

Ein effizientes, sensorunabhängiges Austauschformat für Punktwolken und die 3D-Sensorgeometrie

Peter Johannes Neugebauer

Fraunhofer-Institut für Graphische Datenverarbeitung
Rundeturmstraße 6, 64283 Darmstadt, Germany
email: neugeb@igd.fhg.de
<http://www.igd.fhg.de/www/igd-a7>

Zusammenfassung

Mit dem SCULPTOR-System können vollständige Objektgeometrien aus mehreren 3D-Aufnahmen rekonstruiert werden. Dabei werden sowohl die Koordinaten der Meßpunkte als auch die vektorielle Richtung der Sichtstrahlen verwendet, aus der jeder Punkt erfaßt worden ist. Das hier vorgestellte Austauschformat erlaubt es, die 3D-Daten speichereffizient abzulegen und die vektorielle Richtung des jeweiligen Sichtstrahls in sensorunabhängiger Form vom 3D-Sensor an die Applikation zu übermitteln. Daneben bietet es erhebliche Vorteile bei der Weiterverarbeitung der Daten.

1 Einleitung

Die dreidimensionale Erfassung von Objekten durch 3D-Digitalisierung und Mehrbildintegration hat in den letzten Jahren erhebliche Fortschritte gemacht. Mit dem beim Fraunhofer-IGD entwickelten SCULPTOR-System ist es ohne großen Aufwand möglich, die Mehrbildintegration von 3D-Sensordaten durchzuführen[7, 8]. Nachdem das Objekt aus mehreren Ansichten mit einem 3D-Sensor vermessen wurde, werden die verschiedenen 3D-Aufnahmen durch Registrierung in ein gemeinsames Weltkoordinatensystem überführt. Durch Ausnutzung von Redundanzen in den Überlappungsbereichen der einzelnen Bilder wird dabei die Bildqualität über das Sensorrauschen hinaus gesteigert. Das Modell wird zuerst aus einem Volumenblock herausgearbeitet, indem alle Volumenbereiche eliminiert werden, die zwischen den Meßpunkten und dem 3D-Scanner entlang des (optischen) Meßstrahls zu liegen kommen. Mittels der Skulpturierungs- und Triangulierungsroutinen wird letztlich das dreidimensionale Modell gewonnen.

Eine entscheidende Kernidee bei diesem Ansatz ist, daß etwaige Mehrdeutigkeiten bei der Rekonstruktion dadurch beseitigt werden können, daß die individuelle Beobachtungs- oder Scan-Richtung von jedem Meßpunkt (siehe Abb. 1) bei der Rekonstruktion berücksichtigt wird. Dabei hat sich im Laufe der Entwicklung gezeigt, daß die Zusatzinformation über die Abstrichrichtung und damit der Sensorgeometrie sehr viel mehr an Informationen beisteuert, als es mit einer reinen Punktwolke möglich wäre. Diese Zusatzinformation könnte auch für andere Applikationen von starkem Interesse sein. Ein Ziel der in diesem Artikel beschriebenen Arbeit besteht darin, 3D-Daten von unterschiedlichsten Sensoren mit dem SCULPTOR-System einlesen und bearbeitet zu können.

Die Entwicklung der 3D-Sensorik hat im Laufe der Zeit die verschiedensten physikalischen Meßprinzipien hervorgebracht. Um die weiterverarbeitende Software vom eigentlich Meßprinzip zu entkoppeln, werden die Meßdaten oftmals als Punktwolke $P = \{(x_i, y_i, z_i) | i = 1, \dots, n\}$ an die weiterverarbeitende Software übergeben. Diese *Low-Level* Datenübergabe hat den Vorteil, daß die Weiterverarbeitung in keinsten Weise an das Meßprinzip und den spezifischen Sensor gekoppelt ist. Darüberhinaus können ungeordnete Punktwolken einfach gespeichert und ohne praktische Probleme an die verschiedensten Auswerteprogramme weitergereicht werden. Allerdings wird in dieser einfachen Repräsentationsform nur ein Teil der vom Sensor gewonnenen Information an die weiterführende Software übergeben. Nachbarschaftsbeziehungen zwi-

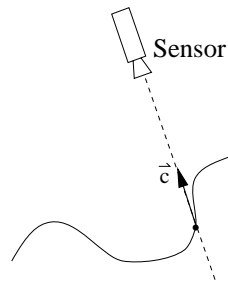


Abbildung 1: Individuelle Scan-Richtung \vec{c} des 3D-Sensors für einen Meßpunkt.

schen den Punkten und die Sensorgeometrie gehen verloren. Eine Beibehaltung der Nachbarschaftsbeziehungen mit dem Vorteil einer einfachen Dateninterpolation kann mit der Speicherung der Meßwerte in einer Matrix erreicht werden.

$$P = \{(x(u, v), y(u, v), z(u, v))^T \mid u = 0, \dots, n_u - 1 \quad v = 0, \dots, n_v - 1\} \quad (1)$$

Aber auch hier wird die Sensorgeometrie nicht repräsentiert. Zudem ist dieses Datenaustauschformat speicherintensiv, da für jeden Meßpunkt 3 Gleitkomma-Werte gespeichert werden müssen. Diese Beispiele zeigen, daß die derzeitigen Austauschformate aus unserer Sicht unzureichend sind. Diesen Umstand haben wir zum Anlaß genommen, um ein neues Austauschformat für bildgebende 3D-Meßverfahren zu definieren, welches sowohl die Sensorgeometrie wiedergibt als auch eine effiziente Datenrepräsentation ermöglicht. Durch die Voraussetzung einer gradlinigen Strahlausbreitung, die bei optischen Sensoren¹ obligatorisch ist, kann die Sensorgeometrie dabei vollständig parametrisiert abgelegt werden.

2 Gängige Sensorgeometrien

In der 3D-Meßtechnik werden eine Reihe von bildgebenden 3D-Sensortypen eingesetzt. Zu den bildgebenden Sensoren zählen sowohl punktsequentielle, liniensequentielle als auch flächenhaft vermessene Verfahren, solange die Meßwerte in einer Bildmatrix $\zeta(u, v)$ gespeichert werden können. Die Bildmatrix sagt dabei noch nichts über die Abtastgeometrie aus.

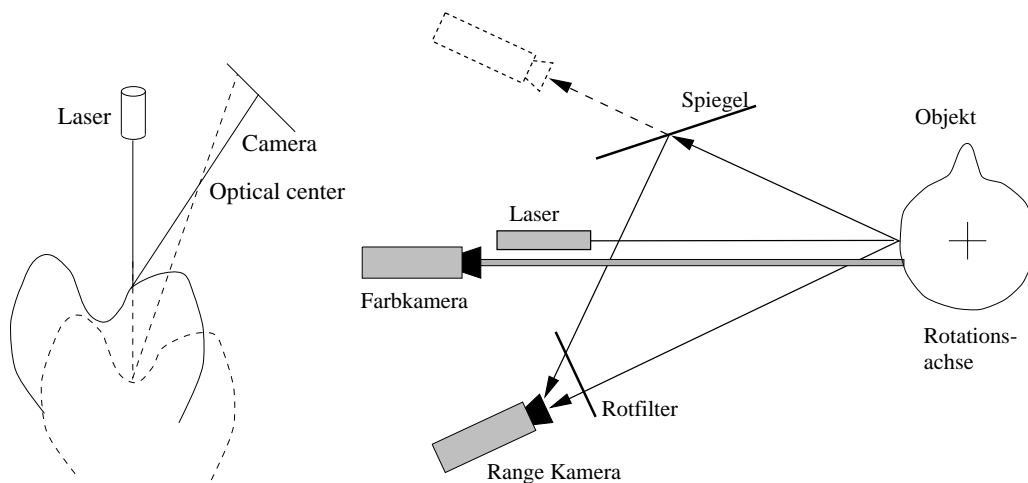


Abbildung 2: Links: Das Triangulationsprinzip. Rechts: Aufbau des Cyberware-Kopfscanners.

¹... außer natürlich im Schwerkraftfeld eines schwarzen Loches.

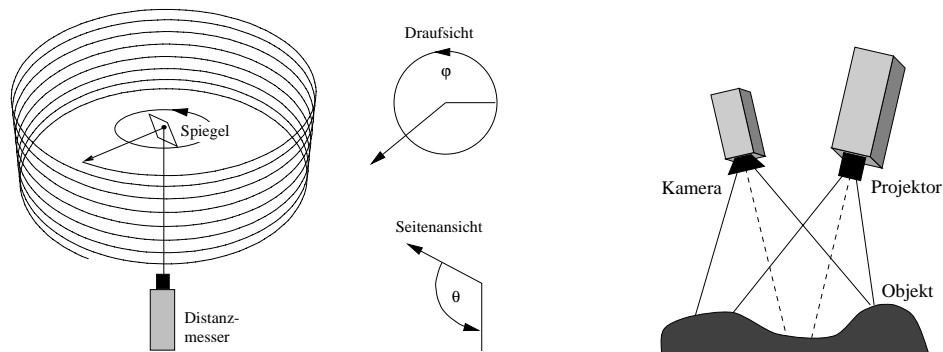


Abbildung 3: Links: 3D-Scanner basierend auf Laufzeit- oder Phasenmessung von modulierten Laserstrahlen. Rechts: Sensoraufbau für die Streifenlichtprojektion.

Für die Vermessung selbst werden verschiedene physikalische Meßprinzipien genutzt. Die Meßprinzipien können grob in die Klasse der passiven und der aktiven Verfahren unterteilt werden. Bei den passiven Verfahren wird das Objekt nur unter Einbeziehung der natürlichen Umgebungsbedingungen, also z.B. der Raumbeleuchtung, mit passiven Sensoren wie CCD-Kameras aufgenommen. Zu diesen Verfahren gehören Fokussierungs-Systeme, Stereo- und photogrammetrische Systeme. Bei den aktiven Verfahren sind neben den passiven Komponenten auch aktive Elemente beteiligt. Unter die aktiven Verfahren fallen taktile Methoden, interferometrische Meßmethoden, Triangulationsverfahren, Streifenlichtprojektion oder Moiré-Verfahren. Einen guten Überblick über häufig eingesetzte Verfahren findet sich in [5, 3, 2, 4, 6].

Zu den punktsequentiellen Verfahren gehören beispielsweise Laufzeitmeßverfahren und punktsequentielle Triangulationsverfahren (siehe Abb. 2). Um ein flächenhaftes dreidimensionales Abbild vom Objekt zu erhalten, wird der Abtaststrahl entweder über rotierende Spiegel (siehe Abb. 3) auf das Objekt projiziert oder das Objekt über einen xy -Tisch verschoben. Im ersten Fall geben die Koordinaten in der Bildmatrix die Rotationswinkel der Spiegel wieder, wohingegen im zweiten Fall die Koordinaten in der Bildmatrix kartesischer Natur sind.

Im Falle der Streifenlichtprojektion (siehe Abb. 3) wird das Objekt flächenhaft parallel vermessen. Die Koordinaten in der Bildmatrix referenzieren dabei die Spalten und Zeilenindizes im Kamerabild, liegen also in projektiven Koordinaten vor. Je nach gewünschter Genauigkeit müssen dabei noch Linsenverzerrungen berücksichtigt werden. Gänge 3D-Sensoren arbeiten häufig mit folgenden Sensorgeometrien:

- Kartesische Koordinaten
- Perspektivische Koordinaten
- Zylinderkoordinaten
- Sphärische Koordinaten
- Kombination der Basiskoordinatensysteme in den beiden Dimensionen des Abtastrasters

Allerdings kann es sein, daß das spezifische Sensorkoordinatensystem je nach Aufbau des 3D-Sensors von den idealisierten Koordinatensystemen abweicht. Daneben sind Sensoren denkbar, deren Sensorgeometrie nicht durch die obigen Beispiele wiedergegeben werden kann. Daher ist es erforderlich, die Sensorgeometrie allgemein zu definieren, ohne dabei eine Spezialisierung für die idealisierten Koordinatensysteme durchzuführen.

3 Anforderungen an die Datenrepräsentation

Um die vom 3D-Sensor erfaßten Informationen zu repräsentieren, sollte das Datenformat für die Sensorgeometrie und die 3D-Meßdaten im Idealfall folgende Eigenschaften aufweisen:

- Vollständige Repräsentation der vom Scanner erfaßten Informationen
- Erhalt von Nachbarschaftsinformationen (und damit auch der *Sampling*-Dichte)
- Repräsentation der Scan-Richtung
- Repräsentation der Meßgenauigkeit
- Effiziente Datenrepräsentation
- Exakte Datenrepräsentation

Alle Anforderungen könnten trivial dadurch erfüllt werden, indem man für jeden Meßpunkt neben den eigentlichen 3D-Koordinaten $\vec{x}(u, v)$ auch die Scan-Richtung $\vec{c}(u, v)$ und die Genauigkeit $\vec{\sigma}(u, v)$ repräsentiert. Eine solche Repräsentation kann jedoch keinesfalls als effizient bezeichnet werden, mit der Folge das die in der Praxis häufig anzutreffenden großen Datenmengen selbst auf sehr großzügig ausgestatteten Rechnern viele praktische Probleme bereiten würden. Ein weiteres Problem ist die zumeist unbekannte Aufnahme-genauigkeit $\vec{\sigma}(u, v)$, deren Bestimmung oftmals gar nicht punktweise möglich ist.

4 Das (geheime) Sensorkoordinatensystem

Um die wesentliche Forderung nach Speicher-Effizienz zu erfüllen, sollten die Meßwerte als Matrix $\zeta(u, v)$ von Skalaren gespeichert werden können. Die eigentlichen 3D-Koordinaten müssen dann aus dem ζ -Bild berechnet werden. Dazu muß eine Funktion $T_p(u, v, \zeta)$ bekannt sein, die die gespeicherten Werte in Abhängigkeit von der Position im Bildraster und den Parametern p der inneren Orientierung in metrische Bildkoordinaten (IC-System) umrechnet (siehe auch Abb. 4).

$$\vec{x} = T_p(u, v, \zeta) \quad (2)$$

Die dazu benötigten Parameter p werden bei der Kalibrierung des 3D-Sensors bestimmt, wobei die Abbildung T_p eine Sensor- und Kalibriermodellspezifische Umrechnungsvorschrift kennzeichnet. Diese Repräsentationsform hat den Vorteil, daß die Scan-Richtung $\vec{c}(u, v)$ für jeden Meßpunkt explizit berechnet werden kann. In Rasterkoordinaten wird die Blickrichtung durch den Vektor

$$\vec{c}(u, v) = \frac{1}{d\zeta} (T_p(u, v, \zeta + d\zeta) - T_p(u, v, \zeta)) \quad (3)$$

bzw. im Grenzübergang gekennzeichnet durch

$$\vec{c}(u, v) = \frac{\partial}{\partial \zeta} T_p(u, v, \zeta) \quad (4)$$

Die Vor- und Nachteile der funktionalen Repräsentation des Sensorkoordinatensystems können wie folgt zusammengefaßt werden:

- Mittels T_p wird nicht nur eine effiziente Datenrepräsentation ermöglicht, sondern auch die Sensorgeometrie repräsentiert.

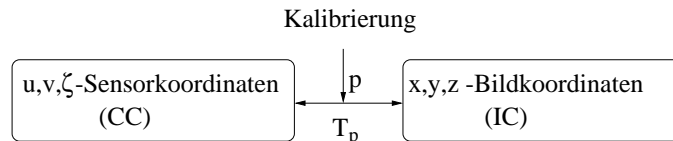


Abbildung 4: Bei der Kalibrierung wird die Abbildungsvorschrift vom Sensorkoordinatensystem zum metrischen Bildkoordinatensystem bestimmt.

- Die Umrechnungsvorschrift kann für einfache Koordinatensystem, wie z.B. kartesischen Koordinaten, sehr effizient implementiert werden.
- Sogar eine nichtgradlinige Strahlausbreitung wird unterstützt.
- Der Wertebereich von $\zeta(u, v)$ hängt i.a. von der Rasterposition (u, v) ab, so daß der gespeicherte ζ -Wert nur in speziellen Fällen als als Fixpunktgröße abgelegt werden kann.
- Da die Abbildungsfunktion vom speziellen Kalibriermodell des Sensorherstellers abhängig ist, ist die Funktion T_p i.a. geheim.
- Die Definition und Berücksichtigung der Funktion T_p auf verschiedenen Plattformen in den unterschiedlichen Applikationen ist in der Regel nicht praktikabel, da die Funktion T_p mit einer Verbesserung des Kalibriermodells z.B. aktualisiert werden müßte.
- Die ζ -Achse ist nicht orthogonal zur u, v -Achse, so daß Meßfehler in w und u, v korreliert sind.

Insbesondere die letzten Punkte weisen darauf hin, daß die Funktion T_p sehr sorgfältig definiert werden muß, um günstige Eigenschaften aufzuweisen. All dies zeigt, daß die funktionale Definition des Sensorkoordinatensystems nur in ausgewählten Einzelfällen (z.B. kartesische Koordinatenraster) sinnvoll ist. Für den allgemeinen Fall führen wir in den nächsten Abschnitten eine vollständig parametrisierte Darstellung des Sensorkoordinatensystems ein.

5 Das (öffentliche) orthogonale Sensorkoordinatensystem

Unter der Annahme einer gradlinigen Scan-Richtung führen wir neben dem u, v, ζ -Sensorkoordinatensystem das orthogonale u, v, w -Sensorkoordinatensystem (siehe Abb. 5) ein, welches bei gradliniger Strahlausbreitung im Meßvolumen immer existiert und die Sensorgeometrie vollständig parametrisiert. Darüberhinaus soll es insbesondere die Eigenschaft aufweisen, daß die w -Achse zumindest näherungsweise orthogonal zu den u - und v Achsen steht, um die Meßfehler in w von der Meßunsicherheit in u, v rechnerisch zu entkoppeln.

Die Kernidee dabei ist, daß jeder Meßpunkt als Skalar $w(u, v)$ gespeichert wird und die Sensorgeometrie für eine konstante Kalibrierung durch die Felder $\vec{x}_0(u, v)$ und $\vec{c}_0(u, v)$ parametrisiert wird, so daß die metrischen Bildkoordinaten durch folgende Umrechnungsvorschrift erhalten werden können.

$$\vec{x}(u, v) = \vec{x}_0(u, v) + w(u, v)\vec{c}_0(u, v) \quad (5)$$

Die Orthogonalitäts-Eigenschaft wird dann erfüllt, wenn die Fläche $\vec{x}_0(u, v)$ an jedem Meßpunkt orthogonal zur Scan-Richtung $\vec{c}_0(u, v)$ verläuft. Die Eigenschaften einer solchen Datenrepräsentation können wie folgt zusammengefaßt werden:

- Die Datenrepräsentation ist effizient.
- Die Umrechnung der w -Werte in 3D-Koordinaten \vec{x} kann mit nur 6 Operationen durchgeführt werden.
- Die Datenrepräsentation ist bis auf Diskretisierungsfehler in w und Ungenauigkeiten in $\vec{c}_0(u, v)$ exakt. Ungenauigkeiten in $\vec{x}_0(u, v)$ haben keinen Einfluß auf die Genauigkeit der Datenrepräsentation, d.h. die Fläche $\vec{x}(u, v)$ ist mehr oder weniger frei definierbar, sollte aber zumindest näherungsweise orthogonal zur Scan-Richtung verlaufen.

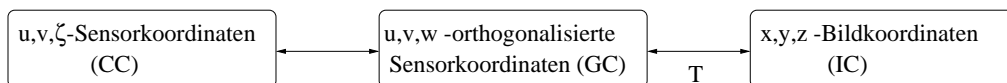


Abbildung 5: Das orthogonale Sensorkoordinatensystem weist für die Weiterverarbeitung der Daten günstige Eigenschaften auf und trennt den geheimen Teil der Kalibrierung von dem öffentlichen Teil der Sensorgeometrie.

- Die Meßfehler in u, v sind nicht mehr mit Fehlern in w korreliert mit vielen Vorteilen für die weitere Bearbeitung der Daten.
- Die Meßfehler σ_u, σ_v in u, v und Meßfehler σ_w in w kann näherungsweise als konstant angenommen werden.

5.1 Berechnung des orthogonalen Sensorkoordinatensystems

Zur Überführung von Meßdaten in die u, v, w -Sensorkoordinaten muß für jeden Punkt der Bildmatrix sowohl die Niveaufläche $\vec{x}_0(u, v)$ als auch die Scan-Richtung $\vec{c}(u, v)$ bekannt sein. Die Scan-Richtung kann zumeist elementar z.B. durch Differenzenbildung von zwei Meßwerten $\vec{x}_1(u, v) - \vec{x}_2(u, v)$ gewonnen werden, wohingegen die Niveaufläche a priori bekannt oder berechnet werden muß. Dies kann je nach Algorithmus besser in der expliziten Form, also in der Form $\vec{x}_0(u, v)$ erfolgen, oder aber auch in der impliziten Form, wobei die Niveaufläche durch $F(x, y, z) = 0$ definiert wird. Die implizite Form hat den Vorteil, daß sie unabhängig von der Parametrisierung in u, v ist. Nichtsdestotrotz muß auch die implizite Form später in die explizite Darstellungsform überführt werden.

5.1.1 Allgemeine Lösung

Ausgehend von der expliziten Repräsentationsform der Niveaufläche kann die Berechnung der Niveaufläche als finites Differenzenproblem betrachtet werden. Wenn der Scan-Strahl durch einen Aufpunkt $\vec{x}(u, v)$ und die vektorielle Richtung $\vec{c}(u, v)$ gegeben ist, kann die Niveaufläche am Rasterpunkt (u, v) definiert werden durch:

$$\vec{x}_0(u, v) = \vec{x}(u, v) + \lambda(u, v)\vec{c}(u, v) \quad (6)$$

Bei der Bestimmung der Niveaufläche müssen jetzt nunmehr noch die Parameter $\lambda(u, v)$ bestimmt werden, so daß die Scan-Richtung immer orthogonal zur Niveaufläche verläuft. Dies führt im diskreten Fall zu folgenden Bedingungen:

$$(\vec{x}_0(u+1, v) - \vec{x}_0(u, v)) \cdot (\vec{c}(u+1, v) + \vec{c}(u, v)) = 0 \quad (7)$$

$$(\vec{x}_0(u, v+1) - \vec{x}_0(u, v)) \cdot (\vec{c}(u, v+1) + \vec{c}(u, v)) = 0 \quad (8)$$

Aus diesen Bedingungen läßt sich ein Gleichungssystem mit $n_u * n_v - 1$ Unbekannten $\lambda(u, v)$ ableiten, welches sich unter der Annahme $\lambda(0, 0) = 0$ lösen läßt. Gibt man sich mit einer sehr guten Näherungslösung zufrieden, wächst der Aufwand zur Lösung des Gleichungssystems nur linear mit der Anzahl der Unbekannten.

Obwohl dieses Vorgehen allgemeingültig ist, führt es dann zu numerischen und algorithmischen Problemen, wenn die Scan-Richtung $\vec{c}(u, v)$ nicht an jedem Rasterpunkt (u, v) ermittelt werden kann. Daher wäre die Kenntnis von $\vec{c}(u, v)$ an jedem Rasterpunkt Voraussetzung für den Einsatz dieses Verfahrens.

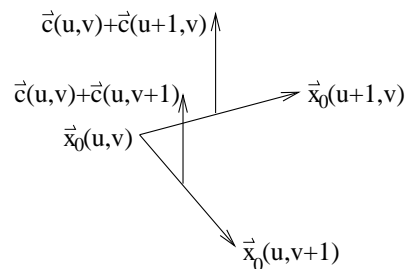


Abbildung 6: Die Niveaufläche ist dadurch definiert, daß die Scan-Richtung \vec{c} orthogonal zu den Tangentenvektoren verläuft.

5.1.2 Spezialisierte Lösung

Für die häufig in der Praxis eingesetzten Sensorkoordinatensysteme, wie kartesische Koordinaten, Zylinderkoordinaten, Kugelkoordinaten, perspektivische Koordinaten und Kombinationen davon kann die Niveaufläche sehr einfach berechnet werden. Die Niveaufläche wird implizit über $F(\vec{x})$ definiert und bis auf eine Konstante durch eine Fläche 2-ter Ordnung mit symmetrischer Koeffizientenmatrix \mathcal{A} approximiert.

$$F(\vec{x}) = \vec{x}^T \mathcal{A} \vec{x} + \vec{b} \cdot \vec{x} + const \quad (9)$$

Zur Approximation wird die folgende Bedingung herangezogen:

$$\frac{\partial}{\partial \vec{x}} F(\vec{x}) \times \vec{c}(u, v) = 0 \quad (10)$$

Die Niveaufläche wird in die explizite Form überführt werden, indem jeweils der Schnittpunkt des bekannten Scan-Strahls mit der Niveaufläche berechnet wird. Dies kann bei diesem Ansatz direkt analytisch erfolgen. Die Auswahl des auf der Niveaufläche liegenden Schnittpunkts aus den zwei potentiell ermittelten Schnittpunkten kann unter Ausnutzung der Nebenbedingung erfolgen, daß der Verbindungslinie zwischen dem Ausgangspunkt und dem gefundenen Schnittpunkt vollständig im erlaubten Meßbereich liegt. Mit Hilfe dieser Nebenbedingung muß der Schnittpunkt eindeutig gefunden werden können, ansonsten wäre die approximierte Niveaufläche nicht für die tatsächlich vorliegende Sensorgeometrie geeignet.

Der quadratische Ansatz ist für kartesische, polare, sphärische, zylindrische und perspektivische Koordinatensystem exakt. Daher wird für gebräuchliche Meßmethoden zumindest eine gute Näherung für die Niveaufläche geliefert. Die Näherung ist — wie schon erwähnt — völlig ausreichend, da die Niveaufläche gemäß unserer Spezifikation nur näherungsweise orthogonal zu den Scan-Richtungen laufen muß.

5.1.3 Sensorspezifische Lösung

Für a priori bekannte Sensorgeometrien kann die Niveaufläche oftmals sehr viel einfacher definiert werden, und zwar

- als Ebene für kartesische Koordinaten
- als Zylinder für Zylinderkoordinaten
- als Kugel für Kugelkoordinaten
- als Kugel oder in guter Näherung auch als Ebene für perspektivische Koordinaten (siehe Abb. 7)

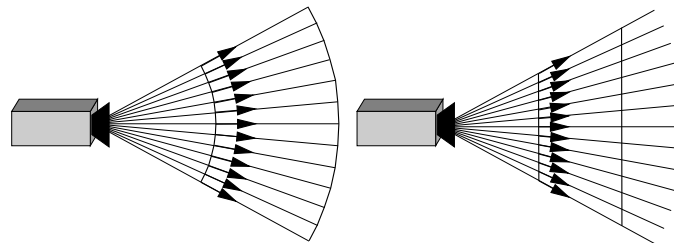


Abbildung 7: Für apriori bekannte Sensorgeometrien ist die Definition der Niveaufläche oftmals sehr einfach.

5.2 Datenkonvertierung

Für die Konvertierung der 3D-Sensordaten in das orthogonale Sensorkoordinatensystem ist zusammengefaßt das folgende Verfahren zweckmäßig. Für jeden Rasterpunkt (u, v) wird Speicher für die 3D-Koordinaten der Niveaufläche $\vec{x}_0(u, v)$ und die Scan-Richtung $\vec{c}_0(u, v)$ reserviert, so daß eine Matrix mit

jeweils 6 Werten pro Pixel benötigt wird. Für jeden Rasterpunkt wird die Scan-Richtung $\vec{c}(u, v)$ z.B. durch die Ableitung der geheimen Funktion $\frac{\partial}{\partial \zeta} T_p(u, v, \zeta)$ oder einfach durch Differenzenbildung von zwei Meßwerten, die demselben Bildpunkt zugeordnet werden, gebildet. Danach wird die Niveauläche in expliziter Form mit einem der oben beschriebenen Verfahren unter Einbeziehung der Scan-Richtung approximiert. Die Niveauläche wird dabei so definiert, daß mit $(w_{max} - w_{min})\vec{c}_0$ der volle Scan-Bereich ausgeschöpft wird. Für undefinierte Datenpunkte wird $w = 0$ reserviert, so daß mit $w_{max} = 2^{16} - 1$ und $w_{min} = 1$ das Bild in voller 16-Bit Genauigkeit repräsentiert werden kann. Mittels der bereits oben definierten Abbildung

$$\vec{x} = T_{GC \rightarrow IC}(u, v, w) = \vec{x}_0(u, v) + w * \vec{c}_0(u, v) \quad (11)$$

wird die Umrechnung der orthogonalisierten Sensorkoordinaten (GC) in metrische Bildkoordinaten (IC) definiert. Die eigentlich Umrechnung der Meßwerte $\vec{x}(u, v)$ erfolgt dann durch Invertierung der Abbildung $T_{GC \rightarrow IC}$ und Abspeicherung des Ergebnisses an der respektiven Stelle (u, v) der Bildmatrix.

$$(u, v, w) = T_{GC \rightarrow IC}^{-1}(\vec{x}) \quad (12)$$

5.2.1 Die Inverse Transformation

Die Inverse Transformation $T_{GC \rightarrow IC}^{-1}(\vec{x})$ kann sehr effizient mit einer verallgemeinerten Newton-Raphson Methode durchgeführt werden. Die Newton-Raphson Methode zeichnet sich dadurch aus, daß sie schnell und sicher konvergiert, sofern der benötigte Startpunkt den inversen Datenpunkt hinreichend gut approximiert. Daher arbeiten wir mit einem dreidimensionalen Startpunktfeld, bei dem das Meßvolumen in eine Menge von Raumwürfeln unterteilt wird und jedem Raumwürfel eine Menge von potentiellen Startpunkten zugewiesen werden. Mit diesem Ansatz konvergiert das modifizierte Newton-Raphson Verfahren bei uns im Mittel in 3 Iterationen.

6 Datenformat

Obwohl die Speicherung der Meßwerte und der Sensorgeometrie unabhängig von der logischen Repräsentation der Daten ist, unterstützen wir bei der Speicherung insbesondere das TIFF-Datenformat[1]. Die Vorteile des TIFF-Datenformats sind offensichtlich: Die TIFF-Bibliothek ist *public domain* auf allen gängigen Hardware-Plattformen verfügbar. Das TIFF-Datenformat erlaubt neben der Speicherung von Grauwert- und Farbbildern auch die Speicherung von 16 Bit Integer Werten und in TIFF 6.0 auch die Speicherung von *Float*-Vektorfeldern. Eine etwaige Datenkompression ist bereits Teil der TIFF-Bibliothek. Daher liegt mit TIFF bereits ein ausgereiftes und effizientes Datenaustauschformat vor. Daneben kann im TIFF-Tag Feld *IMAGEDESCRIPTION* des TIFF-Bildes noch beliebig viel an Zusatzinformation untergebracht werden, ohne eigene Tag-Felder spezifizieren zu müssen.

Für die Speicherung der 3D-Meßwerte ist damit eine Datei für die Speicherung der Sensorgeometrie notwendig und jeweils eine Datei für jede 3D-Aufnahme. Da sich die Sensorgeometrie erst bei einer erneuten Kalibrierung ändert, können viele 3D-Aufnahmen einer Sensorgeometriedatei zugeordnet werden.

6.1 Sensorgeometrie

Die Sensorgeometrie wird als 6-Band Float Bild gespeichert, wobei an jedem Rasterpunkt die 6 Komponenten von $((x)_0(u, v), \vec{c}_0(u, v))$ gespeichert werden. Im TIFF-tag Feld *IMAGEDESCRIPTION* wird die folgende Information untergebracht:

- Eine eindeutige Kennung (*magic string*)
- Das Entstehungsdatum der Daten
- Definitionsbereich w_{min}, w_{max}
- Meßvolumen $x_{min}, y_{min}, z_{min}, x_{max}, y_{max}, z_{max}$

- Reduktionsfaktoren s_x, s_y für Datenkompression
- Drehsinn $\eta = \pm 1$ des Koordinatensystems.

Die Definition des Drehsinns (linksdrehend $\eta = -1$, rechtsdrehend $\eta = +1$) ist unbedingt erforderlich, da in der Praxis verschiedene orientierte Koordinatensystem im Einsatz sind. Der Drehsinn kann nicht aus den Daten berechnet werden und ist folglich reine Definitionssache, obwohl der Drehsinn natürlich richtig herum definiert sein muß, um seine Meßdaten korrekt zu beschreiben. Die Reduktionsfaktoren ermöglichen typischerweise eine deutliche Datenreduktion und werden im folgenden Abschnitt erläutert.

6.1.1 Datenreduktion durch Interpolation

Da die Scan-Richtung typischerweise nur wenig über dem Bildraster variiert, kann die Scan-Richtung an jedem Rasterpunkt aus einem grober aufgelösten Raster mit nur geringfügigen Interpolationsfehlern zurückgewonnen werden. Bei der Festlegung der Niveaufläche besteht ohnehin einiges an Variationsspielraum, so daß auch die Niveaufläche ohne Genauigkeitsverlust durch Interpolation in einem grober aufgelösten Raster definiert werden kann. Diese Eigenschaften werden nun ausgenutzt, um die Menge der zu speichernden Daten deutlich zu reduzieren, indem das Ausgangsraster um die Reduktionsfaktoren s_x, s_y verkleinert wird.

Für die Interpolation kommt die einfache bilineare Interpolation oder die komplexere bikubische Interpolation in Frage. Da die bikubische Interpolation mit deutlich weniger Rechenaufwand verbunden ist und durch die Definition der Abbildungsvorschrift $T_{GC \rightarrow IC}$ sowohl die transformierten 3D-Koordinaten \vec{x} als auch die Ableitung von $\frac{\partial}{\partial w} T(u, v, w)$ jeweils C^0 stetig sind, werden alle in der Praxis notwendigen Eigenschaften bereits durch die bilineare Interpolation erfüllt. Daher haben wir die Reduktion zusammen mit der bilinearen Interpolation definiert.

Sofern die Konvertierung der Meßdaten in das orthogonale Sensorkoordinatensystem bereits mit der in geringerer Auflösung gespeicherten Sensorgeometrie erfolgt, ergibt sich der Fehler in der Datenrepräsentation rein aus der Scan-Richtung, also konkret aus $(w_{max} - w_{min})\vec{c}(u, v)$. Hierbei können die induzierten Fehler in der vektoriellen Scan-Richtung bis zur Größenordnung von 0.1 Pixel im Kameraraster ohne merklichen Genauigkeitsverlust durchaus toleriert werden. Dabei zeichnet sich die bilineare Interpolation durch die Eigenschaft aus, daß die Genauigkeit quadratisch mit der Sampling-Distanz wächst, also linear mit dem Speicheraufwand.

Tests an ausgewählten Koordinatensystemen haben gezeigt, daß damit eine erhebliche Datenreduktion ohne merklichen Genauigkeitsverlust möglich ist. Wesentlich ist nur, daß die Reduktionsrate vor der Konvertierung in das orthogonale Sensorkoordinatensystem festgelegt werden muß und später nicht mehr geändert werden darf.

6.2 Meßdaten

Neben der Sensorgeometrie müssen natürlich auch die eigentlichen Meßdaten abgelegt werden. Auch hier haben wir uns für das TIFF-Datenformat aus den oben aufgeführten Gründen entschieden. Im TIFF-Header werden unter dem Tag-Feld IMAGEDESCRIPTION folgende Informationen gespeichert:

- Typ der Kalibrierung (GENERIC)
- Parameter der Sensorgeometrie. Für das in diesem Artikel beschriebene Niveauflächenmodell (GENERIC) steht hier ein Verweis auf eine TIFF-Datei, die die Sensorgeometrie enthält. Für funktional definierte Kalibriermodelle können die Kalibrierparameter hier auch direkt eingetragen werden, oder ebenfalls ein Verweis auf eine Datei stehen.
- Sofern bekannt, können hier auch optional die Parameter der externen Orientierung eingetragen werden. Die externe Orientierung (Rotation und Translation) wird als homogen 3×4 -Matrix abgelegt. Sie beschreibt die Abbildung vom x, y, z -Bildkoordinatensystem in das allen Bildern gemeinsame Weltkoordinatensystem.

Die eigentlichen Bilddaten werden als 2-Byte oder 4-Byte (wird derzeit noch nicht unterstützt) Datenfelder abgelegt, wobei der Wert $w = 0$ für undefinierte Meßwerte reserviert ist.

6.3 Beispiele

Für eine 512×512 -Sensormatrix kann die Sensorgeometrie in $512 * 512 * 6 * 4\text{Byte} \approx 6\text{MB}$ parametrisiert abgelegt werden. Alleine ein Reduktionsfaktor von 4 reduziert dieses Datenvolumen auf nur 384 KB. Typischerweise sind jedoch noch weit aus stärkere Reduktionsfaktoren ohne merklichen Genauigkeitsverlust zulässig. Die eigentlichen Bilddaten können mit nur 2 Byte pro Rasterpunkt abgelegt werden, so daß in diesem Beispiel ein Bild 512 KB belegt. Je nach Reduktionsfaktor ist dieses Datenformat ab 1-3 Bildern zur selben Sensorgeometrie wesentlich effizienter als die explizite Speicherung der 3D-Koordinaten, obwohl es neben den 3D-Koordinaten auch die Scan-Richtung parametrisiert für beliebige Sensorgeometrien erfaßt.

7 Zusammenfassung/Ausblick

Das Sensorkoordinatensystem enthält eine Reihe von wichtigen Informationen, die für die Auswertprogramme von Bedeutung sein können. Durch die Konvertierung der Meßdaten in ein orthogonalisiertes Sensorkoordinatensystem liegt eine exakte und adäquate Beschreibung der 3D-Meßwerte vor, die durch die Orthogonalität der w -Achse zur u - und v -Achse eine Reihe von günstige Eigenschaften für die Weiterverarbeitung aufweist. Die explizite Speicherung der Sensorgeometrie erlaubt dem Auswertprogramm insbesondere einen Zugriff auf die Scan-Richtung und damit auf eine wesentliche und bislang oftmals unzugängliche Information der 3D-Daten. Durch die Verwendung des weit verbreiteten TIFF-Dateiformats werden die Daten standardisiert abgelegt.

Als Unterstützung zur Anbindung eines breiten Spektrums von 3D-Sensoren an das SCULPTOR-System werden wir Source-Code Beispiele zur Speicherung der Sensorgeometrie und der Bilddaten in Kürze auf unseren WWW-Seiten veröffentlichen:

<http://www.igd.fhg.de/www/igd-a7/Projects/model/index.htm>

Literatur

- [1] Aldus Corporation, Seattle. *Tag Image File Format Specification, Revision 6.0*, 1992.
- [2] P. J. Besl. Active Optical Range Imaging Sensors. In J.L.C. Sanz, editor, *Advances in Machine Vision*, pages 1–63. Springer-Verlag, New York, 1989.
- [3] P.J. Besl. *Surfaces in Range Image Understanding*. Springer-Verlag, Berlin, 1988.
- [4] G.R. Hofmann (Hrsg.), C. Blum and A. Hildebrand and P.J. Neugebauer and L. Neumann and U. Schneider and R. Strack. *Imaging: Bildverarbeitung und Bildkommunikation*. Springer-Verlag, 1993.
- [5] R. A. Jarvis. A perspective on range finding techniques for computer vision. *IEEE Trans. Pattern Anal. and Mach. Intell.*, PAMI-5(2):122–139, 1983.
- [6] R. A. Jarvis. *Range Sensing for Computer Vision*, pages 17–56. Elsevier Science Publishers, 1993.
- [7] P. J. Neugebauer. Reconstruction of Real-World Objects via Simultaneous Registration and Robust Combination of Multiple Range Images. *International Journal of Shape Modeling*, 3(1&2):71–90, 1997.
- [8] P.J. Neugebauer and K. Klein. Adaptive Triangulierung komplexer Oberflächen. In *Workshop 3D-Bildverarbeitung*, TAE Esslingen, 20./21. Januar 1998.