GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Programmierung eines tomographischen Algorithmus zur Rekonstruktion von 3D Dichtefeldern aus BOS[1] Messergebnissen

am Institut für Aerodynamik und Strömungstechnik
von

Christiane Maria Struck

Betreuer:

Prof. Dr. Robert Schaback
Institut für Numerische und Angewandte Mathematik
Georg-August-Universität Göttingen

Dipl.-Ing. Erik Goldhahn
Dipl.-Ing. Tania Kirmse
Institut für Aerodynamik und Strömungstechnik
Deutsches Zentrum für Luft- und Raumfahrt

Tag der Anmeldung: 10. Februar 2005
Tag der Abgabe: 10. Mai 2005

[1] BOS, Background Oriented Schlieren

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Göttingen, den 10. Mai 2005

Inhaltsverzeichnis

Symbolverzeichnis

$\%$	Modulooperator
$*$	Faltungsoperator
Δs	Abstand der Daten in einer Projektion
Δx	s-Komponente einer Verschiebung
Δy	z-Komponente einer Verschiebung
$\lceil s \rceil$	Aufgerundeter Wert von s
$\lfloor s \rfloor$	Abgerundeter Wert von s
$\mathscr{B}$	Backprojektion (Rueckprojektion)
$\mathscr{F}$	Fouriertransformation
$\mathscr{R}$	Radontransformation
ρ	Dichte
θ	Projektionswinkel
ε	Ablenkungswinkel
ϱ	Dichte
c	Lichtgeschwindigkeit im Vakuum
$c(x)$	Faltungsfunktion im Ortsraum
$c_f(f)$	Filterfunktion im Frequenzraum
d	Abstand zwischen Dichtefeldmitte und Hintergrundbild
f	Frequenz
f_c	Grenzfrequenz (Cutoff-Frequenz)
$g(x, y)$	Punkt im rekonstruierten Dichtefeld
$G_\theta(s, t)$	Eine Schicht des Brechungsindexfeldes

K	Gladstone-Dale Konstante
M	Anzahl der Projektionen
N	Anzahl der Daten in einer Schicht
n	Brechungsindex
$p(s, \theta)$	Projektion
s	Koordinate in einer Projektion
t	Lichtstrahl
v	Lichtgeschwindigkeit im Medium
$w(f)$	Fensterfunktion

1. Einleitung

Wenn transparente Medien Dichteunterschiede aufweisen, so erkennt man Schlieren, die sich auf die unterschiedliche Ablenkung der Lichtstrahlen zurückführen lassen. Dieses Phänomen beobachtet man z.B. über Feuer oder an heißen Sommertagen über dem Asphalt und man kann es nutzen, um die Dichte von Gasen zu bestimmen.

Interessant ist das beispielsweise für die Luftfahrt. Wirbelschleppen hinter fliegenden Flugzeugen können andere Flugzeuge gefährden. Insbesondere auf Flughäfen spielt dies eine wichtige Rolle, denn dort sollen Luftfahrzeuge in möglichst kurzen Abständen starten und landen. Daher wird schon während der Konstruktion versucht, diese Luftverwirbelungen bei großen Flugzeugen möglichst gering zu halten. Um an Techniken zu deren Vermeidung arbeiten zu können, ist es nötig, diese Strukturen auch messtechnisch zu erfassen.

Bei der Computertomographie werden aus verschiedenen Winkeleinstellungen Aufnahmen (Projektionen) vom zu untersuchenden Dichtefeld (z.B. Luft über einer Kerze) gemacht. Dadurch sind dann von einem dreidimensionalen Dichtefeld mehrere zweidimensionale Aufnahmen vorhanden. Aus diesen Projektionen wird die dreidimensionale Darstellung der Verteilung der Dichte rekonstruiert.

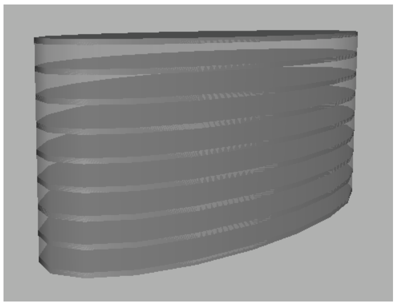


Abbildung 1.1: Schichtdarstellung eines Dichtefeldes

Bei dem in dieser Arbeit vorgestellten Rekonstruktionsverfahren, der gefilterten Rückprojektion, betrachtet man ein Dichtefeld als mehrere aufeinandergestapelte zweidimensionale Schichten. Die Rekonstruktion des dreidimensionalen Feldes reduziert sich also auf die Rekonstruktion von einem Satz zweidimensionaler Schichten. Um diese einzeln zu analysieren, werden Information eines bestimmten Streifens, welcher der späteren Schicht entspricht, aus allen Projektionen ausgewählt und dann die Schicht berechnet. Radon[1] stellte 1917 fest, dass es möglich ist, die Berechnung einer Schicht auszuführen, wenn genügend Aufnahmen aus verschiedenen Projektionswinkeln vorhanden sind.[15]

[1] Johann Radon (1887 - 1956), Wiener Mathematiker [20]

1.1 Zielsetzung der Arbeit

Ziel der vorliegenden Arbeit ist es, den Algorithmus der gefilterten Rückprojektion zu implementieren. Die Ausgangsdaten für die tomographische Rekonstruktion liefert das am DLR entwickelte optische Hintergrund-Schlieren-Verfahren (BOS, engl.: Background Oriented Schlieren). Aus der Ablenkung von Lichtstrahlen in einem Dichtegradientenfeld werden so die Dichtedaten tomographisch rekonstruiert. Die gewonnenen Daten können dann mit TecPlot[2] graphisch dargestellt werden.

Derzeitig liegt dem DLR[3] der Algorithmus in Matlab[4] vor. Die Laufzeit dieses Programms unter Matlab ist sehr lang und auch die erforderliche Flexibilität der Dateneingabe ist nicht gegeben. Aus diesem Grund soll der Algorithmus in eine Programmiersprache implementiert werden, welche zum einen das Laufzeitproblem löst und zum anderen die erforderliche Flexibilität in Hinsicht sowohl auf die Dateneingabe, als auch auf die Programmerweiterbarkeit liefert.

1.2 Gliederung der Arbeit

Bei dem Hintergrund-Schlieren-Messverfahren wird, wie oben beschrieben, die Lichtbrechung genutzt, um ein Vektorfeld zu berechnen, welches Informationen über die Dichteverteilung enthält. In Kapitel 2 wird der Zusammenhang zwischen Dichte und Lichtbrechung in einem Medium erklärt. Des weiteren wird das Verfahren und der Aufbau eines Versuches zur Dichtebestimmung Thema dieses Kapitels, wobei jedoch nur auf die Grundlagen eingegangen wird. Eine detailliertere Beschreibung vom Messverfahren Background Oriented Schlieren und seinen theoretischen Grundlagen kann man in *Raffel*[4] nachlesen.

Zur Rekonstruktion von Objekten aus Projektionen sind mehrere Verfahren entwickelt worden. Das Programm, welches im Rahmen dieser Arbeit geschrieben wird, benutzt die gefilterte Rückprojektion zum Rekonstruieren von dreidimensionalen Dichtefeldern. Rekonstruktionsverfahren und insbesondere der Algorithmus der gefilterten Rückprojektion sind Inhalt des dritten Kapitels.

Im Kapitel 4 wird genauer auf die Programmierung eingegangen und ein Überblick über die Programmstruktur und die Implementierung gegeben. Außerdem werden die Ergebnisse und Folgen einer Laufzeitanalyse dargestellt und erläutert. Mit der praktischen Anwendung des Programms und den Ergebnissen beschäftigt sich Kapitel 5.

Den Abschluss der vorliegenden Arbeit bildet das Kapitel 6, welches eine Zusammenfassung und einen Ausblick auf weiterführende Arbeiten liefert.

---

[2] eine Datenvisualisierungssoftware

[3] Deutsches Zentrum für Luft- und Raumfahrt

[4] eine mathematische Software der Firma *The MathWorks* zur Lösung diverser mathematischer Probleme und zur grafischen Darstellung der Ergebnisse [29]

2. BOS Verfahren: Idee und Realisierung

Die Hintergrund-Schlieren-Methode (BOS, Background Oriented Schlieren) ist ein optisches Messverfahren zur Bestimmung der Dichte von Gasen. Die Ablenkung eines Lichtstrahles aufgrund von Dichtegradienten wird ausgenutzt, um das Dichtefeld zu bestimmen.

2.1 Dichte und Lichtablenkung eines Mediums

Die Brechung eines Lichtstrahls beim Durchgang durch die Grenzfläche zweier Medien bezeichnet die Richtungsänderung seiner Ausbreitung. Diese Richtungsänderung ist darauf zurückzuführen, dass sich Licht in Medien mit verschiedener Dichte unterschiedlich schnell ausbreitet.

Einem einzelnen Medium kann der Quotient aus Lichtgeschwindigkeit im Vakuum c und Lichtgeschwindigkeit im Medium v als Brechungsindex zugeordnet werden.

$$n = \frac{c}{v} \tag{2.1}$$

Die Gladstone-Dale Gleichung (2.2) (ausführliche Herleitung bei *Merzkirch* [2]) beschreibt, wie der Brechungsindex und die Dichte zusammenhängen. Sie gilt nur für Gase, deren Brechungsindex annähernd eins beträgt. [2] [5]

$$n - 1 = \left(\frac{\varrho\, N_A\, e^2}{2\,\pi\, m_e\, m} \right) \sum \left[\frac{f_i}{(\nu_i^2 - \nu^2)} \right] \tag{2.2}$$

N_A – Avogadro Zahl
e – Ladung eines Elektrons
m_e – Elektronenmasse
m – Molekulargewicht des Fluides
f_i – Oszillatorstärken, Wichtungsfaktor mit $\sum_i f_i = 1$
ν_i – Resonanzfrequenzen des Gases
ν – Frequenz des Lichtes

Durch die Vereinfachung:

$$K = \left(\frac{N_A\, e^2}{2\,\pi\, m_e\, m}\right) \sum \left[\frac{f_i}{(\nu_i^2 - \nu^2)}\right] \tag{2.3}$$

kann Gleichung (2.2) geschrieben werden zu:

$$n - 1 = K\,\varrho \tag{2.4}$$

n – Brechungsindex

K – Gladstone-Dale Konstante (Sie ist von der Wellenlänge des verwendeten Lichtes und von der Gasart abhängig.)

ϱ – Dichte

Durch diese Beziehung ist eine Umrechnung zwischen Brechungsindexfeld und Dichtefeld direkt möglich.

Da die Werte des Brechungsindex zur Darstellung der Dichteverteilung ausreichend sind, werden das Programm und der Algorithmus, welche im Rahmen dieser Arbeit beschrieben werden, nur den Brechungsindex berechnen.

2.2 Das Messsystem BOS

Zunächst wird das Prinzip des BOS Messsystems anhand eines einfachen Beispiels erklärt. Im Anschluss wird dann die Erweiterung für die Rekonstruktion eines dreidimensionalen Dichtefeldes erläutert.

2.2.1 Der Aufbau

Für dieses Verfahren benötigt man ein Hintergrundbild mit zufälligem Punktemuster, eine stationäre Kamera zur Bildaufnahme und eine ausreichende Beleuchtung des Hintergrundes.
Die Versuchsanordnung ist wie im Bild 2.1 gezeigt so, dass das Dichtefeld zwischen Kamera und Hintergrundbild platziert ist.

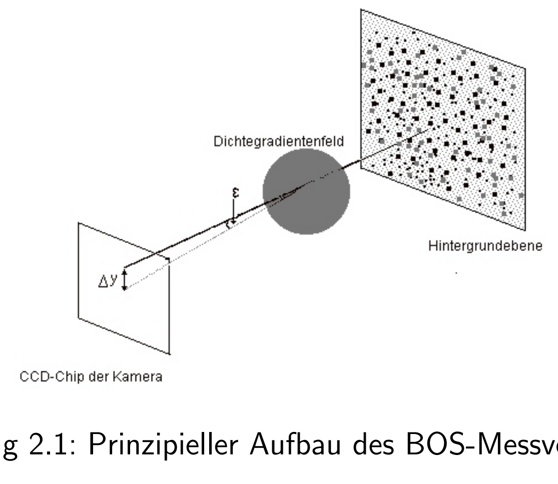


Abbildung 2.1: Prinzipieller Aufbau des BOS-Messverfahrens

2.2.2 Durchführung

Die auf den Hintergrund gerichtete Kamera nimmt lediglich zwei Bilder auf, eines davon als Referenzbild ohne und eines als Messbild mit Dichtefeld. Möchte man mehrere Zeitpunkte analysieren, wird zu den entsprechenden Zeiten ein weiteres Messbild aufgenommen.

2.2.3 Auswertung

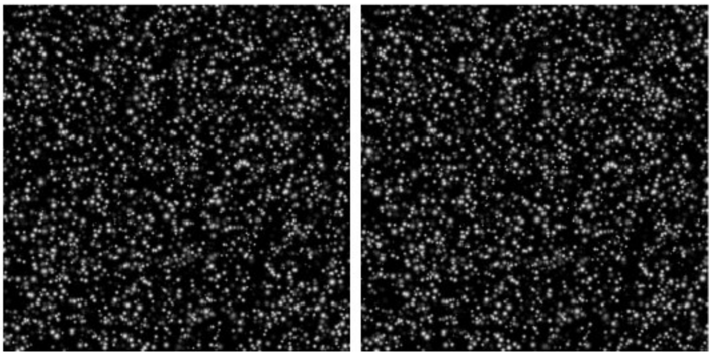


Abbildung 2.2: Links ist das Referenzbild ohne Dichtefeld, rechts ist das Messbild mit Dichtefeld [8]

Das menschliche Auge kann nur schwer erkennen, dass das Punktemuster im zweiten Bild (mit Dichtefeld) im Gegensatz zum ersten Bild (ohne Dichtefeld) unterschiedlich stark verschoben ist.

Mit einer Auswertungssoftware kann man diese Differenzen bestimmen. Anhand von Kreuzkorrelationen ermittelt das Programm die mittlere Verschiebung der Punkte innerhalb festgelegter Bereiche. Jeweils zugehörige Bereiche werden miteinander verglichen. Aus der Position des Maximums der zweidimensionalen Kreuzkorrelation erhält man für jeden Bereich die Verschiebung $\Delta w = (\Delta x, \Delta y)$ (Abb. 2.3). Die Richtung und Größe aller Verschiebungen in den oben gezeigten Bildern ist mit Hilfe von Vektoren in Abbildung 2.4 dargestellt.

Die Größe der Bereiche hat Einfluss auf die Qualität des Ergebnisses. Zu große Bereiche verringern die Detailschärfe. Wurden hingegen zu kleine Bereiche gewählt, so dass zu viele Punkte den ursprünglichen Quadranten verlassen, wird das Ergebnis ungenau. Um es zusätzlich zu verbessern, werden die Quadranten zu 50% überlappt.

Wichtig für die Qualität der Auswertung ist außerdem das zufällige Punktemuster. Bei regelmäßigen Strukturen kann die Kreuzkorrelation kein zuverlässiges Ergebnis liefern. [5]

Auf diese Weise erhält man Verschiebungsvektoren, welche die jeweils mittlere Verschiebung der Punkte eines Bereiches anzeigen. Diese Verschiebungsvektoren sind als Matrix im Tecplot-Format gespeichert $(s - Koord, z - Koord, \Delta x, \Delta y)$

$$\begin{aligned} s - Koord &\quad - \quad \text{Pixelposition in s-Richtung} \\ z - Koord &\quad - \quad \text{Pixelposition in z-Richtung} \\ \Delta x &\quad - \quad s - \text{Komponente der Verschiebung} \\ \Delta y &\quad - \quad z - \text{Komponente der Verschiebung} \end{aligned}$$

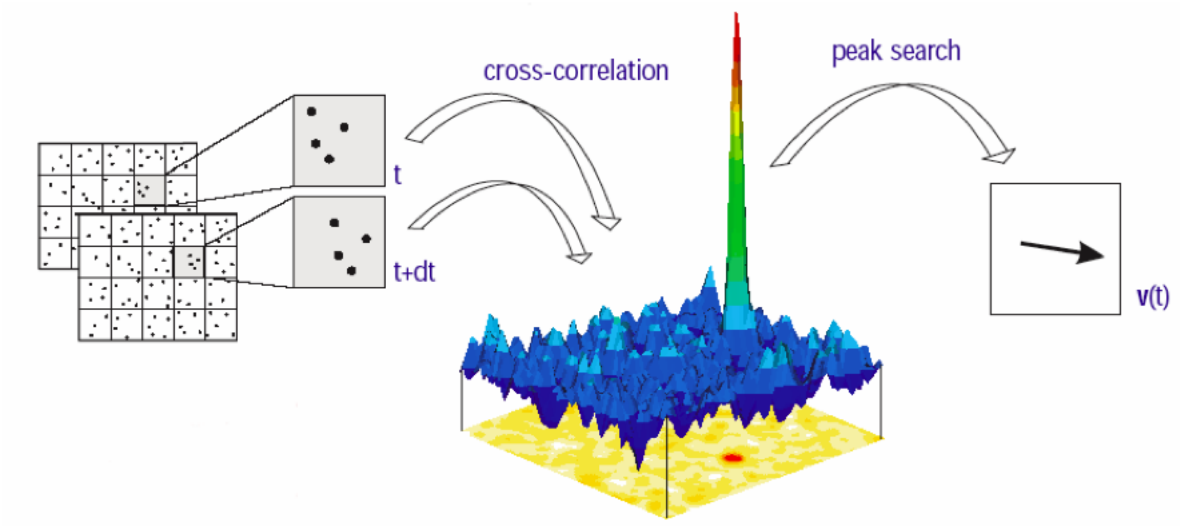


Abbildung 2.3: Prinzipielle Vorgehensweise bei der Auswertung [6]

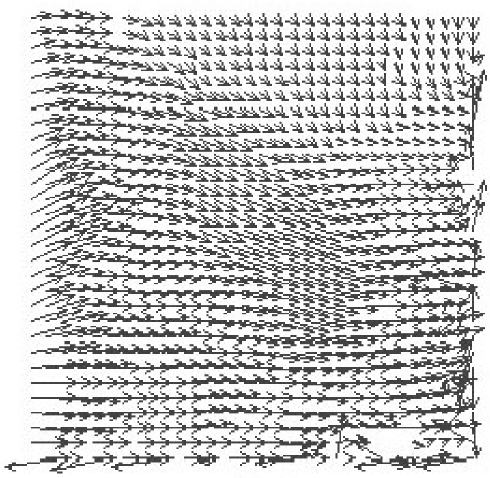


Abbildung 2.4: Das aus den oberen Aufnahmen errechnete Verschiebungsvektorfeld [8]

Man unterscheidet zwischen paralleler und nichtparalleler Projektionsgeometrie (Abb. 2.5). In der Parallelgeometrie werden alle Werte einer Projektion $p(s)$ mit dem gleichen Winkel θ zusammengefasst. In der nichtparallelen Projektionsgeometrie muss berücksichtigt werden, dass der Strahlengang einer Kamera fächerförmig verläuft (Abb. 2.6). Die Werte an den Rändern dieser Aufnahmen werden somit von unterschiedlichen Blickwinkeln aufgenommen. Diese Unterschiede können vernachlässigt werden, wenn zum einen meist nur der zentrale Bereich der Projektionen und somit nicht die komplette Breite verwendet wird und zum anderen die Objektivöffnung relativ klein ist. Aus diesem Grund wird im weiteren davon ausgegangen, dass die parallele Projektionsgeometrie vorliegt.

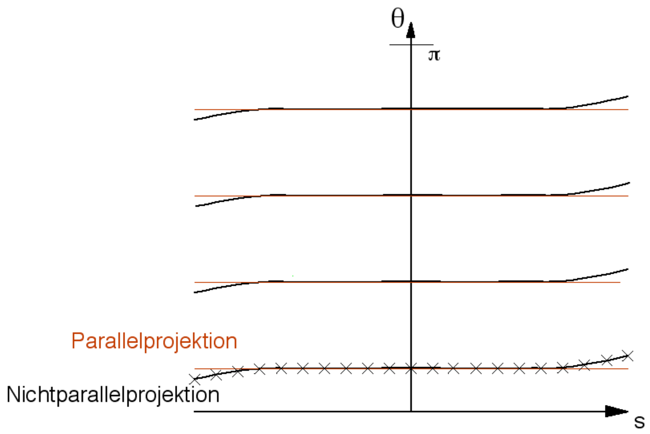


Abbildung 2.5: Projektionsgeometrien

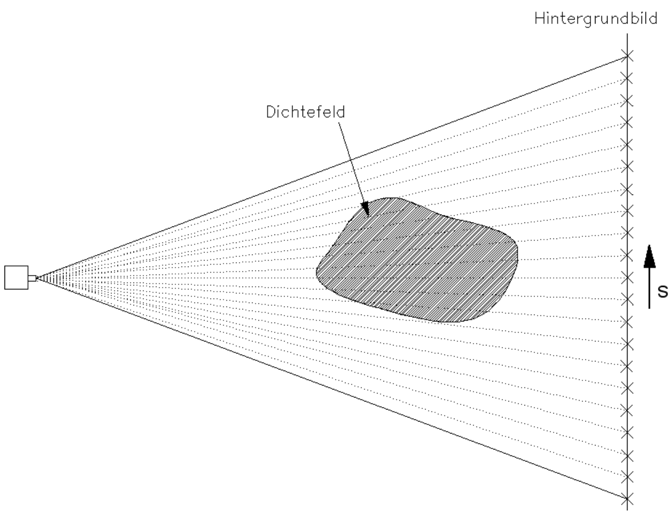


Abbildung 2.6: Fächergeometrie

2.3 Das BOS 3D-Verfahren

Ein zu untersuchendes Dichtefeld besteht aus unbekannt vielen Dichteänderungen. Die Daten der Verschiebungsvektormatrix (vgl. Abb. (2.4)) sind als Summe der kontinuierlichen Ablenkungen entlang des Lichtstrahles zu verstehen. Diskretisiert bedeutet dies, dass der Lichtstrahl viele kleine Brechungen erfährt.

Nimmt man einige Vereinfachungen an (wie beispielsweise ein axialsymmetrisches Untersuchungsobjekt), so kann man das Dichtefeld mit einer einzigen Projektion rekonstruieren. Soll jedoch ein beliebiges Dichtefeld rekonstruiert werden, so sind zusätzliche Informationen nötig, um die Verteilung im gesamten Untersuchungsfeld zu bestimmen. Daher greift man auf ein Tomographiesystem mit mehreren Kameras zurück. Details über die Realisierung dieses Systems mit einem dem BOS-Messsystem ähnlichen Verfahren können in *Fomin* [11] nachgelesen werden.

An dieser Stelle werden nur die Grundlagen dargestellt, da eine detaillierte Erklärung solcher Systeme nicht Thema dieser Arbeit ist.

2.3.1 Aufbau und Durchführung

Um genügend Informationen über das zu untersuchende Dichtefeld zu erhalten, werden mehrere Kameras (üblicherweise 8-16) in festgelegten Winkelabständen auf einem Halbkreis um das Dichtefeld herum aufgestellt (Abb. 2.7).

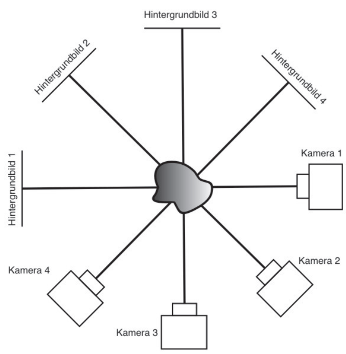


Abbildung 2.7: Aufbau des Tomographiesystems mit mehreren Kameras

Alle Kameras nehmen zur gleichen Zeit die Hintergrundbilder mit den Informationen über das Dichtefeld auf. Dafür kann die oben beschriebene Methode BOS oder die Speckle Fotografie[1] benutzt werden.

2.3.2 Auswertung und Kalibrierung

Von jeder Kamera liegt nun ein Referenzbild und ein Messbild mit verschobenem Punktemuster vor. Für jedes dieser Bilderpaare wird daraufhin nach dem in Kapitel 2.2.3 beschriebenen Verfahren eine Verschiebungsvektor-Matrix erstellt.

Allerdings können diese Matrizen (bzw. die "Bilder" mit den Verschiebungsvektoren) nicht ohne weiteres zur Rekonstruktion des Dichtefeldes verwendet werden. Eine "Kalibrierung"

[1] Ein dem BOS-Messsystem sehr ähnliches Verfahren, bei dem LASER (paralleles Licht) benutzt werden (siehe *Fomin* [11])

der Bilder, d.h. Bestimmung des Maßstabes und der Positionen der Kameras zueinander, ist nötig.

Die erste Kalibrierung dient dem Zweck, die Werte Δx bzw. Δy, die nach der Kreuzkorrelation in Pixel vorliegen, in Meter umzurechnen. Um den Maßstab zwischen Hintergrundbild und aufgenommenem Messbild zu erhalten, wird ein Kalibriergitter verwendet, welches bei der BOS-Aufnahme vor das Hintergrundbild montiert wird. Zur Errechnung des Maßstabes wird vor den eigentlichen BOS-Messungen eine Aufnahme des Hintergrundbildes mit dem Kalibriergitter gemacht. Der Abstand der einzelnen Gitterstäbe ist bekannt. Der Abstand der Gitterlinien im aufgenommenen Bild kann mit dem Computer ermittelt werden, so dass ein Wert in Pixel vorhanden ist. Anhand dieser beiden Daten erhält man den Maßstab (Umrechnungsfaktor) von Hintergrundbild zu Aufnahmebild.

Für die Kalibrierung der Kameras zueinander wird während der BOS-Aufnahmen, statt des Dichtefeldes vor einem Punktemuster, ein einfaches Bild eines Kalibrierungsobjektes (einer Kugel) aufgenommen. Auf Grund kleiner Ungenauigkeiten in der Ausrichtung der Kameras befindet sich das Kalibrierungsobjekt auf den einzelnen Bildern nicht immer an der gleichen Position. Für jedes Bild wird nun ein Bereich bestimmt, in dessen Mitte sich das Kalibrierungsobjekt befindet. Werden alle diese Bereiche in der gleichen Größe gewählt, so ist garantiert, dass sie alle denselben Ausschnitt des Dichtefeldes zeigen.
Nur der so bestimme Bereich jedes Bildes wird im folgenden für die Rekonstruktion des Dichtefeldes benutzt.

3. Rekonstruktion aus Projektionen

Die vorliegende Arbeit beschreibt eine Möglichkeit zur tomographischen Rekonstruktion von Dichtefeldern. Die Grundidee bei diesem Verfahren besteht darin, die Ablenkung von Licht durch Dichtegradienten auszunutzen, um die kontinuierliche Dichteverteilung eines Feldes zu errechnen. Dazu wird aus einem Versuch, wie in Kapitel 2.2 beschrieben, ein Verschiebungsvektorfeld gewonnen. An jedem Punkt dieses Vektorfeldes ist die jeweilige Verschiebung in Form eines Vektors eingetragen. Mit Hilfe der Beträge der einzelnen Vektoren wird der Winkel der Ablenkung des Lichts an der entsprechenden Position im Dichtefeld berechnet. Für jeden einzelnen Punkt im Dichtefeld wird nun die Information aller Lichtstrahlen herangezogen, welche den Punkt durchliefen und mit den jeweiligen Informationen wird versucht, die Dichte im Feld zu rekonstruieren.

3.1 Hinrechnung - Berechnung der Projektionen

Um später unbekannte Dichtefelder aus gegebenen Projektionen heraus rekonstruieren zu können, muss zunächst mit einem bekannten Dichtefeld gearbeitet werden. In dieser Arbeit wird dafür ein Freistrahl herangezogen. In einem Strömungslabor kann ein Freistrahl erzeugt werden, indem Luft horizontal mit einer Austrittsgeschwindigkeit zwischen 2 und 6 m/s ausgeblasen wird (Abb. 3.1).

Wie in Kapitel 1 beschrieben, besteht ein dreidimensionales Dichtefeld aus mehreren zweidimensionalen Schichten. Im Fall des Freistrahles werden diese Schichten entlang der horizontal liegenden Freistrahlachse "aneinandergestapelt". Im weiteren Verlauf der Arbeit wird zunächst eine einzelne Schicht betrachtet, um die Rekonstruktion eines Dichtefeldes durchzuführen.

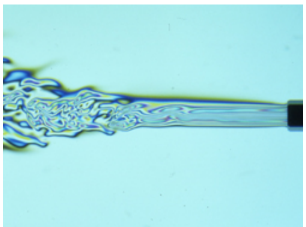


Abbildung 3.1: Freistrahl [9]

Es ist bekannt, dass die Verteilung der Dichte in solch einem Strahl "zuckerhutförmig" bzw. axialsymmetrisch verläuft. Aus Symmetriegründen reicht es daher, eine einzige Projektion zu erzeugen und diese für jeden Projektionswinkel zu verwenden.

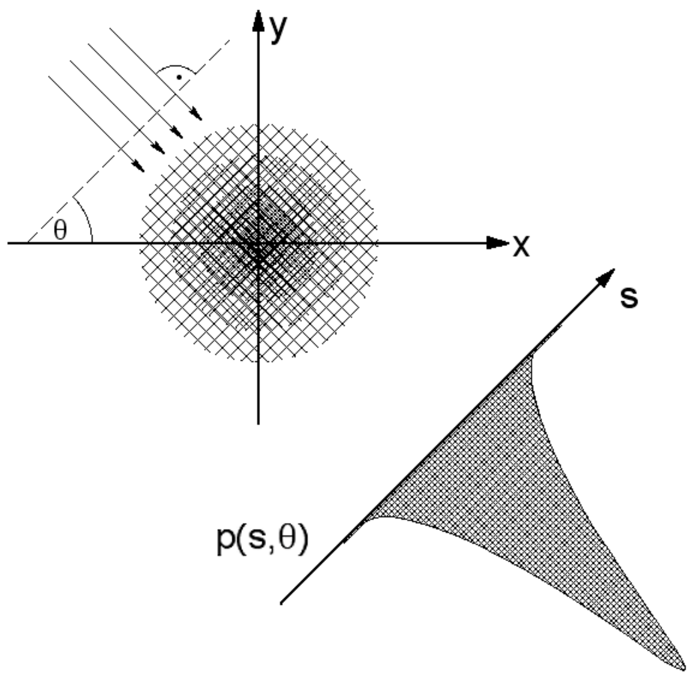


Abbildung 3.2: Gewinnung einer Projektion. In diesem Beispiel wird das Dichteprofil abgebildet.

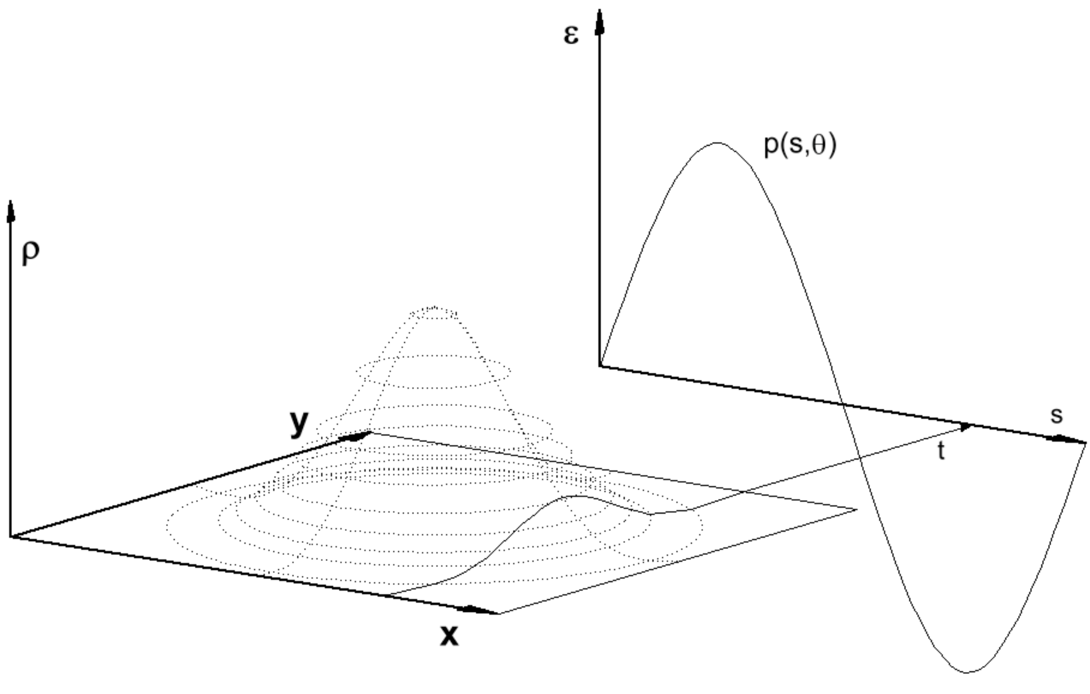


Abbildung 3.3: Verteilung der Dichte in einer Schicht des Freistrahls. In dieser Projektion wird der Ablenkungswinkel abgebildet.

Solch eine projektive Datenerfassung, wie sie durch das in Kapitel 2 beschriebene Verfahren vorgenommen wird, liefert das gleiche Ergebnis wie eine Radontransformation sie liefern würde. Objektinformationen werden entlang einer Linie t aufsummiert (integriert) und als Wert einem Punkt im Projektionsraum zugeordnet. Die Objektinformation ist in diesem Fall die Steigung der Dichtefunktion (Abb. 3.3) in s-Richtung entlang des Strahles t. Für den Ablenkungswinkel dieses Lichtstrahles gilt [3]:

$$\varepsilon(s,\theta) = \int \frac{\partial\varrho}{\partial s}\, G_\theta(s,t)\; dt \tag{3.1}$$

Die Projektionsdaten können nicht direkt umgerechnet werden, um die zugehörige Informationen über die zu analysierende Schicht zu erhalten. Statt dessen muss in mehreren Schritten die durch die Projektion kodierte Objektinformation entschlüsselt werden. Mathematisch wird die Datenkodierung in der Projektion wie folgt beschrieben. [16] [18]

$$\begin{aligned} [\mathscr{R}\varrho](s,\theta) &= \int_{-\infty}^{+\infty} \varrho(x,y)\; dt \\ &= \int_{-\infty}^{+\infty} \varrho(s\ cos\theta - t\ sin\theta, t\ cos\theta + s\ sin\theta)\; dt \end{aligned} \tag{3.2}$$

3.2 Errechnung des Ablenkungswinkels

Um die tomographische Rekonstruktion durchführen zu können, wird der Ablenkungswinkel des Lichtstrahles am Dichtefeld in Abhängigkeit von der Einfallsposition benötigt. Was jedoch im Vektorfeld vorliegt, ist nur die Information der Ablenkung in Form von Δx und Δy an der Pixelstelle s in der Projektion. Es ist also notwendig, das anfängliche Verschiebungsvektorfeld erst einmal umzurechnen, so dass es einer Matrix mit Ablenkungswinkeln entspricht.

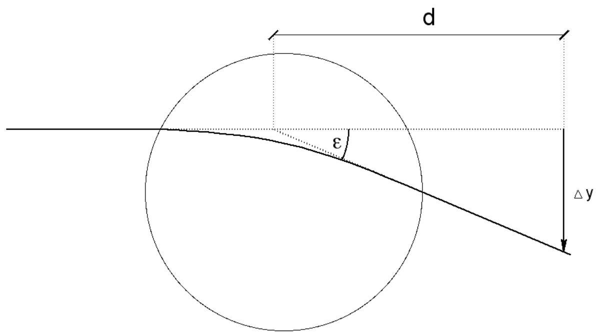


Abbildung 3.4: Lichtablenkung

Im Bild 3.4 ist der Weg eines Lichtstrahls abgebildet der ein Dichtefeld durchdrungen hat und dort kontinuierlich abgelenkt wurde. Der Winkel ε ist der Ablenkungswinkel des Lichtstrahles. Gerechnet wird mit dem rechtwinkligen Dreieck, welches durch *d* und Δy aufgespannt wird. Die Größen *d* und Δy sind bekannt. Für *d* wird angenommen, dass es der Abstand zwischen Objektmitte und Hintergrundbild ist. Δy erhält man aus der Auswertung der Messbilder. Bei der Rechnung für den Ablenkungswinkel muss Δy mit dem Maßstab (siehe Kapitel 2.3.2) multipliziert werden. Mit dem Tangens kann ε nun errechnet werden:

$$\varepsilon = arctan(\frac{\Delta y}{d}) \tag{3.3}$$

Die nun vorliegenden Matrizen liefern den Ablenkungswinkel eines Lichtstrahls in Abhängigkeit von seiner Einfallsposition in das Dichtefeld. Sie repräsentieren die unterschiedlichen Projektionen, wobei jede Projektion Informationen eines anderen Aufnahmewinkels darstellt.

3.3 Umrechnung des Ablenkungswinkels in die Dichte

In der klassischen Computertomographie enthalten die Projektionen direkte Informationen über das zu rekonstruierende Objekt, d.h. dass beispielsweise die Projektion eines Dichtefeldes das Dichteprofil abbildet (vgl. Abb. 3.2). Im Fall der BOS-Messungen wird statt der Dichte jedoch die Punkteverschiebung (Ablenkungswinkel, Kapitel 3.2) auf den Projektionen abgebildet. Wie erhält man aus dieser Information nun den Brechungsindex bzw. die Dichte?

Gleichung 3.1 ist grundlegend für die Rekonstruktion des Brechungsindexes (Dichtefeldes) aus den einzelnen Projektionsdaten (Ablenkungswinkeln).

Der mathematischen Herleitung der Berechnung der Dichte wird die Erklärung der Rekonstruktion und des Filters vorgezogen. Im darauffolgenden Abschnitt wird die Herleitung der im Programm verwendeten Formeln beschrieben.

3.4 Rekonstruktion durch Rückprojektion

Das Ziel der tomographischen Rekonstruktionsverfahren besteht darin, aus gewonnenen Messwerten die räumliche Verteilung der Dichte zu rekonstruieren. Eine Projektion stellt dabei die Integration der Dichte längs eines Strahles dar. Die Radon-Transformation beschreibt die Gesamtheit aller Projektionen. Die tomographische Rekonstruktion ist also die Suche nach der Inversion, d. h. der Umkehrung der Projektion.

Es gibt mehrere Arten von Rekonstruktionsverfahren. Die beiden wichtigsten sind:

- Iterative Rekonstruktionstechniken
- gefilterte Rückprojektion (engl.: filtered backprojection, FBP)

Bei ersterem wird zunächst eine Annahme über die Verteilung der zu untersuchenden Schichten gemacht; meist ist es eine Gleichverteilung. Aus einem so angenommenen Ausgangsbild werden erste Projektionen berechnet. Diese werden dann mit den aufgenommenen Projektionen verglichen. Stimmen sie nicht überein, wird ein entsprechender Korrekturfaktor gebildet und die Projektionen werden damit neu berechnet. Anschließend wird erneut mit den aufgenommenen Projektionen verglichen. Die Iteration wird ab dem Vergleichen fortgesetzt oder beendet, sobald definierte Abbruchkriterien erfüllt sind. Nachteil dieser Technik ist der erhöhte Rechenaufwand, den die Iteration mit sich bringt. Mehr über dieses Verfahren kann man im Buch 'Image Reconstruction' von G. Herman [13] nachlesen.

Bei der gefilterten Rückprojektion werden die Projektionswerte entlang ihres Strahlengangs (Integrationspfad) aufgetragen. Dazu werden in jedem Punkt des rekonstruierten Dichtefeldes all jene Projektionswerte aufsummiert, deren Integrationspfad durch diesen Punkt ging.

Dafür wird für jede Schnittfläche zunächst ein quadratisches Gitter definiert, wobei jede Zelle dieses Gitters einem Punkt im rekonstruierten Dichtefeld entspricht. Die Seitenlänge des Gitters entspricht der Datenlänge der Schichten in den Projektionen. Haben beispielsweise die Schichten der Projektionen N Daten, so definiert man ein N x N Gitter.

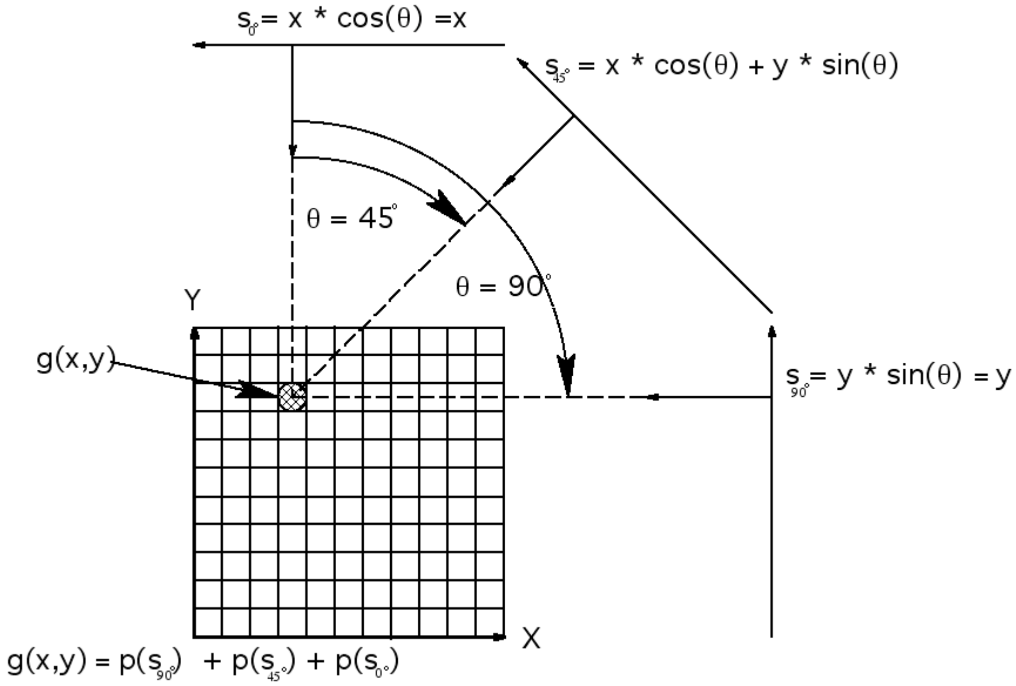


Abbildung 3.5: Prinzipielle Vorgehensweise bei der Rückprojektion

Im Hinblick auf praktische Anwendungen wird ein Kreis mit Radius N/2 als Definitionsbereich der Funktion g angenommen. Außerhalb dieses Kreises ist g gleich Null.

$$g(x, y) = 0 \ \forall (x, y) : \sqrt{x^2 + y^2} > \frac{N}{2} \tag{3.4}$$

Um nun für jeden Punkt $g(x, y)$ innerhalb des Kreises den Dichtewert zu erhalten, müssen aus allen Projektionen die jeweiligen Werte an der Stelle $s = x * cos(\theta) + y * sin(\theta)$ aufsummiert werden.

Nicht immer kann einem Punkt $g(x, y)$ im rekonstruierten Dichtefeld ein Wert aus einer Projektion eindeutig zugewiesen werden, wenn $p(s, \theta)$ mit $s = x*cos(\theta)+y*sin(\theta)$ eine Zahl zwischen zwei Projektionswerten liefern würde. In einem solchen Fall wird zwischen den beiden Werten aus der Projektion interpoliert. Die zwei am häufigsten genutzten Interpolationsverfahren sind: *nächster Nachbar* und die *lineare Interpolation*.

Die *nächster Nachbar* Interpolation (engl.: nearest neighbor) ist die schnellste Interpolationsmethode, da keine Arithmetik benutzt wird. Die aus $s = x * cos(\theta) + y * sin(\theta)$ gewonnene Zahl wird einfach kaufmännisch gerundet. Diese Methode ist jedoch meist die am wenigsten adäquate, da sie häufig Diskontinuitäten aufweist.

Die *lineare Interpolation* ist rechnerisch aufwendiger als die des *nächsten Nachbarn*, sie liefert jedoch bessere Werte:

$$p_{interpoliert}(s,\theta) \quad = \quad (p(\lceil s \rceil,\theta) - p(\lfloor s \rfloor,\theta)) * s\%1) + p(\lceil s \rceil,\theta) \tag{3.5}$$

Abbildung (3.6) zeigt in mehreren Schritten, wie die Rückprojektion das ursprüngliche Bild rekonstruiert.

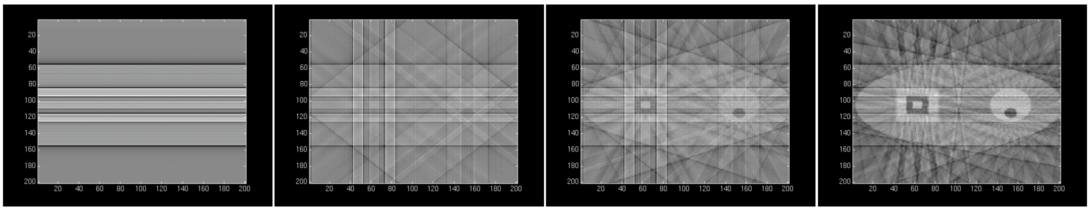


Abbildung 3.6: Rückprojektion mit einer, vier, acht und 15 Projektionen [22]

Wird allein die Rückprojektion verwendet, um die Rekonstruktion durchzuführen, so erhält man ein verschwommenes Bild, welches sich vom ursprünglichen Dichtefeld inakzeptabel unterscheidet. Um diesen Effekt zu verringern, wird vor der eigentlichen Rekonstruktion der Datensatz gefiltert.

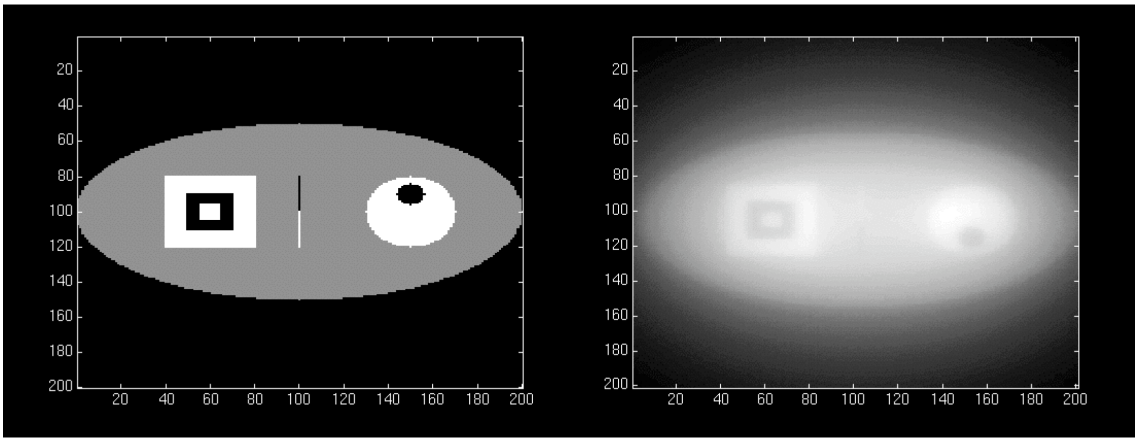


Abbildung 3.7: links das Bild vor der Rekonstruktion und rechts das Bild der einfachen Rückprojektion [22]

3.5 Filterung

Es gibt unterschiedliche Filtermethoden. Sie alle haben die Funktion, Verwischungen (engl.: smoothing) oder Störungen zu verringern. Durch die Anwendung unterschiedlicher Filter kann an jeweils anderen Ursachen der Bildstörungen gearbeitet werden.

Eine anschauliche Deutung dieses Schrittes in der Bildrekonstruktion ist die Gewichtung einer Funktion mit einer anderen. Der Funktionswert der Gewichtsfunktion an einer Stelle gibt an, wie stark der Wert der gewichteten Funktion in den Wert der Ergebnisfunktion eingeht.

Eine Filterfunktion multipliziert mit dem Frequenzraum ist die Filterung. Nach dem Faltungstheorem:

$$F() \cdot G() = \mathscr{F}^{-1}[F() * G()] \tag{3.6}$$

kann man die Multiplikation der Filterfunktion im Frequenzraum mit der Faltung im Ortsraum gleichsetzen.

Die meist benutzten Methoden für die Computertomographie aus Daten paralleler Lichtprojektionen sind die Faltungsmethoden. Der Grund hierfür ist die einfache Implementierung und die hohe Genauigkeit, die sie liefern. [13]

Im Weiteren wird jedoch der Frequenzraum herangezogen, um die Filterfunktion und seine Auswirkung auf die Bilder zu erklären. Dazu werden die Daten fouriertransformiert.

Ein Filter ist die Multiplikation der Rampenfunktion $|f|$ im Frequenzraum mit einem Fenster[1].

Eine Grenzfrequenz (engl.: *cutoff-frequency*) wird gesetzt, um den im Fourierraum zu definierenden Bereich einzugrenzen. Alle Frequenzen oberhalb der Grenzfrequenz werden "abgeschnitten", d.h. die diesen Frequenzen zugehörigen Werte werden gleich '0' gesetzt.

Fenster dienen zum einen dazu, eine solche Grenzfrequenz festzulegen, und zum anderen dazu, die Rampenfunktion $|f|$ so zu beeinflussen, dass bestimmte Frequenzen nur einen Bruchteil ihrer ursprünglichen Amplitude besitzen. In dieser Arbeit wird nur ein einfaches *Rechteck* Fenster benutzt (Abb. 3.9) welches ausschließlich eine Grenzfrequenz festlegt.

Man erhält also:

$$\text{Modifizierte Projektion} \;=\; \mathscr{F}^{-1}[|f|w(f)\mathscr{F}(\text{Projektionsdaten})] \tag{3.7}$$

$$\begin{aligned} \mathscr{F} \quad &- \quad \text{beschreibt die eindimensionale Fouriertransformation} \\ w(f) \quad &- \quad \text{beschreibt eine der Fensterfunktionen} \end{aligned}$$

Nach dem Faltungstheorem (3.6) ist das gleichzusetzen mit:

$$\text{Modifizierte Projektion} \;=\; \mathscr{F}^{-1}[|f|w(f)] * (\text{Projektionsdaten}) \tag{3.8}$$

$$\begin{aligned} \mathscr{F}^{-1}[|f|w(f)] \quad &- \quad \text{beschreibt die Faltungsfunktion} \\ \text{Symbol} * \quad &- \quad \text{bedeutet Faltung} \end{aligned}$$

[1]Diese Fenster finden in der Elektrizität und in der Signalverarbeitung großen Gebrauch. Mehr über diese Fenster kann man in den Büchern [25] und [26] nachlesen.

RAMP Filter

Die in Abbildung 3.7 gezeigte Unschärfe ist mit folgenden Bildern der Rückprojektion besser nachvollziehbar.

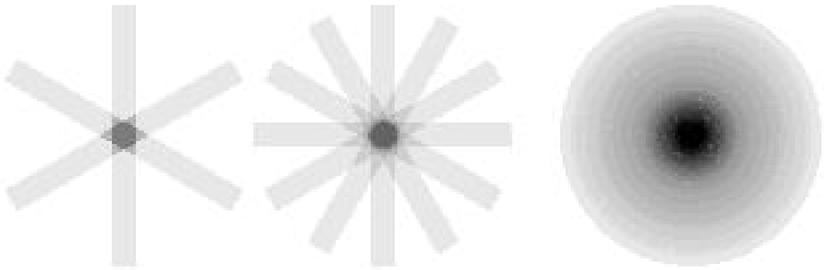


Abbildung 3.8: Rückprojektion eines Punktes aus drei, sechs und unendlich vielen Projektionen einer Punktquelle [24]

Wenn eine kleine Anzahl von Projektionen benutzt wird, sind sternförmige Artefakte beim rekonstruierten Bild zu sehen. Eine Rekonstruktion mit einer unendlichen Anzahl von Projektionen kann mit einer Faltung der Ausgangsverteilung mit der Funktion $g(r,\theta) = |r|^{-1}$ gleichgesetzt werden. Der Zusammenhang zwischen der ursprünglichen Verteilung und der Rückprojizierten kann wie folgt beschrieben werden [24]:

$$\mathscr{B}(r,\theta) = p(r,\theta) * |r|^{-1} \tag{3.9}$$

Wendet man das Faltungstheorem (3.6) an, so erhält man:

$$\mathscr{F}(\mathscr{B}(r,\theta)) = \mathscr{F}(p(r,\theta)) \cdot \mathscr{F}(|r|^{-1}) \tag{3.10}$$

mit $\mathscr{F}(|r|^{-1}) = |r_f|^{-1}$ gilt dann:

$$p(r,\theta) = \mathscr{F}^{-1}(\mathscr{F}(\mathscr{B}(r,\theta)) \cdot |r_f| \tag{3.11}$$

Mit f_c als Grenzfrequenz ist die Rechteck-Fensterfunktion definiert als:

$$w(f) = \begin{cases} 1, & \text{wenn } |f| \leq f_c \\ 0, & \text{wenn } |f| > f_c \end{cases} \tag{3.12}$$

Multipliziert man die Rechteck-Fensterfunktion mit der *Rampenfunktion* im Frequenzraum, so ergibt dies [14]:

$$c_f(f) = \begin{cases} |f|, & \text{wenn } |f| \leq f_c \\ 0, & \text{wenn } |f| > f_c \end{cases} \tag{3.13}$$

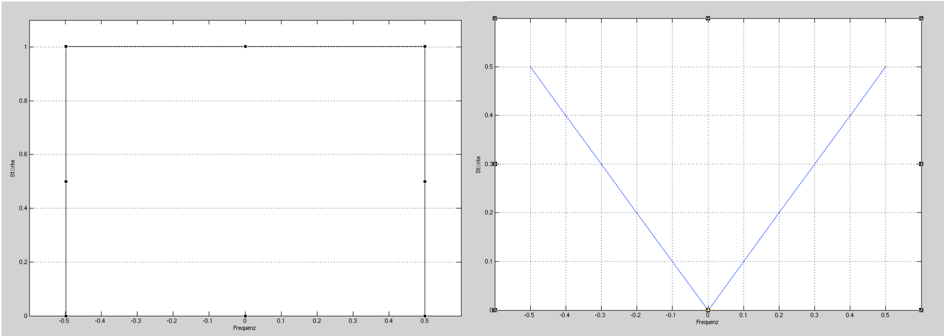


Abbildung 3.9: Die Rechteck - Fensterfunktion (links) und der RAMP Filter (rechts)

Die Fouriertransformierte des RAMP Filters ergibt im Ortsraum dann die Faltungsfunktion :

$$c(x) = 2f_c^2 \left(\frac{sin 2\pi f_c\ x}{2\pi f_c\ x} \right) - f_c^2 \left(\frac{sin \pi f_c\ x}{\pi f_c\ x} \right) \tag{3.14}$$

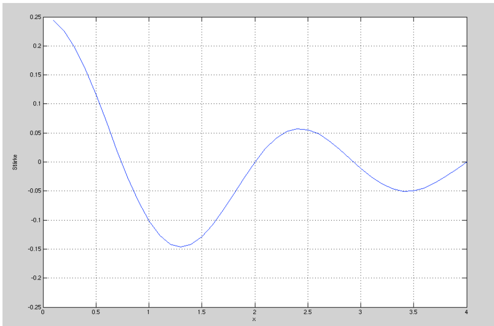


Abbildung 3.10: Faltungsfunktion des *RAMP Filters*

In Bild 3.11 wird das Ergebnis der Rückprojektion unter Anwendung des *RAMP Filter* mit dem ursprünglichen Bild verglichen. Die Unschärfe konnte behoben werden. Jedoch zeichnen sich anderweitige Störungen im Bild ab [14].

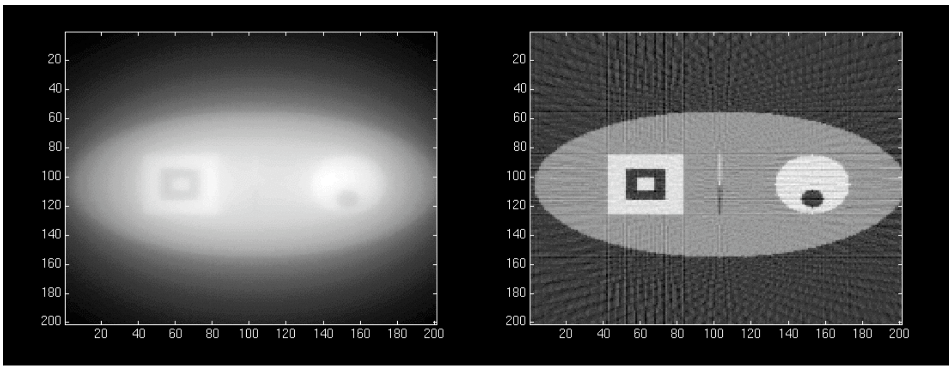


Abbildung 3.11: links die Rückprojektion ohne und rechts mit Filterung (*RAMP Filter*) [22]

Der *RAMP Filter* ist der meist genutzte Filter in der Computertomographie. Er hat die Eigenschaft, niedrige Frequenzen herauszufiltern und hohe Frequenzen zu verstärken.

Zwar wird die Unschärfe verringert, welche sich als niedrige Frequenzen im Frequenzraum auszeichnet, jedoch werden die kontrastreichen Eigenschaften (hohe Frequenzen) hervorgehoben, die Störungen im Bild verursachen. Aus diesem Grund können weitere Filter benutzt werden, um diesen Effekt zu verringern. Dafür müssen die hohen Frequenzen stärker rausgefiltert werden als es bei dem *RAMP Filter* der Fall ist. Um das zu erreichen, wird versucht, die Kurve der Filterfunktion in den hohen Frequenzen abzurunden.

3.6 Herleitung des Algorithmus

Die in Abb. 3.12 gezeigten Variablen seien definiert[2]:

$$\begin{aligned} x &= s\,cos(\theta) - t\,sin(\theta) \\ y &= t\,cos(\theta) + s\,sin(\theta) \\ s &= x\,cos(\theta) + y\,sin(\theta) \\ t &= y\,cos(\theta) - x\,sin(\theta) \end{aligned} \tag{3.15}$$

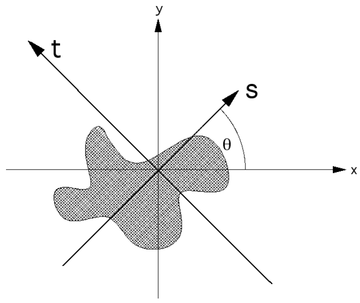


Abbildung 3.12: Der Strahlengang von t

Die Fouriertransformation[3] der Radontransformation (3.2) wird beschrieben durch:

$$\begin{aligned} \mathscr{F}\left[\mathscr{R}\varrho\right](l,\theta) &= \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} G_\theta(s,t)\;dt\right] e^{-2\pi i l s}\,ds \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \varrho(s\,cos(\theta) - t\,sin(\theta), t\,cos(\theta) + s\,sin(\theta))\, e^{-2\pi i l s}\,ds\,dt \\ &= \mathscr{R}(l,\theta) \end{aligned} \tag{3.16}$$

Die Fouriertransformation des Ablenkungswinkels (3.1) bezüglich s entspricht:

$$\begin{aligned} [\mathscr{F}\varepsilon](l,\theta) &= \int_{-\infty}^{\infty} \varepsilon(s,\theta)\; e^{-2\pi i l s}\,ds \\ &= \int\int_{-\infty}^{\infty} \frac{\partial\varrho}{\partial s}\;dt\; e^{-2\pi i l s}\;ds \\ &= \int\int_{-\infty}^{\infty} \frac{\partial\varrho}{\partial s}\; e^{-2\pi i l s} ds\;dt \\ &= E(l,\theta) \end{aligned} \tag{3.17}$$

[2]Die hier vorgestellten Überlegungen wurden von T.C. Liu, W. Merzkirch und K. Oberste-Lehn in ihrer Arbeit [3] gemacht.

[3]Jean Baptiste Joseph Fourier (1768 - 1830), Französischer Mathematiker [20]

Mit dem Ableitungstheorem ist das:

$$\begin{aligned} E(l,\theta) &= 2\,\pi\,i\,l\,\mathscr{R}(l,\theta) \\ &= 2\,\pi\,i\,l\,\mathscr{F}[\mathscr{R}\varrho] \end{aligned} \tag{3.18}$$

oder:

$$\mathscr{R}(l,\theta) = \frac{E(l,\theta)}{2\,\pi\,i\,l} \tag{3.19}$$

Die inverse Fouriertransformation von (3.16) in Polarkoordinaten ergibt:

$$\begin{aligned} \varrho(x,y) &= \varrho(s\;cos(\theta) - t\;sin(\theta), t\;cos(\theta) + s\;sin(\theta)) \\ &= \int_0^{2\pi} \int_0^{\pi} \mathscr{R}(t,\theta) e^{2\,\pi\,i\,l\,s} |l|\;dl\;d\theta \end{aligned} \tag{3.20}$$

Die Integralgrenzen wurden geändert. $(0, 2\,\pi)$ wurde zu $(0,\pi)$ für θ, $(0,\infty)$ zu $(-\infty,\infty)$ für l und l, welches immer positiv ist, zu $|l|$ im Fourierraum.

Wird (3.19) in (3.20) eingesetzt so erhält man:

$$\begin{aligned} \varrho(x,y) &= \int_0^{\pi} \int_{-\infty}^{\infty} \left[\frac{E(l,\theta)}{2\,\pi\,i\,l}\right] |l|\; e^{2\,\pi\,i\,l\,s}\;dl\;d\theta \\ \varrho(x,y) &= \int_0^{\pi} d\theta \int_{-\infty}^{\infty} Q(l)\;E(l,\theta)\;e^{2\,\pi\,i\,l\,s}\;dl \qquad \text{mit } Q(l) = \frac{|l|}{2\,\pi\,i\,l} \end{aligned} \tag{3.21}$$

Mit dem Faltungstheorem (3.6) ergibt sich aus (3.21):

$$\varrho(x,y) = \int_0^{\pi} q(s) * \varepsilon(s,\theta)\;d\theta \tag{3.22}$$

$$\begin{aligned} \text{mit } q(s) &= \int_{-\infty}^{\infty} Q(l)\;e^{2\,\pi\,i\,l\,s}\,dl \\ \text{und } \varepsilon(s,\theta) &= \int_{-\infty}^{\infty} E(l,\theta)\,e^{2\,\pi\,i\,l\,s} dl \end{aligned}$$

wobei $q(s)$ die Filterfunktion ist. Das $q(s)$ ist mathematisch hier nicht ganz korrekt, da das Integral von $-\infty$ bis ∞ geht. $q(s)$ mit den Grenzen $\pm f_c$ wird $q_c(s)$ genannt. Das $q_c(s)$ ist als Näherung korrekt, da das Integral dann im weiteren mit endlichen Grenzen genutzt wird.

Es werden in einem Versuch M Projektionen in regelmäßigen Winkelabständen von $\Delta\theta$ gemacht. Für die Rekonstruktion des Dichtefeldes im Rekonstruktionsgitter wird die Auswertung der Projektionen in s-Richtung an N-Punkten mit dem regelmäßigen Abstand von Δs vorgenommen (Kapitel 3.4). Die Gleichung (3.22) sollte somit in diskretisierter Form vorliegen.

Die Projektionen werden im Ortsraum mit dem Intervall Δs abgetastet. Damit ergibt sich über das Abtasttheorem eine Grenzfrequenz f_c im Fourierraum.

$$f_c = \frac{1}{2\Delta\, s} \tag{3.23}$$

Der Filter aus Gleichung (3.22) kann somit erweitert werden zu:

$$q(s) \quad = \quad \int_{-\infty}^{\infty} Q(l)\, W(l)\, e^{2\,\pi\, i\, l\, s}\, dl \tag{3.24}$$

$W(l)$ ist hier die Fensterfunktion (3.12). Mit ihr wird $q(s)$ neu definiert zu:

$$\begin{aligned} q_c(s) \quad &= \quad \int_{-f_c}^{f_c} \frac{|l|}{2\,\pi\, i\, l}\, e^{2\,\pi\, i\, l\, s}\, dl \\ &= \quad \frac{1}{\pi} \int_0^{f_c} sin(2\,\pi\, l\, s)\, dl \\ &= \quad \frac{1 - cos(\pi \frac{s}{\Delta s})}{2\,\pi\, s} \end{aligned} \tag{3.25}$$

Mit $s = m\,\Delta s$ wird diskretisiert. Es ergibt sich:

$$\begin{aligned} q_c(s) \quad &= \quad \frac{1}{2\,\pi^2\, m\, \Delta s} \left[1 - cos\left(\frac{\pi\, m\, \Delta s}{\Delta s} \right) \right] \\ &= \quad \frac{1}{\pi^2\, \Delta s} \left[\frac{1 - cos(\pi\, m)}{2\, m} \right] \\ &= \quad \frac{1}{\pi^2\, \Delta s}\, q(m) \end{aligned} \tag{3.26}$$

wobei $q(m)$ die diskretisierte Filterfunktion darstellt:

$$q(m) \quad = \quad \begin{cases} \frac{1}{m}, & \text{wenn m ungerade} \\ 0, & \text{wenn m gerade oder 0} \end{cases} \tag{3.27}$$

$$\text{mit } m \quad = \quad 0, \pm 1, \pm 2, \ldots, \pm \frac{N-1}{2}$$

Mit Hilfe der Trapezregel kann die Gleichung (3.22) diskretisiert werden.
m sei definiert durch x und y (vgl. Kapitel 3.4),
wobei $x = (k \cdot \triangle x)$ und $y = (i \cdot \triangle y)$:

$$m = k \, \Delta x \; cos(\theta) + i \; \Delta y \; sin(\theta) \tag{3.28}$$

Wie in Kapitel 3.4 beschrieben, wird m gerundet, um auf die in (3.27) gezeigten Ganzzahlen zu kommen. Im Programm muss dann zwischen den Werten interpoliert werden.

Mit $s' = m' \cdot \Delta s$ gilt:

$$\varrho(k \cdot \Delta x, i \cdot \Delta y) = \Delta\theta \, \Delta s \sum_{j=0}^{M-1} \sum_{m'=-\frac{(N-1)}{2}}^{\frac{(N-1)}{2}} \varepsilon(m' \, \Delta s, \; j \, \Delta\theta) \; \underbrace{q((m-m')\Delta s)}_{(\pi^2 \Delta s)^{-1} \cdot q(m)} \tag{3.29}$$

Mit $\Delta\theta = \frac{\pi}{M}$ gilt:

$$= \frac{\pi}{M} \frac{1}{\pi^2 \Delta s} \Delta s \sum_{j=0}^{M-1} \sum_{m'=-\frac{(N-1)}{2}}^{\frac{(N-1)}{2}} \varepsilon(m' \, \Delta s, j \, \Delta\theta) \, q(m-m')$$

$$= \frac{1}{M\pi} \sum_{j=0}^{M-1} \sum_{m'=-\frac{(N-1)}{2}}^{\frac{(N-1)}{2}} \varepsilon(m' \, \Delta s, j \, \Delta\theta) \, q(m-m') \tag{3.30}$$

wobei k, i, j, m und $m' \in \mathbf{R}$, Δx und Δy die Gitterabstände in x und y Richtung des diskretisierten Rekonstruktionsgitters sind. Dabei wird die Formel (3.28) für den Zusammenhang von k und i mit m benutzt.

Es ergibt sich folgende Vorgehensweise:

- Faltung jedes Wertes in den Projektionen mit der Filterfunktion (3.26) im Ortsraum
- Rückprojektion der gefilterten Projektionen mit Interpolation für jeden Punkt im Rekonstruktionsgitter

4. Implementierung

4.1 Ziel und Anforderungen

Ziel der Arbeit ist es, ein Programm zur tomographischen Rekonstruktion von Dichtefeldern zu erstellen. Als Eingangsdaten soll das Programm die Messwerte aus dem BOS - Messsystem erhalten. Außerdem soll das Programm den folgenden Anforderungen gerecht werden:

- Die Anzahl der dem Programm übergebenen Projektionen soll flexibel gehalten sein.
- Die Rekonstruktion des Dichtefeldes erfolgt aus zweidimensionalen Schichten. Es soll die Möglichkeit bestehen, einen definierten Ebenenbereich anzugeben und vom Programm bearbeiten zu lassen.
- Die Erweiterung um neue Filterfunktionen soll möglichst einfach gestaltet werden.

Um der inhaltlichen Trennung, der Berechnung der Ablenkungswinkel zum einen und der gefilterten Rückprojektion zum anderen, gerecht zu werden, werden die Aufgaben auf zwei voneinander getrennte Programme verteilt.

Das Programm *calculateDeflectionAngle* berechnet die Ablenkungswinkel der Vektorfelder. Das Programm *filteredBackprojection* berechnet im Anschluss das Brechungsindexfeld.

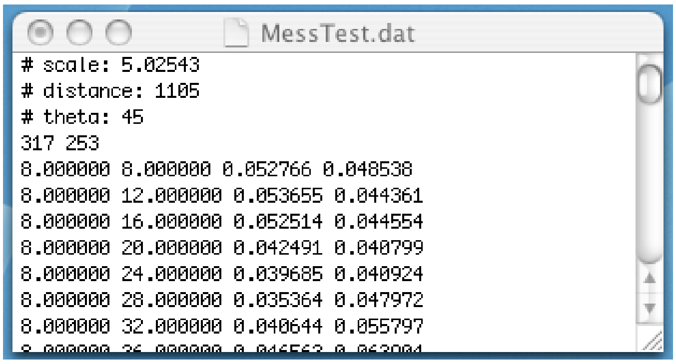
4.2 Programmiersprache

Für die Mitarbeiter der DLR ist es wichtig Programmiercode zu erhalten, der leicht erweiterbar ist. Daher fällt die Wahl der Programmiersprache auf eine objektorientierte. C++ ist für dieses Programm besser geeignet als Java, da die reine Compilersprache im Vergleich zur Interpretersprache Java schneller läuft. Der Geschwindigkeitsvorteil zahlt sich für die Programme in jedem Fall aus, so dass in Kauf genommen wird, die Programme für jede Rechnerarchitektur neu kompilieren zu müssen.

4.3 Einlesen der Daten

Die Eingangsdaten sind in Dateien abgelegt, die vom Programm gelesen werden. Um die Daten in das Programm zu laden, werden mittels der Bibliothek *load_data.h* alle für den Dateizugriff notwendigen Funktionen zur Verfügung gestellt. Damit in der Datei ersichtlich ist, welche Zahlen beispielsweise den Maßstab oder den Projektionswinkel beschreiben, werden diese Daten hinter einem bestimmten Schlüsselwort gespeichert. Der Maßstab steht beispielsweise hinter dem Schlüsselwort *scale*. Die Funktionen der Bibliothek suchen nach den jeweiligen Schlüsselwörtern in der Datei und lesen die darauf folgenden Daten aus. Sind keine Daten hinter einem Schlüsselwort zu finden, wird eine Fehlermeldung ausgegeben. Um die Daten im Programm aufzunehmen, werden die Template Klassen *list* und *vector* der STL (Standard Template Library) verwendet.

Das Programm *calculateDeflectionAngle* erhält als Parameter eine Datei. Aus dieser werden alle für das Berechnen des Ablenkungswinkels benötigten Informationen ausgelesen. Das Programm *filteredBackprojection* benötigt mehr als eine Datei. Statt der einzelnen Dateinamen kann hier jedoch ein Verzeichnis übergeben werden, welches alle benötigten Daten enthält. Um sämtliche Dateien in einem Verzeichnis zu erfassen und die Dateinamen in einer Liste abzulegen steht die Funktion *read_files()* in der Bibliothek zur Verfügung.


MessTest.dat

```
# scale: 5.02543
# distance: 1105
# theta: 45
317 253
8.000000 8.000000 0.052766 0.048538
8.000000 12.000000 0.053655 0.044361
8.000000 16.000000 0.052514 0.044554
8.000000 20.000000 0.042491 0.040799
8.000000 24.000000 0.039685 0.040924
8.000000 28.000000 0.035364 0.047972
8.000000 32.000000 0.040644 0.055797
```

Abbildung 4.1: Ausschnitt einer Datei im Tecplot-Format

Abbildung 4.1 zeigt mögliche Eingangsdaten. Der Maßstab, der Abstand der Dichtefeldmitte zum Hintergrundbild und der Projektionswinkel befinden sich in den ersten Zeilen der Datei. Die Angaben über Schichtanzahl und Größe einer Schicht stehen in der vierten Zeile. Daraufhin folgen zeilenweise die Daten der Matrix $(s - Koordinate, z - Koordinate, \Delta x, \Delta y)$.

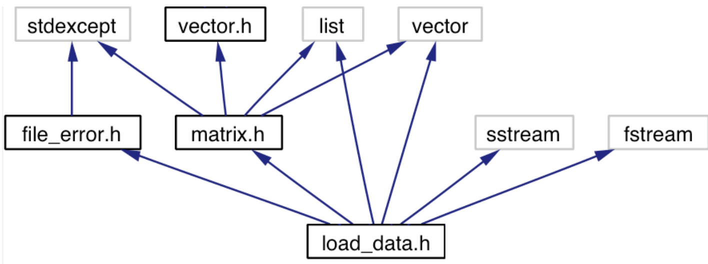


Abbildung 4.2: Abhängigkeiten der Bibliothek *load_data.h*

4.4 Die Klassen Vector und Matrix

Für die weitere Implementierung des Algorithmus werden die Klassen *Vector* und *Matrix* eingeführt. Beide Klassen beinhalten die Implementierung mathematischer Operatoren, sowie nützliche Methoden zum Arbeiten mit diesen Objekten.

Zu den Anforderungen an das Programm gehört unter anderem, die Rechnungen mit einer Genauigkeit von 10 Nachkommastellen auszuführen. Aus diesem Grund ist die Verwendung eines Datentyps notwendig, der die erforderliche Präzision liefert. Die nachfolgende Tabelle[1] veranschaulicht die Eigenschaften der Gleitkommazahlen [19][2].

Typ	Speicherplatz	Wertebereich	kleinste positive Zahl	Genauigkeit (dezimal)
float	4 Bytes	$\pm 3.4E+38$	$1.2E-38$	6 Stellen
double	8 Bytes	$\pm 1.7E+308$	$2.3E-308$	15 Stellen
long double	10 Bytes	$\pm 1.1E+4932$	$3.4E-4932$	19 Stellen

Die Tabelle zeigt, dass zum Erreichen der erforderlichen Genauigkeit der Datentyp *double* notwendig ist.

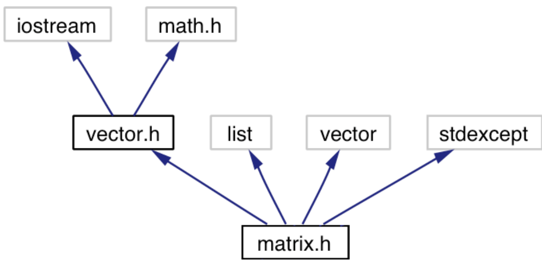


Abbildung 4.3: Abhängigkeiten der *Matrix*- und *Vector* Klassen

4.5 Programm: 'calculateDeflectionAngle'

Das Programm *calculateDeflectionAngle* ist dafür zuständig, den Ablenkungswinkel in Abhängigkeit von der Einfallsposition zu errechnen (vgl. Kapitel 3.2).

In einer Datei enthält das Programm folgende Eingangsdaten:

- Maßstab
- Abstand des Objektes zum Hintergrundbild
- Vektorfeld aus der Kreuzkorrelation

Für jede Projektion gibt es eine Datei, die das zugehörige Vektorfeld im Tecplot-Format enthält $(s-Koord., z-Koord., \Delta x, \Delta y)$. Die einzelnen Zahlen sind in der Datei durch ein Leerzeichen (genaugenommen jede Art von "whitespaces") getrennt.

[1]Zur Darstellung der Gleitkommazahlen wird üblicherweise das IEEE-Format (IEEE = Institute of Electrical and Electronical Engineers) verwendet. Von dieser Darstellung wird in obiger Tabelle ausgegangen. [19]

[2]Es ist zu beachten, dass die in der Tabelle aufgeführten Werte nicht für jeden Compiler und jede Rechnerarchitektur gleich sind. Diese Werte sollen lediglich als Richtwerte dienen.

Der Maßstab und der Abstand des Objektes zum Hintergrundbild sind, wie in Kapitel 4.3 beschrieben, als Kommentar hinter den Schlüsselwörtern *scale* und *distance* gespeichert.

Für das Einlesen der Daten wird die Funktion *getPIV_matrix()* benutzt. Sie liest neben dem Maßstab, dem Abstand und dem Vektorfeld zwei weitere Zahlen am Anfang der Datei aus. Diese Zahlen geben die Schichtanzahl und die Datenmenge pro Schicht an. Die Daten des Vektorfeldes werden nacheinander in einem Vektor gespeichert.

Daraufhin wird die Funktion *calcDeflectionAngle()* benutzt, um aus dem Maßstab, dem Abstand des Objektes zum Hintergrundbild und Δy den jeweiligen Ablenkungswinkel zu berechnen. Abschließend werden alle Ablenkungswinkel in eine neue Datei geschrieben. Um die Weiterverarbeitung durch das zweite Programm zu ermöglichen, wird am Anfang der Datei die Dimension der Matrix, d.h. die Schichtanzahl und -breite gespeichert.

Obwohl das zweite Programm *filteredBackprojection* keine weiteren Informationen benötigt, wird -zusätzlich zu jedem Ablenkungswinkel- die jeweilige *s*- und *z*-Koordinate gespeichert, um eine allgemeine Verwendung der Daten zu ermöglichen.

Ablauf der Arbeitsschritte im Code der *main*-Funktion:

```
int main(int argc, char* const argv[])
{
    .
    .
    try
    {
        vector<long double> field;
        vector<int> dimensions(2);
        list<Matrix> projections;
        .
        .
        /** loads the data from a file **/
        get_PIVmatrix(argv[1], field, dimensions);

        /** calculates the deflection angle and saves the result in a file **/
        calculateDeflectedAngle(field, dimensions, distance, scale, col);
    }
    catch(){...}
    .
    .
}
```

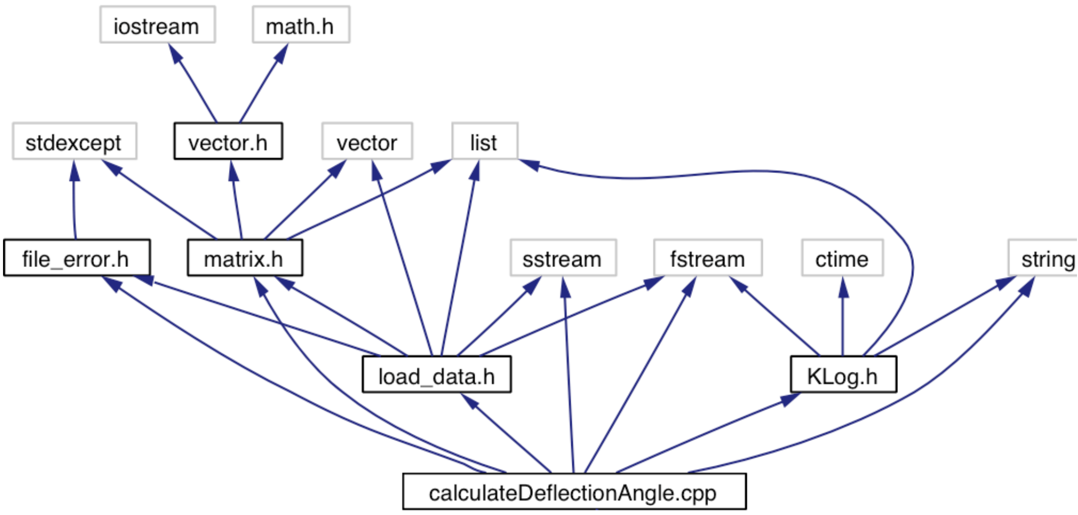


Abbildung 4.4: Klassendiagramm des Programms *calculateDeflectionAngle*

4.6 Programm: 'filteredBackprojection'

Die Berechnung des Brechungsindex ist im Programm *filteredBackprojection* in zwei Arbeitsschritte unterteilt. Im ersten Schritt, der sog. Filterung, werden aus den Eingangsdaten Objekte erstellt. Dazu wird die Klasse *filteredField* verwendet.
Im zweiten Schritt, der sog. Rekonstruktion, wird das dreidimensionale Brechungsindexfeld berechnet.

Als Eingangsdaten dienen die vom ersten Programm erzeugten Dateien. Sie enthalten die Ablenkungswinkel und den Projektionswinkel.

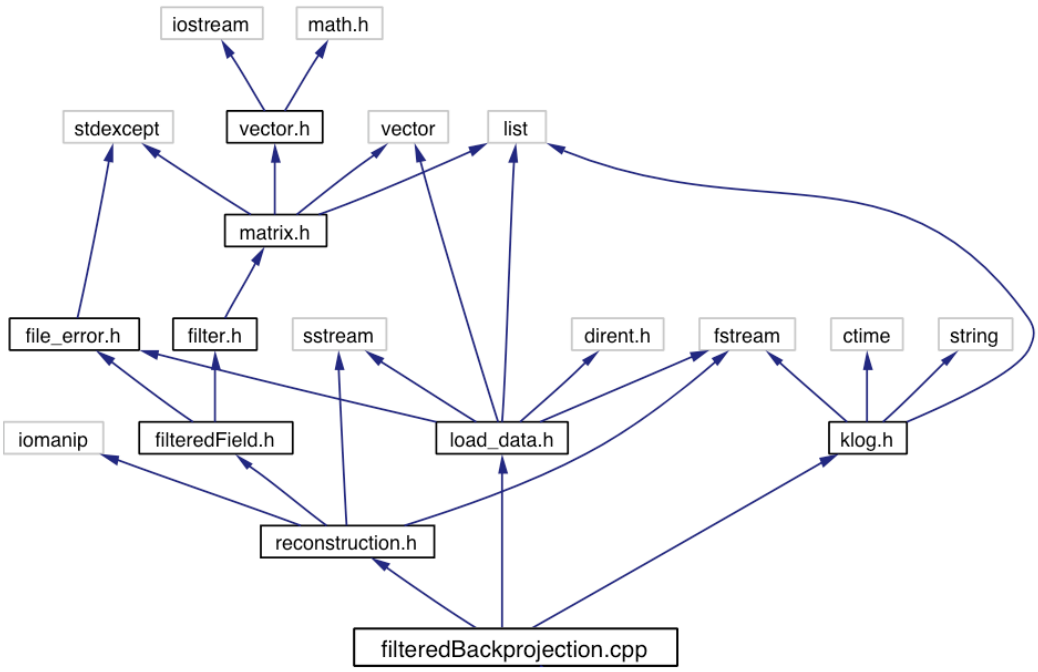


Abbildung 4.5: Klassendiagramm des Programms *filteredBackprojection*

4.6.1 Filterung

Als Eingangsdaten dienen Dateien mit Ablenkungswinkeln, die vom ersten Programm erstellt wurden. (Zur Erinnerung: Die Daten jeder einzelnen Kamera (Kapitel 2.3.1) werden vom ersten Programm zu je einer Datei verarbeitet (Kapitel 4.5).) Mittels eines Filteralgorithmus wird aus den Daten jeder Datei ein *filteredField*-Objekt erstellt.

Die Bibliothek *filter.h* stellt Filter zur Beseitigung von Störungen zur Verfügung. Im Rahmen dieser Arbeit habe ich bereits einen Filteralgorithmus in der Bibliothek implementiert. Da beim Erstellen eines *filteredField*-Objekts der Filter als Funktionszeiger übergeben wird, können weitere Filter problemlos implementiert und zur Laufzeit (z.B. durch Benutzereingabe) ausgewählt werden. Somit ist es möglich, den Filter zu wechseln, ohne das Programm noch einmal erneut kompilieren zu müssen. Neue Filterfunktionen dürfen sich im Namen, nicht jedoch in Rückgabetyp und Parameterliste von der bereits implementierten Funktion *rampFilter()* unterscheiden.

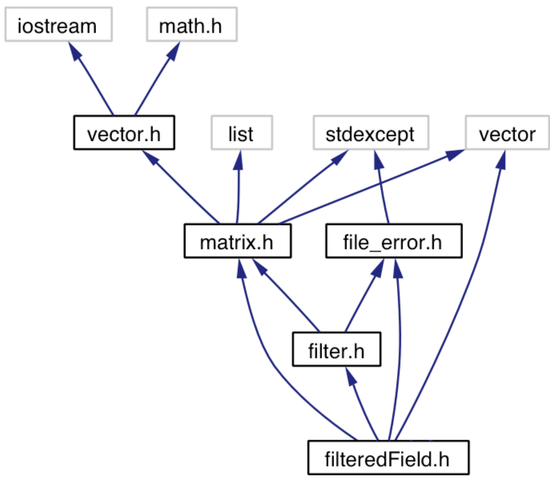


Abbildung 4.6: Abhängigkeiten der Klasse *filteredField* und der Bibliothek *filter.h*

4.6.2 Rekonstruktion

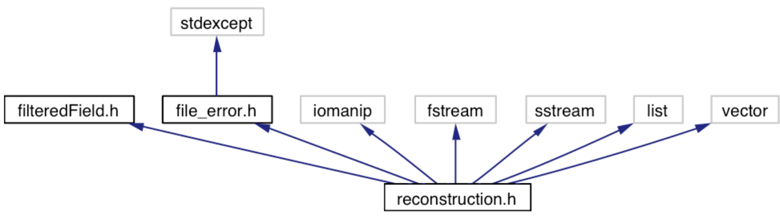


Abbildung 4.7: Abhängigkeiten der Bibliothek *reconstruction.h*

Die Funktion *reconstruction()* berechnet der Reihe nach die Brechungsindizes aller Schichten des Dichtefeldes. Mittels einer Parameterübergabe beim Programmstart, ist es möglich, statt allen Schichten, nur eine bestimmte Auswahl zu berechnen. Für die Berechnung des Brechungsindex jedes einzelnen Punktes aus den Daten der *filteredField*-Objekte wird die Funktion *backprojection()* verwendet.

Wie in Kapitel 3.3 beschrieben, kann man sich ein Dichtefeld aus zweidimensionalen Schichten aufgebaut vorstellen. Setzt man diese zusammen, so erhält man wieder die dreidimensionale Darstellung des Feldes. Da eine Datei, welche die Ergebnisse für alle Schichten enthält, so groß wäre, dass sie von der Auswertungssoftware (Tecplot, graphische Darstellung des Brechungsindexfeldes) nur sehr langsam verarbeitet werden könnte, wird das Brechungsindexfeld jeder Schicht in einer separaten Datei abgelegt. Für die Berechnung der einzelnen Brechungsindizes werden Daten aus allen *filteredField*-Objekten benötigt. Diese werden daher der Funktion *backprojection()* übergeben, welche den Index berechnet und zurückliefert.

Die Funktion *reconstruction()* geht dann mit Hilfe von 3 *for*-Schleifen für jede Schicht jeden Punkt im Gitter durch und ruft zur Berechnung des jeweiligen Brechungsindex die Funktion *backprojection()* auf.

Indexberechnung in der Funktion *reconstruction()*:

```
Fuer jede Schicht mache
    Fuer jeden Index der Gitterlaenge mache
        Fuer jeden Index der Gitterbreite mache
            rekonstruiere diesen Punkt durch die Rueckprojektion
            bzw. berechne fuer jeden Punkt den Brechungsindex
        Ende
    Ende
Ende
```

Die Funktion *backprojection()* berechnet für den Punkt (x, y) in der aktuellen Schicht den Brechungsindex. Dafür benötigt sie aus jeder Projektion einen diesem Punkt entsprechenden Wert. Diese Werte ermittelt sie mit Hilfe linearer Interpolation (vgl. Kapitel 3.4).

4.7 Fehlerbehandlung

Für die Fehlerbehandlung werden *exceptions* benutzt. In erster Linie wird die Standard Fehlerklasse *logic_error* benutzt. Für Fehler, die im Zusammenhang mit Dateien entstehen, findet die Fehlerklasse *file_error* Verwendung, welche von der Fehlerklasse *logic_error* erbt. Diese Klasse speichert zusätzlich zur Fehlerinformation auch den Dateinamen. So ist es für den Anwender einfacher, Fehler, die beispielsweise auf ungültige Zeichen in einer Datei zurückzuführen sind, aufzufinden und zu korrigieren.

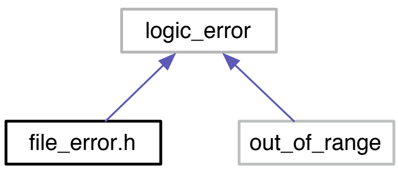


Abbildung 4.8: Übersicht der verwendeten Fehlerklassen

4.8 Programmanalyse

4.8.1 Fehleranalyse

Die Fehleranalyse wird mit dem Datensatz eines Freistrahls berechnet. Zum Vergleich der Ergebnisse stehen Referenzdaten zur Verfügung. Der absolute Fehler wird aus der Differenz dieser Daten und den Ergebnissen des C++ Programms berechnet. Daraufhin wird der aussagekräftigere relative Fehler, die prozentuale Abweichung bestimmt. Es zeigt sich, dass durch die Verwendung der Programmiersprache C++ eine höhere Genauigkeit erzielt werden kann.

Abweichungen zu den Referenzdaten:

- C++: 12,914 ‰
- Matlab: 12,915 ‰

4.8.2 Laufzeitanalyse

- Programm zur Berechnung des Ablenkungswinkels: *calculateDeflectionAngle*

Operation	Laufzeit (O-Notation)
Einlesen der Daten	$O(n)$
Berechnung des Winkels	$O(n)$

n - Anzahl der Daten in einer Datei

- Programm zur Berechnung des Brechungsindex: *filteredBackprojection*

Operation	Laufzeit (O-Notation)
Einlesen der Daten	$O(n * k)$
Filtern der Daten	$O(n/4 * k)$
Rückprojektion	$O(n/4 * k)$

n - Anzahl der Daten in einer Datei
k - Anzahl der Projektionen

Die Laufzeit des Programms *calculateDeflectionAngle* ist mit wenigen Sekunden so kurz, dass sich eine genaue Untersuchung der einzelnen Arbeitsschritte erübrigt.
Ganz anders die Situation beim zweiten Programm. Eine Laufzeit von einer halben Stunde (siehe unten) gilt es durch geschickte Optimierung zu verkürzen.

Um die Laufzeit des Programms besser analysieren zu können, wird an bestimmten Positionen im Programm die Zeit ausgegeben, die zum Beenden der entsprechenden Operationen benötigt wird. Zur Messung der Laufzeit wird die Rekonstruktion aus 4 Projektionen eines Freistrahls benutzt, die jeweils 319 Schichten und 255 Daten pro Schicht besitzen.

Die nachfolgende Tabelle zeigt die Laufzeit in Minuten und Sekunden an, die benötigt wird, um den Brechungsindex zu berechnen:

Operation	Laufzeit (min : sek)
Einlesen der Daten pro Projektion	00 : 03
Filtern der Daten pro Projektion	00 : 08
Rückprojektion	02 : 45
Schreiben der Daten	19 : 52
Gesamtdauer	26 : 54

Das Speichern der Daten nimmt offensichtlich den größten Teil der Zeit in Anspruch. In diesem Arbeitschritt werden die Ergebnisse, welche in einem Objekt des Typs *Matrix* gespeichert sind einzeln in die Ausgabedateien geschrieben. Das Problem dabei ist die Größe des Objekts. Mit 319 x 255 x 255 (Schichten x Breite x Länge), also mehr als 20 Millionen Einträgen dauert das Erstellen und Löschen des *Matrix*-Objektes sehr lange, da Daten vom Hauptspeicher auf die Festplatte ausgelagert werden müssen. Dabei ist das Vorhandensein der kompletten Ergebnisdaten im Programm gar nicht notwendig. Der Verzicht auf das *Matrix*-Objekt und das Schreiben der Daten in die Ausgabedateien unmittelbar nach deren Berechnung führt zur folgenden neuen Laufzeit:

Operation	Laufzeit (min : sek)
Einlesen der Daten pro Projektion	00 : 03
Filtern der Daten pro Projektion	00 : 08
Rückprojektion und gleichzeitiges Schreiben der Daten	10 : 27
Gesamtdauer	11 : 22

Eine deutliche Verbesserung!
Das eigentliche Schreiben der Daten in die Ausgabedateien lässt sich jedoch noch weiter optimieren. Der Schwachpunkt bei diesem Arbeitschritt sind die häufigen Zugriffe auf die, im Vergleich zum Arbeitsspeicher, wesentlich langsamere Festplatte. Es ist daher sinnvoll, zuerst einmal möglichst viele Daten im Ausgabepuffer zu sammeln und dann in großen Blöcken auf die Festplatte zu schreiben. Dies kann erreicht werden, indem nicht wie üblich ein '*endl*', sondern ein einfacher Zeilenvorschub '\n' verwendet wird. Dadurch wird das Leeren des Ausgabepuffers und somit der Festplattenzugriff am Ende jeder einzelnen Zeile verhindert. Statt dessen werden so viele Daten in den Puffer geschrieben, wie es die Speicherverwaltung des Betriebssystems zulässt und dann wird ein entsprechend großer Block auf die Festplatte übertragen. Die folgende Laufzeit ist das Ergebnis dieser Technik:

Operation	Laufzeit (min : sek)
Einlesen der Daten pro Projektion	00 : 03
Filtern der Daten pro Projektion	00 : 08
Rückprojektion und gleichzeitiges Schreiben der Daten	06 : 38
Gesamtdauer	08 : 11

Diese Laufzeit dürfte zufriedenstellend sein, wenn man bedenkt, dass die Berechnung eines solchen Datensatzes mit Matlab etwa 10 Stunden dauert.

Um auch während des späteren Einsatzes des Programms einen Überblick über die Laufzeit (zum Beispiel auf verschiedenen PCs) und den Ablauf der Arbeitsschritte zu behalten, werden mittels der Klasse *KLog* während des Programmdurchlaufs verschiedene Meldungen automatisch in eine Log-Datei eingetragen.

4.9 Verwendete Werkzeuge und Software

- Im Rahmen der Softwareentwicklung wurde "Mac OS X 10.0" und "Xcode" als Entwicklungsumgebung genutzt.

- Für das Kompilieren der Programme unter "Windows XP" kam "Microsoft Visual C++ .NET" zum Einsatz.

- Zur Visualisierung der Daten wurde "Tecplot" verwendet.

- Zum Datenvergleich wurde "Octave" verwendet.

- Die vorliegende Arbeit ist mit "LateX" verfasst worden.

- Weitere nützliche Werkzeuge waren "Omni Graffle" zum Erstellen des Klassendiagramms und "Doxygen" zum Erstellen der Dokumentation der Programme.

- Die verwendeten Bilder wurden mit der CAD- Software "Graphite" konstruiert.

5. Praktische Anwendung

Das im Rahmen dieser Arbeit entwickelte Programm ist hauptsächlich für Mitarbeiter wissenschaftlicher Einrichtungen wie der DLR gedacht. Bisher wurden während der Durchführung eines Versuchs Messdaten aufgenommen, die dann später im Büro ausgewertet werden mussten. Durch die deutlich verkürzte Laufzeit des hier vorgestellten Programms kann die Auswertung nun direkt vor Ort erfolgen. Bereits nach wenigen Minuten stehen Ergebnisse für die Visualisierung zur Verfügung. Die Auswirkungen von Änderungen an einer Messung können sofort dargestellt werden.

Nachdem das Brechungsindexfeld mit dem Programm *filteredBackprojection* errechnet wurde, können die Daten mit dem Programm *Tecplot*, einer Datenvisualisierungssoftware, ausgewertet werden.
Abbildung (5.1) zeigt die zuckerhutartige Dichteverteilung des Freistrahles, der in dieser Arbeit zu Testzwecken verwendet wurde.

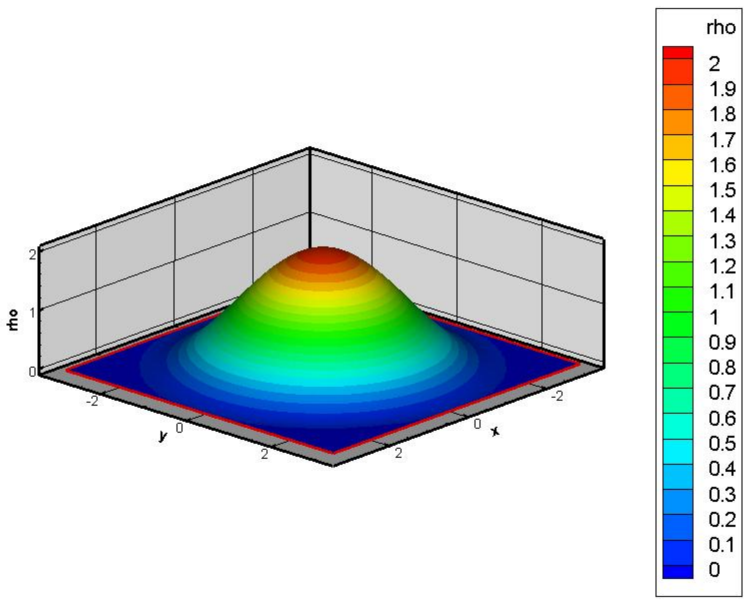


Abbildung 5.1: zuckerhutartige Dichteverteilung eines Freistrahls

Nachfolgend ist ein Vergleich der Ergebnisse des Matlab Programms und des C++ Programms zu sehen. Der Unterschied in der Genauigkeit (siehe Kapitel 4.8.1) ist sichtbar. Man kann im Zentrum des unteren Bildes eine zusätzliche Abstufung erkennen, die auf eine höhere Genauigkeit der Daten zurückzuführen ist.

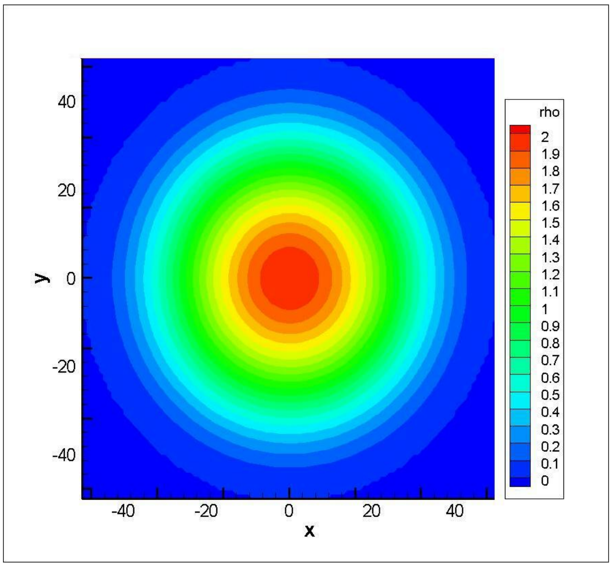


Abbildung 5.2: Ergebnisse der gefilterten Rekonstruktion eines Freistrahls mit dem Matlab Programm

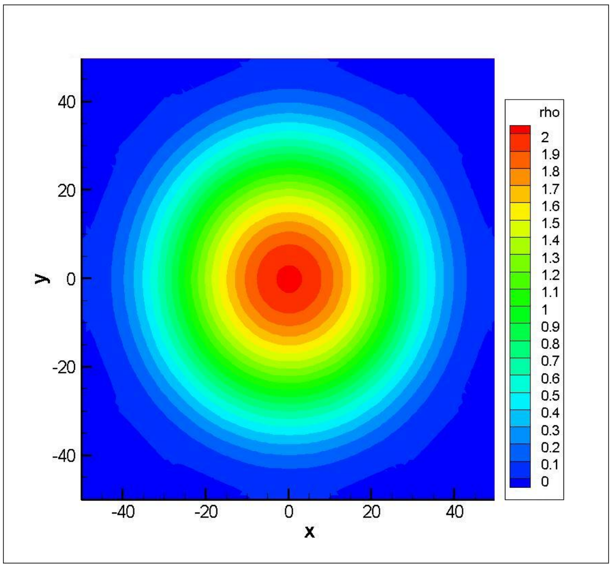


Abbildung 5.3: Ergebnisse der gefilterten Rekonstruktion eines Freistrahls mit dem C++ Programm

Abbildung (5.4) zeigt die Werte der ungefilterten Rekonstruktion des Freistrahls. Die Verschwimmung ist deutlich zu sehen. Außerdem sind die Werte zu hoch, da durch das Weglassen des Filters auch die Wichtung der Werte weggelassen wurde. Die Werte im Zentrum sind daher viel zu stark in das Ergebnis mit eingeflossen.

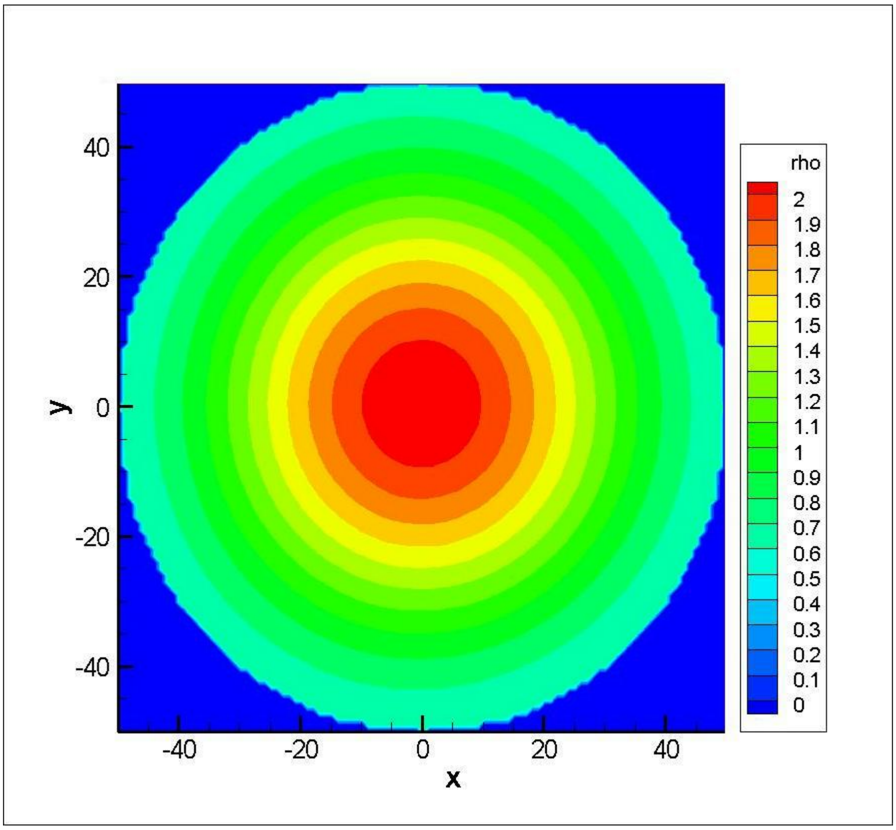


Abbildung 5.4: ungefilterte Rekonstruktion eines Freistrahls

Folgende Abbildungen zeigen die Ergebnisse der gefilterten Rückprojektion mit einer, zwei, drei, vier, sechs und acht Projektionen. Bei einem axialsymmetrischen Untersuchungsfeld wie es bei diesem Freistrahl vorliegt, sind vier Projektionen ausreichend.

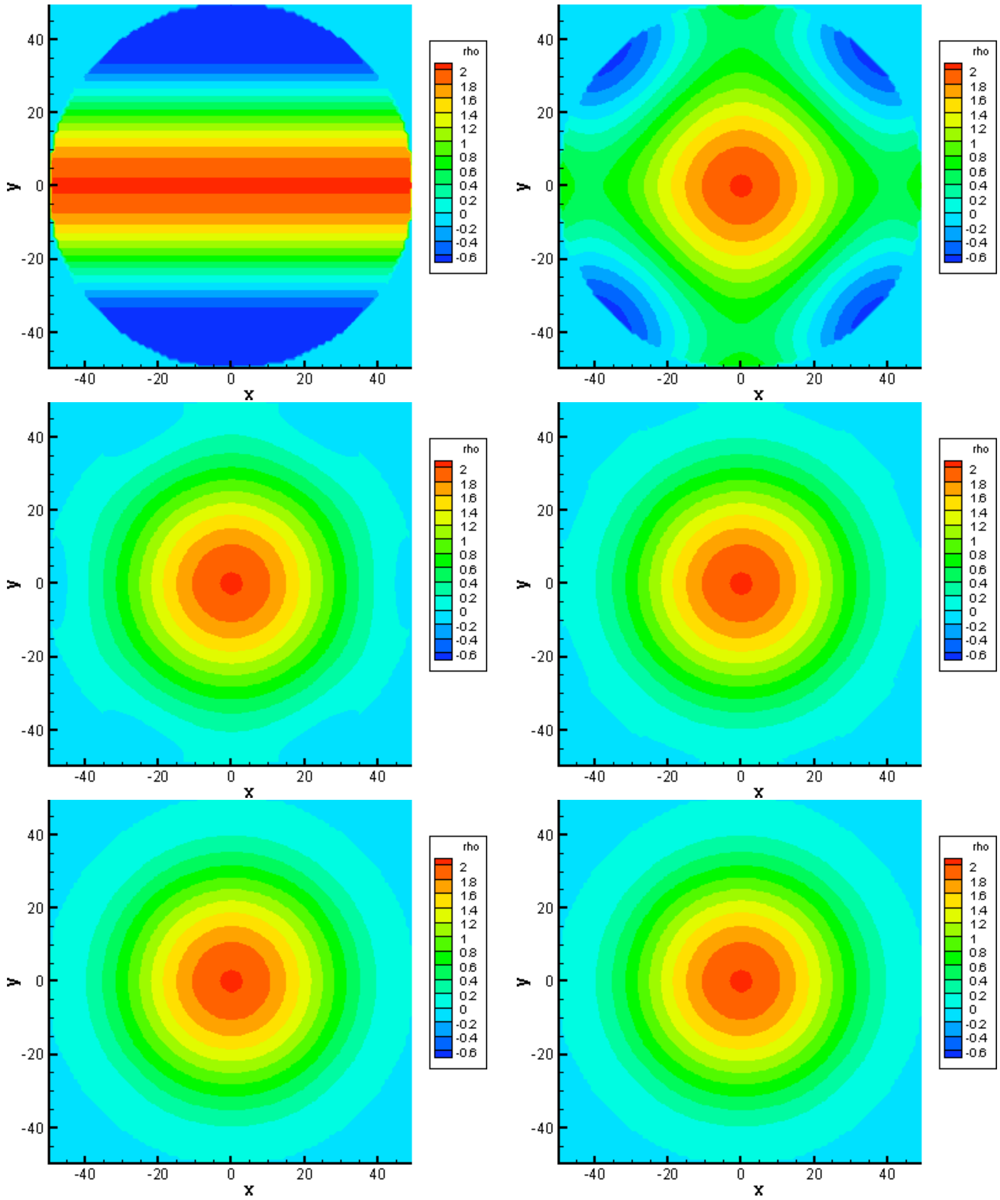


Abbildung 5.5: gefilterte Rückprojektion mit einer, zwei, drei, vier, sechs und acht Projektionen

Auswertungen komplexerer Dichtefelder sehen wie folgt aus. Abbildung (5.6) zeigt die Dichteverteilung in einem unterexpandierten Doppelfreistrahl. Für solche Dichtefelder (Brechungsindexfelder) ist es notwendig, wesentlich mehr als vier Projektionen zu verwenden.

Abschließend soll noch ein konkretes Anwendungsbeispiel vorgestellt werden: In einem Strömungslabor wird ein Objekt platziert. Da es oftmals technisch schwierig ist, Kameras um das Objekt herum aufzustellen, können auch wenige Kameras verwendet werden. Um genügend Projektionen zu erhalten, wird das Objekt dann einfach gedreht, die Kameras bleiben an ihrer Position. Im Beispiel soll ein Doppelfreistrahl untersucht werden. Durch zwei Düsen tritt Luft aus einem Gebläse in die Umgebung aus. Mit bloßem Auge lassen sich möglicherweise Schlieren erkennen, die Kameras nehmen das leicht verschwommene Punktmuster des Hintergrunds exakt auf und speichern es auf einem PC ab. Nachdem der Doppelfreistrahl von mehreren Seiten fotografiert wurde, wird noch das Referenzbild (ohne Strahl) aufgenommen. Nachdem daraufhin für jede Projektion das Verschiebungsvektorfeld erstellt wurde, können mit dem Programm *calculateDeflectionAngle* die Ablenkungswinkel und mit *filteredBackprojection* das Indexfeld berechnet werden. Mit wenigen Mausklicks können die Daten in *Tecplot* angezeigt werden und was bereits als Schlieren zu erkennen war, entpuppt sich nun als laminare, teils turbulente Strömungen und Luftverwirbelungen. Möglicherweise werden nun die Geräte neu justiert und der Versuch wiederholt, vielleicht werden aber auch mehr als ein Gigabyte Daten, die vom Programm errechnet wurden, archiviert und später für eine andere Bachelorarbeit verwendet.

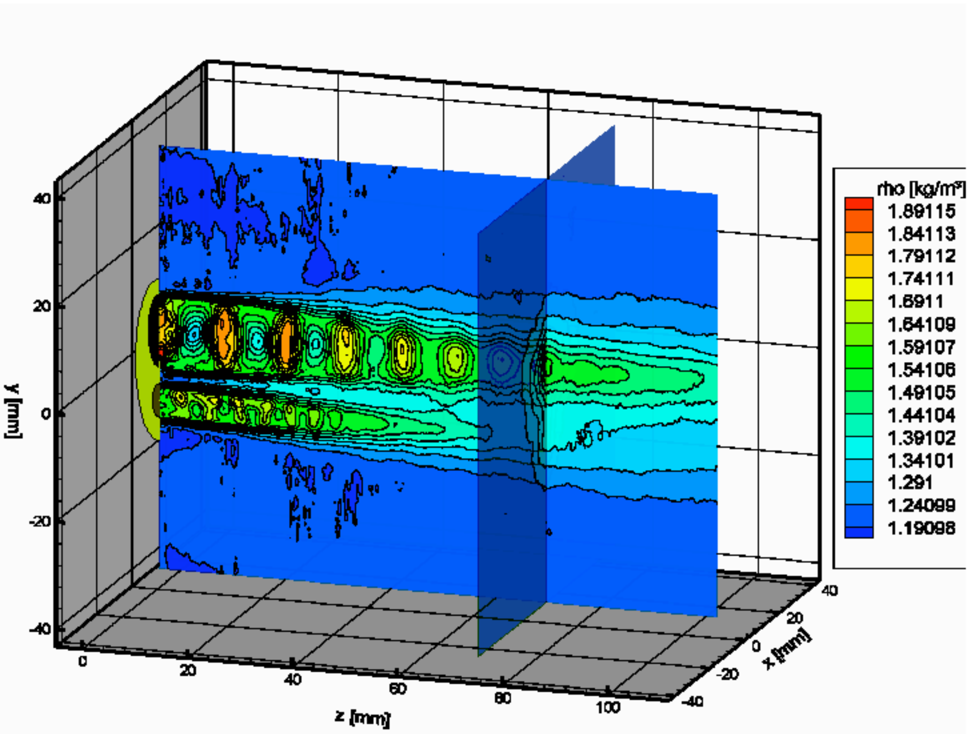


Abbildung 5.6: Mittelschnitt eines Doppelfreistrahls

6. Zusammenfassung und Ausblick

In der vorliegenden Arbeit wird ein Algorithmus zur tomographischen Rekonstruktion von Dichtefeldern aus BOS[1] Messergebnissen implementiert.
Zunächst wird der Zusammenhang zwischen Dichte und Lichtbrechung, also die Grundidee des Hintergrund-Schlieren-Messverfahrens erläutert. Dann wird das Verfahren und die Auswertung der daraus erhaltenen Daten geschildert. Das Vorgehen der gefilterten Rückprojektion wird erläutert und die Herleitung des im Programm verwendeten Algorithmus mathematisch dargestellt.
Der in Matlab implementierte Algorithmus hat eine sehr lange Laufzeit. Die Programmparameter sind fest in den Code eingebunden, was für den allgemeinen Gebrauch eher ungeeignet ist. Ziel der Arbeit ist es daher, den Algorithmus so zu implementieren, dass eine akzeptable Laufzeit und eine flexible Dateneingabe gesichert sind.
Die Implementierung erfolgt in der objektorientierten Sprache C++. Der Algorithmus zur Berechnung eines Brechungsindexfeldes aus Verschiebungsvektorfeldern wird in zwei getrennten Programmen implementiert. Das erste Programm berechnet aus den Verschiebungsvektorfeldern die Ablenkungswinkel und das zweite Programm führt die gefilterte Rückprojektion durch. Beide Programme werden mit dem Datensatz eines Freistrahles getestet. Die Ergebnisse sind korrekt und die Laufzeit angenehm kurz.
Die Laufzeit kann durch eine Optimierung des Algorithmus auf mathematischer Ebene noch weiter verkürzt werden. Ebenso kann dies durch den Gebrauch von Bibliotheken, welche in Assembler programmiert sind, erreicht werden.
Die Wahl der Eingangsdaten ist nun flexibel möglich. Statt einer Änderung des Programmcodes zur Wahl neuer Eingangsdaten, kann einfach ein anderes Verzeichnis als Parameter übergeben werden. Die Implementierung einer graphischen Benutzeroberfläche würde eine noch einfachere Bedienung des Programms ermöglichen. Außerdem kann eine Implementierung der in Kapitel 2.3.2 beschriebenen Kalibrierung angestrebt werden.
Es wird gezeigt, dass die Benutzung eines Filters, der Verschwimmungen unterdrückt, unentbehrlich für die Rekonstruktion von Dichtefeldern (Brechungsindexfeldern) ist. Der im Programm *filteredBackprojection* implementierte *RAMP*-Filter bringt die erwarteten Verbesserungen. Eine weitere Verbesserung durch andere Filter ist allerdings für zukünftige Arbeiten an dem Programm erstrebenswert. Diese Filter sollten Störungen, welche sich durch hohe Frequenzen im Frequenzraum auszeichnen, verringern.

[1] Background Oriented Schlieren

Wie in Kapitel 2 gezeigt wird, ist der Arbeitsaufwand für den Aufbau und die Durchführung des Hintergrund-Schlieren-Verfahrens in Versuchsanlagen relativ gering. Auch die Genauigkeit der Daten, die man durch die gefilterte Rückprojektion erhält, dürfte ausreichend gut sein. Obwohl die auftretende Datenmenge ($>$ 1 GB) sehr groß ist, können durch immer leistungsstärkere Prozessoren und immer größere Speichermedien in Zukunft auch wesentlich umfangreichere Datenmengen verarbeitet werden.

Literatur

[1] K.-H. Ahlheim: Duden Physik, Dudenverlag Vol. 2 (1989)

[2] Wolfgang Merzkirch, Flow Visualization, Academic Press (1974)

[3] T.C. Liu, W. Merzkirch, K. Oberste-Lehn: Optical tomography applied to speckle photographic measurement of asymmetric flows with variable density, Springer-Verlag (1989)

[4] H. Richard, M. Raffel: Principle and application of the background oriented schlieren (BOS) method, Meas. Sci. Technol. 12 pp. 1576-1585 (2001)

[5] Tania Kirmse: Weiterentwicklung des Messsystems BOS (Background Oriented Schlieren) zur quantitativen Bestimmung axialsymmetrischer Dichtefelder (2003)

[6] Dr. H. Chaves: Meßtechnik, http://www.imfd.tu-freiberg.de/lehre/ (2005)

[7] Giovanni Tanda: Application of the schlieren technique to convective heat transfer measurements (1999)

[8] http://gaydon.kokushikan.ac.jp/kagoshima/htm/yano2.htm

[9] http://www.lstm.ruhr-uni-bochum.de/seiten/html/b_gale01.htm

[10] D. Halliday, R. Resnick J. Walker: Fundamentals of Physics, Wiley-VCH Extended 6th Edition (2001)

[11] Experiments in Fluids part 33: N. Fomin, E. Lavinskaya, D.Vitkin: Speckle tomography of turbulent flows with density fluctuations, Springer-Verlag (2002)

[12] www.hochschulstellenmarkt.de

[13] Gabor T. Herman: Image Reconstruction From Projections, The fundamentals of Computerized Tomography, Academic Press (1980)

[14] Gabor T. Herman: Image Reconstruction From Projections, Implementation and Applications, Springer Verlag Berlin Heidelberg New York (1979)

[15] Stefanie Maria Fadgyas: Vergleich der Emissionscomputertomographie (ECT): Gefilterte Rückprojektion versus dreidimensional iterativen Algorithmus (1999)

[16] Prof. Dr.-Ing. J. Hornegger: Medizinische Bildverarbeitung (2005)

[17] Anita Just, Erik Danckwardt, Franz Jacobs: Geoelektrische Tomographie - Prinzip und Anwendungsbeispiele

[18] Werner Backfrieder: Computertomographie, Universität Wien (1999)

[19] Peter Prinz, Ulla Kirch-Prinz: C++ Lernen und professionell anwenden, MITP-Verlag GmbH (1999)

[20] www-groups.dcs.st-and.ac.uk

[21] John J. Benedetto, Ahmed I. Zayed: Sampling, Wavelets, and Tomography, Birkhäuser Verlag (2004)

[22] Waqas Akram, Steve Gee, Charles Gamiz, Christine Pan, Justin Romberg: Image Processing Using SPECT Analysis, http://ipnp00.troja.mff.cuni.cz (1996)

[23] Stefan Horbelt, Michael Liebling, Michael Unser: Filter design for filtered back-projection guided by the interpolation model, http://bigwww.epfl.ch/publications/ (2002)

[24] Ramsey Badawi: Introduction to PET Physics, http://depts.washington.edu/nucmed/IRL/pet_intro/toc.html, (1999)

[25] R.K.Otnes, L.Enochson: Digital Time Series Analysis, (John Wiley, New York 1972)

[26] R.W.Hamming: Digital Filters (Prentice - Hall, Englewood Cliffs, N.J. 1977)

[27] Matthias Bonn: Computer - Tomographie, online-media.uni-marburg.de/ radiologie/bilder/kap2/ct_technik.htm (2000)

[28] Jürgen Braun: Computertomographie, webster.fh-hagenberg.at/staff/wbackfri/ Teaching/MBV/Vorlesung/MBV05ctbasicsBW.pdf

[29] www.wikipedia.de