Gedächtnisstütze, als einfache Ausdrucksmöglichkeit, und Mittel der Selbstkommunikation (Schön 1983), (Tversky 2003).

Designern dient die Skizze als:

- Denkwerkzeug
- Modellierungswerkzeug
- Kommunikationsmittel
- Dokumentationsmittel
- Festhalten von Interaktionsszenarien (Storyboard)
- Problemlösung
- Schnelle Variantenbildung
- Möglichkeit verschiedene Detailstufen des Problems zu erkennen
- Unterstützung des Kurzzeitgedächtnis

Abschließend kann gesagt werden, dass Kreativität der Intuitivität sehr nahe steht und innovative Ergebnisse durch diese geprägt sind. Innovation ist von rationaler Natur und als Umsetzung kreativen Denkens zu verstehen. Wie es Einstein sagte:

“The intuitive mind is a gift and the rational mind is a faithful servant. We have created a society that honors the servant and has forgotten the gift.”

3.3 Die 3D-Skizze und ihre weiteren Rollen im Entwicklungsprozess

Das Skizzieren ist also eine Möglichkeit ein Problem, etwa im Entwicklungsprozess, zu lösen. Vielleicht bietet skizzieren sogar die beste Möglichkeit überhaupt einem Problem zu begegnen. Was aber geschieht, wenn die Skizze ein Problem ausreichend skizziert hat und somit der mögliche Lösungsraum ausreichend erschöpft wurde?

An diesem Punkt, wenn mehrere Skizzen vorliegen, gilt es, die erarbeiteten Varianten zu beurteilen und auszuwählen. Die vermeintlich besten Design-Lösungen werden weiter verfolgt und in Prototypen umgesetzt und getestet. Dabei ist der Prototyp an sich wiederum als Denkwerkzeug zu verstehen, ebenso wie es die Skizze ist. Prototypen helfen Designern zu denken und dienen dazu, ein Design-Problem zu lösen (Beaudouin-Lafon & Mackay 2002). Um einen Prototyp erstellen zu können, werden Skizzen meist digitalisiert und in CAD-Systeme importiert. Hier entsteht dann ein mehr oder weniger exaktes, virtuelles Modell der erdachten und bereits skizzierten Lösung. Wie elaboriert ein Prototyp bereits ist, hängt meist von der Philosophie der Entwicklungsabteilung ab. Während Rapid-Prototyping auch Papier basierende Prototypen zulässt, ist es z.B. bei Apple im HCI-Design üblich, pixelperfekte Prototypen zu erstellen. Die Aufgabe eines Prototyps ist jedoch immer gleich. Abbildung 8 zeigt die Unterschiede eines Prototyps im Vergleich zur Skizze.

Skizze		Prototyp
heraufbeschwörend	→	didaktisch
vermuten	→	beschreiben
erforschen	→	definieren
fragen	→	antworten
vorschlagen	→	testen
provozieren	→	beschließen
tastend	→	spezifizierend
unverbindlich	→	Beschreibung

Abbildung 8 Unterschied Skizze – Prototyp (Buxton 2007)

Während die Skizze dem Denken durch seine vagen Linien hilft weitere neue Lösungen in divergenter Weise zu erkunden, legt ein Prototyp Entscheidungen vor (wirkt also konvergent). Interessante Möglichkeiten kann an dieser Stelle die immersive Umgebung der CAVE bieten. Dies kann am Anfang beschriebenen Beispiel der Entwicklung einer Kaffeemaschine am besten verdeutlicht werden.

Nimmt man an, dass z.B. technische Lösungen bezüglich Pumpwerk und Druckkammer einer Kaffeemaschine existieren, so gibt es meist auch CAD-Modelle dieser Elemente. Diese virtuellen Modelle können in den dreidimensionalen Raum der CAVE (siehe Kapitel 6.3) geholt und dort exploriert werden. Der Designer bekommt einen Eindruck davon, was die Funktion ist und kann durch direktes dreidimensionales Skizzieren in der CAVE Formen für die Funktionen entwickeln und evtl. bessere Affordances schaffen. Denn aus der Form schließt der Anwender auf die Funktion eines Produktes.

In weiteren Iterationsphasen entstehen so genaue CAD-Modell und neue Ideen als 3D-Skizzen, in direktem Kontext stehend mit bereits Entwickeltem. Designprobleme können im Gespräch mit Entwicklern, beispielsweise an einem virtuellen 1:1-Modell, in der CAVE besprochen werden. Mittels Skizze können ferner neu entstandene Gedanken ad-hoc direkt kommuniziert werden.



Abbildung 9 Informationsumsatz nach (Pahl & Beitz 1986)

Auf diese Weise bekommen die Kommunikationspartner die Möglichkeit, Informationen schneller und eindeutiger zu kommunizieren. Designer können bereits erarbeitete 3D-Skizzen, wie z.B. die Idee über eine neue Form eines Wassertanks in die CAVE-Umgebung laden und Konstrukteure beispielsweise den Anschluss der derzeitigen Wasserpumpe. Beide Mitarbeiter erhalten somit neuen Informationen, die auch interpretiert werden können, da die virtuelle Realität die Sache so darstellt wie sie ist (nur immateriell). Beide sprechen somit die gleiche Sprach und können Informationen innerhalb ihres Horizontes weiter verarbeiten und anschließend durch skizzenhafte Anmerkungen am virtuellen Objekt externalisieren.

Hier wird klar, dass *individuelle* Denkprozesse und *gemeinschaftliche* Kommunikationsprozesse durch Einsatz von virtuellen Umgebungen auf intensivste Weise verkoppelt werden. Zum einen dient die dreidimensionale Skizze dem Konstrukteur als Informationsgewinn, der für alle weiteren Schritte des Lösungsprozess (vgl. Abbildung 6) essentiell ist. Ist also somit in der Lage ein evtl. bestehendes Problem doch zu lösen, da er neue Informationen erlangt. Zum anderen kann er seine Ansicht durch Skizzieren wiederum dem Designer verdeutlichen oder durch hinzuholen (in die CAVE) früherer Konstruktionsvarianten neuen Informationsgewinn des Designers bewirken. Ein sehr enger digital gestützter Kommunikationsprozess entsteht, in einer visuellen und zeichnerischen, körperlich geprägten Sprache, die beide verstehen. Denn Gesten und das Deuten auf virtuelle Elemente können schneller verstanden werden als lange Erklärungen, die meist durch eigenes Domänenwissen und dazugehöriger Fachsprache geprägt ist.

4 Stand der Wissenschaft und Technik

Die virtuelle Produktentwicklung und ihre Prozesse stellen hohe Anforderungen an Mensch und Technik. Was genau die Virtuelle-Realität ist und wie Computertechnik immer effektiver die Handlungen des Menschen unterstützt, soll hier erläutert werden.

Weiterhin wird ein eigenes Modell entwickelt, welches die Vorzüge von greifbaren Eingabeschnittstellen darstellt und den Mensch enger mit dem Computer koppelt, ohne dass dieser eine dominierende oder gar behindernde Rolle einnehmen würde.

4.1 Virtuelle Realität

Im Gegensatz zu Büro-Desktop, jenseits von Multimedia und abseits von Hypermedia gibt es weit intensivere Methoden Virtualität zu erfahren. Ben Schneiderman vermag diesen Übergang vom anschauen zum „drin sein“ vortrefflich zu beschreiben, weshalb hier ein kurzer Abschnitt wiedergegeben sei (Schneiderman 2002).

„Seit zwei Jahrzehnten nutzen Architekten die Computer, um dreidimensionale Darstellungen von Gebäuden zu zeichnen. Die meisten ihrer Design-Systeme zeigen das Gebäude auf einem Standard- oder etwas größeren Display, aber wenn man einen Großprojektor nutzt, um ein großes Bild auf die Wand zu werfen, bekommen die potentiellen Kunden einen realistischeren Eindruck. Nun füge man Animation hinzu, um es den Klienten zu ermöglichen zu sehen, was passiert, wenn sie sich nach rechts bewegen oder in das Bild eintreten. Dann erlaube man dem Klienten, die Animation durch Gehen auf einem Laufband zu kontrollieren (schnelleres Gehen bringt einen rascher zum Gebäude), und erlaube ihnen, die Treppe hoch oder durch die Türen zu gehen. Schließlich ersetzt man den Großprojektor durch ein Head-Mounted-Display und überwacht die Kopfbewegungen, mit Polhemus-Trackern.

Jede Änderung nimmt die User ein kleines Stück weiter in der Bandbreite von „anschauen“ bis hin zu „drin sein“. Auf die Wände stoßen, (sachte) die Treppe hinunterfallen, andere Leute treffen oder auf einen Fahrstuhl warten zu müssen, könnte die nächsten Variationen sein.“

Die von Schneiderman beschriebene Bandbreite von „anschauen“ bis „drin sein“ wird in der VR gemeinhin als Immersion beschrieben. Demnach gibt es also visuelle Displays mit geringeren Möglichkeiten der Immersion und solche, wie sie Großdisplays bieten, mit höheren. Das normale große Computer Display, liegt in einer Sichtweite von 70cm und bietet einen Sichtbereich von 20° bis 30°. Ein Head-Mounted-Display dagegen kann einen Sichtbereich von 100° horizontal und 60° vertikal abdecken und somit leicht eine höhere Immersion in die Virtualität bieten, zumal zusätzlich die Sicht auf die

restliche Realität versperrt wird. Weiterhin entsteht durch Head-Tracking der Eindruck, man hätte einen Sichtbereich von 360°, indem das Bild je nach Lage des Kopfes entsprechend dargestellt wird.

Der Grad der Immersion kann noch zusätzlich gesteigert werden, wenn z.B. sog. haptische Displays hinzukommen. D.h., wenn die Virtualität nicht nur mit den Augen perzipiert, sondern auch mit den Händen ertastet werden kann. Die hierfür notwendige Technik ist leider um einiges aufwändiger. Dennoch hat auch sie bereits ihren Platz außerhalb der Wissenschaft gefunden. Computerspieler nutzen beispielsweise ein sog. Force-Feedback-Lenkrad um einen noch größeren „Spielspaß“ zu erhalten. Ein „realistischeres Fahrgefühl“ mag von ihnen als Grund für den erhöhten „Spielspaß“ genannt werden. Doch wissenschaftlich betrachtet ist es ein tieferer Immersionsgrad, dem der Spieler unterliegt. Erreicht wurde dies durch das zusätzliche, haptische Feedback, welches das visuell erfahrene Geschehen ergänzt.

Das Beispiel des Computerspiels ist vielleicht sogar die beste Art und Weise zu verdeutlichen, welche Möglichkeiten der Immersion bereits bestehen, wie ausgereift diese sind, und wie sich Computerspieler versuchen, eine noch immersivere, virtuelle Realität zu verschaffen. Nicht nur teure High-End-Grafikkarten mit einem oder mehreren großen Displays oder Force-Feedback-Lenkräder steigern den Immersionsgrad, auch 3D-Soundsysteme (wie z.B. Dolby-Surround, EAX, etc.) zählen bei den meisten Spiele-Computern schon zum Standard. Gerade auch im Bereich des Audio ist die Generierung von virtueller Realität so weit fortgeschritten, dass sie (bei entsprechender Qualität) nicht mehr von der Realität zu unterscheiden ist.

Die audi-visuelle und haptische VR kann durch weiteres Hinzufügen von Sinnessimulationen z.B. Temperatur (taktile Wahrnehmung), Geruch (olfaktorische Wahrnehmung), Geschmack (gustatorische Wahrnehmung) theoretisch bis zur Voll-Immersion fortgeführt werden. Ob diese Simulationen dazu genutzt werden, um einen realen Raum zu simulieren, sei zunächst dahingestellt. Sicherlich ist dies möglich, aber es könnte genauso gut eine totale Immersion in eine Phantasiewelt sein. Insofern führt auch der Begriff Virtuelle Realität ein wenig in die Irre. Er bedeutet nicht etwa das Abbilden der realen Welt in eine virtuelle Welt, sondern das Vorhandensein einer „erdachten Realität“.

Eine Definition von Burdea aus dem Buch „Virtual Reality Technology“ lautet:

„Virtual reality is a high-end user-computer interface that involves realtime simulation and interactions through multiple sensorial channels. These sensorial modalities are visual, auditory, tactile, smell, and taste.“ (Grigore Burdea 1994)

Ein weiterer Aspekt, den uns diese Definition liefert, ist der der Interaktion. Demnach ist ein 3D-Film nicht als virtuelle Realität zu betrachten, da ihm die interaktive Eigenschaft fehlt. Burdea schlägt

weiterhin neben Immersion und Interaktion eine dritte Dimension der VR vor - die Imagination – und komplettiert somit sein Modell der I³ (siehe Abbildung 10).

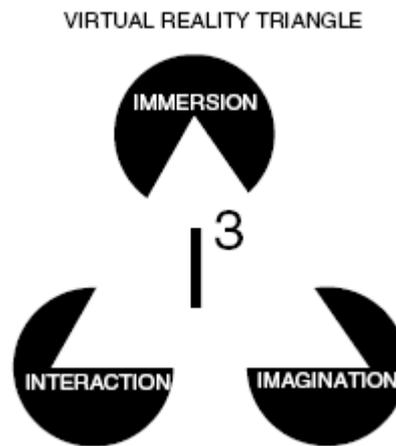


Abbildung 10 VR-Dreieck (Grigore Burdea 1994)

Mit dem Begriff der Imagination verdeutlicht Burdea den Sinn und Anwendungszweck von VR in seinem Modell, den er darin sieht, reale Probleme aus Konstruktion, Medizin, Militär, etc. mittels VR lösen zu können. Da etwaige VR-Applikationen von VR-Entwicklern hergestellt werden, hängt es von ihnen ab, in wie fern ein spezielles Problem mit der Applikation gelöst werden kann, was wiederum davon abhängt, wie weit der Anwender sich das Wahrgenommene überhaupt vorstellen kann. D.h., es handelt sich auch in der VR um ein klassisches HCI-Problem. Das System muss es dem Anwender ermöglichen, Dargestelltes richtig interpretieren zu können. Don Norman beschreibt diesen Umstand als „Gulf of Evaluation“ (Donald.A. Norman 1986) und meint damit (wie Burdea) den Grad, den ein Systemartefakt an Repräsentation veräußert und direkt, in Bezug auf Erwartung und Absicht, vom Anwender wahrgenommen und interpretiert werden kann. Der Golf (oder besser die Spalte) zwischen dem System und dem Anwender ist also klein, wenn ein Systemartefakt leicht zu verstehen und zu interpretieren ist und so vorliegt, wie der Anwender über ein System denkt. Der „Gulf of Evaluation“ bezeichnet also die psychologische Spalte, die überquert werden muss, um das perzipierte System imaginieren und interpretieren zu können. Hier spiegelt sich auch wieder, warum in der VR meist Abbildungen der Realität wiedergegeben werden. Denn die Imagination (Vorstellungskraft) des Anwenders wird so am einfachsten gefördert; und das System kann somit besser seinem Zweck nachkommen.

Alle drei Dimensionen des I³-Modells stellen Beziehungen zwischen Mensch und Computer dar. Während Immersion und Imagination hauptsächlich von der Darstellungsweise des Systems abhängt, und passiv von der Rezeption bzw. Perzeption und schließlich komplizierteren kognitive Prozessen des

Anwenders aufgenommen werden, beinhaltet die Interaktion das „aktiv werden“ des Anwenders. Wie sich Interaktion zwischen Anwender und Computer darstellt, soll im folgenden Kapitel behandelt werden.

Festzuhalten ist, dass die Abstufungen zwischen Realität und Virtualität fließend sind, also ein Kontinuum bilden, und dadurch Begriffe wie Augmented Reality (AR), Mixed Reality (MR) und Augmented Virtuality (AV) durchaus der VR zugeordnet werden können, wenn sie dem I³-Modell gerecht werden (vgl. (Billinghurst, Kato et al. 2001)).

4.2 Interaktion in immersiven Umgebungen

Eines der größten Probleme in immersiven Umgebungen ist die Interaktion mit virtuellen Objekten. Während bereits gute Ausgabemedien (Displays) für die visuelle und auditive Wahrnehmung des Menschen bestehen, gibt es bei der taktilen bzw. haptischen Wahrnehmung bisher nur beschränkte sowie technisch aufwändige Sonderlösungen. Doch gerade das taktile Feedback ist für basale Aufgaben wie auswählen, greifen und bewegen von virtuellen Objekten von enormer Relevanz. Zwar gibt es einige Vorrichtungen, die eine Simulation versuchen, wie z.B. der HapticMASTER von FCS, der omega 6 von Force Dimension, die Virtuose Geräte von Haption, die HapticWorkstation von Immersion, das Reachin Display von Reachin oder die Phanom Gerätereihe von SensAble. Aus praktischer, qualitativer, ergonomischer, ästhetischer oder preislicher Sichtweise sind diese Lösungen jedoch keinesfalls zufriedenstellend.

Die Schwierigkeit einer qualitativ hochwertigen virtuellen Realität für die menschlichen taktilen Fähigkeiten rührt daher, dass der Mensch viele verschiedene unterschiedliche Mechanorezeptoren besitzt (z.B. zum Sensieren von Widerstand, Elastizität, Oberflächentextur, Temperatur, etc.), die je nach Charakter unterschiedliche Wahrnehmungen hervorrufen. Zudem sind diese über den gesamten Körper verteilt, so dass von einem holistischen Ansatz zur Simulierung der taktilen menschlichen Sinne z.Z. Abstand genommen wird. Hilfslösungen neben den haptischen Simulatoren könnten sog. Tangible User Interfaces (TUI) sein, auf die im folgenden Kapitel näher eingegangen wird.

Wie gerade erwähnt, gibt es für die Sensorik des Menschen verschiedene Ausgabegeräte, sog. Displays. Während dieser Begriff im umgangssprachlichen Gebrauch für ein visuell ausgerichtetes Ausgabemedium (wie beispielsweise einem Computermonitor) steht, bedeutet er in der VR-Terminologie lediglich, dass es sich um die Darstellung einer für den Menschen rezipierbaren Information handelt. Ein Lautsprecher ist somit also auch als Display zu bezeichnen.

Aus systemtheoretischer Sicht dient das Display als Ausgabemedium für den Computer und als Eingabemedium für den Mensch. Für den entgegengesetzten Informationsfluss, ist das (Eingabe)Interface verantwortlich. Abbildung 11 zeigt Display und Interface als Transformatoren (Übersetzer) „digitaler“ Informationen aus dem Computer (VR) in „analoge“ Informationen (hinterlegt im RR) und umgekehrt.

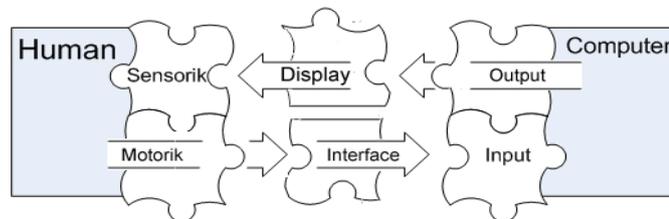


Abbildung 11 Transitionen von Informationen zwischen Mensch und Computer

Geht man noch weiter in der systemtheoretischen Betrachtungsweise, so regt uns die Abbildung zu weiteren interessanten Überlegungen an. Etwa kann ein zweiter Mensch in den Kontext des Systems treten. Dieser könnte mit seiner Motorik auf die Sensorik des anderen Menschen wirken oder er kann seine Sensorik ebenfalls an das Display binden. Auch könnte er mit seiner Motorik auf das Interface einwirken. Interfaces wie der Ubitable (Shen 2006) unterstützen z.B. die gleichzeitige Eingabe durch ein Multi-User Touch-Interface.

Genauso könnte auf der Seite des Computers der Output eines anderen Computers als Input für diesen gelten, etc. viele Variationen sind auch hier denkbar.

Auf welcher Granularitätsebene man das im Zuge dieser Arbeit entstandene Modell betrachtet, bleibt freigestellt. Im Bereich des Computers könnte man beispielsweise bis auf Bit-Ebene In- und Output Phänomene beobachten, beispielsweise in einem Quad-Core-Prozessor, einem RAID-System, einem Dual-Grafikkarten-System, einem Rechnercluster, etc. Die Granularität ist also beliebig. Gleiches gilt für die Betrachtungsweise des Menschen.

Wichtig ist hier zu erkennen, dass einige Systeme (Mensch, Computer, etc.) enger miteinander gekoppelt werden können als andere. Bisweilen benötigt man „Hilfsmittel“, die es überhaupt ermöglichen, Systeme koppeln zu können. Ohne eine Tastatur oder ein Display wäre etwa ein Computersystem für die meisten Menschen zwecklos und eine Kopplung unmöglich. So bedarf es also dieser Transformatoren (Übersetzer); Display und Interface. Die Kopplung spielt besonders in Hinsicht auf TUI eine Rolle und wird im Kapitel 4.3 näher behandelt. Es sei vorerst nur soviel gesagt, dass die bisherige fehlende Kopplung der Transformatoren, Display und Interface, zueinander eine Rolle spielt. Genauso, wie die Intensität der Kopplung eine wichtige Rolle spielt.

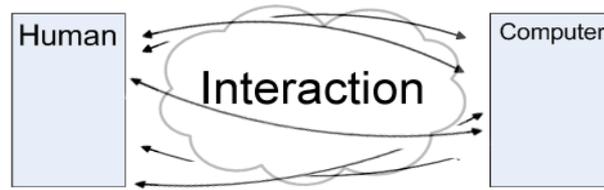


Abbildung 12 Abstrahiertes HCI Modell

In Bezug auf die Interaktionen in der Realität und der VR gibt es also, wie anfangs erwähnt, spezielle Probleme beim Greifen und Handeln mit virtuellen Objekten. Weitere Unterschiede zwischen RR und VR sind z.B. die Gewichtslosigkeit von Objekten, die Möglichkeit durch sie hindurch zu schauen, sie zu skalieren oder sie zu manipulieren, obwohl sie außer Armreichweite liegen. Jeder der beiden Räume (RR und VR) hat seine speziellen Vor- und Nachteile. Das Resultat sind *Hybride Interfaces* wie der Virtual Tricorder (Wloka 1995), der Pick-and-Drop-Ansatz von Rekimoto (Rekimoto 1997) und einige Tangible User Interfaces von Ullmer und Ishii (B. Ullmer 1997). Einige kombinieren zudem verschiedene Interaktionstechniken der Desktop 2D-Umgebung und 3D-VR, wie z.B. das „Eye of Ra“ (Alexander Bornik 2006) der TU Graz. Untersuchungen, wie die von Krause (Krause, Israel et al. 2007) zeigen, dass hybride Interaktionstechniken durchaus zu genaueren Manipulationen von virtuellen Objekten verhelfen können als rein virtuelle Interaktionstechniken. Es zeigt sich also, dass es durchaus Sinn macht sich über die Interaktionen zwischen Anwender und Computer und die verschiedenen Räume wie RR und VR Gedanken zu machen.

Der Rest dieses Kapitels beschäftigt sich mit einer Auswahl an innovativen dreidimensionalen Displays und Eingabeinterfaces, um ansatzweise die Vielfalt der Anstrengungen eine besonders guten Kopplung zwischen VR und RR dar zu stellen. Hierbei kann jedoch keine vollständige Liste wiedergegeben werden, da dies bei weitem die Arbeit sprengen würde.

4.2.1 3D Interfaces



Abbildung 13 The Bat von Ware und Jessome

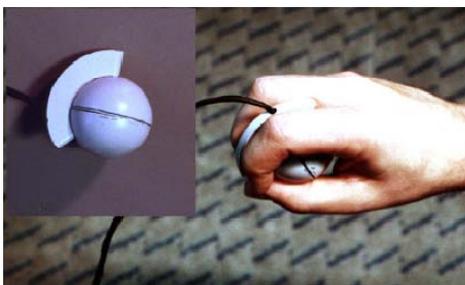


Abbildung 14 6DOF Maus von Dan Vanolia



Abbildung 15 The GyroPoint von Gyration Inc.



Abbildung 16 Nintendo Wii Controller

The Bat von Ware und Jessome (Colin Ware 1988) ist ein Eingabeinterface, das Bewegungen in sechs Freiheitsgraden (6DOF) ermöglicht. Es war eines der ersten User-Interfaces, welches dies durch einen sog. Polhemus-Sensor ermöglichte. Zusätzlich hatte *The Bat* einen „Maus-Knopf“. Einen ähnlichen Prototyp entwickelte Dan Vanolia für Apple Computer. Auch er nutzte einen Polhemus-Sensor. Die Affordance der Maus gibt hier allerdings viel besser zu verstehen, wie das Interface zu nutzen ist und dass es sich hier um eine Maus für dreidimensionale Eingaben handelt, da sie keine Fläche hat, auf der sie hin und her geschoben werden könnte. Der GyroPoint von Gyration Inc. sieht zwar der 6DOF-Maus von Apple ziemlich ähnlich, allerdings beinhaltet sie im Gegensatz zu den beiden oberen Interfaces ein Gyroskop anstatt eines Polhemus-Sensor. Der Unterschied besteht darin, dass beim GyroPoint nur Bewegungsdaten, also relative Bewegungen gemessen werden können, während der Polhemus-Sensor eindeutige absolute Positionen misst. Der Vorteil dieser Technik äußert sich vor allem im Preis, da Gyroskop-Sensoren wesentlich günstiger sind. Zudem kann es, im Gegensatz zu den beiden anderen, kabellos betrieben werden.

Der Wii Controller kann wohl als das Erste, weltweit kommerziell erfolgreiche 3D-Interface genannt werden. Er kombiniert mehrere Sensoren in einem Gerät, wie Beschleunigungssensor, Infrarotkamera, Vibrationseinheit und benutzt Bluetooth-Technologie zur Datenübertragung.

4.2.2 3D Displays



Abbildung 17 Haptisches Display



Abbildung 18 3D Audio Display



Abbildung 19 Propriozeptorisches Display



Abbildung 20 3D Visuelles Display

Die Geräte der Phantomreihe von SensAble Technologies sind sowohl Eingabeinterfaces mit denen sich virtuelle Objekte manipulieren lassen, als auch haptische Displays dank dessen sich virtuelle Objekte auch ertasten lassen. Der Interaktionsraum der Geräte variiert zwischen 38,1 W x 26,7 x H x 19,1 D cm und 90 W x 90 H x 30 D cm. Die möglichen Rotationen liegen bei 297° (yaw), 260° (pitch) und 335° (roll). Als dreidimensionale Displays für die auditive Wahrnehmung haben sog. Sourroundsound-Systeme bereits weite Verbreitung gefunden. Vor allem im Computerspiele- und Home-Cinema-Bereich wird ein sehr hoher Aufwand betrieben, um einer realen Geräuschkulisse gleich zu kommen. 3D-Soundkarten berechnen beispielsweise in Echtzeit die räumliche Verteilung der Geräuschquellen. Auf diese Weise kann ein Spieler z.B. erfahren, aus welcher Richtung ein herannahendes Auto kommt oder ob er sich auf Teppich oder Marmor mit seiner Spielfigur bewegt. Eine Untergruppe der haptischen Simulation ist die Vortäuschung der Propriozeption. Ihr Ziel ist die Simulation von Bewegungskräften, wie sie etwa beim Gehen auftreten, und von unserer Körperposition im Raum, von der Position unserer Arme und Hände. Das „Active Surface“ der Firma Virtual Space Devices Inc. bietet hierbei die Möglichkeit auf einem Laufband in alle Richtungen zu laufen, ohne seine relative Position zum Raum zu ändern. Vor allem in CAVE-Situationen kann diese Technik hilfreich eingesetzt werden, um sich in virtuellen Landschaften bewegen zu können. Eine „künstliche“ Fernsteuerung der Bewegung, etwa durch einen sog. Flightstick, würde somit entfallen.

Neben diesen Beispielen von Interfaces und Displays, gibt es noch eine Vielzahl weiterer, deren Auflistung zusätzlich bestätigen würde, dass die Mensch-Computer-Interaktion noch keineswegs als befriedigend empfunden wird.

4.3 Tangible User Interfaces und ihre Vorteile gegenüber herkömmlichen Benutzerschnittstellen

Der Begriff Tangible User Interfaces (TUI) hat besonders in den letzten beiden Jahren an Popularität gewonnen. Selten jedoch wissen diejenigen, die davon fasziniert sind, was es eigentlich genau bedeutet, und warum viele Forscher dieses Konzept für so relevant halten. Wesentliche Konzepte sollen hier geklärt werden.

Zunächst jedoch sei erwähnt, dass sich auch in der Wissenschaft erst seit den letzten beiden Jahren ein gemeinsames Verständnis über TUI ausgebildet hat. Ein wichtiger Schritt hierbei war die erstmals Februar 2007 stattfindende eigene „International Conference on Tangible and Embedded Interaction“ in Baton Rouge, Louisiana, USA. Mittlerweile glaubt man den Begriff Tangible User Interface verstanden zu haben und untersucht nun die weit interessantere Variante von TUI: „Tangible User Interaction“. Als dessen Vorreiterin Eva Hornecker (Hornecker 2002) genannt werden kann. Erkannt wurde der Unterschied zwischen *TU-Interface* und *TU-Interaction* zwar bereits schon von Paul Dourish im Jahr 2000, aber damals wurden die beiden Trends noch als „Tangible Computing“ und „Social Computing“ bezeichnet. Als Ausgangspunkt für beide Paradigmen ist die Vision des Ubiquitous Computing anzusehen, welche unter anderem von Mark Weiser (Weiser 1991) oder Donald A. Norman in „The invisible Computer“ visioniert wurden. Normans Designlösung für „digitale Werkzeuge“,

„Design the tool to fit so well that the tool becomes a part of the task.“ (Norman 1999)

gilt sicherlich auch als Maxime für TU-Interfaces. Der Computer tritt in den Hintergrund und soll genau wie beim Ubiquitous Computing Paradigma „unsichtbar“ werden.

Als eine der ersten Umsetzungen von echten TUI-Anwendungen, neben metaDesk, transBoard, ambientRoom (Hiroshi Ishii 1997), Wellners „Digital Desk“, Schäfers „Real Reality“, Natalie Jeremijenkos „Live Wire“, Durrell Bishops „Marble Answering Machine“, David Merrill Siftables (David Merrill 2007), sei hier URP (Urban Planning Workbench) genannt (J. Underkoffler 1999). URP nutzt physikalische, maßstabsgetreue Architektur-Modelle zur Simulation von Computer generierten Schattenwürfen, Lichtreflektionen, Windflüssen, etc. Zusätzlich bietet URP eine Vielzahl von interaktiven Tools mit denen man die Simulation beeinflussen kann. Beispielsweise kann man mittels einer physikalischen Uhr die Position der Sonne verändern, mit einem „Zauberstab“ die Oberflächen der Gebäude von Stein- zu

Glasflächen wechseln, mit einem „Wind-Tool“ die Richtung und Geschwindigkeit der projizierten Strömungspfeile ändern, etc. Somit geht die Interaktionsweise mit URP über übliche graphische User-Interfaces (GUI) mit Maus-Tastatur- Bildschirm Konfiguration weit hinaus.

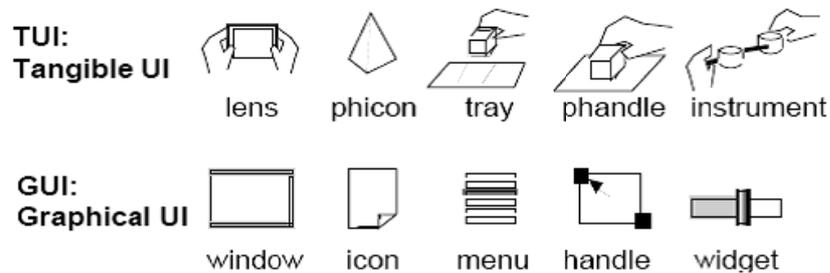


Abbildung 21 Semantischer Vergleich zwischen TUI- und GUI-Artefakten (Uderkoffler & Ishii 1998)

Der Unterschied zu herkömmlichen Simulationen ist, dass die physikalischen Gebäudemodelle für greifbare (also tangible), physikalische Repräsentationen digitaler Versionen des Gebäudes stehen. Um den Ort oder die Orientierung von Gebäuden zu ändern, kann der Anwender einfach das physikalische Modell in die Hand nehmen und es wo anders hinstellen, im Gegensatz zum Selektieren und Ziehen (welches eher einem fernsteuern gleicht) einer graphischen Repräsentation mittels Maus. Hier werden mehrere Unterschiede zu herkömmlichen Interaktionen mit Computern besonders deutlich.

1. Es gibt einen fundamentalen Unterschied zwischen GUI und TUI. Während GUIs „digitale“ Informationen als *graphische* Repräsentationen anzeigen, zeigen TUI die „digitalen“ Informationen mittels *physikalischen* Repräsentationen an. Man mag zwar einwenden, dass der Computer nicht in der Lage ist, physikalische Repräsentationen aktiv als Informationsträger zu nutzen, da er etwa das Gebäudemodell nicht verändern kann; aber auch das stimmt nur bedingt, wie die „Actuated Workbench“ von Pangaro (G. Pangaro 2002) zeigt.
2. Die Interaktion zwischen Mensch und Computer mittels TUI gleicht weniger einem fernsteuern, wie es sonst mittels Maus und Tastatur der Fall ist, sondern einem direkten interagieren.
3. Im Gegensatz zu relativ jungen Desktopmetaphern wie GUIs sie bieten (Scrollen, Windows, Drag & Drop, etc.) , benötigen TUIs keine Metapher zur Unterstützung kognitiver

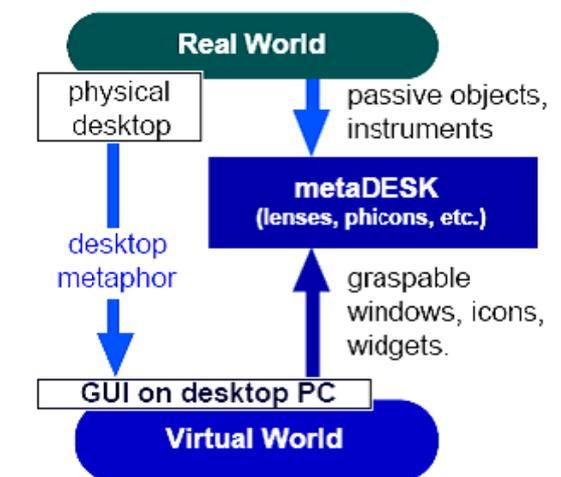


Abbildung 22 GUI versus TUI (Hiroshi Ishii 1997)

Prozesse, da wir es gewohnt sind, uns in der physikalischen Welt zu bewegen und von der Natur darauf angepasst wurden. Somit können TUIs auch Jahrhunderte alte Affordances von Objekten zu Ihrem Vorteil nutzen. Mögliche Nutzungsweisen sind also inhärent und bedürfen keiner zusätzlichen Interpretation wie ein Artefakt zu verwenden ist (vgl. Abbildung 22). Somit wäre auch das Wahrnehmungs- und Interpretations-Problem welches von Norman als „Gulf of Evaluation“ (Norman & Draper 1986) formuliert wurde und bezüglich GUI-Artefakten besteht, hinfällig.

4. Die Rechenleistung wird in alltägliche Objekte integriert und der eigentliche Rechner verschwindet (digital embodiment), wie es die Vision des Ubiquitous Computing fordert. Eine nahtlosere Unterstützung der Interaktion zwischen Mensch und Computer wäre das Resultat.

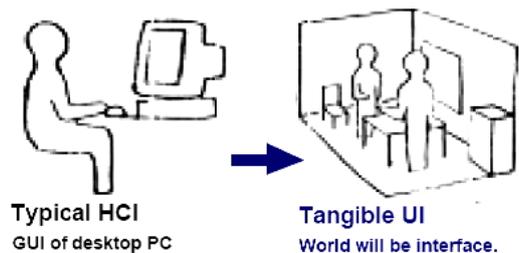


Abbildung 23 Vom User Interface zum Tangible User Interface (Hiroshi Ishii 1997)

5. TUIs sind in ihrer Verwendung spezieller als GUIs, wie das URP-Beispiel verdeutlicht.
6. Display und Interface verschmelzen zunehmend und sind keine getrennten Transformatoren zwischen Mensch und Computer, wie es Kapitel 4.2, Abbildung 11 dargestellt wurde.

Für den letzt genannten Unterschied wurde von Ullmer ein Framework entwickelt, das diese Tatsache näher erläutert. In (Ullmer & Ishii 2000) werden die Schlüsseleigenschaften von TU-Interfaces dargelegt. Im Wesentlichen geht es um die Erkenntnis, dass Eingabegeräte (Control) und Displays (View) in physikalische Repräsentationen (REP-P) und digitale Repräsentationen (REP-D) zerfallen. TUIs rücken somit das Interface näher in den physikalischen Raum (symbolisiert durch das Wasser) und dienen als „Griffe (handles)“ (G. W. Fitzmaurice 1995) für digitale Informationen (Model). Abbildung 24 versucht diesen Sachverhalt darzustellen.

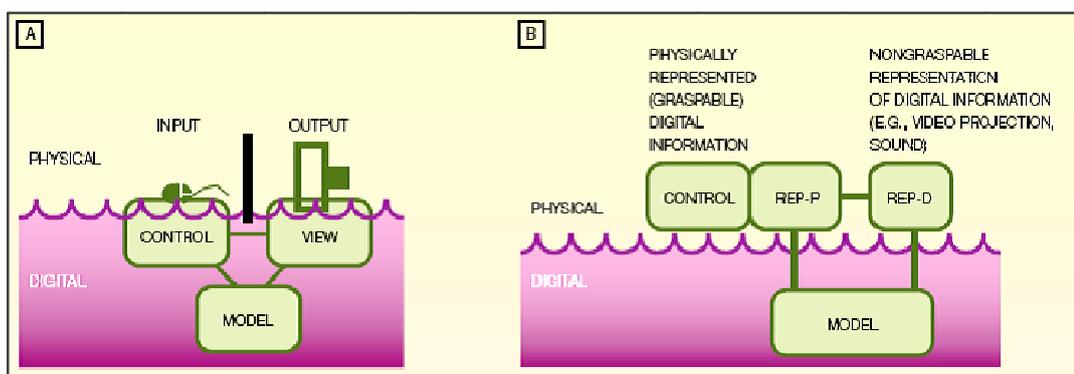


Abbildung 24 TUI Framework (Brygg Ullmer 2000)

Während Fall A eine typische GUI-Situation zeigt, in der „seeing and pointing“ die Haupthandlung beherrscht, wird in B die physikalische Eigenschaft stärker hervorgehoben und die Haupthandlung verschiebt sich zu „greifen und handeln“. Somit gelingt es, eine engere Kopplung innerhalb der Interaktion zwischen Mensch und Computer zu schaffen und die kognitive Belastung beim Anwender kann gesenkt werden. Dass GUIs eine losere Kopplung bedeuten, kann z.B. mit der Existenz von Fitts-Law (Fitt 1954) begründet werden. Dieses Gesetz besagt, dass die Ausführungszeit, z.B. das Drücken eines Knopfes mit der Maus, in Abhängigkeit dessen Größe der graphischen Repräsentation mehr oder weniger schnell durchgeführt werden kann. Da durch TUIs das „Fernsteuern“ mittels Maus entfällt, gibt es ein solches Problem, welches eine enge Kopplung zwischen Mensch und Computer verhindern würde, nicht.

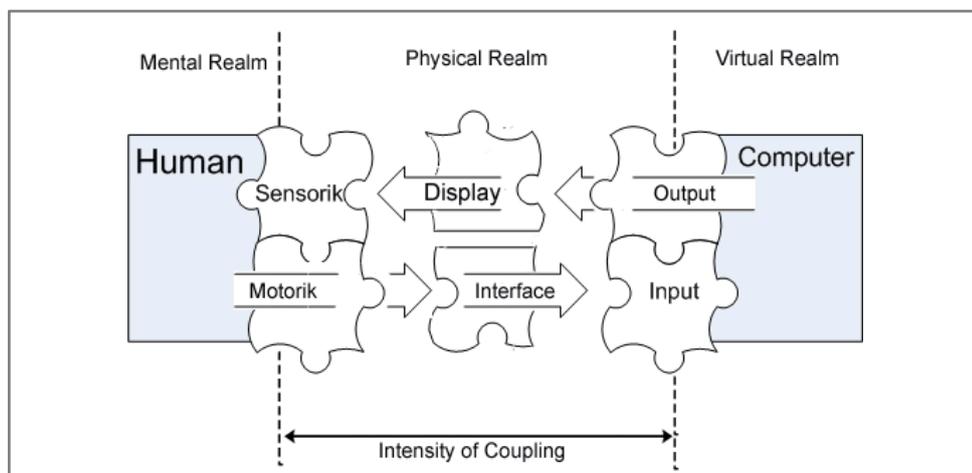


Abbildung 25 Kopplung der Räume

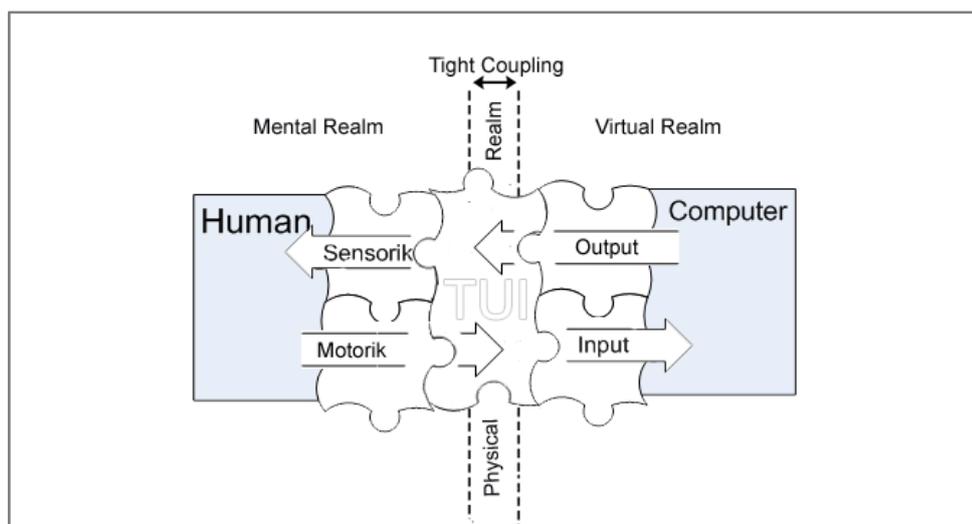


Abbildung 26 Enge Kopplung der Räume durch TUI

Abbildung 25 und Abbildung 26 illustrieren den Unterschied zwischen bisherigen Interaktionsparadigmen und denen, wie sie durch TUIs ermöglicht werden. Dadurch, dass Display und Interface verschmelzen, ergibt sich eine engere Kopplung zwischen Mensch und Computer. Psychologisch lässt sich diese engere Kopplung damit begründen, dass Handlungsraum und Wahrnehmungsraum zusammenfallen. Diese Tatsache vermag besonders in Bezug auf das dreidimensionale Skizzieren von Bedeutung zu sein. Für weitere Ausführungen soll an dieser Stelle auf die Dissertationsschrift von Johann Habakuk Israel (Israel 2008) verwiesen werden, da er eingehende Untersuchungen zu diesem Thema unternommen hat.

Festzuhalten bleibt, dass TU-Interfaces eine direktere Interaktion zwischen Mensch und Computer ermöglichen und sehr fokussiert auf die Handlung selbst sind, während GUIs eher auf die Technologie fokussieren. Weiterhin ermöglichen sie beidhändige Computereingaben und eine reduzierte kognitive Belastung im Vergleich zu GUIs sowie eine engere Kopplung zwischen der Hand des Anwenders und den Berechnungen seitens des Computers (Fitzmaurice 1997). Der Grund für die Ignoranz gegenüber einer besseren Möglichkeit mit Computern zu interagieren sieht Buxton im Mangel an geeigneten Interfaces und an den zugunsten der Displays verschobenen wissenschaftlichen Anstrengungen (Buxton 1997). Die Interface-Beispiele aus dem Kapitel 4.2.1 und 4.2.2 zeigen auch, dass Displays weitaus unterschiedlicher in ihrer Art ausfallen als Eingabeinterfaces, die meist an eine Fernbedienung erinnern. Der derzeit herrschende Hype der TU-Interfaces könnte dieses in Zukunft ändern. Auch Fortschritte in der Forschung bezüglich optischer Eingabeinterfaces, die z.B. Gesten interpretieren, tragen dazu bei natürlichere Interaktionen mit dem Computer zu ermöglichen. Optische Systeme spielen im Zusammenhang mit TU-Interfaces also die Rolle einer „enabling technology“. Sie können z.B. wie beim „reactable“ von Jordâ (Jordâ 2003) helfen, neue TUIs zu kreieren.

5 Methodik zum praktischen Teil der Arbeit

Die Einarbeitung in das Thema der Computer Vision (CV), bietet dem Softwareentwickler meist ein unüberschaubar großes Gebiet der Wissenschaft. Es besteht eine Vielzahl von Ansatzmöglichkeiten, aus denen nur schwer der richtige Weg zur Lösung eines Problems auf Anhieb erkannt und eingeschlagen werden kann. Gründe hierfür sind:

- Die Computer Vision ist ein sehr altes Themengebiet der Computerwissenschaft und bietet somit eine Fülle an Literatur.
- Viele Konzepte bzw. Algorithmen (welche in der CV als Werkzeug fungieren) sind der Mathematik (altgr. Adj. „[die Kunst des] Lernen, zum Lernen gehörig“, altgr. Verb „ich lerne“) entliehen, die wiederum noch mehr Lösungsmöglichkeiten bietet, da sie eine der ältesten Wissenschaften überhaupt ist.
- Viele Algorithmen werden als Teillösungen in der Literatur präsentiert und nicht im Gesamtkontext betrachtet. Das Auffinden des passenden Algorithmus bzw. die Kombination mehrerer ist daher recht schwierig.
- Es mangelt dem Entwicklungsprozess von Tracking Systemen an Formalismen und Vorgehensmodellen, die besonders in zunehmend komplexen Systemen notwendig sind.

Besonders die große Anzahl an verfügbaren Lösungen macht es schwer ein geeignetes System zu entwickeln. Auch Mathematiker kennen oft nur Teilbereiche bzw. –probleme der Computer Vision und überblicken mögliche Lösungen für ein Gesamtsystem nur selten. Geometrie, lineare Algebra, Statistik, Bayes-Netzwerke, Optimierung, Funktionenanalyse, Signalverarbeitung, Gabor-Wavelets, Hough-Transformation, etc. sind nur einige Themenbereiche mit denen man es zu tun bekommen kann. Einige dieser Werkzeuge sind leichter verstehbar, andere dagegen nicht. Die Aufgabe des Informatikers bzw. Medieninformatikers besteht weniger in der Erstellung oder Optimierung dieser Werkzeuge. Er ist der Anwender, kann sich also das Werkzeug aus dokumentierten Algorithmen selber herstellen, sofern dies nötig ist, oder es direkt verwenden, um somit seine Aufgabe zu erledigen.

Beginnt man mit der Arbeit, stellt man häufig fest, dass man nicht genau weiß, welches Werkzeug das richtige ist. Es kommt sogar noch schlimmer. Der Entwickler muss Algorithmen aus vorliegenden Softwarebibliotheken kombinieren, ohne dass dafür geeignete Formalismen zur Verfügung stehen. Wie unter anderem Pustkas Arbeiten (Pustka, Huber et al. 2006) zu den sog. Sensor-Netzwerken zeigen, wird daran zwar gearbeitet, aber erweisen sich bis heute als nicht ausreichend.

Somit hatte die Entwicklung der im Rahmen dieser Arbeit erstellten Prototypen eher explorativen Charakter. Verschiedene Strategien der Lageerkennung (Pose Estimation) wurden implementiert und Erkenntnisse über die Vor- und Nachteile bestimmter Lösungsansätze und Kombinationen von Algorithmen wurden erarbeitet. Hilfreich war hier besonders, dass die verwendete C++ Bibliothek OpenCV dieses Vorgehen aktiv durch Strukturen und Funktionen der Computer Vision unterstützte.

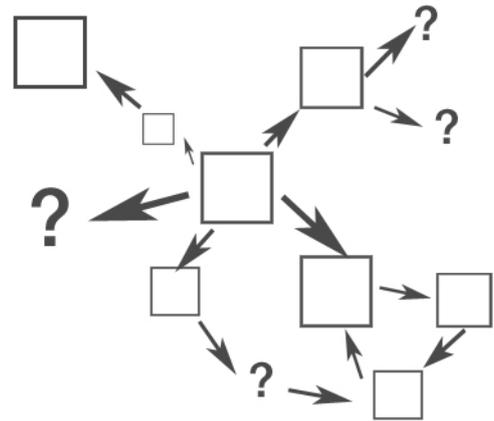


Abbildung 27 Exploratives Vorgehen

Aus didaktischer Sicht kommt diese Situation der anspruchsvollen, herausfordernden Lernsituation gleich, in der der Lernende versucht die Situation für sich durch exploratives Verhalten zu meistern. Um aber nicht in alle Richtungen gleichzeitig laufen zu müssen, gab es zwei hilfreiche, komplexitätsreduzierende Hilfsmittel:

- Es gab ein Systemziel.
- Es gibt bereits funktionierende Systeme aus denen man lernen konnte, obwohl diese nicht im Detail beschrieben waren.

Als Ergebnis dieses explorativen Vorgehens in dieser Arbeit, ist eine alternative Möglichkeit des optischen Trackings zu Tage gefördert worden, die derzeit kommerziell erhältliche Systeme nicht berücksichtigen.

Ergänzende Experimente haben zudem, zu einem phänomenologischen Verständnis über Interaktion im immersiven Raum und inhärenten Beschränkungen durch die Physikalität des Körpers und dem menschlichen Handeln geführt.

6 Dreidimensionales Skizzieren in der Cave mit SketchApp

Nachdem nun die theoretischen Grundlagen in den vorangegangenen Kapiteln erläutert und die Hauptabsichten indirekt vermittelt wurden, widmet sich der zweite Teil der Arbeit der konkreten Unterstützung des Produktentwicklungsprozesses durch eine CAVE-Anwendung. Hierzu wurde von Johann Habakuk Israel eine Anwendung Namens SketchApp erstellt, die er im Rahmen seiner Dissertation als Experimentierfeld für seine Hypothesen im Bereich TUI verwendet hat. Die Cave-Anwendung wurde schließlich von Diplomanden und Masterstudenten aus den verschiedensten Bereichen weiterentwickelt und verbessert. Dabei realisierte der Design-Student Christian Zöllner im Rahmen seiner Diplomarbeit, Konzepte zu 3D-Skizzier- und Interaktionstechniken sowie die hybriden Skizzierwerkzeuge zum skizzieren im 3D-Raum. Gunnar Bittersmann übernahm einen Teil der Programmierung der hybriden Skizzierwerkzeuge und Rasim Korkot implementierte die Drucksensoren der Skizzierwerkzeuge sowie die Programmierung einiger Glättungsfilter.

Die folgenden Kapitel in diesem Abschnitt beschreiben einzelnen Teile des gesamten 3D-Skizziersystems, um weiterführend die Ergänzung um ein mögliches optisches Tracking-System zu verstehen und Theorie und Technik zu verbinden.

6.1 Allgemeine Anforderungen zum 3D-Skizzieren

Aus den theoretischen Grundlagen im ersten Teil dieser Arbeit, ergeben sich folgende nicht-funktionale Anforderungen für das dreidimensionale Skizziersystem SketchApp.

Das System soll

a) die Art und Weise des **Erfassens und Analysierens** des zu lösenden Designproblems unterstützen.

D.h.:

- Umfassende und präzise Möglichkeit zur Problemanalyse mit
- der Konzentration auf die bedeutsamen Elemente für die Funktionsfindung (also nicht nur auf die auffälligsten Elemente)bieten.
- das vollständige Bestimmen der Hauptfunktionen ermöglichen.
- ein abstrahierendes und gewichtendes Zusammenfassen der Zielmerkmale unterstützen
- zeichnerisches und schriftliches Fixieren ermöglichen.
- ein auf das Gesamtziel bezogenes aber flexibles Vorgehen mit den herausgearbeiteten Merkmalen unterstützen.

- b) die Art und Weise der **Suche nach den Prinziplösungen** unterstützen. Hierzu zählt:
- Ein generierendes, d.h. neue Lösungen erzeugendes Vorgehen, im Gegensatz zum lediglichen Korrigieren schon gegebener Lösungen.
 - Das divergierende, schöpferische Suchen vielfältiger Lösungsprinzipien vor dem weiteren Entwerfen eines ausgewählten Prinzips.
 - Eine konvergierende Eingrenzung der Lösungsvielfalt des sog. Suchraumes, auf möglichst wenige Varianten
 - Ein methodisches, aber dennoch flexibles Vorgehen beim Entwerfen mit einer begrenzten Anzahl von Vorgriffen auf spätere Lösungsschritte und korrigierende Rücksprünge auf schon bearbeitete Abschnitte
 - Das Fördern unterbewusster, kognitiver Prozesse durch Entlastung des „working memories“
- c) das rückkoppelnde **Beurteilen der Lösungsschritte ermöglichen**. Dabei werden
- die verfolgten Lösungsvarianten häufig auf ihre Güte beurteilt und
 - die Angemessenheit des abstrakten Prinzips wie auch der konkreten Detaillösungen wiederholt bewertet.

6.2 Funktionale Anforderungen an SketchApp

Funktionale Anforderungen an das System sind:

- Die Bewegung der Skizzierwerkzeuge muss in 6DOF für die Anwendung erfassbar sein.
- Das System muss erkennen, welches Skizzierwerkzeug vom Anwender gerade in der Hand gehalten wird. Es muss also für das System identifizierbar sein.
- Zum Tracken der Skizzierbewegung sollen Low-Cost Kameras eingesetzt werden die max. 30 Frames pro Sekunde liefern.
- Das System soll dem Benutzer einen möglichst großen Interaktionsraum bieten, in dem er seine 3D-Skizzen anfertigen kann. Idealerweise, sollte der Interaktionsraum die Größe der Cave haben.
- Die Latenz des Systems sollte 100ms nicht überschreiten, dieser Wert ist empirisch als der Wert festgestellt worden, ab dem ein User eine Verzögerung als störend empfindet (Schneiderman 2002) eventuelle Flowprozesse werden dadurch verhindert.
- Die Genauigkeit des Systems sollte eine Abweichung von 2cm in der Translation und 2° in der Rotation nicht überschreiten.
- Das Tracking der Zeichenwerkzeuge sollte Robustheit aufweisen. D.h., auch trotz Abweichungen (z.B. durch Ausreißern oder Verdeckung) ein akzeptables Ergebnis liefern.

- Das System soll als Low-Cost-System realisiert werden. Low-Cost bedeutet, dass keine Spezialkameras zum Einsatz kommen, die den Preis des Trackingsystems zum Großteil bestimmen würden.
- Die Zeit, die für das Kalibrieren des Systems aufgebracht werden muss, soll minimal gehalten werden. D.h., eine Einmalige Kalibrierung ist wünschenswert im Gegensatz zu wiederholt nötigen Kalibrierungen vor dem eigentlichen Einsatz.
- Das Trackingsystem soll in das Bestehende Cave-System integrierbar realisiert werden. D.h., Infrarotsensoren des aktiven Stereoskopischen Verfahren der Cave dürfen nicht gestört werden, das Datenaufkommen muss über das vorhandene Netzwerk handhabbar bleiben, Positionsdaten sind der Applikation SketchApp in geeignetem Format z.B. über einen Socket zu übergeben.
- Das Trackingsystem soll auf einem Linux System laufen und die Firewire-Schnittstelle nutzen

6.3 Interaktionsraum CAVE

Wie bereits angemerkt wurde die Skizzieranwendung speziell für die CAVE der Fraunhofer Gesellschaft, IPK, Berlin entwickelt. Die im Institut untergebrachte CAVE hat fünf Projektionswände auf die jeweils ein



Abbildung 28 CAVE des Fraunhofer IPK

Projektor über einen Umlenkspiegel gerichtet ist. Es handelt sich bei dem Gesamtsystem um ein aktives System. D.h., die Bilder für das linke und rechte Auge werden von einem Projektor mit einer Bildfrequenz von 100Hz projiziert. Um die jeweils rechten und linken Bilder auch den rechten und linken Augen der Betrachter zuzuordnen, muss eine sog. Shutterbrille getragen werden. Damit diese synchron mit den Projektorbildern arbeiten, sind auf jede Projektionsfläche drei Infrarot Emitter gerichtet, welche Infrarotimpulse (pulse period left = 122 μ sec und pulse period right = 61 μ sec, laut inpo Support) im Wellenlängenbereich von 925nm senden, aus denen die Shutterbrille erkennt, wann sie welches LCD-Brillenglas verdunkeln soll.

Die projizierten Bilder stammen von einem insgesamt 12 Rechner umfassenden Cluster. Je fünf Rechner für die Projektion der Bilder für das linke und rechte Auge, ein Rechner für die Berechnung des Szenegraphen und ein Rechner für die jeweiligen Applikationen (z.B. sketchApp). Verbunden sind die Rechner durch ein Fast-Ethernet Netzwerk zur Kommunikation der zu rendernden Szenen.

Damit zum richtigen Zeitpunkt auf allen fünf Projektionswänden das jeweils für ein bestimmtes Auge richtiges Bild gezeigt wird, sind die Grafikkarten der zehn Rendering-Rechner synchronisiert.

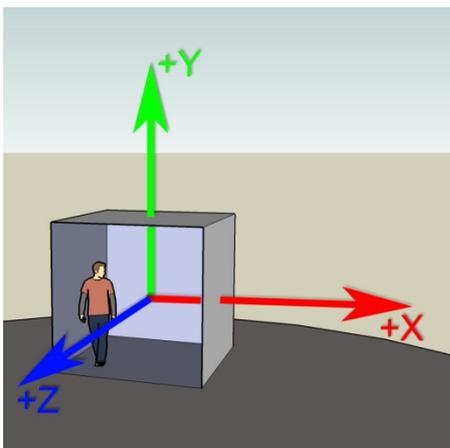


Abbildung 29 Koordinatensystem der CAVE

Die zu rendernden Objekte einer CAVE-Echtzeitanwendung werden in einem Szenegraphen durch OpenGL verwaltet. Dabei nutzt OpenGL die OpenGL Library, wodurch es auch seine Plattformunabhängigkeit bewahrt.

Ist die CAVE richtig kalibriert und wird beispielsweise ein VRML-Objekt in den Szenegraph der Cave eingebunden, so entstehen für den Benutzer der Eindruck eines nahtlosen, virtuellen Objektes, das vor ihm im Raum schwebt. Da an den Shutterbrillen ein magnetischer Tracking Sensor angebracht ist, kann der Benutzer um dieses Objekt herumgehen und es von

allen Seiten betrachten, wie im Realen Raum (RR). Trotz der Möglichkeit Objekte dreidimensional zu

rezipieren und um sie herum gehen zu können, bleibt bei der Benutzung ein Gefühl der Unwirklichkeit. Dies liegt vor allem daran, dass das menschliche Auge nicht in der Lage ist, das virtuelle Objekt auf natürliche Art zu fokussieren. Die Akkommodation des Auges auf ein bestimmtes Objekt ist nicht möglich, da sich die gesamte Umgebung im Fokus befindet. Es werden alle Objekte gleich scharf dargestellt. Dieser Effekt kann zuweilen sogar zu starken Kopfschmerzen, Schwindelgefühl oder Erbrechen führen.

6.4 3D-Skizzierwerkzeuge (TUIs)

Wie auch bei 2D-Skizzen im RR kommen bei SketchApp verschiedene Zeichenwerkzeuge zum Einsatz. Hauptwerkzeug ist dabei der Stift. Er ist ca. 16cm lang, hat eine Halterung für den magnetischen Trackingsensor und birgt im inneren einen sog. Force-Sensor. Drückt man auf die Oberseite des Stiftes, so biegt sich das



Abbildung 30 Stift Prototyp



Abbildung 31 Shaper Werkzeug

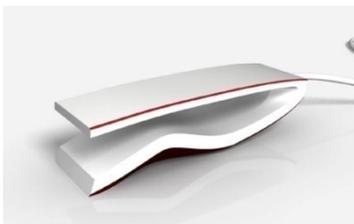


Abbildung 32 3D-Modell des Snapper

Material des Stiftes nach innen und der Sensor wird mit einer bestimmten Kraft betätigt. Je nachdem wie stark oder leicht man den Stift drückt, übermittelt der an den Force-Sensor angeschlossene Mikrokontroller einen höheren oder niedrigeren Zahlenwert an den PC. Auf diese Weise können etwa kräftigere oder feinere Linien gezeichnet werden.

Es gibt zwei Modi des Stiftes. Zum einen kann man mit ihm einfach zeichnen. D.h., wenn man den Stift drückt und gleichzeitig zieht, erscheint ein dreidimensionaler Strich in der CAVE. Zum anderen, kann man eine sog. virtuelle Spitze des Stiftes hinzu schalten. Diese wird ständig angezeigt. Dieser Modus macht den Stift zu einem Hybriden Skizzierwerkzeug, das teils real und teils virtuell ist. Die Wirkung auf den Benutzer ist in diesem Modus sehr positiv. Man fühlt sich noch mehr mit dem System verbunden, gegenüber der Verwendung des Werkzeuges ohne virtueller Spitze.

Abbildung 31 zeigt den sog. Shaper als hybrides Werkzeug. Bei ihm wird bei Benutzung eine virtuelle Fläche von der Cave hinzu projiziert. Auf ihr kann der Benutzer mit dem Stift malen und anschließend kann die auf die Fläche gemalte Kontur, durch ziehen des Shapers durch den Raum, extrudiert werden.

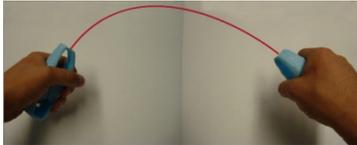


Abbildung 33 Bezier Werkzeug Prototyp

Eines der besonderen Probleme der VR sind die sog. Selektieraufgaben. Sei es in CAD-Programmen, 3D-Modelling Umgebungen oder in einem einfachen Zeichenprogramm, die im RR so einfache Handlung, stellt im VR eine besonderes Problem dar, denn es muss explizit ausgewählt werden, was zu verändern ist.

Erst wenn diese Auswahl getätigt wurde (die zuweilen recht schwierig ist), kann die eigentliche Modifikation stattfinden. Das Selektionswerkzeug für gezielte Selektionen in SketchApp ist der Snapper. Mit ihm kann eine greifähnliche Bewegung durchgeführt werden (Abbildung 32).

Das Letzte von vier realisierten Zeichenwerkzeugen, ist das Bezier-Tool. Es ist das einzige Tool, welches nur beidhändig genutzt werden kann. Mit Ihm lässt sich eine virtuelle Kurve im Raum aufspannen. Wie auch beim Shaper lässt sich dann aus dieser Kurve eine dreidimensionale Fläche extrudieren.

Weiterhin gibt es noch einen Schieberegler, mit dem es möglich ist gezeichnete Objekte in ihrer Entstehung „zurückzuspulen“. D.h., man kann die Zeichnung wie ein Video hin und her spulen.

Die Ideen zu diesen Werkzeugen sind aus einer Fokusgruppe 2007 entstanden (Israel & Zacharias 2007).

Die Wichtigsten Interaktionen beim Skizzieren waren den Befragten hierbei:

<i>Funktionsgruppe</i>	<i>Beispiel</i>
<i>Klassisches Zeichnen</i>	<i>Zeichnen, radieren</i>
<i>Klassische Modellierungstechniken</i>	<i>Abtragen, auftragen</i>
<i>Klassische CAD Modellierung</i>	<i>Geometrische Standardvolumen erstellen und manipulieren, skalieren, spiegeln, schneiden, kopieren</i>
<i>Neue 3D Zeichentechnik</i>	<i>Abzeichnen realer Gegenstände, rückgängig Funktion (Undo), Einsatz virtueller Schablonen</i>
<i>Neue 3D Modellertechniken</i>	<i>Kneten, stauchen, zerren, drücken, falten, dehnen</i>
<i>Abstraktionstechniken</i>	<i>Modelle überlagern, Ungenauigkeit einführen, Zwiebelschichten mit Transparenz, Nebelfiguren, Partikel aufträufeln</i>
<i>Dynamik</i>	<i>Darstellen von Bewegungsspuren und Prozesswirkungen, History: Bewahrung der Varianten einer Skizze</i>
<i>Umgebungsbedingungen</i>	<i>Schritt- und Körpermaße, Kontext einbeziehen, kreative Umgebung, reale Proportionen</i>

Tabelle 3 Ideen zu Interaktionstechniken des 3D-Skizzierens

Weiterhin wurde die Möglichkeit mit dem Raum zu arbeiten, 1:1 Skizzieren zu können, vorhandene Objekte in den VR zu holen und räumliche Skizzen zu verformen als positiv nutzbar genannt.

Als negativ wurde das fehlende Material („Material ist Ideengebend“) genannt sowie die fehlende Auflagefläche und die Bezugsebenen („Kein Blatt Papier, das die Grenze der Zeichnung bedeutet.“, „ Ein Punkt im Raum wieder zu finden, ist schwierig.).

Eine häufige Tätigkeit, die leider nicht in dieser Fokusgruppe erkannt wurde, ist die Möglichkeit die dreidimensionalen Skizzen mit Anmerkungen versehen zu können.

6.5 Schwächen des aktuellen Skizziersystems

Das derzeitige genutzte System für das Tracking des Stiftes ist ein MotionStar, der Firma Ascension. Das System ist ein magnetisches Tracking System. Der Arbeitsbereich des 6DOF Trackers beträgt +-3.0m kann aber durch den Tandembetrieb auf 3m x 4,88 m erweitert werden. Zu den Vorteilen lässt sich bei diesem System vor allem die hohe Frequenz der Messungen nennen. Im Datenblatt wird sie mit 144 Messungen pro Sekunde angegeben und liegt somit um ein vielfaches höher, als die Framerate einer herkömmlichen Kamera. Als Output liefert das System X, Y und Z Koordinaten sowie die jeweiligen Rotationswinkel im Euklidischen-Koordinatensystem oder wahlweise Quaterniones. An das System können bis zu 18 Sensoren angeschlossen werden, welches für die bestehenden und angedachten Skizzierwerkzeuge mehr als ausreichend ist. In Bezug auf das optische Tracking-System, sind 18 eindeutig zu unterscheidende Targets, so nennt man die mit Sensoren bestückte Objekte, sehr hoch und bereits Grenzwertig in der Realisierbarkeit.

Einziges großes Manko des magnetischen Tracking-System ist die hohe Empfindlichkeit gegenüber anderen Magnetfeldern sowie Metallobjekten. Beides lässt sich in Cave-Nähe kaum vermeiden, da etwa Gehäuse von Rechnern standardgemäß aus Metall bestehen und aus praktischen Gründen nahe der Cave installiert werden müssen.

Abbildung 34 bis Abbildung 37 zeigen deutlich die Schwächen des magnetischen Tracking Systems auf. In 729 Messpunkten wurde hier die Abweichung zwischen tatsächlichen und gemessenen Koordinaten ermittelt, die zu der Erkenntnis führten, dass sich das magnetische Feld vor allem im hinteren Teil der CAVE besonders verschlechtert.

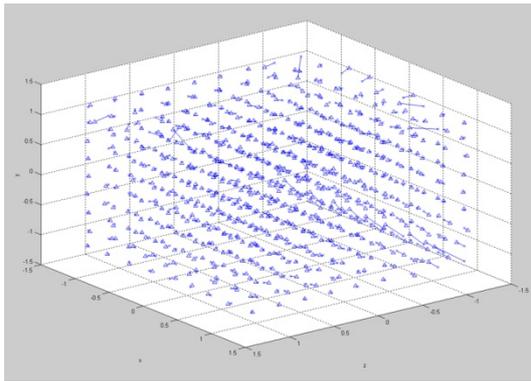


Abbildung 34 Verzerrung des magnetischen Feldes

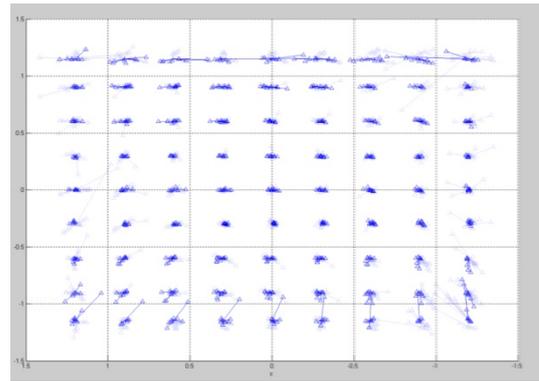


Abbildung 35 Verzerrung in XY-Ebene

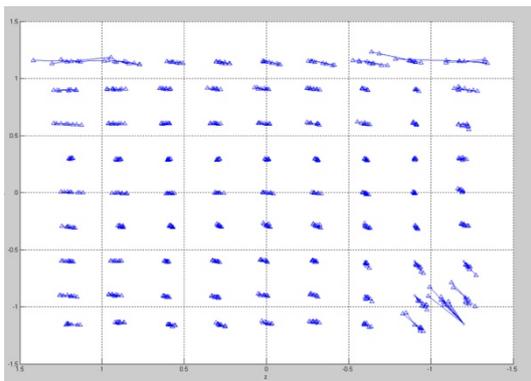


Abbildung 36 Verzerrung in YZ-Ebene

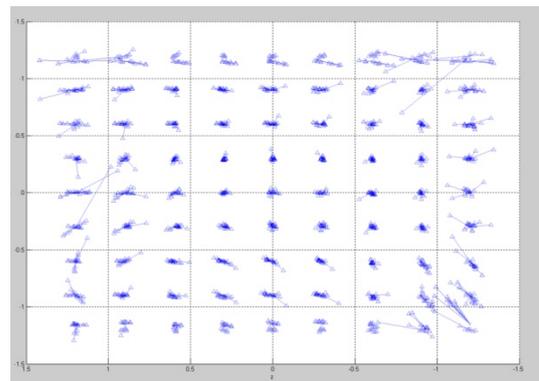


Abbildung 37 Verzerrung in XZ-Ebene

Neben den räumlichen Abweichungen in den Trackingdaten bestehen auch Verzögerungen in zeitlicher Dimension, die besonders dann offensichtlich werden, wenn schnellere Skizzierbewegungen durchgeführt werden. Dabei hat die Verzögerung des Systems für den Skizzierenden den Effekt, seine Externalisierung der Gedanken und sein Entwurfsdenken zu behindern, da zusätzliche kognitive Prozesse zur Kompensierung dieser Situation stattfinden. Messungen von Rasim Korkot haben ergeben, dass derzeit die Verzögerungen bis zu 250ms groß sein können (Korkot 2008).

Weitere Schwächen des derzeitigen Systems gehen über das magnetische Tracking hinaus und betreffen Wahrnehmungspsychologische Probleme von virtuellen Umgebungen. So ist es ein allgemein bekanntes Problem der Akkommodation der Augen, dass Entfernungen zu virtuellen Objekten überschätzt werden. Gerade in Bezug auf das Skizzieren ist dies eins der größten Probleme des Systems, da hierdurch der Externalisierungsprozess beim Skizzieren stark beeinflusst wird und somit Sinn und Zweck des dreidimensionalen Skizzierens im Kerne bezweifeln lässt.

7 Experimente (Analyse)

Da der Skizzierprozess ein ureigener menschlicher Prozess ist, sollte ihm beim Systemdesign von SketchApp so weit wie möglich entgegen gekommen werden. Somit wurde auch dem, in der Softwareentwicklung neue Ansatz, des User-Centered-Design Folge geleistet, der die Bedürfnisse, Forderungen und Grenzen des Users mit einbezieht. Dieser extensiven Philosophie in der Softwareentwicklung nachkommend, wurden eigene Experimente durchgeführt. Von besonderem Interesse waren die Geschwindigkeiten des Zeichenstiftes beim dreidimensionalen Skizzieren und mögliche Bewegungen in der Cave.

7.1 Maximale Handgeschwindigkeiten

Untersucht wurde die maximal mögliche Handgeschwindigkeit mit Hilfe einer Highspeed Kamera, die es erlaubt bis zu 8000 Bilder pro Sekunde aufzunehmen. Bei Durchführung des Experiments wurde die Kamera auf 500fps eingestellt, da höhere Bildraten den aufnehmbaren Bildausschnitt stark einschränken und zusätzliche Scheinwerfer aufgrund der kurzen Belichtungszeit erfordert hätten. Die zu messenden Handbewegungen wurden im stehen von einer mit der Skizzieranwendung vertrauten Testperson durchgeführt. Die Bewegungen wurden einzig aus der Armbewegung heraus getätigt und nicht etwa durch zusätzliche Rotation des eigenen Körpers weiter beschleunigt. Auch ist anzumerken, dass die Bewegungen mit der dominierenden Hand (in diesem Fall die Rechte) gemacht wurden.

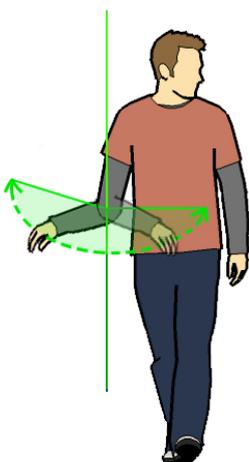


Abbildung 39 Bewegung entlang der X-Achse

Ausführung dieser extrem schnellen Bewegung fiel auf, dass die Form des Stiftes eine unbewusste, automatische Versteifung des Handgelenkes begünstigt. Ob die Versteifung bei langsameren Bewegungen ebenfalls auftritt, wäre noch zu Prüfen.

Entlang der X-Achse des Koordinatensystems des Probanden, passierte die transversale rechts-links Bewegung hauptsächlich durch die Rotation des Oberarmes. Es wurde dabei versucht, in einer Ebene parallel zum Boden zu bleiben, was auch während der Durchführung des Experimentes sehr gut gelang.

Entlang der Y-Achse wurde eine Art „Hackbewegung“ aus dem Ellbogengelenk heraus (aber steifem Handgelenk) mit minderem Einsatz der Schulter getätigt. Bei der

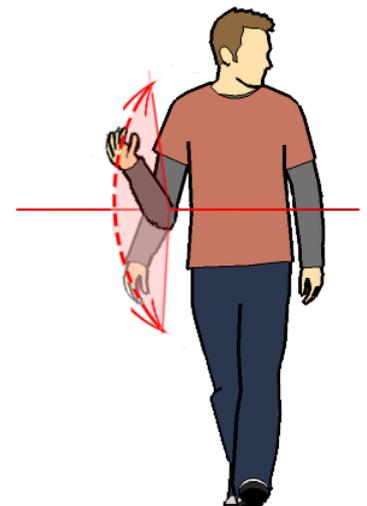


Abbildung 38 Bewegung entlang der Y-Achse

Die Bewegung entlang der in Z-Achse wies eine „schubsende“ Charakteristik auf. Ellbogen und Schulter waren in etwa gleich aktiv. Es war bei Durchführung des Experiments die spürbar anstrengendste Bewegung.

Desweiteren wurden die Bewegungen in aktive und passive Bewegungen eingeteilt, da bereits während der Durchführung des Experiments festgestellt wurde, dass die als Erstes durchgeführte schnelle Bewegung durch bewussten Einsatz der Muskelkraft durchgeführt wurde, während die darauffolgende zweite Bewegung eher einem Zurückfedern aufgrund der ersten Bewegung glich und somit zwangsläufig langsamer sein musste. Dies bestätigt sich auch durchweg in den vorliegenden Messergebnissen. Die passive Bewegung war immer langsamer als die aktive.

In X-Richtung war die von der Körpermitte nach außen gerichtete (links-rechts) Bewegung die Aktive. In Y-Richtung die von Oben nach Unten gerichtete Bewegung. In Z-Achse war die Bewegung vom Körper weg die aktive.



Abbildung 40 Bewegung entlang der Z-Achse

Bei Durchführung des Experiments wurde der originale SketchApp-Stift mit angeschlossenem magnetischem Empfänger in der Hand gehalten um die realen 3D Zeichenbewegungen beizubehalten. Die Nachfolgende Tabelle zeigt die gemessenen Ergebnisse:

Translation	X-Achse	Y-Achse	Z-Achse
1. aktiv	54ms	42ms	-
1. passiv	55ms	48ms	111ms
2. aktiv	80ms	56ms	120ms
2. passiv	78ms	56ms	120ms
3. aktiv	60ms	50ms	121ms
3. passiv	62ms	56ms	131ms
4. aktiv	54ms	54ms	116ms
4. passiv	61ms	57ms	132ms
5. aktiv	66ms	44ms	125ms
5. passiv	66ms	-	-
Mittelwert aktiv	62,8ms	49,2ms	120,5ms
Standardabweichung	(+ -10,83ms)	(+ - 6,10ms)	(+ - 3,7ms)
Mittelwert passiv	64,4ms	54,25ms (+-	123,5ms
Standardabweichung	(+ - 8,56ms)	4,19ms)	(+ - 9,95ms)
Geschw. aktiv	576,11 cm/s	744,86 cm/s	316,02 cm/s
Geschw. passiv	561,8 cm/s	675,52 cm/s	308,34 cm/s

Tabelle 4 Maximale Translationsgeschwindigkeit der Hand

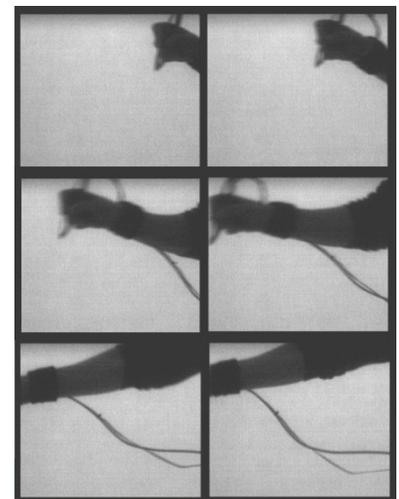


Abbildung 41 Bilder der Highspeed-Kamera

Es ist festzustellen, dass alle maximalen Handgeschwindigkeiten in X- und Y-Richtung ähnlich schnell sind, wohingegen sich die Bewegung in Z-Richtung sogar in um die Hälfte von ihnen unterscheidet. In Bezug auf das zu entwerfende System, könnte dies in Form bestimmter anzuwendender Heuristiken bei der Pose Estimation oder als Parameter für zustandsschätzende Algorithmen wie denen der linearen Prädiktion, Kalman oder Partikel-Filter von Interesse sein.

Rotation	X-Achse (+35°/-32°)	Y-Achse¹ (+ 25°/-25°)	Z-Achse² (+102°/-72°)
1. passiv	164ms	140ms	-
2. aktiv	156ms	128ms	124ms
2. passiv	174ms	136ms	-
3. aktiv	166ms	140ms	110ms
3. passiv	144ms	110ms	-
4. aktiv	174ms	160ms	120ms
4. passiv	124ms	224ms	-
5. aktiv	132ms	196ms	110ms
5. passiv	160ms	224ms	-
6. aktiv	154ms	194ms	-
6. passiv	130ms	-	-
<i>Mittelwert</i>	<i>152,55ms</i>	<i>165,2ms</i>	<i>116ms</i>
<i>Standardabweichung</i>	<i>(+/-17,67ms)</i>	<i>(+/- 41,2ms)</i>	<i>(+/-7,12ms)</i>
Mittlere Rotations-Geschwindigkeit	439,2 °/s	302,66 °/s	1500 °/s
Mittlere Bahn-Geschwindigkeit	141,12 cm/s	48,62 cm/s	481,73 cm/s

Tabelle 5 Maximale Rotationsgeschwindigkeit der Hand

Bedenkt man, dass eine Low-Cost Kamera nur 30fps liefert, also nur jede 33,33ms ein Bild liefert, so stellt man fest, dass bei schnellen Bewegungen leicht bis zu 25cm Distanz zwischen zwei Bildern zurückgelegt werden kann. Dieser Schluss berücksichtigt allerdings nur die im Experiment durchgeführten Bewegungen. Kombinierte Skizzierbewegungen, die zudem Drehungen des eigenen Körpers oder anderen Gliedmaßen beinhalten, würden diese maximalen Geschwindigkeiten noch erhöhen. Zu erwartende Geschwindigkeiten aus Rotationen können der Tabelle 5 entnommen werden. Die Rotation um die X-Achse wurde bei diesem Experiment aus dem Handgelenk ausgeführt. Hier war eine Rotation nach Oben um +35° und nach unten um -32° möglich. Auch die Rotation um die Y-Achse wurde nur durch Bewegung des Handgelenkes (+25°/-25°) getätigt. Diese Bewegung erschien sehr mühselig und unnatürlich und war deshalb auch die langsamste in ihrer Geschwindigkeit. Als schnellste Bewegung konnte die Drehung um die Z-Achse durchgeführt werden, da sie aus dem Unterarm heraus entstand. Mit ihr ist eine Rotation von 50° bei 30fps zwischen zwei aufeinanderfolgenden Bildern möglich und bewegt sich mit ihrer Geschwindigkeit somit auf gleicher Höhe der Bewegungen, die eher positionierende Eigenschaften hatten.

¹ „+“ = links, „-“ = rechts In der Messreihe ist ein deutlicher Ermüdungseffekt zu erkennen, der auch schon während der Durchführung des Experiments aufgefallen ist.

² „+“ = im Uhrzeiger aus Ich-Perspektive, „-“ = gegen den Uhrzeiger

Bedenkt man, dass eine Low-Cost Kamera nur 30fps liefert, also nur jede 33,33ms ein Bild liefert, so stellt man fest, dass bei schnellen Bewegungen leicht bis zu 25cm Distanz zwischen zwei Bildern zurückgelegt werden kann. Dieser Schluss berücksichtigt allerdings nur die im Experiment durchgeführten Bewegungen. Kombinierte Skizzierbewegungen, die zudem Drehungen des eigenen Körpers oder anderen Gliedmaßen beinhalten, würden diese maximalen

Die Durchführung dieses Experimentes der maximalen Handgeschwindigkeiten zeigt in etwa, mit welchen Grenzwerten bei Skizzierbewegungen zu rechnen ist. Es können nun in Frage kommende Techniken zum Tracking dieser Bewegungen ausgewählt werden. Z.B. schließen die Ergebnisse das Tracking durch MeanShift oder CamShift Algorithmen weitestgehend aus (bei Kameras mit nur 30fps), da die Bewegungen potentiell in der Lage sind von linken Bildrand der Kamera zum Rechten innerhalb von nur einem Bild zu gelangen. Das würde für den Meanshift-Algorithmus ein nötiges Suchfenster von der Größe des gesamten Kamerabildes (640x480px) bedeuten, was beim Tracking mehrerer Blobs zu Problemen führt (nähere Erläuterungen siehe Kapitel 9.4.1). Auch ist anzunehmen, dass die Echtzeitfähigkeit aufgrund des großen „Search-Window“ nicht mehr zu realisieren wäre. Bei den unter Kapitel 6.1 genannten Randbedingungen schließen diese Erkenntnisse eine Ermittlung der Pose mittels MeanShift oder CamShift Tracking von Beginn an aus.

7.2 Mittlere Handgeschwindigkeiten beim 3D-Skizzieren

Aus den vorangegangenen Experimenten konnte einige Fakten über etwaige Skizzierbewegungen gewonnen werden und eine Vorstellung darüber entwickelt werden, worum es sich bei menschlichen Bewegungen eigentlich handelt und mit welchen Geschwindigkeiten man rechnen muss. Da das erste Experiment eher die Ausnahmen, als die Regel analysierte, soll in diesem Kapitel ein Einblick in die in-vivo Skizzierbewegungen gegeben werden.

Die ermittelten Geschwindigkeiten resultieren im Gegensatz zu den im Kapitel 7.1 stark beschränkten Bewegungen aus der freien Skizzierhaltung heraus. Diese ist im Normalfall aufrecht stehend auf Hüft- bis Augenhöhe (vgl. Kapitel 7.3) und wird weitestgehend im Zentrum der Cave ausgeführt.

Im ersten von zwei Experimenten, skizzierte eine Testperson ohne bestimmtes Ziel in den dreidimensionalen Raum hinein³. Dabei stand die Testperson in Richtung CAVE-Eingang (positive Z-Richtung). Koordinatensystem der Person und der CAVE stimmten also überein. Weiterhin wurde im Stehen gezeichnet. D.h., die Testperson bewegte sich nicht durch den Raum. Die Skizzierbewegungen wurden über den magnetischen Tracker sensiert und gespeichert.

Rotationen des Körpers waren im Gegensatz zum ersten Experiment möglich sowie alle anderen Bewegungen auch (z.B. hocken, bücken, etc.). In der Log-Datei wurden X-, Y- und Z-Koordinaten sowie die zugehörige Zeit (in μs) gespeichert, woraus sich folgende Diagramme ergaben.

³ APPENDIX A zeigt die entstandene Skizze.

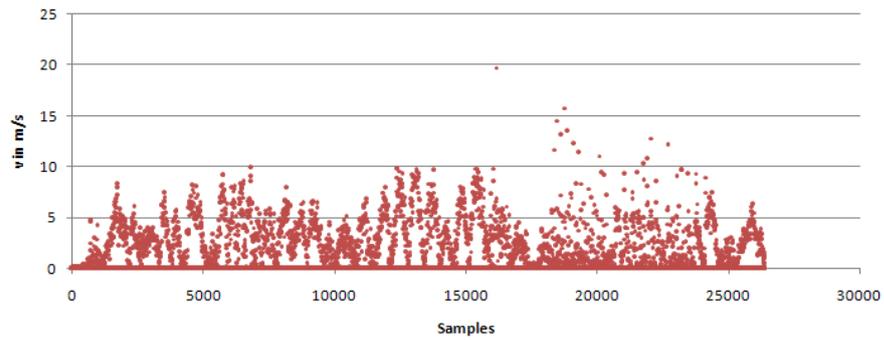


Abbildung 42 Geschwindigkeiten entlang der X-Achse

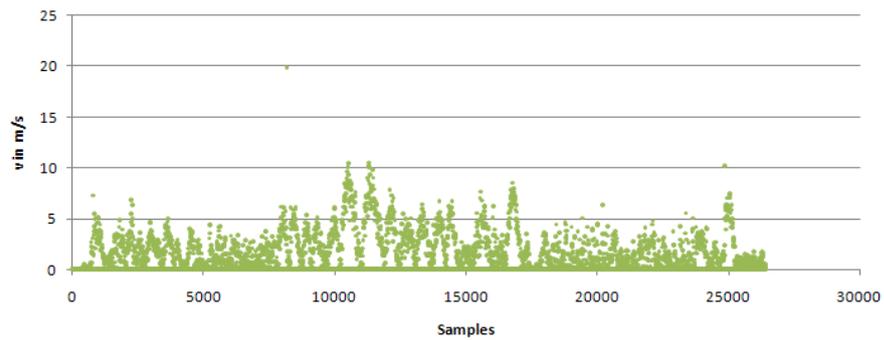


Abbildung 43 Geschwindigkeiten entlang der Y-Achse

Aus den Daten bestätigen sich einige bereits aus dem in Kapitel 7.1 gewonnenen Erkenntnisse. So wird z.B. klar, dass Bewegungen in X- und Y-Richtung etwa gleiche Geschwindigkeiten haben, während solche in Z-Richtung, im Mittel, wesentlich langsamer sind.

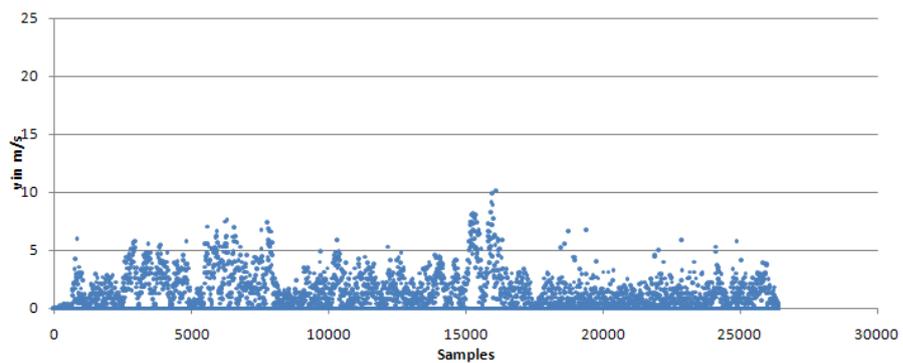


Abbildung 44 Geschwindigkeiten entlang der Z-Achse

Die gezeichneten Trajektorien sind in den Geschwindigkeitsdiagrammen deutlich sichtbar. Man erkennt, dass ein gesetzter Strich mit einer Geschwindigkeit von 0 cm/s beginnt und endet. Weiterhin wurde bei der Auswertung festgestellt, dass es auch in dieser Situation typische Bewegungen gibt, die schneller als exakt entgegengesetzte ausgeführt werden, also den aktiven und passiven Bewegungen aus dem

vorangegangenen Experiment gleichen. So sind Bewegungen in X-Richtung schneller, wenn sie zur Körpermitte (bei Rechtshändern also nach links) ausgeführt werden. In Y-Richtung sind Bewegungen nach Oben schneller und in Z-Richtung die Bewegungen nach Vorne.

Die aus den Teilgeschwindigkeiten in X-, Y- und Z-Richtung zusammengesetzte Gesamtgeschwindigkeit wird nachfolgend dargestellt:

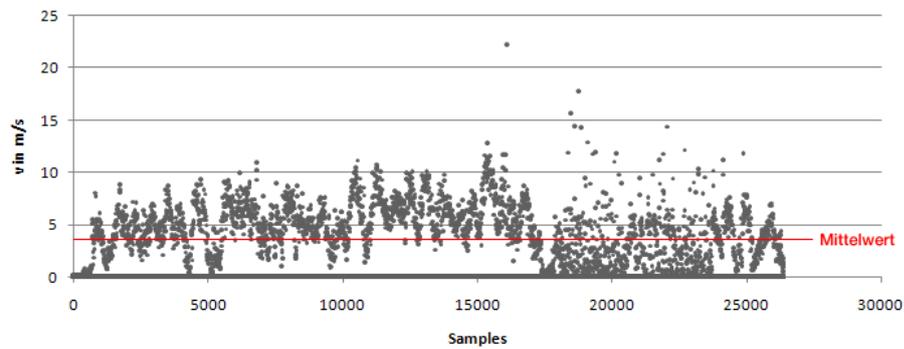


Abbildung 45 Gesamtgeschwindigkeit

Wie man im obigen Diagramm sieht, gibt es zwei Geschwindigkeitsphasen in denen skizziert wurde. Bis ca. Sample 18000 wurde mit einer Gesamtgeschwindigkeit von etwa 5m/s skizziert, wohingegen die restlichen Bewegungen deutlich darunter lagen. Grund hierfür mag der bereits erschöpfte, zur Verfügung stehende, Skizziererraum sein. D.h., es wurde bereits soviel gezeichnet, dass nach freiem Platz zum Zeichnen gesucht werden muss, welches anscheinend die Zeichengeschwindigkeit beeinflusste. Der Mittelwert der Geschwindigkeit aller Samples lag bei 4,39m/s.

Im zweiten Experiment sollten sinnvolle dreidimensionale Objekte gezeichnet werden. Dabei wurde bei allen Skizzen außer der „Amateurskizze“ (siehe APPENDIX B) zuvor überlegt, was zu malen sei. D.h., die Skizzen wurden nicht im Sinne der Selbstkommunikation „entwickelt“ sondern bestanden bereits als mentales Modell. In wie weit sich diese Gegebenheit auf die Zeichengeschwindigkeit auswirkt ist nicht genau zu sagen, da keine Vergleichstests durchgeführt wurden. Zu vermuten ist jedoch, dass diese evtl. schneller gezeichnet wurden als eine entstehende Skizze, bei der das Ziel noch nicht klar ist. Die niedrigeren Geschwindigkeitswerte der „Amateurskizze“ in Tabelle 6 könnten allerdings ein Indiz dafür liefern.

Abbildung 46 bis Abbildung 49 zeigen exemplarisch ein in der Cave skizziertes Objekt.

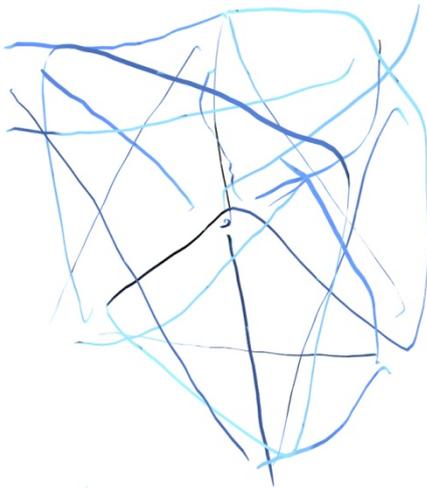


Abbildung 46 Quader - isometrische Ansicht

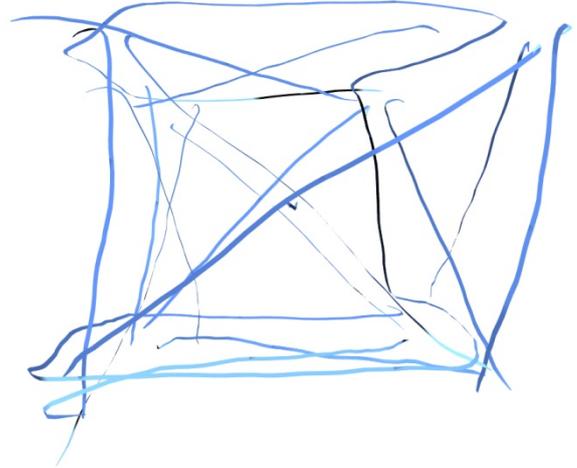


Abbildung 47 Quader - XY-Ebene

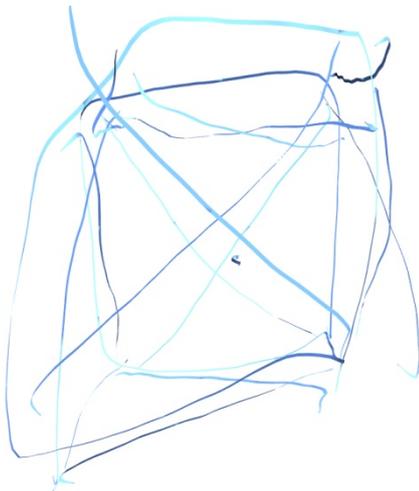


Abbildung 48 Quader - ZY-Ebene

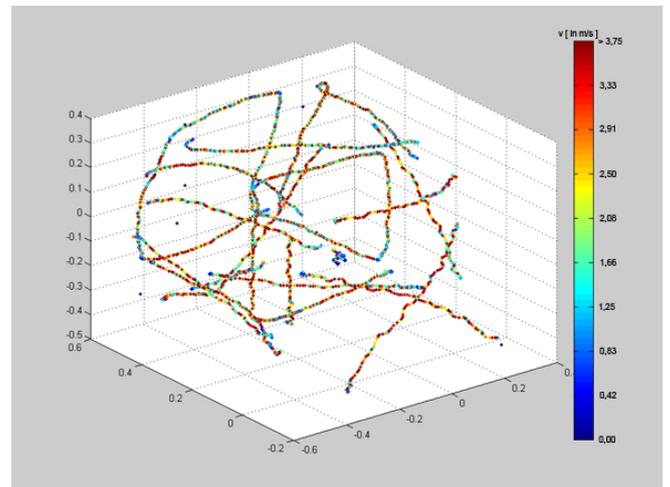


Abbildung 49 3D-Plot der Geschwindigkeiten

Abbildung 49 zeigt farblich die berechneten Geschwindigkeiten als 3D-Plot. Die Trajektorien sind deutlich zu erkennen sowie erste Auffälligkeiten von gezeichneten Objekten, wie beispielsweise niedrige Geschwindigkeiten am Anfang sowie am Ende einer Linie.

Die nachfolgende Tabelle fasst die mittleren Geschwindigkeiten der insgesamt fünf verschiedenen 3D-Skizzen zusammen⁴. Dabei ist vor allem der Median als aussagekräftigster Wert zu berücksichtigen, da der Mittelwert durch die teilweise sehr großen Ausreißer-Koordinaten weniger robust ist und es sich bei der Verteilung der Koordinaten-Samples nicht um eine Normalverteilung handelt (vgl. Abbildung 57).

⁴ Screenshots und Geschwindigkeitsplots der Skizzen befinden sich im APPENDIX B

	<i>Quader</i>	<i>Spirale</i>	<i>Kurven</i>	<i>Schrift</i>	<i>Amateurskizze</i>
Sampleanzahl	3312	1924	2338	5747	11340
Zeitraum t	117,28 s	46,03 s	69,96 s	148,53 s	321,1 s
Median V(x)	16,02 cm/s	143,87 cm/s	26,32 cm/s	45,45 cm/s	14,05 cm/s
Median V(y)	13,32 cm/s	42,57 cm/s	30,81 cm/s	76,92 cm/s	14,48 cm/s
Median V(z)	12,48 cm/s	141,875 cm/s	19,89 cm/s	62,5 cm/s	9,05 cm/s
Median gesamt	39,56 cm/s	236,07 cm/s	78,16 cm/s	141,3 cm/s	32,84 cm/s
Mittelwert V(x)	23,64 cm/s	164,21 cm/s	46,77 cm/s	72,44 cm/s	26,39 cm/s
Mittelwert V(y)	20,52 cm/s	62,57 cm/s	47,94 cm/s	97,97 cm/s	26,7 cm/s
Mittelwert V(z)	20,41 cm/s	156,36 cm/s	44,48 cm/s	81,41 cm/s	16,9 cm/s
Mittelwert gesamt	44,93 cm/s	264,9 cm/s	99,25 cm/s	172,67 cm/s	48,32 cm/s
Standardabw. gesamt	38,91 cm/s	211,88 cm/s	120,16 cm/s	180,24 cm/s	55,81 cm/s
75%-Quantil gesamt	53,73 cm/s	293,25 cm/s	117,60 cm/s	197,64 cm/s	62,35 cm/s
25%-Quantil gesamt	26,00 cm/s	186,15 cm/s	47,06 cm/s	95,47 cm/s	16,51 cm/s

Tabelle 6 Mittlere Skizziergeschwindigkeiten

Während dieses Experimentes fielen mehrere Effekte, die sich in den Skizzen widerspiegeln, auf. Die gezeichneten Linien beginnen und enden sehr häufig mit einer geringen Geschwindigkeit. Dies liegt an der Art und Weise der Abfolge der Schritte, wie der Stift bedient wird.

Meist sind diese:

- Positionieren des Stiftes
- Drücken des Druckknopfes am Stift
- Zeichnen der Linie
- Stoppen des Stiftes
- Loslassen des Druckknopfes am Stift
- Stift neu positionieren



Abbildung 50 Schriftzug geprägt durch anatomische Gegebenheiten

Nach dem Drücken und vor dem Loslassen des Knopfes verweilte der Stift meist ein wenig, als ob die Testperson ein Stoppen und Beginnen des Striches besonders „gut“ machen wollte. Kurven werden zudem im Allgemeinen langsamer gezeichnet als Geraden, besonders dann, wenn sie auf eine Gerade folgen.

Bei der „Spiralen-Skizze“ (vgl. Tabelle 6) ist deutlich zu erkennen, dass diese entlang der Y-Achse der CAVE skizziert wurde. Während Bewegungen in X- und Z-Richtung sehr hoch sind, ist die Bewegung in Y-Richtung von mäßiger Geschwindigkeit.

Die „Schrift-Skizze“ (vgl. Abbildung 50) zeigt sehr deutlich, dass Skizzen durch anatomische Gegebenheiten stark beeinflusst werden. So wird deutlich, dass das intendierte Schreiben auf eine „unsichtbare“ Wand ungewollt zu einem Schriftzug auf einer gebogenen „unsichtbaren“ Wand führt. Beide in diesem Kapitel beschriebenen Experimente, zeigen, dass die in Kapitel 7.1 ermittelten Maximalgeschwindigkeiten zwischen 308 cm/s – 744cm/s im Durchschnitt nicht erreicht werden, obwohl zusätzliche Rotationen möglich waren und somit die die Bewegung potentiell zusätzlich beschleunigt hätten werden können. Einzig die „Spiralen-Skizze“ kommt mit einem Median von 236 cm/s in die Regionen der langsamsten Maximalgeschwindigkeit. Da sie aber aus einer Vorüberlegung und einer bereits geübten Skizzierbewegung heraus entstand, stellt sie auch eine Ausnahme dar. Somit können die Maximalgeschwindigkeiten aus Kapitel 7.1 als Ausnahmewerte betrachtet werden.

Übliche Geschwindigkeiten beim Skizzieren bewegen sich demnach zwischen ca. 15 – 60 cm/s, wie es die „Amateurskizze“ demonstriert. In Bezug auf das optische Tracking wäre ein theoretischer Weg von 2cm (30fps) zwischen zwei aufeinanderfolgenden Bildern möglich. Die funktionale Anforderung von max. 2cm Abweichung in der Translation wäre also rein hardwaretechnisch, also ohne zusätzliche eingesetzte präzisierende Algorithmen, einzuhalten.

7.3 Betrachtung verschiedener Räume der Cave

Während der Experimente in der Cave hat sich herausgestellt, dass die Cave einige Schwächen in Bezug auf den gewünschten Interaktionsraum bietet. Ihre Raumgestalt ist maßgeblich durch die einfachere technische Realisierung geprägt und weniger auf den Anwender, welcher in ihr agiert, angepasst. So wurden z.B. die Projektionsflächen mit Kantenlänge 2,5m in Quaderform angeordnet. Dem steht das natürliche Gesichtsfeld des Menschen gegenüber, welches in horizontaler Ausbreitung (190°) weit mehr Informationen erfassen kann als in vertikaler Richtung (ca. 100°-120°). Technische Gründe standen bei der Konstruktion der CAVE anscheinend im Vordergrund. In wie weit die Interaktionen des Nutzers der Skizzieranwendung durch diese technischen Voraussetzungen beschränkt werden, wird in diesem Kapitel beschrieben.

Dass bereits Anstrengungen unternommen werden die technisch dominierte Umgebung CAVE menschlicher zu machen, zeigt beispielsweise die „VirtuElle“ von Poupyrev. Er schlägt wegen der Körperunabhängigkeit zwischen Pixel und Meter eine neue Maßeinheit vor, die vor allem in immersiven Umgebungen sinnvoll sein kann. Es handelt sich hierbei um ein benutzerzentriertes Koordinatensystem, bei dem die menschliche Elle als Maßeinheit verwendet werden soll. Eine Formalisierung von Skizzen ohne zusätzliches Lineal als Tool wäre somit möglich.

Die hier untersuchten Limitationen beschränken sich vornehmlich auf den zur Verfügung stehenden Interaktionsraum der CAVE und seine mögliche Wirkung auf die Interaktion in SketchApp. Auf diese Weise kann gezeigt werden, wo Interaktionsareale mit erhöhter Interaktion sind und solche mit geringerer.

7.3.1 Wahrnehmungsraum des Anwenders in Bezug zur CAVE

Abbildung 51 zeigt ein Interaktionsmodell aus dem Bereich der Kognitionswissenschaft. Es verdeutlicht die verschiedenen Wahrnehmungsräume und ordnet diesen bestimmte neuropsychologische Prozessen von Stimuli und Reaktionen zu.

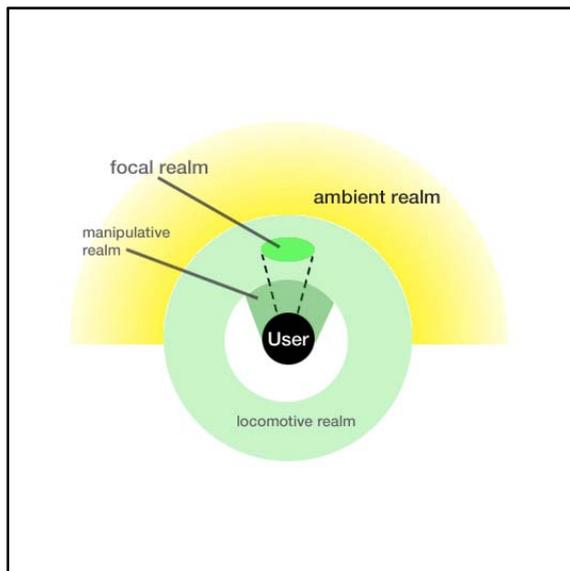


Abbildung 51 Theoretisches räumliches 3D Interaktionsmodell (Previc 1998)

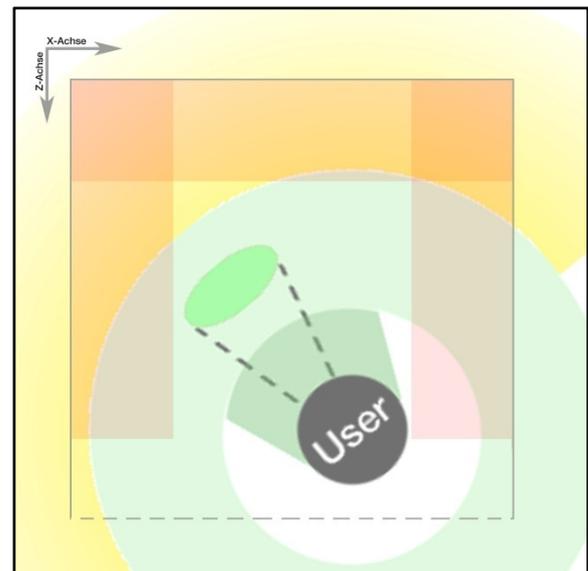


Abbildung 52 Mapping des räumlichen Interaktionsmodell auf die CAVE

Auf diese Weise kann z.B. entschieden werden, welche Informationen wo angeordnet werden sollten, um beispielsweise als eher hintergründig als vordergründig wahrgenommen zu werden. Bildet man das Modell von Previc maßstabsgetreu auf die Maße der CAVE ab, so fällt auf, dass die meisten Wahrnehmungsbereiche (realms) außerhalb der Cave fallen. Der „manipulative realm“ mit einer Größe von 0m – 2m und einer lateralen Ausbreitung von $\pm 60^\circ$ (May 2006) ist der einzige Wahrnehmungsraum, der komplett in die CAVE fallen kann. Er wird als der Raum verstanden, der in Armreichweite liegt, auch wenn diese Armreichweite von 2m nur mit einem entsprechenden Werkzeug erreicht werden kann. Was die visuelle Perzeption betrifft, ist dieser Bereich vor allem für Formen, Tiefen und Bewegungen spezialisiert. Der „focal realm“ hat eine Wahrnehmungsreichweite von 20cm bis zu einer Entfernung, ab der das fokussierte Objekt nicht mehr auflösbar ist. Im Gegensatz zum „manipulative realm“ hat dieser

Bereich eine Ausrichtung im oberen Bereich (Herbon & Rötting 2007). Die beiden anderen Wahrnehmungsbereiche fallen beide außerhalb des möglichen Interaktionsraums.

Allgemein kann gesagt werden, dass die CAVE zwar Raum für die wichtigsten Wahrnehmungen wie für das Suchen, das Objekt erkennen, Farbe, Form und Konturanalyse bietet, dieser Raum mit 2,5m jedoch sehr knapp bemessen ist, wenn man bedenkt, dass wenn man in der Mitte der Cave steht alle vier Wahrnehmungsbereiche durch die CAVE-Wände beschränkt werden. In wie weit sich dies auf die kognitive Performance auswirkt und in wie weit durch die CAVE Objekte, die in den „ambient realm“ fallen simulieren kann, bleibt dahin gestellt. Es ist jedoch anzunehmen, dass die Positionierung von virtuellen Objekten sinnvollerweise so gewählt werden, dass sie in die jeweiligen für eine bestimmte Sache spezialisierten Bereiche fallen.

7.3.2 Analyse des tatsächlichen Interaktionsraum der CAVE

Neben diesen theoretischen Betrachtungen wurde die tatsächliche Nutzung der Cave beim Skizzieren analysiert. Die hierzu herangezogenen Trackingdaten stammen aus den Versuchen von Israel, die er in (Israel, Korkot et al. 2008) beschreibt. Die Punktwolken unten zeigt die Ausnutzung des Interaktionsraumes⁵ jeweils auf eine 2D-Ebene projiziert.

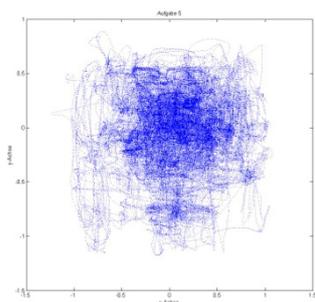


Abbildung 53 Alle Daten von Aufg. 5 (XY-Ansicht)

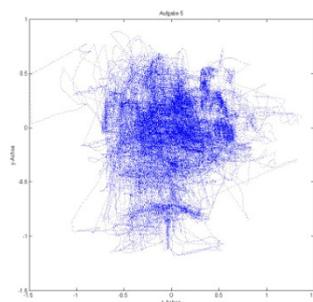


Abbildung 54 Alle Daten von Aufg. 5 (ZY-Ansicht)

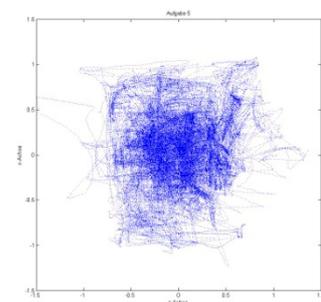


Abbildung 55 Alle Daten von Aufg. 5 (XZ-Ansicht)

⁵ Plots aller weiteren Aufgaben sind im APPENDIX D aufgeführt

Abbildung 56 zeigt die Punktwolke von Aufgabe 1, in der 24 Möbeldesigner einen neuartigen und stabilen, von der Decke herablassbaren Stehtisch entwerfen sollten. Man erkennt sehr deutlich, dass ein Großteil der Skizzen von den Testpersonen innerhalb eines Radius von 0,75m um den CAVE Mittelpunkt herum skizziert wurden. Also in der XZ-Ebene (parallel zum Fussboden) im Allgemeinen 50cm Abstand von der Projektionswand beim Skizzieren eingehalten wurden. Dieser Raum gilt somit als weniger attraktiv für zu zeichnende Objekte, was sich auch durch die Auswertung aller fünf Zeichenaufgaben bestätigt. Weiterhin ist festzustellen, dass XY- und ZY-Ansichten einander sehr stark gleichen, wie die Abbildung 53 und Abbildung 54 zeigen.

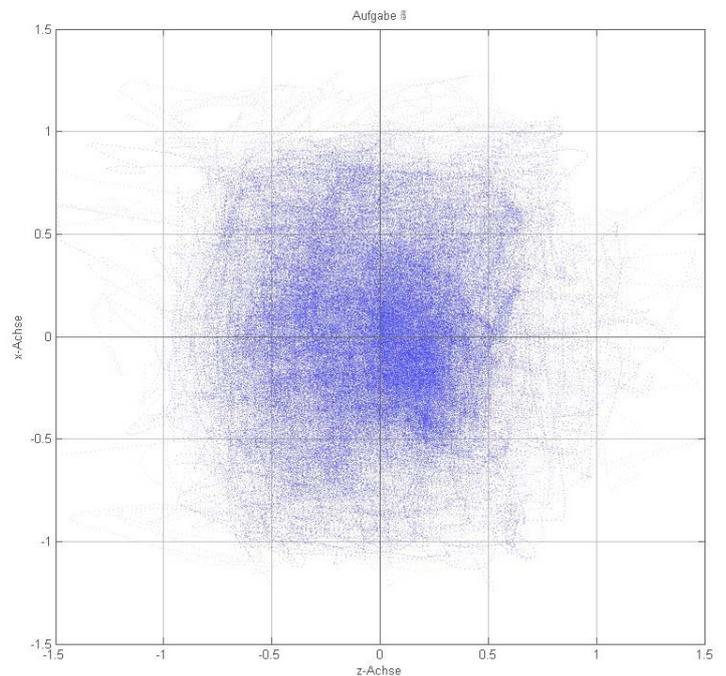


Abbildung 56 Alle Daten von Aufg. 1 (XZ-Ansicht)

Um besonders frequentierte Bereiche auszumachen, wurden die Daten aus allen fünf Versuchen zusammengeführt und in einem Histogramm dargestellt (siehe Abbildung 57). Es zeigt, dass es in X-Richtung eine Präferenz gibt, links neben der CAVE-Mitte zu zeichnen. Der Grund dafür ist unbekannt. In der Höhe (Y-Achse) dominieren die Koordinaten zwischen dem Koordinatenursprung, also 1,25m über dem Boden bis 1,50m über dem Boden. Das ist in etwa die Höhe, in der sich bequem skizzieren lässt. Ab ca. 1,75m (also Augenhöhe) nehmen die gezeichneten Koordinaten stark ab. Dies deckt sich mit dem Anstrengungsgefühl, welches man in dieser Skizzierhöhe leicht erlebt.

Die langgestreckte linke Flanke des Y-Achsen-Histogramm zeigt, dass dieser Raum bis auf den Fussboden der CAVE ausgenutzt wird, obgleich dies auch selten geschieht. Über einer Höhe von 2,25m über dem Boden wird dagegen so gut wie gar nicht skizziert.

Interessant ist die Frequenzverteilung in Z-Richtung. Ihr zufolge ziehen es viele Anwender vor ca. 0,5m vor oder hinter dem Mittelpunkt der CAVE zu zeichnen. Wobei der vordere Teil der CAVE gegenüber dem hinteren Teil im Bereich +0,25m bevorzugt wurde. Nach außen hin aber dann schnell an Attraktivität verliert zugunsten der hinteren Bereiche der CAVE.

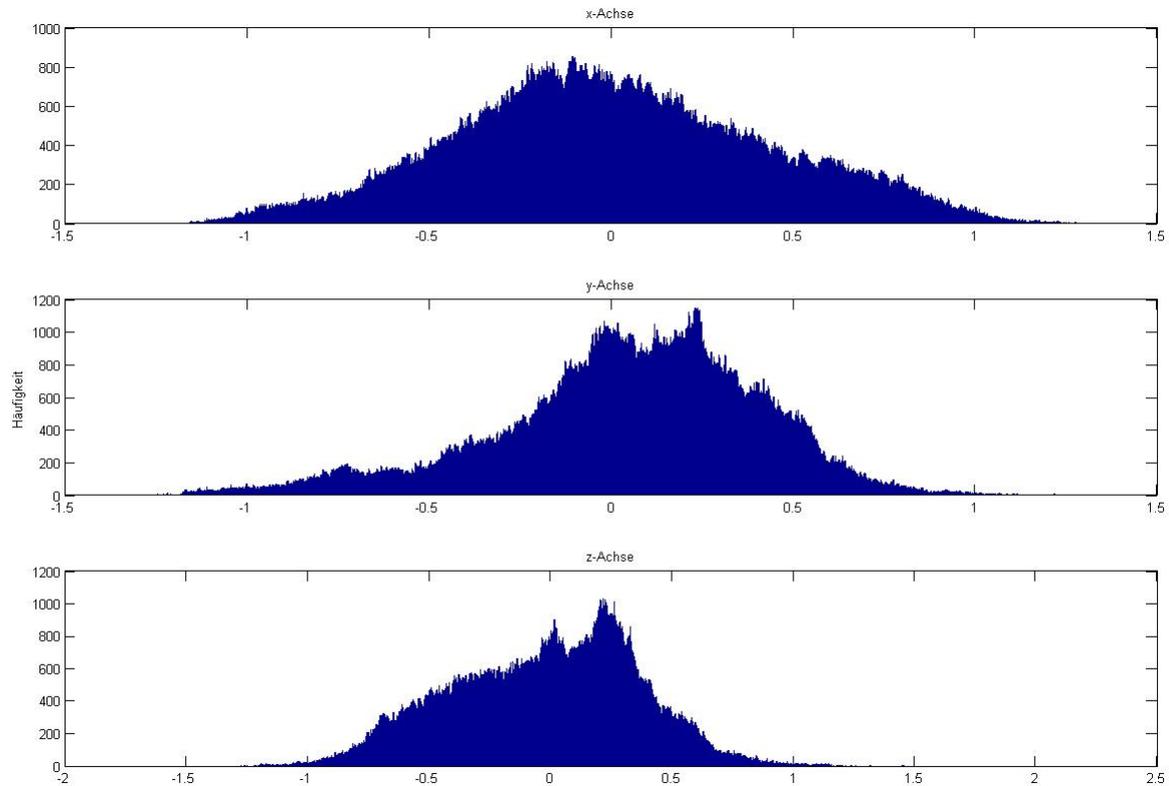


Abbildung 57 Histogramm aller Koordinaten

Weiterhin fiel auf:

- Die Punktwolken der verschiedenen Probanden gleichen innerhalb der Aufgaben einander stark. D.h., Viele Probanden zeichnen bei bestimmten Vorgaben an die gleiche Position.
- Die Ansicht der YZ-Ebene gleicht der YX-Ebene sehr stark. Nur die Draufsicht ist anders. Diese Feststellung mag zunächst trivial sein, aber die Testpersonen hätten das zu zeichnende Objekt genauso gut von oben zeichnen können.
- Es gibt eindeutig präferierte Positionen wenn etwas gezeichnet werden soll. So wurde beispielsweise eine zu entwerfende Bar eher an den Rand der Cave skizziert und ein Stuhl eher in die Mitte der Cave

8 Ergänzung der 3D-Skizzieranwendung durch optisches Tracking

Nachdem in den vorangegangenen Kapiteln Sinn und Zweck, theoretischen Grundlagen sowie eine eingehende Untersuchung des Interaktionsraumes präsentiert wurden, widmet sich dieser Teil der Arbeit der weiteren Analyse und dem Entwurf eines konkreten, optischen Tracking Moduls als Erweiterung der bestehenden SketchApp Anwendung.

Das optische Tracking System ist als sog. *outside-in Tracking* zu realisieren. Das bedeutet, dass stationäre Kameras für die Positionsbestimmung der zu trackenden Gegenstände genutzt werden. Als zu trackende Gegenstände sind die Skizziertools und die Shutterglases zu nennen. Die Shutterglases sind zu tracken, da die Position des Kopfes wichtig für das Rendern der richtigen Position des virtuellen Objektes genutzt wird.

8.1 Bestehende Systeme

Während der Entwicklung des Prototyps war es immer wieder hilfreich die Eigenschaften bestehender optischer Tracking-Systeme heranzuziehen. Somit konnten einerseits erste Abschätzungen über die Machbarkeit und auch Grenzen erkannt werden. Die folgende Tabelle fasst Eigenschaften von Systemen aus ähnlichen Bereichen derer von SketchApp zusammen:

Name	ART Tracking	iotracker	Cyclope
<i>URL d. Herstellers</i>	<i>Ar-tracking.eu</i>	<i>www.iotracker.com</i>	<i>Inrialpes.fr/sed/6doftracker</i>
<i>Verfahren</i>	<i>Stereoskopisch</i>	<i>Stereoskopisch</i>	<i>Monoskopisch</i>
<i>Genauigkeit (Jitter)</i>	<i>0.15mm in der Translation (bei einem Abstand von 2,25m)</i>	<i>Max. 0,1mm in der Translation. 0,02° in der Rotation. Genauigkeit für einen 30m² Interaktionsraum liegt bei 0,5cm</i>	<i>1mm in der Translation 0,3° Rotation (bei einem Abstand von 1,5m)</i>
<i>Arbeitsbereich</i>	<i>< 4m mit passive Markern bis zu 6m, in Abhängigkeit der Markergröße, z.B. 2,5m mit 12mm Marker</i>	<i>0.5 – 5.5 m, je nach Markergröße 4 x 4 x 3 m</i>	<i>Hor. 80° x vert. 60° x 5m</i>
<i>Latenz</i>	<i>< 20 ms</i>	<i>18ms- 40ms</i>	<i>< 30 ms (bei 60 Hz)</i>
<i>IR Quelle</i>	<i>IR LED Flash (8 Helligkeitsstufen)</i>	<i>LED-Flash mit +-60° FOV, Intensität durch Software steuerbar</i>	<i>IR-LED Flash</i>
<i>IR Wellenlänge</i>	<i>850 nm</i>	<i>850nm</i>	
<i>CCD-Chip</i>		<i>1/3" global-shutter CMOS</i>	
<i>Kameratyp</i>		<i>0.36 Megapixel (752x480pixel)</i>	

<i>Objektiv</i>	<i>3.5 mm, 4,5 mm, 6 mm, 8 mm, 12 mm</i>		
<i>FOV</i>	<i>Hor. 72,8°, vert. 58,2° Hor. 57°, vert 45,3° Hor. 42,9°, vert 33,0° Hor. 33,9°, vert 25,8° Hor. 22,2°, vert. 16.8°</i>	<i>90° diagonal, 3.6mm Weitwinkel, f2.0, IR Bandpassfilter</i>	<i>Hor. 60°, vert. 56°</i>
<i>Update Rate (fps)</i>	<i>Max. 60 (adjustable)</i>	<i>Max. 60 Hz</i>	<i>Max. 60 Hz</i>
<i>Max. Targets</i>	<i>20 verschiedene Markermodelle im fast mode und 2 im accurate mode</i>	<i>12</i>	<i>4</i>
<i>Interfaces</i>			<i>VRPN Schnittstelle</i>
<i>Sonstiges</i>		<i>Max. 8 synchronisierte IR Kameras</i>	<i>Kugelgröße der Marker 10-25mm</i>
<i>Preis:</i>	<i>12.400 EUR</i>	<i>11.400 EUR</i>	<i>Kommerziell noch nicht verfügbar</i>

Tabelle 7 Kommerzielle Tracking-Systeme

Die oben vorgestellten Systeme sind nur eine Auswahl der derzeit erhältlichen Systeme, die jedoch besonders im Bereich der VR sehr bekannt sind. Zu bemerken ist, dass bei allen diesen Systemen die mögliche Frame Rate relativ niedrig ist (max. 60 fps). Andere Systeme, wie z.B. das MX System von Vicon (Vicon 2008) übertrifft diese um das mehrfache. Mit Bildraten von 240 bis 2000 Bilder pro Sekunde sind hier noch weit höhere Frequenzen möglich.

Auch gibt es Unterschiede bei der Prozessierung der Daten. Einige Systeme berechnen die Rotation und Translation der Targets „on Chip“, andere liefern das gesamte Bild zunächst an einen Rechner von dem anschließend die Pose (Rotation, Translation) bestimmt wird. Einer der größten Unterschiede zwischen den Systemen stellt die Anzahl der verwendeten Kameras dar. Die meisten kommerziellen Systeme (besonders aus der Medizin) verwenden zwei Kameras (stereoskopisches Tracking), da damit zum einen Positionen genauer ermittelt und Korrespondenzprobleme einfacher gelöst werden können. Dazu später jedoch mehr.

8.2 Hardware

Die Wände der Cave stellen ein besonderes Problem beim Setup eines optischen Tracking Systems dar. Die Kameras müssen einerseits so angeordnet werden, dass sie einen möglichst großen Interaktionsraum bieten (idealerweise 2,5m x 2,5m x 2,5m), andererseits dürfen sie nicht die Sicht auf die Projektionswände behindern. Versuche, die Kameras außerhalb der CAVE zu positionieren und Objekte durch die Projektionswände hindurch zu tracken gibt es zwar (Rakkolainen 2006), erwiesen sich in der Praxis aber als unbrauchbar.

Abbildung 58 zeigt einen anschaulichen Kameraaufbau in der CAVE. Die Kameras werden in den oberen Ecken der Cave angebracht und sind auf den Mittelpunkt ausgerichtet. Ein Überschneiden der Sichtbereiche der verschiedenen Kameras ist besonders erwünscht, da auf diese Weise Verdeckungen der zu trackenden Objekte (Targets)

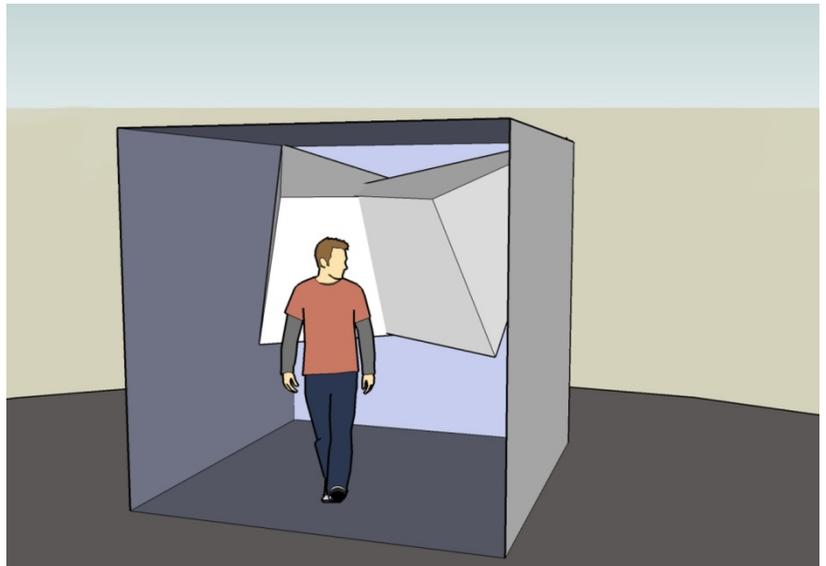


Abbildung 58 Sichtbereich der Kameras innerhalb der CAVE

kompensiert werden können, denn ist ein Target in einem Kamerabild verdeckt so kann eine andere Kamera das Target evtl. doch noch „sehen“. Der resultierende Interaktionsbereich, also der Bereich, in dem es möglich ist, sich frei zu bewegen, ohne das Target zu verlieren, ist dennoch sehr beschränkt, wenn man die Kameras auf diese Weise anordnet. Vor allem der geringe Öffnungswinkel von ca. 42°

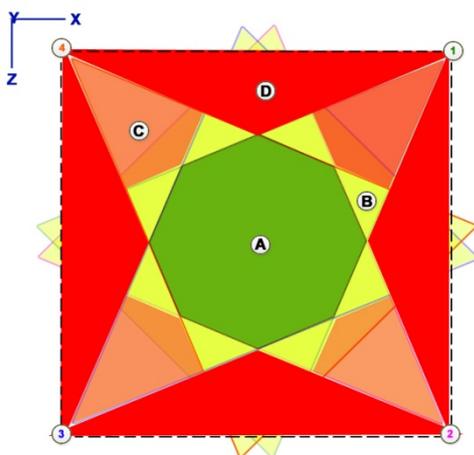


Abbildung 59 Kamerasetup Nr. 1 - Beschränkter Interaktionsraum

(hor.) eines Normalobjektives erlaubt nur einen stark begrenzten Interaktionsraum. Abbildung 59 zeigt einen theoretischen Aufbau von vier Kameras innerhalb der CAVE. Der grüne Bereich (A) kennzeichnet den Raum, der von allen vier Kameras abgedeckt würde und somit ein in der Hand gehaltenes Target höchstwahrscheinlich erkennen könnte. Steht man etwa mit dem Rücken zu Kamera 2, so kann das Target von dieser zwar nicht erkannt werden, dafür aber von Kameras 4 und 1, evtl. auch von 3. Leider ist der Überschneidungsbereich aller vier Kameras relativ klein und nur ca. 15% des Raumes sind auf diese Weise als

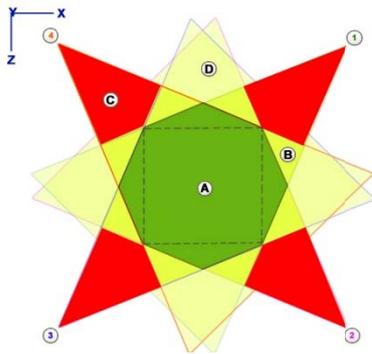


Abbildung 60 Optisches Tracking durch die CAVE Wand

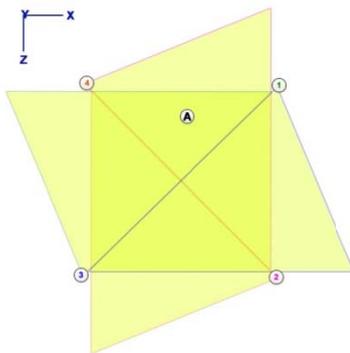


Abbildung 61 Kamerasetup Nr. 2 - Seitlich ausgerichtetes Kamerasetup

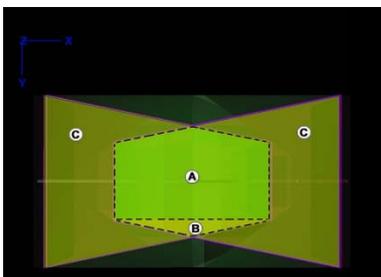


Abbildung 62 Kamerasetup Nr. 1 - Seitenansicht

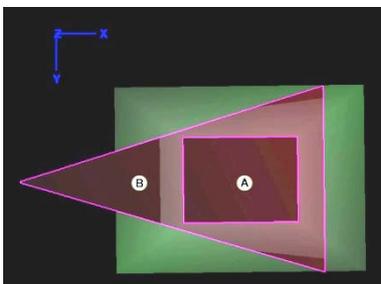


Abbildung 63 Kamerasetup Nr. 2 - Seitenansicht

Interaktionsraum erschließbar. Zwar erweitern die Bereiche B und C diesen möglichen Interaktionsraum, aber können dafür nicht alle Skizziersituationen handhaben. Steht man beispielsweise im Bereich C mit dem Rücken zur Kamera 4, so kann nur noch eine Kamera das Objekt sehen. Ein Tracking des Targets wäre noch möglich, aber sobald die skizzierende Hand oder der Körper des Nutzers das Target selbst verdeckt, geht das Tracking verloren. Die schematischen Darstellungen links zudem das Problem nicht vollständig auf, da es sich um ein dreidimensionales Problem handelt.

Eine weitere Möglichkeit der Kameraanordnung könnte darin bestehen, die Sichtbereiche der Kameras von Ecke zu Ecke der CAVE auszurichten. Auf diese Weise wäre es möglich nahezu die gesamte Cave als Interaktionsraum zu erschließen. Der resultierende Nachteil ist aber, dass wie zuvor im Bereich C, der gesamte Raum nur mit je zwei sich überschneidenden Sichtbereichen der Kameras ausgestattet wäre. Abbildung 61 zeigt ein solches Kamerasetup.

Steht man beispielsweise im Raum A, so kann das Target von Kamera 3 und 4 nicht gesehen werden, da man sich außerhalb deren Sichtbereichs befindet. Lediglich Kamera 1 und 2 sehen evtl. das Target, wenn der Nutzer nicht zur oberen Wand hin schaut, denn dann würde es evtl. von Rücken und skizzierender Hand verdeckt. Die Größe des Interaktionsraumes verhält sich zu den existierenden mehrfach überlappten Sichtbereichen entgegengesetzt.

Die beiden Seitenansichten Abbildung 62 und Abbildung 63 illustrieren das Problem der zwei Kamerasetups von der Seite. Während sich im ersten Fall ein zweckmäßiger Schnittbereich der pyramidenförmigen Sichtbereiche ergibt, entstehen im zweiten Fall einige Probleme, da lediglich der Bereich A als gemeinsame Schnittfläche auszumachen ist. Sie vergrößert sich zwar in zunehmender Z-Richtung (es entsteht eine schiefe Pyramide) ist aber dennoch zu klein. Eine effektive Überlappung von zwei Kameras ist also kaum möglich.

Wie aus diesen Überlegungen hervorgeht, ist es schwierig die gesamte CAVE als Interaktionsraum mittels Kameras vollständig zu erschließen. Mit zusätzlichen Kameras (erheblich mehr als vier) sollte dieses Problem jedoch lösbar sein. Dem gegenüber ist eine möglichst geringe Anzahl eingesetzter Kameras erstrebenswert, um die Projektionen nicht zu stören.

8.2.1 Kamera, Objektiv und IR Setup

Für die optische Tracking Anwendung standen zwei Unibrain Fire-i⁶ und eine Unibrain Fire-i 620c Kamera zur Verfügung. Alle drei Kameras sind mit einem Farb-CCD-Chip ausgestattet und per FireWire-Bus an den PC anschließbar. Die beiden Unibrain Fire-i können als Low-Cost Kamera bezeichnet werden, während die Fire-i 620c eine professionelle Kamera mit höheren Anschaffungskosten ist. Die folgende Tabelle vergleicht die beiden verschiedenen Modelle:

	 Unibrain Fire-i 1.2 40041905	 Unibrain Fire-i 620c 16389
<i>Sensortyp</i>	<i>Sony Wfine ¼" color CCD, progressive scan</i>	<i>Sony ICX-204AK 1/3" color CCD, progressive scan</i>
<i>Max. Auflösung</i>	<i>VGA 640 x 480</i>	<i>SVGA 1024 x 768</i>
<i>Optik</i>	<i>4.3mm Brennweite, 42° hor., 32° vert. Öffnungswinkel (Normalobjektiv)</i>	<i>Pentax 4.2mm Brennweit, Weitwinkel (separat erworben)</i>
<i>Video-Mode</i>	<i>YUV (4:1:1, 4:2:2, 4:4:4), RGB-24bit, Monochrome-8bit</i>	<i>YUV (4:2:2), 8bit or 12bit color RAW</i>
<i>Bildwiederholrate</i>	<i>30, 15, 7.5, 3.75 fps</i>	<i>36 (Format-7 mode 0), 30, 15, 7.5, 3.75, 1.875 fps</i>
<i>Shutter</i>	<i>1/3400s - 1/31s</i>	<i>1 μs - 65s</i>

Tabelle 8 Vergleich der verwendeten Kameras

Die wichtigste Eigenschaft der Kamera beim Tracking ist eine hohe Framerate, gefolgt von der maximalen Auflösung und Video Modus. Die höchstmögliche Framerate bei der Fire-i Kamera ist 30fps und kann bei YUV 4:1:1 in 640x480px erreicht werden. Die Fire-i 620c Kamera kann diese Bildrate im Y-Monochrom Modus mit einer Auflösung von 1024x768px erreichen. Im Format-7, Mode-0 Modus, der die Definition einer Region of Interest (ROI) erlaubt, sind sogar 36fps möglich. Bei Tests der verschiedenen Video-Modes konnte selbst diese Bildrate nochmals auf ca. 46fps erhöht werden, wenn nicht alle horizontalen Linien ausgelesen werden.

⁶ Unibrain Part No. 2036, www.unibrain.com

Wie aus dem vorangegangenen Kapitel hervor geht, sind für SketchApp mehrere Kameras notwendig um einen angemessenen großen Interaktionsraum zu erhalten. Dabei kann der FireWire-Bus unter Umständen ein Problem darstellen. Während die Auslastung des FireWire-Bus mit zwei Fire-i im Dolly-Chain-Modus (mehrere Kameras an einem Bus) im YUV 4:1:1 Modus bei einer Auflösung von 640x480px und 30fps bei 79% liegt, lastet die Fire-i 620c einen Standard 400MBit/s IEEE-1394-Bus bereits bei 30fps im Y800 (Monochrom-Modus) mit einer Auflösung von 1024x768px mit 62% aus. Handelsübliche IEEE 1394 Hostcontroller können somit Daten von sechs der Low-Cost Fire-i Kameras aufnehmen oder Daten von drei Fire-i 620c Kameras. Die Anzahl der Kameras kann verdoppelt werden, wenn ein 800MBit/s FireWire Hostcontroller verwendet wird. Da die meisten FireWire Karten lediglich drei Anschlüsse besitzen, sind dann aber weitere zwischen geschaltete Hubs nötig, um weitere Kameras aufzunehmen. In Abhängigkeit vom eingesetzten PCI-Bus können sogar bis zu fünf Fire-i 620c (bei PCI-Spezifikation 2.1 und einer Bandbreite von 32Bit) oder zehn Fire-i 620c (bei einer Bandbreite von 64Bit und einem Frontside-Bus von 66MHz) angeschlossen werden. Somit ist eine Abdeckung des geforderten Interaktionsraumes mit handelsüblicher, moderner Hardware möglich.

Wie sich im Verlauf der Entwicklung herausgestellt hat, ist es sinnvoll beim Kauf einer Kamera auf die Empfindlichkeit des CCD-Chips in den verschiedenen Farbspektren zu achten. Viele CCD-Chips können langwelliges Infrarot-Licht sensieren. Dies kann dazu ausgenutzt werden, um beispielsweise auf analoge bzw. technische Weise gute Blobs generieren zu können. Die in Kapitel 8.2.2 beschriebenen retroreflektiven Multipunktmarker nutzen diesen Effekt.

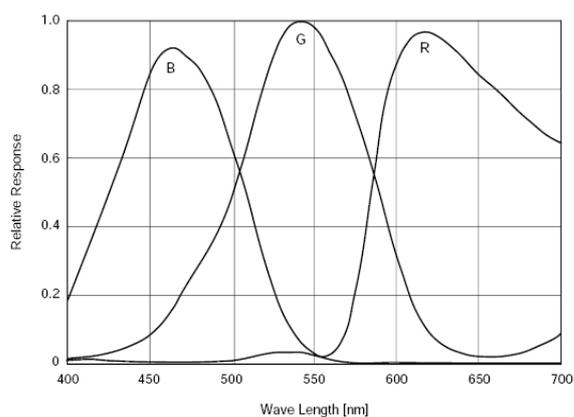


Abbildung 64 Relative Empfindlichkeit des Sony ICX-204AK

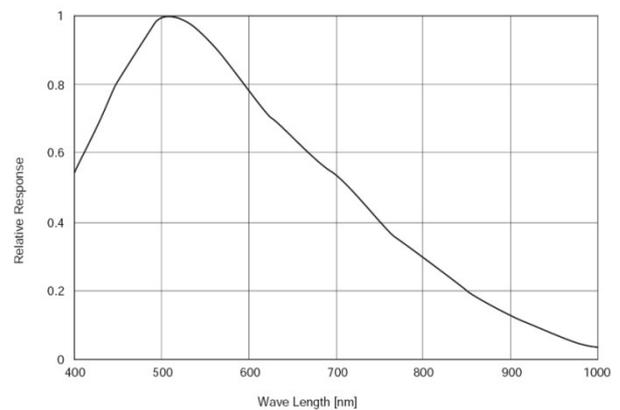


Abbildung 65 Relative Empfindlichkeit des Sony ICX-204AL

Die obigen Graphen zeigen den Verlauf der relativen Empfindlichkeit bzw. das spektrale Verhalten der in den Fire-i eingesetzten CCD-Chips. Im Gegensatz zum menschlichen Auge sind diese meist auch sensitiv für das nahe IR-Spektrum (NIR). Einige Kameras bzw. Objektive sind mit einem IR-Sperrfilter

ausgerüstet, um einen korrekten Weißabgleich durchführen zu können. Die verwendeten Fire-I Low-Cost Kameras werden in der Standardversion ohne IR-Sperrfilter ausgeliefert, was dem Einsatz von IR ermöglicht. Bei einigen anderen Low-Cost Kameras lassen sich diese Filter, die im Normalfall auf die zum CCD-Chip gerichtete Seite des Objektivs aufgebracht sind, durch vorsichtiges kratzen entfernen. Häufig kann ein Objektiv einer Webcam (Low-Cost Kamera) auch gegen ein anderes Standard Miniobjektiv ohne IR-Sperrfilter ausgetauscht werden. Bringt man nun zusätzlich einen Visible-Light-Blocking Filter vor der Kamera an, so können die mit IR-Licht bestrahlten Markerkugeln sehr gut im Kamerabild erkannt werden. Abbildung 66 zeigt verschiedene käuflich erhältliche Visible-Light-Sperrfilter.

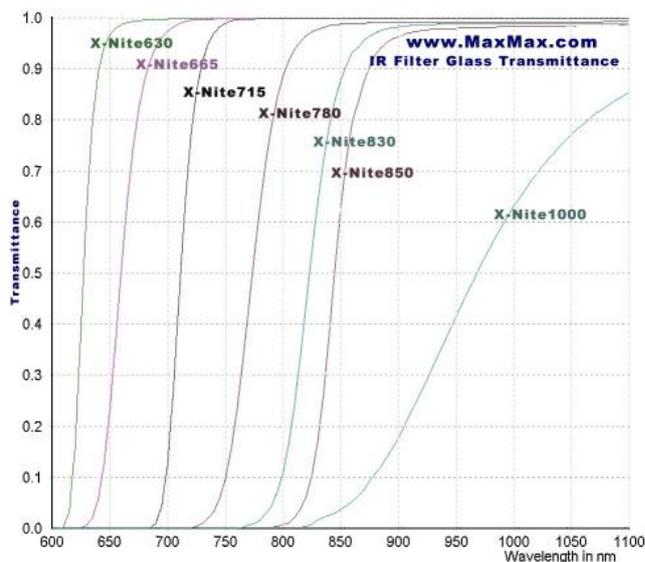


Abbildung 66 Frequenzgang verschiedener Visible Light Filter

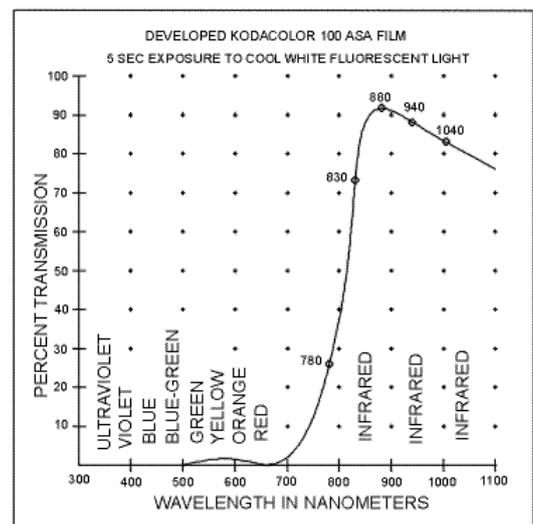


Abbildung 67 Low-Cost Visible Light Filter

IR-Filter lassen sich auf einfache Weise auch selber herstellen. Dazu wird ein Farbfotofilm benötigt, wie z.B. einen 100 ASA Film von Kodak, und belichtet ihn ca. 5 Sekunden mit fluoreszierendem Licht, wie sie Neonröhren erzeugen. Danach muss er entwickelt werden und es resultiert ein Filter mit dem Transmissionsbereich wie er in Abbildung 67 dargestellt ist. Ein solcher Filter wird für das derzeitige System erfolgreich eingesetzt. Bei vollem Sonnenschein kann es vorkommen, dass noch weiterhin sichtbares Licht durch dringt, dies kann aber durch eine doppelte Schicht des Filters verhindert werden, ohne wesentliche Verluste der Helligkeit des IR-Bereiches hinnehmen zu müssen.

Neben der Wahl des geeigneten CCD-Chips ist es ebenso wichtig, ein passendes Objektiv für die Kamera zu finden. Für SketchApp ist ein möglichst großer Öffnungswinkel die wichtigste Eigenschaft, um einen möglichst großen Interaktionsbereich zu erhalten. Folgende Tabelle zeigt beispielhaft verschiedene

Konfigurationen von Objektgröße, Arbeitsbereich, CCD-Chip-Größe und Brennweite für verschiedene Anwendungen.

	<i>Strichcode Erkennung</i>	<i>Sortieren von Obst</i>	<i>Augenkontrolle</i>	<i>Nummernschild Erkennung</i>
<i>Object size</i>	<i>30 mm</i>	<i>200 mm</i>	<i>50 mm</i>	<i>600 mm</i>
<i>Working distance</i>	<i>300 mm</i>	<i>1000 mm</i>	<i>350 mm</i>	<i>10000 mm</i>
<i>CCD format</i>	<i>1/4" (w=3,2 mm)</i>	<i>1/3" (w=3,6 mm)</i>	<i>1/2" (w=6,4 mm)</i>	<i>1/4" (w=3,2 mm)</i>
<i>Focal length of width</i>	<i>28,9 mm</i>	<i>17,7 mm</i>	<i>39,7 mm</i>	<i>53 mm</i>

Tabelle 9 Beispiele für verschiedene Anwendungsgebiete (ImagingSource 2006)

Es lassen sich drei Regeln für die Auswahl des richtigen Objektivs formulieren:

- Das Format des Objektivs sollte vorzugsweise größer als das CCD-Format sein, da Fehler eines Objektivs besonders am Rand dessen auftreten.
- Die zu wählende Brennweite hängt ausschließlich vom Format (Größe) des CCD-Chips, der Arbeitsdistanz und der Objektgröße ab.
- Extensionsringe zwischen Objektiv und Kamera vermindern die minimale Arbeitsdistanz.

Für die optische Tracking Anwendung, lässt sich die zu verwendende Brennweite wie folgt bestimmen:

$$\text{Brennweite für die Breite} = \frac{\text{Working distance} \times \text{CCD width}}{\text{Object width} + \text{CCD width}}$$

$$\text{Brennweite für die Höhe} = \frac{\text{Working distance} \times \text{CCD height}}{\text{Object height} + \text{CCD height}}$$

Bezogen auf die Mitte der CAVE, lässt sich die ungefähre „working distance“ mit 1768mm angeben. Als Objektgröße ist hier die gewünschte Höhe bzw. Breite des Interaktionsraumes heranzuziehen, welcher mit 2000mm bemessen wird. Das CCD-Format der Fire-i 620c ist 1/3“ und hat eine Höhe von 3,6mm und eine Breite von 4,8mm. Somit lassen sich die Brennweiten wie folgt berechnen:

$$\text{Brennweite für die Breite} = \frac{1768 \times 4,8}{2000 + 4,8} = 4,233\text{mm}$$

$$\text{Brennweite für die Höhe} = \frac{1768 \times 3,6}{2000 + 3,6} = 3,177\text{mm}$$

In der Praxis sind Objektive unter 4mm Brennweite für C-Mount Kameras wie die Unibrain 620c, schwer zu bekommen. Üblicher sind bei Objektiven mit geringer Brennweite CS-Mount Aufnahmen. Bei diesen besteht zudem die Möglichkeit auch C-Mount Objektive einzusetzen, was andersherum nicht möglich ist. Letztendlich fiel die Wahl des Objektivs auf das Pentax H416(KP), ein manuell geregeltes Objektiv im ½" Format, einer Brennweite von 4,2mm und einem Blendenbereich von 1,6mm – C. Der horizontale Bildwinkel wird mit 86°46' und die minimale Objektdistanz mit 0,2m angegeben.

Nach einigen Tests mit dieser Kamera-Objektiv Kombination, ist folgendes festzuhalten:

- Die Fire-i 620c hat einen eingebauten IR-Blocking Filter, der mittels kleiner Schraubchen vor dem CCD-Chip fixiert ist. Er sollte entfernt werden, wenn man im IR-Bereich operiert.
- Beim Scharfstellen des Objektiv, sollte der IR-Filter installiert sein und die IR Beleuchtung angeschaltet, da die meisten Objektive unterschiedliche Brennweiten im IR-Bereich, als im Bereich des sichtbaren Lichtes haben.
- Der IR-Emitter sollte nahe der optischen Achse der Kamera aufgestellt sein, damit retroreflektive Marker (siehe nächster Abschnitt) das Licht geeignet reflektieren können. Idealerweise sind die einzelnen LEDs des IR-Emitter um das Objektiv angeordnet.

8.2.2 Der Marker (Target, Fiducial)

Während aktuelle Forschungen des optischen Trackings zunehmend in Richtung markerloses Tracking mit monookularen Kamerasystemen gehen (Andrew I. Comport 2006) und dabei auf Verfahren mittels „Optical Flow“ (Neumann, You et al. 1999) oder „Scale Invariant Feature Points“ (SIFT) (Lowe 2004) zurückgreifen, ist die einfachste Art Position und Rotation (Pose) eines Objektes heraus zu finden, das Befestigen eines Markers an eben diesem Objekt. Marker (auch engl. Fiducials, Targets oder auch Rigid-Marker genannt) sind so designed, dass diese mittels ad hoc Methoden leicht identifiziert werden können. Ihre Position im Kamerabild kann mit viel höherer Genauigkeit gemessen werden, als natürliche

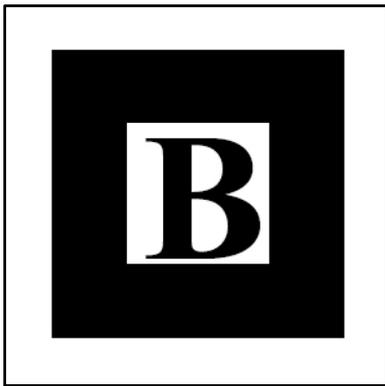


Abbildung 68 2D-Marker aus ARToolkit

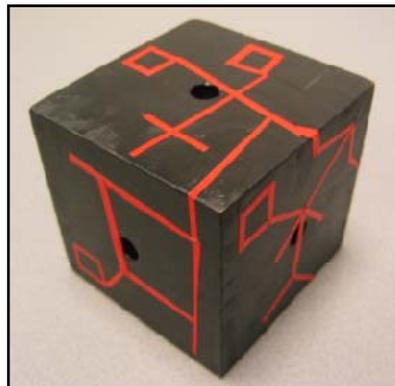


Abbildung 69 Line-Pattern Marker von Smit (Smit, Rhijn et al. 2006)

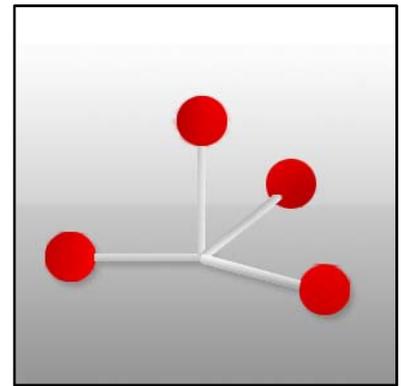


Abbildung 70 3D Multipunktmarker

aus dem Bild (z.B. mittels Corner-Detection) extrahierte Feature-Points. Die Abbildungen oben zeigen einige verschiedene Marker. Planare Marker wie solche in Abbildung 68 und Abbildung 69 haben den Vorteil, dass sie trotz perspektivischer Projektion leicht voneinander unterschieden werden können. Zur Identifikation können z.B. aufgedruckte Buchstaben oder Grafiken verwendet werden. Dabei besitzt die Grafik für die Erkennung der Pose in Raum keine Bedeutung. Bei ARToolkit geschieht dies über den schwarzen Rand des Markers. Durch ihn können die Eckpunkte mittels Computer Vision Algorithmen leicht bestimmt und daraus Zuhilfenahme der Oberflächen-Normalen die 3D-Position bestimmt werden. Ein ähnliches Verfahren wird auch von PTrack erfolgreich genutzt (Pedro Santos 2006). Der Nachteil dieser Methode besteht in der eingeschränkten Bewegungsfreiheit bzw. dem geringen Erkennungswinkel. So kann beispielsweise in senkrechter Lage zur XY-Fläche des Kamerakoordinatensystems nichts mehr erkannt werden und eine Rekonstruktion der Pose ist unmöglich. Smit versucht dieses Problem durch eine Kombination von sechs planaren Markern als Würfel zu lösen (Smit, Rhijn et al. 2006). Laut Smit funktioniert diese Lösung auch ganz gut. Vor allem für AR-Anwendungen, bei denen man ein virtuelles Objekt von all seinen Seiten begutachten will, bietet sich diese Methode an. Wie auch bei AR-Toolkit werden die verschiedenen Seiten des Würfels durch

spezielle Markierungen identifiziert. Leider bietet diese Art der Marker, wie auch AR-Toolkit in Bezug auf das zu realisierende optische Tracking für SketchApp keine sinnvolle Verwendungsmöglichkeit, da durch die Größe des Markers eine Befestigung am Stift den Benutzer beim zeichnen erheblich stören würden.

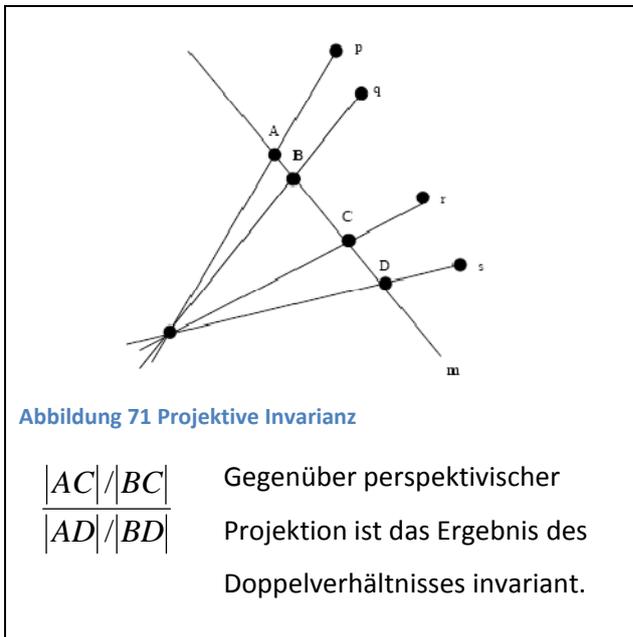


Tabelle 10 projektive Invarianz nach (Liere & Mulder 2003)

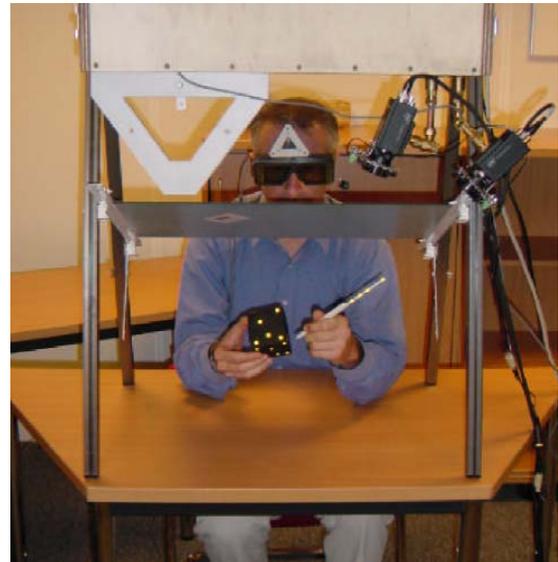


Abbildung 72 Tracking mit planaren und kollinearen Marker Pattern

Eine elegante Lösung, um Stifte zu tracken und zu identifizieren, wird von Liere in (Liere & Mulder 2003) beschrieben. Er nutzt retro-reflektives Klebeband und markiert damit das obere Ende eines Stiftes mit vier Ringen um den Stift herum. Der Vorteil hierbei ist, dass das Target je nach Position der Ringe eindeutig identifiziert werden kann, da das Doppelverhältnis von vier zueinander kollinearen Punkten unter perspektivischer Projektion invariant bleibt (auch beschrieben in (Fernandez, Papos et al. 2007)). Nachteil bei dieser Lösung sind die eingeschränkten Freiheitsgrade. Da eine Rotation um den Stift nicht detektiert werden kann, hat man nur 5DOF anstatt der geforderten 6DOF. Somit ist leider auch diese Lösungsmöglichkeit für SketchApp nicht anwendbar.

In medizinischen AR-Anwendungen werden als Marker seit Jahren sphärische, retroreflektive Marker eingesetzt. Es hat sich erwiesen, dass diese Art von Marker-Kugeln sehr gut funktioniert, da ihr rundes Äußeres gegen perspektivische Verzerrung relativ invariant ist. Außerdem bietet ihr Schwerpunkt eine stabile 2D Position, welche mit Subpixelgenauigkeit ermittelt werden kann. Einzelne dieser Kugeln können sog. Point-Models oder Multipunktmodellen (siehe Abbildung 70) zusammengefasst werden. Diese Multipunktmodelle können aktiver oder auch passiver Natur sein. Jeder der beiden Kategorien hat Vor- und Nachteile. Aktive Marker können etwa aus LEDs bestehen, die über eine Batterie zum Leuchten gebracht werden. Ihr Vorteil ist eine bessere Helligkeit gegenüber passiven Markern und die Unempfindlichkeit gegen sich verändernde Lichtsituationen. Allerdings benötigen sie eine Stromquelle, welche das Objekt, an dem der Marker angebracht ist entweder durch das Gewicht der Batterie erschwert oder die Verwendung eines zusätzlichen Stromkabels nötig macht. Beides ist für professionelle Anwendungen meist unakzeptabel, weshalb in den meisten Fällen passive Marker verwendet werden.

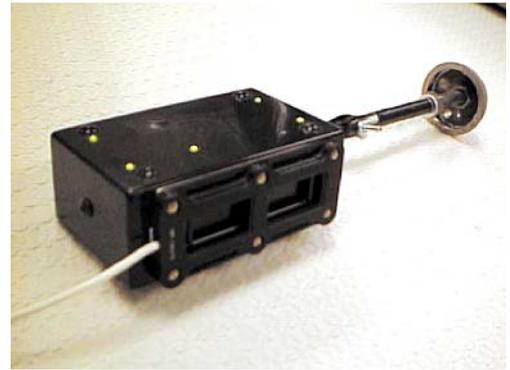


Abbildung 73 Planarer aktiver Multipunktmarker zu (William Hoff 2000)

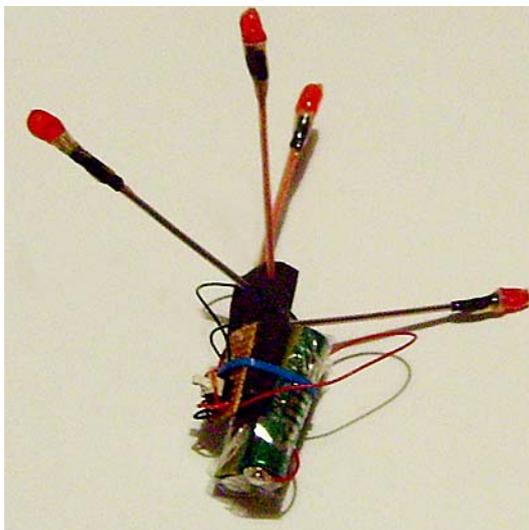


Abbildung 74 Selbstgebaute aktiver 4-Punkt-Marker

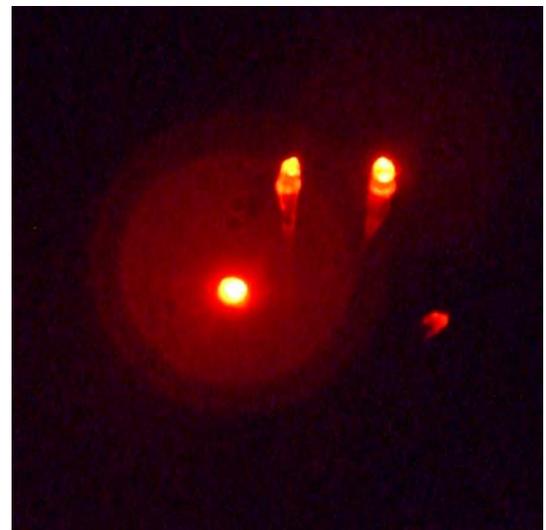


Abbildung 75 Aktiver 4-Punkt-Marker mit Lenseflare-Effekt

Bei dem Versuch einen eigenen aktiven Marker herzustellen, sind weitere Probleme hervorgetreten. Es hat sich herausgestellt, dass der Abstrahlwinkel der LEDs ein besonderes Problem ist. Die meisten handelsüblichen LEDs haben einen Abstrahlwinkel von 16° bis 45° . Es lässt sich mit diesen also nur

schwer ein homogenes, punktförmiges Leuchten in alle Richtungen realisieren. Versuche mit einer Heißkleberschicht um die LEDs herum brachte zwar ein wenig Verbesserung (die LED leuchtete schwach an der Seite) aber die Lichtstärke nahm enorm ab und von hinten leuchtete die LED immer noch nicht. Recherchen förderten eine weitere Verbesserungsmöglichkeit zu Tage. Für LEDs sind z.B. sog. LED-Refraktoren als Aufsatz erhältlich. Dieser ermöglicht einen Abstrahlwinkel von 180° . So könnte mit 2 LEDs einen Abstrahlwinkel von 360° zu erreichen sein.



Abbildung 76 LED Refraktor

Ein weiteres Problem aktiver Marker besteht darin, dass bei direktem Lichteinfall in das Kameraobjektiv zusätzliche Artefakte in Form von Lensflare-Effekten entstehen, die das Tracking negativ beeinflussen können (vgl. Abbildung 75). Auch das Beleuchten von Gegenständen durch den Marker führt zu Problemen. So kann es etwa passieren, dass falsche Blobs aufgrund von Reflektionen von Objekten in der Umgebung erzeugt werden.

Die Vorteile der aktiven Multipunktmodelle liegen darin, auf technische Weise eine recht einfache Art der Identifikation (vgl. Kapitel 9.6) von einzelnen Blobs zu ermöglichen. Die Identifikation von Blobs wird vor allem beim Lösen des 3D-2D Korrespondenzproblems und bei der Identifikation von verschiedenen Multipunktmodellen benötigt. Um ein Blob zu identifizieren, können bei aktiven Multipunktmodellen die einzelnen LEDs unterschiedliche Farbe haben (Hoffmann, Hahne et al. 2005) oder in unterschiedlicher Weise gepulst werden. So kann im Kamerabild zuverlässig die jeweilige LED einem Blob zugeordnet werden. Die 3D-2D Korrespondenz wäre somit für einzelne Marker oder Multipunktmarker gegeben.

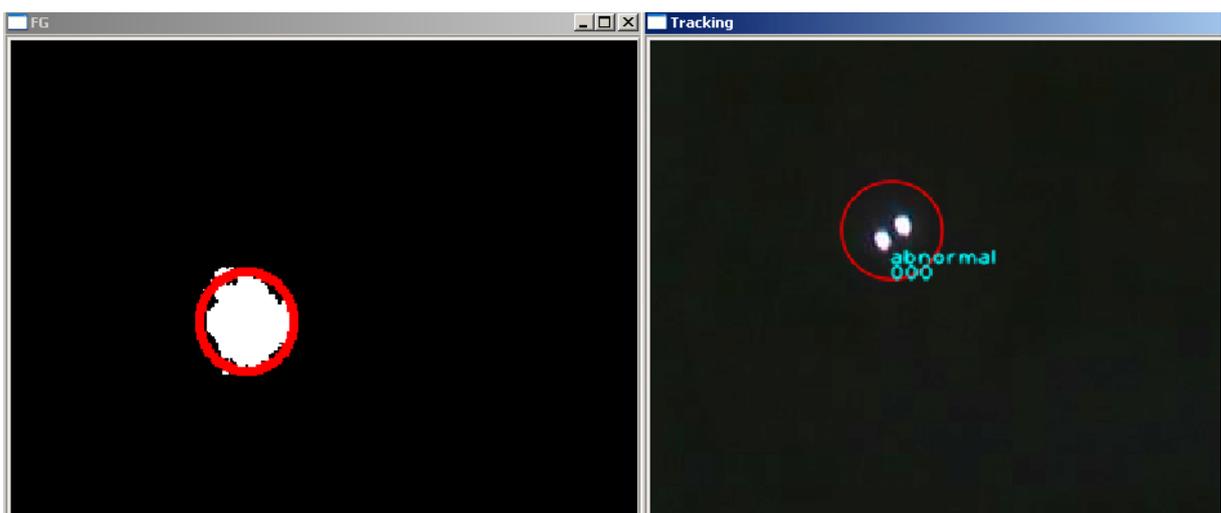


Abbildung 77 Probleme beim Blob-Tracking durch direkten Lichteinfall

Die obige Abbildung zeigt ein typisches Problem bei aktiven Markern: Zwei LEDs leuchten direkt in die Kamera. Da der Threshold für die Binärbildherstellung nicht adaptiv ist, wird aus zwei sehr hellen LEDs ein großer Blob. Stellt man den Threshold niedriger ein, so kann diese Situation zwar vermieden werden, aber die LEDs werden schlechter von der Seite erkannt.

Da die negativen Auswirkungen eines aktiven Multipunktmodells für das SketchApp System überwogen, wurde sich für die Verwendung eines passiven Multipunktmodells entschieden.

Auch bei der Konstruktion von passiven Multipunktmodellen können verschiedene Wege eingeschlagen werden. So sind Farbe, Größe, Anzahl und Position der Markerkugeln mit Bedacht zu wählen. Eine erste Version für SketchApp bestand aus vier roten Kugeln mit einem Durchmesser von $d = 2\text{cm}$, die per Color-Tracking verfolgt wurden. Veränderte Lichtsituationen bedingten allerdings bei dieser Methode eine häufige Neujustierung, da sich die Farbe der roten Kugeln in Abhängigkeit der Beleuchtung sehr stark ändern. Ein Testvideo das in der CAVE erstellt wurde, erwies, dass der vermeintlich für hell befundener Ort gar nicht hell ist und aus diesem Grund das Color-Tracking der Kugeln unmöglich wird. Eine Verbesserung dieser Methode wurde durch Einsatz von neon-farbenen Kugeln erreicht. Im Gegensatz zu üblichen roten Farbe wandeln Neonfarben (auch Tagesleuchtfarben genannt) weniger hell erscheinende oder unsichtbare kurzwellige Spektralanteile des Tageslicht (blau, violett, ultraviolett) durch Fluoreszenz in heller erscheinendes, langwelligeres sichtbares Licht um.

Durch Einsatz von neon-roten Kugeln konnte die Zuverlässigkeit des Color-Tracking zwar gesteigert werden, allerdings blieb ein weiteres Problem, welches auch die roten Kugeln aufwies, bestehen. Da die Oberfläche der Farbe einfallende Lichtstrahlen nicht diffus reflektieren, entstehen (über die Kugel verteilt) eine Farbverlauf und kein Kreis mit einheitlicher Farbe auf dem Kamerabild. Bei ungünstigen Lichtsituationen, z.B. einer Deckenbeleuchtung, verursacht dies, dass der Blob auf dem Kamerabild oben die Farbe Weiß hat und nach unten hin immer röter wird. Eine genaue Mittelpunktbestimmung des Blobs wird somit nahezu unmöglich, da selten ein runder Blob erkannt wird (eher ein Halbmond) und der so errechnete Mittelpunkt abhängig von der Lichtsituation wird. Um dieses Problem zu vermeiden, ist es ratsam Retroreflexfarben zu nutzen. Diese besitzen die Fähigkeit Licht gerichtet zur Lichtquelle zurück zu strahlen, was durch Beigabe von feinen transparenten Kügelchen erreicht wird. Da eine solche Farbe für das Multipunkt-Prototyp-Modell nicht verfügbar war, wurden die einzelnen Kugeln mit Retroreflexgewebe beklebt. Das Resultat war unerwartet gut und bietet die Möglichkeit sehr gut erkennbare Blobs zu generieren.

Neben der Farbe der Marker spielt die Größe der Markerkugeln eine Wichtige Rolle. Laut (Lepetit & Fua 2005) sollten die Marker auf der Bildebene mindestens 4-5pixel groß sein und gegenüber dem Hintergrund drei Mal so groß sein. Zudem sollten digitale Filter (z.B. Scharfzeichnungsfilter) zur

Verbesserung des Bildes vermieden werden. Es ist sogar ratsam, die Linse der Kamera auf einen sog. *defocus* von 0.5pixel einzustellen. Laut Lepetit können so 1/10 Pixelgenauigkeit leicht erreicht werden.

Eine Berechnung der theoretischen Pixelgröße (m) ist über die Kameraauflösung $[R_x; R_y]$, die Brennweite (F) und den Abstand (D) möglich. Unter der Voraussetzung eines mathematischen Lochkameramodells ergibt sich hierfür folgende Gleichung:

$$m_x = \frac{M_d \cdot R_x}{\tan \frac{\alpha}{2} \cdot D \cdot 2}; \text{ bzw. } m_y = \frac{M_d \cdot R_y}{\tan \frac{\beta}{2} \cdot D \cdot 2}$$

Folgende Tabelle zeigt einige theoretische Pixelgrößen der Markerkugeln in horizontaler Richtung bei verschiedenen Brennweiten mit verschiedenen Abständen und Kugelgrößen:

		F= 4,3 (42.25° hor. 32° vert.)				F=3,5 (72.8° hor. 58.1° vert.)				F=2,1(80.95° hor.)			
Auflösung horizontal 640px													
$\frac{M[mm]}{D[mm]}$	0,5	1,0	1,5	2,0	0,5	1,0	1,5	2,0	0,5	1,0	1,5	2,0	
500	9px	17px	25px	34px	5px	9px	13px	18px	4px	8px	12px	15px	
1000	5px	9px	13px	17px	3px	4px	7px	9px	2px	4px	6px	8px	
1500	3px	6px	9px	11px	2px	3px	5px	6px	2px	3px	4px	5px	
2000	2px	5px	7px	9px	1px	3px	4px	5px	1px	2px	3px	4px	
2500	2px	4px	5px	7px	1px	2px	3px	4px	1px	2px	3px	3px	
Auflösung horizontal 800px													
500	11px	21px	31px	42px	6px	11px	17px	22px	5px	10px	14px	19px	
1000	6px	11px	16px	21px	3px	6px	9px	11px	3px	5px	7px	10px	
1500	4px	7px	11px	14px	2px	4px	6px	8px	2px	4px	5px	7px	
2000	3px	6px	8px	11px	2px	3px	4px	6px	1px	4px	4px	5px	
2500	2px	4px	7px	9px	1px	3px	4px	5px	1px	3px	3px	5px	
Auflösung horizontale 1024px													
500	14px	27px	40px	53px	7px	14px	21px	28px	6px	12px	18px	24px	
1000	7px	14px	20px	27px	4px	7px	11px	14px	3px	6px	9px	12px	
1500	5px	9px	14px	18px	3px	5px	7px	10px	2px	4px	6px	8px	
2000	4px	7px	10px	14px	2px	4px	6px	7px	2px	3px	5px	6px	
2500	3px	6px	8px	11px	2px	3px	5px	6px	2px	3px	4px	5px	

Tabelle 11 Theoretische Blobgrößen⁷

⁷ Pixelwerte sind mit einer Nachkommastelle auf die nächsthöhere ganze Zahl gerundet.

Aus der obigen Tabelle lässt sich für die CAVE-Situation in etwa ablesen, welche Kugelgrößen sinnvoll eingesetzt werden können. Alle Konfigurationen bei denen Blobs mit einer Größe von weniger als 4-5px entstehen, sind nicht zu empfehlen (rot markiert), da sie nur schwer vom Rauschen der Kamera getrennt werden können. Je niedriger die Brennweite des Kameraobjektivs ist, desto kleiner fallen die Blobs aus. Der Umkehrschluss, dass große Blobs viel besser sind als kleine, stimmt nur bedingt, da große Blobs bzw. große Markerkugeln sich gegenseitig häufiger überdecken, was wiederum zu einem Problem beim Tracking der Blobs führt. Es gilt also das geeignete Maß zwischen zu großen und zu kleinen Blobs in Abhängigkeit der Hardware zu finden. Oft kann dies nur durch Erfahrung und Ausprobieren geschehen. Einige wenige Anstrengungen auf dem Gebiet der Image Registration versuchen durch synthetische Tests die optimale Anzahl, Position und Größe zu bestimmen. Versuche von Vogt et al. (Sebastian Vogt 2002) haben ergeben, dass die Erhöhung der Markeranzahl keine wesentlichen Verbesserungen mit sich bringen. Die Position dagegen kann bei ungünstig gewählter Lage Ungenauigkeiten verursachen. Dennoch sind diese im Rahmen dieser Arbeit vernachlässigbar, da die die Ungenauigkeit in Translation unter $\pm 0,1$ mm und in Rotation $\pm 0,03^\circ$ liegen, also weit unter den gestellten Anforderungen liegen. In wie weit sich die Verzerrung bei hohen Linsenkrümmungen (kleinen Brennweiten), also Weitwinkel- oder Fischaugenobjektiv auswirkt wurde hier nicht berücksichtigt. Es kann aber davon ausgegangen werden, dass durch die softwareseitige Korrektur eines stark gekrümmten Kamerabildes gerade die Pixel am Rand des Bildes für falsche Erkennung der Blobposition anfällig sind.

Aus Tabelle 11 würde sich bei einer Brennweite von 4,2mm und einer durchschnittlichen Distanz von $D=1,5$ m bei einer Auflösung von 640x480 von 1-2cm ergeben. Sollten allerdings nicht die Fire-I Low-Cost Kameras, sondern die Fire-I 620c, so kann die Markergröße noch weiter verringert werden, da eine höhere Auflösung zum Einsatz kommt (1024x768). Bei den Test-Implementierungen haben sich Marker von 1cm bis zu einer Distanz von 2,3m als praktikabel erwiesen.

Die zu verwendende Anzahl der Markerkugeln ist eine weitere Variable bei dem Design der Multipunktmarker. Sie hängt vom Anwendungszweck und der Anzahl der zur Verfügung stehenden Kameras bzw. des zu verwendenden Trackingalgorithmus ab. So reicht beispielsweise bereits eine Markerkugel beim Einsatz von zwei Kameras, um X, Y und Z-Koordinaten (3DOF) berechnen zu können, während mit einer Kamera nur X und Y-Koordinaten (2DOF) berechnet werden können. Um die geforderten 6DOF mittels zwei Kameras zu erlangen, sind mindestens drei nicht-linear angeordnete Markerkugeln notwendig. Um 6DOF mit einer Kamera zu berechnen sind mindestens vier Markerkugeln notwendig, wobei die vierte Kugel nicht coplanar zu den anderen drei sein darf bzw. alle Kugeln voneinander linear unabhängig sind. Gründe hierfür liefert die Natur der projektiven Geometrie.

Ob Vektoren linear unabhängig sind, kann man mit leicht überprüfen:

Falls die einzigen Skalare die die Gleichung x erfüllen $v_0 = v_1 = \dots = v_{n-1} = 0$ sind, dann sind die Vektoren u_0, \dots, u_{n-1} linear unabhängig.

$$v_0 u_0 + \dots + v_{n-1} u_{n-1} = 0$$

Formel 1 lineare Unabhängigkeit

Beispiel: $u_0 = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$ und $u_1 = \begin{bmatrix} -8 \\ -6 \end{bmatrix}$ sind nicht linear unabhängig, da $v_0 = 2$ und $v_1 = -1$ die Gleichung oben erfüllen.

Die optimale Anordnung für die Kugeln eines Multipunktmarkers kann nur schwer ermittelt werden, z.B. durch synthetische Test, in denen mögliche Kugelpositionen einfach ausprobiert werden (Steinicke, Jansen et al. 2006).

Erfahrungen der TU-Graz bezüglich der Marker (ähnliches Anwendungsszenario) sind gute Ergebnisse bei der Verwendung von Holzkugeln im Durchmesser von 2cm, beklebt mit M3 Signal Dekorband, verbunden mit 2mm Fiberglasstäbe von der Länge 5-15cm, zu erreichen. Dabei ist es ratsam die einzelnen Markerkugeln mindestens 5cm voneinander zu platzieren. Im Eigenbau haben sich auch Korkkugeln als praktikabel erwiesen, da sie leichter als Holz sind und somit die haptischen Eigenschaften des Stiftes weniger beeinflussen.

9 Allgemeine Schritte des Trackings

Um ein den Anforderungen aus Kapitel 5 gerecht zu werden, wurden während der Entwurfsphase verschiedene Möglichkeiten des Trackings erprobt. Testimplementierungen trugen vor allem dazu bei, das Gesamtproblem zu überblicken und die inhärenten Probleme eines jeden Weges zu erkennen. Oberstes Ziel während der Entwicklung war es, eine angemessene Genauigkeit und Geschwindigkeit in Abhängigkeit von geringen Kosten und geringster Beeinflussung der Projektionen innerhalb der CAVE zu ermöglichen. Konkret bedeutet dies für das System so wenige Kameras einzusetzen, wie es die Verdeckungsproblematik der Targets erlaubt. Monoskopische Verfahren sind demzufolge den Stereoskopischen vorzuziehen. Da einige Schritte des Trackings sowohl bei mono- als auch bei stereoskopischen Verfahren anwendbar sind, wurde auch eine Testimplementierung mit Stereoskopischem Verfahren erstellt. Im Folgenden wird der stereoskopische Prototyp als *StereoTracker* und die monoskopische als *OptiTrack* bezeichnet.

Wie bereits aus den vorangegangenen Kapiteln ein zu sehen war, beginnt das Trackingproblem bereits mit der Auswahl einer geeigneten Hardware. Kapitel 8.2.2 beschreibt beispielsweise, welche Eigenschaft ein Marker haben kann, und wie er aufgebaut ist. Obwohl aktuelle Forschungen zunehmend markerloses Tracking erforschen (Grembowietz 2005; Rosenhahn, Perwass et al. 2005), ist die Verwendung eines Markers die einfachste Methode die Position des Zeichenstiftes zu ermitteln. Im finalen Prototyp wurde deshalb ein starrer 3D-

Multipunktmarker mit vier retroreflektiven Kugeln verwendet, die an den Zeichenwerkzeuge angebracht werden können. Auf diese Weise bekommt der physikalische Teil des Stiftes ein Koordinatensystem, welches wiederum hilft die virtuelle Spitze des Stiftes vor diesem zu platzieren. Zusätzlich können die Koordinaten vom Stift-Koordinatensystem mittels einer Transformationsmatrix leicht in das Cave-Koordinatensystem transformiert werden. Die Transformation zwischen Marker-Koordinatensystem und CAVE-Koordinatensystem ist deshalb möglich, da das System durch eine vorangehende Kalibrierung die Kameraposition im CAVE-Koordinatensystem kennt und eine Rekonstruktion des Marker-Koordinatensystems über die

Topologie des Markers errechnet werden kann. Gleiches gilt zur Ermittlung der Sicht auf das virtuelle Objekt. Über einen zweiten Multipunktmarker, der an der Shutterbrille der Skizzierenden Person

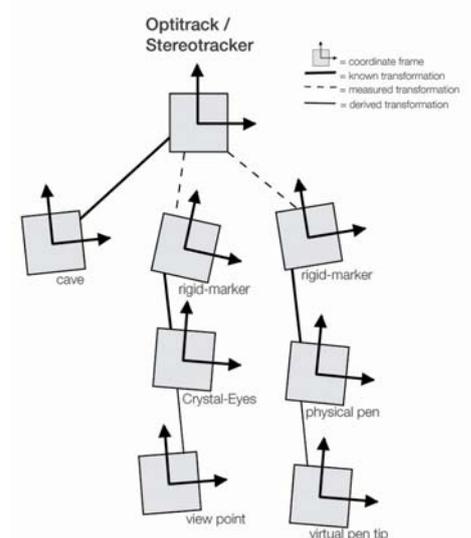


Abbildung 78 Zusammenhang der Koordinatensysteme

angebracht ist, wird dem System ein weiteres Koordinatensystem hinzugefügt, welches für die Berechnung der Sicht genutzt werden kann (vgl. Abbildung 78).

Das Hauptproblem ist also die Ermittlung der Lagen (Pose) der verschiedenen Koordinatensysteme zueinander. Dabei zerfällt das Problem in zwei Unterprobleme des optischen Trackings:

1. *Image Processing*, bei dem es vor allem um die geeignete Ermittlung von distinkten Bildmerkmalen geht (z.B. feature point extraction, blob extraction).
2. *Pose Estimation* (auch *view point solving*, *camera localization*, *3D registration*), welches die 3D-Pose aus 2D Bilddaten rekonstruiert.

Zudem bestimmt die Wahl des anzuwendenden *Pose Estimation* Algorithmus die Wahl der *Image Processing* Algorithmen und umgekehrt. Abbildung 79 illustriert einen kleinen Ausschnitt möglicher Lösungswege. Die grünen und roten Pfeile dabei zeigen, welche Wege in Testimplementierungen ausprobiert wurden. Die roten zeigen dabei die als weniger geeigneten Lösungswege, welches vor allem den Meanshift und den Camshift zum Tracken der Blobs in den Kamerabildern betrifft.

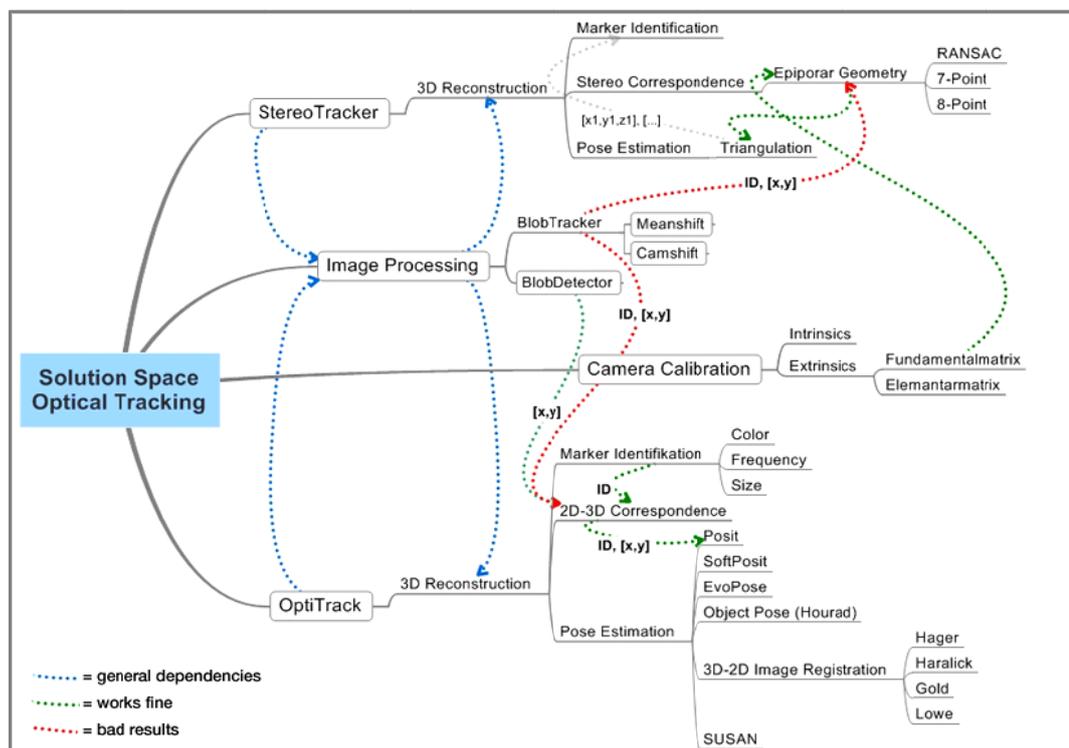


Abbildung 79 Ausschnitt möglicher Lösungswege

Meanshift bzw. Camshift (Continuously Adaptive Meanshift) ist ein Algorithmus aus dem Bereich der Bewegungsanalyse (Motion Analysis) und des Objekt Trackings (Object Tracking). Neben anderen

Algorithmen aus diesem Bereich wie Background Substraction, Motion Templates, Active Contours und Optical Flow, bieten diese beiden Algorithmen die Möglichkeit Objekte aufgrund ihrer Farbverteilung zu tracken. Da die beiden Algorithmen mit Suchfenstern (search windows) arbeiten, eignen sie sich weniger für das Tracken von schnellen Bewegungen. Denn bewegt sich ein Blob zu schnell innerhalb des Bildes, kann das Suchfenster eventuell nicht nachgeführt werden. In der Praxis trat dies sehr häufig auf und disqualifizierte somit den Mean- bzw. Camshift zum sinnvollen Einsatz in einer Tracking Anwendung (mehr dazu im Kapitel 9.4.1).

Die Vorgehensweise zu Ermittlung der Pose über den Multipunktmarker, gliedert sich für OptiTrack in die im folgenden Flussdiagramm dargestellten Schritte:

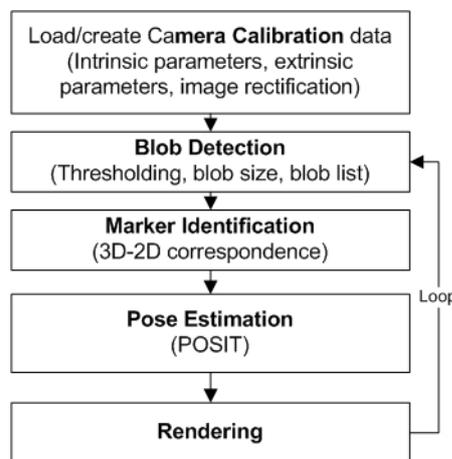


Abbildung 80 Flussdiagramm Optitrack

Bevor eine Pose Estimation, also die Ermittlung der X-, Y-, Z –Koordinaten und der Rotationswinkel θ , ϕ , ψ , möglich ist, muss ein genaues Kameramodell erstellt werden und anschließend eventuelle Verzerrungen des Objektivs kompensiert werden (Camera Calibration). Erst dann kann mit der Feature-Point- bzw. Blob-Extraction, Marker-Identifikation und der Pose Estimation fortgefahren werden.

Im Folgenden werden die einzelnen Schritte, die den Gesamtablauf des optischen Trackings bilden, näher erläutern und besondere Probleme dabei dargestellt.

9.1 Kamerakalibrierung

Als Voraussetzung für eine genaue Bestimmung der Pose, muss ein geeignetes Kameramodell erstellt werden. Es besteht aus den sog. intrinsischen Kameraparametern, die in einer Kalibrierungsmatrix (K) zusammengefasst werden.

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{aligned} \alpha_x &= k_x \cdot f_u \\ \alpha_y &= k_y \cdot f_v \\ s &= 0 \end{aligned}$$

- α_x und α_y sind Skalierungsfaktoren in x und y Richtung und setzen sich aus Produkt von Pixel pro Einheit (k_x, k_y) und der Brennweite zusammen.
- s ist der sog. Verzerrungsparameter (Skrew Factor), der für die üblicherweise angenommene Orthogonalität bei normalen Kameras gleich Null ist. Er ist ungleich Null, wenn x und y Richtung nicht senkrecht aufeinander stehen. Die Fire-i Kameras haben beispielsweise quadratische Pixel ($4,65 \mu\text{m} \times 4,65 \mu\text{m}$) und somit einen Skrew Factor von Null.
- Die optische Achse (Principle Point) wird mit $c = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$ angegeben.

Zusätzlich zu den intrinsischen Kameraparametern gibt es noch die extrinsischen Kameraparameter, die Orientierung und Position der Kamera $[R | t]$ zum Weltkoordinatensystem spezifizieren.

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}; t = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

Mit Hilfe der Kalibrierungsmatrix (K) und den extrinsischen Parametern kann die Projektion eines Punktes P im Weltkoordinatensystem auf den Punkt p im Bildkoordinatensystem berechnet werden.

$$p = K[R | t] \cdot P$$

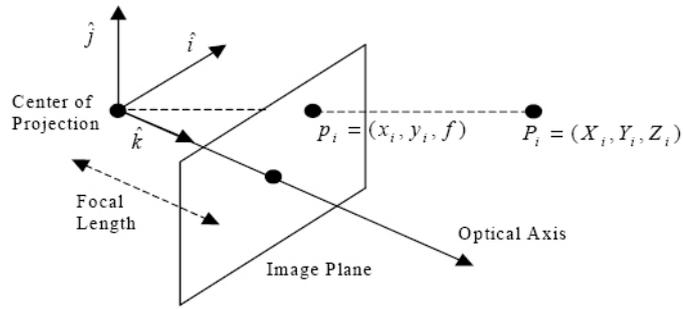


Abbildung 81 Projektion in die Bildebene einer Kamera

Abbildung 81 verdeutlicht die Projektion eines Punktes im dreidimensionalen Raum in die 2D Bildebene.

9.2 Entzerrung der Bildebene

Die Bilder einer Kamera weisen üblicherweise eine signifikante radiale Verzerrung auf. Gerade wenn niedrige Brennweiten, also Weitwinkelobjektive, genutzt werden sollte das Bild vor einer weiteren Prozessierung korrigiert werden.

Aus den verzerrten Bildkoordinaten $\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}$ können durch Modellierung der Verzerrung die idealen

Koordinaten approximiert werden durch:

$$\bar{x} = x(1 + \kappa r^2) \quad r^2 = x^2 + y^2$$

$$\bar{y} = y(1 + \kappa r^2) \quad \kappa \text{ ist ein Koeffizient für die radiale Verzerrung } r$$

Wobei x und y die idealen, verzerrungsfreien Bildkoordinaten sind und \bar{x} , \bar{y} die verzerrten Koordinaten repräsentieren.

Auch der CCD-Chip kann die Ursache von Verzerrungen sein, deshalb muss das Mapping

der Sensor Koordinaten $\begin{bmatrix} m \\ n \end{bmatrix}$ in das Bildkoordinatensystem modelliert werden. Das Mapping der

Sensorkoordinaten zu den verzerrten Bildkoordinaten kann mit

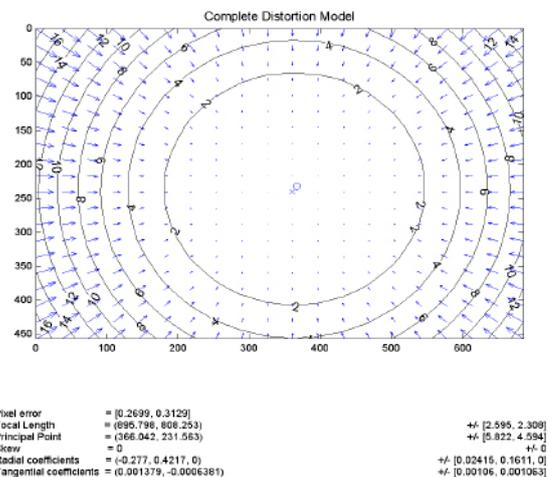


Abbildung 82 Radiale Verzerrung der Bildkoordinaten (Joshua A. Staples 2006)

$$x = \frac{(m - m_0)}{f_u} ; y = \frac{(n - n_0)}{f_v}$$

angegeben werden. Dabei bezeichnet m_0, n_0 den Bildmittelpunkt im Sensor Koordinatensystem und f_u, f_v die horizontale und vertikale Skalierung des Bildes die durch die Linse und Aufbau des CCD-Chips verursacht wird. Die Skalierung durch die Linsenbrennweite f kann in u und v Richtung unterschiedlich sein, da auch der horizontale und vertikale Abstand zwischen den einzelnen Sensorelementen unterschiedlich sein kann. Die Somit unterschiedlichen „Brennweiten“ lassen sich mit

$$f_u = \frac{f}{d_u} ; f_v = \frac{f}{d_v}, \text{ berechnen.}$$

Somit wäre ein vollständiges verzerrungsfreies Kameramodell mit intrinsischen und extrinsischen modelliert.

Da im Normalfall die benötigten Parameter des gesamten Kamerasystems nicht vorliegen, lag es lange daran, diese Kalibrierung der intrinsischen und extrinsischen Parameter zu automatisieren. Ein bekannter Algorithmus zur Kamerakalibrierung von Zhang (Zhang 1999; Zhang 2000) kalibriert eine Kamera wie folgt:

1. Es wird ein Kalibrierungsmuster (Calibration Pattern) mit leicht extrahierbaren Bildmerkmalen und bekanntem Aufbau in den Sichtbereich der Kamera gehalten. Anschließend wird die Homographie zwischen Punkte des Kalibrierungsmusters und den Punkten in der Bildebene der Kamera hergestellt. Es resultiert eine Homographie-Matrix mit der Bildkoordinaten in Weltkoordinaten umgerechnet werden können.
2. Initialisierung der intrinsischen Parameter . Die Verzerrung wird mit einem Wert von Null initialisiert.
3. Finden der extrinsischen Parameter für jedes während der Kalibrierung aufgenommenen Bildes.
4. Optimierung der angenommenen intrinsischen Parameter durch Minimierung des Fehlers in den Projektionspunkten mit allen Parametern.

Üblicherweise wird als Kalibrierungsmuster eine Schachbrettmuster oder ein Punktmuster verwendet, wie es in (Tsai 1986) erstmals beschrieben wurde.

9.3 Kamerakalibrierung in Bezug auf Opti- und StereoTracker

Während der Entwicklung der Programme StereoTracker und OptiTrack stellte sich heraus, dass die Kamerakalibrierung in besonderen Fällen vernachlässigbar ist. Abhängig ist dies von der Kombination des CCD-Chips mit einem entsprechenden Objektiv. Denn je kleiner die Brennweite eines Objektivs gewählt wird, desto größer wird der Öffnungswinkel (Weitwinkel) und gleichzeitig auch die Verzerrung des Kamerabildes. Da es für jede CCD-Chip-Größe eine Idealbrennweite gibt (Normalobjektiv), durch die eine minimale Verzerrung im Kamerabild entsteht, sind dann Methoden zur Entzerrung des Bildes vernachlässigbar. Als intrinsische Parameter lassen sich Brennweite und Mittelpunkt der Linse leicht ermitteln, da sie meist in den Datenblättern der Kameras vermerkt sind. Für die Fire-i Kamera ließ sich so eine Brennweite von 4,5mm und ein Mittelpunkt von einer halben Bildauflösung (bei 640x480px folglich $x_0 = 320px$ und $y_0 = 240px$) nutzen. Da die Maßeinheit in der Kalibrierungsmatrix in Pixel angegeben wird, muss die Maßeinheit der Brennweite entsprechend umgerechnet werden. Dies kann durch Multiplikation der Brennweite (in mm) mit der Pixelanzahl pro cm des CCD-Chips einfach errechnet werden. Bei der Fire-i ergibt sich somit eine Brennweite von 666px.

Im Falle der Fire-i 620c inkl. Pentax Objektiv ist eine „Abschätzung“ der Kamerakalibrierung nicht möglich, da das Objektiv bereits ein Weitwinkelobjektiv ist und starke Krümmungen innerhalb des Kamerabildes verursacht. Es muss also unbedingt eine Kalibrierung stattfinden, bevor die Bilddaten weiter verwendet werden können.

9.4 Ermittlung und Tracking von distinkten Bildmerkmalen

Die Extraktion von Bildmerkmalen aus den Kamerabildern ist ein wesentlicher Schritt im optischen Tracking, denn aufgrund der ermittelten Bildkoordinaten finden alle weiteren Berechnungen zur Ermittlung der Pose des Markers statt. Dabei können auch hier verschiedene Strategien angewandt werden, um zuverlässig Bildmerkmale extrahieren zu können. Neben Background-Subtraction, konturbasierender Extraktion von Bildmerkmalen (contour based feature extraction) und Extraktion von Kanten, ist das Tracking von Farben und Funktionen wie Thresholding, Binarization, Erosion und Dilatation in Bezug auf Opti- und StereoTracker besonders nützlich.

Während der Entwicklung der Testimplementierungen wurden zwei Wege für eine geeignete Blobextraktion eingeschlagen. Das Blobtracking mittels Mean-/Camshift und ein eigener Blobdetection Algorithmus mit zusätzlichen Heuristiken. Beide werden nachfolgend beschrieben, um die Vorteile und die Nachteile jeweils zu erläutern.

9.4.1 Color Blob Tracking via Meanshift/Camshift

Wie bereits im Voraus schon mehrfach erwähnt, hat sich der Mean-/Camshift Algorithmus für ein eventuelles Produktivsystem der optischen Trackinganwendung nicht als einsetzbar erwiesen. Dies wurde jedoch erst nach einer Implementierung in C++ deutlich. Dennoch erwies er sich bei der Entwicklung als nützlich, da zum Test des Pose Estimation Algorithmus die Bildkoordinaten auf irgendeine Weise aus den Bildern extrahiert werden mussten. Im Folgenden wird kurz beschrieben, wie dieser Algorithmus arbeitet.

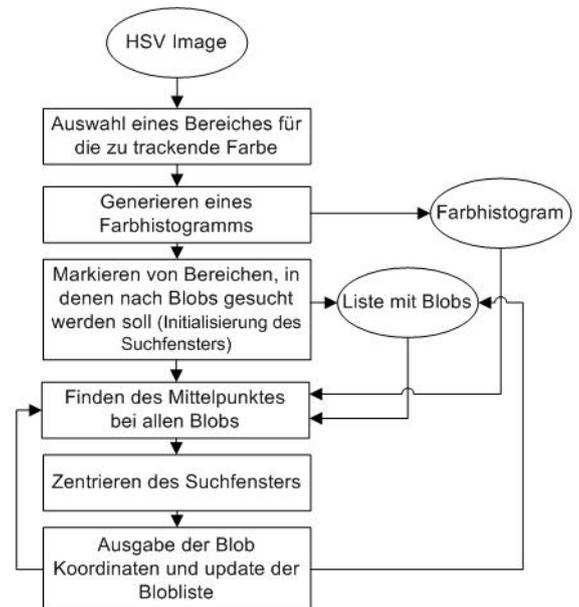


Abbildung 83 Blobtracking mittels Meanshift

Als Eingabe dient ein Bild im HSV Format. Dieses Farbmodell hat den Vorteil, dass der Farbton sehr leicht extrahiert werden kann, da der resultierende Farbton, nicht wie im RGB Raum auf drei Kanäle verteilt ist, sondern in einem Kanal (H also Hue) vorliegt. Somit haben die Farbwerte eine hohe Trennschärfe gegeneinander und bieten eine leichte Weiterverarbeitung zur Berechnung eines Farbhistogramm. Zur Initialisierung des Blobtrackers muss bei der Testimplementierung ein Rahmen per Maus aufgespannt werden um einen Bereich zu Markieren. Aus diesem Pixelbereich wird dann ein Farbhistogramm der Einzelnen Pixelfarbwerte erstellt, welches die häufigsten Farben in einer Struktur speichert.

Danach markiert man Farbbereiche, die man im Kamerabild tracken möchte. In diesem Falle sind dies die neon-farbenen Markerkugeln des in Kapitel 8.2.2 beschriebenen Multipunktmarkers. Es werden sog. Suchfenster um die zu trackenden Kugeln erstellt, deren Mittelpunkt als initiale Koordinate des Blobs in einer Liste gespeichert wird. Somit bekommt jedes Suchfenster eine eindeutige ID die es an die Liste mit den Blobs weitergibt. Somit sind mehrere eindeutig identifizierbare Blobs in der Liste gespeichert.

Die einzelnen Pixel in den Suchfenstern werden nun mit dem gespeicherten Farbhistogramm verglichen ist der Farbton (Wert) eines Pixels mit den im Histogramm hinterlegten in etwa (gesetzter Threshold) gleich, so wird der Pixel markiert. Auf diese Weise ergeben sich viele markiere Pixel die letzten Endes den Blob bilden. Von diesem Blob wird dann der Mittelpunkt bestimmt und als Koordinate in der

entsprechenden Stelle der Blobliste gespeichert. Da sich der Blob bewegt, wird zudem das Suchfenster dem neu bestimmten Mittelpunkt nachgeführt. Somit ergibt sich ein kontinuierliches Tracking einer Markerkugel.

Die angesprochenen Probleme dieses Algorithmus begründen sich einerseits auf den eingesetzten Suchfenster und andererseits auf dem Farbhistogramm.

Abbildung 84 verdeutlicht das Problem der Suchfenster. Die Abbildung zeigt den Multipunktmarker in verschiedenen Abständen zur Kamera. Im linken Bild sind die vier Suchfenster an ihrer jeweils richtigen Position.

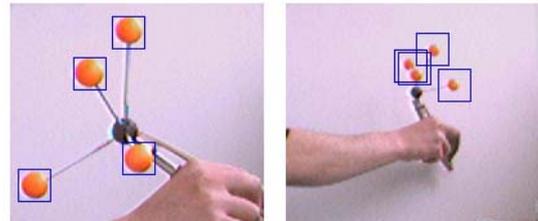


Abbildung 84 Meanshift Suchfensterproblem

Entfernt man sich nun von der Kamera, so werden die Kugeln immer kleiner und die Suchfenster springen auf andere Kugeln mit gleicher Farbe über. Initialisiert man kleinere Suchfenster wie in Abbildung 85, so kann kein genauer Mittelpunkt der Markerkugel ermittelt werden, da wenn man mit dem Marker näher an die Kamera heran geht, die Markerkugel größer als das Suchfenster wird.

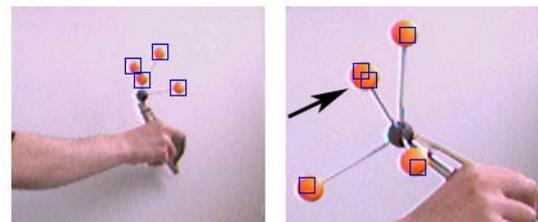


Abbildung 85 Meanshift Initialisierungsproblem

Die resultierenden Koordinaten sind somit sehr instabil, da das Suchfenster innerhalb der Kugel hin und her springt, im Versuch, die passendsten Farbwerte aus dem Histogramm innerhalb der Kugel zu finden. Man kann dieses Problem durch adaptive Suchfenster verringern, wie es beispielsweise der Camshift Algorithmus macht. Leider ergibt sich daraus ein neues Problem, welches die Zuverlässigkeit des Tracking beeinflusst. Befindet sich der Marker nämlich sehr weit weg, so wird das Suchfenster immer

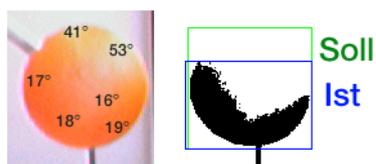


Abbildung 86 Beeinflussung des Lichtes auf den Blob

empfindlicher gegenüber den Bewegungen des Markers und das Tracking geht verloren.

Die Tatsache, dass bei dieser Art von Tracking die Blobs aus dem Farbton heraus erstellt werden, führt zu einem weiteren Problem.

Denn ist ein Farbhistogramm nicht richtig initialisiert worden, so kann es vorkommen, dass wie in Abbildung 86 kein runder sondern ein halbmondförmiger Blob entsteht. Dies wirkt sich natürlich auf die Ermittlung des Mittelpunktes, also der Blobkoordinaten aus, welche aufgrund dessen bis zu 5mm abweichen können. Auch wenn das Histogramm zunächst richtig initialisiert wurde, so kann eine Veränderung der Lichtsituation die Verteilung der Farbtonwerte (wie sie beispielhaft in der Abbildung 86 eingetragen wurden) erheblich

beeinflussen. Besonders in der Cave ist dies ein Problem, da die Projektoren je nach Skizziersituation eine andere Lichtsituation schaffen. Vor allem dieser Grund disqualifizierte das Farbtracking, im Allgemeinen, und den Camshift, im Besonderen, für den Einsatz in Optitrack.

9.4.2 Blobdetection

Die Idee der Blobdetection resultierte aus der Entscheidung, dass ein Farbtracking für OptiTrack nicht zielführend ist. Neben den oben genannten Problemen bestand zusätzlich das Problem des Verlustes des Trackings. Verlieren die Suchfenster nämlich das zu trackende Objekt, so konnte keine automatische Reinitialisierung durchgeführt werden. Auch das hängen bleiben mehrere Suchfenster an einem Blob, für den Fall, dass sich Kugeln kurzzeitig überlappen sollte vermieden werden. Ein Überdenken der Gesamtsituation führte zudem dazu, Marker, Kamera und Beleuchtung zu modifizieren und auf ein IR-Setup auszurichten wie in den Kapiteln 8.2 bereits erläutert wurde.

Der Begriff Blobdetector wurde hier bewusst gewählt, da für jedes Eingabebild die Blobs neu erkannt werden und kein echtes Tracking (also Verfolgen) der Blobs durchgeführt wird. Dies hat den Vorteil, dass bei Verdeckung oder schnellen Bewegungen der Algorithmus nicht abbricht, wie es beim Mean-/Camshift Algorithmus der Fall ist.

Abbildung 87 zeigt die einzelnen Schritte des Blobdetectors. Als Eingabebild dient hier ein Graustufenbild (Grey Image), welches mit einer Kamera mit IR-Hochpass-Filter aufgenommen wurde. Anschließend wird eine Binarisierung des Bildes durch die Anwendung eines Grenzwertes (Threshold) durchgeführt. Das so entstandene Schwarz-Weiß-Bild wird dann mehrmals gefiltert. Eine Erosion eliminiert verbliebene Pixelartefakte. Wobei die anschließende Dilatation eine Auffüllung der Blobs bewirkt, um einerseits die ursprüngliche Größe wieder herzustellen und andererseits evtl. entstehende „Risse“ im Blob zu schließen.

Der dritte Filter besteht derzeit aus einer Funktion zum Herausfinden des größten Blobs. Hierbei wird ermittelt, welcher Blob die meiste Fläche besitzt. Dieser Filter behebt ein wesentliches Problem, welches bezüglich des verwendeten

POSIT-Algorithmus in Optitrack besteht (siehe Kapitel 10.3.1). Dieser fordert nämlich, dass ein Referenzpunkt der erkannten Blobs „genannt“ wird. Die Identifizierung dieses Punktes, der beim Mean-/Camshift-Algorithmus implizit durch die Initialisierung der Suchfenster geschah, muss in diesem Fall

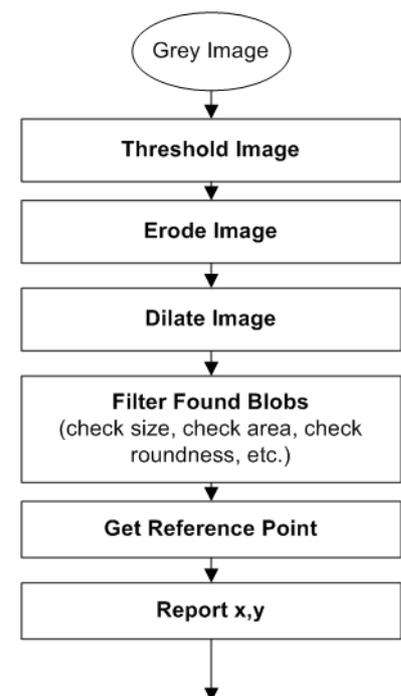


Abbildung 87 Flussdiagramm des Blobdetector

durch eine Filterfunktion realisiert werden. Abbildung 88 zeigt den ermittelten Referenzpunkt in gelb. Diese Methode konnte leider nur mit der Fire-i Kamera getestet werden, welche lediglich eine Auflösung von 640x480 zuließ. Aber bereits in dieser Auflösung wurde der Referenzpunkt des Multipunktmarkers zuverlässig bis zu einer Entfernung von ca. 2,30m erkannt. Dabei hatte die Referenzkugel einen Durchmesser von 2cm und war somit nur 0,5cm größer als die restlichen drei Markerkugeln. Zwar ist diese Situation bereits grenzwertig (vgl. Tabelle 11 Theoretische Blobgrößen), kann aber durch Optimierung des Markers bzw. eine höhere Auflösung leicht verbessert werden.

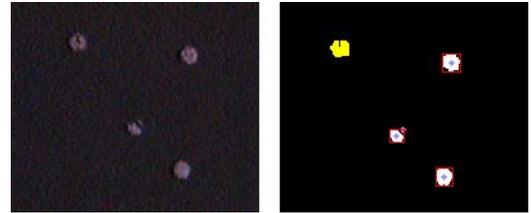


Abbildung 88 Blobdetector Ergebnis

Letztendlich liefert der Blobdetection-Algorithmus die ermittelten Mittelpunktkoordinaten als Liste aus, welche zur Pose Estimation weiter genutzt wird.

Falls nötig, lassen sich später sogar weitere Filter integrieren. So wäre z.B. ein Filter zur Erkennung „falscher Markerkugeln“ wie sie etwa durch Reflektionen entstehen können, sinnvoll. Dies könnte beispielsweise durch eine Methode zur Erkennung der Rundheit geschehen, wie in (Russ 1995) beschrieben.

9.4.3 Ergebnis

Die Extraktion von Bildmerkmalen zählt zu der Kategorie des *Image Processings* und ist ein wichtiger Schritt vor der *Pose Estimation*. Wie die beiden vorangehenden Kapitel beschreiben, wurden zwei verschiedene Wege erkundet, die zum optischen Tracking im Umfeld der CAVE beitragen sollen. Dabei wurde klar, dass eine Ermittlung der Koordinaten über Mean-/Camshift nicht zielführend ist. Ein Überdenken des gesamten Systems führte dazu, auf ein IR-System umzusteigen. Ausstattungen der Kamera mit IR-Pass-Filter, Bau eines IR-LED-Scheinwerfers sowie eines Multipunktmarkers mit retroreflektierenden Kugeln waren die Folge. Mit einem auf diesem IR-Setup ausgerichteten Algorithmus konnte schließlich eine zuverlässige Ausgabe von Blob-Koordinaten erreicht werden. Nachfolgend werden die Vor- und Nachteile der beiden Wege noch einmal zusammengefasst.

	Blobtracking (Mean-/Camshift)	Blobdetection (IR-Setup)
Vorteile	<ul style="list-style-type: none"> • Flexibilität: Der Farbtracker kann beliebige Farben eines HSV-Bildes tracken. • Blobs können eindeutig identifiziert werden. 	<ul style="list-style-type: none"> • Lichtverhältnisse haben einen geringen Einfluss • Tracking kann nicht abbrechen, da in jedem Bild die Blobs neu bestimmt werden • Referenzpunkt kann angegeben werden
Nachteile	<ul style="list-style-type: none"> • Zuverlässigkeit ist stark von den Lichtverhältnissen abhängig. • Tracking geht bei schnellen Bewegungen verloren • Tracking kann nicht wiederaufgenommen werden • Tracker muss initialisiert werden 	<ul style="list-style-type: none"> • Flexibilität: Der Blobtracker erfordert ein IR-Setup

Tabelle 12 Vergleich zweier Methoden zur Bildmerkmalsextraktion

Ein Problem, welches keiner der beiden Wege löst, ist das Problem der Verdeckung, welche im Normalfall, nur durch Prädiktionsalgorithmen wie lineare Prädiktion oder dem Kalmanfilter in einem nachgelagertem Post-Processing-Schritt (vgl. Schritt 4 in Abbildung 89) gelöst werden kann. Dennoch kann gesagt werden, dass der Blobdetector weniger anfällig gegenüber der Verdeckung ist, da auch kurzfristig verdeckte Markerkugeln ohnehin bei wieder erscheinen im Bild erkannt werden und ein Tracking nicht verloren gehen kann.

Wie Abbildung 89 zeigt, gehören zum robusten Blob-Tracking noch weitere Schritte, als die hier angesprochenen (grün markiert). Besonders die nachgelagerten Schritte der Prozessierung der entstehenden Trajektorien bieten weiteres Optimierungspotential, unter die auch die Prädiktion fällt.

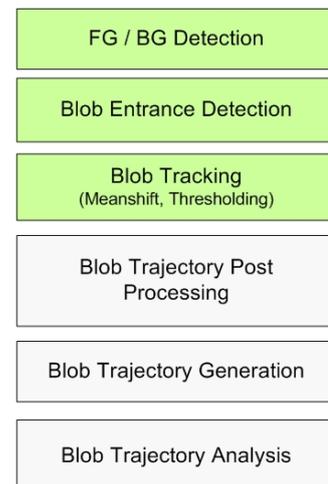


Abbildung 89 Blob-Tracking Pipeline

9.5 Pose Estimation

Wie bereits erwähnt, handelt es sich bei der Pose Estimation um ein Problem der Projektion. Ziel ist es, Koordinatenpunkte aus dem 3D-Raum, welche lediglich als Punkte in der 2D-Bildebene vorliegen, zu bestimmen. Hierbei lässt sich die Korrespondenz zwischen 3D-Punkten und 2D-Bildkoordinaten wie folgt formalisieren:

$$\text{Es sei } P = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

ein Punkt eines gegebenen Objektes in seinem jeweiligen Koordinatensystem (beispielsweise der Punkt eines Multipunktmarker-Modells)

$$\text{und } p = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \text{ die Projektion dessen auf die Bildebene,}$$

wobei die Tiefeninformation z verloren geht und deshalb $z' = 1$ ist. Dieses Vorgehen wird als Normalisierung der Bildkoordinaten bezeichnet. Der resultierende normalisierte Punkt wäre also:

$$\mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix}$$

Die Koordinaten sind voneinander abhängig. D.h., wenn sich der Punkt des Modells bewegt, bewegt sich logischerweise auch dessen Projektion. Sie lassen sich wie folgt zueinander transformieren:

$$p = RP + t$$

$$\text{Wobei } R = \begin{bmatrix} r_1^t \\ r_2^t \\ r_3^t \end{bmatrix} \text{ und } t = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \text{ die Rotationsmatrix und der Translationsvektor ist.}$$

Unter der Verwendung des zuvor in Kapitel 9.1 beschriebenen idealen Lochkamera-Modells,

$$\text{sind } p = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}, \text{ der Bildpunkt } p = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \text{ und der Mittelpunkt der Kamera kollinear.}$$

Die obige Gleichung ($p = RP + t$) kann also als $p = \frac{1}{r_3^t + t_3}(RP + t)$ geschrieben werden.

Diese Gleichung wird in der Photogrammetrie auch als Kollinearitätsgleichung bezeichnet. Die erhaltenen Bildkoordinaten sind meist jedoch fehlerbehaftet, da das System einem gewissen Rauschen unterliegt. Dies kann mit $\bar{p} = P + \varepsilon$ modelliert werden.

Sind die Vektoren p und P inklusive der z' Bildkoordinaten, die etwa durch Entfernungsmessung oder stereo-matching oder structure-from-motion bestimmt wurden, bekannt, so ist auch die Translationsmatrix bekannt. Die Rotationsmatrix lässt sich durch Least-Squares-Methoden wie das Gauss-Newton- oder das Lavenberg-Marquardt-Verfahren berechnen. Das Vorgehen zum Finden der Rotations- und Translationsmatrix um P in p transformieren zu können wird auch 3D-3D Pose Estimation genannt.

9.5.1 Stereoskopische Pose Estimation

Bisher sind die durchzuführenden Schritte, wie z.B. die Kalibrierung der Kamera, Blob-Extraction und Berechnung der Translations- und Rotationsmatrix bei monoskopischen und stereoskopischen Tracking-Verfahren gleich. Um die Pose aus 2D-Bildkoordinaten berechnen zu können muss nun also der Abstand des Markers in Z-Richtung ermittelt werden (auch 3D-Rekonstruktion genannt). Dies geschieht bei dem stereoskopischen Tracking über die sog. Triangulation. Als Voraussetzung zur Durchführung der Triangulation müssen die Kameras von ihrer gegenseitigen Position und Orientierung wissen. Anschließend muss von jeder Kamera ein Bild geliefert werden. Hierbei ist es wichtig, dass die beiden Bilder zum möglichst gleichen Zeitpunkt aufgenommen wurden, um Ungenauigkeiten zu vermeiden. Mit Hilfe der beiden Kamerabilder, in denen je der Punkt P als Projektion p sichtbar ist, kann z über den Strahlensatz der Elementargeometrie berechnet werden.

$$\frac{z}{\text{Basis}} = \frac{f}{\text{Abstand}} \quad \rightarrow \quad z = \frac{b \cdot f}{(x_1 - x_2)}$$

Dabei verhält sich z zum Abstand der beiden Kameras (auch Basis genannt) wie die Brennweite f zu dem Abstand der Punkte p in den beiden Bildebenen.

Ist nur ein Feature Point im Bild zu sehen, ist diese Methode einfach durchzuführen. Zur vollständigen Rekonstruktion der Pose sind aber mindestens drei Punkte zu rekonstruieren, welche die Basisvektoren, des Koordinatensystem des zu trackenden Markers bilden. Es sind im Falle der 6DOF Pose Estimation also mehrere Blobs (≥ 3) im linken und rechten Kamerabild zu sehen. Da für die Triangulation der Abstand zweier korrespondierender Punkte benötigt wird, ergibt sich ein neues Problem, welches als Stereokorrespondenzproblem in der Literatur bekannt ist. Es muss also eine Methode gefunden werden, wie Punktepaare in den beiden Bildern ermittelt werden können. Ein Weg wäre hier wiederum das

Blobtracking per Meanshift oder Camshift. Durch kontinuierliches Tracking wüsste man nach einer Initialisierung immer welche Punkte zusammengehören. Wie jedoch bereits erwähnt, verliert diese Art von Tracking bei schnellen Bewegungen den zu trackenden Blob.

Elegantere Methoden zur Ermittlung von Punktepaaren bietet die Epipolargeometrie (vgl. Testimplementierung in Kapitel 10.2.1). Durch sie kann eine Translationsmatrix (Fundamentalmatrix) zwischen den Kamerabildern erstellt werden, die es erlaubt aus Punkten im ersten Bild sog. Epipolarlinien im zweiten Kamerabild zu berechnen, auf denen dann die korrespondierenden Punkte in etwa liegen müssen. Meist liegt der gesuchte Punkt nicht ganz auf der Epipolarlinie, weswegen vorausschauender Weise in der Nähe um die Epipolarlinie herum der entsprechende Punkt gesucht werden sollte. Somit wäre die sog. Stereokorrespondenz herzustellen und zu rekonstruieren.

9.5.2 Monoskopische Pose Estimation

Im Gegensatz zum stereoskopischen Tracking werden beim monoskopischen Tracking nur Informationen aus einem Bild für die Pose Estimation verwendet. Veröffentlichte Algorithmen dazu sind vor allem von Ansar, Huttenlocher, Ullmann, Hager, Hourad, Feldmar, Wunsch, Gold, Lowe, Davis, DeMenthon. Dabei wird das Gesamtproblem meist auf nicht-lineare Minimierungstechniken, wie z.B. das Gauß-Newton oder Lavenberg-Marquard Verfahren zurückgeführt. Neuere Algorithmen setzen statt auf nicht-lineare auf lineare Regressions-Methoden, wie z.B. Iterative Re-weighted Least Squares (IRLS), die Projective Transformation Matrix (PTM) Methode oder die Radial Alignment Constraint (RAC) Methode. Einige Algorithmen wie der Two-Step Algorithmus kombinieren lineare Algorithmen (um zunächst eine ungefähre Lösung zu bekommen) mit nicht-linearen (um finale Genauigkeit zu bekommen), bei denen die ungefähre Lösung aus dem linearen Ansatz als Ausgangspunkt für den nicht-linearen Ansatz genutzt wird. Neben den linearen und nicht-linearen Least-Squares Methoden gibt es weiterhin noch Ansätze eine Lösung auf analytischem Weg, wie dem der Algebra oder der Geometrie, zu finden. Das Anwendungsgebiet für diese Ansätze liegt jedoch eher im Bereich der Objekterkennung.

Das Problem der monoskopischen Pose Estimation beruht vor allem darauf, dass sich bei der Projektion der 3D-Punkte in die 2D-Bildebene ein nicht-lineares Verhalten ergibt und nicht-lineare Algorithmen wie Lavenberg-Marquard nur dann konvergieren, wenn bereits eine ungefähre Lösung existiert. D.h., die 3D-Position des zu trackenden Marker müsste fast vollständig bekannt sein (vorhandene initiale Pose). Weiterhin lassen sich die bestehenden Algorithmen in modell-basierende (model based) und modell-freie (model free) Ansätze einteilen. Bei den Modellbasierenden liegt ein 3D-Modell als Datensatz (z.B. CAD-Modell) oder Struktur vor und dient der Berechnung dessen Pose im dreidimensionalen Raum

(auch Image Registration oder 3D-2D Registration genannt). Der modellfreie Ansatz vergleicht dagegen ein Referenzbild mit dem aktuellen.

9.5.3 POSIT als mögliche Lösung zur Pose Estimation

Die für die Anwendung SketchApp angedachten Lösungen beruhen auf den modellbasierenden und linearen Lösungsmethoden unter Verwendung der sog. Weak-Perspective-Projection⁸ beschrieben in (R. Hourad 1997). Die Methode wird als POS (Position from Orthography and Scaling) bzw. POSIT (POS mit Iterationen) bezeichnet und wurde von DeMenthon und Davis (DeMenthon & Davis 1995) entwickelt. Als Voraussetzungen für die Pose Estimation über die Weak-Perspective-Projection, sollte die Kamera kalibriert sein (intrinsischen Parameter bekannt und Verzerrungen im Kamerabild beseitigt). Als Marker sollte ein 4-Punkt-Marker oder ein Marker mit mehr als vier Kugeln verwendet werden, wobei mehrere Markerkugeln die Genauigkeit in Rotation und Translation nicht besonders erhöhen (I. Lopez 2002). Die obere Grenze der Feature-Points (bzw. Markerkugeln) kann für bleibende Echtzeitfähigkeit des Systems (>24fps) kann mit ca. 30 Punkten angegeben werden (Groher 2008). Weiterhin muss die 3D-2D Korrespondenz eines Referenzpunktes bekannt sein, was für die Anwendung SketchApp zunächst zu praktischen Problemen führte, da einzelne Blobs nicht zuverlässig getrackt werden konnten. Kapitel 9.4.2 beschreibt ein Workaround dieses Problems.

<i>Algorithmus</i>	<i>Haralick</i>	<i>Hager</i>	<i>DeMenthon</i>
Kategorie	Klassische iterative Lösung	Minimierung des Objekt Space Errors	Weak-Perspective Approximation
Methode	Einführung einer Tiefenwertes d_i für jeden Punkt in der Bildebene. Anschließende Rückprojektion in das Kamerakoordinatensystem und Berechnung der Rotation und Translation (3D-3D Pose Estimation). Ist der minimierte Fehler klein genug, stoppt der Algorithmus, ansonsten Berechnung neuer Tiefenwerte mit neuer Rotationsmatrix.	Einführung einer Orthogonal-Projektions-Matrix V_i für jeden Punkt in der Bildebene. Mit dieser Matrix werden die Punkte in der Bildebene in das Kamerakoordinatensystem transformiert und eine 3D-3D Pose Estimation durchgeführt. Ist der Fehler klein genug, bricht der Algorithmus ab, wenn nicht werden aus den Tiefenwerten neue Orthogonal-Projektions-Matrizen berechnet.	Nutzt eine Skalierte Orthographische Projektion (SOP), im Gegensatz zur Orthographischen Projektion, zur Approximation der Rotation und Translation. Ist der minimierte Fehler noch zu groß, werden die Objektpunkte in das Kamerakoordinatensystem transformiert und eine neue SOP generiert.

⁸ Grundlagen zur 3D-2D Projektion siehe APPENDIX E

Vorteile	Robust gegenüber Rauschen	Schnelle Konvergenz in wenigen Iterationen	Schneller Konvergenz (4-5 Iterationen)
Nachteil	Konvergiert sehr langsam und Initiale Rotation ist nötig		Funktioniert nicht bei nahen Objekten und das Objekt darf nicht zu weit von der Z-Achse der Kamera entfernt sein

Tabelle 13 Vergleich iterativer Pose Estimation Algorithmen

Die Wahl des POSIT-Algorithmus für das Lösen des Pose Estimation Problems lässt sich dahingehend begründen, als dass er sehr schnell konvergiert und wenige Iterationen benötigt und somit Echtzeitfähigkeit des Gesamtsystems erlaubt. Zudem ist er sehr Robust gegenüber Verdeckungen einzelner Kugeln eines Multipunktmarkers. Funktionierende bestehende Implementierungen wie das OpenSource Programm FreeTrack (FreeTrack 2008) bestätigen zudem seine Anwendbarkeit. Einzig bestehendes Problem zur Anwendbarkeit ist wie bereits erwähnt die notwendige 3D-2D Korrespondenz eines Referenzpunktes.

9.5.4 SoftPOSIT

Da während des Entwurfes die Prämisse der notwendigen 3D-2D Korrespondenz eines Referenzpunktes als zunächst unlösbares Problem angesehen wurde, wurde nach einer Alternativen gesucht, die dieses Problem nicht hatte. Diese wurde mit dem sog. SoftPOSIT-Algorithmus, welcher eine Weiterentwicklung des POSIT von David (Philip David 2004) ist, auch gefunden. Beim SoftPOSIT handelt es sich um einen kombinierten Algorithmus, des iterativen SoftAssign-Algorithmus von Gold (S. Gold 1996) mit dem POSIT-Algorithmus. Um die vollständige Pose rekonstruieren zu können ist außer dem Modell eines Objektes und den dazugehörigen 2D-Punkten keine weitere Information nötig. Die Exzellenz des Algorithmus besteht zudem darin, Pose und Korrespondenz simultan lösen zu können, ähnlich dem Hypothesis-and-Test Ansatz der bei dem RANSAC Algorithmus genutzt wird. Wobei jedes der beiden Probleme leicht zu lösen wäre, wenn bereits eines gelöst ist. Der beim SoftPosit angewandte SoftAssign Algorithmus (S. Gold 1998) bewältigt das Korrespondenzproblem, während POSIT die Pose bestimmt.

Da der SoftPOSIT derzeit die eleganteste Lösung zum Tracken von Multipunktmarkern zu sein schien, wurde eine Testimplementierung in C++ auf Basis von (DeMenthon 2003) erstellt. Wie sich jedoch herausstellte, ist der Algorithmus derzeit nicht Echtzeitfähig, da er im Mittel eine Ausführungsdauer von 140ms⁹ benötigt. Sicherlich ist diese Zeit durch weitere Optimierung der Testimplementierung zu verkürzen. Diese hätten aber weit aus mehr Zeit in Anspruch genommen, als im Rahmen dieser Arbeit zur Verfügung stand.

⁹ Das Testsystem bestand aus einem Intel Core2 Duo T7300 Prozessor mit 2,0 GHZ

9.6 Identifikation der Marker

Wie bereits in den vorangegangenen Kapiteln immer wieder erwähnt wurde, führt die Identifikation von Markerkugeln in den verschiedenen Koordinatensystemen (vgl. Abbildung 78) immer wieder zu Problemen, an der die Realisierung eines Gesamtsystems durchaus scheitern kann. Dabei geht es im Bezug auf SketchApp vor allem um zwei Probleme die unter dem Begriff „Identifikation“ zusammengefasst sind. Die Identifikation eines bestimmten 2D-Punktes in der Bildebene der Kamera zu einem dazugehörigen 3D-Modellpunkt und weiterhin die Identifikation eines bestimmten Multipunktmarkers gegenüber einem anderen Multipunktmarkers. Ersteres von beiden wird auch unter dem Begriff des Korrespondenzproblem es referenziert.

9.6.1 3D-2D-3D Korrespondenz

Das Problem der Korrespondenz stellt sich in der Computer Vision immer dann ein, wenn für Punkte aus einem Koordinatensystem der jeweilige „gleiche“ Punkt in einem anderen Koordinatensystem gefunden werden soll. Der POSIT-Algorithmus schafft eine solche Situation, da er als Eingabewert, die 2D-Koordinaten eines Referenzpunktes aus dem Kamerabild benötigt, um das im Speicher vorliegende 3D-Punktmodell mit diesem abgleichen und gleichzeitig die Pose errechnen zu können. Es ist also eine Methode notwendig, mit der man eine Korrespondenz zwischen Markerkugel des Multipunktmarkers und letztendlich dem 3D-Modell schafft. In Kapitel 9.4 „Ermittlung und Tracking von distinkten Bildmerkmalen“ wurden hierzu bereits zwei verschiedene Wege vorgeschlagen. Das Tracking der projizierten Markerkugeln und die Differenzierung des Referenzpunktes durch eine andere Größe (gegenüber den anderen Markerkugeln). Beide Methoden schaffen eine eindeutige Korrespondenz mit eigenen Vor- und Nachteilen.

Da sich aus in Kapitel 9.4 genannten Gründen für einen Blobdetector entschieden wurden, sind an dieser Stelle keine weiteren Korrespondenzprobleme vorhanden, solange nur ein Multipunktmarker verwendet wird. Bei der Verwendung eines weiteren Multipunktmarker, wie er beispielsweise für das Headtracking benötigt wird, wären zwei Referenzpunkte im Kamerabild zu sehen und der implementierte Algorithmus funktioniert somit nicht mehr bzw. bedarf einer Ergänzung. Im Rahmen der zur Verfügung stehenden Zeit, konnte hier keine geeignete Lösung gefunden werden.

9.6.2 Identifikation eines Multipunktmarkers

Auch wenn sich das im vorangegangenen Kapitel beschriebene Problem evtl. lösen lässt (beispielsweise durch entsprechende Heuristiken) und man zwei Multipunktmarker gleichzeitig Tracken kann, ergibt sich dennoch das Problem, dass Multipunktmarker A von Multipunktmarker B unterschieden werden

muss, da sie etwa für verschiedene Skizzierwerkzeuge eingesetzt werden. Auch hier kann derzeit keine Lösung gegeben werden. Ansätze wären eventuell:

- Weitere verschiedene Größen der Referenzpunkte
- Aktive Marker wie Beispielsweise in (William Hoff 2000)
- Tracking der Punkte in 2D oder in 3D
- Bewegungsmodelle und Wahrscheinlichkeitstheorie
- Treemaps

Existierende Systeme wie der Cyclope Tracker von Herve Mathieu (Mathieu 2005) demonstrieren, dass es auch hierzu eine Lösung gibt. Zwar kann das System, wie aus den Spezifikationen des Cyclope Trackers hervorgeht, nur vier Multipunktmarker unterschieden, aber auch das würde für die spezielle Situation der CAVE ausreichen.

10 Testimplementierungen für die Skizzieranwendung

Für die Erweiterung der Skizzieranwendung SketchApp wurden mehrere Prototypen bzw.

Testimplementierungen auf Linux-Basis erstellt. Aus Gründen der Performance wurde dabei die Programmiersprache C/C++ verwendet in Kombination mit der OpenCV Bibliothek.

OpenCV ist eine Computer Vision Bibliothek von Intel und basiert auf der Image Processing Library (IPL) und wurde erstmals im Jahr 2000 als OpenSource veröffentlicht. Die Entwicklung zielt vor allem auf Echtzeitanwendungen im Bereich Computer Vision ab und bietet mit Intels Integrated Performance Primitives (IPP) Bibliothek einen Prozessor optimierten, äußerst leistungsfähigen Funktionsumfang.

Die Testimplementierungen bestehen, wie bereits erwähnt, in zwei Basisprogrammen – StereoTracker und OptiTrack, die jeweils mit verschiedenen aber auch teilweise gleichen Klassen erweitert wurden. Beide Programme beinhalten, z.B. neben den zu testenden Pose Estimation Funktionen, ein Modul zur Speicherung bereits prozessierter Bildfolgen im DIVX-Format. Dazu wurde OpenCV mit der OpenSource Video-Bibliothek ffmpeg kompiliert.

Weitere verwendete C/C++ Bibliotheken waren:

- OpenGL zum Rendern der aktuellen Pose des Skizzierstiftes im VR
- ARToolkit aus denen Funktionen zur Bildaufnahme genutzt wurden
- Libdc 1394-1.2.2 und Libraw1394-1.2.0 zur Bildaufnahme über den FireWire-Bus
- UniCap, als mögliche universelle Bibliothek für die Fire-i als auch die Fire-i 620c Kamera
- BobLib bzw. das VS Modul von OpenCV für den Blobdetector

Um die Testprogramme so flexibel wie möglich zu halten, wurden die nötigen Schritte zum optischen Tracking in die Module, wie sie Abbildung 90 zeigt, eingeteilt.

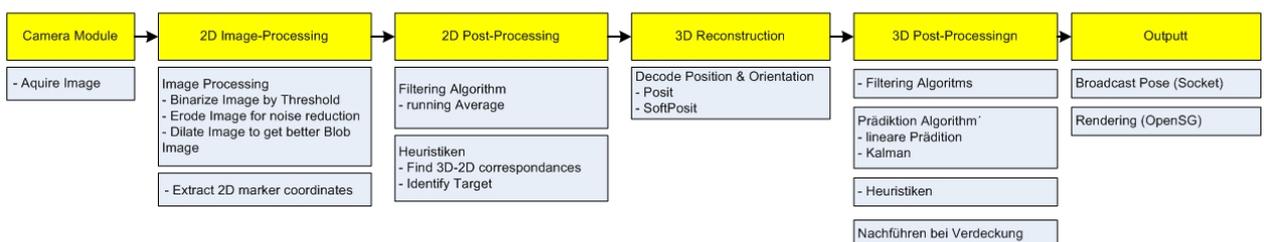


Abbildung 90 Modulansicht der einzelnen Trackingschritte

10.1 Bild-Akquisition

Das akquirieren der Bilddaten über den FireWire-Bus stellt unter Linux ein besonderes Problem dar, da hier keine standardisierte Programmierschnittstelle, wie DirectX unter Windows, vorhanden ist. Die Hauptprobleme lagen vor allem darin den gewünschten Videomodus in der Implementierung auszuwählen. So gelingt es beispielsweise in OpenCV zwar leicht, ein Bild zu aquirieren, aber `cvSetCaptureProperty(...)` Funktionen mit Parametern wie `CV_CAP_PROP_FPS`, mit denen man die Framerate steuert, werden von der Library ignoriert. Die Bildaquire über OpenCV gestaltet sich auch weiterhin problematisch, da über oben genannte Funktion nicht die geforderten 30fps erreicht werden können. Dieses Problem besteht sowohl unter Windows bei der Verwendung von „Video for Windows“ (VfW) als auch unter Linux unter Verwendung der V4l-Schnittstelle.

Im Allgemeinen gibt aber mehrere Optionen ein Bild zu aquirieren. Video4Linux (v4l bzw. v4l2) wird unter Linux als der gebräuchlichste Standard geführt, obwohl er schon sehr alt ist und einige Fehler offenbart. Über v4l können sowohl USB als auch Firewire Kameras angesteuert werden.

Eine andere Möglichkeit bietet die raw1394 Schnittstelle mit der direkt auf den FireWire-Bus zugegriffen werden kann. Das Problem hierbei besteht darin, dass es sich um eine low-level Schnittstelle handelt, bei der direkte Speicherzugriffe (DMA) auf das Gerät selber implementiert werden müssen.

ARToolkit bietet dagegen in seinem Framework ähnlich wie OpenCV eine Methode auf höherer Abstraktionsebene an, mit der Bilddaten direkt aquiriert werden können, ohne sich um die Hardware kümmern zu müssen wie bei libdc1394 und raw1394. Für die Fire-i Kameras wurde ein Teil der ARToolkit Library verwendet. Bei der Fire-i 620c funktionierte diese allerdings nicht, da der entsprechende Videomode nicht unterstützt wird.

Als die derzeit beste Lösung, erscheint die UniCap Library, die aber (entgegen DirectX unter Windows) ebenso nicht zuverlässig alle Modi einer Kamera ansteuern lässt. Trotzdem soll sie hier als Empfehlung gelten.

10.2 Stereoskopisches Tracking – StereoTracker

Zum stereoskopischen Tracking wurden zwei Testimplementierungen erstellt, die dem Modul „Kamera-Kalibrierung“ zugeordnet werden können und somit auch für Methoden des monoskopischen Tracking interessant sind. Beide Testimplementierungen werden nachfolgend kurz beschrieben.

10.2.1 Testimplementierung zur Epipolargeometrie

Abbildung 91 zeigt eine Szene aus zwei verschiedenen Kamerablickwinkeln, wobei einige Objekte sowohl im linken, als auch im rechten Kamerabild zu erkennen sind (beispielsweise der in der Hand gehaltene Marker). Ziel dieser Implementierung war es, die Genauigkeit der Epipolarlinien in vivo zu testen. Die Epipolargeometrie wird benötigt, wenn Korrespondenzen zwischen 2D-Punkten in verschiedenen Koordinatensystemen ermittelt werden sollen.



Abbildung 91 Testimplementierung zur Epipolargeometrie

Vergleicht man die markierten roten Punkte im linken Bild mit denen im Rechten, so sieht man, dass die aus der Fundamentalmatrix berechneten grünen Epipolarlinien diese sehr genau treffen.

Die Berechnung der Epipolarlinien bzw. der Fundamentalmatrix, geschieht über die OpenCV Funktion `cvFindFundamentalMat(points1, points2, fundMatr, CV_FM_RANSAC, 1.0, 0.99, status,` bei der die sog. RANSAC-Algorithmus (Fischler & Bolles 1981) genutzt wird. Auch weitere Minimierungsalgorithmen wie 7-Point, 8-Point und LMedS sind über andere Parameter auswählbar.

Als Eingabeparameter verlangt die Funktion je zwei Arrays mit Punktkoordinaten aus dem linken und rechten Bild. Die Punkte werden in der Testimplementierung per Mausklick auf einen gewissen Punkt im Bild dem Punkttarray hinzugefügt. Dabei ist darauf zu achten, dass die Punkte im ersten Array in der Reihenfolge mit den Punkten im zweiten Array übereinstimmen. Wurde dies gewissenhaft durchgeführt, kann eine Fundamentalmatrix errechnet werden, mit der man aus jedem weiteren Punkt im linken Bild, eine Epipolarlinie für das rechte Bild errechnen lassen kann (siehe grüne Linie in Abbildung 91).

10.2.2 Triangulation

Für die Triangulation sind wie in Kapitel 9.5.1 beschrieben die extrinsischen Kameraparameter notwendig. OpenCV bietet die Möglichkeit diese automatisch über den von Tsai vorgeschlagenen Algorithmus (Tsai 1986) per Schachbrettmuster zu ermitteln. Die Funktion `cv3dTrackerCalibrateCameras (...)` ermöglicht die Position und Rotation mehrerer Kameras gleichzeitig zu ermitteln. Dazu sind lediglich die intrinsischen Parameter der Kameras und die Größe der einzelnen Quadrate des Schachbrettmusters nötig. Abbildung 92 zeigt ein Screenshot aus StereoTracker während der Kamerakalibrierung. Im rechten Bild wurde das gesamte Muster des Schachbrettes bereits erkannt. Links nur die einige Ecken, da nicht das gesamte Brett innerhalb des Kamerabildes zu sehen ist.

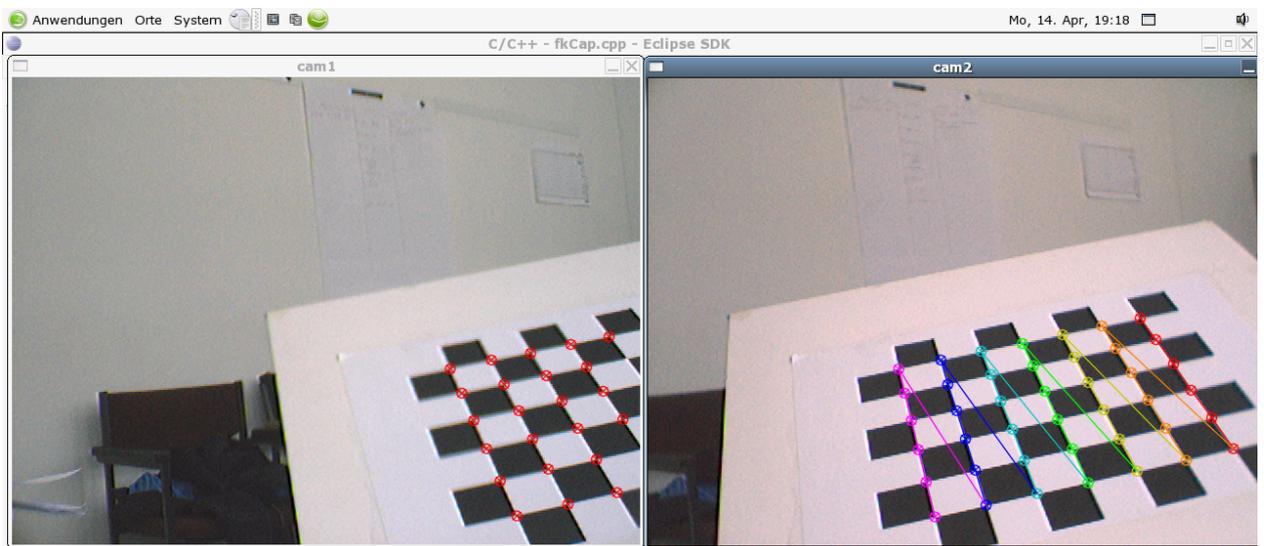


Abbildung 92 Kamerakalibrierung durch Schachbrettmuster

Hat man die extrinsischen Parameter auf diese Weise ermittelt und in einer Struktur gespeichert, kann durch Übergabe der Blobpositionen in die Funktion `cv3dTrackerLocateObjects (...)` die Triangulation durchgeführt werden und der Abstand eines Objektes zur Kamera bestimmt werden. Die für die Triangulation notwendigen Koordinaten des zu trackenden Objektes wurden durch den Meanshift-/Camshift-Algorithmus geliefert.

10.3 Monoskopisches Tracking – Optitrack

Das optische Tracking mittels einer Kamera basiert maßgeblich auf dem von DeMenthon entwickeltem POSIT Algorithmus (DeMenthon & Davis 1995) bzw. SoftPOSIT. Zwei Testimplementierungen werden nachfolgend kurz beschrieben.

10.3.1 POSIT

Der Posit Algorithmus ermittelt in OptiTrack die 6DOF Pose eines bekannten 4-Point-Markers, der am Skizzierwerkzeug angebracht wird. Über die projizierten 3D Koordinaten des Markers in die Bildebene der Kamera kann die Pose in Echtzeit ermittelt werden (>30fps). Abbildung 93 zeigt die Kontrollfenster von Optitrack. Das linke obere Fenster der Applikation zeigt die erkannten Markerkugeln aus dem rechten unteren Fenster. Ein blauer Punkt markiert die Mitte des Blobs. Der für den POSIT-Algorithmus nötige Referenzpunkt wurde gelb gefüllt, um seine Erkennung durch das System leichter zu kontrollieren. Rechts oben wird die aus den 2D-Koordinaten rekonstruierte Pose durch ein Koordinatenkreuz dargestellt.

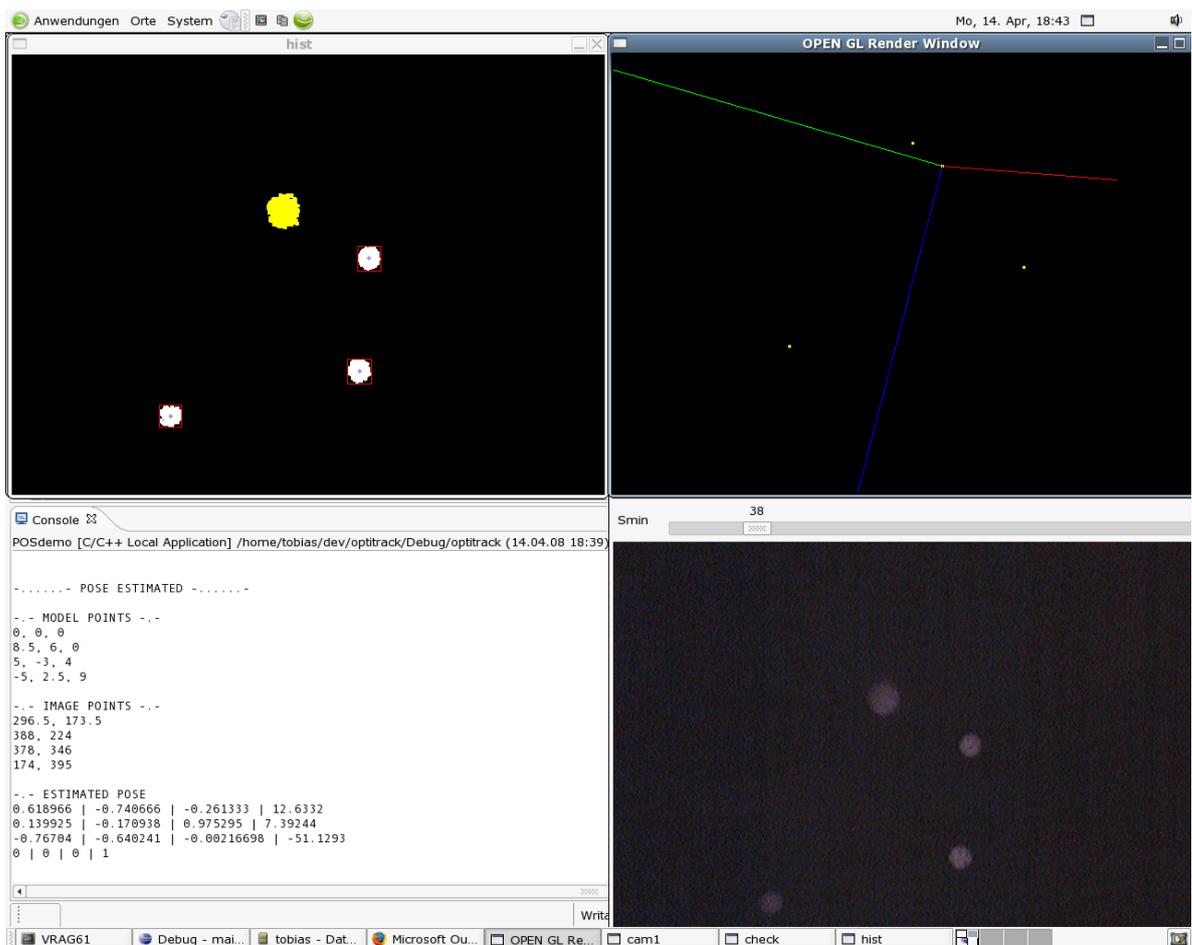


Abbildung 93 OptiTrack Screenshot

10.3.2 SoftPOSIT

Während der Entwicklung wurde offensichtlich, dass der beim POSIT notwendige Referenzpunkt ein Problem darstelle würde. Aus diesem Grund wurde nach Alternativen gesucht, die nicht einen solche Korrespondenz als Funktionseingabeparameter benötigen. Recherchen förderten den SoftPOSIT Algorithmus zu Tage der von DeMenthon sogar als Matlab-Code umgesetzt wurde (DeMenthon 2003).

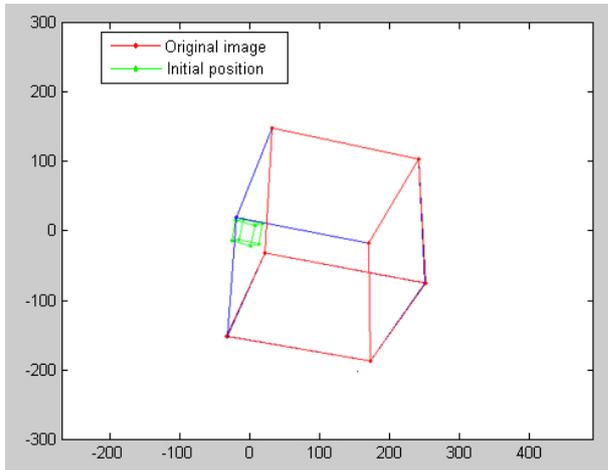


Abbildung 94 SoftPosit - Initiale Pose

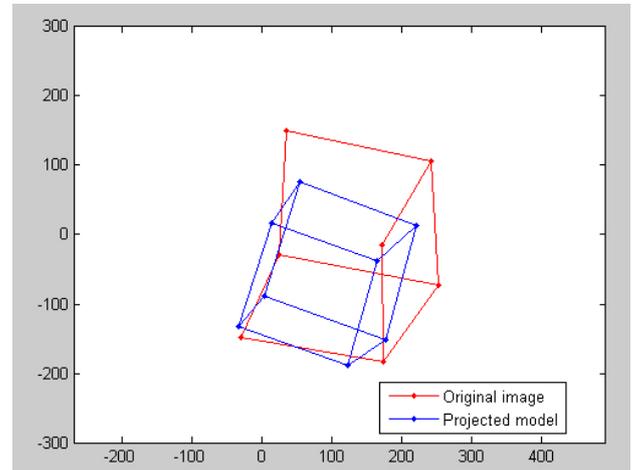


Abbildung 95 SoftPOSIT - 6. Iteration

Um ihn auch für OptiTrack nutzbar zu machen, wurde er in C++ unter der Verwendung der OpenCV Bibliothek nachprogrammiert. Leider stellte sich heraus, dass er für eine Echtzeitanwendung mit einer Konvergenzdauer von 141ms nicht schnell genug ist. Dieser Ansatz wurde im weiteren Verlauf der Entwicklungen also fallen gelassen. Abbildung 96 zeigt die endgültige Rotation und Translation nach 43 Iterationen des Algorithmus. Hier ist festzustellen, dass es geringe Abweichungen zwischen eigener Implementierung und der von DeMenthon nach der vierten Kommastelle gab. Grund hierfür mag ein unterschiedliches Runden der Datentypen bei den unterschiedlichen Compilern zu sein.

```

c:\Dokumente und Einstellungen\realtime\Eigene Dateien\Visual Studio Projects\SoftPosit\Debug\So...
Info: Best Rotatoion Vector so far:
0.972866, 0.0518899, -0.225476, row: 0
-0.172823, 0.810913, -0.559063, row: 1
0.153832, 0.582861, 0.797877, row: 2

Info: Best Translationsvektor so far:
0.498088, row: 0
-0.101357, row: 1
6.99478, row: 2

Info: Best Rotatoion Vector so far:
0.969737, 0.0564272, -0.237543, row: 0
-0.186173, 0.800338, -0.56991, row: 1
0.157957, 0.596887, 0.786623, row: 2

Info: Best Translationsvektor so far:
0.499709, row: 0
-0.100176, row: 1
6.99976, row: 2

Info: Best Rotatoion Vector so far:
0.969271, 0.0566471, -0.239387, row: 0
-0.187711, 0.799271, -0.570903, row: 1
0.158995, 0.598295, 0.785343, row: 2

```

Abbildung 96 SoftPOSIT Test

11 Ergebnisse und Diskussion

Im Rahmen dieser Arbeit wurde erstmalig die Gesamtsituation des dreidimensionalen Skizzierens mittels optischem Tracking in der CAVE im Kontext der virtuellen Produktentwicklung, betrachtet. Eingehende Recherchen zum Thema Problemlösen in der Entwicklung und im Design haben gezeigt, dass innovative Produkte ein kreatives Denken erfordern. Es wurde festgestellt, dass die sekundäre Kreativität sowohl divergente, als auch konvergente Vorgehensweisen innerhalb des Problemlöseprozesses kennt. Somit gibt es also generierende und reduzierende Vorgehensweisen im Designprozess, die im Kerne der Analyse und der Synthese gleichkommen. Der Prozess des Skizzierens führt mit dem Reinterpretationskreislauf einen ähnlichen Mechanismus mit sich, den die Schritte, Analyse und Synthese beinhalten. Dieser Kommunikationsprozess zwischen Designer und Skizze ist Ideen generierend und sollte eine Unterstützung erhalten. Nach einer eingehenden Betrachtung zum Thema Tangible User Interfaces wurde festgestellt, dass hybride Interfaces den Immersionsgrad des Designers in seinem Tun begünstigen können, da durch sie eine stärkere Kopplung zwischen realem Raum und virtueller Welt möglich ist. Das eigens entwickelte Modell verdeutlicht dieses Phänomen. Weiterhin haben die theoretischen Betrachtungen zu einem Anforderungskatalog geführt, den ein optisches Trackingsystem antizipieren sollte. Um festzustellen, ob diese Anforderungen in einem realistischen und praktisch realisierbaren Rahmen gestellt sind, wurden mehrere in vivo Experimente in der CAVE durchgeführt, die durchschnittliche Skizzierbewegungen von 4,39m/s ergeben haben und somit verdeutlichen, dass dreidimensionale Skizzierbewegungen per Kamera getrackt werden können. Die Analyse von 24 Probanden, die je fünf verschiedene Designprobleme lösen sollten, hat ergeben, dass es bevorzugte Skizzierpositionen innerhalb der CAVE gibt und dass die Skizzen durch anatomische Beschränkungen des eigenen Körpers geprägt sind. In wie fern dies positiv für den Designprozess ist, kann derzeit nicht abgeschätzt werden. Denkbar ist, dass aus diesem Phänomen „menschlichere“ oder natürlichere Produkte entstehen.

Um die CAVE als kreativen Interaktionsraum mittels optischem Tracking zu erschließen zu können, wurden mögliche Systemsetups analysiert. Hiernach ist es empfehlenswert, die Kameras in den oberen Ecken der CAVE anzubringen. Die Kameras selbst sollten eine möglichst kleine Brennweite besitzen und im Infrarotbereich arbeiten. Zur Lagebestimmung der Skizzierwerkzeuge sind 4-Punkt-Marker zu empfehlen, bei dem die einzelnen Kugeln einen Durchmesser zwischen 1-2cm aufweisen. Durch ihn kann die Pose des Skizzierwerkzeuges unter Anwendung des POSIT-Algorithmus mit nur einer Kamera in Echtzeit bestimmt werden, wie die Testimplementierung OptiTrack demonstriert.

12 Zukünftige Arbeit und Ausblick

In dieser Arbeit wurde gezeigt, das monoskopisches Tracking dank des POSIT Algorithmus möglich ist. Dennoch gibt es einige Probleme, die für ein robustes Tracking noch zu lösen sind.

Im Bereich des Image Processing bzw. der Blob Extraktion sollten neben den bestehenden Funktionen zusätzliche Post Processing Methoden für die Trajektorien der Blobs implementiert werden. Auf diese Weise können bisher nicht behandelte Probleme der Verdeckung gelöst werden. Lineare Prädiktionsalgorithmen oder der Kalman-Algorithmus sollten an dieser Stelle eine Anwendung finden. Ein zuverlässigeres Erkennen des Referenzpunktes ist hierdurch zu erwarten. Weiterhin sollten andere Methoden der Korrespondenzfindung des Referenzpunktes evaluiert werden. Ansätze hierzu bietet Claudio Rossi in (Rossi, Abderrahim et al. 2005), indem er einen „Extensive Search“ – Algorithmus vorschlägt.

Weiterhin sollte eine Möglichkeit gefunden werden, die es erlaubt mehrere 4-Punkt-Marker voneinander zu unterscheiden. Hierzu sind Eigenschaften des Markers zu finden, die sich unter der einer 3D-2D-Projektion invariant verhalten. Es ist zu prüfen, ob das Quadrangle einen Lösungsansatz dazu bietet.

Neben dem Ansatz des Trackings über den POSIT-Algorithmus, sollte unbedingt gleichzeitig der SoftPOSIT Ansatz weiter verfolgt werden. In den Testimplementierungen hat sich dieser zwar als zu langsam für eine Echtzeitanwendung erwiesen, doch durch eine Optimierung und entsprechende Hardware ist zu erwarten, dass Echtzeitfähigkeit erreicht werden kann. Forschungsergebnisse und Indizien hierfür liefert Diaz in (Diaz & Abderrahim 2007).

Um die Positionsdaten von mehreren Kameras verwenden zu können, ist eine geeignete Methode der sog. „Sensor Fusion“ zu finden. Dies kann durch eigene Programmierung geschehen, oder durch Frameworks, wie sie OpenTracker bietet.

Weiterhin ist zu erwarten, dass im Bereich der Pose Estimation linien- und flächenbasierende Verfahren die punktbasierenden Verfahren ablösen werden. Weitere Informationen hierzu sind unter (Phong, Houraud et al. 1995; Rosenhahn, Perwass et al. 2005) nach zu lesen.

In Bezug auf die Interaktion zwischen Mensch und Computern ist festzustellen, dass Kameras als Sensor des Computers an Relevanz gewinnen. Eingabe-, Berechnungs- und Ausgabetechnologien werden zudem aus Sicht des Anwenders miteinander verschmelzen, und eine einheitliche Erfahrung zu schaffen. Die Vision des Ubiquitous Computing wird somit zur Realität.

13 Schlusswort

Stereoskopische Tracking-Systeme haben seit einigen Jahren die Marktreife erlangt und werden in verschiedenen Arbeitsbereichen erfolgreich eingesetzt. Im Bereich der kreativen Unterstützung dagegen, gibt es derzeit keine Lösungen außerhalb der Forschungslandschaft. Hauptursache dafür mögen die hohen Hardware-Kosten solcher Systeme sein - dabei gibt es Wege diese zu senken.

Diese Arbeit hat bewiesen, dass monoskopische Pose Estimation für das Skizzieren im immersiven Raum praktisch umsetzbar ist und sogar einige Vorteile gegenüber dem stereoskopischen Tracking besitzt. So kann der Einsatz der zu verwendenden Kameras beispielsweise halbiert und dennoch eine angemessene Genauigkeit beibehalten werden. Mit dem Einsatz von nur vier Kameras kann der Interaktionsraum einer CAVE ausreichend erschlossen werden. Multipunktmarker verhelfen dabei das System so universell wie möglich zu halten, indem sie z.B. an verschiedenen Skizzierwerkzeugen angebracht werden können. Zudem bietet ein monoskopisches Kamera-System einen geringeren Störfaktor innerhalb der CAVE als dies ein stereoskopisches System bieten würde. Die Genauigkeit ist dabei größer, als die eines magnetischen Trackingsystem mit dem Effekt, dass die Kopplung zwischen Mensch und Computer verbessert werden kann. Die Möglichkeit Display und Interface durch das Paradigma des hybriden Skizzierwerkzeugs miteinander zu verschmelzen, birgt dabei das Potential, noch tiefer in den virtuellen Raum einzutauchen und mentale Modelle, durch Ausdruck des eigenen Körpers ohne Umweg über verzögernde Eingabeschnittstellen, wie Maus oder Tastatur, zu externalisieren, dokumentieren und kommunizieren.

Das in dieser Arbeit analysierte und ergänzte Skizziersystem bietet kreative Unterstützung im Entwurfsprozess und hat das Potential innovativere Produkte entstehen zu lassen.

APPENDIX A

Zur Ermittlung der durchschnittlichen Handgeschwindigkeiten, wurden die Skizzierbewegungen einer Testperson aufgezeichnet. Folgende Grafiken zeigen das skizzierte Objekt.

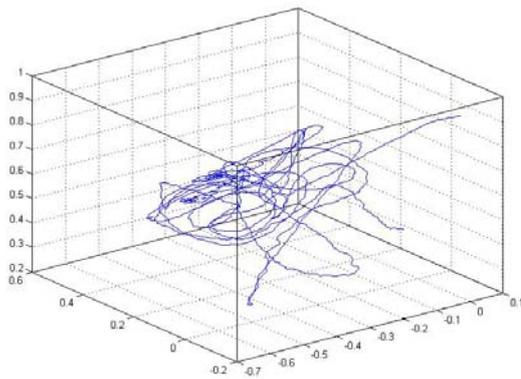


Abbildung 97 Skizze im Stehen

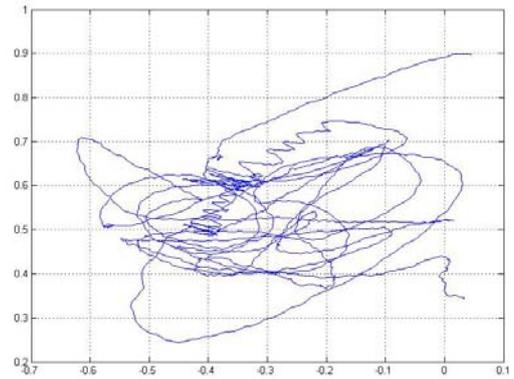


Abbildung 98 Skizze im Stehen - XZ-Ebene

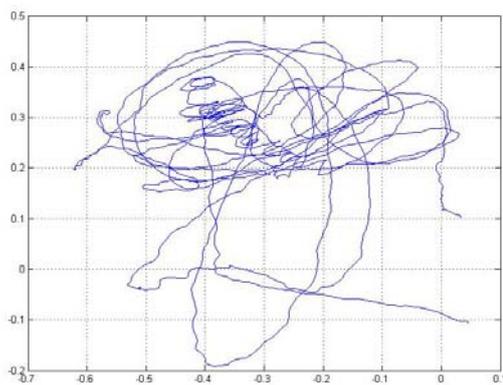


Abbildung 99 Skizze im Stehen - XY-Ebene

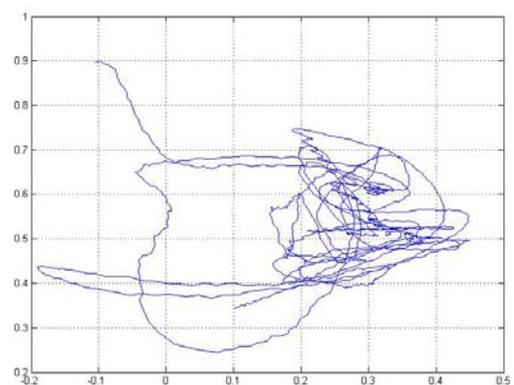
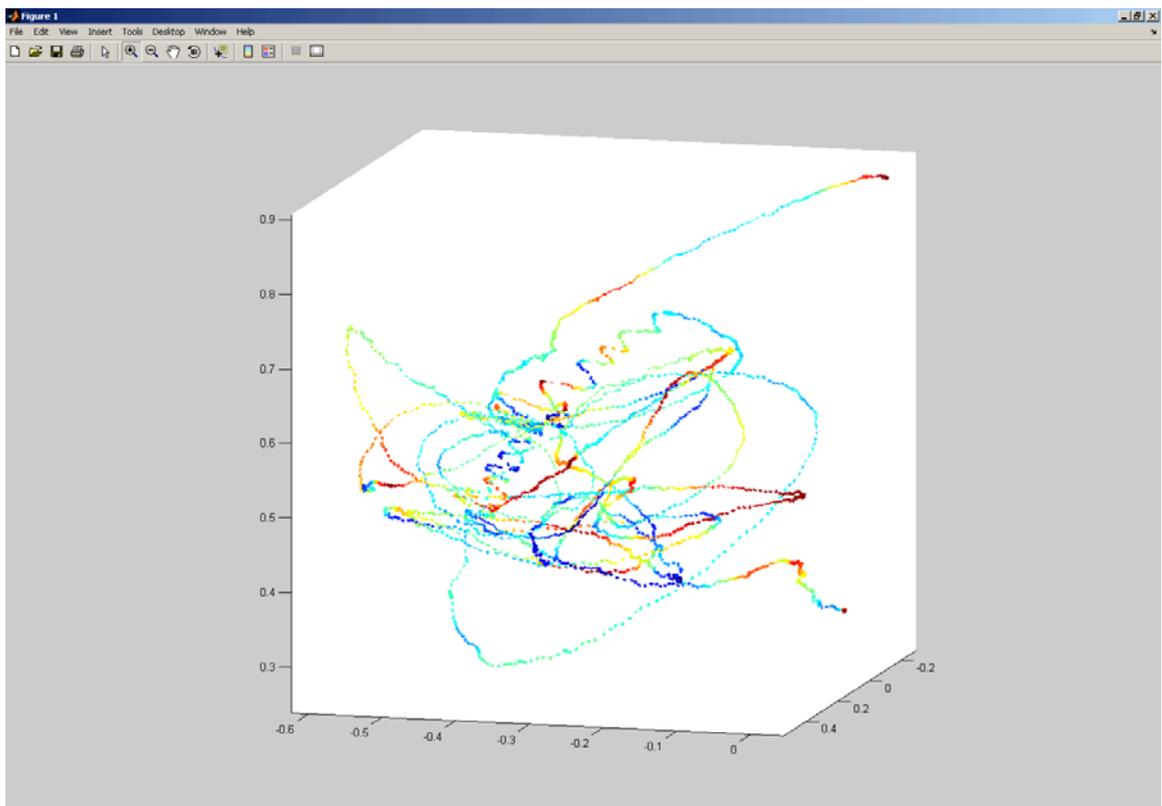
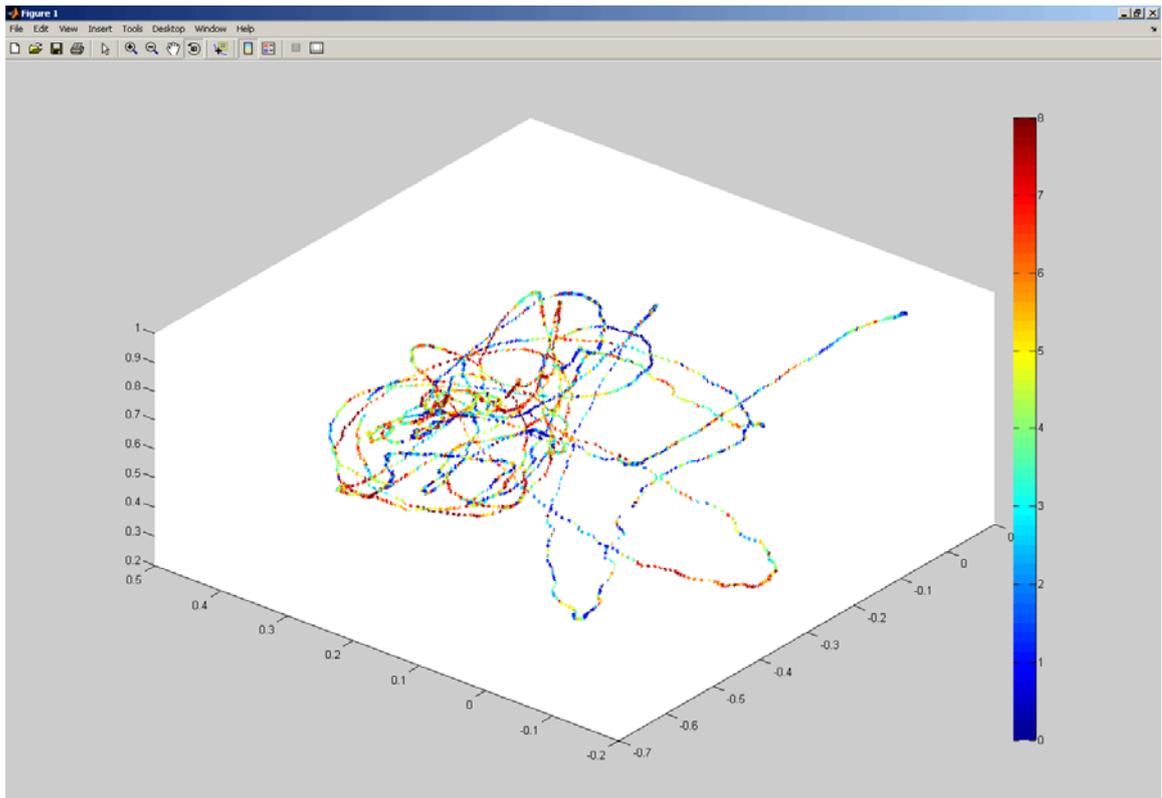
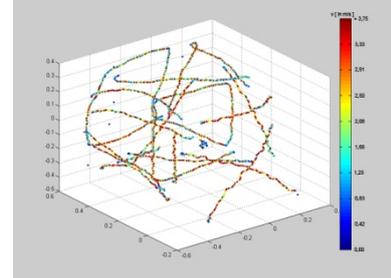
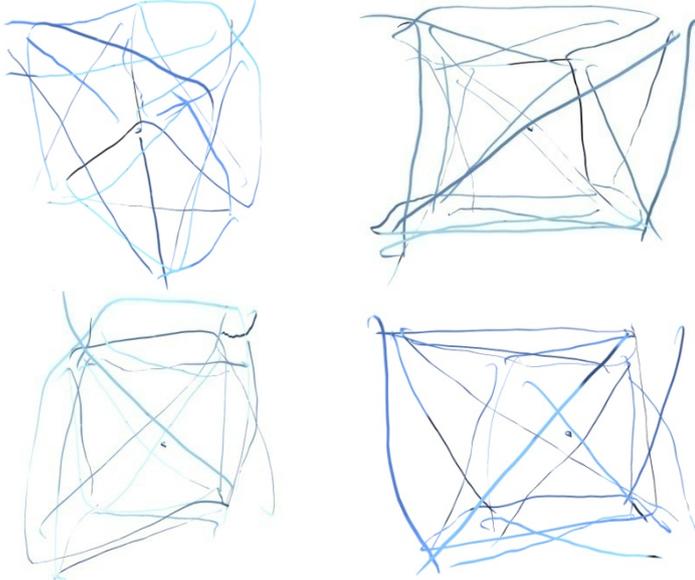


Abbildung 100 Skizze im Stehen - YZ-Ebene



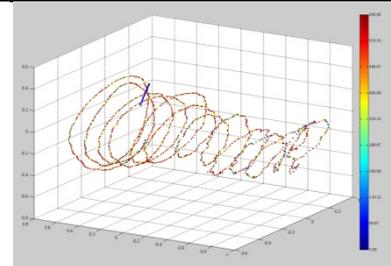
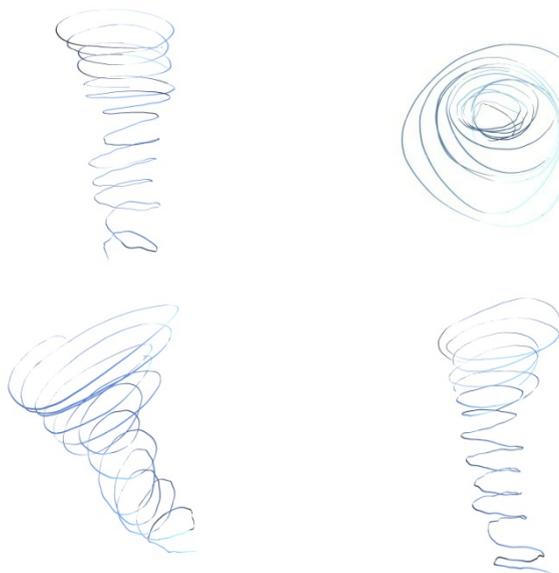
APPENDIX B

3D Skizzen - Geometrische Körper



Die 3D-Skizze zeigt einen Quader mit eingezeichneten Diagonalen.

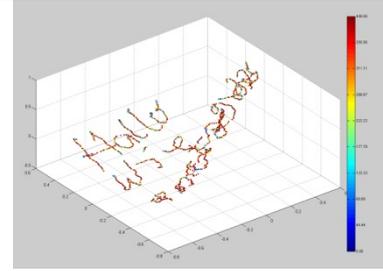
Spirale



Die Spirale wurde von der CAVE-Decke zum Boden hin gezeichnet, wo sie sich verjüngt. In der Analyse der Geschwindigkeiten fällt auf, dass der obere Teil der Spirale schneller gezeichnet wurde, als der Untere.

3D Skizze - Handschrift

Hallo
Wie gehts
euch?



Hallo
Wie gehts
euch?

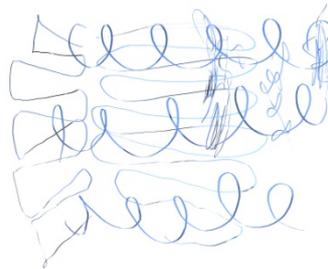


Diese Skizze zeigt den Schriftzug:
„Hallo wie geht’s euch!“
In der Seitenansicht ist deutlich
zu sehen, dass der Schriftzug
anatomisch geprägt ist. Der
Proband zog es vor nicht in die
Knie zu gehen, um einen
planaren Schriftzug zu schreiben.

3D Skizze - Kurven

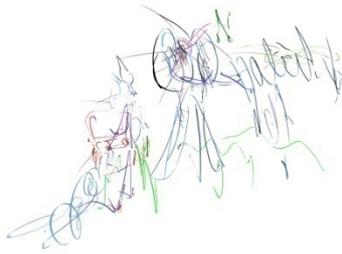


Die Skizze zeigt verschiedene
Formen von regelmäßigen
Kurven parallel zu den CAVE-
Wänden. Hier viel bei der
Analyse auf, dass Geraden
schneller gezeichnet werden als
Kurven, und dass vor Kurven
meist die Handgeschwindigkeit
absinkt.



Spiralen dagegen haben meist
eine sehr hohe Geschwindigkeit.

3D Skizze von einem Amateur



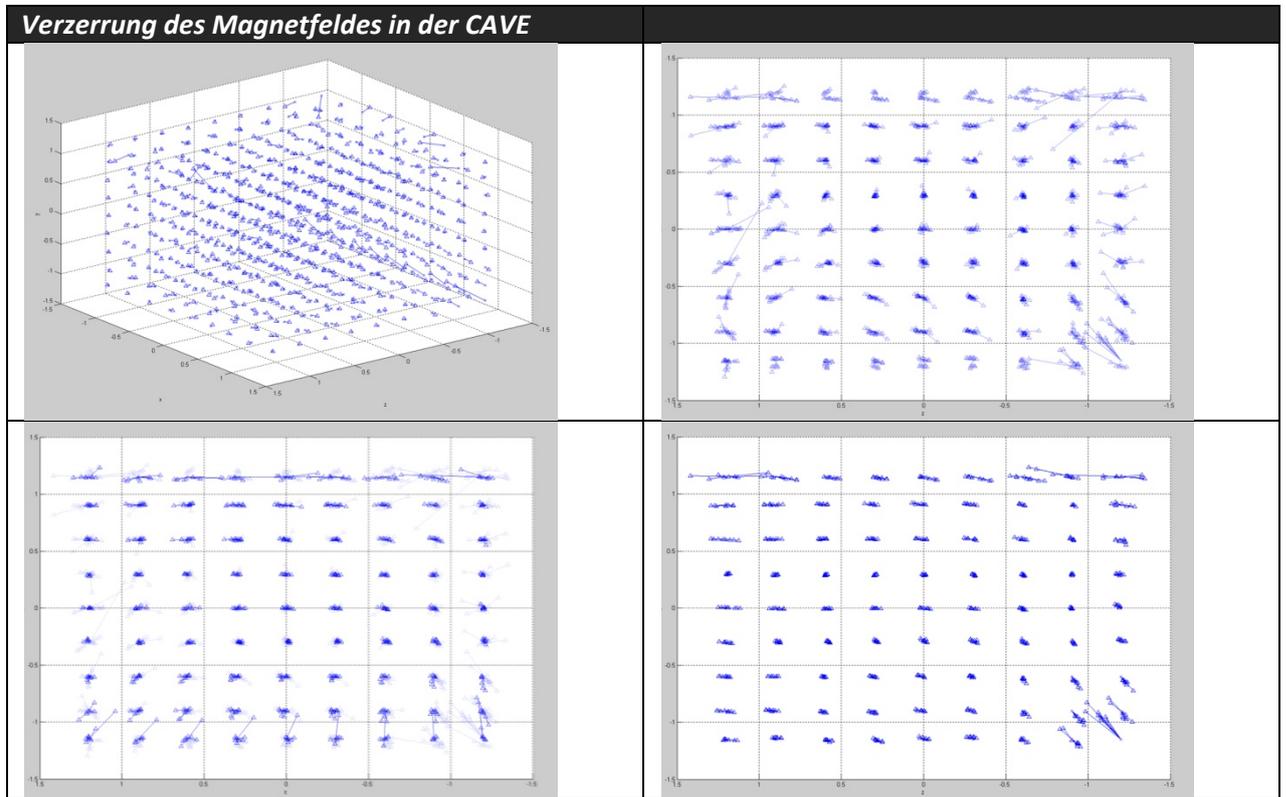
Diese Skizze zeigt verschiedene Elemente auf. Haus, Sonne, Auto, Baum, Person und ein Schriftzug wurde von einem Amateur skizziert.



Es fällt auf, dass die Szene in 2D-Ebenen aufgebaut ist einzig das Auto wurde als 3D-Objekt skizziert.

APPENDIX C

Die Diagramme zeigen tatsächliche und durch den magnetischen Tracker gemessene Positionsdaten und vergleicht diese.

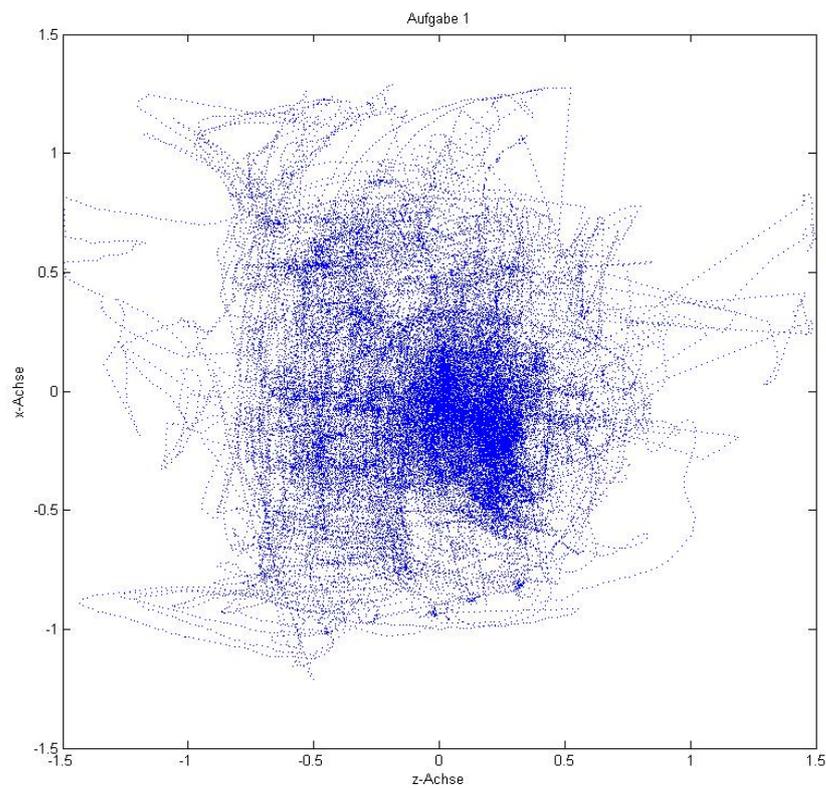
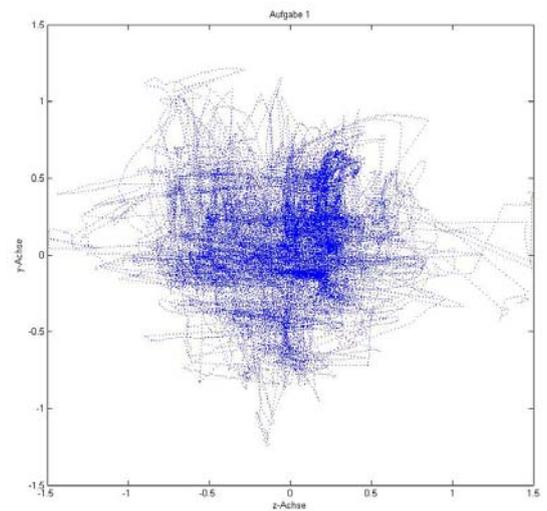
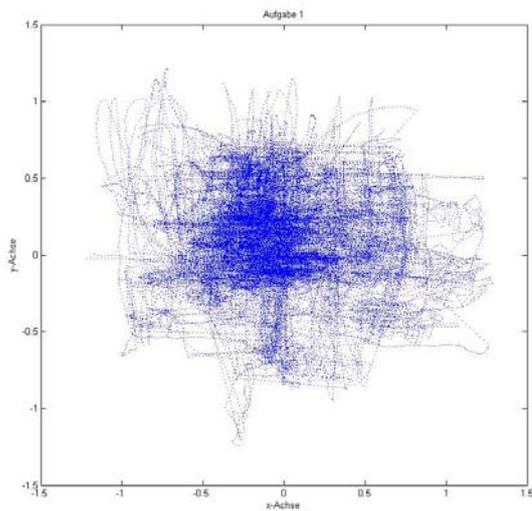


X = tatsächlicher Punkt, Dreieck = gemessener Punkt

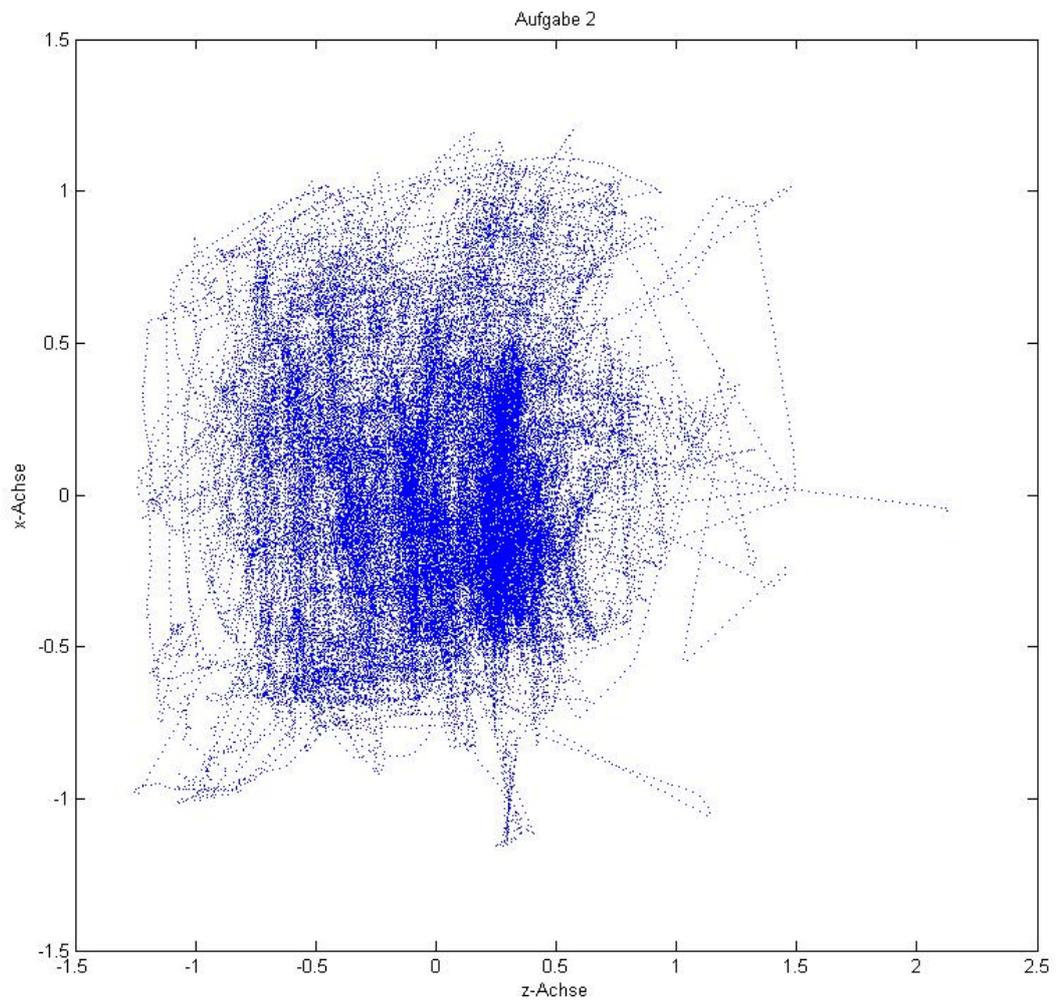
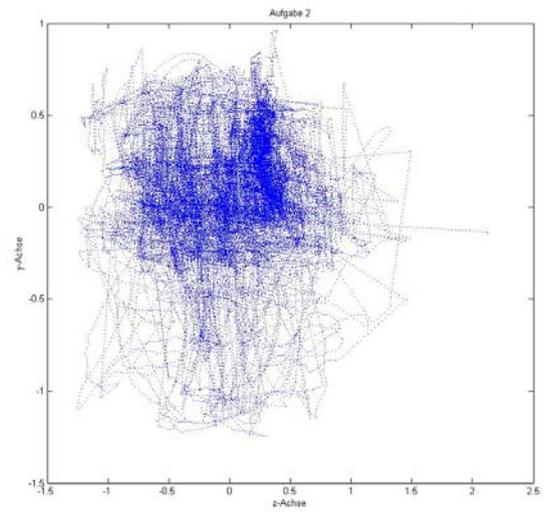
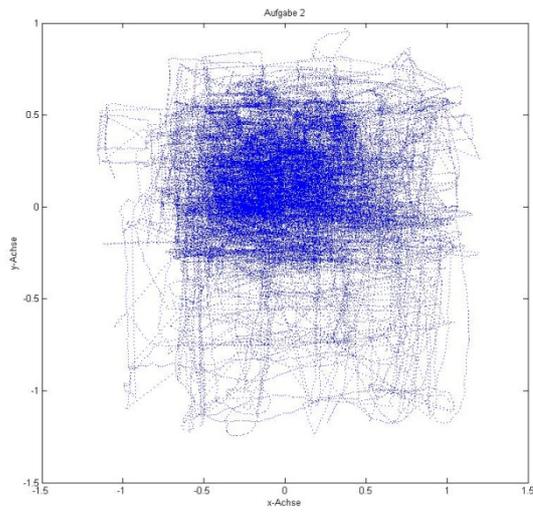
APPENDIX D

Die Punktwolken kombinieren Skizzen von 24 Probanden die fünf verschiedene Designlösungen in der CAVE anfertigen sollten.

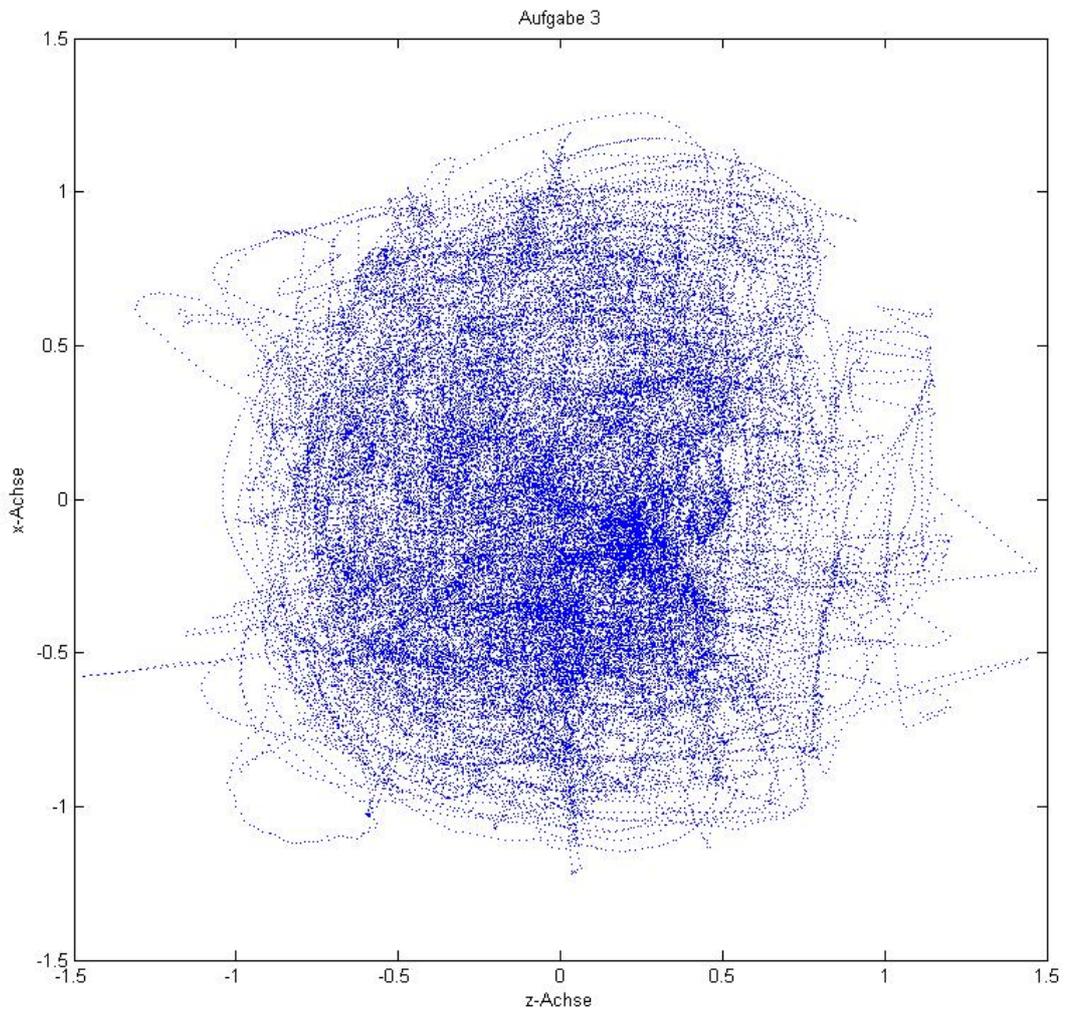
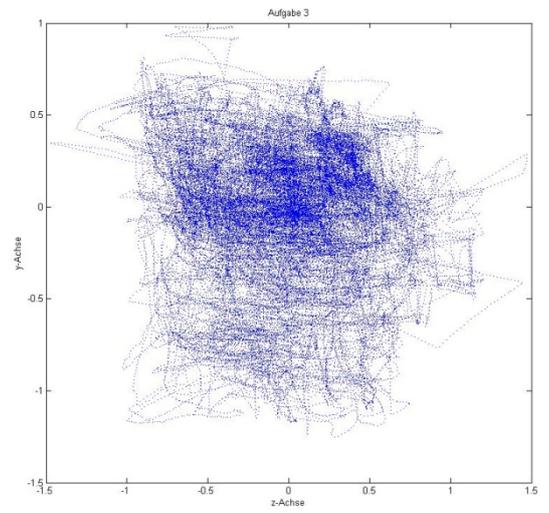
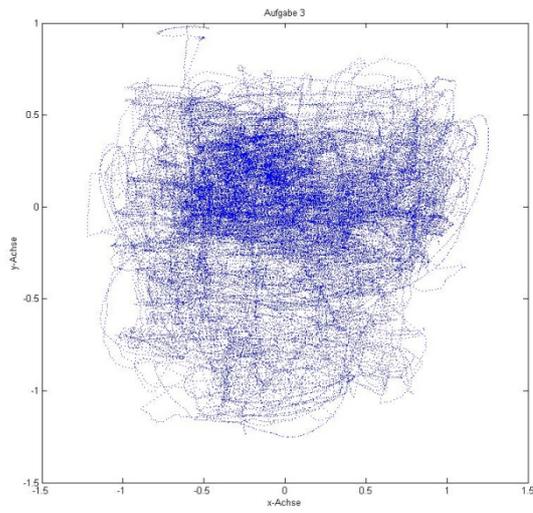
Aufgabe 1 - Hängetisch



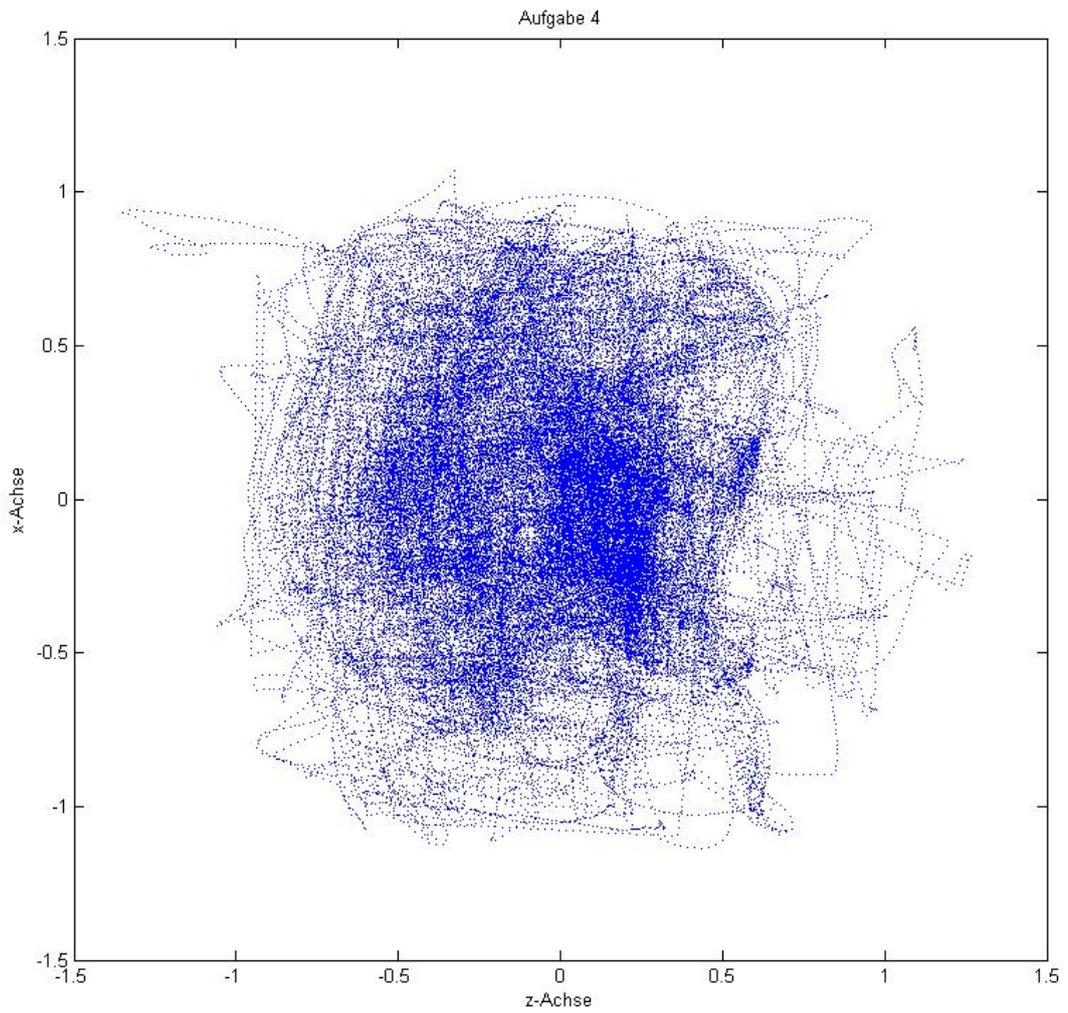
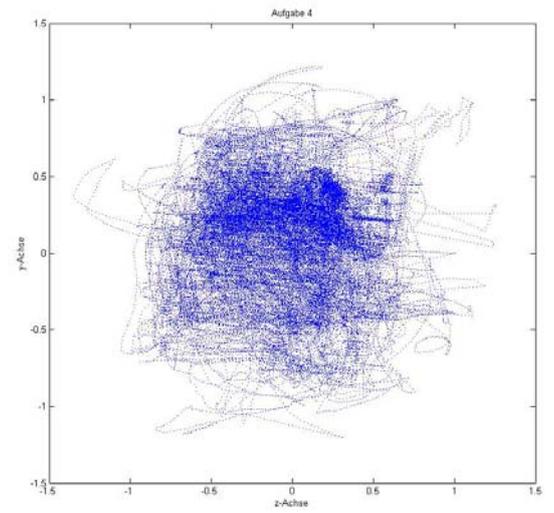
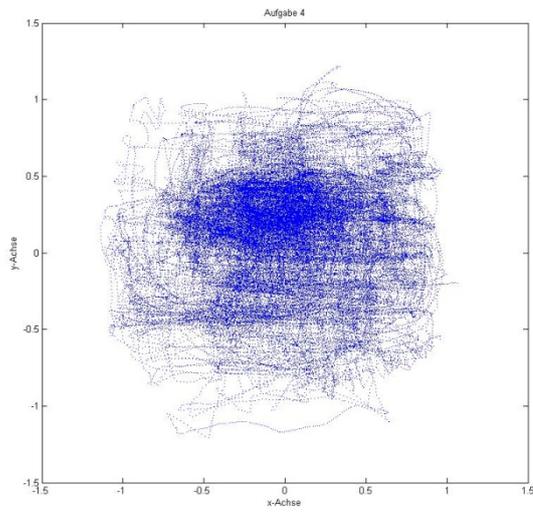
Aufgabe 2 – Innovative Bar



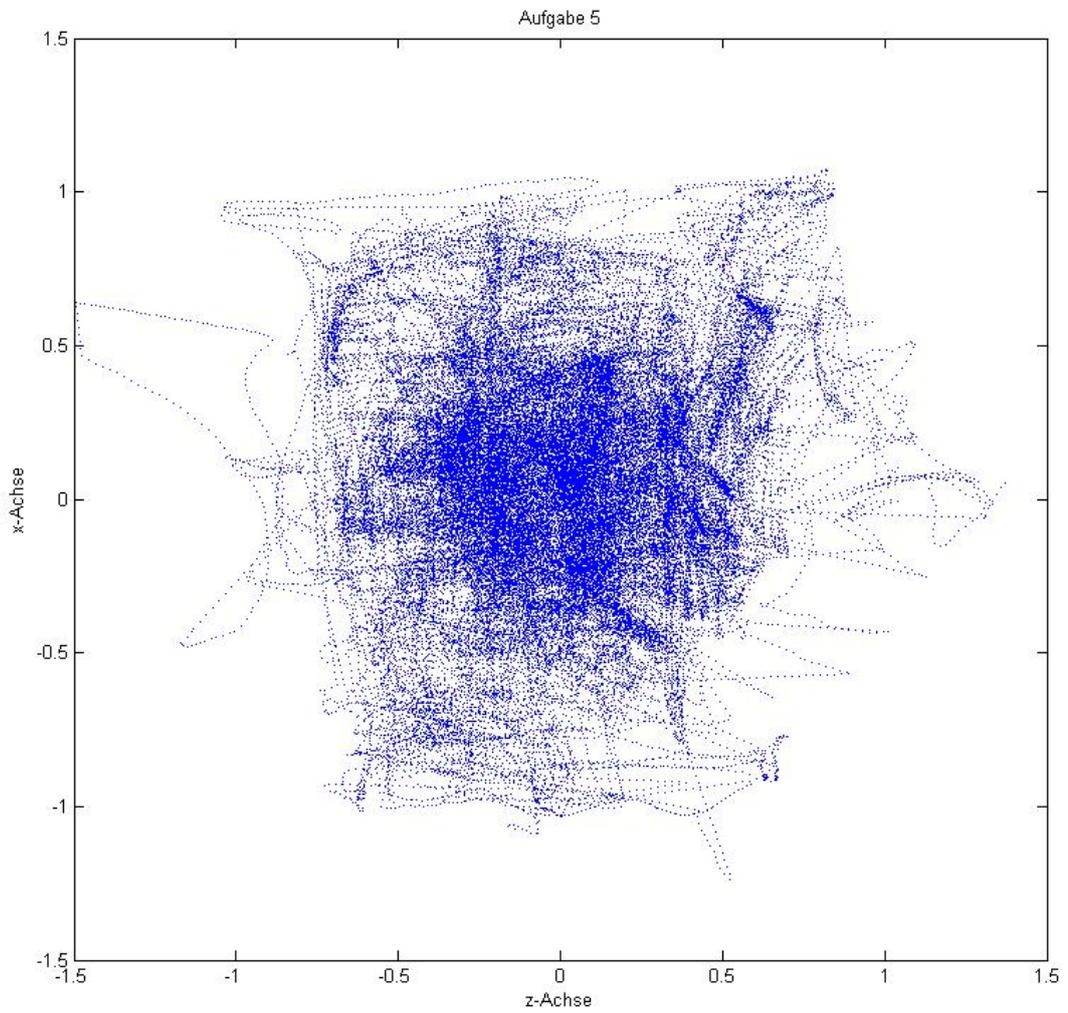
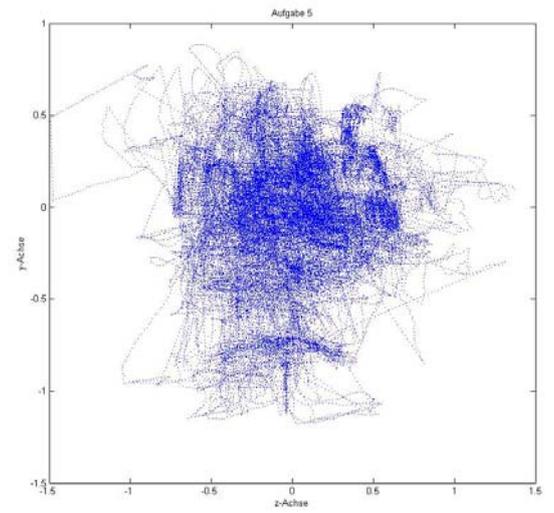
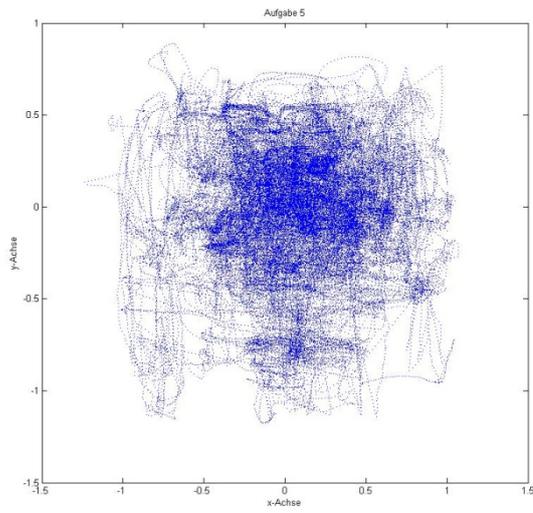
Aufgabe 3 – Sitzbereich mit Couch und Tisch



Aufgabe 4 - Schlafkapsel



Aufgabe 5 - Massagstuhl



APPENDIX E

Dieser Abschnitt stellt die Grundzüge der geometrischen Projektion dar, da sie eine wesentliche Rolle bei der Pose Estimation spielen. Vor allem die orthographische und perspektivische Projektion sind Ausgangspunkt weiterer mathematischer bzw. algorithmischer Überlegungen.

Orthographische Projektion

Bei der orthographischen (ortho- griechisch für Senkrecht) Projektion liegt die Bildebene der Kamera immer parallel zu einer der Koordinatenachse aus dem zu projizierenden Koordinatensystem. Um einen 3D-Punkt a_x, a_y, a_z auf einen 2D-Punkt b_x, b_y mit Hilfe der orthographischen Projektion parallel der Y-Achse zu projizieren, können folgende Gleichungen benutzt werden:

$$b_x = s_x a_x + c_x; \quad b_y = s_z a_z + c_z$$

Wobei der Vektor \vec{s} einen beliebigen Skalierungsfaktor darstellt, und \vec{c} ein beliebig großes Offset ist. Diese beiden Konstanten sind optional und positionieren im Grunde den Darstellungsbereich.

Die Projektion an sich, kann in Matrixnotation wie folgt angegeben werden (sie führt zusätzlich den Hilfsvektor \vec{d} zur Verdeutlichung ein):

$$\begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$$
$$\begin{bmatrix} b_x \\ b_y \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_z \end{bmatrix} \begin{bmatrix} d_x \\ d_y \end{bmatrix} + \begin{bmatrix} c_x \\ c_z \end{bmatrix}$$

Während das orthographisch projizierte Bild zwar die dreidimensionale Erscheinung des Objektes darstellt, zeigt es allerdings nicht, wie das Objekt aussehen würde, wenn es von einer Kamera aufgenommen oder von einem Beobachter gesehen würde. Im speziellen hat dies den Effekt, dass die Linien (von Objekten) in orthographisch projizierten Bildern immer gleich lang sind, egal wie weit oder wie nah sie am Beobachter bzw. der Bildebene der Kamera sind. Das Resultat ist, dass nahe Linien verkürzt erscheinen.

Perspektivische Projektion

Die perspektivische Projektion erfordert mehrere Definitionen.

$a_{x,y,z}$ = Der Punkt im RR der projiziert werden soll

$c_{x,y,z}$ = Die Position der Kamera

$\theta_{x,y,z}$ = Die Rotation der Kamera

$e_{x,y,z}$ = Die Position des Betrachters im in der Bildebene

Alle diese Parameter werden benötigt, um $b_{x,y}$ in der 2D-Projektion aus $a_{x,y,z}$ zu berechnen.

Zunächst wird ein Punkt $d_{x,y,z}$ als Translation eines Punktes $a_{x,y,z}$ in das Koordinatensystem der Kamera

$c_{x,y,z}$ definiert. Dies kann erreicht werden, indem c von a subtrahiert wird und die Rotationsmatrizen

hinzu multipliziert werden.

$$\begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos-\theta_x & \sin-\theta_x \\ 0 & -\sin-\theta_x & \cos-\theta_x \end{bmatrix} \begin{bmatrix} \cos-\theta_y & 0 & -\sin-\theta_y \\ 0 & 1 & 0 \\ \sin-\theta_y & 0 & \cos-\theta_y \end{bmatrix} \begin{bmatrix} \cos-\theta_z & \sin-\theta_z & 0 \\ -\sin-\theta_z & \cos-\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} - \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} \right)$$

Der transformierte Punkt kann dann mit folgender Formel in die 2D-Bildebene (xy-Ebene) projiziert werden.

$$b_x = (d_x - e_x) \left(\frac{e_z}{d_z} \right); \quad b_y = (d_y - e_y) \left(\frac{e_z}{d_z} \right)$$

Literaturverzeichnis

- 1 Alexander Bornik, R. B., Ernst Kruijff, Bernhard Reitinger, Dieter Schmalstieg (2006). A Hybrid User Interface for Manipulation of Volumetric Medical Data. In: *Proc. Proceedings of the 3D User Interfaces (3DUI)*, IEEE Computer Society. 29-36.
- 2 Amabile, T. M. (1998). How to kill creativity. *Harvard Business Review* 76(5).
- 3 Amit Konar, L. J. (2005). *Cognitive Engineering - A Distributed Approach to Machine Intelligence*. London, Springer Verlag.
- 4 Andrew I. Comport, E. M., Muriel Pressigout, Francois Chaumette (2006). Real-Time Markerless Tracking for Augmented Reality: the Virtual Servoing Framework. *IEEE Transactions on Visualization and Computer Graphics* 12(4): 615-628.
- 5 ARToolkit. (2006). *ARToolkit*. Retrieved 13. April 2008, 2008, from <http://www.hitl.washington.edu/artoolkit/>.
- 6 Azuma, R. (1995). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments* 6(4): 355-385.
- 7 B. Ullmer, H. I. (1997). The metadesk: Models and prototypes for tangible user interfaces. . In: *Proc. Proceedings of the 1997 ACM Symposium on User Interface Software and Technology (UIST'97)*, ACM Press. 223-232.
- 8 Beaudouin-Lafon, M. & W. Mackay (2002). Prototyping Tools and Techniques. 1007-1029.
- 9 Billingham, M., H. Kato & I. Poupyrev (2001). The MagicBook - Moving Seamlessly between Reality and Virtuality. *IEEE Computer Graphics and Applications* 21(3): 6-8.
- 10 Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*, Springer Science+Business media, LLC.
- 11 Bowman, D. A., E. Kruijff, J. J. LaViola & I. Poupyrev (2005). *3D User Interfaces: Theory and Practice*, Addison Wesley.
- 12 Brygg Ullmer, H. I. (2000). Emerging frameworks for tangible user interfaces. *IBM Systems Journal* 39(3/4): 915-931.
- 13 Buxton, B. (1997). *Living in Augmented Reality: Ubiquitous Media and Reactive Environments*. Hillsdale, N.J., Erlbaum.
- 14 Buxton, B. (2007). *Sketching User Experiences: Getting the Design right and the Right Design*, Morgan Kaufmann.
- 15 Colin Ware, D. J. (1988). Using the bat: a six-dimensional mouse for object placement. *IEEE Computer Graphics and Applications* 8(6): 65-70.

- 16 Couger, D. (1996). Creativity & Innovation. *Information Systems Organisations*.
- 17 Couger, J. D. (1995). Developing information systems with creativity techniques: an exploratory study. In: *Proc. 28th Hawaii International Conference on System Sciences*, Hawaii, IEEE. 720-728.
- 18 David Merrill, J. K., Pattie Maes (2007). Siftables: Towards Densor Network User Interfaces. In: *Proc. Tangible and Embedded Interaction (TEI'07)*, Baton Rouge, Louisiana, USA, ACM Press.
- 19 DeMenthon, D. F. (2003). *Matlab SoftPOSIT Demonstration Code*. Retrieved 04. March 2008, 2008, from www.cfar.umd.edu/~daniel/SoftPOSIT.txt.
- 20 DeMenthon, D. F. & L. S. Davis (1995). Model-Based Object Pose in 25 Lines of Code. *International Journal of Computer Vision* 15(1-2): 131-141.
- 21 Diaz, J. C. & M. Abderrahim (2007). Modified SoftPOSIT algorithm for 3D visual tracking. *IEEE*.
- 22 Donald.A. Norman, S. W. D. (1986). *User Centered System Design: New Perspectives in Human-computer Interaction*, CRC.
- 23 Dörner, D. (1979). *Problemlösen als Informationsverarbeitung*. Stuttgart, Kohlhammer.
- 24 Fernandez, M. E. L., A. B. Paposo & M. Gattass (2007). *A Novel Optical Trackin Algorithm for Point-Based Projective Invariant Marker Patterns*. Rio de Janeiro, Dissertation, Pontificia Universidade Catolica do Rio de Janeiro.
- 25 Fischler, M. A. & R. C. Bolles (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Autonates Cartography. *Communications of the ACM* 24(6): 381-395.
- 26 Fitt, P. M. (1954). The information capacity of human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47(6): 381-391.
- 27 Fitzmaurice, G. (1997). *Graspable User Interfaces*. Toronto, Dissertation, University of Toronto.
- 28 FreeTrack. (2008). *FreeTrack Software*. Retrieved 04. April, 2008, from www.free-track.net.
- 29 G. Pangaro, D. M.-A., H. Ishii (2002). The actuates workbench: computer-controlled actuation in tabletop tangible interfaces. In: *Proc. Proc. UIST 2002*, ACM Press. 181-190.
- 30 G. W. Fitzmaurice, H. I., B. Buxton (1995). Laying the foundations for graspable user interfaces. In: *Proc. CHI'05*, ACM Press. 422-449.
- 31 Grembowietz, S. (2005). *3D Pose Estimation*. Munich, Technische Universität München.
- 32 Grigore Burdea, P. C. (1994). *Virtual Reality Technology*, John Wiley & Sons Inc.
- 33 Groher, M. (2008). *Pose Estimation POSIT SoftPOSIT*. Berlin.
- 34 Heap, J. (1989). *The management of innovation and design*. London, Cassell.

- 35 Herbon, A. & M. Rötting (2007). Detection and Processing of Visual Information in Three-Dimensional Space. *MMI Interaktiv, Human-Machine Interaction in Virtual Environments* 12: 18-26.
- 36 Hiroshi Ishii, B. U. (1997). Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. *Proc. of CHI'97*.
- 37 Hoffmann, J., U. Hahne, S. Knödel & G. Wetzstein. (2005). *Optical Tracking - Dokumentation zum Projekt*. Retrieved 20.03.2008, 2008, from www.cs.ubc.ca/~wetzste1/projects/optotrack/.
- 38 Holm-Hadulla, R. (2005). *Kreativität - Konzept und Lebensstil*, Vandenhoeck & Ruprecht, Göttingen.
- 39 Hornecker, E. (2002). Understanding the Benefits of Graspable Interfaces for Cooperative Use. In: *Proc. Coop'02*.
- 40 I. Lopez, A. F., B. van der Wijst, H. Broers (2002). *Pose estimation from 2D to 3D for computer vision in an assembly node*. Eindhoven, Philips Centre for Industrial Technology Eindhoven.
- 41 ImagingSource, T. (2006). *Objektive - Auswahl und Montage*. Retrieved 20.03.2008, 2008, from <http://theimagingsource.com/de/resources/whitepapers/download/choosinglenswp.en.pdf>.
- 42 Israel, J. H., R. Korkot & P. T. Fischer (2008). Investigating User Requirements and Usability of Immersive Three-dimensional Sketching for Early Conceptual Design - Results from Expert Discussions and User Studies. In: *Proc. Eurographics Workshop on Sketch-Based Interfaces and Modeling*. submitted.
- 43 Israel, J. H. & E. Zacharias (2007). Möglichkeiten und Grenzen des Skizzierens im Dreidimensionalen Raum aus Designersicht. In: *Proc. 7. Berliner Werkstatt Mensch-Maschine-Systeme: Prospektive Gestaltung von Mensch-Technik-Interaktion*, Berlin-Brandenburgische Akademie der Wissenschaft, VDI Verlag GmbH Düsseldorf.
- 44 J. Underkoffler, H. I. (1999). Urp: a luminous-tangible workbench for urban planning and design. In: *Proc. Proc. CHI*, ACM Press. 386-393.
- 45 Jordâ, S. (2003). Interactive Music Systems for Everyone: Exploring visual Feedback as a way for Creating more Intuitive, Efficient and Learnable Instruments. In: *Proc. Music Acoustics Conference (SMAC'03)*, Stockholm.
- 46 Joshua A. Staples, T. A. D. (2006). *Motion Capture for the Masses*, Department of Computer Science, Clemson University, USA.
- 47 K. Ullrich, S. E. (1995). *Product Design and Development*, McGraw-Hill.
- 48 Klauer, K. C. (1995). *Grundlagen der Problemlöseforschung*. Göttingen, Verlag für Angewandte Psychologie.

- 49 Knieß, M. (2005). *Kreativitätstechniken - Methoden un Übungen*, Beck im dtv.
- 50 Korkot, R. (2008). *Die Entwicklung von Tangible User Interfaces für immersive Umgebungen*. Berlin, Dissertation, Technische Fachhochschule Berlin.
- 51 Krause, F.-L., J. H. Israel, J. Neumann & T. Feldmann-Wüstenfeld (2007). Usability of Hybrid, Physical and virtual Objects for Basic Manupulation Tasks in Virtual Environments. *3d User Interfaces*.
- 52 Laseau, P. (1980). *Graphic Thinking for Architects and Designers*. New York, Van Nostrand Reinhold Company.
- 53 Lepetit, V. & P. Fua (2005). Monocular Model-Based 3D Tracking of Rigid Objects: A Survey. *Foundations an Trends in Computer Graphics and Vision* 1(1): 1-89.
- 54 Liere, R. v. & J. D. Mulder (2003). *Optical Tracking Using Projective Invariant Marker Pattern Properties*, IEEE Computer Society.
- 55 Lowe, D. (2004). Distinctive Image Features from Scale-Invariant Key-Points. *International Journal of Computer Vision* 60(2): 91-110.
- 56 Mariottini, G. L. & D. Prattichizzo *The Epipolar Geometry Toolbox: multiple view geometry and visual servoing for MATLAB*, Dipartimento di Ingegneria dell'Informazione, Universit`a di Siena.
- 57 Mathieu, H. (2005). A 6DOF Optical Tracker Based on a Single Camera. In: *Proc. INTUITION 2005*, Senlis France.
- 58 May, M. (2006). *Raum*. Göttingen, Hogrefe.
- 59 Neumann, U., S. You, Y. Cho, J. Lee & J. Park (1999). Augmented Reality Tracking in Natural Environments. In: *Proc. Proceedings of the International Symposium of Mixed Realities*.
- 60 Norman, D. A. (1999). *The Invisible Computer*. London, MIT Press.
- 61 Norman, D. A. & S. W. Draper (1986). *User Centered System Design: New Perspectives in Human-computer Interaction*, CRC.
- 62 OpenCV. (2006). *OpenCV reference guide*. Retrieved 12.April 2008, 2008, from <http://sourceforge.net/projects/opencvlibrary/>.
- 63 Pahl, G. & W. Beitz (1986). *Konstruktionslehre*. Berlin - Heidelberg - New York - London - Paris - Tokyo, Springer-Verlag.
- 64 Paul Hekkert, M. v. D. (2003). Designing in Context: Foundations and Applications of the ViP approach. In: *Proc. Proceedings of Design thinking Research Symposium 5.*, Delft: DUP Science.
- 65 Pedro Santos, A. S., Alexandre Buaes, Joaquim Jorge (2006). PTrack: Introducing a Novel Iterative Geometric Pose Estimation for a Marker-based Single Camera Tracking System. In: *Proc. IEEE Virtual Reality Conference 2006 (VR'06)*, Alexandria, Virginia, USA, IEEE.

- 66 Philip David, D. D., Ramani Duraiswami, Hanan Samet (2004). SoftPOSIT: Simultaneous Pose and Correspondence Determination. *Int. J. Comput. Vision* 59(3): 259-284.
- 67 Phong, T. Q., R. Houraud, A. Yassine & P. D. Tao (1995). Object Pose from 2-D and 3-D Point and Line Correspondences. *International Journal of Computer Vision* 15(3): 225-243.
- 68 Polya, G. (1945). *How to solve it*. Princeton, PrincetonUniversity Press.
- 69 Previc, F. H. (1998). The neuropsychology of 3D Space. *Psychological Bulletin* 124(2): 123-164.
- 70 Pugh, S. (1990). *Total Design: Integrated Methods for Successful Product Engineering*, Addison-Wesley.
- 71 Pustka, D., M. Huber, M. Bauer & G. Klinker (2006). *Spatial Relationship Patterns: Elements of Reusable Tracking and Calibration Systems*. Garching bei München, Technische Universität München, Fakultät für Informatik: 88-97.
- 72 R. Hourad, F. D., B.Lamiroy, B. Christy (1997). Object pose: The link between weak perspective, paraperspective and full perspective. *International Journal of Computer Vision* 22(2): 173-189.
- 73 Rakkolainen, I. (2006). Tracking users trough a projection screen. In: *Proc. International Multimedia conference*, Santa Barbara, CA, USA, ACM Press. 101-104.
- 74 Rekimoto, J. (1997). Pick-and-Drop: A direct manipulation technique for multiple computer environments. *ACM Symposium on User Interface Software and Technology*: 31-39.
- 75 Rhodes, M. (1961). An analysis of creativity. *Phi Delta Kappan* 42: 305-311.
- 76 Rosenhahn, B., C. Perwass & G. Sommer (2005). *Foundations about 2D-3D Pose Estimation*.
- 77 Rossi, C., M. Abderrahim & J. C. Diaz (2005). EvoPose: A Model-based Pose Estimation Algorithm with Correspondences Determination. In: *Proc. Proc. of the International Conference on Mechatronics & Automation*, Niagara Falls, Canada, IEEE.
- 78 Russ, J. C. (1995). *The Image Processing Handbook*, CRC PRESS.
- 79 S. Gold, A. R. (1996). A Graduated Assignment Algorithm for Graph Matching. *IEEE Trans. PAMI* 18: 377-388.
- 80 S. Gold, A. R., C.P. Lu, S. Pappu, E. Kjolsness (1998). New algorithms for 2D and 3D Point Matching: Pose estimation and Correspondence. *Pattern Recognition* 31: 1019-1031.
- 81 Sachse, P. (2002). *Idea materials: Entwurfsdenken und Darstellungshandeln*. Berlin, Logos Verlag.
- 82 Schneiderman, B. (2002). *User Interface Design*. Bonn, mitp-Verlag.
- 83 Sebastian Vogt, A. K., Frank Sauer, Heinrich Niemann (2002). Single Camera Tracking of Marker Clusters: Multiparameter Cluster Optimization and Experimental Verification. In: *Proc. Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR'02)*.

- 84 Shen, C. (2006). Multi-User Interface and Interactions on Direct-Touch Horizontal Surfaces: Collaborative Tabletop Research at MERL. In: *Proc. Proceedings of the First IEEE International Workshop in Horizontal Interactive Human-Computer Systems (TABLETOP'06)*, IEEE.
- 85 Shuang Song, A. M. A. (2004). Insights on Designers Sketching Activities in new Product Design Teams. In: *Proc. ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Salt Lake City, Utah, USA, ASME.
- 86 Smit, F. A., A. v. Rhijn & R. v. Liere (2006). GraphTracker: A Topology Projection Invariant Optical Tracker. In: *Proc. Eurographics Symposium on Virtual Environments*, the eurographics Association.
- 87 Steinicke, F., C. Jansen, K. Hinrichs, J. Vahrenhold & B. Schwald (2006). *Generating optimized Marker-Based Rigid Bodies for Optical Tracking Systems*, Institut für Informatik, WWU Münster.
- 88 Tsai, R. Y. (1986). An efficient and accurate camera calibration technique for 3D machine vision. In: *Proc. Proc. IEEE Conf. computer Vision Pat. Rec.* 364-374.
- 89 Tuckman, B. W. (1965). Developmental sequences in small groups. *Psychological Bulletin* 63: 348-399.
- 90 Tversky, B. (2003). *Sketching for Design and Design of Sketches. Human Behaviour in Design*. Berlin, Springer-Verlag.
- 91 Ullmer, B. & H. Ishii (2000). Emerging frameworks for tangible user interfaces. *IBM Syst. J.* 39(3-4): 915-931.
- 92 Underkoffler, J. & H. Ishii (1998). Illuminating Light: An Optical Design Tool with a Luminous-Tangible Interface. In: *Proc. Proceedings of the 1998 ACM Conference on Human Factors in computing Systems (CHI'98)*, ACM Press. 542-549.
- 93 Vicon. (2008). Retrieved 23. March, 2008, from www.vicon.com.
- 94 Ware, C. & B. Ravin (1994). Reaching for objects in VR displays: lag and frame rate. *ACM Trans. Comput.-Hum. Interact.* 1(4): 331-356.
- 95 Weiser, M. (1991). The Computer for the 21st Century. *Scientific America* 265(3): 94-104.
- 96 William Hoff, T. V. (2000). Analysis of Head Pose Accuracy in Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics* 6(4): 319-334.
- 97 Wloka, M. (1995). *Interacting with Virtual Reality*, Chapman & Hall.
- 98 wp (2008) *Neue Wege im Design*. DOI:
- 99 Zhang, Z. (1999). Flexible Camera Calibration by Viewing a Plane From Unknown Orientations. In: *Proc. International Conference on Computer Vision (ICCV'99)*, Corfu, Greece. 666-673.

100 Zhang, Z. (2000). A Flexible New Technique for Camera Calibration. *IEEE Trans. PAMI* 22(11): 1330-1334.