



UNIVERSITÄT KARLSRUHE (TH)

FAKULTÄT FÜR INFORMATIK

Institut für Prozeßrechentechnik und
Robotik (IPR)

Prof. U. Rembold

Prof. H. Wörn

Prof. R. Dillmann

FAKULTÄT FÜR ELEKTROTECHNIK

Institut für Biomedizinische Technik (IBT)

Prof. O. Dössel

Prof. K. Meyer-Waarden

Lagebestimmung mit Hilfe eines 3D- Oberflächenscanners

Diplomarbeit von: Claude LOULLINGEN

Hauptreferent: Prof. DÖSSEL

Koreferent: Prof. REMBOLD

Betreuer: Hartwig GRABOWSKI

Mai 1998

Wie durch §19 Absatz 7 der Prüfungsordnung der Universität Karlsruhe für den Diplomstudiengang Elektrotechnik vom 31. Januar 1990 und Änderung vom 31. März 1993 vorgesehen, erkläre ich hiermit, daß ich diese Arbeit selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Karlsruhe, den 18. Mai 1998

(Claude Loullingen)

Mein besonderer Dank gilt:

Hartwig Grabowski für die exzellente und geduldige Betreuung dieser Arbeit.

Der ganzen **Megi-Gruppe** die immer mit Rat und Tat zur Seite stand.

Der **Firma RSI** für die hervorragende Unterstützung in allen Angelegenheiten des
Oberflächenscanners.

Inhaltsverzeichnis

1.	Einleitung	2
2.	Stand der Technik	8
2.1.	Markierungsverfahren	8
2.1.1.	Schrauben mit Innengewinde	9
2.1.2.	geklebte Landmarken	9
2.2.	Methoden der Positionsbestimmung	9
2.2.1.	Stereotaktischer Rahmen	9
2.2.2.	Pointer	10
3.	Grundlagen	12
3.1.	Triangulation	12
3.2.	Lichtschnittechnik	13
3.3.	Codierter Lichtansatz	15
3.4.	Phasen-Sampling-Methode	16
4.	Spezifikation des Oberflächenscanners	22
4.1.	Digiface 3D-Oberflächenscanner	22
4.1.1.	Meßprinzip	24
4.1.2.	Kalibration	25
4.1.3.	Geometrie des Versuchsaufbaus	27
4.1.4.	Software	28

4.2.	Verfahren zur Bestimmung der Meßgenauigkeit des Scanners	29
4.2.1.	Verfahren zur Bestimmung der Genauigkeit der z-Koordinate	31
4.2.2.	Verfahren zur Bestimmung der Genauigkeit der x- und y-Koordinate	32
4.3.	Ergebnisse der Fehlerbestimmung	36
4.3.1.	Ebenen konstanter Höhe	37
4.3.2.	Schiefe Ebenen	42
4.3.3.	Schlußfolgerung	43
4.4.	Untersuchung der Störeinflüsse	44
4.5.	Abschließende Betrachtung	46
5.	Matchingverfahren	48
5.1.	Methodik	49
5.2.	Homogene Koordinaten	50
5.3.	Fehlerfunktion	52
5.3.1.	Extraktion der Oberfläche aus dem Volumenmodell	53
5.3.2.	Distanztomogramm	54
5.4.	Normierungsmatrix	55
5.5.	Initiale Transformation	57
5.6.	Optimierung	60
5.7.	Implementierung	62
6.	Ergebnisse & Ausblick	68
6.1.	Ergebnisse & Diskussion	68
6.2.	Zusammenfassung & Ausblick	76
A.	Anhang A	A1

Kapitel 1

Einleitung

1. Einleitung

Die rechner- und sensorgestützte Chirurgie ist eine der neuen Herausforderungen der Medizintechnik. Dabei ist in dem jetzigen Stadium nicht einmal daran zu denken den Chirurgen auch nur ansatzweise zu ersetzen. Ziel ist es vielmehr dem Chirurgen mit dem Roboter ein Werkzeug zur Seite zu stellen, das es ihm ermöglicht die Grenze des ‘Menschenmöglichen’ wieder ein Stück zu verschieben.

Eines der ersten dazu zu lösenden Probleme besteht dabei in einer zuverlässigen Lagebestimmung des Patienten auf dem Operationstisch. Mathematisch gesehen ist dies nichts anderes als die Bestimmung einer Transformation zwischen einem patienteneigenen und einem beliebig anderen Koordinatensystem, z.B. das eines Operationsroboters. In dieser Arbeit wird das Bezugssystem aber ein Volumenmodell des Patienten sein. Anhand eines solchen Volumenmodells soll nämlich im Rahmen der Operationsplanung die Möglichkeit eröffnet werden später vom Roboter auszuführende operative Eingriffe zu planen. Speziell in der Kopfchirurgie gibt es eine ganze Reihe von Operationen bei denen Roboter geradezu prädestiniert zur Anwendung kommen könnten. Im wesentlichen sind dies Bohr-, Fräs- und Sägearbeiten am Schädelknochen. Aus diesem Grund werden sich im weiteren die praktischen Betrachtungen am Kopf orientieren.



Abb. 1.1: Volumenmodell eines Kopfes

Die zur Zeit angewendeten Methoden der Lagebestimmung (stereotaktischer Rahmen, Marker, Infrarotsysteme) haben oft den Nachteil, daß Sie relativ belastend für den Patienten sind. Außerdem wird die Transformation klassischerweise dadurch erstellt, daß man drei exakt korrespondierende Punktpaare in beiden Systemen bestimmt und daraus die entsprechende Abbildung berechnet. In der Praxis ist aber die Exaktheit nie hundertprozentig zu erreichen (siehe Kapitel 2).

In dieser Arbeit kommt deshalb ein *3D-Oberflächenscanner* zum Einsatz, der durch ein berührungsloses optisches Verfahren die Koordinaten von Punkten (10000 Punkte und mehr) auf der Oberfläche eines Meßobjektes bestimmt. In unserem Fall zum Beispiel der Kopf eines Patienten. Die gemessenen Punkte als Gesamtheit bilden eine Punktwolke, die implizit die Lage des aufgenommenen Objektes als Information in sich trägt. Die Datenerfassung erfolgt also ohne jeglichen präparativen Aufwand am Patienten und schließt durch die Verwendung normalen Lichts eine Belastung des Patienten aus.



Abb. 1.2: Punktwolke aus Oberflächenscan des gleichen Kopfes

Auch zur Bestimmung der Transformation wird ein anderer Weg eingeschlagen. Anstelle der Suche nach exakt korrespondierenden Punkten wird eine Oberflächenregistrierung der Punktwolke mit dem Volumenmodell vorgenommen. Es soll also die von der Punktwolke beschriebene Oberfläche mit der Oberfläche des Modells in Übereinstimmung (engl.: matching) gebracht werden.

Innerhalb der Diplomarbeit werden daher die folgenden zwei Teilaufgaben zu bearbeiten sein:

1. Spezifizierung des Oberflächenscanners

Da die Anwendung eines Oberflächenscanners zur Lagebestimmung in der Medizin Neuland ist, ist es erforderlich sich zuerst umfassend mit dem Scanner zu beschäftigen. Es sind also geeignete Verfahren zur Bestimmung der relevanten Systemparameter und Betriebsbedingungen zu entwickeln (siehe Kapitel 4).

Da es sich um ein optisches Meßverfahren handelt, geht es dabei im speziellen um die Untersuchung:

- einer für unsere Zwecke ausreichenden Genauigkeit des Gerätes
- der Störanfälligkeit bei ungünstigen Lichtverhältnissen
- der Störanfälligkeit gegenüber ungünstigen Oberflächenbeschaffenheiten

Um gezielt diese Spezifizierung des Oberflächenscanners vornehmen zu können wird sich basierend auf den in Kapitel 3 skizzierten Grundlagen am Anfang des Kapitels 4 grundlegend mit dem Aufnahmeverfahren beschäftigt.

2. Implementierung eines Algorithmus zur Oberflächenregistrierung der Punktwolke mit einem Volumenmodell

Zur expliziten Lagebestimmung eines Kopfes soll die erwähnte Oberflächenregistrierung der Punktwolke mit dem Modell implementiert werden. Dieses Matching ist durch einen Algorithmus in C++ zu realisieren. Als Resultat wird eine Transformationsmatrix ausgegeben, die die Übertragung von Ortsangaben aus dem Modell auf den Patienten ermöglicht. Die Eingaben des Algorithmus wären also ein korrespondierendes Punktwolke-Modell-Paar, und die Ausgabe eine Transformationsmatrix, die die Koordinatentransformation zwischen beiden System beschreibt.

Der Entwurf dieses Verfahrens findet in Kapitel 5 statt, während die erzielten Ergebnisse in Kapitel 6 dokumentiert sind. Abschließend erfolgt eine Zusammenfassung und eine Reflexion über die getane Arbeit sowie ein Ausblick auf zukünftige Verbesserungs- und Anwendungsmöglichkeiten.

Terminologie

Um Verwirrung vorzubeugen will ich noch kurz auf die Bedeutung von verschiedenen Begriffen innerhalb dieser Arbeit eingehen:

Die einfache Bestimmung der Koordinaten bestimmter Punkte des Meßobjektes zu einem definierten Koordinatensystem wird im Gegensatz zu der eben beschriebenen Lagebestimmung mit *Positionsbestimmung* bezeichnet.

Unter *Punktwolke* verstehe ich sowohl die Gesamtheit der aus einer Messung hervorgegangenen Liste an Punkte als auch die Datei unter der diese Liste abgespeichert werden kann.

Mit *CT* (Computertomogramm) wird im weiteren nicht eine einzelne Schichtbildaufnahme bezeichnet sein sondern immer ein sogenannter 3D-Datensatz, also eine Folge von mehreren in einem bestimmten örtlichen Abstand aufgenommenen CT-Schichtbildern. Weiter spielt es im folgenden auch keine Rolle, ob es sich bei einem *CT* wirklich um ein echtes Computertomogramm oder ein Magnetresonanztomogramm (MRT) handelt. Beide Verfahren sind Schnittbildverfahren und können im gleichen Format gespeichert werden. Von den aufgenommenen Körpern wird innerhalb dieser Arbeit nur die Oberfläche interessieren.

Die *reale Welt* bezeichnet den Raum in dem das Meßobjekt liegt, im Gegensatz zu seinem Bild- oder Modellraum in den das Objekt durch ein bildgebendes Verfahren projiziert wird.

Kapitel 2

Stand der Technik

2. Stand der Technik

Mathematisch betrachtet reduziert sich die Lagebestimmung eines Objektes auf die Bestimmung einer Transformation zwischen einer objekt eigenen Basis und einem beliebig anderen Koordinatensystem. Prinzipiell genügt es also drei exakt korrespondierende Punktpaare zwischen dem vorliegenden Patienten und dem Modell zu finden. Sind die drei resultierenden Vektoren linear unabhängig so kann man die Transformationsmatrix bestimmen.

$$\begin{pmatrix} \vec{v}_1' & \vec{v}_2' & \vec{v}_3' \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \cdot \begin{pmatrix} \vec{v}_1 & \vec{v}_2 & \vec{v}_3 \end{pmatrix}$$

Dabei werden an die Exaktheit der Korrespondenz sehr hohe Maßstäbe gesetzt da sonst die Transformation nicht isometrisch also nicht längen- und winkeltreu ist.

In der Praxis stellt sich dabei aber das Problem wie man erstens über Tage hinweg und zweitens in unterschiedlichsten Aufnahmeverfahren am Menschen eindeutig die gleichen Punkte immer wiederfindet und dabei noch der geforderten Exaktheit gerecht wird.

Nichts desto trotz ist diese Vorgehensweise Stand der Dinge und deshalb sind im folgenden einige Realisierungen davon skizziert indem zuerst auf die Markierungsverfahren der zu suchenden Punkte eingegangen wird und danach die Methoden deren Positionsbestimmung in der realen Welt beschrieben werden.

2.1 Markierungsverfahren

Landmarken werden in der Medizin Merkmale genannt, die einen Punkt enthalten, der sowohl in einem auf bildgebender Technik basierenden präoperativen Model des Patienten, als auch am Patienten selber, eindeutig identifizierbar sind. Dabei unterscheidet man zwischen künstlich am Patienten angebrachte Landmarken und sogenannten natürlichen oder anatomischen Landmarken, wie zum Beispiel die Mittelpunkte der Augäpfel oder die Kreuzung der Sehnerven [Pokrandt 97 (S. 16)]. Für die Lagebestimmung relevant sind im Moment aber nur die künstlichen Landmarken, deshalb soll im folgenden nur auf diese eingegangen werden.

Wichtig ist, daß die Landmarken ab dem verwendeten Aufnahmeverfahren bis zur Operation am Patienten bleiben. Je nach Art der Fixierung kann man also grob zwei Typen unterscheiden:

2.1.1 Schrauben mit Innengewinde

Hierbei werden Schrauben in den Knochen des Patienten eingesetzt. Mittels eines Gewindebolzens können dem jeweiligen Aufnahmeverfahren angepaßte Landmarken in das Innengewinde der jeweiligen Schraube gedreht werden. Zum Einsatz kommen des öfteren Kugel aus unterschiedlichen Materialien. Durch die Fixierung am Knochen führen diese Landmarken zu äußerst genauen Ergebnissen. Sie stellen jedoch durch das Einsetzen und Entfernen der Schrauben eine erhebliche Belastung des Patienten dar.

2.1.2 Geklebte Landmarken

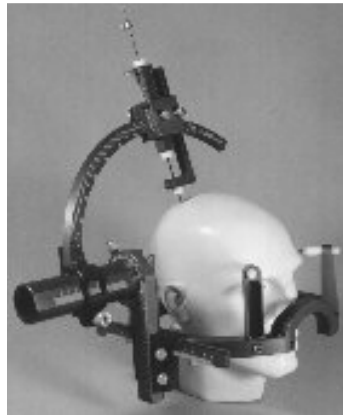
Diese Landmarken bestehen aus ca. 1-2 mm dickem Material und besitzen einen Durchmesser von etwa 10-15 mm. Sie werden auf die Haut aufgeklebt und können durch ihre Form relativ gut aus den Aufnahmen extrahiert werden. Für den Patienten ergibt sich dadurch eine wesentlich geringere Belastung. Die Genauigkeit dieses Verfahrens ist allerdings durch die leichte Verschiebbarkeit der Haut wesentlich schlechter.

2.2 Methoden der Positionsbestimmung

Die Position der im vorherigen Abschnitt beschriebenen Marker wird klassisch durch folgende Techniken bestimmt oder abgetastet:

2.2.1 Stereotaktischer Rahmen

Im allgemeinen findet der stereotaktische Rahmen Anwendung am Schädel eines Patienten. Dazu wird für die Operation an vier bis acht Schrauben mit Innengewinde ein bis zu mehreren Kilogramm schwerer Metallring fixiert. An diesem Metallring sind ein oder zwei bewegliche Halbkreisbögen befestigt, die mit einer Winkelbemaßung versehen sind. Am Kreuzungspunkt der Bögen können durch eine Führung Werkzeuge zentrisch eingebracht werden. Der Metallring mit seinen Halbkreisbögen spannt also praktisch ein Kugelkoordinatensystem auf.



Quelle: www.micromar.com/frameing.htm

Abb. 2.1: stereotaktischer Rahmen

Da die Schrauben im CT gut erkennbar sind, erhält man so korrespondierende Punkte im CT und dem Kugelkoordinatensystem des stereotaktischen Rahmens. Hieraus kann man die Transformation zwischen den beiden Systemen berechnen.

Durch den absolut festen Sitz am Kopf ist eine stabile Registrierung garantiert. Stereotaktische Rahmen stellen bis zum heutigen Tag die meistverwendete Methode zur Gewinnung von Punkten für die Registrierung dar.

2.2.2 Pointer

Ein anderes Verfahren zur Positionsbestimmung der Landmarken erfolgt mit Hilfe eines Pointers. Dieser Pointer besteht im wesentlichen aus einem Stift mit spitz zulaufendem Ende. Diesen Stift bewegt man dann mit der Spitze auf die zu bestimmenden Stellen. Die eigentliche Positionsbestimmung erfolgt durch Lagebestimmung des Stiftes. Dazu ist er entweder an einem beweglichen Arm fixiert, in dessen Gelenken über Winkelgeber die Lage abgetastet wird [Microscribe], oder am Stift sind Infrarotdioden befestigt deren Position wiederum durch ein Infrarottrackingsystem ausgewertet wird.

Kapitel 3

Grundlagen

3. Grundlagen

Da es von Nutzen sein wird zu wissen wie der Oberflächenscanner an die Tiefeninformation der bestimmten Punkte kommt, werden hier die Grundlagen zu dessen Beschreibung im Kapitel 4 gelegt. Dazu sind mehrere Verfahren zur Gewinnung von Tiefeninformationen aus 2D-Bilder beschrieben. In [Breuckmann 92] Kapitel 6.2 & 4.4 und [Gerl 97] Kapitel 4.2 findet man hierzu auch einen guten Überblick, während die mathematische Formulierung des Phasen-Sampling sehr gut in [Osten 91] beschrieben ist.

3.1 Triangulation

Die Triangulation ist das einfachste optische Verfahren zur Gewinnung von Abstandsdaten. Mit einem Projektor wird ein punktförmiger Lichtstrahl auf die Stelle eines auszumessende Objektes gerichtet von der man den Abstand zu einem bestimmten Empfänger (Sensor) wissen will. Klassischerweise ist der Sensor eine Kamera.

Richtet man nun die Kamera so aus, daß der Lichtpunkt auf dem Meßobjekt genau auf der Verlängerung der optischen Achse des Sensors (normalerweise die Bildmitte) zum Liegen kommt, so kann der Abstand z unmittelbar aus einer einfachen mathematischen Beziehung gewonnen werden.

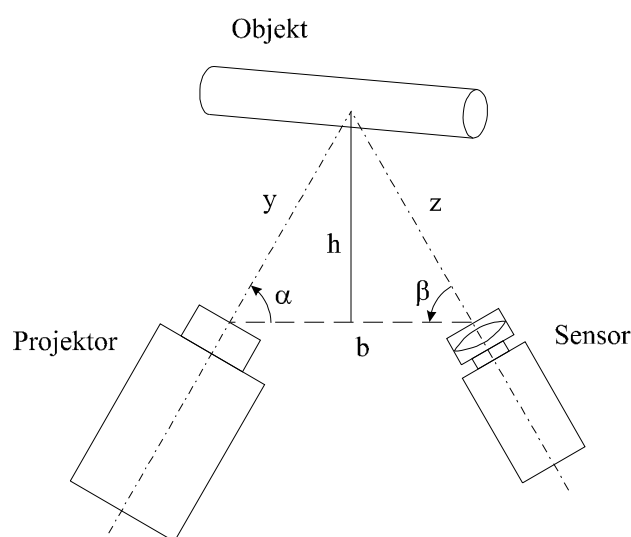


Abb. 3.1: Prinzip der optischen Triangulation

Der Ort des Projektors, der beleuchtete Objektpunkt und die Kamera bilden eine Dreieck. Kennt man nun den Abstand b zwischen dem Projektor und der Kamera, sowie deren Position gegeneinander in Form der Winkel α und β , so kann man aus den folgenden geometrischen Abhängigkeiten den Abstand des Meßpunktes zur Kamera berechnen:

$$\begin{cases} b = z \cdot \cos(\beta) + y \cdot \cos(\alpha) & (1) \\ h = z \cdot \sin(\beta) = y \cdot \sin(\alpha) & (2) \end{cases}$$

aus (2) in (1) folgt dann:

$$z = \frac{b}{\sin(\beta) \cdot \left(\frac{1}{\tan(\alpha)} + \frac{1}{\tan(\beta)} \right)}$$

Nachteilig bei der Erfassung ganzer Flächen ist die Aufnahmegeschwindigkeit und die zur Bewegung des Projektors erforderliche Mechanik.

3.2 Lichtschnitttechnik

Statt lediglich einen Punkt auf ein Objekt zu projizieren, kann man auch einen Lichtstreifen oder einen scharfen Hell-Dunkel-Übergang auf das Objekt werfen. Nimmt man wiederum die Szene mit einer Kamera auf und berücksichtigt man weiter deren Optik, so kann man gleichzeitig alle Punkte längs des Lichtschnittes durch Triangulation auswerten.

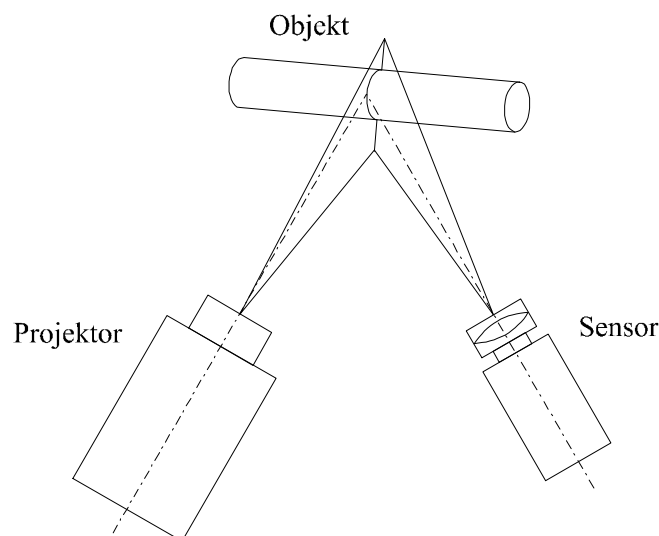


Abb. 3.2: Lichtschnittverfahren

Der nächste Gedankenschritt liegt nun fast auf der Hand; warum nicht gleich mehrere Lichtschnitte gleichzeitig projizieren?

Effektiv kann man durch gezielte flächenhafte Projektion von strukturiertem Licht mit einer Aufnahme die gesamte Meßszene erfassen. Anwendungen und Behandlungen dieser Lichtschnitttechnik finden sich auch in [Hof 92], [Ioannides 93], [Klette 96], [Schlipf 93], [Strutz 92], [Wahl 86] und [Wolf 92].

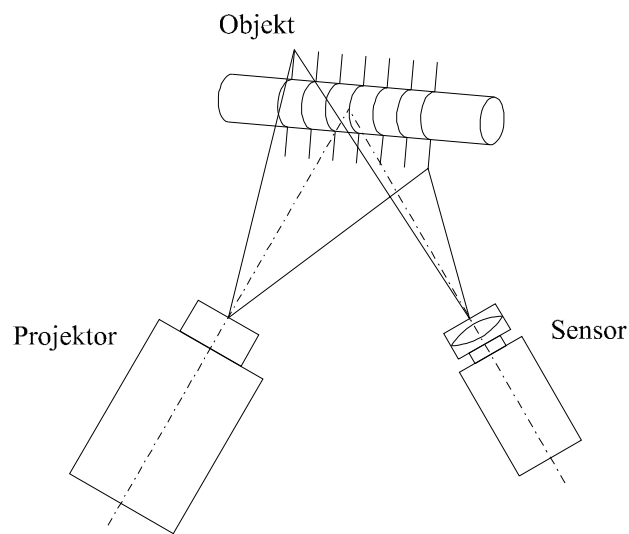


Abb. 3.3: Bildhafte Triangulation

Je feiner dabei die projizierten Linien sind, um so höher kann man bei entsprechend angepaßter Auflösung der Kameras die Tiefe auflösen.

Die “bildhafte” Triangulation funktioniert allerdings nur solange als man immer noch eindeutig jeden Bildpunkt dem entsprechenden Lichtschnitt zuordnen kann. Man sagt auch: Die Ordnung des Lichtschnittes muß in jedem Bildpunkt bekannt sein.

Genau dies ist aber bei starken womöglich noch für die Kamera verdeckten Höhesprüngen nicht mehr möglich. Bild 3.4 zeigt einen denkbaren Kamerabild. Es wird ersichtlich, daß die von oben kommenden Streifen je nach Größe des zugrundeliegenden Höhesprunges unten prinzipiell durch jeden anderen Streifen fortgesetzt werden könnte. Die Lichtschnitte sind also nicht mehr eindeutig zuzuordnen. Abhilfe schafft hier der *codierte Lichtansatz*.

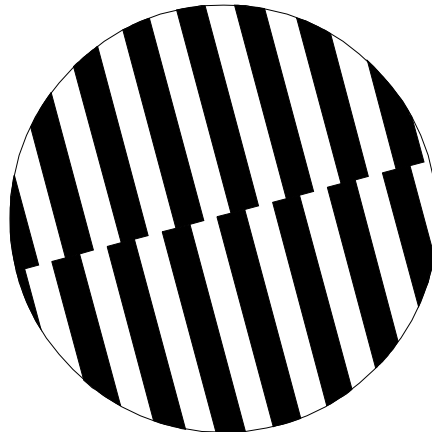


Abb. 3.4: Lichtschnitte an Höhengsprung

3.3 Codierter Lichtansatz

Beim codierten Lichtansatz projiziert man eine Serie (z.B.: 8) von immer feiner werdenden schwarzweiß Streifenmuster auf ein Meßobjekt. Jedes feinere Streifenmuster enthält dabei gerade die doppelte Anzahl an Streifen als das vorhergehende Muster. Für jeden Bildpunkt beobachtet man währenddessen die Abfolge der Hell-Dunkel-Wechsel und codiert sie binär. Wird ein Pixel durch einen hellen Streifen beleuchtet so notiert man sich zum Beispiel eine '1' ansonsten eine '0'. So verfährt man für jedes Streifenmuster.

Für jedes Bildpixel erhält man so eine bestimmte Bitfolge. Diese Bitfolge codiert nun gerade die Ordnung des Lichtschnittes auf dem der dem Pixel entsprechenden Szenenbereich liegt, wenn das zuletzt projizierte feinste Streifenmuster die Lichtschnitte definiert.

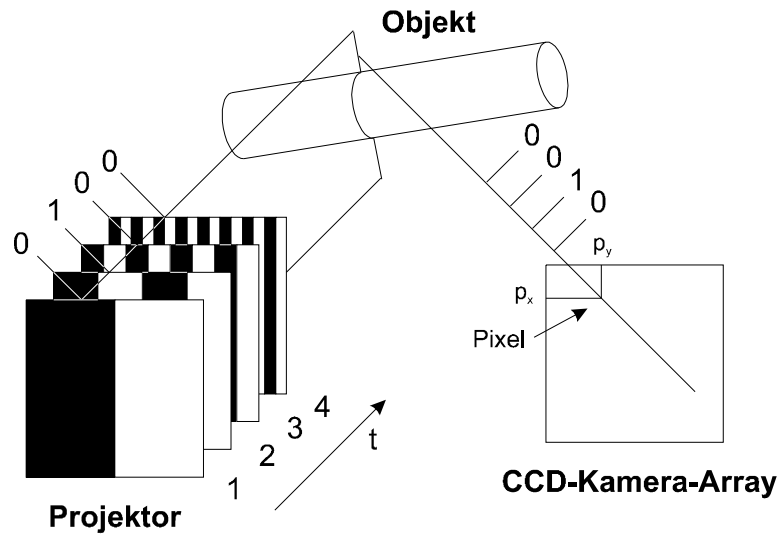


Abb. 3.5: Codierter Lichtansatz

Nehmen wir zum Beispiel an alle Streifenmuster begännen links mit einem schwarzen Streifen. Wäre ein Bildpunkt also anschließend mit der Folge "0,0,0,0,0,0,0,0" kodiert, so muß dieser Bildpunkt notgedrungenerweise auf dem ersten schwarzen Streifen des feinsten Musters gelegen haben. Mit 8 Bit und Streifenmustern können so $2^8 = 256$ Streifen kodiert werden.

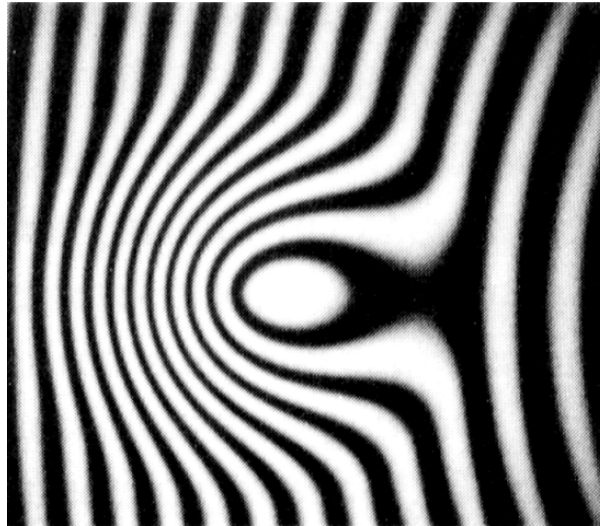
Die Auflösung der Tiefe ist irgendwann durch die Auflösung der Kamera und der nachfolgenden optischen Elemente begrenzt. Wie in Kapitel 4 zu sehen sein wird kann man aber durch einen Trick noch mehr Information gewinnen. Zum Verständnis ist aber ein kleiner Exkurs in die Interferometrie nötig.

3.4 Phasen-Sampling-Methode

Bringt man zwei phasenverschobene monochromatische linear polarisierte ebenen Lichtwelle zur Interferenz so ist bekannt, daß sich die Intensität der resultierende Lichtwelle nicht aus der Summe der Intensitäten der Teilwellen zusammensetzt, sondern von der Phasendifferenz zwischen beiden Teilwellen abhängig ist.

In der Physik macht man sich diesen Effekt zur Abstandsmessung im Wellenlängenbereich des Lichtes zunutze. Laufen nämlich zwei Wellen unterschiedliche Strecken und werden dann zur Interferenz gebracht, so ergeben sich charakteristische Interferenzbilder mit Bereichen

gegenseitiger Auslöschung oder Verstärkung (s. Abb. 3.6). Der Abstand von einem Intensitätsmaximum zum Nächsten entspricht dann gerade einem Laufstreckenunterschied von einer halben Wellenlänge des verwendeten Lichts. Die Interferenzlinien entsprechen also praktisch den Höhenlinien des abgetasteten Profils, wobei der Höhenunterschied zwischen zwei Höhenlinien genau $\lambda/2$ entspricht.



Quelle: [Osten 91]

Abb. 3.6: Beispiel eines Interferenzmusters

In der Meßtechnik bedient man sich nun der Interferenztechnik zur hochgenauen Oberflächenprofilabtastung. Zwecks einer automatischen Auswertung der Interferenzbilder nimmt man das Interferenzbild mit einer Kamera auf. Nun muß aus dem so entstandenen Intensitätsbild der Phasenverlauf rekonstruiert werden. Dabei kann man folgenden Zusammenhang ansetzen:

$$I(x, y) = a(x, y) + b(x, y) \cdot \cos[\delta(x, y) + \varphi_r] \quad (\text{Gl. 3.1})$$

Hierbei sind x und y die zweidimensionalen Ortskoordinaten. $a(x, y)$ bezeichnet die Grundintensität, $b(x, y)$ die Schwingungsamplitude der Intensität und $\delta(x, y)$ die interessierende Phase. Ebenfalls eingeführt wurde hier die Referenzphase φ_r , deren Rolle von der verwendeten Meßmethode abhängt. Ein Vergleich unterschiedlicher Phasenmeßverfahren wird in [Creath 86] und [Creath 88] angestrengt.

Eine dieser Methoden ist die Phasen-Sampling-Methode. Hier wird φ_r in m Schritten der Größe φ_0 durchgestimmt. Nach jeder Teilverschiebung der Referenzphase werden die phasenverschobenen Streifenbilder zur Weiterverarbeitung aufgenommen. Man erhält also für jeden Punkt im Interferogramm m Intensitätswerte I_n :

$$I_n(x, y) = a(x, y) + b(x, y) \cdot \cos[\delta(x, y) + \varphi_n]$$

mit:

$$\varphi_n = (n - 1) \cdot \varphi_0 \quad n = 1 \dots m$$

$$\varphi_0 = \frac{2\pi}{m} \quad m \geq 3$$

Zur Auswertung sind mindestens 3 Intensitätsmessungen erforderlich, um die 3 Unbekannten $a(x, y)$, $b(x, y)$ und $\delta(x, y)$ zu bestimmen. Der in [Osten 91] beschriebene Weg zur Lösung des Gleichungssystems führt für $m=4$ allerdings zu einer wesentlich einfacheren Beziehung:

$$m = 4, \quad \varphi_0 = \frac{\pi}{2}$$

$$\delta(x, y) \pmod{\pi} = \arctan \frac{I_3(x, y) - I_2(x, y)}{I_1(x, y) - I_2(x, y)} \quad (\text{Gl. 3.2})$$

Unter Berücksichtigung der Vorzeichen von Zähler und Nenner (s. Abb. 3.7) liefert Gl. 3.2 den Hauptwert der arctan-Funktion im Intervall $[-\pi \dots \pi]$.

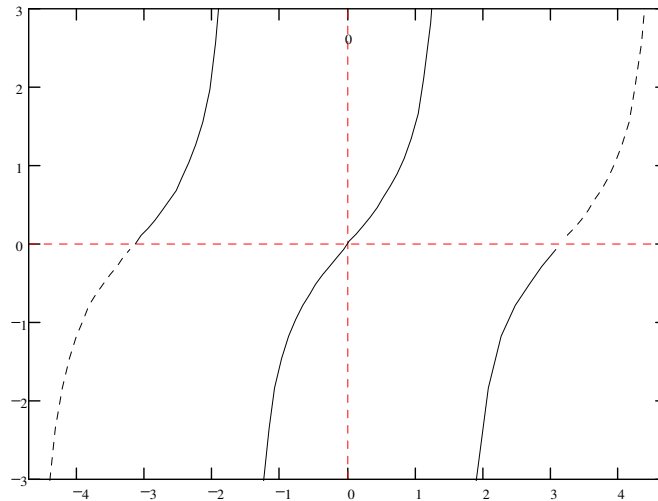


Abb. 3.7: Hauptwert der arctan-Funktion

Bedingt durch die im Abstand von 2π auftretenden Unstetigkeiten der arctan-Funktion entsteht ein unstetiger Phasenverlauf der erst phasenrichtig wieder zusammengesetzt werden muß. Diese Demodulation bezeichnet man als “phase unwrapping”.

Soweit soll der Einblick in die Interferometrie genügen. Wir können festhalten, daß der codierte Lichtansatz das geeignete Hilfsmittel ist aus einer “bildhaften” Triangulation eindeutige Meßergebnisse zu erhalten. Durch eine noch zu klärende Verbindung mit dem Phasen-Sampling der Interferometrie, kann man zusätzlich die Tiefe bis in den Subpixelbereich hinein auflösen.

Kapitel 4

Spezifikation des Oberflächenscanners

4. Spezifikation des Oberflächenscanners

Wie kaum eine andere Technik hat die Bildverarbeitung Einzug in den praktischen Alltag gehalten. Durch die drastische Kostenreduzierung bei Rechenanlagen sowie die rasante Entwicklung immer leistungsfähigerer CCD-Kamerasensoren, findet die Bildverarbeitung verstärkte Anwendung in den unterschiedlichsten Gebieten der Industrie.

In dieser Arbeit bildet die Bildverarbeitung den zentralen Teil der Datenaufnahme in der Form eines 3D-Oberflächenscanners der Firma RSI. Da die Anwendung eines solchen Scanners, zur Lagebestimmung in der Medizin Neuland ist, ist es innerhalb dieser Arbeit erforderlich die externen Systemparameter sowie Betriebsbedingungen näherer zu untersuchen.

In dem folgenden Kapitel wird deshalb zunächst die Funktionsweise des Scanners, basierend auf den Grundlagen aus Kapitel 3, genauer beschrieben. Anschließend werden theoretische Überlegungen zur Bestimmung der Genauigkeit angestrengt um daraus praktische einsetzbare Methoden abzuleiten.

Soll das Verfahren praktisch anwendbar sein, muß es robust gegenüber schwankenden Umgebungsbedingungen sein. Diese Störquellen werden deshalb diskutiert und beurteilt.

4.1 Digiface 3D-Oberflächenscanner

Der Oberflächenscanner besteht im Wesentlichen aus einem LCD-Projektor, zwei CCD-Kameras, einem PC mit entsprechender Hardware und einer bildverarbeitenden Software.

Der Versuchsaufbau ist in Abbildung 4.1 a) zu sehen. In der Mitte oben ist der LCD-Projektor zu sehen. Er bestrahlt die Szene senkrecht nach unten. Rechts und links davon sind die CCD-Kameras montiert. Neben dem Gerüst steht der PC der den LCD-Projektor steuert und die Kamerabilder auswertet. Unter dem Scanner befindet sich der Kalibrationsplattenhalter mit der eingeschobenen Kalibrationsringplatte.

In Abbildung 4.1 b) ist der funktionalen Aufbau dargestellt.

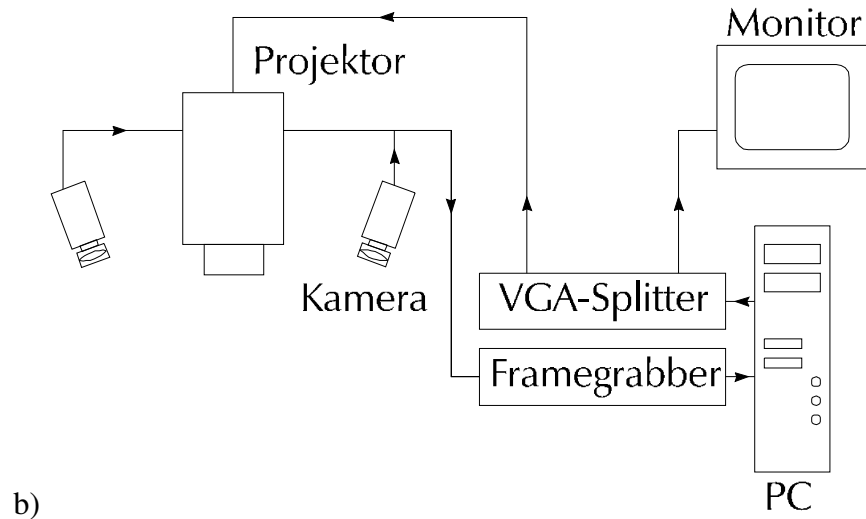
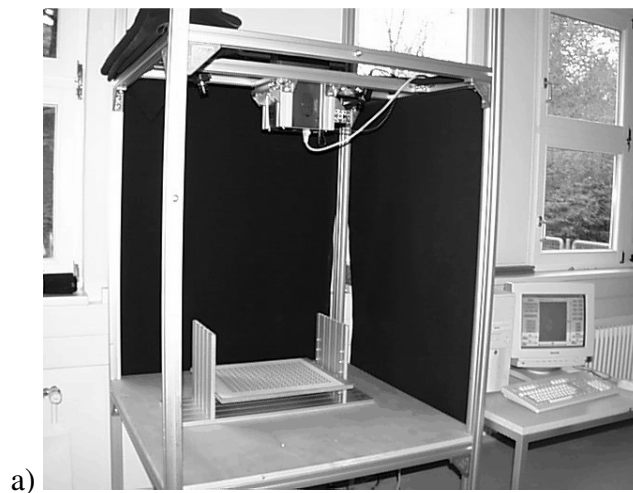


Abb. 4.1: Aufbau des Scanners: a) Photo Versuchsaufbau, b) Prinzipskizze

Das von der Grafikkarte des PC erzeugte Bild, mit einer Auflösung von 640×480 Punkten, wird über einen VGA-Splitter sowohl dem Monitor als auch dem LCD-Projektor zugeführt. Die beiden Kamerasignale werden über den grünen (links) und roten (rechts) Kanal der Framegrabberkarte zugeführt. Die Kameras haben eine Auflösung von 740×572 Punkten. Zur Erhöhung des Signal/Rauschverhältnisses werden aber, laut Herstellerangaben, jeweils 4 Punkte zusammengefaßt wodurch sich für alle Aufnahmen eine maximale Auflösung von 370×286 Punkten ergibt. Sowohl die Kameras als auch der Projektor sind so ausgerichtet, daß sie eine Portraitaufnahme der Szene erzeugen.

Es wird bereits jetzt darauf hingewiesen, daß beide Kameras unabhängig voneinander zwei getrennte Punktwolken liefern. Es handelt sich nicht um ein stereoskopisches Verfahren. Nur zur Erfassung größerer Objektbereiche und zur gegenseitigen Ergänzung von Abschattungen werden zwei Kameras verwendet.

4.1.1 Meßprinzip

Das Meßprinzip des Oberflächenscanners gehört zur topometrischen 3D-Meßtechnik und ist eine Abwandlung des Phasen-Sampling-Verfahrens der Interferometrie (s. Abschn. 3.4). Hierbei werden allerdings keineswegs zwei kohärente Lichtstrahlen zur Überlagerung gebracht, sondern es werden mit dem LCD-Projektor ganz feine schwarzweiße Liniengitterstrukturen direkt auf das Meßobjekt projiziert (s. Abb. 3.3). Das Phasen-Sampling, das in der Interferometrie durch Verschieben der Referenzphase oder des Meßobjektes erreicht wird, wird hier durch einen Pixelshift des Bildes simuliert.

Der Ablauf einer Aufnahme könnte also folgendermaßen aussehen:

Mit einem LCD-Projektor wird ein schwarzweißes Streifenmuster auf das Meßobjekt geworfen, dabei betrage die Linienbreite gerade 4 Pixel des LCD-Gitters. Nach der Aufnahme der Szene durch die Kamera, projiziert man das gleiche Linienmuster um einen Bildpixel senkrecht zu den Linien verschoben. So verfährt man drei mal und erhält vier Aufnahmen. Die Auswertung der Aufnahme erfolgt nach dem Prinzip des Phasen-Sampling wie in Abschnitt 3.4 bereits besprochen.

In zwei Punkten unterscheidet sich aber das “interferometrische” und das “topometrische” Phasen-Sampling:

Erstens ist die im Moment mit Projektoren erreichbare Auflösung, der Auflösung der Interferometrie weit unterlegen. Mit 4 Pixel breiten Streifen, projiziert auf ein 1 Meter entferntes Objekt, ist man weit von der Auflösung im Wellenlängenbereich entfernt. Eine solche Auflösung ist aber für die hiesige Anwendung auch gar nicht wünschenswert.

Zweitens erfüllen die scharf begrenzten Schwarzweißstreifen nicht den Ansatz des sinusförmigen Intensitätsverlaufs (s. Gl. 3.1). Durch die Unzulänglichkeiten der Optik und vor allem der geringen Tiefenschärfe erfahren die feinen projizierten Strukturen aber eine Tiefpaßfilterung, was dazu führt, daß der Intensitätsverlauf annähernd sinusförmig wird.

Auch beim topometrischen Phasen-Sampling ist, bedingt durch die im Abstand von 2π auftretenden Unstetigkeiten im Phasenverlauf, ein phase-unwrapping nötig. Hierzu bedient man sich des codierten Lichtansatzes (s. Abschn. 3.3). Benutzt man nämlich das feinste Streifenmuster des codierten Lichtansatzes ebenfalls zum Phasen-Sampling, so codiert die Musterfolge des codierten Lichtansatzes gerade alle die Pixel gleich, die zu einer Phasenperiode gehören. Man nutzt das Verfahren also nicht zur direkten Tiefendatengewinnung sondern nur zum phasenrichtigen Zusammensetzen der stetigen Phasenabschnitte. Der Phasenverlauf enthält nun die Information über den relativen Höhenverlauf des abgetasteten Profils.

Bis jetzt wurde lediglich eine Tiefeninformation für jeden Kamerapixel gewonnen. Natürlich ist die Position eines Punktes nicht alleine durch seinen Tiefenabstand bestimmt, er muß noch um zwei Komponenten eines Koordinatensystems erweitert werden. Der Oberflächenscanner bedient sich eines kartesischen Koordinatensystems zur vollständigen Positionsbeschreibung. Das xy-Wertepaar wird für jeden der 105 820 Bildpixel (370×286) aus dem Tiefenwert des entsprechenden Pixels, den Pixelkoordinaten und einer Transformationsmatrix bestimmt. Diese Transformationsmatrix bestimmt sich aus der folgenden Kalibration.

4.1.2 Kalibration

Obwohl man rein mathematisch mit den Pixelkoordinaten, den Abstand eines triangulierten Punktes zur Kamera und einer bekannten Geometrie der Optik, die x- und y-Werte bereits berechnen könnte (s. Abschn. 3.1), geht man aus Flexibilitäts- und wahrscheinlich Genauigkeitsgründen zu einer anderen Methode über.

Eine ausreichend große und absolut ebene weiße Platte wird ungefähr senkrecht zur optischen Achse des Projektors plaziert. Die Platte ist mit schwarzen Ringen bedruckt. Sie sind matrixförmig angeordnet und haben einen definierten, der Software bekannten, Abstand zueinander. Der mittlere Ring ist etwas größer als die anderen.

Von der Ringplatte wird nun durch die Kameras ein Grauwertbild aufgenommen. Durch Schwerpunktbildung werden nun die Flächenmittelpunkte der Kreisinnenflächen der einzelnen Ringe im Bild ermittelt. Die Positionen dieser Schwerpunkte sind in der realen Welt wiederum bekannt. Nun wird die Platte parallel um einen definierten, der Software bekannten, Betrag a_{Kalib} in Richtung des Projektors verschoben. Es erfolgt wieder eine Aufnahme. Dies wiederholt man noch 3 mal, so daß die Platte in 5 äquidistanten parallelen Position aufgenommen wurde.

Wir erhalten also eine ganze Serie an korrespondierenden Punkten des Bildraums und der realen Welt. Diese Punkte spannen in beiden Systemen ein Raumgitter an Stützstellen für die spätere Berechnung der x- und y-Werte auf. Anhand dieses Raumgitters läßt sich weiterhin die gesuchte Transformationsmatrix berechnen.

Bei der Höhenbestimmung hatten wir vorerst nur relative Höhenaussagen treffen können. Um zu absoluten Werten zu kommen, muß auch für die z-Koordinate eine Kalibration durchgeführt werden. Hierzu wird wiederum eine weiße ebene Platte, aber diesmal ohne Ringe, in die gleichen fünf Höhen gebracht wie die Ringplatte vorher. In jeder Höhe wird diesmal ein Oberflächenscan von der Platte vorgenommen.

Nun wird der ersten Aufnahme der Höhenwert $z = 0$ zugeordnet, sowie den folgenden Aufnahmen der Höhenwert:

$$z = 0 + n \cdot a_{\text{Kalib}}$$

Die Platten definieren die Ebenen konstanter Höhe und bilden so für jeden Bildpunkt Stützstellen in der Tiefe, aus denen jeder späteren Aufnahme durch Interpolation eindeutig ein Meßwert zugewiesen werden kann.

In unserem Fall beträgt der Abstand zwischen den Positionen 40 mm und die Platte hat die Größe eines DIN A3 Blattes. Die Platte ist auf $\frac{1}{10}$ mm genau eben.

Das System ist nun auf das durch die Kalibrationsplatten begrenzte Meßvolumen kalibriert. Dieses Volumen hat die Abmaße: 400 * 300 * 160 mm (Höhe * Breite * Tiefe). Meßobjekte sollten sich idealerweise innerhalb dieses Meßvolumens befinden.

Als weiteres Resultat der Bemühungen hat man auch das Koordinatensystem des Scanners aufgebaut. Es hat seinen Ursprung im Zentrum des mittleren Kreises der Ringplatte der ersten Kalibrationsposition (im folgenden *Nullebene* genannt). Die z-Achse steht senkrecht auf den Kalibrationsebenen in Richtung steigender Ebenenordnung. Wie in Abbildung 4.2 zu sehen liegt die x-Achse in der Nullebene in Ringzeilenrichtung, die y-Achse in Ringspaltenrichtung. Somit spannen die Achsen ein rechtshändiges Koordinatensystem auf. Die Koordinaten der Punkte werden in Millimetern ausgegeben.

Eine Kalibration ist prinzipiell solange gültig wie keine Änderung an der Geometrie des optischen Aufbaus vorgenommen wird.

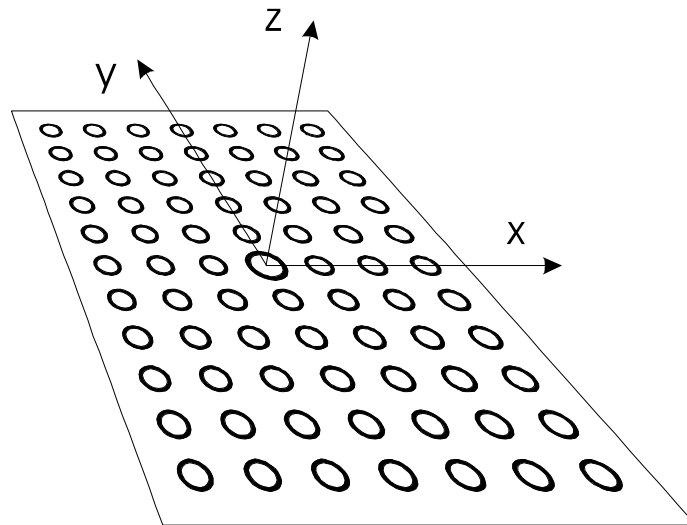


Abb. 4.2: Koordinatensystem des Oberflächenscanners

4.1.3 Geometrie des Versuchsaufbaus

Durch die Kalibration können die Kameras und der Projektor relativ frei zueinander positioniert werden. Bei der Wahl sind aber folgende Faktoren zu berücksichtigen:

1. Größe der Kalibrationsplatten

Es ist erforderlich, daß sich bei allen Aufnahmen der Kalibration nichts anderes im Blickfeld der Kameras befindet, als die Kalibrationsplatte selber.

Das heißt, daß bei gegebenem Objektiv, die hinterste Plattenposition nicht beliebig weit von den Kameras entfernt werden darf.

2. Genauigkeit

Da das Meßverfahren seine Information aus den durch das Profil erzeugten Verwerfungen der Linienmuster erhält, ist es prinzipiell von Interesse die Kameras möglichst flach auf die Szene blicken zu lassen (α und β klein, siehe Abb. 3.1). Dies bedingt allerdings, daß die Kameras immer weniger Lichtschnitte aufnehmen und die Abschattungen gegenseitig immer schlechter ergänzen können.

Bemerkung:

Die Kalibrationsplatten können um die optische Achse des Projektors verdreht, oder gegen die zu ihr senkrechte Ebene um mehrere Grad verkippt sein, ohne daß die Kalibration dadurch gleich scheitert. Das Koordinatensystem wird sich in dem Fall immer nach den obigen Regeln an den Kalibrationsplatten ausrichten. Wichtig ist hingegen, daß die Ringplatte ohne seitliche Verschiebung exakt senkrecht zu der Nullebene verschoben wird.

4.1.4 Software

Die Software ist im Handbuch zum Oberflächenscanner soweit beschrieben [RSI 97]. Auf einige Details, die zum weiteren Verständnis nötig sind, wird dennoch eingegangen.

Es ist möglich bei der Aufnahme zwischen 3 verschiedenen “Auflösungen” zu wählen, sie sind mit 1, $\frac{1}{2}$ und $\frac{1}{4}$ bezeichnet. Durch diese Einstellung wird festgelegt, ob jeder, jeder zweite oder jeder vierte Punkt der Aufnahme ausgewertet wird. Das heißt die Aufnahme enthält pro Kamera entsprechend maximal 105 820, 26 455, oder 6532 Punkte in der berechneten Punktwolke. Da die Bildaufnahme stets mit der vollen Auflösung durchgeführt wird und nur die Auswertung unterschiedlich erfolgt, werden die Punkte immer mit maximaler Genauigkeit bestimmt.

Ein sehr wichtiges Tool ist der Kontrastfilter mit dem eine Punktwolke nachbearbeitet werden kann. Auf dunklen oder sehr fein strukturierten Oberflächen können Punkte nämlich nur sehr ungenau bestimmt werden. Solche Punkte sind für das spätere Matching sehr ungeeignet. Durch das Kontrastfilter wird nun der Kontrast der aufgenommenen Grauwertbilder verstärkt. Dies führt dazu, daß Punkte von den genannten Problemoberflächen von der Software ausmaskiert werden (s. Abb. 2a). Ausmaskieren heißt, daß die Punkte aus der Punktwolke entfernt werden indem ihre Koordinaten auf das Trippel (0,0,0) gesetzt werden, sie werden also in den Koordinatenursprung projiziert.

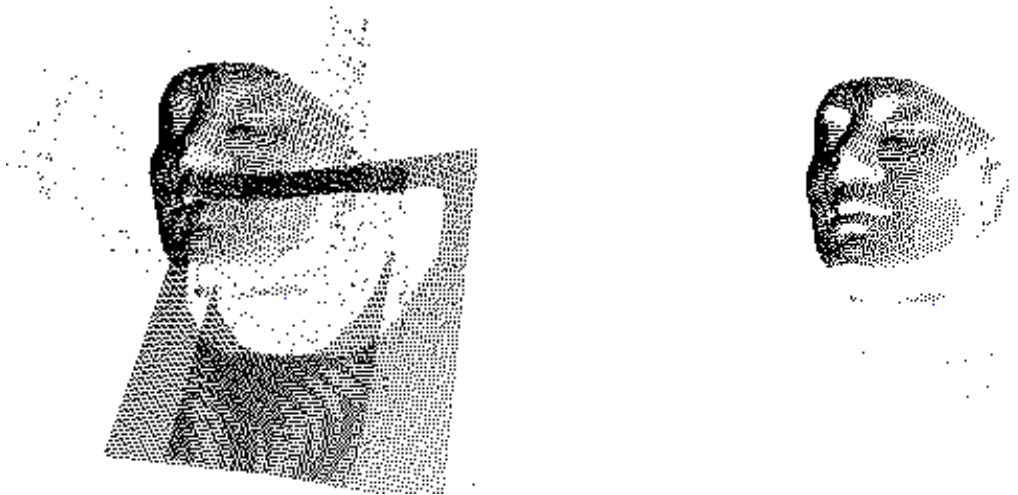


Abb. 4.2a: Ungefilterte (links) und gefilterte (rechts) Meßdaten

4.2 Verfahren zur Bestimmung der Meßgenauigkeit des Scanners

Um eine Aussage über die Qualität der Meßergebnisse machen zu können ist es notwendig sich über die Meßgenauigkeit und mögliche Störquellen klar zu werden. Es wird hier weniger darum gehen die exakte Meßgenauigkeit zu bestimmen, als vielmehr zu überprüfen, ob das eingesetzte Verfahren einem in der sensorgestützten Chirurgie vorläufig maximal toleriertem Fehler von

$$\Delta f_{\max} = 1 \text{ mm}$$

gerechtfertigt wird.

Eine genauere Bestimmung des Fehlers wäre auch in soweit sinnlos, als sich die Genauigkeit der Lichtschnitttechnik mit höheren Auflösungen der Kameras und des Projektors beliebig verbessern läßt.

Entsprechend dem gerade definierten maximal tolerierbaren Fehler muß das Prüfverfahren selber natürlich eine höhere Genauigkeit aufweisen. Als Richtwert soll deshalb für das Prüfverfahren eine Genauigkeit von einem Zehntel des maximal tolerierbaren Fehler gelten, sprich $\Delta f_{\max, \text{Prüf}} = \frac{1}{10} \text{ mm}$.

Der Fehler soll getrennt für die drei Koordinaten bestimmt werden, so daß unterschiedliche Verfahren zu deren Bestimmung zur Anwendung kommen.

Klassischerweise würde man nun versucht sein aus dem Oberflächenscan eines Prüfkörpers bekannter Position, markante Punkte, zum Beispiel die Eckpunkte eines Würfels, zu extrahieren und deren ausgegebenen Koordinaten mit der wahren Position zu vergleichen.

Welche Situation könnte dabei aber entstehen?

In Abbildung 4.2b ist das CCD-Raster der Kamera zusehen, das sich bei der Aufnahme wie ein Gitternetz über das auszumessende Objekt legt. Für jede Masche dieses Netzes wird der mittlere Höhenwert auf der Oberfläche des Meßobjektes bestimmt und dem Maschenmittelpunkt zugewiesen. Nun könnte es passieren, daß der auszumessender Punkt auf dem Prüfkörper gerade so ungünstig positioniert ist, daß er gerade zwischen 4 CCD-Pixel zum Liegen kommt. In der Punktwolke wird man aber einen dieser 4 Nachbarpixel als den zu suchenden Punkt identifizieren und somit einen maximalen Fehler von einem halben Abtastabstand $\delta/2$ auf der x- und y-Koordinate erhalten. Er würde beim vorliegenden Scanner bis zu 1,5 mm betragen.

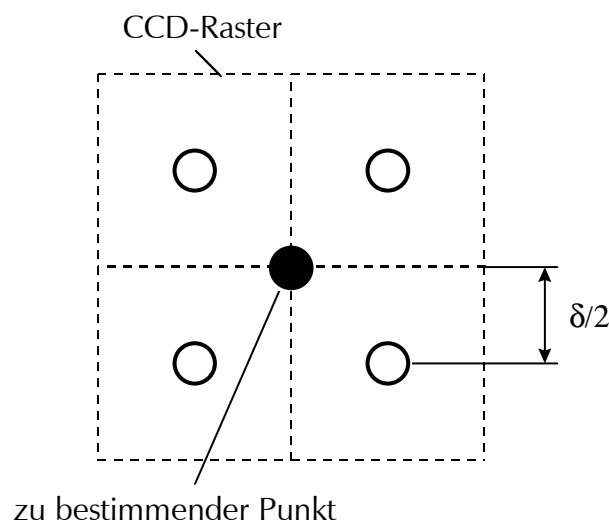


Abb. 4.2b: Abtastfehler des Pixelrasters.

Wir haben zwar somit theoretisch den maximalen Abtastfehler bei der Bestimmung der Koordinaten eines ganz bestimmten Punktes auf der Oberfläche eines Objektes ermittelt, aber dieser Fehler hat für uns keine Relevanz, da wir uns für den Fehler der vom Scanner "ausgesuchten" Punkte interessieren. Das heißt wir müssen den Fehler untersuchen mit dem der Scanner die Koordinaten des Oberflächenmittelpunktes eines Elementarsensors bestimmt hat.

Unter Meßgenauigkeit ist also hier die Genauigkeit zu verstehen, mit der einzelne vom Scanner gelieferte Punkte auch wirklich auf der abgetasteten Oberfläche gelegen haben.

Auch das Schwerpunktverfahren, wie es bereits bei der Kalibration zur Anwendung kam, wird den gesuchten Fehler nicht bestimmen können. Zur Erinnerung: Man geht hin und bestimmt jeweils die Punkte aus einer Punktwolke, die zu einer Teiloberfläche wie zum Beispiel einen kreisförmigen Ausschnitt gehören. Aus den Koordinaten der ausgewählten Punkte bestimmt man den Schwerpunkt der Fläche in der Punktwolke und vergleicht ihn mit der bekannten Position des Schwerpunktes auf dem Prüfobjekt. Wie man leicht nachrechnen kann erhält man aber wieder einen Abtastfehler, der aber nicht wie man vielleicht erwartet hätte mit der Größe der Fläche oder der Anzahl der in der Fläche eingeschlossenen Punkte abnimmt, sondern konstant maximal einen viertel Abtastabstand $\delta/4$ beträgt.

4.2.1 Verfahren zur Bestimmung der Genauigkeit der z-Koordinate

Hierzu kann man hingehen und ebene Fläche konstanter Höhe in den Meßraum hineinführen. Von diesen Ebenen führt man einen Oberflächenscan durch. Die gemessenen Punkte müssen also alle einen konstanten z-Wert aufweisen der zudem noch der eingestellten Höhe entsprechen muß. Wiederholt man nun diese Messung in verschiedenen Höhen, so kann man mittels weniger Aufnahmen Aussagen über die Genauigkeit der z-Koordinate quer durch den gesamten Meßraum treffen, indem man zum Beispiel die Abweichung der gemessenen von der eingestellten Höhe in Abhängigkeit der drei Koordinatenrichtungen betrachtet.

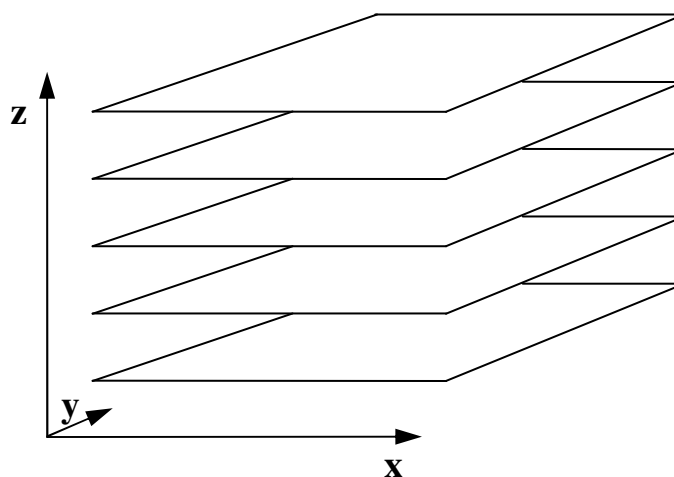


Abb. 4.3: Ebenen konstanter Höhe des Scanners (3D-Darstellung)

Praktisch ist dieses Verfahren sehr leicht umsetzbar indem man sich der weißen Kalibrationsplatte und des Kalibrationsplattenhalters zu deren Positionierung bedient. Diese Vorgehensweise hat zwei entscheidende Vorteile: Durch mechanische Anschläge ist die Platte sehr genau relativ zum Kalibrationsplattenhalter positionierbar. Zudem kann die Platte durch den Kalibrationsplattenhalter genau in eine Ebenen konstanter Höhe gebracht werden, da die Platte bei der Kalibration ja selber die Ebenen konstanter Höhe definiert hat.

Zwei Tatsachen unterscheiden das vorgeschlagene Verfahren allerdings von dem Vorgehen bei der Kalibration: Erstens werden nicht die von den Kameras aufgenommenen Grauwertbilder ausgewertet, sondern die Punktliste der auf dem Prüfobjekt liegenden Punkte und zweitens wird die Platte nicht wieder exakt in die Kalibrationsebenen ($z = 0, 40, 80, 120$ und 160 mm) gelegt. Letzteres wäre nämlich in der Tat unzulässig, da diese Ebenen durch die Kalibration gerade die Stützstellen der Höhenberechnung darstellen. Der Fehler wäre hier also Null, respektive minimal. Statt dessen wird die weiße Platte genau in die Zwischenebenen ($z = 20, 60, 100$ und 140 mm) gelegt indem der ganze Kalibrationsplattenhalter um 20 mm angehoben wird.

4.2.2 Verfahren zur Bestimmung der Genauigkeit der x- und y-Koordinate

Die Bestimmung der Genauigkeit der x- und y-Koordinate ist bei weitem nicht so einfach wie das bei z der Fall war. Was müßte man nämlich tun um den gesuchten Positionierungsfehler eines Punktes zu bestimmen?

Da der Scanner mit jedem Kamerapixel einen Oberflächenpunkt bestimmt, müßte man die Position des Mittelpunktes des Sichtbereiches eines Pixels auf einer abgetasteten Oberfläche bestimmen und diese Position mit den vom Scanner gelieferten Werten vergleichen. Oder anderes ausgedrückt, man müßte ein Objekt in den Meßraum hineinbringen, das exakt ein Pixel beleuchtet. Die Koordinaten des vom Scanner auf der Oberfläche dieses Objektes bestimmten Punktes, könnten dann mit der Position des Mittelpunktes der Objektoberfläche verglichen werden.

Wie leicht einzusehen ist, ist dieses Verfahren wenig praktikabel. Abgesehen von dem enormen zeitlichen Aufwand den man betreiben müßte um Aussagen über den gesamten Meßraum zu machen, ergäben sich starke Probleme bei der Positionierung des Meßobjektes, da der Koordinatenursprung des Scanners sehr schwer mit der geforderten Genauigkeit erfaßbar ist. Dabei hätte man dann noch immer vernachlässigt, daß der Scanner ja jeweils vier Kamerabildpixel zur Bestimmung eines Oberflächenpunktes zusammenfaßt.

Leider gibt es aber keine Alternativen zu diesem Verfahren. Es muß also daraus geschlossen werden, daß es zwar möglich aber nicht praktikabel sein wird einen absoluten Fehler für die x- und y-Koordinate zu bestimmen.

Verfahren zur Bestimmung des globalen Positionierungsfehlers einer Punktwolke

Genauso wie es später für die z-Koordinate getan wird, könnte man aber versuchen, mit Rücksicht auf das später verwendete Matchingverfahren, einen globalen Positionierungsfehler der Punktwolke zu bestimmen. Hierunter wird der Fehler verstanden, um den die ganze Punktwolke als Gesamtheit im Mittel falsch positioniert wurde.

Aus der Funktionsweise des Scanners ist ja bekannt, daß es eine starre Bindung zwischen dem ermittelten z- und den xy-Werten gibt (s. Abschn. 4.1.1). Es ist gerade die Transformation mit der man die x- und y-Werte aus dem z-Wert bestimmt. Der Fehler von den x- und y-Werten setzt sich also zusammen aus dem Bestimmungsfehler der Höhe z und dem Berechnungsfehler der Transformation. Da der Berechnungsfehler der Transformation aber keiner Statistik unterliegt wie der z-Fehler (s. Abschn. 4.3.1), wird der Fehler von x und y ähnlich verteilt sein wie der von z.

Würde also das Matching dazu führen, daß die Streuung der Koordinaten ohnehin “herausgemittelt” würde, so würde ein Verfahren zur Bestimmung des globalen Positionierungsfehlers das dies ebenfalls täte, durchaus eine sehr wertvolle Aussage liefern können. In der Tat genügt das später zur Anwendung kommende Matchingverfahren durch die Wahl der Fehlerfunktion dieser Anforderung. Somit kann man sich folgendes Verfahren zur Bestimmung des oben genannten Fehlers überlegen:

Man legt zum Beispiel eine um die y-Achse gedrehte Ebenen in den Meßraum hinein. Diese mißt man mit dem Oberflächenscanner aus. Kennt man die genaue Position der Ebene, zum Beispiel durch eine Hochpräzisionsverschiebeeinrichtung, so ergibt sich eine für diese Ebene charakteristische Beziehung zwischen z und x, nämlich gerade eine Ebenengleichung der Form:

$$z = a \cdot x + b$$

Dabei sind a und b durch die Verschiebeeinrichtung bekannt. Da aber sowohl z als auch x um die wahren Werte streuen, kann man keine direkte Folgerung von dem Fehler auf z auf den Fehler von x ziehen. Statt dessen könnte man aber durch die Methode der kleinsten Quadrate (LMS) die Ebenenparameter aus allen gemessenen Punkten schätzen und mit den exakten Werten verglei-

chen. Die Abweichung der geschätzten Position zu der wahren ist dann gerade der globale Positionierungsfehler der Punktwolke in x-Richtung.

Aber auch dieser Weg wurde, aufgrund der Positionierungsschwierigkeiten zu dem frei schwebenden Koordinatenursprung, sowie mangels Zeit, in der vorliegenden Arbeit nicht verfolgt. Statt dessen wurde ein Gütekriterium entwickelt, das eine Aussage über die Qualität des Scanners erlaubt und sich auf die vorherigen Erkenntnisse stützt.

Entwicklung eines Gütekriteriums

Ähnlich dem im vorherigen Abschnitt beschriebenen Verfahren bringt man, um einen bekannten Winkel, verdrehte Ebenen in den Meßraum hinein. Dabei ist die genaue Position der Ebene weniger interessant. Der Winkel soll nun mit Hilfe des Oberflächenscanners geschätzt werden. Die relative Abweichung des geschätzten Winkels zu dem wahren stellt ein integrales Gütemaß dar.

Praktisch realisieren kann man das Verfahren wieder mit Hilfe des Kalibrationsplattenhalters und der weißen Kalibrationsplatte. Die Neigung der Platte um eine Achse, erzielt man, indem sie links und rechts auf Halter unterschiedlicher Höhen aufgelegt wird (s. Abb. 4.4). Durch diese Maßnahme dreht man die Platten gerade um die Halterachsen. Da das Ringmuster aber exakt parallel zu den Außenkanten der Platte aufgeklebt ist, stimmt dieses Achse mit der durch die Spaltenrichtung der Ringe definierten y-Achse überein (s.a. Abb. 4.2).

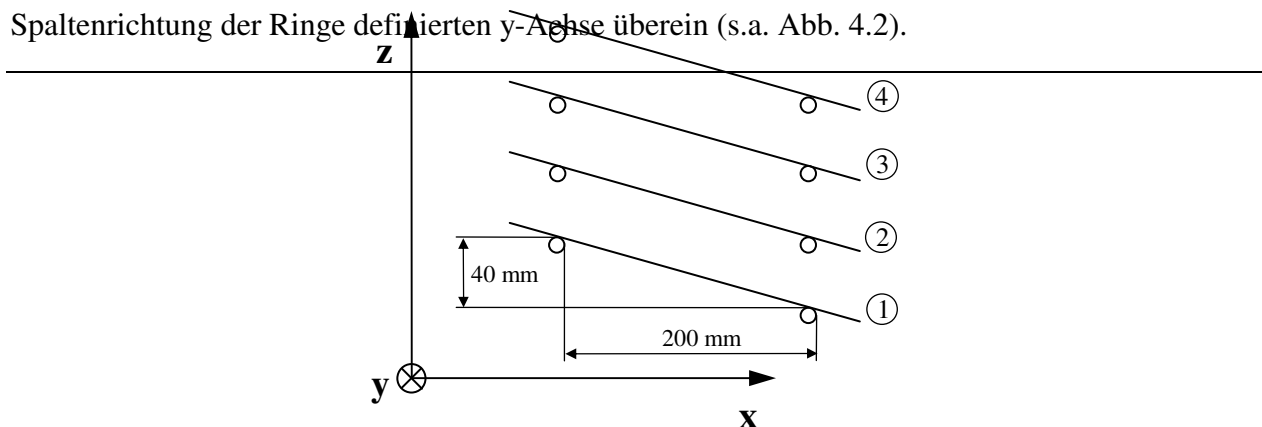


Abb. 4.4: Um die y-Achse gedrehte schiefe Ebenen (2D-Darstellung)

Die Schätzung des Winkels wird sowohl für eine Verkipfung um die y- als auch um die x-Achse und bei verschiedenen Höhen vorgenommen. Danach soll die Steigung der Ebene aus allen Meßpunkten nach der Methode der kleinsten Quadrate berechnet werden.

Für einen normalverteilten Fehler läßt sich diese Steigung nach folgender Formel geschlossen berechnen [Bronstein 91 (Abschn. 7.1.5)]:

$$\hat{a} = \frac{\frac{1}{n} \cdot \sum_{i=1}^n x_i \cdot z_i - \bar{x} \cdot \bar{z}}{\frac{1}{n} \cdot \sum_{i=1}^n x_i^2 - \bar{x}^2} \quad (\text{Gl. 4.1})$$

Hierbei sind x_i und z_i die Koordinaten der gemessenen Punkte, sowie \bar{x} und \bar{z} deren Mittelwerte.

Weiter oben wurde bereits gefolgert, daß der x- und y-Fehler die gleichen Verteilungen wie der z-Fehler haben müssen. Wie in Abbildung 4.6 zu sehen ist, ist diese Verteilung effektiv annähernd normalverteilt, so daß Gleichung 4.1 zur Anwendung kommen kann.

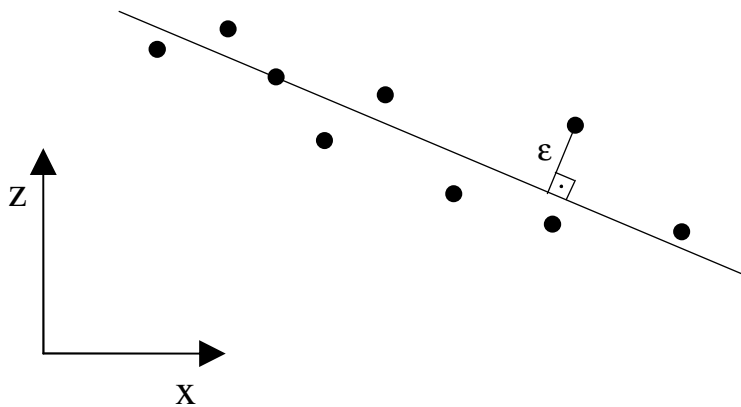


Abb. 4.5: Schätzung der Steigung einer Ebene aus n Meßpunkten durch das LMS-Verfahren

4.3 Ergebnisse der Fehlerbestimmung

Folgendes Meßprogramm wurde zur Datengewinnung durchgeführt:

1. Kalibration
2. Aufnahme der weißen Platte in den Höhen $z = 20, 60, 100$ und 140 mm
3. Aufnahme der um die y-Achse verkippten weißen Platte bei 4 verschiedenen Höhen
4. Aufnahme der um die x-Achse verkippten weißen Platte bei 4 verschiedenen Höhen

Bei *Punkt 1-4* lagen dabei folgende Meßbedingungen vor:

Der Oberflächenscanner ist zu allen vier Seiten mit Matten aus schwarzem Zellkautschuk abgehängt. Dies verhindert erfolgreich jeden seitlichen Lichteinfall und schafft somit bereits exzellente Meßbedingungen. Zum Abführen der vom Projektor abgegebenen Wärme muß der Meßaufbau nach oben hin offen bleiben. Die Neonlichtreihe die direkt über dem Aufbau verläuft ist allerdings ausgeschaltet. Alle Aufnahmen wurden mit einer Auflösung von $\frac{1}{4}$ (s. Abschn. 4.1.4) aufgenommen.

Wie in Abschnitt 4.2.1 bereits erwähnt, werden die Höhen $z = 20, 60, 100$ und 140 mm dadurch eingestellt, daß der ganze Kalibrationsplattenhalter um 20 mm angehoben wird. Dies wurde mit 4 Leiterplattenabstandshaltern erreicht. Die genauen Längen betrugen: $19.955, 19.925, 19.925$ und 19.895 mm. Als Mittelwert ergibt sich also 19.925 mm. Bei den Berechnungen werden dementsprechend die genauen Sollhöhen $19.925, 59.925, 99.925$ und 139.925 mm herangezogen.

4.3.1 Ebenen konstanter Höhe

Zur Auswertung der unter *Punkt 2* gemachten Aufnahmen, werden die Punktwolken im ASCII-Format abgespeichert. Zur weiteren Reduktion der nicht unerheblichen Datenmengen (eine Aufnahme mit einer viertel Auflösung belegt immerhin 524 kByte) wird aus der Punktwolke nur jeder sechste Punkt in die Auswertung einfließen gelassen. Damit werden immerhin noch über 1000 Punkte pro Kamera berücksichtigt. Die Auswirkungen auf die Statistik sind vernachlässigbar da 1000 Punkte von 6000 ohne Zweifel eine repräsentative Menge darstellen.

Erstellt man nun von den restlichen $2 \cdot 1000$ Punkten ein Histogramm über die gemessenen Höhenwerte mit einer Klassenbreite Δz von 0.04 mm, so ergibt sich folgendes Bild:

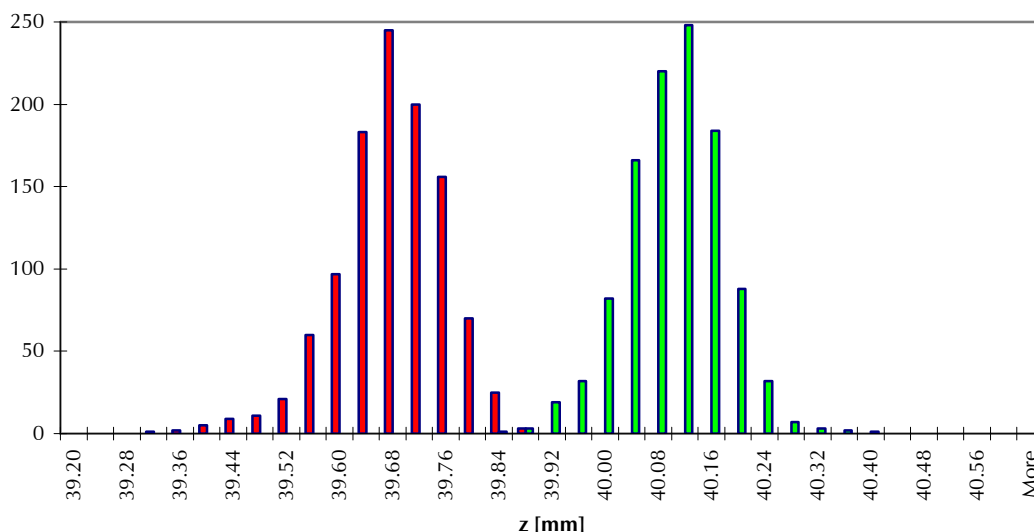


Abb. 4.6: Histogrammverteilung der aufgenommenen Punkte einer Ebene $z = \text{const.}$

Rot eingezeichnet ist die Verteilung der Punkte der rechten Kamera, grün die der linken Kamera. Man erkennt deutlich den Gaußschen Charakter der Verteilung.

Im Anhang A befindet sich die detaillierte statistische Auswertung aller 4 aufgenommenen Ebenen, während hier nur die Interpretation der Ergebnisse erfolgt.

Betrachtet man die Verteilungen genauer, so stellt man fest, daß die Standardabweichung zum Projektor hin geringer wird. In Abbildung 4.7 ist dazu die Standardabweichung des rechten Kanal in Abhängigkeit von der Ebenenhöhe dargestellt. Interessehalber wurden dabei die Standardabweichungen in den Kalibrationsebenen noch hinzugefügt (s. a. Abschn. 4.2.1). Ebenfalls im Diagramm enthalten ist der Verlauf über die Differenz zwischen den Mittelwerten des rechten und linken Kanals (Kanaldivergenz).

Man erkennt, daß die Standardabweichung wie erwartet für die Kalibrationsebenen jeweils minimal wird, aber sowohl in den Kalibrationsebenen als auch in den Zwischenebenen zum Projektor hin abnimmt. Dies läßt sich dadurch erklären, daß die Linienbreite der projizierten Streifenmuster mit abnehmendem Abstand zum Projektor geringer wird und dadurch die Auflösung steigt.

Für die Kanaldivergenz kann man dagegen keine sicher Aussage über die Abhängigkeit von z machen.

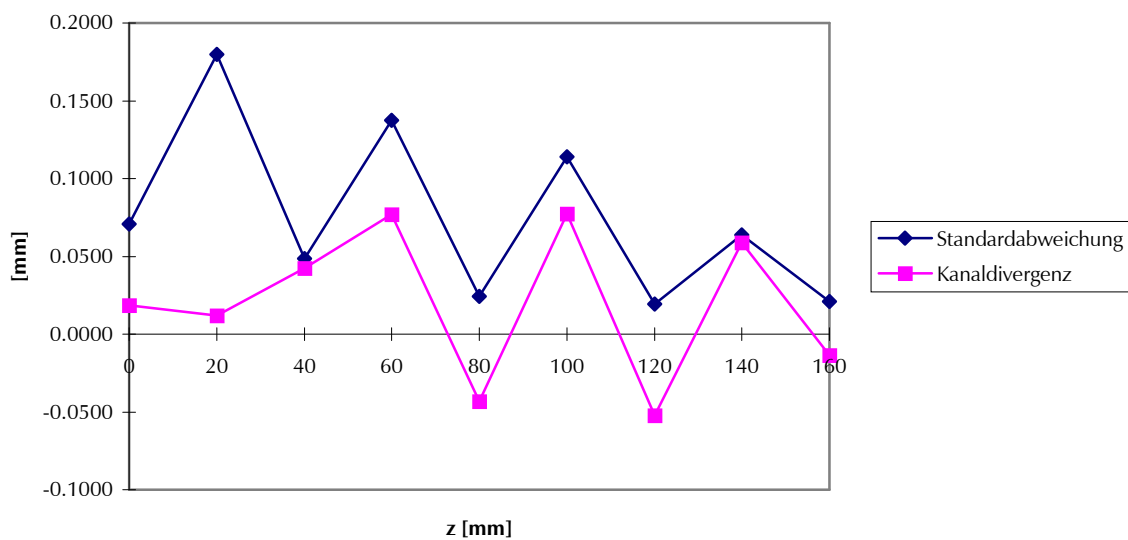


Abb. 4.7: Standardabweichung (rechter Kanal) und Teilpunktwellenabstand als Funktion von z .

Nachdem soweit die globale Verteilung des Fehler betrachtet wurde, findet nun eine Untersuchung der lokalen Abhängigkeit des Fehler statt. Dazu wird zunächst die Abweichung des Istwertes jedes Punktes von seinem Sollwert in Abhängigkeit von seinem x-Wert analysiert.

In Abbildung 4.8 ist der Fehler jedes Punktes des rechten Kanals der $z = 140$ mm Ebene als Funktion seines x-Wertes eingetragen.

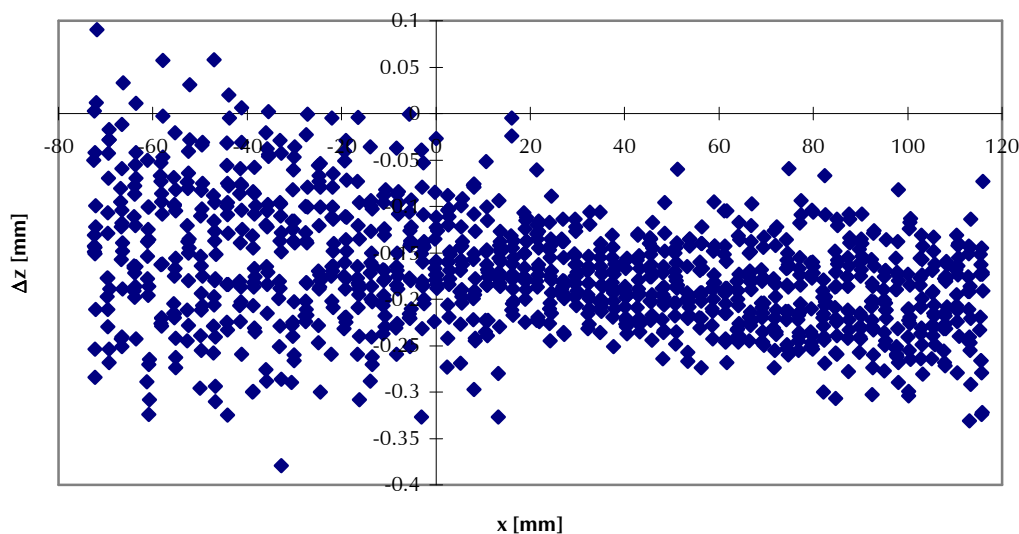


Abb. 4.8: Fehler jedes Punktes des rechten Kanals der $z = 140$ mm Ebene als Funktion seines x-Wertes.

Es ist zu sehen, wie die Streubreite des Fehlers von negativen zu positiven x-Werten zuerst abnimmt um dann wieder leicht zuzunehmen. Dies läßt sich auf zwei Effekte zurückführen. Erstens nimmt die Linienbreite, der ungefähr parallel zur y-Achse verlaufenden Linien, zu betragsmäßig hohen x-Werten zu, da der Abstand der Platten zum Projektor an den Seiten etwas zunimmt. Zweitens sieht die Kamera die Linien zu kleiner werdenden x dünner aber auch unschärfer, da in dieser Richtung der Abstand der Platte zur rechten Kamera zunimmt. Der Einfluß der Unschärfe scheint dabei zu überwiegen.

Strengt man die gleiche Analyse für den Fehler in Abhängigkeit vom y-Wert an, so ist kaum eine Abhängigkeit zu erkennen (s. Abb. 4.9). Da sich die Linienbreite in dieser Richtung auch nicht ändert war dies auch nicht anderes zu erwarten.

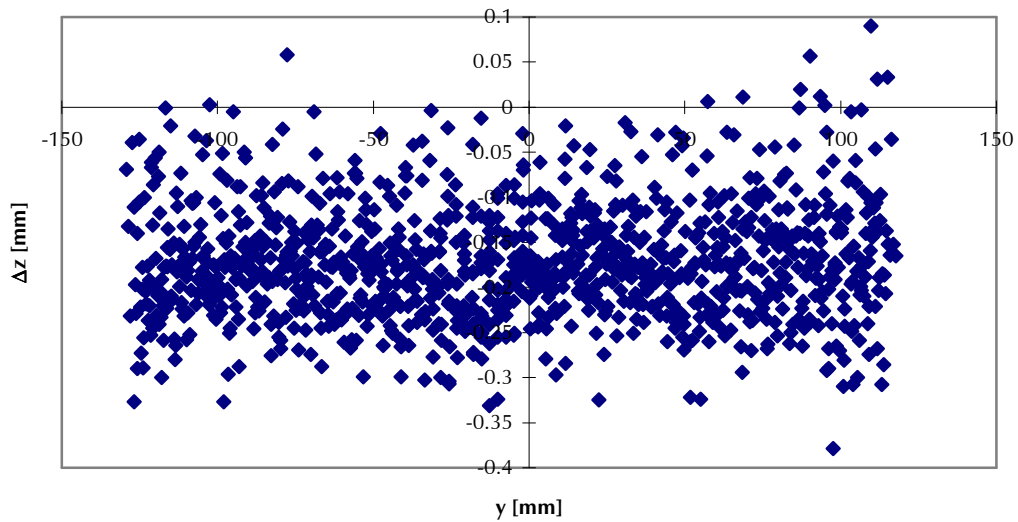


Abb. 4.9: Fehler jedes Punktes des rechten Kanals der $z = 140$ mm Ebene in Funktion seines y -Wertes.

Bleibt noch die Betrachtung des Verlaufs der Abweichung des Höhenmittelwertes der gesamten Punktwolke vom Sollwert in Abhängigkeit von z zu betrachten.

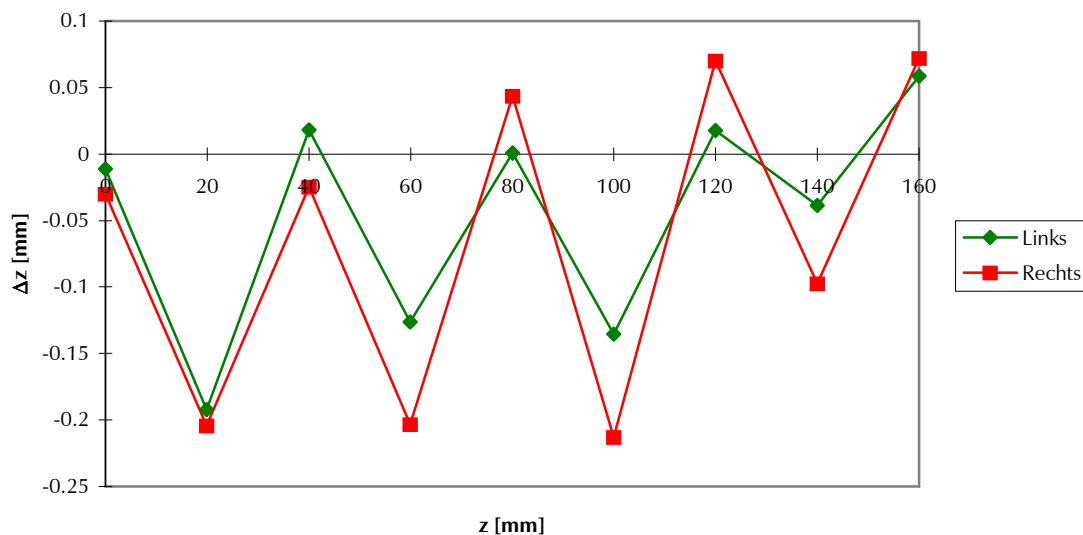


Abb. 4.10: Abweichung des Höhenmittelwertes der Punktwolke vom Sollwert in Funktion von z .

In rot ist die Kurve der aus der rechten Punktwolke berechneten Mittelwerte gezeichnet, in grün die der linken Punktwolke. Man erkennt wieder wie der Fehler für die Kalibrationshöhen durchgängig geringer ist.

Somit können wir zur abschließenden Diskussion der z-Meßreihen übergehen:

Die 3σ -Breite, die 99,73 % aller aufgetretenen Fälle begreift, beträgt für den “worst case” der Ebene $z = 19.925$ mm:

$$3\sigma_{z=19.925\text{mm, rechter Kanal}} = 0.539 \text{ mm}$$

Dieser Wert entspricht dem *maximal zu erwartenden absoluten Fehler pro Punkt* und beträgt:

$$\Delta f_{\text{maxz}} = \pm 0.54 \text{ mm}$$

Betrachtet man die Punktwolke als Gesamtheit, was wir durch die Summation der quadrierten Abstände in der Fehlerfunktion tun werden (s. Abschn. 5.3), wird der Fehler des einzelnen Punktes sekundär. Vielmehr interessiert wie groß der *integrale Fehler über die gesamte Punktwolke* ist, also der Fehler mit der die gesamte Punktwolke im Mittel falsch in der Höhe positioniert wurde. Für ihn gilt die Differenz des Mittelwertes der Verteilung von der Sollhöhe.

Der “worst case” ist hier bei der Höhe $z = 99.925$ mm zu finden und beträgt:

$$\Delta f_{\text{maxz,int}} = \pm 0.21 \text{ mm}$$

Dieser Fehler stößt damit fast in einen Bereich vor in dem wir nicht mehr sicher sagen können, ob die Ursache beim Scanners oder beim Positionierungsfehler der Platte zu suchen ist. In Abschnitt 4.2 hatten wir ja eine Genauigkeit von $\frac{1}{10}$ mm für das Prüfverfahren gefordert. Der Scanner ist also wahrscheinlich noch genauer nur sind wir an die Grenzen des Prüfverfahrens gelangt.

Nichts desto trotz erfüllt der Scanner für die z-Koordinate eindeutig die Maßgabe genauer als 1 mm zu sein. Das war gefordert.

4.3.2 Schiefe Ebenen

Wie in 4.2.2 beschrieben wurden jeweils 4 um die y- und x-Achse verdrehte Ebenen aufgenommen. Zur Datenreduktion wurde wieder nur jeder 6. Punkt ausgewertet. Die Details findet man wie gehabt in Anhang A.

Steigung in x-Richtung

Die Platten wurden in die 4 Positionen, wie in Abbildung 4.4 zu sehen, gebracht und ausgemessen. Für jede der Ebenen wurde die Steigung nach dem in Abschnitt 4.2.2. beschriebenen Verfahren geschätzt und der relative Fehler zum Sollwert bestimmt. Die prozentualen Abweichungen der Meßwerte vom Sollwert ist in Abb. 4.11 gezeigt.

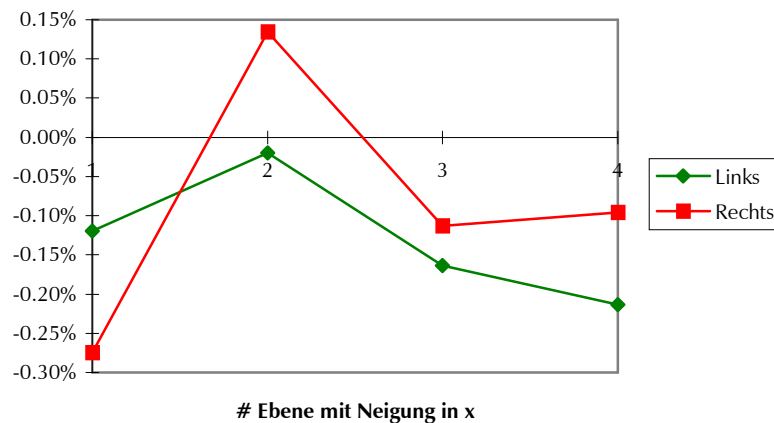


Abb. 4.11: Prozentuale Abweichung der Steigung in x-Richtung (Ebene 1 = unten, 4 = oben).

Die Höhenunterschiede der Auflagestützen betragen gerade 40 mm, ihr Abstand in x-Richtung 200 mm. Daraus ergibt sich eine Sollsteigung von:

$$\tan \alpha_{\text{soll},x} = 0,2$$

Der maximale Fehler bei der Bestimmung der Steigung in x-Richtung betrug:

$$f_{\text{max},x} = 0,27 \%$$

Steigung in y-Richtung

Für die y-Richtung wurde analog vorgegangen. Die prozentualen Abweichungen vom Sollwert ist hier in Abb. 4.12 gezeigt.

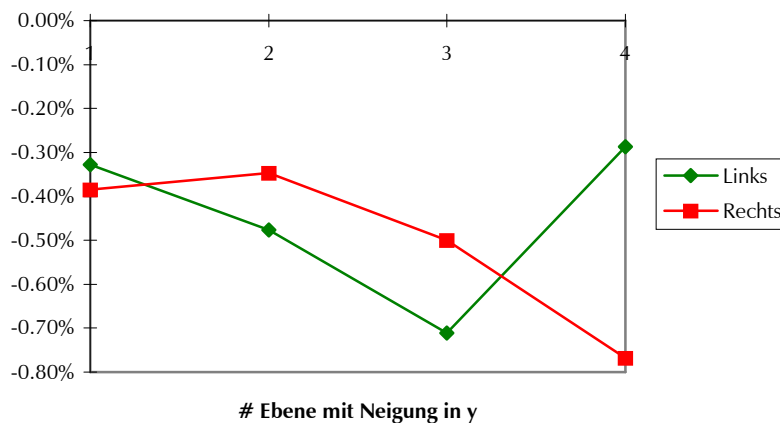


Abb. 4.12: Prozentuale Abweichung der Steigung in y-Richtung (Ebene1 = unten, 4 = oben).

Die Höhenunterschiede der Auflagestützen betragen weiterhin 40 mm, ihr Abstand in y-Richtung 491 mm. Daraus ergibt sich eine Sollsteigung von:

$$\tan \alpha_{\text{soll},y} = 0,0814$$

Der maximale Fehler bei der Bestimmung der Steigung in y-Richtung betrug:

$$f_{\text{max},y} = 0,77 \%$$

4.3.3 Schlußfolgerung

Man muß betonen, daß die Resultate keine Schlußfolgerungen auf den absoluten Fehler in x- oder y-Richtung zulassen. Alle Aussagen diesbezüglich wären reine Mutmaßungen.

Welche Schlußfolgerungen können aber für den praktischen Einsatz des Scanners aus dem Gütemaß gezogen werden. Aufgabe des Scanners ist es geometrische Formen wie zum Beispiel ein Gesicht möglichst gut wiederzugeben, denn dies ist für die Qualität des Matchings von entscheidender Bedeutung. Die schiefe Ebene ist nun genau eine solche geometrische Form die der Scanner, wie aus dem erzielten Gütemaß zu ersehen ist, mit verblüffender Genauigkeit zu erfassen imstande ist.

Nach dieser ausführlichen Betrachtung des Scannerfehlers lenken wir nun unser Augenmerk auf die Störempfindlichkeit des Scanners. Da es sich beim Aufnahmeverfahren um ein optisches Verfahren handelt ist mit Schwierigkeiten bei dunklen Oberflächen und bei zu hoher Umgebungshelligkeit zu rechnen. Deshalb wurden einige Aufnahmen mit zunehmender Oberflächenschwärze und Umgebungshelligkeit gemacht.

Die Auswertung erfolgte wie in allen Messungen davor mit einer Auswahl von 1000 Punkten pro Punktwolke.

4.4 Untersuchung der Störeinflüsse

Oberfläche zunehmender Schwärze

Um die zunehmende Schwärzung zu simulieren wurde mit unterschiedlich eingefärbten Oberflächen gearbeitet. Hierzu wurde zuerst die braune Rückseite der weißen Kalibrationsplatte in der Höhe $z = 160$ mm ausgemessen, danach wurde auf die Platte ein ebenes Stück mattschwarzes Papier gelegt. Es lagen die gleichen Meßbedingungen vor wie in den Meßreihen davor.

Im folgenden Diagramm ist der Verlauf der 3σ -Breite über drei Aufnahmen mit diskret zunehmender Oberflächenschwärze dargestellt. Hierbei wurde die erste Aufnahme unter optimalen Bedingungen durchgeführt.

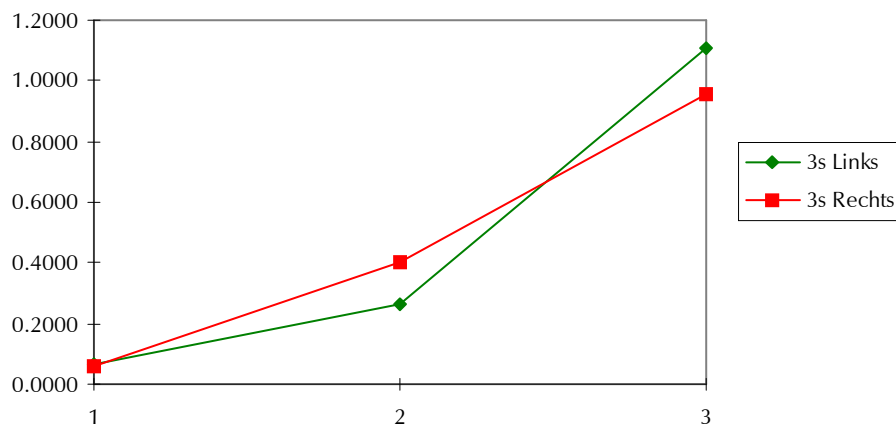


Abb. 4.13: 3 Aufnahmen mit zunehmender Oberflächenschwärze.

Wie erwartet steigt die Streuung der Punkte und damit der maximal zu erwartende absolute Fehler pro Punkt stark an.

Aufnahmen unter zunehmendem Lichteinfall

Um durch diese Untersuchung gleichzeitig die praktische Notwendigkeit der in den Meßbedingungen beschriebenen Maßnahmen zur Abdunklung zu demonstrieren, wurde zuerst die Neonlichtreihe über dem Scanneraufbau eingeschalten, danach die vordere Matte entfernt, um dann auch die restlichen Neonlichtreihen in der Versuchshalle zuzuschalten. Abschließend wurden auch die seitlichen Matten entfernt was allerdings zu keiner verwertbaren Aufnahme mehr führte.

Im folgenden Diagramm ist wiederum der Verlauf der 3σ -Breite über die 4 Aufnahmen mit zunehmendem Lichteinfall dargestellt. Die erste Aufnahme ist wieder eine Aufnahme unter optimalen Bedingungen.

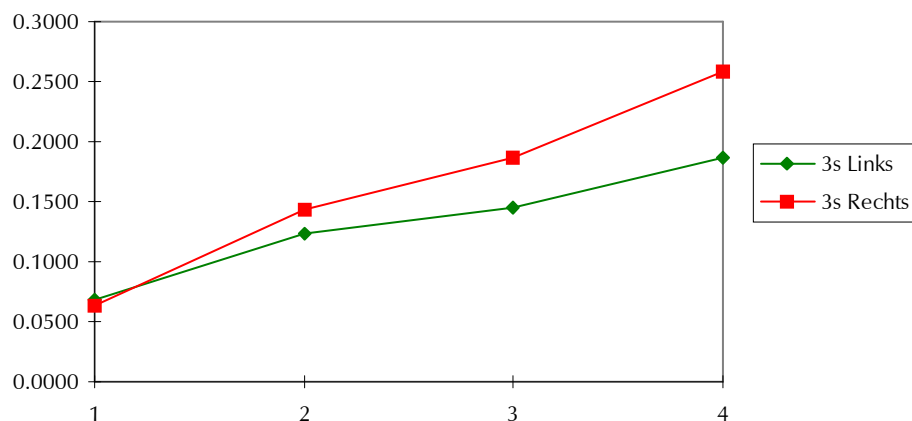


Abb. 4.14: 4 Aufnahmen mit zunehmendem Lichteinfall.

Wiederum steigt die Streuung der Punkte und damit der maximal zu erwartende absolute Fehler pro Punkt mit zunehmender Umgebungshelligkeit an.

4.5 Abschließende Betrachtung

Der Oberflächenscanner hat sich als Aufnahmeverfahren bewährt. Trotz abnehmender Genauigkeit bei ungünstigen Betriebssituationen ist es dennoch oft erstaunlich, unter welchen Bedingungen der Scanner überhaupt noch imstande ist Punkte von der Oberfläche zu liefern. Für den Einsatz im OP wird es aber wahrscheinlich nötig sein die Umgebungshelligkeit und den OP-Strahler für die kurze Zeit der Aufnahme abzudunkeln.

Das wichtigste Tool für den praktischen Einsatz ist ohne Zweifel das Kontrastfilter. Es wäre nämlich sehr ungünstig gewesen Punkte von sehr stark veränderlichen Oberflächen wie den Haaren oder der Kleidung zu erhalten. Da das Kontrastfilter aber Punkte, die von dunklen Oberflächen oder sehr feinen Strukturen stammen, sicher entfernt, erhält man ohne großen Aufwand eine Punktwolke hoher Übereinstimmung mit dem Volumenmodell. Eine Kontrasterhöhung von 50-60% ist in den Fällen guter Aufnahmebedingungen ausreichend. Notfalls bietet die Software immer noch die Möglichkeit Punkte manuell wegzuschneiden.

Die in Zukunft steigende Auflösung der LCD-Projektoren und der CCD-Kameras wird dazu führen, daß man auch von feinen Strukturen sicher Punkte erhalten können wird. Dies ist für die hiesige Anwendung aber nicht wünschenswert. Trotzdem wird man die höheren Auflösungen zur weiteren Rauschunterdrückung (s.a. Abschn. 4.1 Einleitung) nutzen können.

Wichtig ist auch zu betonen, daß eine sichere Aufnahme von Punkten auf dunkler Haut sicher nicht unproblematisch ist. Vielleicht kann man hier mit weißer Kreide, oder weißer Farbe nachhelfen. Mangels Probanden konnten in dieser Richtung allerdings keine Versuche gemacht werden.

Probleme gab es dagegen mit der Langzeitstabilität des vorliegenden Scanners. Nach dem aktuellem Kenntnisstand kommt es, durch einem Auseinanderdriften der Synchronisation zwischen Grafikkarte und LCD-Projektor, ebenso zu einem kontinuierlichen symmetrischen Auseinanderdriften der Teilpunktwolken um die Sollhöhe. Dieser Drift nimmt Werte um mehrere zehntel Millimeter an. Nach mehreren Stunden steigt der Höhenabstand zwischen den Teilpunktwolken dann sprunghaft auf 26 mm an und macht eine Neukalibration erforderlich.

Hierbei handelt es sich aber um ein reines Hardwareproblem des in dieser Arbeit verwendeten Scanners. Es folgt daraus keine allgemeine Beschränkung der Anwendbarkeit der eigentlichen Aufnahmetechnik.

Aufgrund der langwierigen Suche nach der Ursache des Fehlers mußten aber die Meßreihen mehrmals komplett wiederholt werden um eine Reproduzierbarkeit der Resultate zu garantieren.

Kapitel 5

Matchingverfahren

5. Matchingverfahren

Nachdem der Oberflächenscanner im vorherigen Kapitel ausgiebig behandelt wurde, wird nun der Fokus auf das verwendete Matchingverfahren gelegt. Erklärtes Ziel des Matchings war es, die Punktwolke mit der Oberfläche des Volumenmodells bestmöglich in Übereinstimmung zu bringen, um daraus die Koordinatentransformation zwischen beiden Systemen zu bestimmen.

Bildlich stellt sich diese Aufgabe wie folgt dar:

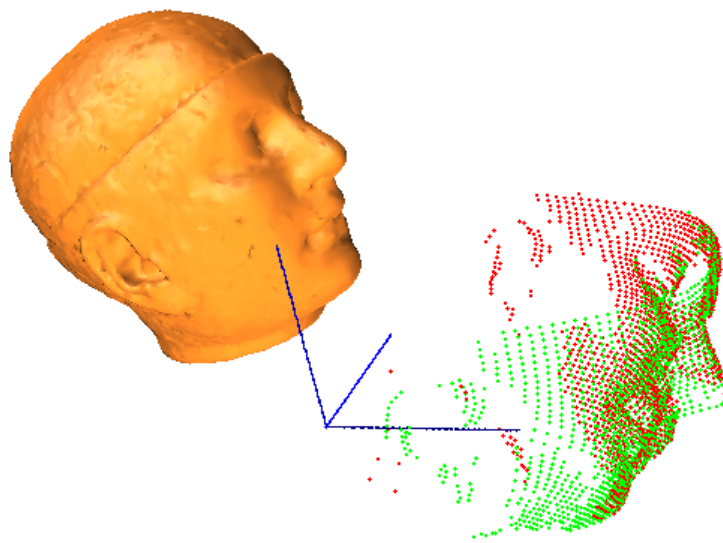


Abb. 5.1: Ausgangslage der Punktwolke relativ zu dem Volumenmodell

Zu sehen ist die Lage der Punktwolke relativ zu dem korrespondierenden Volumenmodell, wenn man die Koordinatensysteme der beiden Systeme übereinanderlegt. Die vorliegende Orientierung entsteht dadurch, daß der Oberflächenscanner längs der z-Achse auf das Gesicht des Kopfes “geblickt” hat, das dem Volumenmodell zugrundeliegende CT aber in Vorschubrichtung (z-Achse) Längsschnitte von hinten nach vorne durch den Kopf erstellt hat.

Nichts desto trotz ist dies eine praxisrelevante Ausgangssituation für das folgende Matching.

5.1 Methodik

Beim Matching soll wie in [Pokrandt 97 (S. 15)] beschrieben vorgegangen werden.

Um den Grad der Übereinstimmung zu quantifizieren ist es dabei in einem ersten Schritt nötig ein Fehlermaß in Form eines Gütekriteriums, einer Kostenfunktion oder einer Fehlersumme zu bestimmen. Diese Fehlerfunktion gilt es dann mit Hilfe eines geeigneten Optimierungsverfahrens zu minimieren. Da die meisten Verfahren versuchen ausgehend von einer Startposition die Fehlerfunktion systematisch zu minimieren, stellen Nebenminima des öfteren Probleme dar. Wie durch Abb. 5.1 leicht einzusehen ist, werden somit die wenigsten Optimierungsverfahren in der Lage sein aus der vorliegenden Startposition heraus ein erfolgreiches Matching durchzuführen. Aus diesem Grund soll eine geschickt gewählte initiale Transformation die Punktwolke in eine gute Ausgangslage für die folgende Optimierung bringen. Letztere wird das Fehlermaß minimieren und damit die Punktwolke in die optimale Lage überführen.

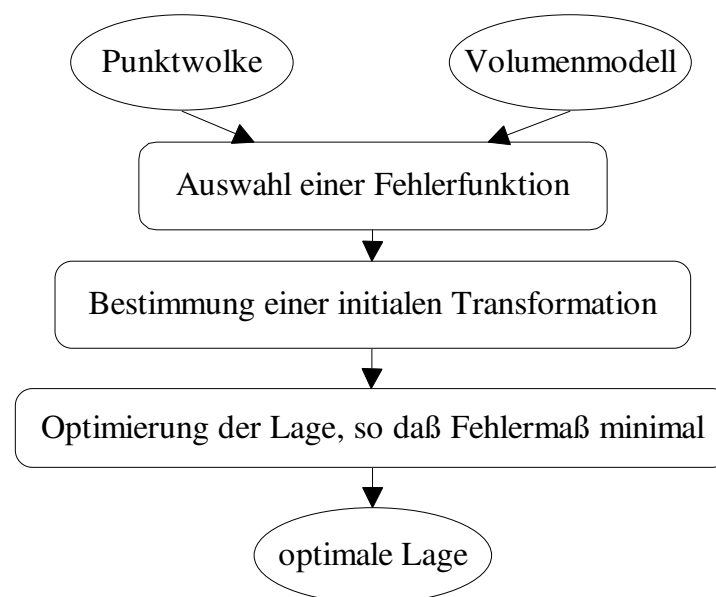


Abb. 5.2: Ablauf des Matching

An dieser Stelle sei noch erwähnt, daß es gleichbedeutend ist, ob man die Punktwolke auf das Volumenmodell matched oder umgekehrt. Die aus dem einen Vorgehen ermittelte Transformation wird gerade die Inverse der anderen darstellen. Aus praktischen Gründen ist es aber sinnvol-

ler die Punktwolke auf das Volumenmodell zu matchen, da hierbei wesentlich weniger Datenmen

gen zu manipulieren sind und damit die Rechenzeit bis zum Erfolg wesentlich reduziert wird. An gegebener Stelle wird noch einmal detaillierter darauf eingegangen.

5.2 Homogene Koordinaten

Wie in der Robotik üblich, werden die Vektoren in homogenen Koordinaten dargestellt. Dabei wird jeder Vektor um eine vierte Komponente erweitert:

$$\vec{v} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

x,y und z entsprechen den kartesischen Koordinaten. Die vierte Komponente wird Skalierungsfaktor genannt. Bei der Transformation zwischen zwei Vektoren in homogenen Koordinaten ist es aber nun im Gegensatz zu den kartesischen Koordinaten möglich, in der Transformationsmatrix gleichzeitig Rotationen, Translationen, Perspektivverzerrungen sowie Skalierungen auszudrücken.

Eine solche 4x4-Transformationsmatrix setzt sich aus den 4 folgenden Untermatrizen zusammen [Dillmann 91]:

$$D = \begin{bmatrix} \begin{pmatrix} R \end{pmatrix} & \begin{pmatrix} T \end{pmatrix} \\ \begin{pmatrix} P \end{pmatrix} & \begin{pmatrix} S \end{pmatrix} \end{bmatrix}$$

- R = 3x3 Rotationsmatrix
- T = 3x1 Translationsmatrix
- P = 1x3 Perspektivtransformationsmatrix
- S = 1x1 Skalierungsfaktor

Da die von uns gesuchte Koordinatentransformation nichtperspektivisch und nichtverzerrend sein soll, muß S immer 1 sein und alle Komponenten der P-Matrix Null sein. Entsprechend ist auch der Skalierungsfaktor der Vektoren immer 1.

Die Transformationsmatrizen zur Rotation um eine der drei Achsen eines festen Koordinatensystems nehmen entsprechend der Achse die folgende Form an:

$$D_{x_\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad D_{y_\beta} = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad D_{z_\gamma} = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

α, β und γ heißen Cardan-Winkel [Meyberg 90].

Eine Translation um den Verschiebungsvektor $\vec{t} = (t_x \quad t_y \quad t_z)^T$ wird durch die Matrix:

$$D_t = \begin{bmatrix} 0 & 0 & 0 & t_x \\ 0 & 0 & 0 & t_y \\ 0 & 0 & 0 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

beschrieben. Die Matrizen $D_{x_\alpha}, D_{y_\beta}, D_{z_\gamma}$ und D_t können zu einer allgemeinen 4×4 -Matrix zusammengefaßt werden:

$$D = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = (D_{x_\alpha} \cdot D_{y_\beta} \cdot D_{z_\gamma}) + D_t$$

Bei der Berechnung einer solch kombinierten Matrix ist darauf zu achten, daß Rotationen nicht kommutativ sind. Bei der dargestellten Berechnungsvorschrift für D würde also zuerst um die z-Achse, dann um die y-Achse und zuletzt um die x-Achse gedreht.

5.3 Fehlerfunktion

Die Fehlerfunktion soll wie bereits erwähnt den Grad der Übereinstimmung zwischen der Oberfläche des Volumenmodells und der von der Punktwolke modellierten Oberfläche für jede relative Position der Punktwolke zum Volumenmodell zahlenmäßig ausdrücken. Die Fehlerfunktion muß also ein integrales Bewertungskriterium über alle Punkte der Punktwolke darstellen. Nur mit hohem Aufwand ist aber eine analytische Berechnung der Fehlerfunktion möglich, also soll die Fehlerfunktion numerisch berechnet werden.

Als Fehlerfunktion bietet sich damit ein Abstandsmaß über alle Punkte an. Tausendfach in der Praxis bewährt hat sich dabei sicherlich das Summenfehlerquadrat. Der Fehler wird dabei in dieser Arbeit mit dem kürzesten euklidischen Abstand eines Punktes zur Oberfläche des Volumenmodells definiert:

$$F = \sum_{i=0}^n d_i^2 = f(t_x, t_y, t_z, \alpha, \beta, \gamma) \quad (\text{Gl. 5.1})$$

Diese Fehlerfunktion ist in der Tat eine Funktion der Lage der Punktwolke und hängt damit von deren 6 Transformationsparametern ab.

Zur Berechnung der einzelnen Abstände wird nun wiederum wie folgt vorgegangen: Damit man nicht jedesmal den zeitaufwendigen Schritt der Abstandsberechnung eines Punktes zur Oberfläche wiederholen muß, soll eine sogenannte “distance map” angelegt werden. Dies ist eine Tabelle in der für eine diskrete Anzahl an Stützstellen im Raum die Abstände zu der gewählten Oberfläche gespeichert sind. Die “distance map” wird durch einen, am Institut für Prozeßrechentechnik und Robotik entwickelten, Algorithmus berechnet und in Form eines Distanztomogramms abgespeichert. Wie der Name suggeriert handelt es sich beim Distanztomogramm um die Speicherform eines normalen Tomogramms. Statt Grauwerte enthält das Tomogramm aber gerade die Distanzwerte zu der gewählten Oberfläche.

Die Voraussetzung für die Berechnung des Distanztomogramms ist natürlich die vorherige Extraktion der Oberfläche aus dem Volumenmodell.

5.3.1 Extraktion der Oberfläche aus dem Volumenmodell

Da das Volumenmodell aus einem 3D-CT-Datensatz generiert wird [Grabowski 96] bietet es sich an auch die Oberfläche aus dem CT zu extrahieren. Dazu wird das Tomogramm zunächst durch ein “region growing”-Verfahren in ein Körper- und ein nicht-Körperteil segmentiert. Danach kann die Extraktion der zum Körper gehörend markierten Oberflächenvoxeln erfolgen.

Der erste genannte Teilschritt wurde dabei durch das am Institut entwickelte Programm “KaFus” [Büscher 98] umgesetzt. Auch die Extraktion der Oberfläche aus dem segmentierten Tomogramm ist in der institutsinternen C++-Template-Klasse “tomoCube” implementiert. Diese Klasse nimmt die Tomogrammdaten auf und stellt Funktion zu deren Manipulation zur Verfügung.

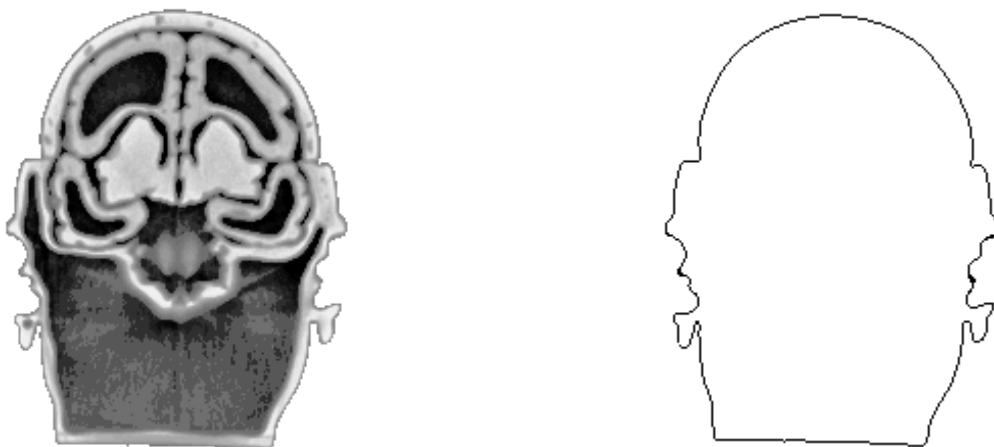


Abb. 5.3: Schicht aus dem 3D-CT-Datensatz (links) und die daraus extrahierte Kontur (rechts).

Die bestimmte Oberfläche wird als Objekt der Template-Klasse “tomoCube” abgespeichert.

5.3.2 Distanztomogramm

Das Distanztomogramm ist ebenfalls ein Objekt der Template-Klasse “tomoCube”. Statt den Grauwerten der Voxel enthält es aber deren Abstand zu einer, durch ein “Oberflächentomogramm” im Sinne des vorherigen Abschnittes, definierten Oberfläche. Die Distanzwerte erhält man aus einem in [Günzl 94] beschriebenen Algorithmus, der den kürzesten euklidischen Abstand eines Voxels zu der Oberfläche berechnet. Aus Speicherplatzgründen kann der berechnete Abstandswert aber nicht mit der vollen Genauigkeit (8 Byte) abgespeichert werden. Statt dessen stehen nur 2 Byte pro Voxel zur Verfügung die aber eine für die hiesige Anwendung absolut ausreichende Genauigkeit von $\frac{1}{100}$ mm zulassen.

Zur Visualisierung kann man das Abstandsspektrum des Distanztomogramms auf die Grauwerteskala projizieren und als normales Tomogramm darstellen. Dies ist in Abb. 5.4 zu sehen.

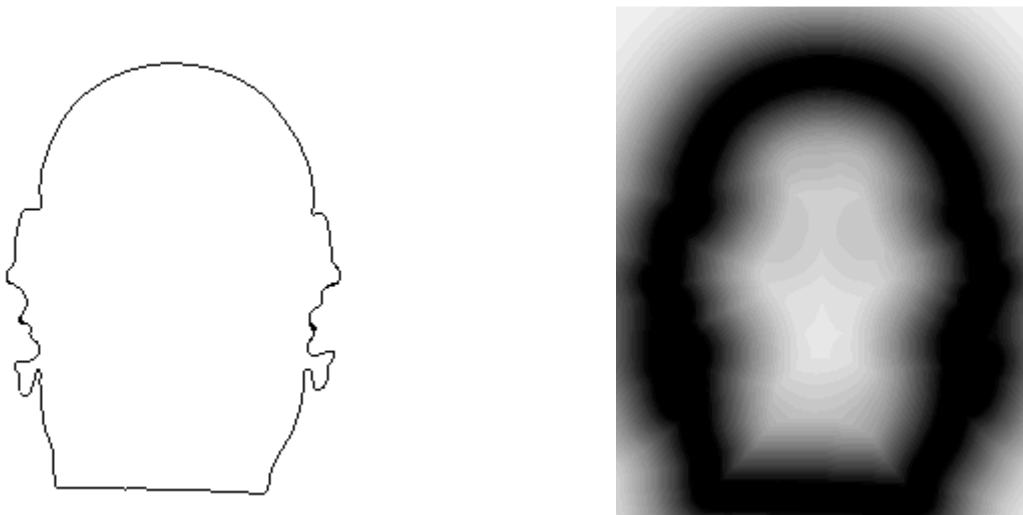


Abb. 5.4: Schicht aus dem 3D-Oberflächentomogramm (links) und dem Distanztomogramm in Grauwertdarstellung (rechts).

Man erkennt deutlich die Gebiete geringen Abstandes zur Oberfläche (dunkel) und die Regionen höheren Abstandes (hell).

Zu erwähnen bleibt noch, daß durch das jetzigen Vorgehen auch Oberflächen im Inneren des Kopfes, wie Nasen- und Rachenraumgrenzen zu der Oberfläche zählen. Da die Punktwolke aber

nach wie vor ein Teilmenge dieser Oberfläche ist, wird dies die Verwendbarkeit des Distanztomogramms im folgenden nicht einschränken.

Nun ist es ein Leichtes das Fehlermaß bei einer bestimmten Lage der Punktwolke numerisch zu berechnen. Man braucht nur noch das Distanztomogramm genau an den Stellen auszulesen an denen sich die Punkte der Punktwolke befinden, die Werte zu quadrieren und alle aufzusummieren. Da das Distanztomogramm aber nur Stützstellen des Abstandes im Raum zur Verfügung stellt, muß man entweder einen gewissen Fehler hinnehmen, oder man interpoliert zwischen den Abstandswerten der 8 Nachbarstützstellen. Da diese Berechnung zwar nur geringfügig aber doch mehr Zeit in Anspruch nimmt als das einfache Auslesen eines Wertes, wird abhängig vom Grad der Optimierung die Entscheidung über die Wahl des verwendeten Verfahrens getroffen. In Abschnitt 5.7 wird noch einmal konkreter Bezug auf diese Entscheidung genommen.

Jetzt wird auch klar warum man die Punktwolke auf das Volumenmodell matched und nicht umgekehrt. Würde man nämlich letzteres tun, müßte man bei jeder Berechnung, das ganze Distanztomogramm transformieren. Bei einem klassischen Tomogramm mit 256×256 Punkten pro Schicht und zum Beispiel 20 Schichten wären dazu bereits über 1,3 Millionen Einträge zu transformieren, statt den paar tausend Punkten der Punktwolke, was sich natürlich direkt auf die Rechenzeit auswirken würde.

Jetzt fehlt noch die Verbindung zwischen dem Voxelkoordinatensystem des Grauwerte- oder Distanztomogramms und dem Weltkoordinatensystem der Punktwolke. Diese Verbindung drückt die Normierungsmatrix aus.

5.4 Normierungsmatrix

Die Voxel eines Tomogramms werden einfach durch ganzzahlige Indizes adressiert. Diese Indizes der Voxel stellen die Koordinaten des Voxelkoordinatensystems dar. Um nun ein Tomogramm von einem Voxelkoordinatensystem in ein Weltkoordinatensystem zu überführen braucht man eine sogenannte Normierungsmatrix. Dies ist eine Transformationsmatrix (s. Abschn. 5.2) die in den meisten Fällen nur eine getrennte reellwertige Skalierung der einzelnen

Koordinaten vornimmt, also von der Form:

$$N = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{ist.}$$

In machen Fällen enthält die Normierungsmatrix noch den Gantrywinkel. Er berücksichtigt eine Verkippung der Aufnahmeeinheit des CTs um die y-Achse. Für den Fall einer Verkippung um den Winkel α gegenüber der Vertikalen erweitert sich die Normierungsmatrix zu:

$$N = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y \cdot \cos(\alpha) & 0 & 0 \\ 0 & s_y \cdot \sin(\alpha) & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Da auch das Distanztomogramm in Voxelkoordinaten vorliegt, wird es nötig sein die Punktwolke mit der inversen Normierungsmatrix in den Tomogrammraum hineinzutransformieren bevor man das Distanztomogramm ausliest. Auch muß die Punktwolke während der Optimierung immer in diesem elliptischen Tomogrammraum bewegt werden. Eine Rotation in diesem Raum läßt sich nun aber nicht mehr durch die in Abschnitt 5.2 besprochenen Rotationsmatrizen beschreiben. Mit Hilfe der Normierungsmatrix läßt sich aber eine Rotation um die x-Achse zum Beispiel durch folgende Form ausdrücken:

$$\begin{aligned} D_{x_{\alpha, \text{elliptisch}}} &= N \cdot D_{x_{\alpha}} \cdot N^{-1} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x^{-1} & 0 & 0 & 0 \\ 0 & s_y^{-1} & 0 & 0 \\ 0 & 0 & s_z^{-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -s_y \cdot \frac{\sin(\alpha)}{s_z} & 0 \\ 0 & s_z \cdot \frac{\sin(\alpha)}{s_y} & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Die beschriebene Transformation bildet die Punktwolke zuerst in das isometrische Weltkoordinatensystem ab, dreht sie dort um die x-Achse und transformiert sie wieder zurück.

Bleibt nur noch die Frage in Bezug zu welchem Koordinatensystem die Abstände in dem Distanztomogramm berechnet werden. Obwohl Distanztomogramm in Voxelkoordinaten vorliegt, beziehen sich die enthaltenen Abstandswerte auf die Oberfläche in Weltkoordinaten. Um also den Abstand eines Punktes zur Oberfläche zu erfahren, transformiert man ihn mit der Normierungsmatrix in das Voxelkoordinatensystem des Distanztomogramms und liest dort seinen Abstandswert zu der Oberfläche in Weltkoordinaten aus.

5.5 Initiale Transformation

Wie in Abschn. 5.1 bereits angedeutet wurde, wird durch eine initiale Transformation die Punktwolke in eine gute Ausgangslage für die folgende Optimierung gebracht. Zur Bestimmung der initialen Transformation gibt es drei denkbare Vorgehensweisen:

- vollautomatisch:
durch die automatische Gewinnung von Merkmalen aus der Punktwolke, wie Breite, Höhe, Hauptachse oder Wölbung könnte man versuchen eine automatische Vorpositionierung vorzunehmen.
- halbautomatisch:
durch eine formale Beschreibung der Orientierung des Kopfes bei der Aufnahme, z.B. lateral oder sagittal könnte man eine grobe Vororientierung der Punktwolke vornehmen. Die Genauigkeit wäre hierbei natürlich stark von der zur Verfügung stehenden Begriffsmenge abhängig.
- interaktiv:
ein Benutzer nimmt die Vororientierung manuell vor.

In der vorliegenden Arbeit wurde eine interaktive Lösung implementiert, da sie die flexibelste Lösung für eine erste Anwendung darstellt.

Die Methodik besteht nun darin, daß ein Benutzer in einem der beiden Modelle (Punktwolke oder Volumenmodell) drei beliebige linear unabhängige Punkte markiert. Danach markiert er in dem anderen Modell in der **gleichen** Reihenfolge die **ungefähr** korrespondierenden Punkte. Zum Beispiel könnte der Benutzer in beiden Systemen die Nasenspitze, das linke und das rechte Auge markieren.

Da die beiden Punkttupel aber keine exakte Korrespondenz aufweisen, darf man die gesuchte Rotations- und Translationsmatrix nicht einfach durch Matrixinversion bestimmen, da die so berechnete Abbildung nicht isometrisch ist, wie in der Einleitung des Kapitel 2 gefordert.

$$\left\{ \begin{array}{l} R = (\vec{v}_1 \quad \vec{v}_2 \quad \vec{v}_3) \cdot (\vec{v}_1' \quad \vec{v}_2' \quad \vec{v}_3')^{-1} \\ T = \vec{v}_1' - \vec{v}_1 \end{array} \right. \quad \text{aber } R \text{ nicht isometrisch}$$

Anschaulich würde eine solche Transformation die Punktwolke in bestimmten Richtungen stauchen oder dehnen, wodurch sie ihre Form verlieren würde. Dies würde ein sinnvolles Matching unmöglich machen.

Eine isometrische Transformation erfüllt die Bedingung, die Norm eines Vektor durch die Abbildung nicht zu ändern, also:

$$\|\varphi(\vec{x})\| = \|\vec{x}\|$$

Die Lösung des Problems erhält man dadurch, daß man *mit Hilfe* der ermittelten Punkte eine orthogonale orthonormale Basis in beiden Systemen aufspannt. Anhand der Basisvektoren kann man dann wie oben beschrieben die Abbildung bestimmen. Die so berechnete Abbildung ist isometrisch, da alle Basisvektoren den gleichen Normwert haben.

Die Methodik zur Erstellung einer solchen Basis auf drei beliebig linear unabhängigen Punkten funktioniert wie folgt:

Zwischen den Punkten p_1 und p_2 sowie den Punkten p_1 und p_3 baut man je einen Vektor \vec{v}_1 und \vec{v}_2 auf. Durch Kreuzprodukt zwischen dem Vektor \vec{v}_1 und \vec{v}_2 berechnet sich ein dritter Vektor \vec{v}_3 der senkrecht auf der durch die Ausgangsvektoren aufgespannten Ebene steht. Aus dem neu berechneten Vektor \vec{v}_3 und Vektor \vec{v}_1 ermittelt man nun durch ein weiteres Kreuzprodukt den letzten Basisvektor \vec{v}_4 (s.a. Abb. 5.5).

In einem letzten Schritt werden die Basisvektoren noch normiert, um der orthogonale Basis noch die Eigenschaft der Orthonormalität zu verleihen.

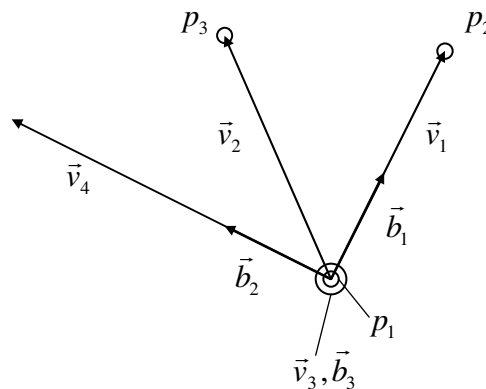


Abb. 5.5: Erstellen einer orthogonalen orthonormalen Basis $(\vec{b}_1 \ \vec{b}_2 \ \vec{b}_3)$ aus den Punkten

p_1, p_2, p_3

Formal lässt sich diese Vorgehensweise wie folgt beschreiben:

$$1. \text{ Schritt: } \vec{v}_1 = \begin{bmatrix} p_{2x} - p_{1x} \\ p_{2y} - p_{1y} \\ p_{2z} - p_{1z} \end{bmatrix} \quad \vec{v}_2 = \begin{bmatrix} p_{3x} - p_{1x} \\ p_{3y} - p_{1y} \\ p_{3z} - p_{1z} \end{bmatrix}$$

$$2. \text{ Schritt: } \vec{v}_3 = \vec{v}_1 \times \vec{v}_2 \quad \vec{v}_4 = \vec{v}_3 \times \vec{v}_1$$

$$3. \text{ Schritt: } \vec{b}_1 = \frac{\vec{v}_1}{\|\vec{v}_1\|} \quad \vec{b}_2 = \frac{\vec{v}_4}{\|\vec{v}_4\|} \quad \vec{b}_3 = \frac{\vec{v}_3}{\|\vec{v}_3\|}$$

Zur Unterstützung des Anwenders bei der Bestimmung der 2 Punktripel wurde das Programm “Geomview” eingesetzt. Es erlaubt in erster Hinsicht die Visualisierung der Punktwolke und des Volumenmodells. Durch die Anwendung des jetzt implementierten Moduls “3 Punkte markieren und abspeichern”, kann die Funktion aber darauf erweitert werden jeweils drei Punkte zu markieren. Die Weltkoordinaten der Punkte werden dann automatisch in der markierten Reihenfolge in eine Datei geschrieben.

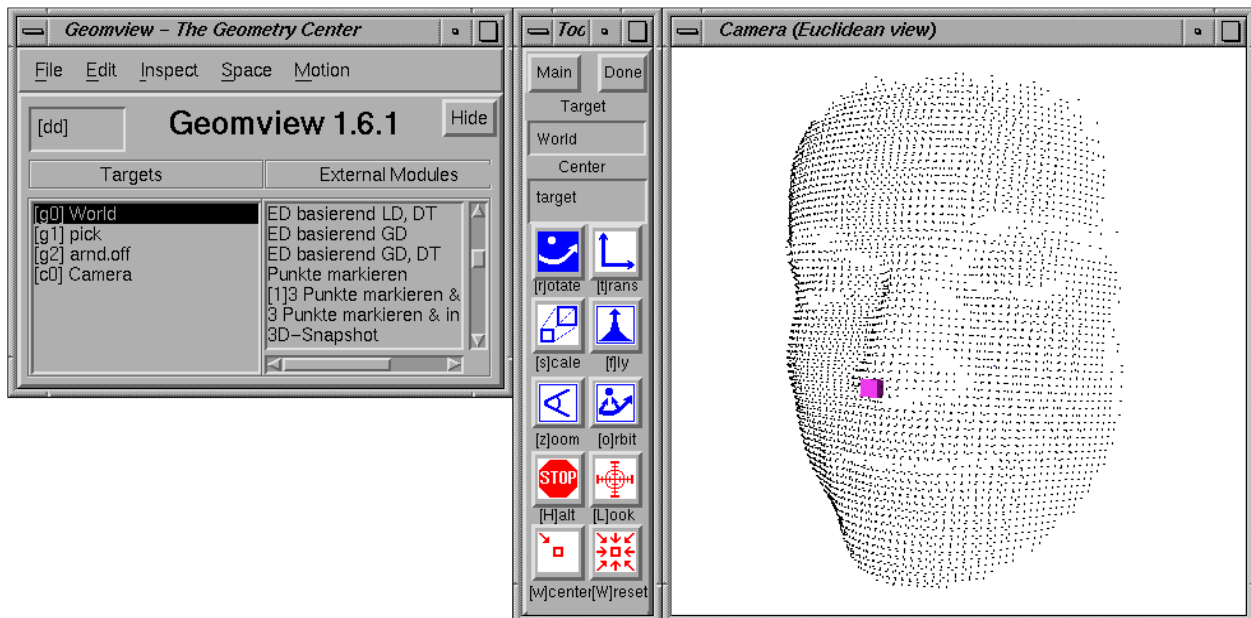


Abb. 5.6: Benutzeroberfläche von Geomview mit dem aktivierten Modul “3 Punkte markieren und abspeichern” und markierter Nase

5.6 Optimierung

Ziel der Optimierung ist es ausgehend von Lage der Punktwolke nach der Initialtransformation die Fehlerfunktion zu minimieren und somit die Punktwolke in die gewünschte optimale Lage zu bringen.

Als Optimierungsverfahren kommen nur Verfahren ohne Gradientenvektor in Frage, da die Fehlerfunktion nicht analytisch vorliegt. Es stellt sich aber noch die Frage, ob man von der Fehlerfunktion ähnlich der Distanztransformierten vor der Optimierung Stützstellen berechnen sollte um die eigentliche Optimierung zu beschleunigen. Würde man den Suchraum um die Initialposition auf 10 mm und 5 Grad pro Achse beschränken und in diesem Suchraum alle 0.5 mm und 0.25 Grad einen Stützpunkt berechnen, so wären dies trotzdem immer noch 64 Millionen Stützstellen. Für die Berechnung des Fehlers an jeder Stützstelle müßten die Abstände mehrerer tausend Punkte ausgelesen, quadriert und summiert werden. Dies macht jeden Zeitvorteil bei der Optimierung um Tage zunichte. Außerdem könnten die Stützstellen nicht Off-Line berechnet werden, da die Punktwolke ja erst bei der Operation selber aufgenommen werden kann.

Als mehrdimensionale sequentielle Optimierungsverfahren kommen Verfahren wie das Gauß-Seidel, Rosenbrock, Hooke-Jeeves mit dem aufbauenden Simplex-Algorithmus oder das Powell-Verfahren in Frage [Stehle 97]. Daneben sind auch statistische Suchverfahren und genetische Algorithmen nicht aus den Augen zu verlieren.

Ein sich am Gradientenabstiegsverfahren orientierendes Verfahren ist das genannte *Mustersuchverfahren nach Hooke-Jeeves*.

Das Verfahren ermittelt dabei zuerst, durch sogenannte Erkundungsschritte (exploratory moves), längs der n Koordinatenrichtungen der Fehlerfunktion, eine günstige Suchrichtung für den folgenden Musterschritt (pattern move), bei dem mit einer bestimmten Schrittweite längs dieser Suchrichtung fortgeschritten wird.

Die Wahl für dieses Verfahren hat praktische Gründe. Bei der ersten Implementierung des Matching hat die Einfachheit des Verfahren, in dem ohnehin schwer zu überschauenden 6 dimensional Raum, absolute Priorität.

Es sei bereits hier erwähnt, daß es ohne Zweifel schnellere und geschicktere Verfahren gibt, was aber aus dem genannten Grund nur zweitrangig ist.

Ein denkbare Szenario der Optimierung nach dem gewählten Verfahren für 2 Parameter ist in Abb. 5.7 dargestellt. Ausgehend von der Startposition oben links wird ein Erkundungsschritt mit

der Schrittweite Δd längs der positiven Parameterrichtung x_1 gemacht (1). Da der Schritt erfolgreich war also das Fehlermaß reduziert wurde, wird an der Stelle im Suchraum verharret und zum nächsten Parameter x_2 übergegangen. Der Erkundungsschritt in positive Parameterrichtung (2) war nicht erfolgreich, deshalb wird es mit der gleichen Schrittweite in negativer Richtung nochmals probiert (3). Nachdem alle Parameter einmal erkundet wurden erfolgt der Musterschritt (4). Er ergibt sich aus der Differenz der aktuell besten Position zu der letzten bekannten besten Position vor der Erkundung. Für die Schritte (6),(7) und (8) wiederholt sich das eben geschilderte Szenario. Man beachte allerdings, daß der Musterschritt diesmal größer ist als der Letzte. Dies kommt dadurch, daß die letzten bekannte beste Position vor der Erkundung, der Punkt nach Schritt (3) ist und nicht wie im Zyklus davor der Punkt vor Schritt (5).

Sind nun alle Erkundungsschritte erfolglos (9, 10, 11, 12), so geht man zurück zur letzten bekannten besten Position und beginnt das Spiel von Neuem. Erst wenn auch die Erkundung hier fehlgeschlagen hat (13, 14, 15) reduziert man die Schrittweite um die Hälfte und verfährt wie gehabt.

Das Verfahren bricht ab, wenn die Schrittlänge Δd kleiner als eine vorgegebene Schranke ϵ wird.

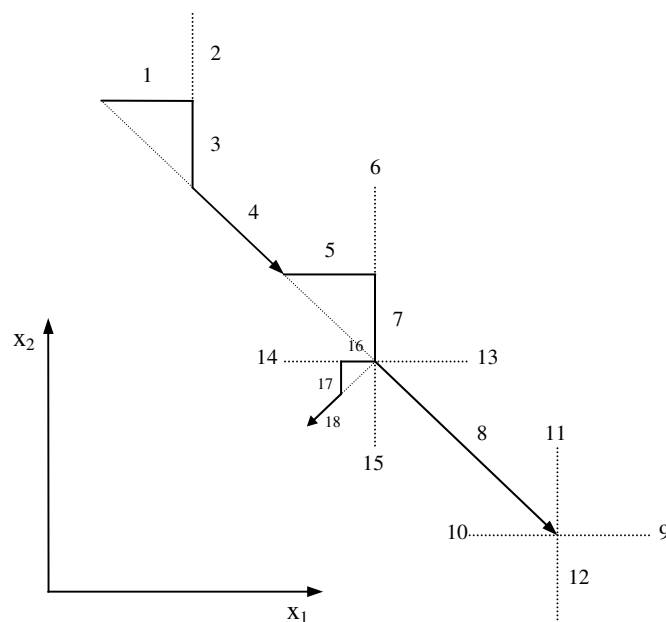


Abb. 5.7: Prinzip des Mustersuchverfahrens

5.7 Implementierung

Das eben beschriebene Optimierungsverfahren wurde in dieser Arbeit als Algorithmus in C++ implementiert und wird durch das Flußdiagramm in Abbildung 5.8 beschrieben.

Alle nicht bei der Initialisierung aufgeführten Variablen sind mit dem Wert 0 initialisiert worden, außer die Schrittweite “Step” sie wird natürlich mit einer maximalen Schrittweite gestartet.

Die Lage der Punktwolke ist durch 3 Translationen und 3 Rotationen um die Koordinatenachsen vollständig bestimmt. Die Optimierung erfolgt deshalb in einem 6 dimensional Parameterraum, wobei “ActualPositon” die aktuelle Position in diesem Raum wiedergibt. Die Einheiten der Translationen und Rotationen sind Millimeter, resp. Grad.

Als Erweiterung zu dem eigentlichen Vorgehen, wurde eine für jede Parameterrichtung unterschiedlich wählbare Schrittweite implementiert. Dies wurde erreicht indem für jeden Parameter feste Gewichtungsfaktoren angegeben werden können mit denen die Schrittweite bei Änderung der Parameter multipliziert wird.

Abhängig von der Schrittweite wird auch das Verfahren der Fehlerfunktionsberechnung geändert. Es wurde bereits bemerkt, daß das Distanztomogramm nur Stützstellen des Abstandes im Raum zur Verfügung stellt, man also entweder einen gewissen Fehler hinnehmen muß oder man eine Interpolation zwischen den 8 Nachbarstützstellen vornimmt. Um die Rechenzeit zu optimieren wurde bei der Implementierung des Algorithmus im Falle großer Schrittweiten, also noch schlechter Übereinstimmung der Modelle, einfach nur das Distanztomogramm an den entsprechenden Stelle ausgelesen, bei kleineren Schrittweiten wurden die Abstandswerte aus der Interpolation berechnet.

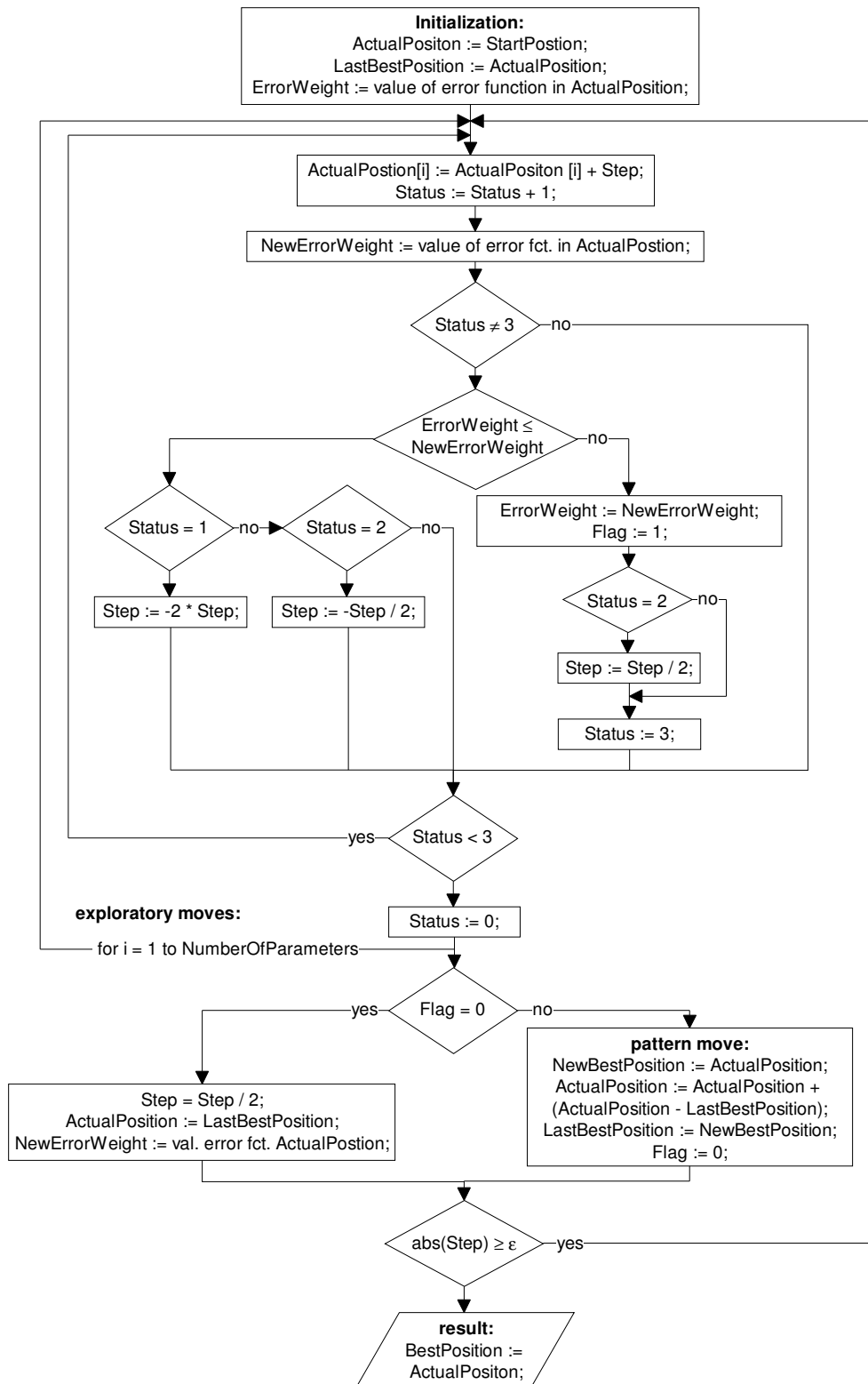


Abb. 5.8: Mustersuchverfahren nach Hooke-Jeeves

Dem Konzept der C++-Programmiersprache folgend [Schumann 91][Eggink 95], bietet es sich förmlich an für die Punktwolke eine eigene Klasse zu definieren die die Meßdaten aufnimmt und Funktionen zu ihrer Manipulation zur Verfügung stellt. Dabei wird in jedem Objekt der Klasse eine linke und eine rechte Teilpunktwolke unterschieden, wie sie auch vom Oberflächenscanner durch die rechte und die linke Kamera geliefert wird. Im folgenden werden deshalb die öffentlichen Funktionen der Klasse “pointCloud” kurz beschrieben:

```
vec3 GetPoint (long Position);
```

Gibt die Koordinaten des Punktes Nummer `Position` als Objekt der Klasse `vec3` aus.

```
void SetPoint (float xCoord,float yCoord,float zCoord);
```

Fügt der linken Punktwolke einen Punkt mit den Koordinaten `xCoord`, `yCoord` und `zCoord` zu.

```
void LoadFileToArray (char *Filename);
```

Lädt eine ASCII-Datei des Scanners mit dem Dateinamen `Filename` in den Speicher.

```
void SaveAsGeomviewFile (char *OriginalDigifaceFilename,
char *OffFilename, int DumpFlag, int CoordinateFlag);
```

Speichert die Punktwolke zur Visualisierung im Geomview-Format unter dem Filenamen `OffFilename` ab, wobei Punkte der Koordinaten (0,0,0) nicht berücksichtigt werden. Zur Dokumentation des Namens der Ursprungsdatei im Header des Geomviewfiles, wird dieser in dem String `OriginalDigifaceFilename` übermittelt. Es kann durch `CoordinateFlag=1` dem Geomviewdatensatz ein graphisches Koordinatensystem zugefügt werden. `DumpFlag` steuert die Ausgabe auf den Bildschirm.

`DumpFlag=0`: nur die wichtigsten Informationen werden auf dem Bildschirm ausgegeben.

`DumpFlag=-1`: keine Informationen werden auf dem Bildschirm ausgegeben.

Sonst: Alle Informationen werden auch auf dem Bildschirm ausgegeben.

```
void TransfPointCloud (mat4 Transf);
```

Transformierte die ganze Punktwolke mit der 4×4-Matrix `Transf` der Klasse `mat4`.

```
double calcErrorSquareSum (tomoCube<unsigned short>
    &distanceCube, int xMax, int yMax, int zMax, double
    StepSize, int DumpFlag);
```

Berechnet in Abhängigkeit der Schrittweite StepSize das Summenabstandsquadrat der Punktwolke zu einem Tomogramm, mit Hilfe des Distanztomogramms distanceCube der Größe xMax, yMax, zMax. (DumpFlag siehe oben)

```
void GetNumberOfAllPoints (long &AmountAllPointsRight,
    long &AmountAllPointsLeft);
```

Gibt die Anzahl aller Punkte der Punktwolke, getrennt nach rechter und linker Teilpunktwolke, in den Variablen AmountAllPointsRight und AmountAllPointsLeft zurück.

```
void GetNumberOfTruePoints (long &AmountTruePointsRight,
    long &AmountTruePointsLeft);
```

Gibt die Anzahl aller Punkte der Punktwolke die nicht dem Vektor $\begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$ entsprechen, getrennt nach rechter und linker Teilpunktwolke, in den Variablen AmountTruePointsRight und AmountTruePointsLeft zurück.

```
void SaveAsDigifaceFile (char *DigifaceFilename, int DumpFlag);
```

Speichert die Punktwolke wieder im ASCII-Format des Oberflächenscanners, unter dem Filenamen DigifaceFilename, ab. (DumpFlag siehe oben)

```
void MergeWithTomoCube (tomoCube<unsigned char> &TomoCube,
    unsigned char GrayValue);
```

Visualisiert die Punktwolke in einem Tomogramm indem an den Stellen der Punkte Voxel mit dem Grauwert GrayValue gesetzt werden.

```
int GetReductionFactor (void);
```

Fordert zur Eingabe einer Zahl, zur Reduzierung der Anzahl der Punkte, auf und übergibt die Zahl als Rückgabewert der Funktion.

```
void DetermineTruePoints (int DumpFlag);
```

Bestimmt die Anzahl der Punkte der Punktwolke die nicht dem Vektor $\begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$ entsprechen. Dies erfolgt getrennt für die rechte und linke Teilpunktwolke. Die Werte werden in klasseninternen Variablen gespeichert. (DumpFlag siehe oben)

Weiterhin enthält das Include-File `pointCloud.hh` noch Funktionen die nicht klassenspezifisch sind.

```
void CreateFilename (char *InputFilename, char *NewExtension,  
                    char *OutputFilename);
```

Schneidet von dem String `InputFilename` die letzten 4 Buchstaben ab und fügt den String `NewExtension` wieder an. Das Ergebnis wird in `OutputFilename` zurückgegeben.

```
void LoadMat4 (mat4 &Norm, char *Filename);
```

Lädt ein Objekt `Norm` der Klasse `mat4` aus einer Datei mit dem Namen `Filename`.

```
void SaveMat4 (mat4 Matrix, char *Filename);
```

Speichert ein Objekt `Matrix` der Klasse `mat4` in eine Datei des Namens `Filename`.

```
void CalcTransfMat (mat4 &TransfMat, double xRot, double yRot,  
                   double zRot, double xTrans, double yTrans, double zTrans);
```

Berechnet eine Transformationsmatrix die nacheinander die Rotationen `xRot` um die x-Achse, `yRot` um die y-Achse, `zRot` um die z-Achse und die Translationen `xTrans` in x-Richtung, `yTrans` in y-Richtung und `zTrans` in z-Richtung, an einem Vektor der Klasse `vec4` vornimmt, und gibt diese Matrix in dem Objekt `TransfMat` der Klasse `mat4` zurück.

```
void erstelleBasis (mat3 &result, vec3 p1, vec3 p2, vec3 p3);
```

Erstellt eine orthogonale orthonormale Basis mit Hilfe von den Vektoren `p1`, `p2` und `p3` der Klasse `vec3`. Die Basisvektoren werden als Spaltenvektoren der 3×3-Matrix der Klasse `mat3` zurückgegeben.

```
void CalcPreOrientTransfMat(mat4 &result, char* Filename1,  
                             char* Filename2);
```

Berechnet eine isometrische Transformation zur Vorpositionierung der Punktwolke, aus den 2 * 3 Punkten in den ASCII-Dateien `Filename1` und `Filename2`. Die Transformation wird in der 4×4-Matrix `result` der Klasse `mat4` zurückgegeben.

Kapitel 6

Ergebnisse & Ausblick

6. Ergebnisse & Ausblick

6.1 Ergebnisse & Diskussion

Anhand eines Dummys fand eine Überprüfung des implementierten Matchingverfahrens statt. Bei dem Dummy handelt es sich um ein anatomisches Modell des menschlichen Kopfes. Das Volumenmodell-Punktwolke-Paar war bereits in den Abbildungen 1.1 und 1.2 der Einleitung zu sehen. Ebenso ist die Ausgangslage der Punktwolke relativ zum Volumenmodell in Abb. 5.1 visualisiert.

Der erste Schritt des eigentlichen Matchings war die Bestimmung einer initialen Transformation. Dazu wurden in beiden Systemen die Nase, das rechte und das linke Auge als korrespondierende Punkte herangezogen. In Abbildung 6.1 ist das *Resultat der Vororientierung* zu erkennen. Die Punkte der Punktwolke sind stark vergrößert dargestellt.

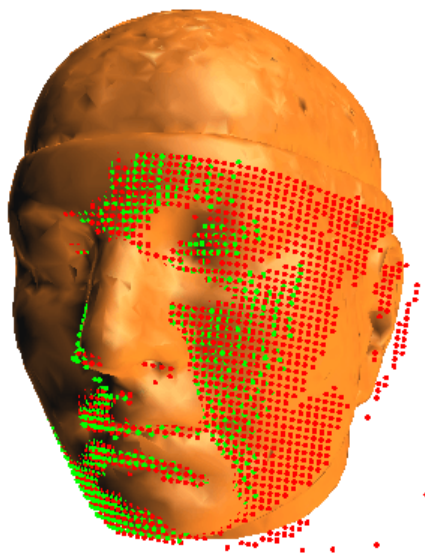


Abb. 6.1: Lage der Punktwolke relativ zu dem Volumenmodell nach der initialen Transformation

Man erkennt, daß die Punktwolke auf der linken Seite hinter der Oberfläche verschwunden ist, dafür steht sie auf der rechten Seite weit von Oberfläche ab. Trotzdem ist dies eine wesentlich günstigere Ausgangslage für die Optimierung. Das Fehlermaß verrät, daß der mittlere Abstand der Punkte zur Oberfläche immerhin noch 3.6 Millimeter beträgt. Es kann also noch keineswegs

von einer optimalen Lage gesprochen werden. Auch bei allen weiteren hier nicht dokumentierten Versuchen konnte die Vororientierung einwandfrei durchgeführt werden, unter der Voraussetzung natürlich, daß man in beiden Systemen die Punkte auch wirklich in der gleichen Reihenfolge markiert hat.

Kommen wir nun zur Optimierung: Sie wurde mit einer Schrittweite von 1 gestartet und nach 0.25 unterbrochen. Die Gewichtungsfaktoren betrugen für die Translationen gerade 1, für die Rotationen 0.5, d.h. die Punktwolke wurde im Falle der feinsten Schritte, jeweils um 0.25 mm verschoben und um 0.125 Grad gedreht. Die reine Optimierungszeit bis zum Abbruch betrug 45 Sekunden, dabei durchlief der Algorithmus 21 Optimierungszyklen mit jeweils 6 Parameteroptimierungen.

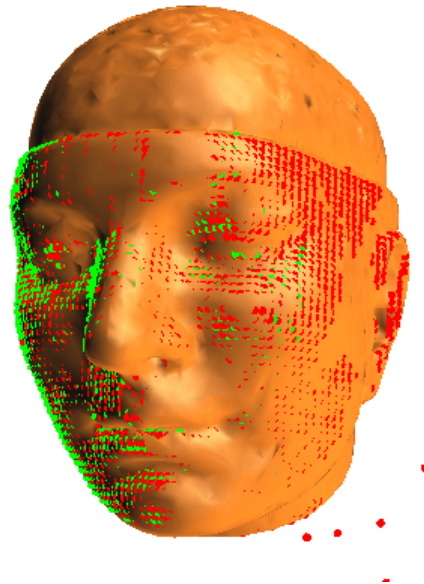


Abb. 6.2: Lage der Punktwolke relativ zu dem Volumenmodell nach der Optimierung

In Abbildung 6.2 ist nun die *Punktwolke nach der Optimierung* dargestellt. Man erkennt deutlich die erfolgreiche Optimierung. Die Punktwolke hat sich sehr exakt an die Kontur des Volumenmodells angeschmiegt. Die Punkte liegen sehr gemischt mal hinter der Oberfläche mal vor der Oberfläche. Auch an der sauber mit dem Absatz des Volumenmodells abschließenden Oberkante der Punktwolke erkennt man die Qualität der Registrierung. Eine detaillierte optische Untersuchung hat auch keine Mängel erkennen lassen.

Die Fehlerfunktion bestätigt den optischen Eindruck, der mittlere Abstand der Punkte zur Oberfläche beträgt jetzt nur noch 0.6 Millimeter. Bedenkt man, daß die dem Distanztomogramm zugrundeliegende Oberfläche durch das Tomogramm mit einer Genauigkeit von ± 0.4 mm, neben den Fehlern des Scanners und der Approximation der Abstände, bestimmt wird, so ist dieses Resultat sehr zufriedenstellend.

Die Ergebnisse des Matchings in einem früheren Stadium der Arbeit mahnen aber zur Vorsicht:

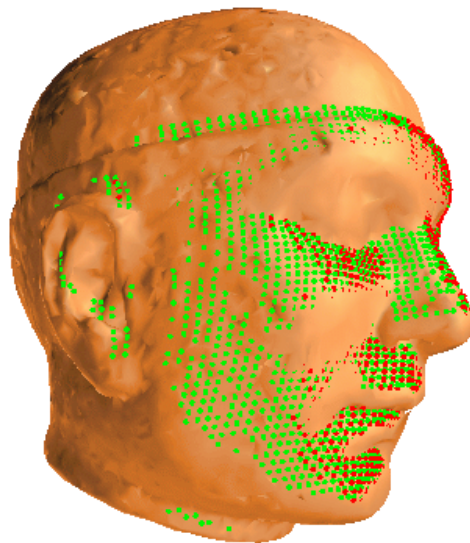


Abb. 6.3: Lage der Punktwolke nach Optimierung bei spiegelverkehrtem Modell.

In der Abbildung 6.3 ist das Resultat eines offensichtlich schlechteren Matchings zu sehen. Das Bild wurde retuschiert um die Problemstellen hervorzuheben. Auch bei diesem Matching betrug der mittlere Abstand der Punkte zur Oberfläche nur noch 1.0 Millimeter. Aus Sicht der Optimierung ist das Resultat also durchweg als zufriedenstellend anzusehen. Trotzdem erkennt man an dem Ohr und an der Absatzkante, daß die Punktwolke nicht paßgenau auf dem Modell liegt. Effektiv wäre aus einer genaueren Betrachtung ersichtlich, daß die Punktwolke mehrere Grad um die Nasenspitze nach rechts verdreht auf dem Gesicht sitzt, das Matching also nicht der erwarteten Deckung entspricht.

Welche Gründe kommen dafür in Frage?

1. das Tetraedernetz als Visualisierung des Volumenmodells ist falsch berechnet, d.h. das Matching wäre korrekt, nur die Visualisierung würde etwas anderes vortäuschen.
2. die Punktwolke oder das Volumenmodell ist spiegelverkehrt.
3. die Startposition für die Optimierung ist ungünstig gewählt, daher
4. bleibt die Optimierung in einem lokalem Minimum der Fehlerfunktion hängen.
5. Streupunkte mit sehr großen Abstand gewinnen ein zu großes Gewicht im Fehlermaß.
6. die Wahl der Fehlerfunktion oder des Abstandsmaßes sind ungeeignet
7. die Aufnahme der Punktwolke ist schlecht gewählt.

Es soll gleich vorweg genommen werden, daß sich herausgestellt, daß das Volumenmodell spiegelverkehrt war. Da aber bereits bei der Implementierung des Matchings auf diese Problematik geachtet wurde, ist die Ursache der Spiegelung nicht dort zu suchen. Es hat sich herausgestellt, daß *bei der Erstellung der Normierungsmatrix ein Fehler* gemacht wurde. Die Normierungsmatrix muß zur Zeit nämlich noch für jeden 3D-CT-Datensatz manuell erstellt werden, da der verwendete Computertomograph diese Funktion nicht zur Verfügung stellt.

Bei der Erstellung der Matrix ist die Vorschubrichtung der Aufnahmen aufeinanderfolgender Schichten von grundlegender Bedeutung. Sie entscheidet darüber, ob bei der späteren Modellerstellung aus dem Tomogramm, die aufeinanderfolgenden Schichten in Richtung größer werdender z oder kleiner werdender z hintereinander gelegt werden müssen. Geschieht dies genau verkehrt herum, ist das erstellte Modell rechts/links-gespiegelt. Das rechte Ohr zum Beispiel befindet sich danach links und umgekehrt.

Die Suche nach der Ursache der geringeren Qualität des eben besprochenen Matchings, hat weitere interessante Fragen in Bezug auf das eigentlich Matchingverfahren aufgeworfen, die auch nach der jetzigen Behebung des Problems von Interesse sind. Im Ausblick (s. Abschn. 6.2) soll darauf eingegangen werden.

Wenden wir uns aber wieder der Untersuchung des Matchingverfahrens zu. Es soll ein Matching unter *verschärfteren Randbedingungen* durchgeführt werden. Erstens wird durch eine extrem ungünstige Wahl der korrespondierenden Punktpaare eine ungünstige Vorpositionierung der Punktwolke für das Matching provoziert und gleichzeitig eine nicht zusammenpassendes Punktwolke-Volumenmodell-Paar verwendet. In Abbildung 6.4 a) ist das Paar in der Initialposition zu sehen.

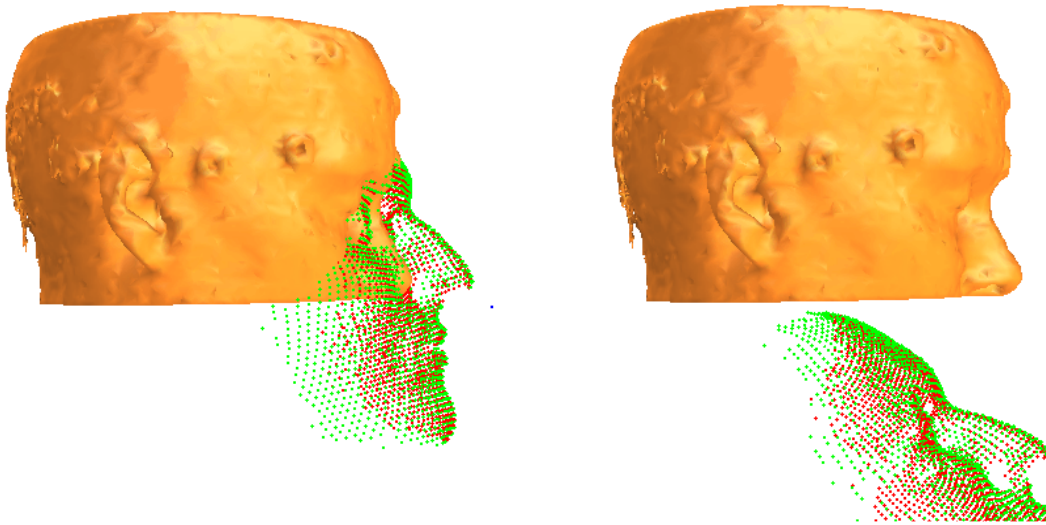


Abb. 6.4: Matching bei schlechter Initialposition: a) vorher b) nachher.

Wie zu sehen ist schlug das Matching fehl. Die Punktwolke hat sich komplett aus dem Volumenmodell herausbewegt. Trotzdem zeigt die Fehlerfunktion ein perfektes Matching an, das Fehlermaß beträgt nämlich gerade Null. Die Fehlerfunktion hat also offensichtlich ein globales Minimum in der vorliegenden Position. Im folgenden wird gezeigt warum dieser Fall eingetreten ist und daß dies kein Fehler, sondern eine unumgängliche Notwendigkeit ist.

Wie zu sehen ist besteht das Volumenmodell nur aus einem Teilausschnitt des Gesichtes. Dieser Fall hat eine sehr hohe Praxisrelevanz, da wegen der Belastung durch die zur Bildgebung verwendeten Röntgenstrahlung immer nur so viele Schichten wie unbedingt nötig aufgenommen werden. Da aber auch das Distanztomogramm nicht größer als das ursprüngliche Grauwerttomogramm ist, modelliert die Punktwolke im vorliegenden Fall ein größeren Bereich des Gesichtes als das Distanztomogramm dafür Abstände zur Verfügung stellen kann. Nun stellt sich die Frage welchen Abstandswert man den Punkten der Punktwolke zuweist die im Moment nicht im Tomogramm liegen. Weist man diesen Punkten große Abstandswerte zu so wird der Optimierungsalgorithmus abhängig von der Größe des gewählten Abstandswertes tendenziell versuchen ein Maximum an Punkten in das Tomogramm hineinzulegen, egal ob diese Punkte auf der Oberfläche liegen oder nicht. Es kommt also notgedrungenerweise zu einem völlig unbrauchbaren Matching. Versuche haben diese Aussage bestätigt.

Geht man hingegen hin und weist den Punkten, wie im vorliegenden Fall getan, den Abstand Null zu, so ist es für den Optimierungsalgorithmus von Vorteil, möglichst wenig Punkte in das Tomogramm hineinzulegen. Mit jedem Punkt der außerhalb des Tomogramms zum Liegen kommt reduziert sich nämlich das Fehlermaß. Dies heißt aber im Gegensatz zum vorherigen Vorgehen nicht, daß es zwangsweise zu einem erfolglosen Matching kommen muß. Vielmehr ist der Erfolg von der Initialposition abhängig. Abbildung 6.5 zeigt das Resultat des Matchings nach einer geschickten Vorpositionierung, die durch eine normale Wahl der Punktpaare zustande kam.

Die Frage ist nun, ob es einen anderen oder variablen Wert gibt den man den Punkten zuweisen kann um die genannten Problem zu vermeiden. Dies kann verneint werden. Jede Wahl eines Wertes oder einer Funktion wäre äquivalent zu einer künstlichen Erweiterung der Oberfläche des Tomogramms. Wie soll aber eine solche Erweiterung genauer der wahren also der Aufnahme zugrundeliegenden Oberfläche entsprechen als es die Punktwolke bereits tut. Eine auch noch so geschickte Wahl dieser Erweiterung würde zwar bewirken könne daß die Punktwolke durch die Optimierung nicht mehr aus dem Tomogramm herausbewegt wird, der Grad der Übereinstimmung der Oberflächen und damit die Qualität der gelieferten Transformationsmatrix wäre aber immer schlechter als das Resultat eines Matchings mit der Wertzuweisung Null.

Letzteres Vorgehen entspricht anschaulich einem Abschneiden der Punktwolke über und unterhalb des Tomogramms, da diese Punkte durch ihren Abstandswert Null nicht mehr zu der Fehlerfunktion beitragen und damit die weitere Optimierung nicht mehr beeinflussen können.

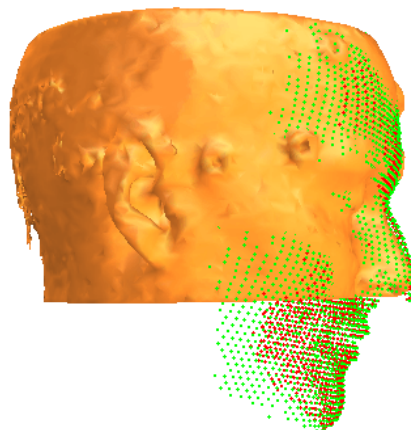


Abb. 6.5: Matching bei nicht korrespondierendem Punktwolke-Volumenmodell-Paar und guter Initialposition.

Damit wäre auch der zweite wichtige Grund für eine Initialtransformation gezeigt, denn nur eine gute Initialtransformation kann sicherstellen, daß es zu einem erfolgreichen Matching kommt.

Dennoch soll noch einmal erwähnt werden, daß wir gerade ein nicht korrespondierendes Punktwolke-Volumenmodell-Paar gematcht haben. Trotz dieser Randbedingung hat ein erfolgreiches Matching stattgefunden. Daraus läßt sich vermuten, daß die Fehlerfunktion wenig Nebenminima hat. Wäre dies nicht der Fall hätte die Optimierung bei diesem Punktwolke-Volumenmodell-Paar mit Sicherheit in einem lokalen Minimum geendet.

Lenken wir nun unser Augenmerk auf den Verlauf des Fehlermaß bei einer solchen Optimierung. In Abbildung 6.6 ist ein solcher Verlauf für das am Anfang des Abschnitts besprochene Matching zu sehen. Dokumentiert ist die Auswirkung jeder Parameteränderung, also auch die Erkundungsschritte des Optimierungsalgorithmus. Die Anzahl der dargestellten Schritte entspricht also nicht der Anzahl der Optimierungsschritte, letztere beträgt nämlich wie oben angegeben 21. Der Ausschnitt zeigt die letzten 6 Parameteroptimierungen.

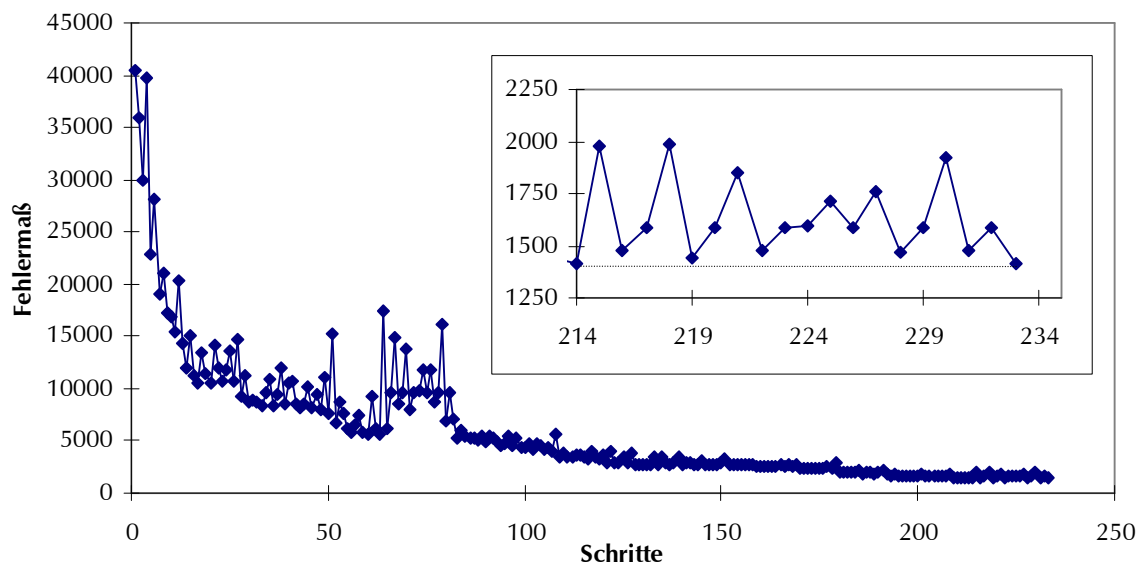


Abb. 6.6: Verlauf der Fehlerfunktion mit Ausschnitt aus den letzten 6 Parameteroptimierungen vor Abbruch.

Es ist deutlich die Verbesserung des Fehlermaßes zu erkennen. Die Peaks sind erfolglose Erkundungsschritte. Die nach Schritt 60 auftretende Unstetigkeit beruht auf dem in Abschnitt 5.7 beschriebenen Wechsel der Bestimmung der Abstände. Die genauere Bestimmung der Abstände

der einsetzenden Interpolation führt in manchen Fällen zu einer kurzfristigen Erhöhung des Fehlerverlaufs.

Im Ausschnitt der letzten 6 Parameteroptimierungen ist zu erkennen wie keiner der Schritte die Funktion mehr zu minimieren vermag und somit zur Position vor dem letzten Musterschritt zurückgekehrt wird bevor der Algorithmus abbricht.

Auch die Fehlerfunktion selber war Ziel einer Überprüfung. Es wurde untersucht, ob ein kubisches oder noch höheres Summenfehlermaß die Oberflächenregistrierung noch weiter verbessern kann. In Abbildung 6.7 ist der mittlere Abstand der Punkte nach der Optimierung für die Exponenten 2, 3, 4 und 5 der Fehlerfunktion aus Gleichung 5.1 skizziert.

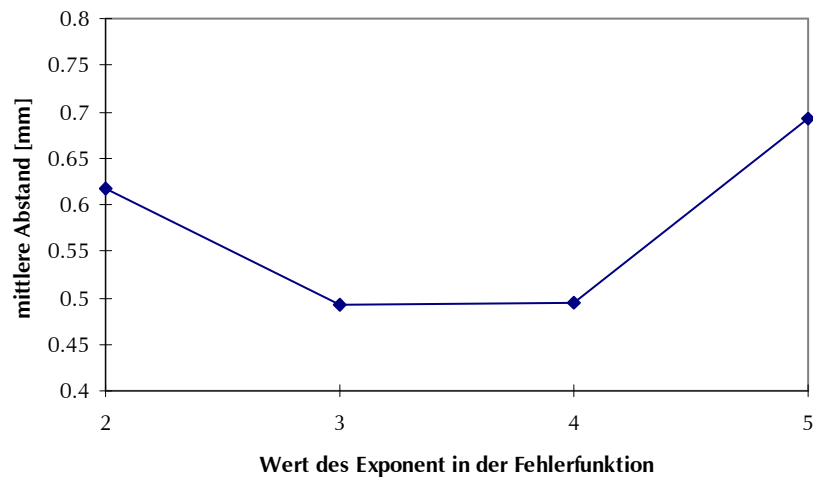


Abb. 6.7: Mittlerer Abstand der Punktwolke zum Volumenmodell in Abhängigkeit des verwendeten Fehlermaßes

Die beste Lage wurde für den kubischen Summenfehler erzielt mit einem mittlerer Abstand der Punktwolke zum Volumenmodell von 0.49 mm. Von diesem einen Experiment kann sicherlich nicht auf die Allgemeinheit geschlossen werden, es zeigt aber daß sich die Fehlerfunktion sehr stark auf die Qualität des Resultats auswirkt.

Damit wäre die Betrachtung der Resultate abgeschlossen. Der vielleicht wichtigste Teil einer Zusammenschrift über 6 Monate Arbeit ist aber die Rekapitulation und der Ausblick auf zukünftige Entwicklungen. Dem wird im nächsten Abschnitt Rechnung getragen.

6.2 Zusammenfassung & Ausblick

Diese Arbeit stellt eine allgemeine Betrachtung der Einsetzbarkeit eines 3D-Oberflächenscanners zur Lagebestimmung in der Medizin dar.

Dazu wurde zuerst eine genaue Untersuchung des Meßfehlers und der Störempfindlichkeit des Scanners durchgeführt. Der dabei ermittelte absolute Positionierungsfehler der gesamten Oberflächenpunkt看ke in z-Richtung beträgt weniger als 0.2 mm. Bei der Bestimmung des Fehlers in x- und y-Richtung mußte auf ein Gütekriterium ausgewichen werden, das die Formtreue des Scanners wiedergeben kann. Auch hier schneidet der Scanner sehr gut ab.

Wie bei einem optischen Verfahren nicht anders zu erwarten liefert das Gerät um so bessere Ergebnisse je weniger Umgebungslicht die Aufnahme stört und je heller die aufzunehmende Oberfläche ist. Trotzdem hat sich der Scanner im praktischen Einsatz bewährt und zeigt sich als durchaus robust gegenüber den genannten Störungen.

Die für Bewegungsartefakte kritische Aufnahmedauer beträgt beim Scannen 4 Sekunden. Danach folgt eine ungefähre Berechnungszeit von 20 Sekunden bis mehrere tausend auf der Oberfläche des Meßobjektes bestimmte Punkte zur Verfügung stehen.

In dem zweiten Teil der Arbeit wurde ein Matchingverfahren zur eigentlichen Lagebestimmung implementiert. Dazu wird die vom Oberflächenscanner gelieferte Punkt看ke mit einem Volumenmodell des gleichen Objektes gematched. Der erste Schritt dieser Oberflächenregistrierung besteht in einer initialen Transformation, die die Punkt看ke in eine gute Startposition für die folgende Optimierung bringen soll. Diese Initialisierung wird interaktiv ausgeführt durch das Markieren von drei korrespondierenden Punktpaaren zwischen der Punkt看ke und dem Volumenmodell.

Zur Optimierung der Lage der Punkt看ke kam das sequentielle Mustersuchverfahren nach Hooke-Jeeves zur Anwendung. Die eigentlich zu optimierende Fehlerfunktion, die den Grad der Übereinstimmung zwischen der Punkt看ke und dem Volumenmodell quantifiziert, wird durch das Summenabstandsquadrat der kürzesten euklidischen Abstände aller Punkte zur Oberfläche des Volumenmodells ausgedrückt. Zwecks einer schnellen numerischen Berechnung dieser Fehlerfunktion kommt ein Distanztomogramm zum Einsatz, das Stützstellen der Abstände zur Oberfläche des Volumenmodells zur Verfügung stellt.

Die Überprüfung des Matchingverfahrens erfolgte an einem Plastikkopf von dem sowohl eine CT-Aufnahme als auch ein Oberflächenscan erstellt werden konnte. Der Grad der erzielten Übereinstimmung beider Modelle durch das Matching ist mit einem mittleren Abstand der

Punktwolke zum Volumenmodell von rund 0.5 mm als sehr gut zu bezeichnen. Dazu mußte aber das Abstandsmaß von quadratisch auf kubisch erhöht werden.

Bei einer normal guten Vorpositionierung der Punktwolke durch die Initialtransformation, ist auf einer Workstation mit einer Registrierung innerhalb einer Minute zu rechnen.

Was sind nun die *Zukunftsansichten* der Lagebestimmung durch Oberflächenregistrierung und welches *Potential zur Verbesserung* halten die Teilschritte des Matchingverfahren noch vor?

Die vorliegende Arbeit hat gezeigt, daß eine Lagebestimmung mit Hilfe eines Oberflächenscanner praktisch umsetzbar ist. Das Verfahren bedingt allerdings ausreichend helle Oberflächen und geringe Umgebungshelligkeit (Problem: OP-Strahler). Obwohl die Oberflächenregistrierung auf der guten Übereinstimmung von Oberflächen basiert, hat sich aber gezeigt (s. Abb. 6.5), daß sich auch unterschiedliche Oberflächen noch sinnvoll matchen lassen. Eine genauere Untersuchungen der Qualität der daraus berechneten Transformationsmatrizen ist aber notwendig. Es wäre aber sicherlich interessant zu wissen wie sich eine Schwellung, eine Wunde oder ein Operationswerkzeug auf die Registrierung auswirken würde.

Ein Tracking, also eine permanente Verfolgung der Lage, ist aufgrund der Aufnahme- und Rechenzeit derzeit aber noch nicht möglich.

Zur weiteren Verbesserung des eigentlichen Matchingverfahren in Bezug auf Rechenzeit, Qualität der berechneten Transformationsmatrix und Benutzerfreundlichkeit enthalten die Komponenten noch folgendes Potential zur Optimierung.

Wie in dem Abschnitt 6.1 beschrieben, hat ein spiegelverkehrtes Volumenmodell zwei interessante Fragen aufgeworfen deren Untersuchung von Interesse sein könnte.

Erstens stellt sich nämlich die Frage wie man eine Aufnahme mit dem Scanner geschickt wählen kann um das folgende Matching zu unterstützen. Die frontale Aufnahme im behandelten Beispiel hat nämlich den Nachteil, daß sie der Punktwolke beim Matching noch zuviel Freiheit bei einer Rotation um die Nasenspitze/Hinterkopf-Achse läßt. Eine Erfassung des Bereichs unter dem Kinn könnte diese Freiheit bestimmt einschränken.

Zweitens ist das Distanztomogramm zu hinterfragen. Ist es nämlich wirklich sinnvoll beim Matching auch Punkte hinter der Oberfläche des Volumenmodells zuzulassen? Wäre dies nicht der

Fall könnte man dies wirksam durch einen sprunghaften Anstieg der Abstandswerte ins Inneren des Kopfes hinein unterbinden.

Das nächste Glied in der Kette ist die Optimierung. Das verwendete Verfahren hat zwar erfolgreich die Fehlerfunktion minimiert und das erklärte Ziel der Überschaubarkeit erfüllt, sicherlich kann man aber die Optimierung noch erheblich beschleunigen. Unter Verwendung von zusätzlichem Wissen über die Fehlerfunktion ist hier noch einiges an Potential vorhanden. Auch muß sich über eine analytische Berechnung der Fehlerfunktion Gedanken gemacht werden, damit andere Klassen von Optimierungsalgorithmen zur Anwendung kommen könnten.

In Sachen Rechenzeit und Fehlerfunktion ist auch zu erwähnen, daß die Rechenzeit stark mit der Anzahl der zu matchenden Punkte gekoppelt ist. Da der rechenintensivste Schritt bei dem Matching die Berechnung des Fehlermaßes in jedem Schritt der Optimierung ist und die Rechenzeit eines Fehlerwertes mit jedem zusätzlichen Punkt in der Punktwolke zunimmt, ist die Frage nach der absolut nötigen Anzahl an Punkten von höchstem Interesse.

Weiterhin stellt sich die Frage nach einer Automatisierung der Registrierung. Das in der vorliegenden Arbeit implementierte Verfahren erfordert noch die Interaktion eines Benutzers zur Vorpositionierung der Punktwolke. Sollte sich eine Vollautomatisierung dieses Schrittes aber als nicht realisierbar herausstellen, so könnte man mit entsprechender Rechenleistung die Punktwolke in Zukunft auch mit Hilfe eines sogenannten haptischen Interfaces vorpositionieren.

Abgesehen davon, daß das Verfahren mit Sicherheit noch andere Anwendungsgebiete als die Medizin hat, kann man durch die Bestimmung der Lage des Patienten mit Hilfe des Scanners, Daten aus der Operationsplanung nicht nur, wie der in der Einleitung erwähnt, auf den vorliegenden Patienten transformieren, sondern es ist ebenfalls daran gedacht diese Daten, mit Hilfe des LCD-Projektors des Scanners, optisch direkt auf den Patienten zu projizieren. Bei gleichzeitiger Beleuchtung durch einen OP-Strahler müßte die Lichtstärke der Projektoren aber noch erhöht werden, damit die Information kontraststark auf dem Patienten zu sehen ist.

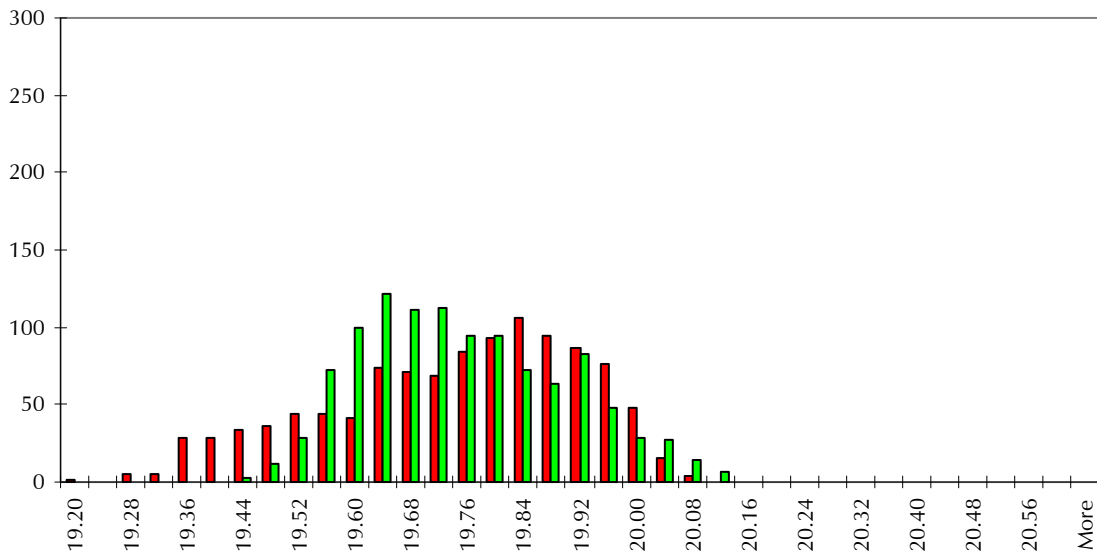
Was die Anwendbarkeit eines Oberflächenscanner zur Lagebestimmung in der Medizin betrifft, so kann abschließend gesagt werden, daß man zum jetzigen Zeitpunkt die Verfolgung dieser Idee nur unterstützen kann.

Damit wünsche ich allen die sich diesem Thema widmen und widmen werden viel Erfolg und gutes Gelingen.

Anhang A

Meßprotokolle

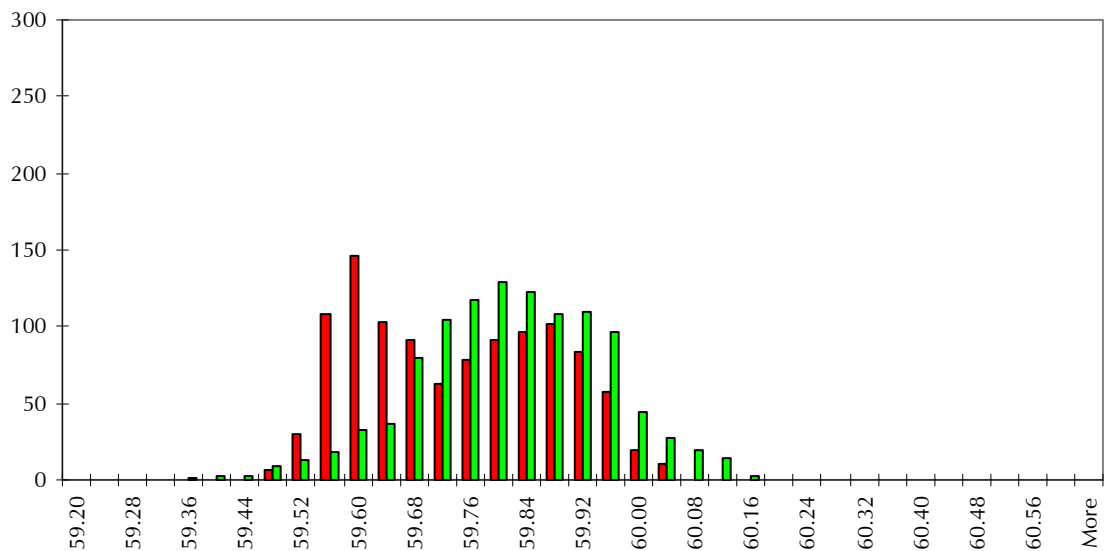
A.1 Zwischenebene konstanter Höhe $z = 19.925$ mm



<i>Links (grün)</i>	
Mean	19.73287
Standard Error	0.004331
Median	19.7154
Mode	19.6507
Standard Deviation	0.142873
Sample Variance	0.020413
Kurtosis	-0.63548
Skewness	0.336345
Range	0.7059
Minimum	19.4074
Maximum	20.1133
Sum	21469.36
Count	1088
Confidence Level(95.0%)	0.008499

<i>Rechts (rot)</i>	
Mean	19.720853
Standard Error	0.0054525
Median	19.75025
Mode	19.7909
Standard Deviation	0.1798502
Sample Variance	0.0323461
Kurtosis	-0.6194
Skewness	-0.478877
Range	0.8575
Minimum	19.1942
Maximum	20.0517
Sum	21456.288
Count	1088
Confidence Level(95.0%)	0.0106986

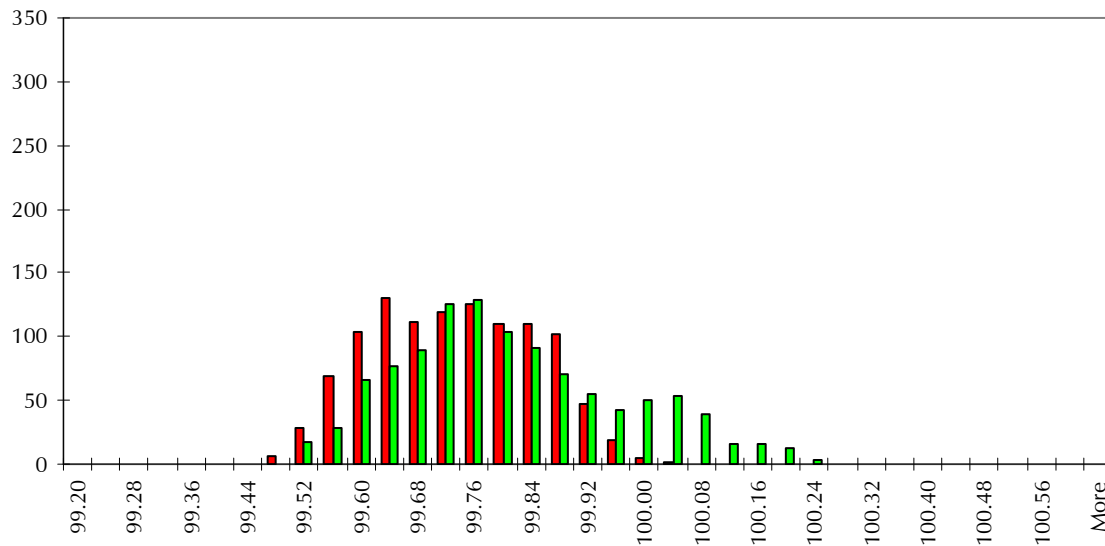
A.2 Zwischenebene konstanter Höhe $z = 59.925$ mm



<i>Links (grün)</i>	
Mean	59.79849
Standard Error	0.004019
Median	59.7991
Mode	59.8052
Standard Deviation	0.132559
Sample Variance	0.017572
Kurtosis	0.00304
Skewness	-0.1816
Range	0.8199
Minimum	59.3317
Maximum	60.1516
Sum	65060.76
Count	1088
Confidence Level(95.0%)	0.007885

<i>Rechts (rot)</i>	
Mean	59.721499
Standard Error	0.004161
Median	59.7188
Mode	59.529
Standard Deviation	0.1372484
Sample Variance	0.0188371
Kurtosis	-1.177973
Skewness	0.1581683
Range	0.5676
Minimum	59.4659
Maximum	60.0335
Sum	64976.991
Count	1088
Confidence Level(95.0%)	0.0081644

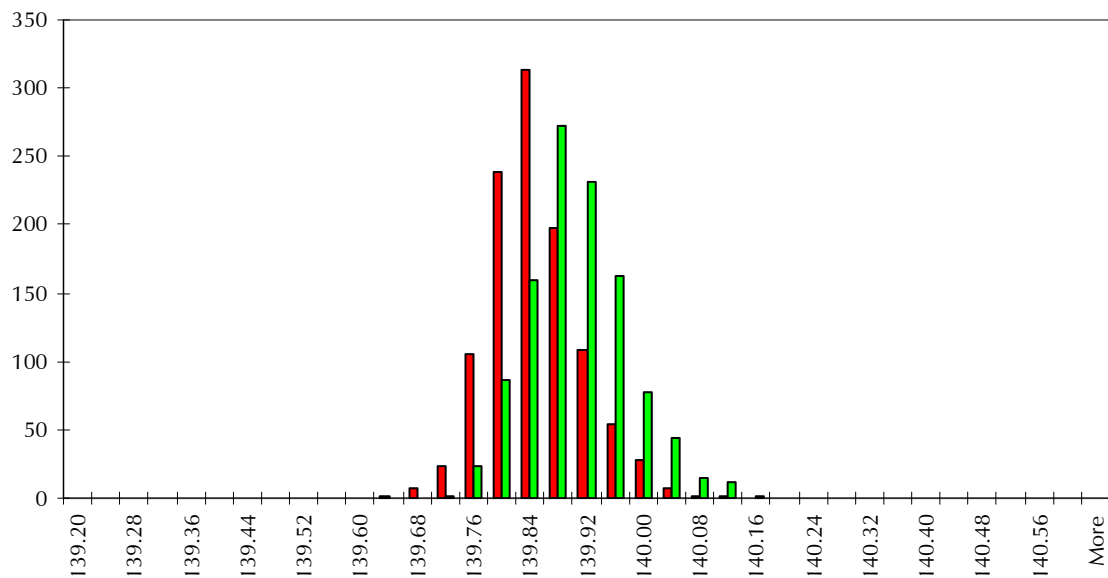
A.3 Zwischenebene konstanter Höhe $z = 99.925 \text{ mm}$



<i>Links (grün)</i>	
Mean	99.78976
Standard Error	0.004776
Median	99.76585
Mode	100.01
Standard Deviation	0.157525
Sample Variance	0.024814
Kurtosis	-0.42961
Skewness	0.480649
Range	0.7379
Minimum	99.4811
Maximum	100.219
Sum	108571.3
Count	1088
Confidence Level(95.0%)	0.009371

<i>Rechts (rot)</i>	
Mean	99.712247
Standard Error	0.003456
Median	99.71345
Mode	99.6129
Standard Deviation	0.113994
Sample Variance	0.0129946
Kurtosis	-0.876618
Skewness	0.0232822
Range	0.5623
Minimum	99.4497
Maximum	100.012
Sum	108486.92
Count	1088
Confidence Level(95.0%)	0.0067811

A.4 Zwischenebene konstanter Höhe $z = 139.925 \text{ mm}$

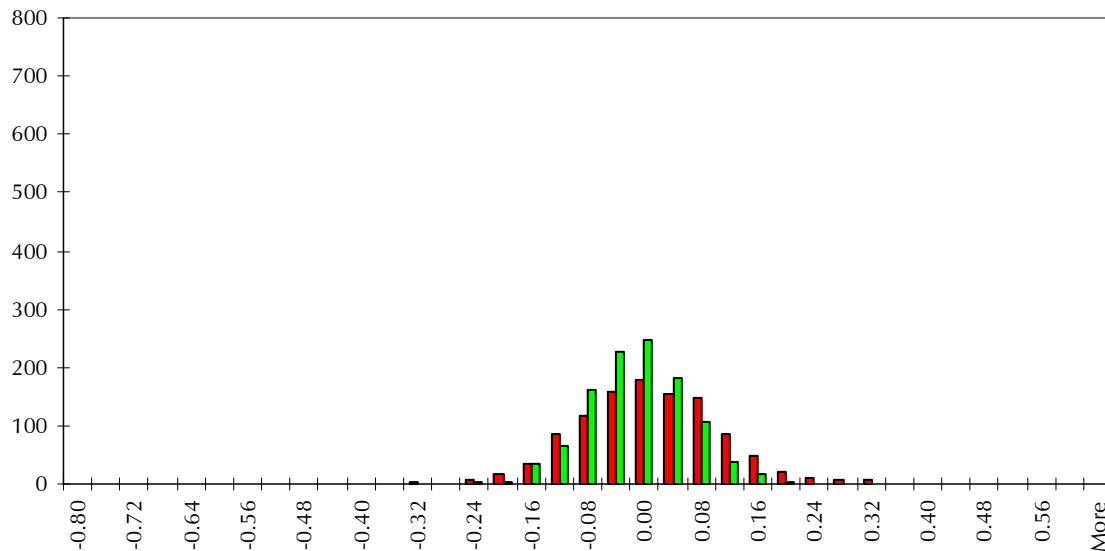


<i>Links (grün)</i>	
Mean	139.8865
Standard Error	0.00214
Median	139.8805
Mode	139.849
Standard Deviation	0.070595
Sample Variance	0.004984
Kurtosis	0.388069
Skewness	0.470645
Range	0.431
Minimum	139.703
Maximum	140.134
Sum	152196.5
Count	1088
Confidence Level(95.0%)	0.004199

<i>Rechts (rot)</i>	
Mean	139.82775
Standard Error	0.0019337
Median	139.822
Mode	139.822
Standard Deviation	0.0637817
Sample Variance	0.0040681
Kurtosis	0.6462785
Skewness	0.5085954
Range	0.469
Minimum	139.621
Maximum	140.09
Sum	152132.59
Count	1088
Confidence Level(95.0%)	0.0037941

A.5 Kalibrationsebene konstanter Höhe $z = 0$ mm

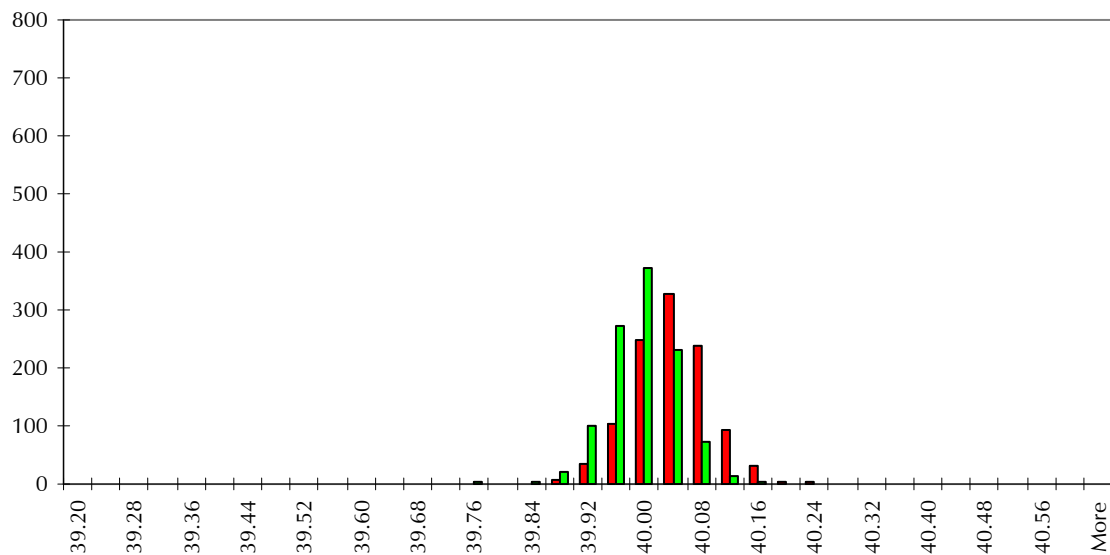
!!! Achtung: Skalierung im Histogramm auf 800 statt 350 wie bisher !!!



<i>Links (grün)</i>	
Mean	-0.01134
Standard Error	0.003091
Median	-0.01215
Mode	0
Standard Deviation	0.101948
Sample Variance	0.010393
Kurtosis	1.10047
Skewness	0.235642
Range	0.8988
Minimum	-0.4021
Maximum	0.4967
Sum	-12.3378
Count	1088
Confidence Level(95.0%)	0.006065

<i>Rechts (rot)</i>	
Mean	-0.030001
Standard Error	0.002141
Median	-0.0296
Mode	0
Standard Deviation	0.0706211
Sample Variance	0.0049873
Kurtosis	0.1357489
Skewness	0.0722328
Range	0.5214
Minimum	-0.2553
Maximum	0.2661
Sum	-32.6416
Count	1088
Confidence Level(95.0%)	0.004201

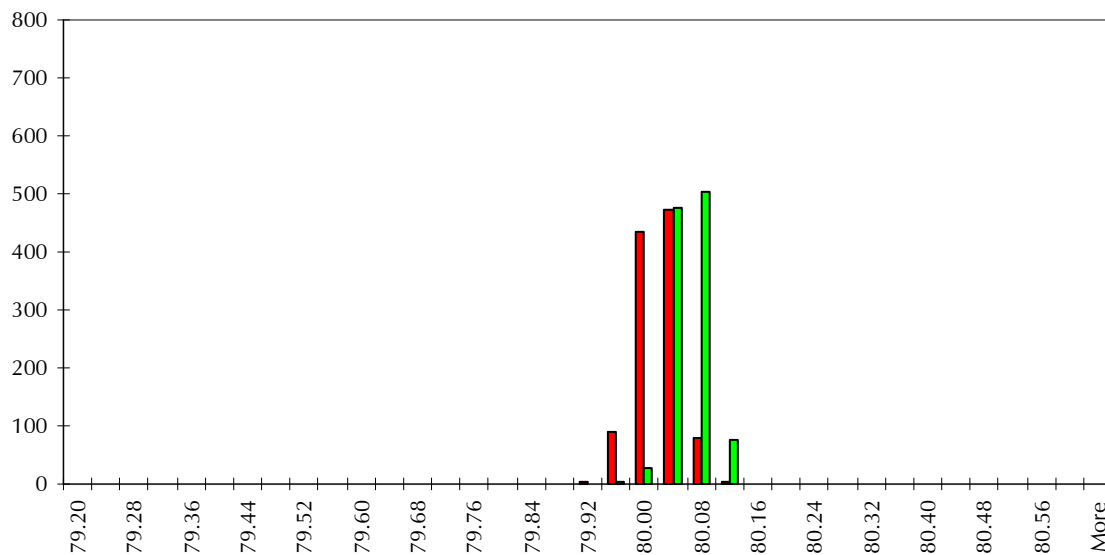
A.6 Kalibrationsebene konstanter Höhe $z = 40$ mm



<i>Links (grün)</i>	
Mean	40.01801
Standard Error	0.001662
Median	40.0188
Mode	40.02
Standard Deviation	0.054805
Sample Variance	0.003004
Kurtosis	0.466832
Skewness	0.064627
Range	0.4043
Minimum	39.8407
Maximum	40.245
Sum	43539.6
Count	1088
Confidence Level(95.0%)	0.00326

<i>Rechts (rot)</i>	
Mean	39.975424
Standard Error	0.0014748
Median	39.97635
Mode	40.02
Standard Deviation	0.0486446
Sample Variance	0.0023663
Kurtosis	1.1613319
Skewness	-0.239069
Range	0.4022
Minimum	39.7415
Maximum	40.1437
Sum	43493.261
Count	1088
Confidence Level(95.0%)	0.0028937

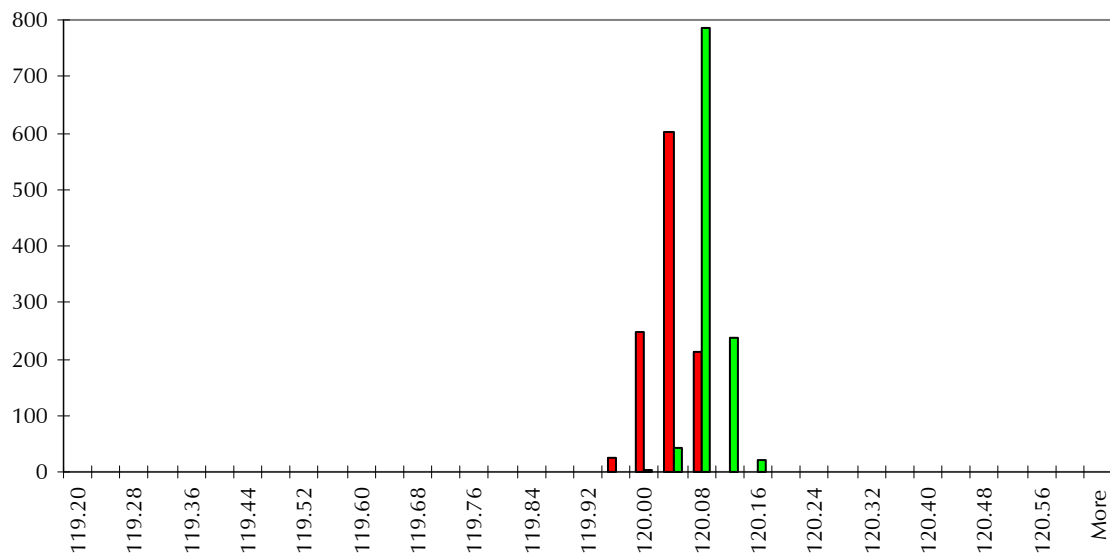
A.7 Kalibrationsebene konstanter Höhe $z = 80$ mm



<i>Links (grün)</i>	
Mean	80.00062
Standard Error	0.000878
Median	80.001
Mode	80.04
Standard Deviation	0.028974
Sample Variance	0.000839
Kurtosis	0.415715
Skewness	-0.16526
Range	0.2065
Minimum	79.8889
Maximum	80.0954
Sum	87040.68
Count	1088
Confidence Level(95.0%)	0.001724

<i>Rechts (rot)</i>	
Mean	80.043724
Standard Error	0.000731
Median	80.04315
Mode	80.04
Standard Deviation	0.0241115
Sample Variance	0.0005814
Kurtosis	0.6823514
Skewness	0.1062538
Range	0.1718
Minimum	79.9472
Maximum	80.119
Sum	87087.572
Count	1088
Confidence Level(95.0%)	0.0014343

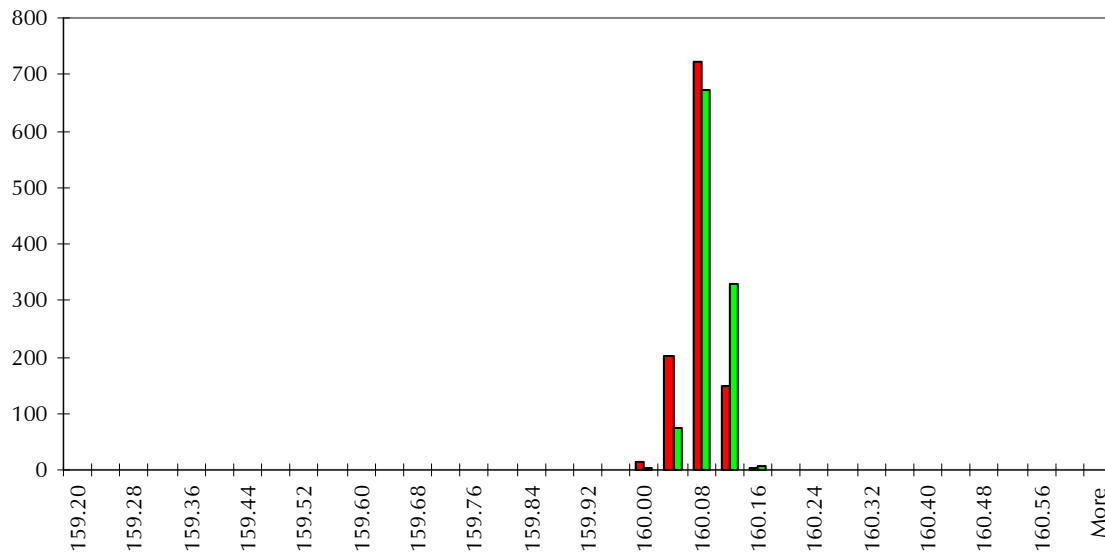
A.8 Kalibrationsebene konstanter Höhe $z = 120$ mm



<i>Links (grün)</i>	
Mean	120.0178
Standard Error	0.000792
Median	120.022
Mode	120.035
Standard Deviation	0.026138
Sample Variance	0.000683
Kurtosis	0.281633
Skewness	-0.59745
Range	0.188
Minimum	119.894
Maximum	120.082
Sum	130579.4
Count	1088
Confidence Level(95.0%)	0.001555

<i>Rechts (rot)</i>	
Mean	120.07021
Standard Error	0.0005905
Median	120.068
Mode	120.06
Standard Deviation	0.019477
Sample Variance	0.0003794
Kurtosis	1.9433101
Skewness	0.4150472
Range	0.172
Minimum	119.973
Maximum	120.145
Sum	130636.39
Count	1088
Confidence Level(95.0%)	0.0011586

A.9 Kalibrationsebene konstanter Höhe $z = 160$ mm



<i>Links (grün)</i>	
Mean	160.0584
Standard Error	0.00069
Median	160.06
Mode	160.055
Standard Deviation	0.022768
Sample Variance	0.000518
Kurtosis	0.593613
Skewness	-0.20342
Range	0.168
Minimum	159.973
Maximum	160.141
Sum	174143.5
Count	1088
Confidence Level(95.0%)	0.001354

<i>Rechts (rot)</i>	
Mean	160.07197
Standard Error	0.0006316
Median	160.074
Mode	160.08
Standard Deviation	0.0208335
Sample Variance	0.000434
Kurtosis	1.6723544
Skewness	-0.32396
Range	0.191
Minimum	159.989
Maximum	160.18
Sum	174158.3
Count	1088
Confidence Level(95.0%)	0.0012393

A.10 Schiefe Ebene 1, Steigung in -x-Richtung

<i>Links (grün)</i>	
Mean	0.200258
Standard Error	4.42E-05
Median	0.200239
Mode	#N/A
Standard Deviation	0.001456
Sample Variance	2.12E-06
Kurtosis	-0.05288
Skewness	0.161997
Range	0.008808
Minimum	0.196178
Maximum	0.204986
Sum	216.8795
Count	1083
Confidence Level(95.0%)	8.68E-05

<i>Rechts (rot)</i>	
Mean	0.2005733
Standard Error	7.105E-05
Median	0.200508
Mode	#N/A
Standard Deviation	0.0023362
Sample Variance	5.458E-06
Kurtosis	-0.164685
Skewness	0.0524855
Range	0.0131505
Minimum	0.1946081
Maximum	0.2077586
Sum	216.81973
Count	1081
Confidence Level(95.0%)	0.0001394

A.11 Schiefe Ebene 2, Steigung in -x-Richtung

<i>Links (grün)</i>	
Mean	0.200222
Standard Error	5.11E-05
Median	0.2003
Mode	#N/A
Standard Deviation	0.00168
Sample Variance	2.82E-06
Kurtosis	0.159856
Skewness	-0.22121
Range	0.011741
Minimum	0.194735
Maximum	0.206475
Sum	216.6404
Count	1082
Confidence Level(95.0%)	0.0001

<i>Rechts (rot)</i>	
Mean	0.1996181
Standard Error	8.82E-05
Median	0.1995584
Mode	#N/A
Standard Deviation	0.0029013
Sample Variance	8.417E-06
Kurtosis	-0.355495
Skewness	0.0062773
Range	0.0162163
Minimum	0.1912209
Maximum	0.2074372
Sum	215.9868
Count	1082
Confidence Level(95.0%)	0.0001731

A.12 Schiefe Ebene 3, Steigung in -x-Richtung

<i>Links (grün)</i>	
Mean	0.20039
Standard Error	5.97E-05
Median	0.200443
Mode	#N/A
Standard Deviation	0.001963
Sample Variance	3.85E-06
Kurtosis	-0.0483
Skewness	-0.11948
Range	0.012989
Minimum	0.19309
Maximum	0.20608
Sum	216.8218
Count	1082
Confidence Level(95.0%)	0.000117

<i>Rechts (rot)</i>	
Mean	0.2000836
Standard Error	0.0001052
Median	0.2000921
Mode	#N/A
Standard Deviation	0.0034594
Sample Variance	1.197E-05
Kurtosis	-0.495945
Skewness	-0.016693
Range	0.0175993
Minimum	0.1911138
Maximum	0.2087131
Sum	216.49043
Count	1082
Confidence Level(95.0%)	0.0002064

A.13 Schiefe Ebene 4, Steigung in -x-Richtung

<i>Links (grün)</i>	
Mean	0.200064
Standard Error	2.73E-05
Median	0.20009
Mode	#N/A
Standard Deviation	0.000897
Sample Variance	8.04E-07
Kurtosis	-0.17932
Skewness	-0.19671
Range	0.005327
Minimum	0.197102
Maximum	0.202429
Sum	216.4691
Count	1082
Confidence Level(95.0%)	5.35E-05

<i>Rechts (rot)</i>	
Mean	0.2002186
Standard Error	4.423E-05
Median	0.2002157
Mode	#N/A
Standard Deviation	0.0014547
Sample Variance	2.116E-06
Kurtosis	0.9455222
Skewness	0.0952729
Range	0.0114329
Minimum	0.1945968
Maximum	0.2060297
Sum	216.63651
Count	1082
Confidence Level(95.0%)	8.678E-05

A.14 Schiefe Ebene 1, Steigung in y-Richtung

<i>Links (grün)</i>	
Mean	0.081744
Standard Error	5.29E-05
Median	0.081755
Mode	#N/A
Standard Deviation	0.001235
Sample Variance	1.52E-06
Kurtosis	-0.49491
Skewness	-0.14419
Range	0.006909
Minimum	0.077931
Maximum	0.08484
Sum	44.46873
Count	544
Confidence Level(95.0%)	0.000104

<i>Rechts (rot)</i>	
Mean	0.0818668
Standard Error	6.748E-05
Median	0.0819193
Mode	#N/A
Standard Deviation	0.001574
Sample Variance	2.477E-06
Kurtosis	-0.532064
Skewness	-0.13265
Range	0.008155
Minimum	0.0774913
Maximum	0.0856463
Sum	44.535553
Count	544
Confidence Level(95.0%)	0.0001326

A.15 Schiefe Ebene 2, Steigung in y-Richtung

<i>Links (grün)</i>	
Mean	0.081856
Standard Error	6.27E-05
Median	0.081912
Mode	#N/A
Standard Deviation	0.001463
Sample Variance	2.14E-06
Kurtosis	-0.29274
Skewness	-0.08178
Range	0.00918
Minimum	0.077224
Maximum	0.086404
Sum	44.52956
Count	544
Confidence Level(95.0%)	0.000123

<i>Rechts (rot)</i>	
Mean	0.0817819
Standard Error	8.106E-05
Median	0.0817315
Mode	#N/A
Standard Deviation	0.0018905
Sample Variance	3.574E-06
Kurtosis	-0.760857
Skewness	0.0195244
Range	0.0089254
Minimum	0.077267
Maximum	0.0861925
Sum	44.489378
Count	544
Confidence Level(95.0%)	0.0001592

A.16 Schiefe Ebene 3, Steigung in y-Richtung

<i>Links (grün)</i>	
Mean	0.082192
Standard Error	9.87E-05
Median	0.082051
Mode	#N/A
Standard Deviation	0.002301
Sample Variance	5.3E-06
Kurtosis	0.278265
Skewness	0.166501
Range	0.013168
Minimum	0.075401
Maximum	0.088569
Sum	44.71271
Count	544
Confidence Level(95.0%)	0.000194

<i>Rechts (rot)</i>	
Mean	0.0817355
Standard Error	8.654E-05
Median	0.0817251
Mode	#N/A
Standard Deviation	0.0020185
Sample Variance	4.074E-06
Kurtosis	0.9863809
Skewness	-0.098858
Range	0.012974
Minimum	0.075276
Maximum	0.08825
Sum	44.464117
Count	544
Confidence Level(95.0%)	0.00017

A.17 Schiefe Ebene 4, Steigung in y-Richtung

<i>Links (grün)</i>	
Mean	0.081775
Standard Error	4.34E-05
Median	0.081742
Mode	#N/A
Standard Deviation	0.001012
Sample Variance	1.02E-06
Kurtosis	-0.22489
Skewness	0.113112
Range	0.006174
Minimum	0.079162
Maximum	0.085336
Sum	44.48582
Count	544
Confidence Level(95.0%)	8.52E-05

<i>Rechts (rot)</i>	
Mean	0.0820764
Standard Error	3.682E-05
Median	0.082099
Mode	#N/A
Standard Deviation	0.0008587
Sample Variance	7.373E-07
Kurtosis	0.6772334
Skewness	-0.175488
Range	0.0061101
Minimum	0.0784873
Maximum	0.0845974
Sum	44.649542
Count	544
Confidence Level(95.0%)	7.232E-05

Literaturverzeichnis

Literaturverzeichnis

Da sich die Suche nach Literatur zum Oberflächenscanner als sehr schwierig herausgestellt hat, habe ich mir erlaubt einige Bücher zu diesem Thema hier kurz zu kommentieren, damit andere nicht wieder bei Null anfangen müssen.

- [AANS 93] American Association of Neurological Surgeons. Interactive Image-Guided Neurosurgery. R. Maciunas (Ed.). USA 1993.
- [Breuckmann 92] B. Breuckmann. Bildverarbeitung und optische Meßtechnik in der industriellen Praxis. Franzis Verlag. München 1993.
(Rem: Referenzbuch um Überblick zu gewinnen)
- [Bronstein 91] I.N. Bronstein und K.A. Semendjajew. Taschenbuch der Mathematik. 25. Auflage. Teubner Verlag. Stuttgart 1991.
- [Creath 86] K. Creath. Comparison of Phase-Measurement Algorithms. in: Proceedings SPIE Vol. 680. Surface Characterization and Testing. 1986.
- [Creath 88] K. Creath. Phase-Measurement Interferometry Techniques. in: E. Wolf (Ed.). Progress in Optics. Volume XXVI. Elsevier Science Publishers B.V. North-Holland 1988.
- [Dillmann 91] R. Dillmann, M. Huck. Informationsverarbeitung in der Robotik. Springer Verlag. Berlin 1991.
- [Eggink 95] B. Eggink. Die C++ iostreams-Library. Hanser Verlag.
(Rem: Füllt die IO-Lücke der meisten C++-Bücher, Prädikat: sehr wertvoll)
- [Gerl 97] S. Gerl. 3D-Gesichterkennung mit selbstorganisierendem mehrkanaligem Matching-Verfahren. VDI-Fortschrittberichte. Düsseldorf 1997.
(Rem: Der Lichtblick, endlich mal was über codierten Lichtansatz u.s.w., vielversprechende Literaturangaben)
- [Günzl 94] M. Günzl. Integration von röntgentomographischen und kernspintomographischen Aufnahmen. Diplomarbeit. Universität Karlsruhe. Fakultät für Informatik. Institut für Prozeßrechner- und Robotik. Karlsruhe 1994.

-
- [Grabowski 96] H. Grabowski. Konsistente Oberflächenmodellierung medizinischer Bilddaten auf der Basis der dreidimensionalen Delaunay-Triangulierung. Diplomarbeit. Universität Karlsruhe. Fakultät für Informatik. Institut für Prozeßrechentechnik und Robotik. Karlsruhe 1994.
- [Grimson 95] W.E.L. Grimson, G.J. Ettinger, S.J. White, P.L. Gleason, T. Lozano-pérez, W.M. Welle III and R. Kikinis. Evaluating and Validating an automated Registration System for Enhanced Reality Visualization in Surgery. in: N. Ayache (Ed.). Lecture Notes in Computer Science Nr. 905. First International Conference on Computer Vision, Virtual Reality and Robotics in Medicine. CVRMed '95. Springer Verlag. Nice 1995.
- [Hamadeh 95] A. Hamadeh, S. Lavallée, R.Szeliski, P. Cinquin, and O. Péria. Anatomy-based Registration for Computer-integrated Surgery. in: N. Ayache (Ed.). Lecture Notes in Computer Science Nr. 905. First International Conference on Computer Vision, Virtual Reality and Robotics in Medicine. CVRMed '95. Springer Verlag. Nice 1995.
- [Hof 92] A. Hof, A. Hanßen, M.Küchel, J.Schütz. Streifenprojektionsmethode in der 3D-Meßtechnik. In: Fertigungsmeßtechnik und Qualitätssicherung. S. 67-80. VDI-Berichte. Nr.1006. Düsseldorf 1992.
- (Rem: Der Aufsatz ist eine ganzheitliche Betrachtung der verschiedenen Streifenprojektionsverfahren, deren Probleme und deren Grenzen. Es wird jeweils auf spezifischere Literatur verwiesen.)
- [Ioannides 93] M. Ioannides, A. Wehr. Von der 3D-Digitalisierung über die Mustererkennung zur Fertigung. In: S.J. Pöpl, H.Handels (Hrsg.). Mustererkennung 1993, S. 717-724. 15. DAGM-Symposium. Mustererkennung im Dienste der Gesundheit. Lübeck 1993. Informatik aktuell. Springer Verlag. Berlin Heidelberg 1993.
- (Rem: wie in [Gerl 97] angekündigt geht es nur um Oberflächenrekonstruktionsalgorithmen, keine Informationen zur Aufnahmetechnik)

- [Klette 96] R. Klette, A. Koschan, K. Schlüns. Computer Vision. Vieweg Verlag. Wiesbaden 1996.
- (Rem: guter Überblick über Lichtschnittverfahren und codierten Lichtansatz, die Sache wird auch rechnerisch durchgeixt, gute kommentierte Literaturhinweise)
- [McCloy 89] D. McCloy, D.M. J. Harris. Robotertechnik. VCH-Verlag. Weinheim 1989.
- [Meyberg 90] K. Meyberg und P.Vachenauer. Höhere Mathematik 1. Springer Verlag. Berlin 1990.
- [Microscribe] <http://www.immerse.com/WWWPages>
- [Osten 91] W. Osten. Digitale Verarbeitung und Auswertung von Interferenzbildern. Akademie Verlag, Berlin 1991.
- (Rem: leitet mathematisch sauber und verständlich 7 Methoden zur Rekonstruktion der Phasenverteilung in Interferenzbildern her)
- [Pokrandt 97] P. Pokrandt. Automatische Überlagerung medizinischer Bilddatensätze. VDI-Fortschrittberichte. Düsseldorf 1997.
- [Reincke 83] M.A. Reincke. Lichtschnitt-Technik: der Versuch einer zerstörungsfreien Prüfung geschweißter Kunststoffrohre. Dissertation. Aachen 1983.
- [RSI 97] Digiface. Komplettsystem zur optischen 3D-Digitalisierung des menschlichen Gesichtes bzw. Kopfes. Bedienungsanleitung und Dokumentaion. RSI GmbH. Oberursel 1997.
- [Schlicht 95] H.-J. Schlicht. Bildverarbeitung digital: Scanner, Drucker, Video, Multimedia unter Windows. Addison-Wesley. 1995.
- (Rem: das war nix)
- [Schröder 84] G. Schröder. Technische Optik, 4.Auflage. Vogel Verlag. Würzburg 1984.
- [Schlipf 93] A. Schlipf. CAD 3D-Modellierung aus Laserscanner-Abstandsdaten. Diplomarbeit. FZI Karlsruhe. Karlsruhe 1993.
- (Rem: optische Messverfahren nur in der Einleitung angerissen, aber vielversprechende Literaturhinweise)

-
- [Schumann 91] H.-G. Schumann. Turbo Borland C++ 3.0 für Windows. Einstieg in die objektorientierte und Windows-Programmierung. Addison-Wesley Verlag. Bonn 1991.
- [Stehle 97] W. Stehle. Allgemeine Optimierungsverfahren. Skriptum zur Vorlesung. Institut für Nachrichtensysteme. Fakultät Elektrotechnik. Universität Karlsruhe. Sommersemester 1997.
- [Strutz 92] T. Strutz, W. Riechmann, T. Stahs. Tiefendatengewinnung mit dem codierten Lichtansatz - Einsatzmöglichkeiten in der Automobilindustrie. Querschnittsverarbeitung Bildverarbeitung. Deutsche Gesellschaft für zerstörungsfreie Prüfung e.V., Berlin 1992.
- [Wahl 86] F. M. Wahl. A coded light approach for depth map acquisition. In: G. Hartmann (Hrsg.). Mustererkennung 1986. S. 12-17. 8. DAGM-Symposium. Paderborn September/Oktober 1986. Proceedings. Informatik-Fachberichte 125. Springer Verlag. Berlin Heidelberg 1986
- (Rem: extrem kurze Beschreibung des codierten Lichtansatzes, wenig hilfreich für die vorliegende Arbeit)
- [Wolf 92] H. Wolf. Schnelle dreidimensionale Bilderfassung mit dem codierten Lichtansatz. VDI-Berichte 939, S. 393-398. VDI-Verlag 1992.
- [Zao 96] J. Zhao, PTE Roberts, ACF Colchester and K. Holton-Tainter. Registration Updating Using Marker Pins in a Video Based Neurosurgical Guidance System. in: K.-H. Höhne (Ed.). Lecture Notes in Computer Science Nr. 1131. Visualization in Biomedical Computing. 4th International Conference, VBC '96. Springer Verlag. Hamburg 1996.

