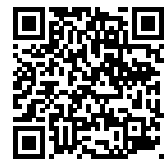
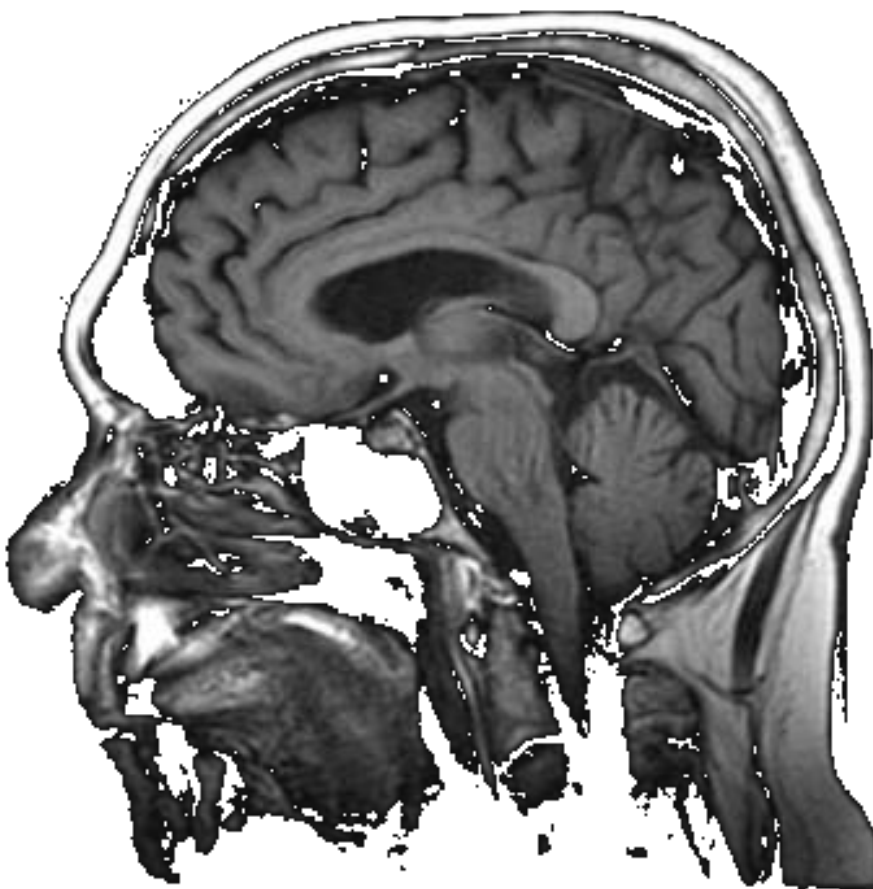


Computertomographie Grundlagen, Simulation und Bildrekonstruktion

(Theorie-Praktikum für Fortgeschrittene)

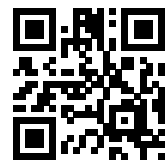
Christian Hoffmann

(Version 0.0400 — 2. August 2023 08:37)



Aktuelle Version

Kontakt



chhof@lusi.uni-sb.de

Weitere Infos



<https://alpha.lusi.uni-sb.de/chhof>



Big Bug Button (?)

Inhaltsverzeichnis

1	Vorbemerkungen, Lernziele und Fragen	3
1.1	Hinweis zum Hinweis zu \LaTeX auf der FoPra-Homepage (\rightarrow Hinweise)	3
1.2	<i>Big Bug Button</i>	4
1.3	Lernziele	4
1.4	Vorkenntnisse	4
1.5	Fragen	4
2	Grundlagen	5
2.1	Tomographie und Radon-Transformation	5
2.2	Sinogramm	5
2.3	Bildrekonstruktion	5
2.4	Diskretisierung	6
2.5	Berechnung von \underline{R}	6
2.6	Veranschaulichung von \underline{R}	8
2.7	Erhöhung der Genauigkeit von \underline{R}	9
2.8	Diskrete Rücktransformation	9
2.9	ART nach <i>Gordon et al.</i>	10
2.10	Visualisierung und Interpretation der Radon-Transformation	11
3	Aufgaben	13
4	Verzeichnisse	16
4.1	Literaturverzeichnis	16
4.2	Attachments	16
4.3	Stichwortverzeichnis	16

1 Vorbemerkungen, Lernziele und Fragen


1.1 Hinweis zum Hinweis zu L^AT_EX auf der FoPra-Homepage (→ Hinweise)

Tatsächlich gibt es 'Kreise', die L^AT_EX bevorzugen. Bezeichnenderweise sind das genau diejenigen, die sich L^AT_EX angeschaut haben, bevor sie darüber urteilen und sich dann einfach für die bessere Alternative entschieden haben. Hier einige Argumente von Nutzern. Mir ist in der Tat kein Vorteil von Word gegenüber L^AT_EX bekannt.

- Die Angst davor, Studenten mit L^AT_EX in Kontakt kommen zu lassen, ist berechtigt: Ich kenne niemanden, der nach dem Wechsel von Word zu L^AT_EX freiwillig wieder zu Word zurückgekehrt ist.
- Die Benutzung von L^AT_EX führt nicht dazu, dass man Word danach nicht mehr verwenden kann. Man kann Studenten also problemlos mit L^AT_EX arbeiten lassen, ohne dass sie die Fähigkeit verlieren, in der 'normalen Arbeitswelt' in Word herumzuklicken.
- Es ist ein weit verbreiteter Mythos, dass in der 'normalen Arbeitswelt' nur Windows verwendet würde. Das gilt vielleicht für Sekretariate und Business-Kasper, aber nicht für die Aufgaben, die ein Physiker ausübt.
- L^AT_EX ist *nicht* nur etwas für Programmierer: Word heißt 'Klicken auf Eigenschafts-Buttons', L^AT_EX heißt 'Hinschreiben der Eigenschaften'. Merken muss man sich die jeweilige Eigenschaft in beiden Fällen.
- In Word arbeitet man zu 70-80% am Aussehen, mit L^AT_EX arbeitet man am Inhalt, L^AT_EX kümmert sich ums Aussehen – und das sogar besser als Word.
- Word ist extrem umständlich bei der Eingabe von Formeln. Es übersetzt sie zwischen verschiedenen Sprachen hin und her, und zwar nicht konvergierend.
- Einbinden von Objekten in Word führt regelmäßig zu einer nur schwer reversiblen Zerstörung des Seitenlayouts. In L^AT_EX kommentiert man Code einfach aus und kompiliert neu.
- Große Dokumente sind schwierig zu handeln in Word. Es gibt Leute, die das Dokument gesplittet und die Seitenzahlen im Inhaltsverzeichnis manuell eingetragen haben.
- Word-Blocksatz lässt u.U. große Wortzwischenräume, weil Microsoft nach 35 Jahren Entwicklung immer noch keine Box-basierte Positionierung hinbekommen hat. Deshalb sehen Word-Texte teilweise schrecklich aus und sind unangenehm zu lesen. Typographie ist sowieso keine Stärke von Word.
- L^AT_EX ist frei und es existieren zahllose Zusatzpakete für nahezu alle Anwendungsbereiche.
- Da ein L^AT_EX-Dokument eigentlich ein L^AT_EX-Programm ist, lassen sich sehr einfach Dokumente automatisch über Shell-Skripte erzeugen.
- Schüler der Mittelstufe erlernen die Grundlagen von L^AT_EX in 30-60 Minuten. Wenn man das Konzept verstanden hat, kann man sehr schnell beliebig komplexe Aufgaben lösen. Viele Aufgaben sind in Word gar nicht erst umsetzbar und wenn, dann nur mit massivem Aufwand, um dann im Zweifel ein typographisch schlechtes Ergebnis zu erhalten.
- Bei den meisten der Möglichkeiten von L^AT_EX ist Word-Nutzern nicht einmal bewusst, dass so etwas überhaupt existiert oder möglich ist (s. nächsten Abschnitt ☺).
- Hier noch ein Zitat von Donald E. Knuth (berühmter Informatiker und Physiker, Erfinder von T_EX und METAFONT): *"If you were young again, would you start writing TeX again or would you use Microsoft Word, or another word processor?"* — *"I hope to die before I have to use Microsoft Word."* — Harald Koenig asking Donald Knuth, Tübingen, 2001 Oct 2.



1.2 *Big Bug Button*

Mit dem *Big Bug Button*  auf jeder Seite unten rechts kommst du direkt auf eine Web-Seite, auf der du einen Fehler melden kannst. Die Position des Fehlers wird automatisch übertragen, kann aber von dir beliebig angepasst werden. Die Angabe eines Kontakts ist freiwillig.

1.3 Lernziele

Das Skript soll anhand des praktischen Beispiels Computertomographie die verschiedenen grundsätzlichen Konzepte und Arbeitsweisen der theoretischen Physik vermitteln.

1. Analyse des physikalischen Problems
2. Mathematische Modellierung
3. Algorithmische Umsetzung
4. Implementierung in einer geeigneten Programmiersprache
5. Fehlersuche und Optimierung
6. Simulation und Variation der Parameter
7. Analyse der Ergebnisse

Im Zentrum steht die Implementierung verschiedener Algorithmen, die sich im wesentlichen auf Matrixoperationen zurückführen lassen. Die Programmierung kann wahlweise in C, C++ oder Python erfolgen. Es werden folgende Teilaufgaben gelöst:

1. Simulation einer computertomographischen Aufnahme (Sinogramm)
2. Rekonstruktion des Urbildes aus dem Sinogramm
3. Variation der Rekonstruktionsverfahren und Parameter
4. Analyse der Urbilder durch Methoden der Bildverarbeitung

1.4 Vorkenntnisse

- Mathematik: Lineare Algebra 1
- Physik: Grundkenntnisse
- Numerik: Grundkenntnisse
- Programmieren: Computerpraktikum/ITG1 bzw. Programmierkenntnisse z.B. in C o.a.

Anmerkung: Das Skript befindet sich im Aufbau und wird nach und nach erweitert.

1.5 Fragen

1. Verfahren zur Erzeugung von Röntgenstrahlung - Energiespektren
2. Absorptionsprozesse von Röntgenstrahlung, Wechselwirkung mit Materie
3. Detektion von Röntgenstrahlung
4. Idealisierte Annahmen über Absorption und Ausbreitung - Modellgrundlagen
5. Herleitung der rechten Seite von Gl. (2.1).
6. Warum kein Gauß-Verfahren zur Bildrekonstruktion?
7. Analytische und numerische Lösungsverfahren für lineare Gleichungssysteme



2 Grundlagen

2.1 Tomographie und Radon-Transformation

Viele Messverfahren liefern integrale Größen der inneren Struktur eines Objekts. Dann besteht das Problem, aus diesen auf die gesuchte innere Struktur zurück zu schließen, z.B. bei der Tomographie (Schnittbildverfahren), bei der ein zweidimensionales Schnittbild durch das Objekt über die orts- und richtungsabhängige Absorption von Röntgenstrahlen (integriert entlang jedes einzelnen Strahls) als integrale Größe entlang des eindimensionalen Randes bestimmt wird. Neben elektromagnetischer (bzw. optischer) Absorption können natürlich auch andere physikalische Messgrößen wie z.B. mechanische Schwingungen (geologische Analysen) etc. verwendet werden.

Mathematisch liegt diesen Verfahren die Radon-Transformation zugrunde. Sei $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ stetig und außerhalb eines Kreises K um den Ursprung identisch Null. Sei $g_{d,\theta} \subset \mathbb{R}^2$ eine beliebige Gerade mit Winkel θ zur x -Achse und Abstand d vom Ursprung, dann heißt das Linienintegral

$$\mathcal{R}[f(x, y)](d, \theta) = \int_{g_{d,\theta}} f(x, y) ds = \int_{-\infty}^{\infty} f(d \cos \theta + t \sin \theta, d \sin \theta - t \cos \theta) dt \quad (2.1)$$

Radon-Transformierte von f entlang der Geraden g .

2.2 Sinogramm

In der Computertomographie werden S parallele Röntgenstrahlen (jeweils mit einer endlichen Breite), welche alle in einer Ebene liegen, durch das Testvolumen (beschrieben durch den ortsabhängigen Absorptionskoeffizienten $f(x, y)$) geführt und auf der gegenüberliegenden Seite von einem Detektor mit S Sensorelementen gemessen. Dadurch entsteht eine eindimensionale Projektion von $f(x, y)$ auf das Detektorarray. Je nach Gesamtabsorption $B_s = \int_{\text{ray}_s} f(x, y) ds$ entlang des s -ten Strahls ray_s ($0 \leq s < S$) wird das s -te Element des Detektors proportional angeregt. Wird dies für alle Winkel $0 \leq \theta < 180^\circ$ wiederholt, wobei der Halbkreis in T Schritten mit $0 \leq t < T$ abgefahren wird, erhält man das sogenannte *Sinogramm* in Abhängigkeit aller $(T \cdot S)$ Strahlengänge

$$B_{t,s} = \mathcal{R}[f(x, y)] \left(d_s, t \cdot \frac{180^\circ}{T} \right), \quad (2.2)$$

wobei d_s der Abstand des s -ten Strahls vom mittleren Strahl darstellt und $0 \leq s < S$ und $0 \leq t < T$ gilt.

2.3 Bildrekonstruktion

Ziel der Bildrekonstruktion ist es nun, aus dem Sinogramm auf die ursprüngliche Dichteverteilung (entspricht der Absorption) $f(x, y)$ zurück zu schließen (*inverse Radon-Transformation*). Da der Detektor und damit auch das Sinogramm eine diskrete Elementzahl besitzen, stehen auch nur endlich viele Gleichungen zur Bestimmung von $f(x, y)$ zur Verfügung, d.h. wir nähern die gesuchte Dichteverteilung auf dem Gitter $0 \leq i, j < N$ durch $f_{i,j} = f(x_i, y_j)$. Um ausreichend viele Gleichungen zu erhalten, sollte das Gleichungssystem überbestimmt sein, also

$$N^2 \lesssim (T \cdot S) \quad (2.3)$$

gelten.



2.4 Diskretisierung

Wir beginnen die Analyse mit der Herleitung der *diskreten* Radon-Transformierten von $f_{i,j} = f(x_j, y_i)$, wobei der Zeilenindex i die y - und der Spaltenindex j die x -Koordinate repräsentieren. Das Objekt, wie auch das rekonstruierte Bild seien quadratisch ($N \times N$ Pixel) und um den Ursprung zentriert. Dies vereinfacht die Berechnung der Projektionen bei Rotation des Objekts um den Winkel θ . Der Einfachheit halber definieren wir die Weltkoordinaten des Zentrums des Pixels (j, i)

$$\begin{aligned} x_j &= j - \lfloor N/2 \rfloor + 0.5 \\ y_i &= i - \lfloor N/2 \rfloor + 0.5. \end{aligned} \quad (2.4)$$

Dadurch beträgt der Pixelabstand immer 1. Wegen $0 \leq i, j < N$ erhalten wir für das Beispiel $N = 100$ folgende Pixelzentrumskordinaten $x_j, y_i = \{-49.5, -48.5, \dots, -0.5, 0.5, \dots, 48.5, 49.5\}$.

Die Radon-Transformation stellt eine lineare Überlagerung der Beiträge $f_{i,j}$ *sämtlicher* Pixel $(j, i) \in \{0, \dots, N-1\}^2$ dar. Es liegt also nahe, die Transformation des Bildes auf das Sinogramm $B_{s,t}$ als lineare Abbildung der Form

$$\underline{\underline{R}} \cdot \underline{F} = \underline{P} \quad (2.5)$$

zu schreiben, wobei \underline{F} und \underline{P} die aus $f_{i,j}$ bzw. $B_{t,s}$ mittels

$$\underline{F}_{i \cdot N + j} := f_{i,j} \quad \text{bzw.} \quad \underline{P}_{t \cdot S + s} := B_{t,s} \quad (2.6)$$

gewonnenen Spaltenvektoren sind. Anschaulich wird also jede Zeile beider Matrizen transponiert und unten an den jeweiligen Spaltenvektor angehängt. Offensichtlich gilt $|\underline{F}| = N^2$ und $|\underline{P}| = T \cdot S$ und damit $\underline{\underline{R}} \in \mathbb{R}^{(T \cdot S) \times N^2}$.

Innerhalb dieser Übung wählen wir $N = 100$. Der gebräuchliche Wert für die Winkelauflösung beträgt $T = 180$, d.h. $\Delta\theta = 1^\circ$. Da der Detektor ein Quadrat der Kantenlänge N für *alle* Drehwinkel vollständig abdecken soll, benötigen wir eine Detektorlänge von $\sqrt{2} \cdot N$ (entspricht der Diagonalen im Objekt bei einem Drehwinkel von $\theta = 45^\circ$ bzw. $\theta = 135^\circ$). Um eine gerade Anzahl an Detektorelementen S zu gewährleisten, was für die Implementierungen einfacher ist, definieren wir:

$$S = 2 \cdot \left\lceil \frac{N}{\sqrt{2}} \right\rceil, \quad (2.7)$$

also hier $S = 142$ für $N = 100$. Diese Wahl erfüllt im übrigen auch die notwendige Bedingung der Gl. (2.3): $T \cdot S = 180 \cdot 142 > 100 \cdot 100 = N^2$. Da im allgemeinen die Hardware (also die Zahl S der Sensoren des Detektorarray und die Winkelauflösung $\Delta\theta$) durch die Mechanik vorgegeben ist, ergibt sich aus Gl. (2.3) die maximal mögliche Auflösung N des rekonstruierten Bildes.

Anmerkung: Eine kleine Abschätzung zeigt die Komplexität des numerischen Problems: Nach Gl. (2.3) erhalten wir z.B. für eine relativ geringe Auflösung des rekonstruierten Bildes von $N = 100$ eine Matrix $\underline{\underline{R}}$ der Mindestgröße 25000×10000 , was klassische Invertierungsverfahren ineffizient macht.

2.5 Berechnung von $\underline{\underline{R}}$

Jedes Element des Vektors $\underline{F} \in \mathbb{R}^{N^2}$ repräsentiert genau einen Pixel des zu rekonstruierenden ($N \times N$)–Bildes. Jedes Element $P_r = P_{t \cdot S + s} = B_{t,s}$ des Vektors \underline{P} repräsentiert den resultierenden Wert des s -ten Strahls beim t -ten Winkel im Sinogramm $\underline{P} \in \mathbb{R}^{T \cdot S}$.

Die r -te Zeile \underline{R}_r der Gewichtsmatrix $\underline{\underline{R}}$ enthält sämtliche relativen Beiträge der Pixel $n = 0 \dots N^2 - 1$ an der r -ten Projektion P_r . Es gilt also $\langle \underline{R}_r, \underline{F} \rangle = P_r$, wobei $\langle \cdot, \cdot \rangle$ das Standardskalarprodukt darstellt.



Das Skalarprodukt gewichtet somit jeden Pixelwert F_n des zu rekonstruierenden Bildes mit $R_{r,n}$. Im einfachsten Fall enthält die r -te Zeile \underline{R}_r eine 1 für jeden Pixel (j, i) , dessen Zentrum in Weltkoordinaten (x_j, y_i) im r -ten Strahlstreifen liegt, ansonsten eine 0 für alle übrigen Pixel. Der r -te Strahlstreifen ist der Strahlstreifen des s -ten Strahls im t -ten Winkel $\theta = t \cdot \Delta\theta$, also $r = t \cdot S + s$. Es gilt somit:

$$R_{r,n} = R_{t \cdot S + s, i \cdot N + j} = \begin{cases} 1, & (x_j, y_i) \in \text{ray}_{t,s} \\ 0, & \text{sonst} \end{cases} \quad (2.8)$$

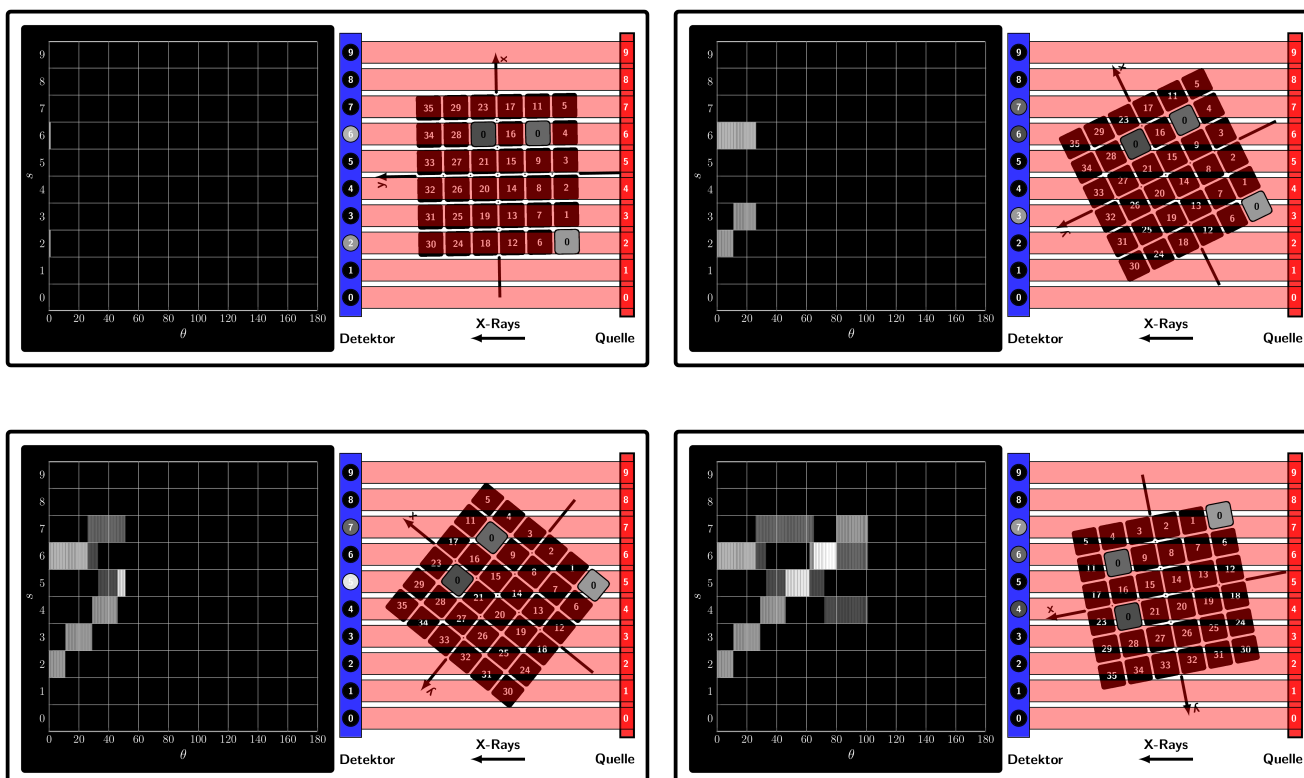
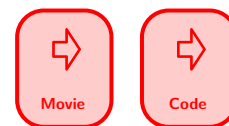
mit $0 \leq t < T$, $0 \leq s < S$ und $0 \leq i, j < N$. Um die Elemente von \underline{R} zu erhalten, bestimmen wir für jeden Winkelindex $t = 0 \dots T - 1$ und für jeden Pixel (j, i) mit $0 \leq i, j < N$ zunächst dessen Position $(x_j, y_i) \in \mathbb{R}^2$ mittels Gl. (2.4) in Weltkoordinaten und rotieren diese mittels Rotationsmatrix um den Winkel $\theta = t \cdot \Delta\theta$

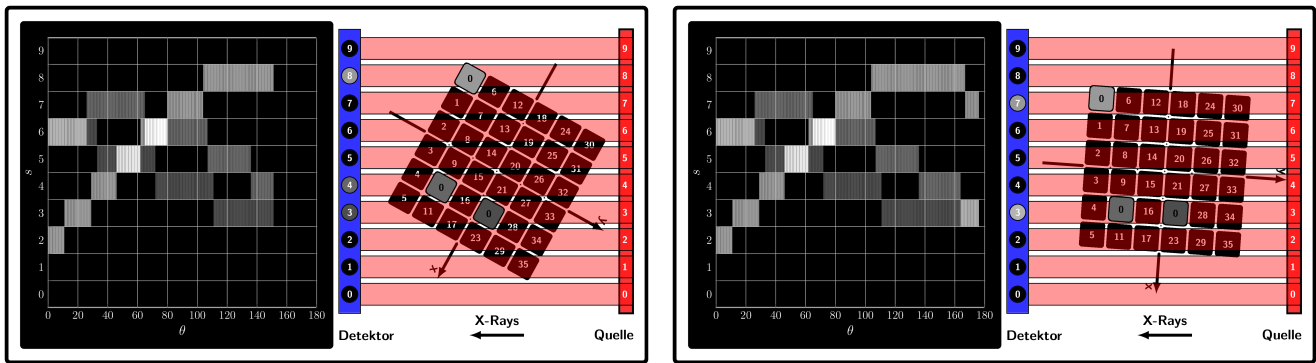
$$\begin{pmatrix} x'_j \\ y'_i \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_j \\ y_i \end{pmatrix} = \begin{pmatrix} x_j \cos \theta - y_i \sin \theta \\ x_j \sin \theta + y_i \cos \theta \end{pmatrix}. \quad (2.9)$$

Die Projektion $s^\perp \in \mathbb{R}$ auf das Detektorarray entspricht nun gerade der ersten Komponente $x'_j = x_j \cos \theta - y_i \sin \theta$, so dass wir den gesuchten Index s durch Abrunden und Verschieben in den positiven Bereich ($s = \lfloor s^\perp \rfloor + S/2$) erhalten:

$$s = \lfloor x_j \cos \theta - y_i \sin \theta \rfloor + S/2. \quad (2.10)$$

Weil sich S nach Gl. (2.7) berechnet und demnach eine gerade Zahl ist, ist $S/2$ ganzzahlig. Mit dem so erhaltenen Quadrupel $[i, j, t, s]$ setzen wir das entsprechende Element von \underline{R} auf Eins: $R_{r,n} = R_{t \cdot S + s, i \cdot N + j} = 1$. Vergleiche hierzu die Snapshot-Folge mit $N = 6$ und $S = 10$ (vollständiges Video inkl. Source-Code als Attachment).



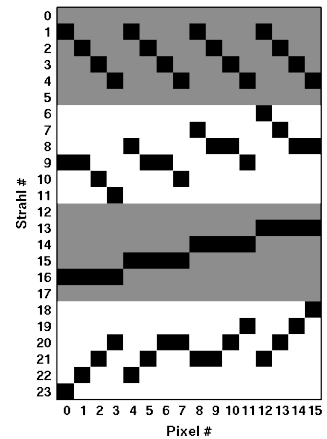


2.6 Veranschaulichung von \underline{R}

Zum besseren Verständnis der Bedeutung der Radon-Transformationsmatrix \underline{R} betrachten wir ein Beispiel mit $N = 4$ (das rekonstruierte Bild hat dann eine Auflösung von 4×4 Pixel) und $T = 4$ verschiedenen Winkeln zwischen 0° und 180° , also $\theta = 0^\circ, 45^\circ, 90^\circ, 135^\circ$.

Nach Gl. (2.7) erhalten wir $S = 6$ und wegen Gl. (2.5) ergibt sich die Länge des Pixelvektors $|\underline{F}| = 16$ und die Dimension der Radon-Transformationsmatrix \underline{R} zu $(T \cdot S) \times (N \cdot N) = (4 \cdot 6) \times (4 \cdot 4) = 24 \times 16$.

Die nach den Gln. (2.8), (2.9) und (2.10) erhaltene Transformationsmatrix besteht aus Nullen und Einsen und ist in der Abbildung rechts gezeigt, wobei die Einsen als schwarze Quadrate dargestellt sind.



Jede der 24 Zeilen repräsentiert einen Gewichtsvektor für jeden der 16 Pixel. Die Zeilen sind in 4 Blöcke ($T = 4$) für die vier Einstrahlwinkel (gekennzeichnet durch die graue Unterlegung) zu je 6 Zeilen ($S = 6$) unterteilt. Es gilt wie bereits in Gl. (2.9) für die Zeile r in der Matrix die Beziehung $r = t \cdot S + s$. Jede Zeile entspricht somit einem einzelnen Röntgenstrahl (endlicher Breite) definiert durch das Detektorelement s und den Winkelindex t .

Zum Verständnis der Belegung von \underline{R} beginnen wir für $(t, s) = (0, 0)$, also mit dem ersten Röntgenstrahl ($r = 0$) für $\theta = 0^\circ$ und dem ersten Detektorelement. Da der Detektor breiter ist als die Kantenlänge N befinden sich keine Pixel im Strahlengang. Der zweite Strahl $r = 1$ für das Detektorelement $s = 1$ und dem Winkelindex $t = 0$ streift eine Kante des Objekts mit 4 Pixeln, nämlich Pixel #0, #4, #8 und #12. Der dritte Strahl $r = 2$ mit $s = 2$ und $t = 0$ streift die benachbarte Kante des Objekts mit den Pixeln #1, #5, #9 und #13. Analog ergeben sich die Einträge für die restlichen 4 Strahlen des Winkels $\theta = 0^\circ$.

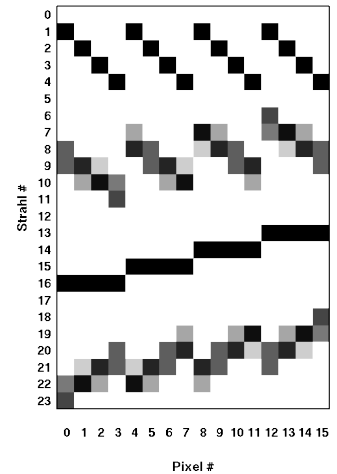
Der zweite Block der Matrix repräsentiert den Winkel $\theta = 45^\circ$ mit dem Index $t = 1$. Die $S = 6$ zugehörigen Strahlen $r = 6 \dots 11$ streifen das um 45° gegen den Uhrzeigersinn gedrehte Objekt, so dass die Randstrahlen $r = 6$ und $r = 11$, die den Randelementen $s = 0$ und $s = 5$ des Detektors entsprechen, jeweils nur einen (Eck-)Pixel #12 für $s = 0$ bzw. #3 für $s = 5$ streifen. Die dazwischen liegenden Strahlen streifen entsprechend mehrere Pixel, jeweils zwei für $s = 1$ und $s = 4$ und jeweils drei für $s = 2$ und $s = 3$.

Beim nächsten Winkel $\theta = 90^\circ$ ($t = 2$) streifen die Randstrahlen $s = 0$ und $s = 5$ wiederum keine Pixel und die innen liegenden streifen jeweils eine Zeile des Objekts, also die Pixel #12, #13, #14 und #15 für $s = 1$, #8, #9, #10 und #11 für $s = 2$ usw. Vergleiche dazu auch die oben gezeigten Snapshots bzw. das Video.

2.7 Erhöhung der Genauigkeit von \underline{R}

Gl. (2.8) beschreibt ein sehr simples Verfahren zur Bestimmung der Gewichtsmatrix \underline{R} . Eine genauere Zuordnung der Pixel berücksichtigt die Lage des Zentrums (x_j, y_i) . Dazu definieren wir dessen relative Position `float frac` innerhalb einer Zelle gegeben durch $\text{frac} := x^\perp - s$. Wir können dann folgende Fälle unterscheiden:

- $\text{frac} = 0.5$: Das Pixelzentrum befindet sich genau in der Mitte der Zelle, so dass diese den Wert 1 erhält.
- $\text{frac} < 0.5$: Das Pixelzentrum befindet sich links von der Mitte der Zelle, so dass ein Teil des Pixels in die linke Nachbarzelle ragt. In diesem Fall wird der Wert der Zelle um $0.5 - \text{frac}$ und der der linken Nachbarzelle um $0.5 + \text{frac}$ erhöht.
- $\text{frac} > 0.5$: Das Pixelzentrum befindet sich rechts von der Mitte der Zelle, so dass ein Teil des Pixels in die rechte Nachbarzelle ragt. In diesem Fall wird der Wert der Zelle um $1.5 - \text{frac}$ und der der rechten Nachbarzelle um $-0.5 + \text{frac}$ erhöht.



Die nebenstehende Abbildung enthält die graukodierten Einträge der Matrix \underline{R} . Für die Winkel $\theta = 0^\circ$ und $\theta = 90^\circ$ gibt es keine Unterschiede zur ersten Variante. Jedoch für $\theta = 45^\circ$ und $\theta = 135^\circ$ überdecken einige der Pixel zwei benachbarte Strahlstreifen.

Intuitiv sollte diese Gewichtsmatrix zu einem besseren Ergebnis der u.a. Rücktransformation führen. In Testmessungen hat sich aber eine Verschlechterung der Konvergenz gezeigt.

2.8 Diskrete Rücktransformation

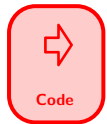
Die Radon-Transformationsmatrix \underline{R} hängt nur von der gewählten Pixelauflösung N und der gewünschten Winkelauflösung T ab, wobei Gln. (2.3) und (2.7) zu berücksichtigen sind. Sie kann also für gegebenes $N = 100$ und $T = 180$ vorab berechnet werden und liefert gemäß Gl. (2.5) für verschiedene Objektpixelvektoren \underline{F} das zugehörige Sinogramm \underline{P} .

Da in der Praxis \underline{F} unbekannt und gesucht ist, \underline{R} berechnet und \underline{P} gemessen werden kann, stellt sich die Frage der Lösung des Gleichungssystems (2.5) mit der schwachbesetzten Matrix \underline{R} :

$$\underline{R} \cdot \underline{F} = \underline{P} \quad (2.11)$$

Es handelt sich um ein überbestimmtes und *schlecht gestelltes Problem*, was eine exakte Lösung erheblich erschwert. In der Praxis stehen eine Vielzahl numerischer Verfahren zur Verfügung, von denen vorallem in der Medizintechnik die sogenannte *Filtered Back-Projection (FBP)* eingesetzt wird. Die dazu notwendige Mathematik ist überschaubar, jedoch setzt sie Kenntnisse über die Fourier-Transformation voraus. Wir wählen daher hier einen bedeutend einfacher zu implementierenden Algorithmus aus der Gruppe der *Algebraic Reconstruction Techniques (ART)*.

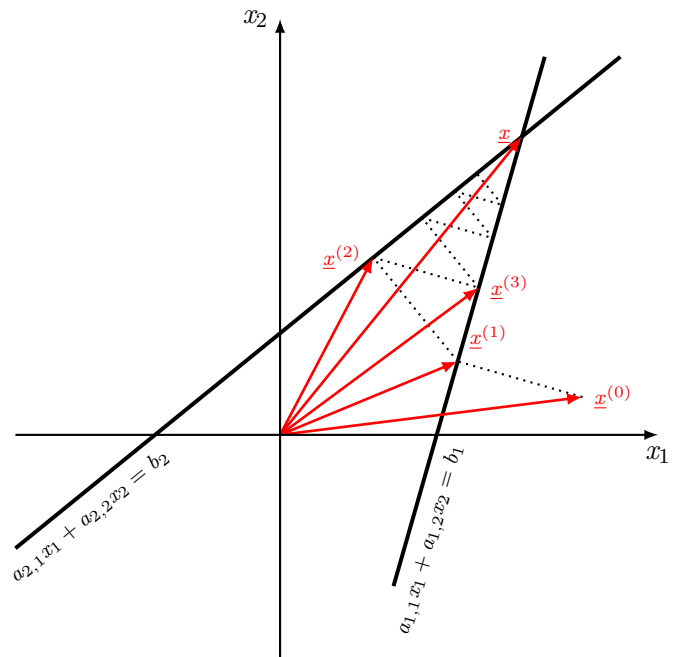
Das Iterationsverfahren nach Kaczmarz, wie es auch in Numerik-Vorlesungen behandelt wird, löst das Gleichungssystem (2.11) durch sukzessive Projektion einer Näherungslösung auf die durch jeweils eine der Einzelgleichungen definierten Hyperebenen. Wir machen dies an einem kleinen Beispiel deutlich.



Gegeben sei das Gleichungssystem $\underline{A} \cdot \underline{x} = \underline{b}$ mit $\underline{A} \in \mathbb{R}^{2 \times 2}$ und $\underline{x}, \underline{b} \in \mathbb{R}^2$, wobei \underline{A} und \underline{b} gegeben und \underline{x} gesucht sind. Das Gleichungssystem lässt sich ausschreiben:

$$\begin{aligned} (0) \quad & a_{1,1}x_1 + a_{1,2}x_2 = b_1 \\ (1) \quad & a_{2,1}x_1 + a_{2,2}x_2 = b_2. \end{aligned}$$

Jede der Gleichungen (0) und (1) stellt einen 1-dimensionalen Unterraum (Gerade) des \mathbb{R}^2 dar (im allgemeinen eine $(n - 1)$ -dimensionale Hyperebene des \mathbb{R}^n). Die gesuchte Lösung $\underline{x} = (x_1, x_2)^T$ stellt den Schnittpunkt beider Geraden dar (s. nebenstehende Abbildung). Wir nähern diese Lösung iterativ durch die Folge $\underline{x}^{(k+1)} = f_m(\underline{x}^{(k)})$ mit einem beliebigen Startpunkt $\underline{x}^{(0)}$, wobei $f_m(\underline{x}^{(k)})$ die senkrechte Projektion von $\underline{x}^{(k)}$ auf die m -te Hyperebene darstellt. Dabei gilt hier $m = k \bmod 2$, da nur zwei Gleichungen vorhanden sind. Im Falle von n Gleichungen gilt natürlich $m = k \bmod n$



2.9 ART nach Gordon et al.

Die allgemeine Iterationsvorschrift für die Projektionsfolge $\underline{F}^{(k)}$ zur Rekonstruktion des Bildes \underline{F} leitet sich aus der Kaczmarz Methode ab und wurde 1971 von Gordon et al. in [1] zum ersten Mal beschrieben und in [2] Kap. 7 sehr anschaulich und verständlich dargestellt. Eine einfache geometrische Herleitung ergibt demnach für $\underline{F}^{(k)}$

$$\underline{F}^{(k+1)} = \underline{F}^{(k)} + \frac{P_r - \langle \underline{R}_r, \underline{F}^{(k)} \rangle}{\| \underline{R}_r \|^2} \underline{R}_r =: \underline{F}^{(k)} + \frac{P_r - q_r^{(k)}}{L_r} \underline{R}_r, \quad \underline{F}^{(0)} \text{ bel., } r = k \bmod (T \cdot S) \quad (2.12)$$

für den k -ten Iterationsschritt. Dabei bedeuten \underline{R}_r die r -te Zeile von \underline{R} und P_r das r -te Element von \underline{P} . Das bedeutet, die Iteration beginnt nach einem vollständigen Durchlauf aller Zeilen wieder von vorne, solange bis die Näherung gut genug ist.

Gl. (2.12) liefert eine Folge von rekonstruierten Bildpixelvektoren $\underline{F}^{(k)}$, deren Grenzwert der gesuchten Lösung \underline{F} entspricht. Zur bequemeren Implementierung schreiben wir diese Vektorgleichung in Komponenten, d.h. für jeden Pixel F_n und erhalten nach einfacher Umformung

$$\begin{aligned} F_n^{(k+1)} &= F_n^{(k)} + \frac{P_r - q_r^{(k)}}{L_r} R_{r,n}, \quad 0 \leq n < N^2 \\ q_r^{(k)} &:= \langle \underline{R}_r, \underline{F}^{(k)} \rangle = \sum_{n=0}^{N^2-1} R_{r,n} F_n^{(k)} \\ L_r &:= \sum_{n=0}^{N^2-1} R_{r,n}^2, \quad r = k \bmod (T \cdot S) \end{aligned} \quad (2.13)$$

Die Betragsquadrate L_r der Zeilen \underline{R}_r ($0 \leq r < T \cdot S$) von \underline{R} sind (wie \underline{R}) unabhängig von der Iteration und können daher vorab berechnet werden. Gemäß Gl. (2.13) wird in jedem einzelnen Iterationsschritt $k \rightarrow k + 1$ eine Schleife über sämtliche Pixel $0 \leq n < N^2$ durchgeführt. In der Implementierung gibt es also zwei Arrays `float Fprev[N*N]` und `float Fnext[N*N]`.

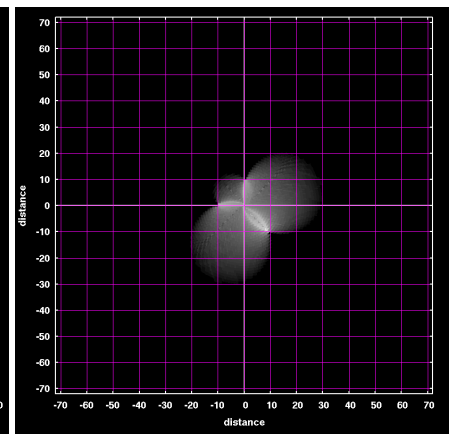
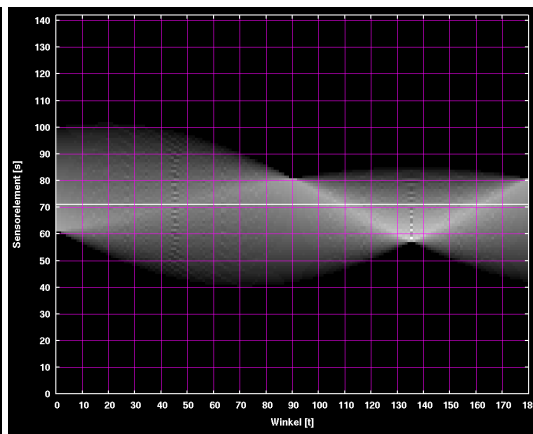
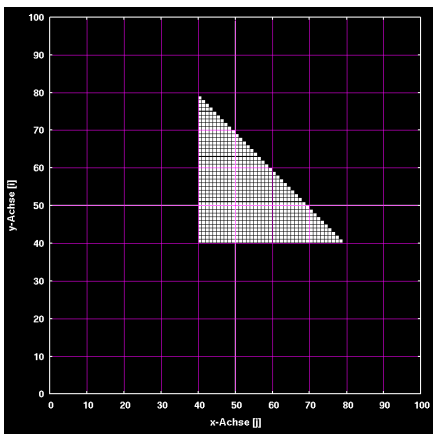
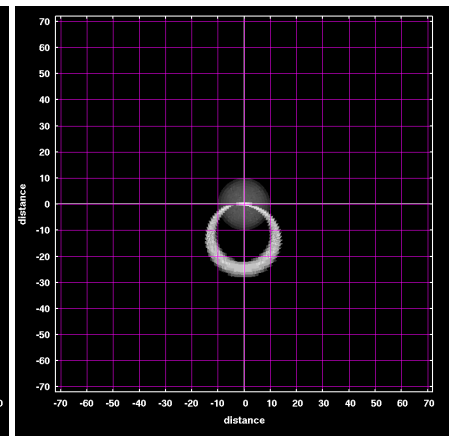
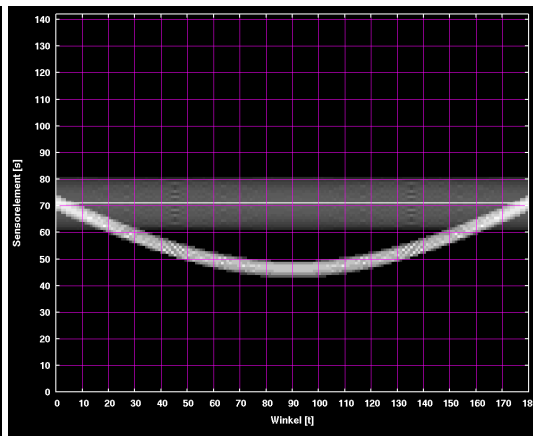
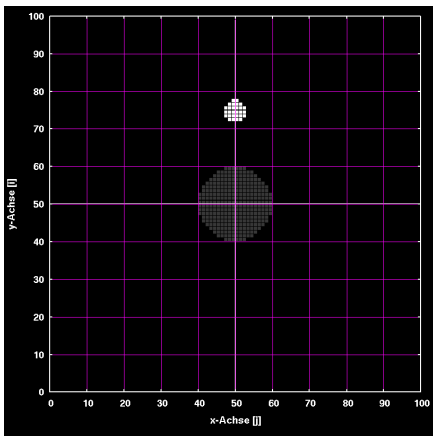


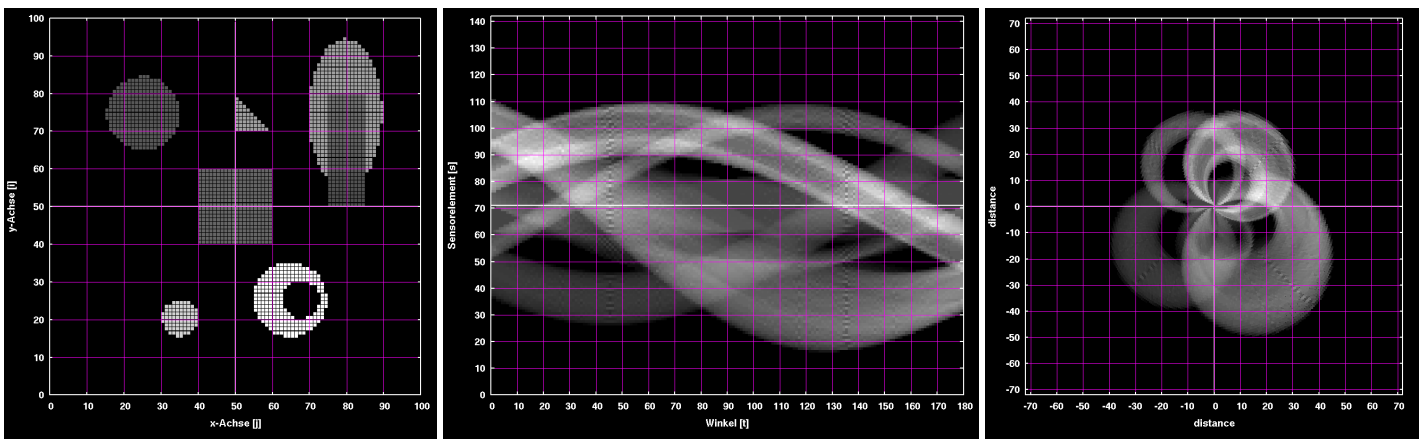
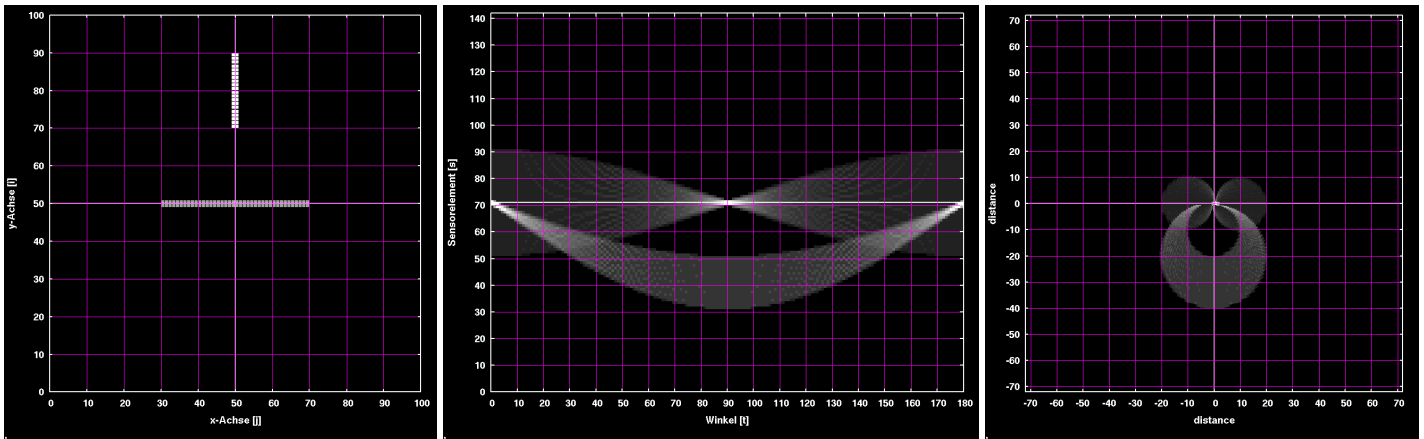
2.10 Visualisierung und Interpretation der Radon-Transformation

Zur Veranschaulichung und zum tieferen Verständnis der bei der Tomographie auftretenden Projektionen (Sinogramm, Radon-Transformation) betrachten wir einige einfache Originalbilder (Weltkoordinaten $[-0.5, 0.5]^2$ skaliert auf die Arraykoordinaten $[0..99]^2$) und den zugehörigen Sinogrammen, sowohl in kartesischen Koordinaten in der (t, s) -Ebene (d.h. Abszisse = Winkelkoordinate, Ordinate = Detektorelement) als auch in Polarkoordinaten.

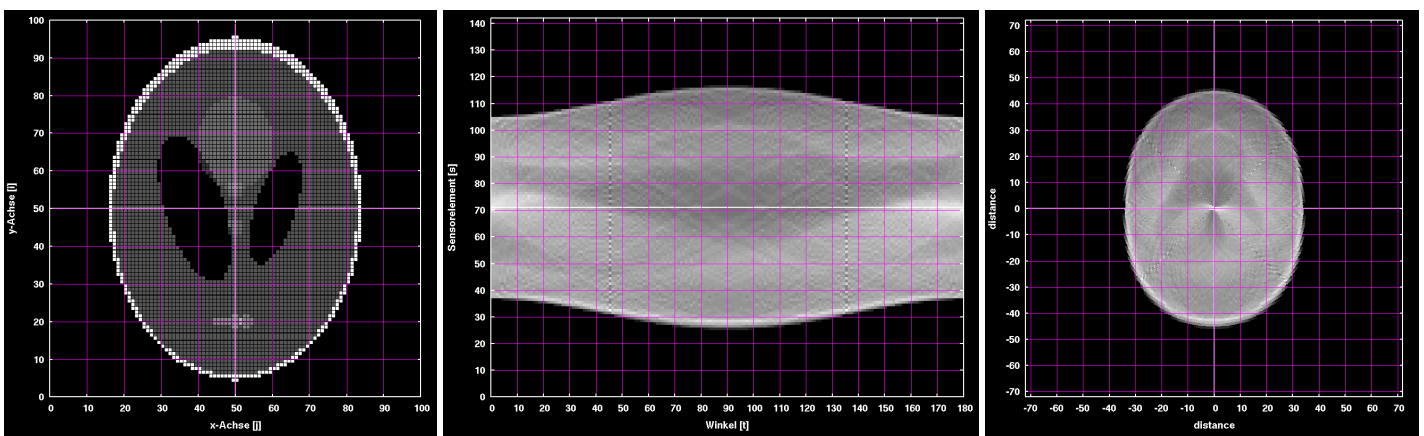
Wir beginnen mit zwei Kreisen, einem großen grauen Kreis mit Mittelpunkt im Ursprung (um den auch das Objekt rotiert wird) und einem zweiten, kleineren und hellerem Kreis auf der y -Achse (linkes Bild). Das Sinogramm (mittleres Bild) lässt sich leicht verstehen, wenn man sich vorstellt, dass das Objekt im Tomographen (Strahlengang senkrecht nach oben) um den Ursprung gegen den Uhrzeigersinn um den Winkel $\theta = f \cdot 180^\circ / T$ gedreht wird.

Da der Strahlengang für Winkel $\theta = 180^\circ + \alpha$ dem für $\theta = \alpha$ entspricht, wird natürlich nur ein Halbkreis gescannt. Da wir aber auch die Detektorelemente umnummerieren $-S/2 < s' < S/2$, kann das Sinogramm auch in einem 360° Polardiagramm dargestellt werden. Dort erkennt man direkt die Periodizität. Versuchen Sie auch die folgenden Tomographien nachzuvollziehen.

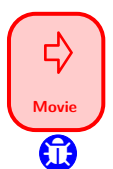


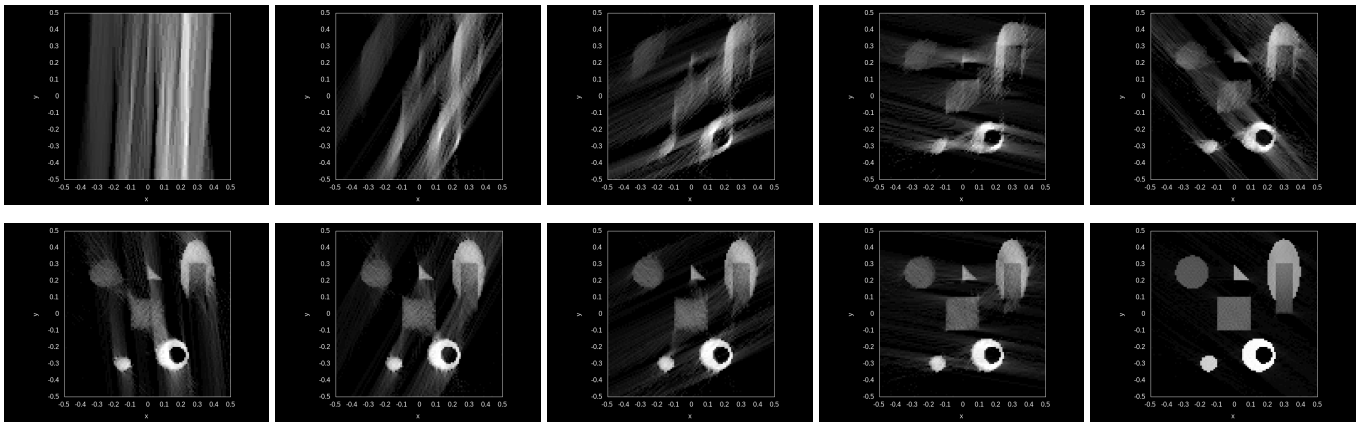


Als letztes Beispiel benutzen wir ein professionelles Testmuster (*Phantom*) von Shepp und Logan, welches sie 1974 in einem Paper über die Rekonstruktion von Tomographieaufnahmen vorgeschlagen haben und als klassisches Testbild verwendet wird. Es besteht aus 10 Ellipsen unterschiedlicher Graustufen, die einen Schnitt durch einen Schädel modellieren sollen. Die Graustufen sind so gewählt, dass sie aufgrund eines zu geringen Kontrastverhältnisses in der Rekonstruktion schlecht erkennbar sein könnten. Man kann also den implementierten Algorithmus so anpassen, dass bei Phantom alle Aneurysmen, Zysten, Thromben, Tumore etc. erkennbar werden.

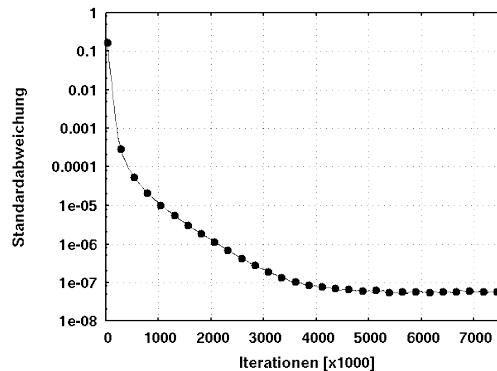


Der Iterationsprozess führt zu einer sukzessiven Rekonstruktion des ursprünglichen Objekts. Die folgenden Snapshots zeigen die Stadien nach jeweils 5000 Iterationen. Die Animation (Attachment rechts) zeigt den kompletten Ablauf.





Betrachtet man die mittlere Abweichung zwischen Original- und rekonstruiertem Bild in Abhängigkeit von der Anzahl der Iterationen, erhalten wir folgende Ergebnisse (s. Abb.: Dargestellt sind erheblich weniger Symbole als Messwerte). Man erkennt, dass die Standardabweichung schon nach einigen wenigen 1000 Iterationen unter 1% fällt (die Grauwerte des Bildes liegen zwischen 0 und 1). Zur Komplexität: 100000 Iterationen benötigen auf einer durchschnittlichen i7-CPU etwa 1.5s.



3 Aufgaben

1. Schreiben Sie eine Funktion, die den Bildvektor `float F[N*N]` z.B. mit `#define N 100` aus einer Datei einliest und ggfs. eine Normierung von `{0..155}` auf das Intervall `[0; 1]` durchführt.

Erstellen Sie eine weitere Funktion `save_vector(float* V, int rows, int cols)`, die einen Vektor `v` im Format `i j v[i*cols + j]` in eine Datei abspeichert. `i` numeriert die `rows` Zeilen, `j` die `cols` Spalten der dem Vektor `v` zugrundeliegende `(rows × cols)`-Matrix.

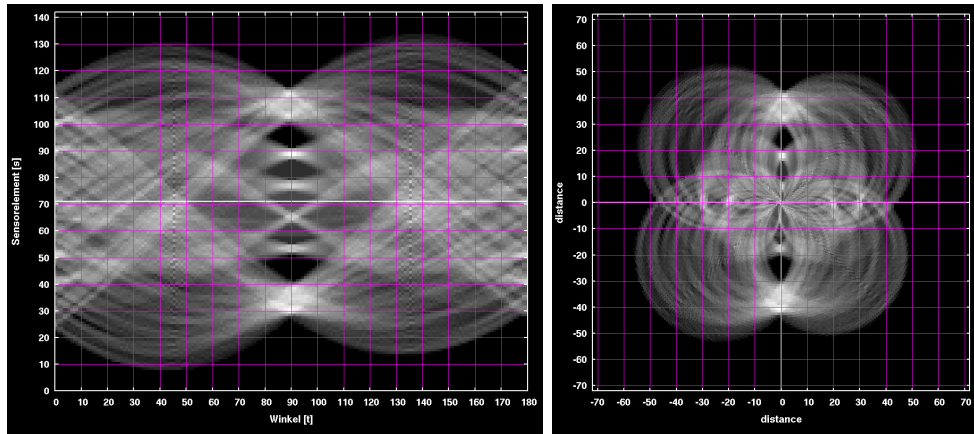
2. Schreiben Sie eine Funktion, welche die Radon-Transformationsmatrix `float R[S*T][N*N]` und die Betragsquadrate `float LR[S*T]` ihrer Zeilenvektoren als globale Variablen berechnet. Bestimmen Sie `s` gem. Gl. (2.3) und wählen Sie z.B. `#define T 180`.

3. Erstellen Sie eine Funktion, die das Sinogramm `float P[S*T]` eines eingelesenen Bildes `F[N*N]` nach Gl. (2.11) erstellt. Das Bild und sein Sinogramm können mittels der Funktionen aus Teil 1 eingelesen bzw. abgespeichert werden. Dateien im oben beschriebenen Format können z.B. mit `gnuplot` dargestellt werden. Tipp: `plot 'sinogram.dat' us 1:2:3 w p palette ps 1 pt 7` erzeugt ein farbkodiertes Bild der eingelesenen Daten. Verwenden Sie nebenstehende Daten eines Testbildes.



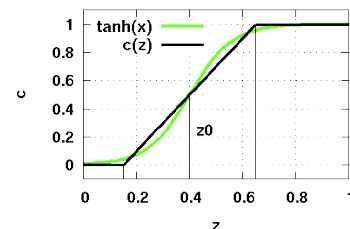
4. Implementieren Sie die diskrete Rücktransformation nach Gl. (2.13) und testen Sie sie an dem im vorangegangenen Teil berechneten Sinogramm. Anmerkung: In der Regel genügen einige wenige vollständige Durchläufe von $(S \cdot T)$ Iterationen, um eine sehr gute Qualität zu erhalten.

5. Die Rekonstruktion kann erheblich beschleunigt werden, wenn man während der Iteration Kenntnisse über Eigenschaften des zu rekonstruierenden Bildes einfließen lässt. In den hier verwendeten Originalbildern liegt der Grauwert stets zwischen 0 und 1. Wird also ein Pixel während der Iteration > 1 bzw. < 0 sein, setzen wir ihn auf 1 bzw. 0.
6. Als kleinen Test für Ihr Programm sollten Sie versuchen, das Originalbild aus folgendem Sinogramm zu rekonstruieren. Das Datenfile ist im Format `t s P[t][s]`. Viel Spaß...



7. **Kontrasterhöhung:** Unabhängig vom Rekonstruktionsverfahren kann ganz allgemein zur Verbesserung der Erkennbarkeit von Strukturen in Graustufenbildern der Kontrast erhöht werden. Im einfachsten Fall wird dem Funktionswert $z \in [0; 1]$ eines Punktes ein Grauwert $c \in [0; 1]$ gemäß $c(z) = z$ zugeordnet. Soll der Kontrast innerhalb einem Teilintervall aus $I \subset [0; 1]$ erhöht werden, bietet sich ein 'Strecken' der Grauwertabbildung $c(z)$ in diesem Intervall an. Sigmoidale Funktionen wie ein (skaliertes und verschobenes) $\tanh(x)$ oder die abschnittsweise definierte Funktion

$$c(z) = \begin{cases} 0, & z < z_0 - \frac{1}{2m} \\ m(z - z_0) + 1/2, & z_0 - \frac{1}{2m} \leq z \leq z_0 + \frac{1}{2m} \\ 1, & z > z_0 + \frac{1}{2m} \end{cases}$$



(3.1)

erfüllen dies. Dabei ist z_0 die Mitte von I und $m \in \mathbb{R}^+$ die Steigung von $c(z)$ innerhalb von I , so dass $I = [z_0 - \frac{1}{2m}; z_0 + \frac{1}{2m}]$ gilt. Anschaulich ändert sich dadurch der dargestellte Grauwert $c(z)$ in einem kleinen Intervall viel stärker als der darzustellende Funktionswert z , so dass der (lokale) Kontrast erhöht wird. Die folgende Abbildung zeigt an einem realen Bild (links oben) den Effekt der Kontrastverstärkung um verschiedene Intervallmitten z_0 (jeweils angegeben in der blauen Box) und einer Steigung von $m = 2$.

Der zur Erzeugung des nachfolgenden Bildes benutzte `gnuplot`-Code lautet

```
1 # Multiplot-Example (verkürzt)
2 N = 200 # Input-Bild Kantenlänge
3 ROWS = 5 # Multiplot rows x cols
4 COLS = 6
5
6 set term x11 nopersist size 200*COLS,200*ROWS
7 set size noratio
8 set font "-adobe-helvetica-bold-r-normal-*-20-*-*-*-*-*"
9 unset colorbox; unset key; unset xlabel; unset ylabel; unset title;
10 set grid front
11 set xtics 50
12 set ytics 50
13 set format x "" # grid w/o tick labels
14 set format y ""
```

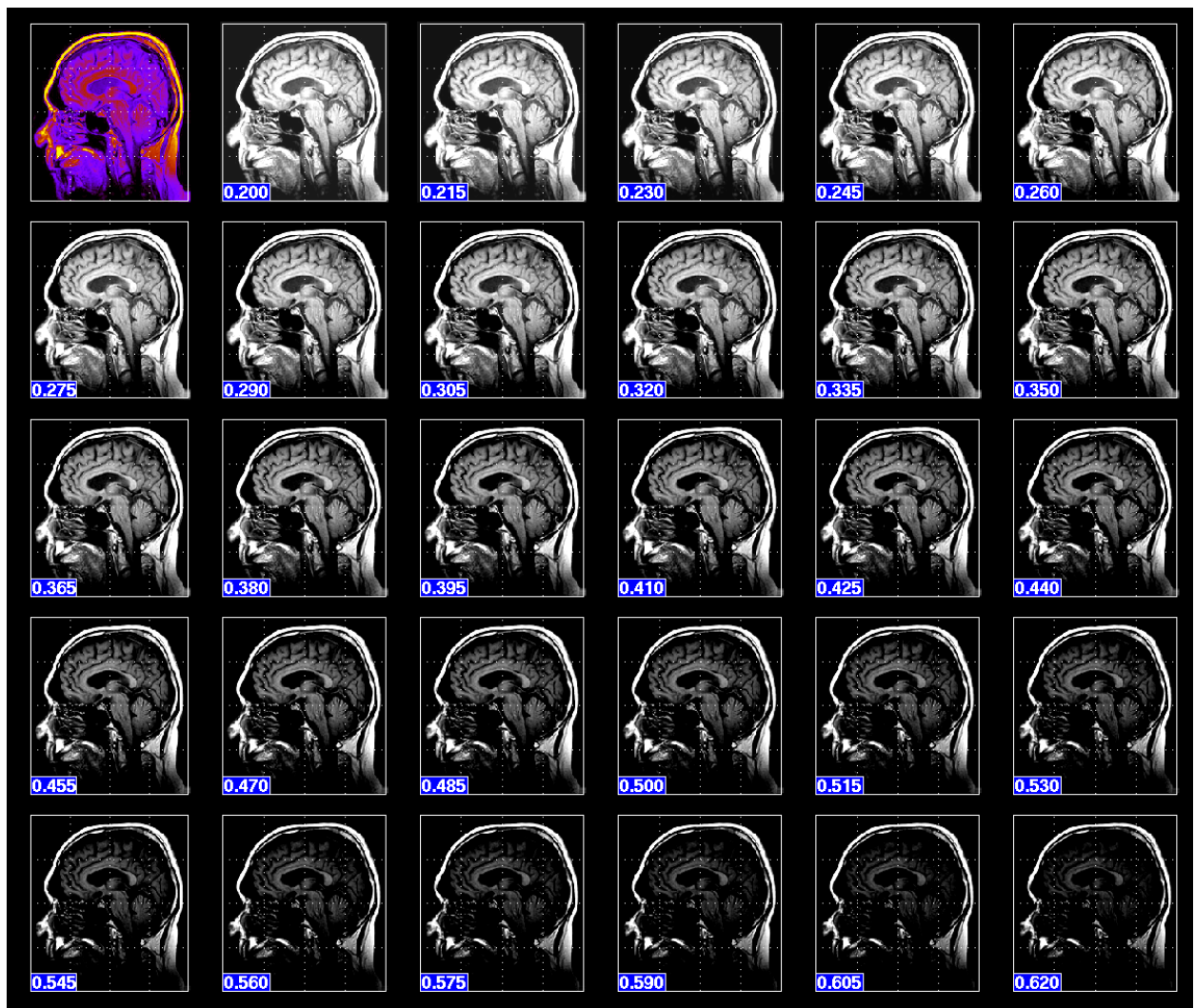


```

15 set xtics scale 0                                # grid w/o tick marks
16 set ytics scale 0
17
18 # Skalierungsfunktionen für Kontrasterhöhung
19 # gray palette with small band
20 c(z,z0,m) = (z<z0-0.5/m) ? 0 : (z>0.5/m+z0 ? 1 : m*(z-z0)+0.5)
21
22 # Einzelbilder-Größe
23 dx = 1./COLS
24 dy = 1./ROWS
25
26 set multiplot
27 set size 1.1*dx, 1.1*dy                          # weniger Platzverlust
28
29 # image in fullscale contrast
30 x = 0; y = 1-dy
31 set origin x-0.03*dx, y-0.08*dy
32 set pal color
33 plot [0:N][0:N][0:1] "CT_imgorg.dat" us ($2+0.5):($1+0.5):3 w p palette ps 1.2 pt 5
34
35 do for [k = 1:ROWS*COLS-1] {
36     x = (k % COLS) * dx
37     y = 1-dy-floor(k/COLS)*dy
38     set origin x-0.03*dx, y-0.08*dy
39
40     z0 = 0.20 + (k-1)*0.015                        # center of stretched contrast
41     LABEL = sprintf("%5.3f", z0)
42     unset label
43     set obj rect from 0,0 to 58, 20 fc rgb "blue" fs solid front
44     set label LABEL at 2,10 front
45
46     set palette model RGB functions c(gray, z0, 2), c(gray, z0, 2), c(gray, z0, 2)
47     plot [0:N][0:N][0:1] "CT_imgorg.dat" us 2:1:3 w p palette ps 1.2 pt 5
48 }
49 unset multiplot

```

Machen Sie sich die Funktionsweise klar und testen Sie den Code an anderen Bildern.



4 Verzeichnisse

4.1 Literaturverzeichnis

- [1] [\[PDF\]](#) R. Gordon, R. Bender, G. Herman, *Algebraic Reconstruction Techniques (ART) for Three-dimensional Electron Microscopy and X-ray Photography*, J.Theor.Biol. (1971)
- [2] [\[PDF\]](#) A. C. Kak, M. Slaney, *Principles of Computerized Tomographic Imaging*, IEEE PRESS (1999)

4.2 Attachments

1	[2.5] CT schematisch, Movie	7
2	[2.5] CT schematisch, Code	7
3	[2.8] ART schematisch, Code	10
4	[2.10] CT Rekonstruktion Example, Movie	12
5	[3.0] Testbild-Matrix, DATA	13
6	[3.0] Test Sinogramm, DATA	14
7	[3.0] Gnuplot-Multiplot Beispiel, Code	14

4.3 Stichwortverzeichnis

A	Lineares Gleichungssystem
Algebraic Reconstruction Technique	Kaczmarz Methode 9
ART	
F	P
FBP	Phantom
Filtered Back-Projection	Shepp-Logan-~ 12
K	R
Kaczmarz Methode	Radon-Transformation 5
Lineares Gleichungssystem	
L	S
	Shepp-Logan-Phantom 12
	Sinogramm 5