
Masterarbeit

# Algorithmus zur Rekonstruktion verrauschter CT Scans

*Maximilian Friedl*
*Matrikelnummer: 77847*
*Studienfach: Master Computational Mathematics*

Betreuung durch
Prof. Dr. Tomas Sauer

Zweitprüfung durch
Prof. Dr. Brigitte Forster-Heinlein

10. September 2020

# Kurzfassung

In der vorliegenden Arbeit wird der ART Algorithmus zur Rekonstruktion von CT Scans hergeleitet und vorgestellt. In den Zeilen der Systemmatrix $A^{m\times n}$ werden dabei die Informationen zu den einzelnen X-Ray Strahlen und im Ergebnisvektor $b^m$ die Scanergebnisse der entsprechenden Strahlen gespeichert. Eine einzelne Zeile von $A$ wird mit $a_i$ bezeichnet. Ausgangspunkt des Algorithmus ist $x^0 = 0 \in \mathbb{R}^n$ und der Iterationsschritt mit $k(i) = (i \mod m) + 1$ lautet:

$$x^{i+1} = x^i + a_{k(i)} \frac{b_{k(i)} - \langle a_{k(i)}, x^i \rangle}{\left\| a_{k(i)} \right\|^2}.$$

Außerdem wird in dieser Arbeit beweisen, dass der Algorithmus zu derjenigen Lösung des Gleichungssystems $A \cdot x = b$ mit kleinster Norm konvergiert.
Für verrauschte CT-Scans wird ebenfalls ein Algorithmus hergeleitet und dessen Konvergenz bewiesen. Dabei wird das Ungleichungssystem $b - \epsilon \leq A \cdot x \leq b + \epsilon$ betrachtet, wobei mit der Variable $\epsilon \geq 0$ versucht wird, eine Toleranz hinzuzufügen, um dem Rauschen entgegenzuwirken. Der Algorithmus wird ART4 genannt:

$$\begin{aligned}
x^0 &= 0 \in \mathbb{R}^n \\
\lambda^0 &= 0 \in \mathbb{R}^m \\
c^i &= \operatorname{mid}\left( \lambda^i_{k(i)}, \frac{(b_{k(i)} + \epsilon) - \langle a_{k(i)}, x^i \rangle}{\left\| a_{k(i)} \right\|^2}, \frac{b_{k(i)} - \epsilon) - \langle a_{k(i)}, x^i \rangle}{\left\| a_{k(i)} \right\|^2} \right) \\
x^{i+1} &= x^i + c^i \cdot a_{k(i)} \\
\lambda^{i+1}_{k(i)} &= \lambda^i_{k(i)} - c^i \cdot e_{k(i)}
\end{aligned}$$

Es werden mehrere Tests durchgeführt, um die optimale Toleranz, abhängig vom Grad des Rauschens, zu bestimmen. Abschließend werden die erhaltenen Testergebnisse auf neue Daten angewandt. Dies soll zeigen, dass die Rekonstruktion auch bei unbekannten Daten gut funktioniert.

# Inhaltsverzeichnis

# Kapitel 1

# Einleitung

Vorliegende Arbeit beschäftigt sich mit der Rekonstruktion von computertomographischen Scans, kurz CT-Scans. Diese finden vor allem in der Medizin Anwendung. Aber auch in der Industrie sind sie zu einem nicht mehr wegzudenkenden Instrument geworden, beispielsweise um sehr kleine Mängel im Metallguss erkennbar zu machen.
Zur Datengewinnung wird bei diesen Scans X-Ray Strahlung von verschiedenen Richtungen auf ein Objekt geschossen. Abhängig vom Einfallswinkel der Strahlen sowie dem Material des Objekts, wird ein Teil der Energie der Strahlen absorbiert. Für jeden Strahl wird dann die übrig gebliebene Energie gemessen und die absorbierte Energie gespeichert.

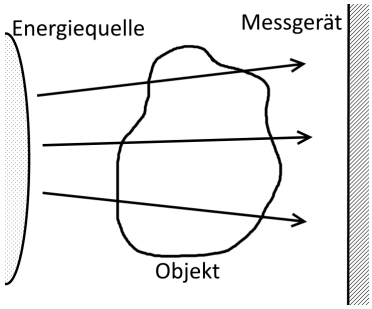



Abbildung 1.1: Beispielhafte zweidimensionale Darstellung des Scanprozesses.

In dieser Arbeit wird ein Algorithmus behandelt, der erstmals von Gabor T. Herman, Arnold Lent und Stuart W. Rowland in „*ART: Mathematics and Applications A Report on the Mathematical Foundations and on the Applicability to Real Data of the Algebraic Reconstruction Techniques*“(1973) vorgestellt wurde. Anschließend wird gezeigt, dass dieser Algorithmus - kurz

ART Algorithmus - aus den gewonnen Daten des Scans das Objekt sehr gut approximiert.
Es ist nicht unüblich, dass diese Daten verrauscht sind. Das passiert zum Beispiel durch Rundungsfehler oder durch fehlerhafte, beziehungsweise zu ungenaue Messmethoden. Für diesen Fall haben Gabor T. Herman und Arnold Lent in „*A Family of iterative quadratic optimization algorithms for pairs of inequalities, with appliation in diagnostic radiology*“(1977) einen weiteren Algorithmus präsentiert, der sowohl auf dem ART Algorithmus als auch auf Hildreths Methode für Ungleichungen [Hil57] basiert. Ausschlaggebend für die Qualität dieser Rekonstruktionen ist eine Variable zur Toleranz, welche vom Grad des Rauschens abhängt. Diese Toleranzen, sowie der Algorithmus selbst, werden ebenfalls in vorliegender Arbeit detailliert ermittelt, hergeleitet und getestet.
Abschließend wird ein unabhängiger Datensatz betrachtet und die durch die Arbeit gewonnenen Ergebnisse angewandt, um die Qualität der Algorithmen auf neuen Daten zu ermitteln.

# Kapitel 2

# Datenstruktur des CT Scans

Der gesamte Scanbereich wird in Teilbereiche unterteilt. Jeder dieser Teilbereiche wird später durch einen Pixel dargestellt werden. Demnach sind diese extrem klein zu wählen. Nun kann jeder Strahl durch einen Vektor ausgedrückt werden. Jeder Eintrag des Vektors steht für einen Teilbereich und der Wert jedes Eintrags repräsentiert eine Gewichtung des Teilbereichs in dem entsprechenden Strahl. Hierbei kann zum Beispiel die Fläche in Prozent, die der Strahl abdeckt, verwendet werden. Alternativ kann auch die Länge des Strahls innerhalb des Bereichs genommen werden, wobei zur Berechnung für jeden Bereich eine Seitenlänge von 1 festgelegt wird. Ein Beispiel hierfür ist in Abbildung 2.2 zu sehen.
Für die Qualität der Rekonstruktion ist die gewählte Methode in der Regel nicht ausschlaggebend. Wichtig ist allerdings, dass die gewählte Methode einheitlich ist.

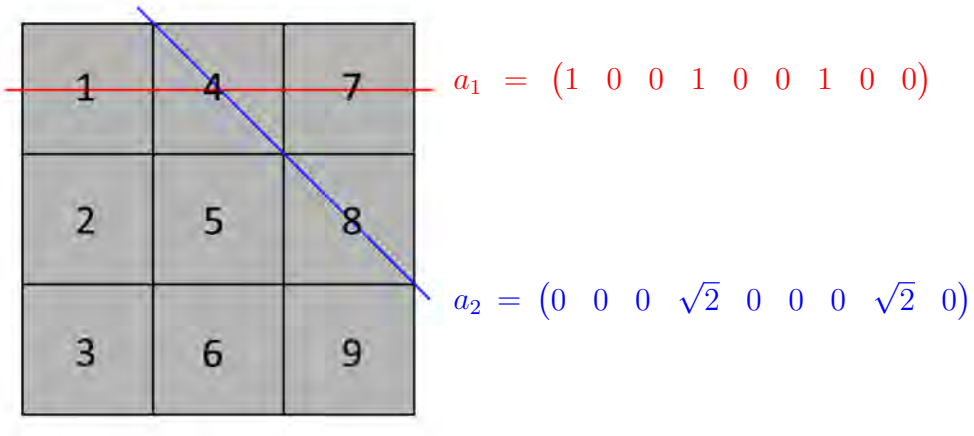
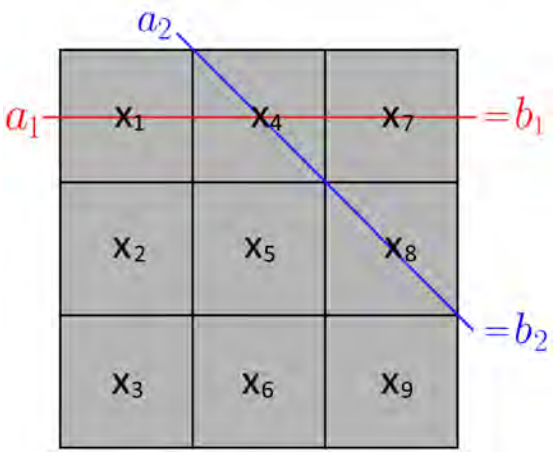



Abbildung 2.1: Beispiel der Vektorrepräsentation zweier Strahlen $a_1$ und $a_2$ durch einen zweidimensionalen abgescannten $3 \times 3$ Bereich.

Die einzelnen Vektoren werden als Zeilen untereinander in eine Matrix - Systemmatrix genannt - geschrieben. Die zu den Strahlen gehörenden Messwerte werden, der Reihenfolge in der Systemmatrix entsprechend, in einem Ergebnisvektor gespeichert.
Nun wird die Gleichung $A \cdot x = b$ betrachtet. $A$ ist dabei die Systemmatrix und $b$ der Ergebnisvektor. Der Lösungsvektor $x$ dieses Problems entspricht dem gesamten abgescannten Bereich. Dieser muss lediglich noch in die selbe Dimension wie der Scanbereich überführt werden. Jeder Eintrag von $x$ steht für einen Teilbereich und der Wert des Eintrags ist der Anteil an Strahlung, der durch den entsprechenden Teilbereich absorbiert wurde. Die Werte liegen folglich alle zwischen 0 und 1. Dadurch kann jeder Eintrag durch einen Pixel repräsentiert werden und der umgeformte Vektor $x$ als Graustufenbild dargestellt werden. 0 steht dabei für einen komplett schwarzen und 1 für einen komplett weißen Pixel.
Um den Scanbereich visuell darstellen zu können, muss demnach das Gleichungssystem $A \cdot x = b$ gelöst werden.




$$a_1 \cdot x \stackrel{!}{=} b_1$$

$$a_2 \cdot x \stackrel{!}{=} b_2$$

Abbildung 2.2: Darstellung des Ergebnisses des Scans als lineares Gleichungssystem.

Fortlaufend wird für die Anzahl an Projektionen, beziehungsweise Zeilen der Systemmatrix, die Variable $m$ und für die Anzahl an Teilbereichen, also der Spalten der Systemmatrix, $n$ verwendet.[Han]

## 2.1 Optimale Lösung

Von hier an wird als Fehler oder Norm eines Vektors stets die euklidische Norm genutzt.

Es wird nicht vorausgesetzt, dass die Systemmatrix vollen Rang besitzt. Die Gleichung $A \cdot x = b$ kann dementsprechend auch mehrere exakte Lösungen besitzen. Des weiteren kann es vorkommen, dass keine Lösung existiert, was bereits durch minimale Messfehler beziehungsweise Ungenauigkeiten hervorgerufen werden kann. In diesem Fall sind die Daten verrauscht. Wie damit umgegangen werden kann, wird ausführlich in Kapitel 4 und 5 behandelt.
Diese Sektion beschäftigt sich mit der Frage, welche der möglichen Lösungen verwendet werden soll. Voraussetzung dafür ist, dass es keine Messfehler gibt und die Daten mit 100%iger Genauigkeit gewonnen wurden. Dadurch gibt es mindestens eine exakte Lösung des Gleichungssystems.
Zuerst wird $\bar{x}$ definiert durch

$$\bar{x} = \frac{\sum_{i=1}^{m} b_i}{m \cdot n}.$$

Dies ist der Durchschnitt der Energie, die pro Teilbereich absorbiert wird und demnach der Durchschnitt der Graustufenwerte von jeder möglichen Lösung der Gleichung $A \cdot x = b$. Dadurch gilt auch $n\bar{x} = \sum_{i=1}^{n} x_i$.
Nun soll jene Lösung $x^*$ als die optimale Lösung, beziehungsweise als das finale Bild verwendet werden, für die die Distanz zu $\bar{x}$ minimal ist:

$$x^* = \underset{x:\{A \cdot x = b\}}{\operatorname{argmin}} \sum_{i=1}^{n} (x_i - \bar{x})^2.$$

Dieser Vektor ist, unter allen möglichen Lösungen, der mit der kleinsten Varianz und somit der mit der größten Gleichmäßigkeit. Diese ist in einem Bild oder Objekt nämlich in der Regel sehr hoch. Der Wert $\bar{x}$ ist eine Konstante. Demnach kann die Definition von $x^*$ folgendermaßen umgestellt werden:

$$\begin{aligned}
x^* &= \underset{x:\{A \cdot x = b\}}{\operatorname{argmin}} \sum_{i=1}^{n} (x_i - \bar{x})^2 \\
&= \underset{x:\{A \cdot x = b\}}{\operatorname{argmin}} \sum_{i=1}^{n} (x_i^2 - 2x_i\bar{x} + \bar{x}^2) \\
&= \underset{x:\{A \cdot x = b\}}{\operatorname{argmin}} \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} 2x_i\bar{x} + \sum_{i=1}^{n} \bar{x}^2 \\
&= \underset{x:\{A \cdot x = b\}}{\operatorname{argmin}} \sum_{i=1}^{n} x_i^2 - 2n\bar{x}^2 + n\bar{x}^2 \\
&= \underset{x:\{A \cdot x = b\}}{\operatorname{argmin}} \|x\|^2 .
\end{aligned}$$

Die optimale Lösung $x^*$ ist also diejenige Lösung unter allen Lösungen der Gleichung $A \cdot x = b$ mit der kleinsten Norm. [HL76]

## 2.2 Existenz einer optimalen Lösung

Falls das Gleichungssystem lösbar ist, so existiert die exakte Lösung mit minimaler Norm $x^*$ immer. Dies folgt aus dem Projektionstheorem:

**Projektionstheorem**
Sei $M$ ein geschlossener Unterraum eines Hilbertraums $H$. Sei weiter $x \in H$ und $V = x + M$. Dann existiert ein eindeutiger Vektor $x_0 \in V$, sodass gilt:

$$x_0 = \underset{x \in V}{\operatorname{argmin}} \|x\|, \qquad \text{sowie} \qquad x_0 \perp M.$$

[Lue69, S.64]

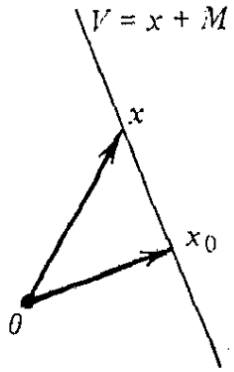



Abbildung 2.3: Visualisierung des Projektionstheorems. Quelle:[Lue69, S.64]

Im Fall der CT Rekonstruktion hat der Lösungsraum von $A \cdot x = b$, also der Schnitt der Lösungsräume der einzelnen Strahlen, die selben Eigenschaften wie der in obiger Definition beschriebene Raum $V$. In Abbildung 2.3 repräsentiert $x_0$ somit die gesuchte eindeutige Lösung mit kleinster Norm $x^*$. [HLR73]

# Kapitel 3

# Der ART Algorithmus

Die Größe der Systemmatrix $A$ hängt von der Anzahl an Strahlen und von der gewollten Auflösung der Rekonstruktion ab. Im zweidimensionalen Fall führt jeder zusätzliche Strahl zu einer weiteren Zeile in der Matrix und jeder Pixel im späteren Ergebnis benötigt eine weitere Spalte. Dadurch wird die Systemmatrix sehr groß und es ist extrem aufwändig und rechenintensiv, die Gleichung $A \cdot x = b$ exakt zu lösen. Deswegen werden für solche Fälle meist iterative Algorithmen verwendet. Im Idealfall konvergieren diese Verfahren dann auch gegen eine exakte Lösung.
Der im Folgenden erläuterte Algebraic Reconstruction Technique - kurz ART - Algorithmus von Gabor T. Herman, Arnold Lent und Stuart W. Rowland [HLR73] ist ein Beispiel hierfür, der sogar zur Lösung mit minimaler Norm $x^*$ konvergiert.

## 3.1 Definition des Algorithmus

Der ART Algorithmus nutzt die Kaczmarz-Methode [Kac37], um das Problem $A \cdot x = b$ iterativ für eine Systemmatrix $A^{m \times n}$ und den dazugehörigen Ergebnisvektor $b^m$ zu lösen. Ausgehend vom Startpunkt $x^0 = 0$ wird im Schritt $i+1$ eine Zeile von $A$ und der entsprechende Eintrag von $b$ betrachtet und die alte Lösung $x^i$ folgendermaßen manipuliert:

$$x^{i+1} = x^i + a_{k(i)} \frac{b_{k(i)} - \langle a_{k(i)}, x^i \rangle}{\left\| a_{k(i)} \right\|^2}.$$

Dabei wird mit $k(i) \in \{0, ..., m\}$ der Index des gerade betrachteten Strahls bezeichnet. In der Zeile $a_{k(i)}$ ist also der Verlauf des $k(i)$-ten Strahls gespeichert, mit der absorbierten Energie durch diesen Strahl in $b_{k(i)}$.

Es gibt diverse Möglichkeiten, die Zeilen von $A$ zu durchlaufen. Eine davon ist, mit jeder Iteration eine zufällige Zeile auszuwählen. Eine andere Möglichkeit ist es, die Matrix wiederholend von oben nach unten zu durchlaufen. Letzteres Verfahren wird in vorliegender Arbeit aufgrund von Reproduzierbarkeit auch hauptsächlich verwendet. Dadurch ergibt sich $k(i) = (i \mod m) + 1$.

## 3.2 Geometrische Erklärung

Das Problem $A \cdot x = b$ kann auch geometrisch dargestellt werden. Jede Gleichung $a_i \cdot x = b_i$ wird dabei als Hyperebene im Raum $\mathbb{R}^n$ gesehen. Ein Strahl spannt somit einen Raum mit Dimension $n-1$ auf, nämlich exakt den Lösungsraum des Strahls. Der Vektor $a_i$ steht dann orthogonal auf diesen Raum. Die möglichen Lösungen von $A \cdot x = b$ sind schlussendlich die Punkte, in denen sich all diese Hyperebenen schneiden.

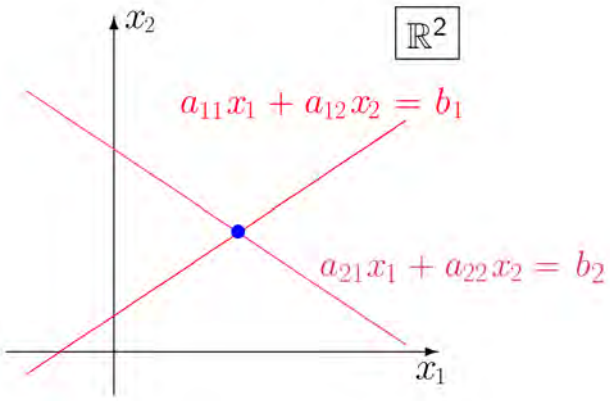



Abbildung 3.1: Beispiel für die Lösungsräume zweier Strahlen im zweidimensionalen Raum $m = n = 2$. Quelle:[Han, S.18]

Die Formel für die orthogonale Projektion $P(z)$ eines Punktes $z$ auf eine affine Hyperebene - gegeben durch $a \cdot x = b$ - lautet:

$$P(z) = z + a \cdot \frac{b - \langle a, z \rangle}{\|a\|^2}.$$

[Han, S.20]

Durch den ART-Algorithmus wird also mit jedem Iterationsschritt $i$ der aktuelle Stand der approximierten Lösung $x^i$ orthogonal auf jenen Lösungsraum

projiziert, der durch die gerade betrachtete Zeile der Systemmatrix $a_i$ und den dazugehörigen Wert des Ergebnisvektors $b_i$ aufgespannt wird. Visuell bedeutet das:

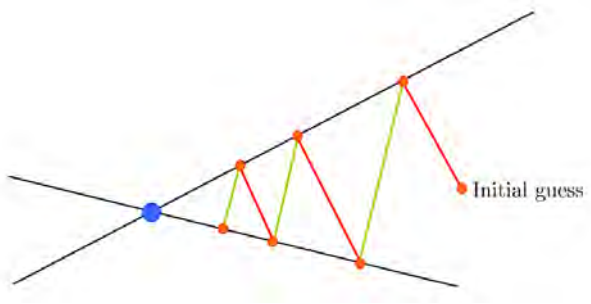



Abbildung 3.2: Alternierende orthogonale Projektion auf die beiden Hyperebenen. Quelle:[Han, S.19]

## 3.3 Konvergenz gegen x*

Der ART Algorithmus konvertiert zur optimalen Lösung $x^*$. Der folgende Beweis hierzu ist eine abgewandelte Form des Beweises in [HLR73]. Sei zunächst der Vektor $y \in L$ ein beliebiges Element aus dem Lösungsraum von $A \cdot x = b$. Der Abstand vom nächsten Iterationsschritt des ART Algorithmus $x^{i+1}$ zu $y$ lässt sich folgendermaßen umschreiben. Zu beachten hierbei ist, dass $a_i \cdot y = b_i$, sowie dass $(\alpha - \beta) \cdot (\beta - \alpha) = -(\alpha - \beta)^2$ für alle $\alpha, \beta \in \mathbb{R}$:

$$
\begin{aligned}
&\left\|x^{i+1} - y\right\|^2 = \\
&= \left\| x^i - y + a_{k(i)} \frac{b_{k(i)} - \langle a_{k(i)}, x^i \rangle}{\left\|a_{k(i)}\right\|^2} \right\|^2 \\
&= \left\|x^i - y\right\|^2 + 2 \cdot \langle x^i - y, a_{k(i)} \rangle \cdot \frac{b_{k(i)} - \langle a_{k(i)}, x^i \rangle}{\left\|a_{k(i)}\right\|^2} + \left\| a_{k(i)} \frac{b_{k(i)} - \langle a_{k(i)}, x^i \rangle}{\left\|a_{k(i)}\right\|^2} \right\|^2 \\
&= \left\|x^i - y\right\|^2 + 2 \cdot (\langle x^i, a_{k(i)} \rangle - b_{k(i)}) \cdot \frac{b_{k(i)} - \langle a_{k(i)}, x^i \rangle}{\left\|a_{k(i)}\right\|^2} + \left\|a_{k(i)}\right\|^2 \cdot \frac{|b_{k(i)} - \langle a_{k(i)}, x^i \rangle|^2}{\left\|a_{k(i)}\right\|^4} \\
&= \left\|x^i - y\right\|^2 - 2 \cdot \frac{|b_{k(i)} - \langle a_{k(i)}, x^i \rangle|^2}{\left\|a_{k(i)}\right\|^2} + \frac{|b_{k(i)} - \langle a_{k(i)}, x^i \rangle|^2}{\left\|a_{k(i)}\right\|^2} \qquad (3.1) \\
&= \left\|x^i - y\right\|^2 - \frac{|b_{k(i)} - \langle a_{k(i)}, x^i \rangle|^2}{\left\|a_{k(i)}\right\|^2} \\
&= \left\|x^i - y\right\|^2 - \left\|x^{i+1} - x^i\right\|^2
\end{aligned}
$$

Die Folge $\{\|x^i - y\|^2\}$ ist dementsprechend monoton fallend und kann auf Grund der Norm nicht kleiner als 0 werden, sie ist demnach nach unten beschränkt. Folglich existiert ein Grenzwert $\lim_{i\to\infty} \|x^i - y\|^2$ und für den Abstand zwischen $x^i$ und $x^{i+1}$ gilt: $\lim_{i\to\infty} \|x^{i+1} - x^i\|^2 = 0$. Des weiteren ist die Folge $\{x^{(i)}\}$ beschränkt.

Aus dem Satz von Bolzano-Weierstraß folgt, aufgrund der Beschränktheit, dass eine konvergente Teilfolge von $\{x^{(i)}\}$ existieren muss.
Sei nun $\{x^{(j)}\}$ eine Teilfolge von $\{x^{(i)}\}$, die gegen einen Vektor $z$ konvergiert. Eine Teilfolge $\{x^{(t)}\}$ von der Teilfolge $\{x^{(j)}\}$ konvergiert dann ebenso gegen $z$, wobei für jedes $t$ gelten soll: $t = lm + k$, für $l \in \mathbb{N}_0$ und ein fixes $k$ mit $1 \leq k \leq m$. Die Teilfolge $\{x^{(t)}\}$ betrachtet somit nur jene $x^j$ nach den Anpassungen durch die $k-$te Zeile $a_k$ der Systemmatrix $A$, also nach den Projektionen auf den Lösungsraum des $k-$ten Strahls. Dadurch ergibt sich die Gleichung $a_k \cdot z = b_k$. Wie oben gezeigt gilt $\|x^t - x^{t+1}\|^2 \to 0$. Die Teilfolge $\{x^{(t+1)}\}$ konvergiert daher ebenso gegen $z$ und es gilt: $a_{k+1} \cdot z = b_{k+1}$. Dies kann für alle Zeilen der Systemmatrix wiederholt werden, außer für den Fall $k = n$, hier wird an Stelle von $k + 1$, 1 verwendet. Der Vektor $z$ ist folglich eine mögliche Lösung, was $A \cdot z = b$ bedeutet.
Sei $\{x^{(j')}\}$ nun eine weitere Teilfolge, die gegen einen Vektor $z'$ konvergiert. Es kann analog zu vorheriger Ausführung gezeigt werden, dass $A \cdot z' = b$ gilt. Weiter wird $\alpha$ definiert durch:

$$\alpha = \lim_{i\to\infty} (\left\|x^{(i)} - z\right\|^2 - \left\|x^{(i)} - z'\right\|^2).$$

Unter Anwendung der Teilfolge $\{x^{(t)}\}$, die gegen $z$ konvergiert, folgt $\alpha = -\|z - z'\|^2$, wohingegen mit $\{x^{(t')}\}$, $\alpha = \|z' - z\|^2$ gilt. Dadurch ergibt sich $\alpha = 0$ und $z = z'$. Alle konvergenten Teilfolgen konvergieren somit gegen $z$. Da $\{x^{(i)}\}$ beschränkt ist, konvergiert die Folge $\{x^{(i)}\}$ ebenso gegen $z$.

Nun muss gezeigt werden, dass $z$ diejenige Lösung $x^*$ ist, deren Norm unter allen Lösungen minimal ist. Dazu wird der Vektorraum $S$ betrachtet, den die Zeilen der Systemmatrix aufspannen: $S := \langle\{a_1, a_2, ..., a_m\}\rangle$. Der Startpunkt $x^0 = 0$ liegt offensichtlich in $S$. Da bei jedem Iterationsschritt lediglich mit einem Vielfachen einer Zeile $a_k$ addiert wird, liegt folglich auch jedes $x^i$ in $S$. Der Algorithmus konvergiert also zum Schnitt von $S$ und der Lösungsmenge $L$. Da weiter $a_i$ orthogonal zur Lösung von $a_i \cdot x = b_i$ für alle $1 \leq i \leq m$ steht, steht auch $S$ orthogonal zu $L$. Aufgrund des Projektionstheorems [Lue69] ist nun der Schnitt von $S$ und $L$ jenes Element aus $L$ mit minimaler Norm und deswegen das gesuchte $x^*$.

In [HLR73] wird außerdem bewiesen, dass der Algorithmus ebenfalls konvergiert, wenn für das Gleichungssystem keine Lösung existiert. In diesem Fall liegt eine zyklische Konvergenz vor.

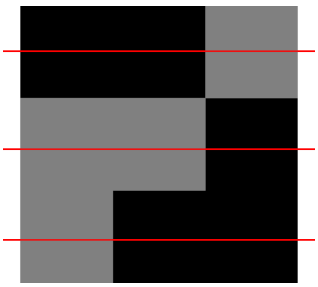
## 3.4 Beispielobjekt in 3×3

Zum besseren Verständnis nun ein kleines Beispiel mit zweidimensionalem Scanbereich der Größe $3 \times 3$. Die Systemmatrix wird insofern aus neun Spalten bestehen. Das gescannte 'Objekt' ist grau dargestellt. Jeder Pixel, beziehungsweise jeder Teilbereich, absorbiert $\frac{1}{5}$ der gesamten Energie eines Strahls. In diesem Beispiel bedeutet $b_i = 0$, dass keine Energie des $i$-ten Strahls absorbiert wurde und $b_i = 1$, dass die komplette Energie absorbiert wurde.

$$\widehat{=} \quad \begin{pmatrix} 0 & 0 & 0.2 \\ 0.2 & 0.2 & 0 \\ 0.2 & 0 & 0 \end{pmatrix}$$

Abbildung 3.3: Abgescannter Bereich der Größe $3 \times 3$.

Die Fläche wird von vier verschiedenen Winkeln bestrahlt und die absorbierte Energie im Ergebnisvektor festgehalten:

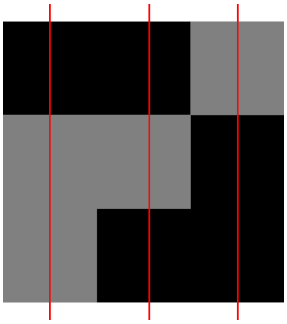


$a_1 = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$
$\rightarrow b_1 = 0.2$

$a_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$
$\rightarrow b_2 = 0.4$

$a_3 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$
$\rightarrow b_3 = 0.2$

Abbildung 3.4: Drei horizontale Strahlen, die repräsentierenden Zeilen $a_i$ der Systemmatrix und die dazugehörigen Messergebnisse $b_i$.

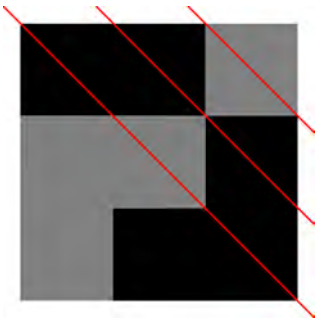


$$a_4 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$
$$\rightarrow b_4 = 0.4$$

$$a_5 = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$
$$\rightarrow b_5 = 0.2$$

$$a_6 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$
$$\rightarrow b_6 = 0.2$$

Abbildung 3.5: Drei vertikale Strahlen bilden die nächsten Zeilen der Systemmatrix und die dazugehörigen Einträge des Ergebnisvektors.



$$a_7 = \begin{pmatrix} \sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} \end{pmatrix}$$
$$\rightarrow b_7 = 0.2 \cdot \sqrt{2}$$

$$a_8 = \begin{pmatrix} 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 \end{pmatrix}$$
$$\rightarrow b_8 = 0$$

$$a_9 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{2} & 0 & 0 \end{pmatrix}$$
$$\rightarrow b_9 = 0.2 \cdot \sqrt{2}$$

Abbildung 3.6: Es folgen drei Strahlen, die von links oben nach rechts unten gehen.

$$a_{10} = \begin{pmatrix} \sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$
$$\rightarrow b_{10} = 0$$

$$a_{11} = \begin{pmatrix} 0 & \sqrt{2} & 0 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$
$$\rightarrow b_{11} = 0.2 \cdot \sqrt{2}$$

$$a_{12} = \begin{pmatrix} 0 & 0 & \sqrt{2} & 0 & \sqrt{2} & 0 & \sqrt{2} & 0 & 0 \end{pmatrix}$$
$$\rightarrow b_{12} = 0.6 \cdot \sqrt{2}$$

Abbildung 3.7: Zuletzt drei Strahlen von rechts oben ausgehend mit den entsprechenden Vektoren und Werten.

Die Systemmatrix und der dazugehörige Ergebnisvektor sehen dann fol-

gendermaßen aus:

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ \sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} \\ 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{2} & 0 & 0 \\ \sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{2} & 0 & \sqrt{2} & 0 & \sqrt{2} & 0 & 0 \end{pmatrix}, b = \begin{pmatrix} 0.2 \\ 0.4 \\ 0.2 \\ 0.4 \\ 0.2 \\ 0.2 \\ 0.2 \cdot \sqrt{2} \\ 0 \\ 0.2 \cdot \sqrt{2} \\ 0 \\ 0.2 \cdot \sqrt{2} \\ 0.6 \cdot \sqrt{2} \end{pmatrix}.$$

Die Matrizen $A$ und $[A|b]$ besitzen einen Rang von 9. Das lineare Gleichungssystem $A \cdot x = b$ ist somit eindeutig lösbar.
Der ART Algorithmus liefert nach bereits 100 Iterationen den Vektor:

$$x \approx \begin{pmatrix} 0.001 \\ -0.002 \\ 0.202 \\ 0.201 \\ 0.198 \\ 0.001 \\ 0.198 \\ 0.004 \\ -0.003 \end{pmatrix} \Rightarrow$$

Dies entspricht fast exakt dem gescannten Objekt und ist visuell nicht mehr vom Original unterscheidbar.

## 3.5 Vollständig absorbierte Strahlen

Falls manche Strahlen vollständig absorbiert werden, beispielsweise durch Blei, so funktioniert die Rekonstruktion durch den ART Algorithmus nicht mehr einwandfrei. Folgendes Beispiel soll dies verdeutlichen. Dabei wird das selbe Objekt wie in vorheriger Sektion verwendet, mit dem Unterschied, dass durch den zweiten Teilbereich die gesamte Energie absorbiert wird:

$$\widehat{=} \quad \begin{pmatrix} 0 & 0 & 0.2 \\ 1 & 0.2 & 0 \\ 0.2 & 0 & 0 \end{pmatrix}$$

Abbildung 3.8: Abgescannter Bereich der Größe $3 \times 3$; der Kontrast wurde zur besseren Erkennbarkeit erhöht.

Die Systemmatrix $A$ ist identisch zum Beispiel aus vorherigem Kapitel, der Ergebnisvektor $b$ lautet also:

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ \sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} \\ 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{2} & 0 & 0 \\ \sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{2} & 0 & \sqrt{2} & 0 & \sqrt{2} & 0 & 0 \end{pmatrix}, b = \begin{pmatrix} 0.2 \\ 1 \\ 0.2 \\ 1 \\ 0.2 \\ 0.2 \\ 0.2 \cdot \sqrt{2} \\ 0 \\ 0.2 \cdot \sqrt{2} \\ 0 \\ 1 \\ 0.6 \cdot \sqrt{2} \end{pmatrix}$$

Der Algorithmus liefert als Ergebnisvektor:

$$x \approx \begin{pmatrix} 0.017 \\ -0.026 \\ 0.208 \\ 0.750 \\ 0.200 \\ 0.030 \\ 0.207 \\ 0.022 \\ -0.034 \end{pmatrix} \Rightarrow$$

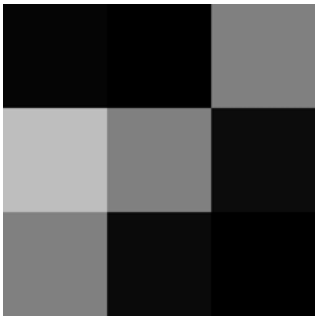


Abbildung 3.9: Der Kontrast wurde zur besseren Erkennbarkeit erhöht.

Die zwei relevanten Ränge zur Lösbarkeit linearer Gleichungssysteme lauten $rang(A) = 9$ und $rang([A|b]) = 10$. Das betrachtete Gleichungssystem besitzt also keine Lösung. Zurückzuführen ist dies darauf, dass die Einträge des Ergebnisvektors $b$ auf $[0, 1]$ beschränkt sind. Es kann nicht mehr als die komplette Energie eines Strahls absorbiert werden. Das Gleichungssystem, welches zum korrekten Ergebnis führen würde, lautet allerdings:

$$b = \begin{pmatrix} 0.2 \\ 1.2 \\ 0.2 \\ 1.2 \\ 0.2 \\ 0.2 \\ 0.2 \cdot \sqrt{2} \\ 0 \\ 0.2 \cdot \sqrt{2} \\ 0 \\ \sqrt{2} \\ 0.6 \cdot \sqrt{2} \end{pmatrix}$$

Dieses Problem kann gelöst werden, indem alle Strahlen $k$ mit $b_k = 1$ ignoriert werden. Ist die Fläche, welche die Strahlung vollständig absorbiert nicht zu groß, werden die Bereiche der ignorierten Strahlen noch durch andere Strahlen abgedeckt. Bereiche, welche die Strahlen vollständig absorbieren, werden allerdings nie besucht. Da der Ausgangsvektor $x^0 = 0$ ist, muss bei der Implementierung dann darauf geachtet werden, dass alle nie besuchten Stellen auf 1 gesetzt werden.

## 3.6 Anzahl Iterationen

Um die optimale Anzahl an Iterationen zu ermitteln, wurden Tests mit Daten unterschiedlicher Dimensionen durchgeführt. Nach jeder Iteration des Algorithmus wurde als Fehler die euklidische Norm von $A \cdot x^i - b$ berechnet und graphisch dargestellt. Zu sehen sind außerdem jeweils zwei Zwischenergebnisse und das finale Ergebnis des Algorithmus.

### 3.6.1 3×3 Objekt

Das erste Beispiel ist das $3 \times 3$ Objekt aus Kapitel 3.4. Das Gleichungssystem $A \cdot x = b$ ist eindeutig lösbar, der Fehler konvergiert demzufolge gegen 0.

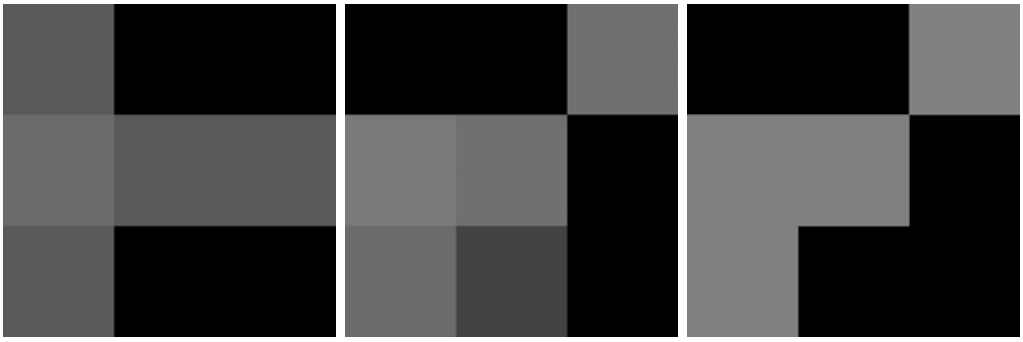


Abbildung 3.10: 3×3 Beispiel nach 5 Iterationen (links), 20 Iterationen(mitte) und 50 Iterationen(rechts). Zur besseren Erkennbarkeit wurde bei allen Bildern der Kontrast erhöht.

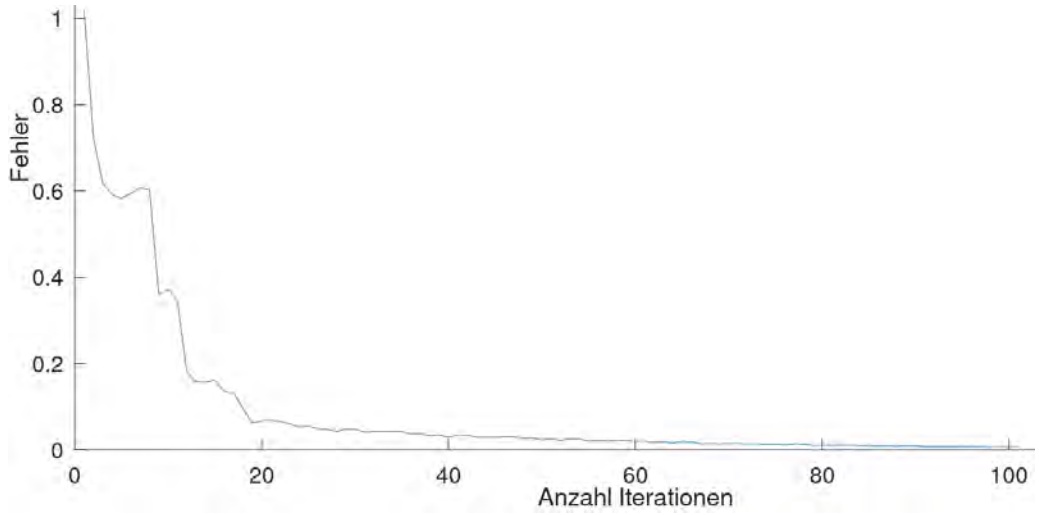



Abbildung 3.11: Anzahl Iterationen $3 \times 3$ Beispiel

### 3.6.2 gestanztes Käsestück

Das nächste Beispiel zeigt ein Stück Käse, in welches die Buchstaben 'CT' eingestanzt sind. Das Ausgangsbild hat eine Dimension von $512 \times 512$ und die Systemmatrix $dim(A) = 14\,835 \times 262\,144$. Es besteht also lediglich aus 14 835 Strahlen. Das System ist nicht exakt lösbar, der Algorithmus konvergiert demnach nicht zu einem Fehler von 0. Der minimale Fehler hier ist $\approx 3.20$.

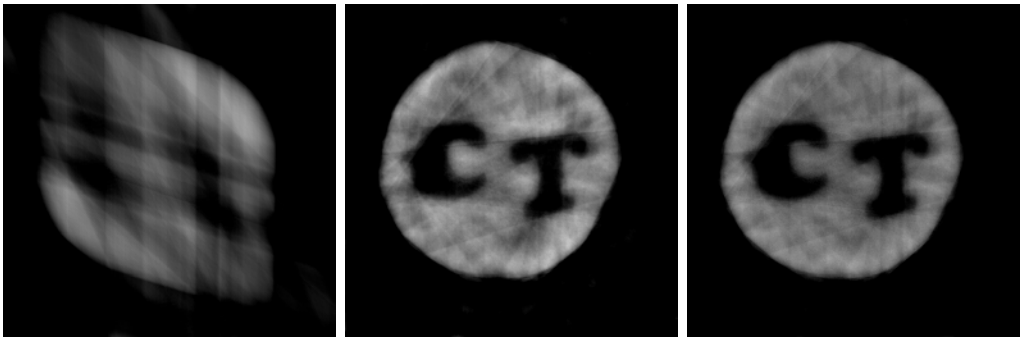


Abbildung 3.12: Scanergebnis nach 4 000 Iterationen (links), nach 20 000 Iterationen (mitte) und nach 50 000 Iterationen (rechts). Zur besseren Erkennbarkeit wurde bei allen Bildern der Kontrast erhöht. Quelle Datensatz:[AMJ+17]

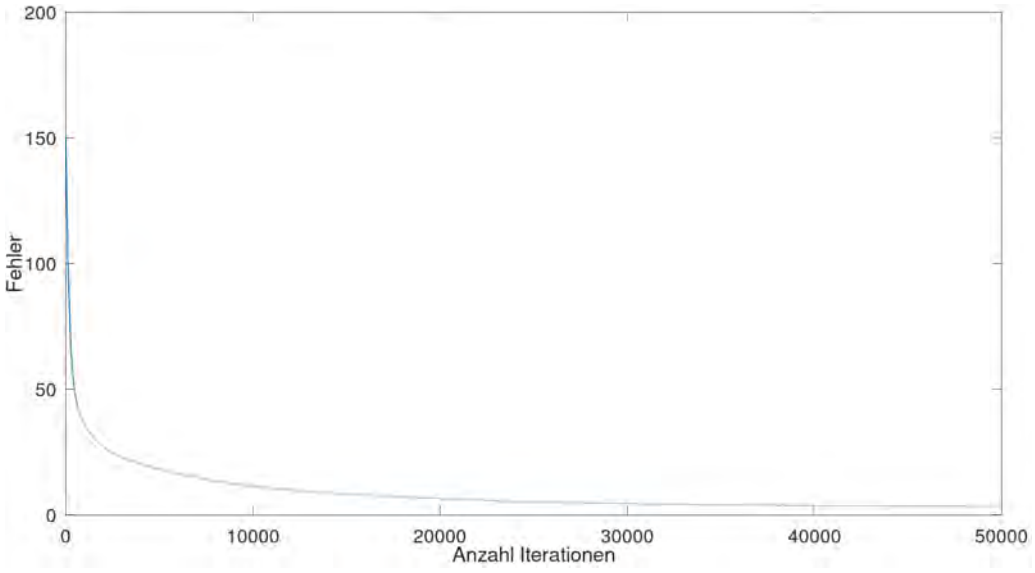



Abbildung 3.13: Fehler nach x Iterationen.

### 3.6.3 Walnuss Querschnitt

Das letzte Beispiel zeigt den Querschnitt einer Walnuss mit Dimension $328 \times 328$. Für die Systemmatrix gilt $dim(A) = 39\,360 \times 107\,584$. Das System ist ebenfalls nicht exakt lösbar, der minimale Fehler $A \cdot x^i - b$ lautet $\approx 12.60$.

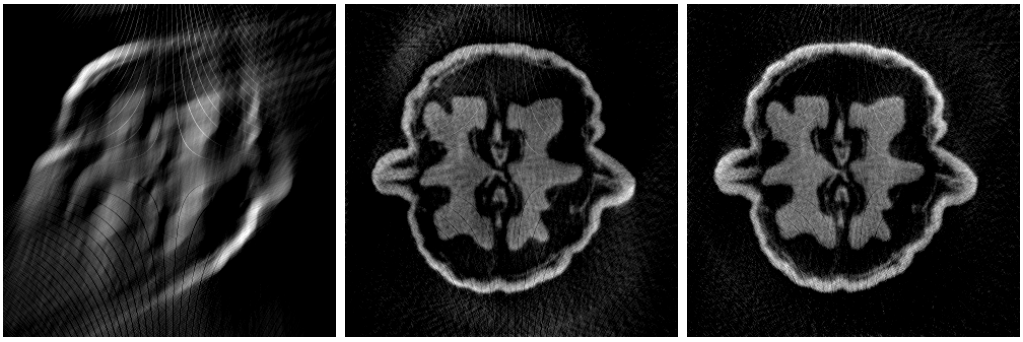


Abbildung 3.14: Scanergebnis nach 8 000 Iterationen (links), nach 40 000 Iterationen (mitte) und nach 100 000 Iterationen (rechts). Zur besseren Erkennbarkeit wurde bei allen Bildern der Kontrast erhöht. Quelle Datensatz:[KLA+15]

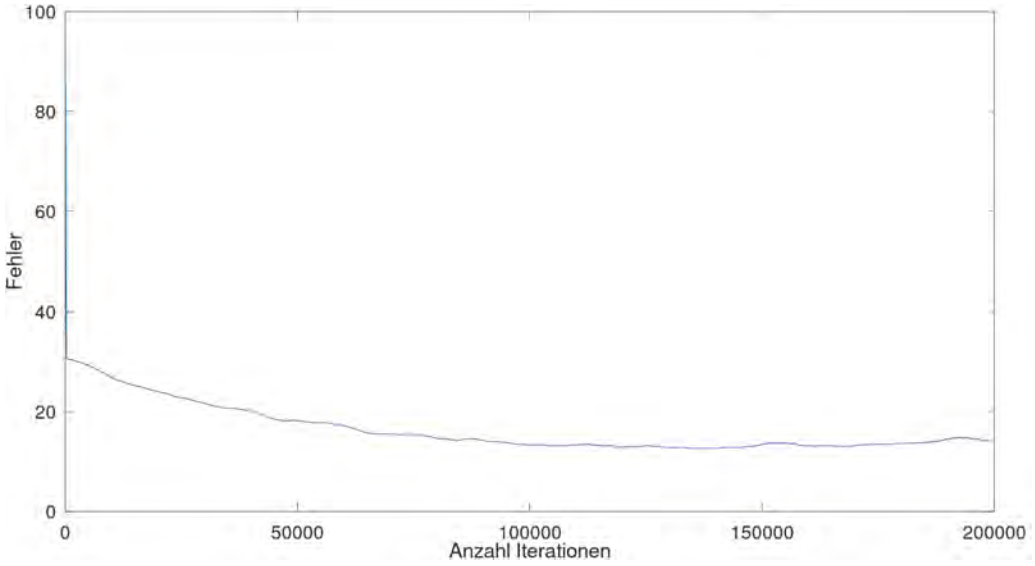



Abbildung 3.15: Fehler nach x Iterationen.

### 3.6.4 Zwischenfazit

Bei allen Beispielen fällt auf, dass der Fehler nach einigen wenigen Iterationen bereits extrem fällt. Des weiteren kann bei den letzten beiden Beispielen beobachtet werden, wie sich nach $\approx 0.25 \cdot m$ Iterationen auch die Umrisse des Objekts erst noch formen müssen. Nach etwas mehr als einem Durchlauf aller Zeilen der Systemmatrix ist das gescannte Objekt dann bereits sehr gut rekonstruiert. Nach drei Durchläufen, also $3 \cdot m$ Iterationen, kann da-

von ausgegangen werden, dass der Fehler nicht mehr geringer wird und der Algorithmus spätestens gestoppt werden kann.

## 3.7 Zufällige Strahlen

Falls die Systemmatrix nicht zyklisch durchlaufen wird, sondern in jedem Schritt ein zufälliger Strahl ausgewählt wird, ist nachfolgend zu sehen, dass der ART Algorithmus schneller zu einem visuell besseren Ergebnis führt. Anders als in vorheriger Sektion reichen hier bereits $\approx 0.25 \cdot m$ Iterationen um das Bild sehr gut erkennen zu können. Der in diesem Test verwendete Datensatz ist der selbe wie der, der in Kapitel 3.6.2 verwendet wurde.

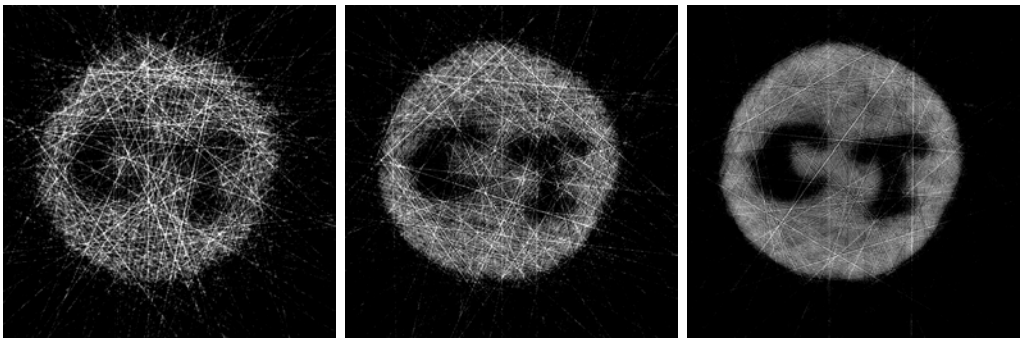


Abbildung 3.16: Scanergebnis nach 4 000 Iterationen (links), nach 8 000 Iterationen (mitte) und nach 20 000 Iterationen (rechts). Zur besseren Erkennbarkeit wurde bei allen Bildern der Kontrast erhöht. Quelle Datensatz:[AMJ+17]

Selbiges gilt auch für den Datensatz aus Kapitel 3.6.3. Auch hier genügen verhältnismäßig wenige Iterationen um das gescannte Objekt bereits gut zu erkennen. In beiden Fällen ist allerdings die Konvergenzgeschwindigkeit, sowie der minimale Fehler, nahezu identisch zu den Tests in Kapitel 3.6.2. beziehungsweise 3.6.3.

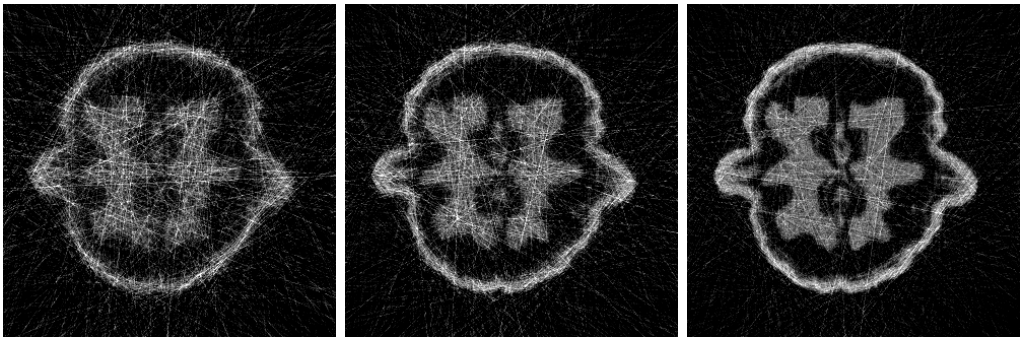


Abbildung 3.17: Scanergebnis nach 8 000 Iterationen (links), nach 18 000 Iterationen (mitte) und nach 40 000 Iterationen (rechts). Zur besseren Erkennbarkeit wurde bei allen Bildern der Kontrast erhöht. Quelle Datensatz:[KLA+15]

## 3.8 Gewichtetes ART

Es ist möglich eine Gewichtung $w_i$ zum Algorithmus hinzuzufügen:

$$x^{i+1} = x^i + w_i \cdot a_{k(i)} \frac{b_{k(i)} - \langle a_{k(i)}, x^i \rangle}{\left\| a_{k(i)} \right\|^2}.$$

Dies wirkt sich natürlich auf die Konvergenz aus. Im Beweis zur Konvergenz in Kapitel 3.3 muss dann ab (3.1) aufgrund der zusätzlichen Variable anders fortgefahren werden:

$$\begin{aligned}
&= \left\| x^i - y \right\|^2 - 2 \cdot w_i \cdot \frac{|b_{k(i)} - \langle a_{k(i)}, x^i \rangle|^2}{\left\| a_{k(i)} \right\|^2} + w_i^2 \cdot \frac{|b_{k(i)} - \langle a_{k(i)}, x^i \rangle|^2}{\left\| a_{k(i)} \right\|^2} \\
&= \left\| x^i - y \right\|^2 - w_i \cdot (2 - w_i) \cdot \frac{|b_{k(i)} - \langle a_{k(i)}, x^i \rangle|^2}{\left\| a_{k(i)} \right\|^2} \\
&= \left\| x^i - y \right\|^2 - \frac{2 - w_i}{w_i} \left\| x^{i+1} - x^i \right\|^2.
\end{aligned}$$

Die Folge $\{\|x^i - y\|^2\}$ ist also nur monoton fallend - und somit die Konvergenz gegen $x^*$ nur gegeben - falls $w_i \in (0, 2)$ gewählt wird. Im Zuge vorliegender Arbeit wurden mehrere Tests mit unterschiedlichen Gewichtungen und Datensätzen gemacht. Diese haben gezeigt, dass eine Gewichtung von $w_i \leq 1$ in der Regel zu einem Ergebnis mit geringerem Fehler konvergiert als Gewichtungen von $w_i > 1$. Jedoch waren die Fehler bei $w_i < 1$ nicht niedriger als bei $w_i = 1$ und auch die Konvergenzgeschwindigkeit war in diesen Tests nahezu identisch. Es wurde in allen Tests eine konstante Gewichtung für jede Iteration $i$ verwendet.

## 3.9 Implementierung

In der Implementierung wird damit begonnen, $x^0$ sowie den 'Flags' Vektor zu definieren. Im 'Flags' Vektor wird festgehalten, welche Teilbereiche bereits besucht wurden. Anschließend wird der Algorithmus so oft ausgeführt, bis die Systemmatrix 3 mal durchlaufen wurde. Dies basiert auf den Kenntnissen aus Kapitel 3.6. Bei jeder Iteration wird dabei zuerst geprüft, ob $b_{k(i)} = 1$ gilt. Falls ja, wurde der gesamte $k(i)$-te Strahl absorbiert und aufgrund Kapitel 3.5 wird dieser Iterationsschritt übersprungen. Falls $b_{k(i)} < 1$ werden alle Teilbereiche, die durch die gerade betrachtete Zeile der Systemmatrix $a_{k(i)}$ abgedeckt werden, als besucht markiert und der Iterationsschritt gemäß Definition ausgeführt. Am Ende des Algorithmus werden alle nie besuchten Teilbereiche von $x$ wie in Kapitel 3.5 erläutert auf 1 gesetzt.
Der Wert, der impliziert, dass ein Strahl vollständig absorbiert wurde, ist abhängig von der Messmethodik. Daher muss dieser Wert in der Implementierung eventuell angepasst werden. Es handelt sich dabei um jenen, der in der if-Abfrage mit $b_{k(i)}$ verglichen wird.

```
function x = solveART(A,b)
  m = size(A)(1);
  n = size(A)(2);
  x = zeros(n,1);
  flags = zeros(size(x));
  for i = 0:(3*m)
      ki = mod(i,m) + 1;
      bki = b(ki);
      aki = A(ki,:);
      if(bki == 1)
          continue;
      end
      flags(aki!=0) = 1;
      x = x + (aki*((bki - aki*x)/(aki*aki')))';
  end
  x(flags==0) = 1;
end
```

# Kapitel 4

# ART für Ungleichungen

Um verrauschte CT Scans gut mit Hilfe von ART zu rekonstruieren, wird das ursprüngliche Problem folgendermaßen umgeformt:

$$b - \epsilon \leq A \cdot x \leq b + \epsilon.$$

Dabei soll $\epsilon \geq 0$ und die Norm von $x$ erneut minimal sein. Durch die Toleranz in Form der Variable $\epsilon$ werden leichte Abweichungen herausgefiltert, die hauptsächlich durch das Rauschen entstehen. Klare Kanten des gescannten Objekts bleiben allerdings weiterhin erhalten.[Her73]
Dazu wird zunächst eine Methode benötigt, um Ungleichungen der Form $A \cdot x \leq b$ zu lösen. Vorausgesetzt das Ungleichungssystem ist lösbar, gibt es immer unendlich viele Lösungen. Auch hier ist die gesuchte Lösung diejenige mit der kleinsten Norm. Für die bisher betrachteten Fälle sind die Einträge der Matrix $A$ und des Vektors $b$ nie negativ. Der Ausgangspunkt $x^0 = 0$ wäre hier also immer die Lösung mit der kleinsten Norm. Aus diesem Grund wird hier nun eine beliebige Matrix $A$ sowie ein beliebiger Vektor $b$ verwendet.

## 4.1 Bedingtes ART für Ungleichungen

Basierend auf dem ART Algorithmus aus dem letzten Kapitel ist ein naheliegender Lösungsansatz für obiges Ungleichungssystem die Anpassung nur anzuwenden, wenn die Ungleichung $a_{k(i)} \cdot x \leq b_{k(i)}$ nicht gilt [HL76]:

$$x^{i+1} = x^i + \begin{cases} 0, \text{ falls } a_{k(i)} \cdot x \leq b_{k(i)} \\ a_{k(i)} \frac{b_{k(i)} - \langle a_{k(i)}, x^i \rangle}{\left\| a_{k(i)} \right\|^2}, \text{ sonst.} \end{cases}$$

Der Algorithmus konvergiert natürlich gegen eine mögliche Lösung. Der Beweis hierfür ist analog zu dem Beweis der Konvergenz des ART Algorithmus

in Kapitel 3.3. Das Resultat des Algorithmus ist allerdings nicht die Lösung mit der kleinsten Norm. Folgendes einfache Beispiel zeigt dies deutlich:

$$A = \begin{pmatrix} -1 & 1 \\ 0.1 & 1 \end{pmatrix}, \qquad b = \begin{pmatrix} -2 \\ -2 \end{pmatrix}$$

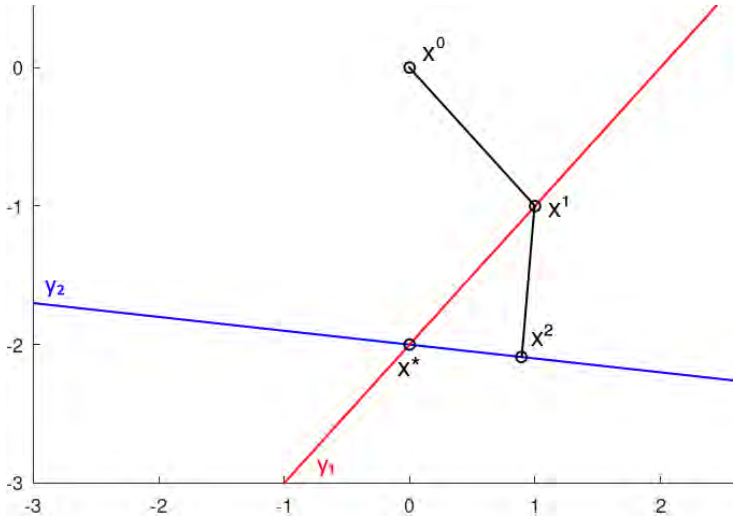



Abbildung 4.1: Iterationsschritte des Algorithmus mit obigen Problem.

Der Algorithmus konvergiert nach 2 Iterationen zum Punkt $x^2 \approx (0.9, -2.1)^T$ mit Norm $\|x^2\| = 2.2712$. Dies ist eine mögliche Lösung des Ungleichungssystems, der Algorithmus stoppt demnach an diesem Punkt. Die Lösung mit minimaler Norm ist allerdings offensichtlich $x^* = (0, -2)^T$.

## 4.2 ART mit Hildreths Prozess

Um die Lösung von Ungleichungssystemen mit der kleinsten Norm zu finden, wird in [HL76] ein Algorithmus basierend auf dem ART Algorithmus und Hildreths Prozess für Ungleichungen [Hil57] vorgestellt:

$$x^0 = 0 \in \mathbb{R}^n \tag{4.1}$$

$$\lambda^0 = 0 \in \mathbb{R}^m \tag{4.2}$$

$$c^i = \min\left(\lambda^i_{k(i)}, \frac{b_{k(i)} - \langle a_{k(i)}, x^i \rangle}{\left\|a_{k(i)}\right\|^2}\right) \tag{4.3}$$

$$x^{i+1} = x^i + c^i \cdot a_{k(i)} \tag{4.4}$$

$$\lambda^{i+1} = \lambda^i - c^i \cdot e_{k(i)} \tag{4.5}$$

Hier wird mit $e_{k(i)}$ der $e_{k(i)}$-te Einheitsvektor bezeichnet, mit 1 am Eintrag $e_{k(i)}$ und 0 sonst. Der Algorithmus speichert somit die Summe der Anpassungen, die durch die Ungleichung $a_{k(i)} \cdot x \leq b_{k(i)}$ gemacht wurden, in $\lambda_{k(i)}$. Mit jedem Iterationsschritt wird $x^i$ entweder orthogonal auf die Lösungsmenge von $a_{k(i)} \cdot x^i = b^i$ projiziert oder der Einfluss des $k(i)$-ten Strahls wird vollständig herausgerechnet. Dies ist der Fall, wenn $c^i = \lambda^i_{k(i)}$. Dieser Ablauf ist im zweidimensionalen Fall in Abbildung 4.2, sowie in Abbildung 4.3 zu sehen. In letzterem entspricht die dritte Anpassung von $x^2$ auf $x^3$ genau dem Negativ der ersten, wodurch nur der Einfluss des zweiten Strahls übrig bleibt:

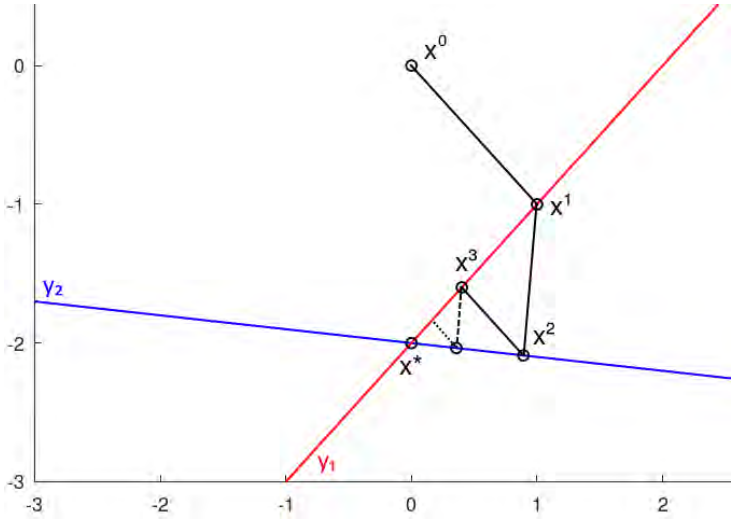



Abbildung 4.2: Iterationsschritte des Algorithmus mit Hildreths Prozess für vorheriges Problem aus Abbildung 4.1.

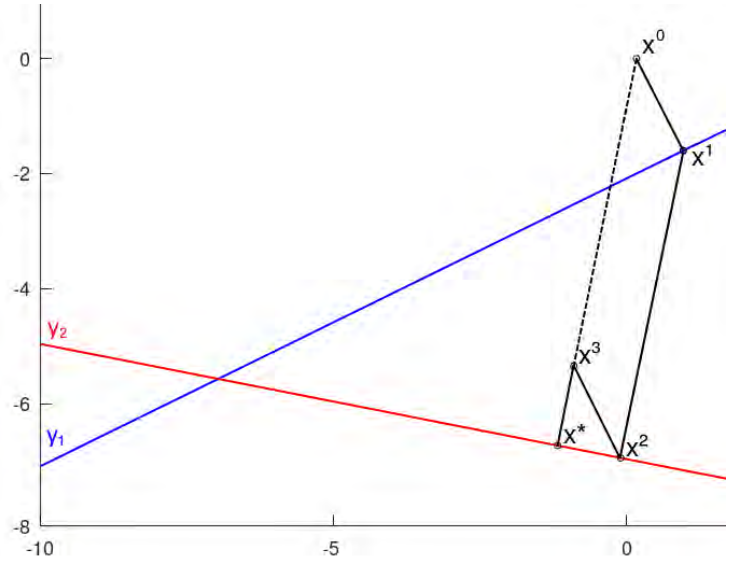



Abbildung 4.3: Iterationsschritte des Algorithmus mit Hildreths Prozess.

Der Algorithmus konvergiert beim zyklischen Durchlaufen der Systemmatrix $A$ gegen die Lösung mit minimaler Norm der Ungleichung $A \cdot x \leq b$. Als Vorraussetzung darf diese Lösungsmenge nicht leer sein und die Matrix $A$ darf keine Nullzeile enthalten.
Der Beweis hierzu ist in [LC80] zu finden, welcher im nächsten Abschnitt in abgewandelter Form wiedergegeben wird. Während des Beweises wird ein Resultat aus [Dan73] verwendet, welches auf Hoffmans Theorem [Hof52] basiert:

Sei $T = \{y | A \cdot y \leq d\}$ und $T' = \{y' | A \cdot y' \leq d'\}$ mit $T' \neq \emptyset$. Es existiert $a$, konstant, welches nur von $A$ abhängt, so dass für alle $y \in T$ ein $y' \in T'$ existiert, mit:

$$\|y - y'\| \leq a \cdot \left\|(d - d')^+\right\|.$$

Hier bedeutet für einen beliebigen Vektor $v$: $(v^+)_i = \max(0, v_i)$.

### 4.2.1 Konvergenzbeweis

Zuerst wird die mögliche Lösungsmenge mit $L = \{x | A \cdot x \leq b\}$, sowie die Lösung mit minimaler Norm mit $x^*$ benannt. Es soll gezeigt werden, dass $x^i \to x^*$, wenn $i \to \infty$. Dies ist der Fall, wenn $\|x^* - x^i\| \to 0$. Mittels Dreiecksungleichung kann nun folgende Ungleichung formuliert werden:

$$\left\|x^* - x^i\right\| \leq \left\|x^* - \hat{x}^i\right\| + \left\|x^i - \hat{x}^i\right\|. \tag{4.6}$$

Durch $\hat{x}^i$ wird dabei die orthogonale Projektion von $x^i$ auf $L$ bezeichnet. Es muss daher sowohl $\|x^* - \hat{x}^i\| \to 0$ als auch $\|x^i - \hat{x}^i\| \to 0$ bewiesen werden.

**Nicht Negativität von $\lambda$**

Es lässt sich leicht per Induktion zeigen, dass die Einträge von $\lambda^i$ nicht negativ sind. Ausgehend vom Ursprung $\lambda^0 = 0$: Aufgrund von (4.3) gilt, dass $c^i \leq \lambda^i_{k(i)}$, wodurch $\lambda^{(i+1)}_{k(i)} = \lambda^i_{k(i)} - c^i \geq 0$ folgt.

**Gleichheit $x^i = -A^T \cdot \lambda^i$**

Außerdem kann $x^i = -A^T \cdot \lambda^i$ durch Induktion gezeigt werden. Per Definition gilt natürlich $x^0 = -A^T \cdot \lambda^0$. Unter Beachtung von $a_{k(i)} = A^T \cdot e_{k(i)}$ folgt der Induktionsschritt:

$$x^{i+1} \overset{(4.4)}{=} x^i + c^i \cdot a_{k(i)} \overset{I.S.}{=} -A^T \cdot \lambda^i + c^i \cdot A^T \cdot e_{k(i)} \overset{(4.5)}{=} -A^T \cdot \lambda^{i+1}. \tag{4.7}$$

**Konvergenz von $\{c^i\}$ gegen 0**

Weiter kann $c^i \to 0$ bewiesen werden. Hierzu wird zunächst das zu beweisende Problem als primales Problem betrachtet

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|x\|^2 \text{ mit } A \cdot x \leq b,$$

mit dazugehörigem dualen Problem

$$\max_{\mu \in \mathbb{R}^m_+} \Phi(\mu) \text{ mit } \Phi(\mu) = \min_{x \in \mathbb{R}^n} L(x, \mu),$$

wobei

$$L(x, \mu) = \frac{1}{2} \sum_{j=1}^{n} x_j^2 + \sum_{k=1}^{m} \mu_k \cdot \left( \sum_{j=1}^{n} (a_{kj} \cdot x_j - b_k) \right)$$

die Lagrange Funktion ist. Somit folgt wegen $x^i = -A^T \cdot \lambda^i$, dass

$$\Phi(\lambda^i) = -\frac{1}{2} \left\| A^T \cdot \lambda^i \right\|^2 - \langle b, \lambda^i \rangle.$$

Der starke Dualitätssatz liefert nun den Zusammenhang:

$$\min_{A \cdot x \leq b} \frac{1}{2} \|x\|^2 = \max_{\lambda \in \mathbb{R}^m_+} \Phi(\lambda), \tag{4.8}$$

was auch zeigt, dass $\Phi(\mu)$ nach oben beschränkt ist.
Anschließend wird $\lim_{i \to \infty} (\Phi(\lambda^{i+1}) - \Phi(\lambda^i)) = 0$ bewiesen. Hierzu muss zunächst

die Differenz $\Phi(\lambda^{i+1}) - \Phi(\lambda^i)$ betrachtet werden, die folgendermaßen umgeschrieben werden kann:

$$\begin{aligned}
\Phi(\lambda^{i+1}) - \Phi(\lambda^i) &= -\frac{1}{2}\left\|A^T \cdot \lambda^{i+1}\right\|^2 - \langle b, \lambda^{i+1}\rangle - \Phi(\lambda^i) \\
&\overset{(4.7)}{=} -\frac{1}{2}\left\|x^{i+1}\right\|^2 - \langle b, \lambda^{i+1}\rangle - \Phi(\lambda^i) \\
&\overset{(4.4)}{=} -\frac{1}{2}\left\|x^i + c^i \cdot a_{k(i)}\right\|^2 - \langle b, \lambda^{i+1}\rangle - \Phi(\lambda^i) \\
&= -\frac{1}{2}\left\|x^i\right\|^2 - c^i \cdot \langle x^i, a_{k(i)}\rangle - \frac{1}{2}\left\|c^i \cdot a_{k(i)}\right\|^2 - \langle b, \lambda^{i+1}\rangle - \Phi(\lambda^i) \\
&\overset{(4.5)}{=} -c^i \cdot \langle x^i, a_{k(i)}\rangle - \frac{1}{2}\left\|c^i \cdot a_{k(i)}\right\|^2 - \langle b, \lambda^i - c^i \cdot e_{k(i)}\rangle + \langle b, \lambda^i\rangle \\
&= -c^i \cdot \langle x^i, a_{k(i)}\rangle - \frac{1}{2}\left\|c^i \cdot a_{k(i)}\right\|^2 + c^i \cdot \langle b, e_{k(i)}\rangle \\
&= -c^i \cdot \langle x^i, a_{k(i)}\rangle - \frac{1}{2}(c^i)^2 \cdot \left\|a_{k(i)}\right\|^2 + c^i \cdot b_{k(i)} \\
&= \left\|a_{k(i)}\right\|^2 \cdot c^i \cdot \left(\frac{b_{k(i)} - \langle a_{k(i)}, x^i\rangle}{\left\|a_{k(i)}\right\|^2} - \frac{c^i}{2}\right) \\
&\overset{(4.3)}{\geq} \left\|a_{k(i)}\right\|^2 \cdot c^i \cdot \left(c^i - \frac{c^i}{2}\right) \\
&\geq \frac{1}{2}\left\|a_{k(i)}\right\|^2 \cdot (c^i)^2 \geq 0.
\end{aligned}$$

Die Folge $\{\Phi(\lambda^i)\}$ ist also monoton steigend und nach oben beschränkt (4.8), wodurch $\lim_{i\to\infty}(\Phi(\lambda^{i+1}) - \Phi(\lambda^i)) = 0$ folgt.
Wird nun ein $a$ gewählt, für das $\left\|a_{k(i)}\right\|^2 \geq a > 0$ für alle $k(i)$ gilt, ergibt sich

$$\Phi(\lambda^{i+1}) - \Phi(\lambda^i) \geq \frac{1}{2} \cdot a \cdot (c^i)^2. \tag{4.9}$$

An dieser Stelle wird deutlich, weswegen vorausgesetzt wurde, dass $A$ keine Nullzeile enthalten darf. In diesem Fall würde nämlich kein solches $a$ existieren.
(4.8) hat schlussendlich $c^i \to 0$ zur Folge und zeigt, dass die Folge $\{x^i\}$ nach dem Cauchy-Kriterium konvergiert.

### Konvergenz von $b^i$ gegen $b$

Nun werden die gestörten Ergebnisvektoren $b^i = q^i + A \cdot x^i$, sowie $L^i = \{x | A \cdot x \leq b^i\}$ betrachtet. Die Folge $\{q^i\}$ wird dabei definiert durch $q^0 = 0 \in \mathbb{R}^m$

und

$$q_s^{i+1} = \begin{cases} q_s^i & \text{wenn } s \neq k(i) \\ -c^i \cdot \left\|a_{k(i)}\right\|^2 + b_{k(i)} - \langle a_{k(i)}, x^i \rangle & \text{wenn } s = k(i). \end{cases} \tag{4.10}$$

Da der zyklische Durchlauf der Matrix $A$ vorausgesetzt wurde, verändert sich $q_r^i$ genau einmal nach $m$ Iterationen. Es gilt also $q_r^i = q_r^{i-m+1}$ und somit auch $b_r^i = b_r^{i-m+1}$, falls $i \geq m$. Deutlich wird, dass für jeden Eintrag $s$ von $b^i$ gilt: $b_s^i \to b_s$ für $i \to \infty$. Hierzu wird zunächst $i$ so gewählt, dass $s = k(i)$ und $i > m$:

$$\begin{aligned} b_s^i &= q_s^i + \langle a_s, x^i \rangle \\ &= q_s^{i-m+1} + \langle a_s, x^i \rangle, \\ &\overset{(4.10)}{=} -c^{(i-m)} \cdot \|a_s\|^2 + b_s - \langle a_s, x^{(i-m)} \rangle + \langle a_s, x^i \rangle \\ &= -c^{(i-m)} \cdot \|a_s\|^2 + b_s + \langle a_s, x^i - x^{(i-m)} \rangle. \end{aligned}$$

Da $i$ gegen $\infty$ geht, geht auch $(i - m)$ gegen $\infty$. Dadurch fällt der erste Summand weg. Für den dritten Summanden gilt:

$$\langle a_s, x^i - x^{(i-m)} \rangle = \sum_{j=i-m}^{i-1} \langle a_s, x^{j+1} - x^j \rangle.$$

Da die Folge $\{x^i\}$ konvergiert, gilt $(x^{i+1} - x^i) \to 0$, wodurch obige Summe ebenfalls gegen 0 geht. Da $s$ beliebig gewählt werden kann, folgt somit $b^i \to b$, wenn $i \to \infty$.

### Konvergenz von $\|x^i - \hat{x}^i\|$ gegen 0

Wird nun für $T$ $L^i$, für $T'$ $L$ und für $y$ $x^i$ in das vor dem Beweis erwähnte Resultat aus [Dan73] eingesetzt, ergibt sich

$$\left\|x^i - \hat{x}^i\right\| \leq a \cdot \left\|(b^i - b)^+\right\|. \tag{4.11}$$

Um abschließend die Konvergenz des zweiten Summanden von (4.2) zu beweisen, wird die Konvergenz von $b^i$ gegen $b$ genutzt wodurch $\|x^i - \hat{x}^i\| \to 0$ für $i \to \infty$ folgt.

### Konvergenz von $\|x^* - \hat{x}^i\|$ gegen 0

Für die Konvergenz des ersten Summanden von (4.6), $\|x^* - \hat{x}^i\|$, wird gezeigt, dass $\hat{x}^i \to x^*$, wenn $i$ gegen $\infty$ geht. $L$ ist eine abgeschlossene konvexe Menge

und deswegen ist $x^* \in L$ der eindeutige Punkt in $L$ mit minimaler euklidischer Norm. Aus diesem Grund genügt es zu zeigen, dass $\|\hat{x}^i\| \to \|x^*\|$. Durch die Dreiecksungleichung gilt die Ungleichung:

$$\|x^*\| \leq \left\|\hat{x}^i\right\| \leq \left\|\hat{x}^i - x^i\right\| + \left\|x^i\right\|.$$

Da $\|\hat{x}^i - x^i\| \to 0$, existiert für jedes $\varepsilon_1 > 0$ ein hinreichend großes $i$, sodass $\|x^*\| \leq \|x^i\| + \varepsilon_1$ gilt.
Weiter optimiert wegen (4.8) der Punkt $x^i$ die Lagrange Funktion $L(x, \lambda^i)$, woraus für $i \to \infty$

$$\frac{1}{2}\left\|x^i\right\|^2 + \langle A \cdot x^i - b, \lambda^i\rangle \leq \frac{1}{2}\|x\|^2 + \langle A \cdot x - b, \lambda^i\rangle \quad \forall x \in \mathbb{R}^n$$

und

$$\frac{1}{2}\left\|x^i\right\|^2 \leq \frac{1}{2}\|x\|^2 + \langle A \cdot x, \lambda^i\rangle - \langle A \cdot x^i, \lambda^i\rangle \quad \forall x \in \mathbb{R}^n \tag{4.12}$$

folgt. Per Induktion kann gezeigt werden, dass $q_s^i \cdot \lambda_s^i = 0 \ \forall s \in \{1, ..., m\}$:

Induktionsanfang: Nach Initialisierung: $q_s^0 \cdot \lambda_s^0 = 0$
Induktionsschritt:
Falls $s \neq k(i)$ ist laut Definitionen $q_s^{i+1} = q_s^i$ und $\lambda_s^{i+1} = \lambda_s^i$, somit $q_s^{i+1} \cdot \lambda_s^{i+1} = 0$.
Falls $s = k(i)$:

$$\begin{aligned} q_s^{i+1} \cdot \lambda_s^{i+1} &= \left(-c^i \cdot \left\|a_{k(i)}\right\|^2 + b_i - \langle a_{k(i)}, x^i\rangle\right) \cdot \left(\lambda_{k(i)}^i - c^i\right) \\ &= \begin{cases} \left(-c^i \cdot \left\|a_{k(i)}\right\|^2 + b_i - \langle a_{k(i)}, x^i\rangle\right) \cdot 0 & \text{wenn } c^i = \lambda_k^i(i) \\ 0 \cdot \left(\lambda_{k(i)}^i - c^i\right) & \text{wenn } c^i \neq \lambda_k^i(i). \end{cases} \end{aligned}$$

Dadurch kann $+\langle q^i \cdot \lambda^i\rangle$ zu (4.12) hinzugefügt werden:

$$\begin{aligned} \frac{1}{2}\left\|x^i\right\|^2 &\leq \frac{1}{2}\|x\|^2 + \langle A \cdot x, \lambda^i\rangle - \langle A \cdot x^i + q^i, \lambda^i\rangle \\ &= \frac{1}{2}\|x\|^2 + \langle A \cdot x - b^i, \lambda^i\rangle \quad \forall x \in \mathbb{R}^n. \end{aligned}$$

Da alle Einträge von $\lambda^i$ nicht negativ sind und da $A \cdot x - b^i \leq 0$ für alle $x \in L^i$, folgt nun für alle $x \in L^i$: $\frac{1}{2}\|x^i\|^2 \leq \frac{1}{2}\|x\|^2$. Sei $x_*^i$ nun die orthogonale Projektion von $x^*$ auf $L^i$. Unter erneuter Anwendung von obigem Resultat aus [Dan73] bedeutet das wegen

$$\left\|x^i\right\| \leq \left\|x_*^i\right\| \leq \left\|x_*^i - x^*\right\| + \|x^*\|,$$

dass für jedes $\varepsilon_2 > 0$ ein $i$ existiert, sodass $\|x^i\| \leq \|x^*\| + \varepsilon_2$. Sowohl $\varepsilon_1$ als auch $\varepsilon_2$ wird für $i \to \infty$ beliebig klein und es folgt schlussendlich die Konvergenz $\|x^i\| \to \|x^*\|$. Aufgrund von $\|x^i - \hat{x}^i\| \to 0$ folgt auch die Konvergenz $\|\hat{x}^i\| \to \|x^*\|$ und dadurch $\hat{x}^i \to x^*$, was die Konvergenz des ersten Summanden von (4.6) zeigt und den Beweis abschließt.

# Kapitel 5

# Verrauschte CT Scans

Der Algorithmus aus dem letzten Kapitel kann nun verwendet werden, um $b - \epsilon \leq A \cdot x \leq b + \epsilon$ für $\epsilon \geq 0$ und $x$ mit minimaler Norm zu lösen.

## 5.1 Der ART4 Algorithmus

Der in diesem Abschnitt beschriebene Algorithmus, sowie dessen Herleitung, basiert auf [HL77]. Zunächst wird das Problem $b - \epsilon \leq A \cdot x \leq b + \epsilon$ in zwei Ungleichungen aufgeteilt. Diese können anschließend mit dem Verfahren aus dem vorherigen Kapitel gelöst werden. Die erste Ungleichung $A \cdot x \leq b + \epsilon$ hat die gleiche Form wie die Probleme aus dem letzten Kapitel, wohingegen die zweite Ungleichung $b - \epsilon \leq A \cdot x$ zuvor noch umgeformt werden muss:

$$b - \epsilon \leq A \cdot x \Leftrightarrow -b + \epsilon \geq -A \cdot x.$$

Gesucht ist also jenes $x$ mit minimaler Norm, für das sowohl $A \cdot x \leq b + \epsilon$ als auch $-A \cdot x \leq -b + \epsilon$ gilt.
Es ergibt sich somit folgender kombinierter Algorithmus, in dem beide Ungleichungen in einer Iteration abgearbeitet werden:

$$\begin{aligned} x^0 &= 0 \in \mathbb{R}^n \\ \lambda^{(+)0} &= 0 \in \mathbb{R}^m \\ \lambda^{(-)0} &= 0 \in \mathbb{R}^m \end{aligned}$$

Im Fall $\lambda_{k(i)}^{(+)i} \geq \lambda_{k(i)}^{(-)i}$ wird die Ungleichung $A \cdot x \leq b + \epsilon$ zuerst behandelt. Nachfolgend wird ausschließlich dieser Fall betrachtet, der umgekehrte Fall gilt analog, lediglich die Reihenfolge der Anpassungen dreht sich um. Der

Übersichtlichkeit halber wird nunmehr $k$ anstatt $k(i)$ geschrieben. Es gilt folgender Iterationsschritt:

$$c' = \min\left(\lambda_k^{(+)i}, \frac{(b_k+\epsilon) - \langle a_k, x^i\rangle}{\|a_k\|^2}\right) \tag{5.1}$$

$$x^{i+0.5} = x^i + c' \cdot a_k$$

$$\lambda^{(+)i+1} = \lambda^{(+)i} - c' \cdot e_k \tag{5.2}$$

$$\begin{aligned} c'' &= \min\left(\lambda_k^{(-)i}, \frac{(-b_k+\epsilon) + \langle a_k, x^{i+0.5}\rangle}{\|a_k\|^2}\right) \\ &= \min\left(\lambda_k^{(-)i}, \frac{(-b_k+\epsilon) + \langle a_k, (x^i + c' \cdot a_k)\rangle}{\|a_k\|^2}\right) \\ &= \min\left(\lambda_k^{(-)i}, \frac{(-b_k+\epsilon) + \langle a_k, x^i\rangle + c' \cdot \langle a_k, a_k\rangle}{\|a_k\|^2}\right) \\ &= \min\left(\lambda_k^{(-)i}, c' - \frac{(b_k-\epsilon) - \langle a_k, x^i\rangle}{\|a_k\|^2}\right) \end{aligned} \tag{5.3}$$

$$\begin{aligned} x^{i+1} &= x^{i+0.5} + c'' \cdot (-a_k) \\ &= x^i + (c' - c'') \cdot a_k \end{aligned} \tag{5.4}$$

$$\lambda^{(-)i+1} = \lambda^{(-)i} - c'' \cdot e_k \tag{5.5}$$

Wie in Kapitel 4 erwähnt, können sowohl $\lambda_k^{(+)i}$ als auch $\lambda_k^{(-)i}$ nicht negativ sein. Weiter kann mittels Induktion bewiesen werden, dass für alle $i$ entweder $\lambda_k^{(+)i} = 0$ oder $\lambda_k^{(-)i} = 0$ oder beides gilt:

Induktionsanfang: Nach Initialisierung: $\lambda^{(+)0} = 0$ und $\lambda^{(-)0} = 0$
Induktionsschritt:

1.Fall: $\lambda_k^{(+)i} \leq \frac{(b_k+\epsilon)-\langle a_k, x^i\rangle}{\|a_k\|^2}$

$\xrightarrow{(5.1)} c' = \lambda_k^{(+)i} \xrightarrow{(5.2)} \lambda_k^{(+)i+1} = 0$

2.Fall: $\lambda_k^{(+)i} > \frac{(b_k+\epsilon)-\langle a_k, x^i\rangle}{\|a_k\|^2}$

$\xrightarrow{(5.1)} c' = \frac{(b_k+\epsilon)-\langle a_k, x^i\rangle}{\|a_k\|^2}$

$\xrightarrow{(5.3)} c'' = \min\left(\lambda_k^{(-)i}, \frac{(b_k+\epsilon)-\langle a_k, x^i\rangle}{\|a_k\|^2} - \frac{(b_k-\epsilon)-\langle a_k, x^i\rangle}{\|a_k\|^2}\right)$

$\xrightarrow{\text{I.A.}} c'' = \min\left(0, \frac{(b_k+\epsilon)-\langle a_k, x^i\rangle}{\|a_k\|^2} - \frac{(b_k-\epsilon)-\langle a_k, x^i\rangle}{\|a_k\|^2}\right)$

$\rightarrow c'' = 0$, da $\epsilon \geq 0$ und somit $(b_k + \epsilon) \geq (b_k - \epsilon)$

$\xrightarrow{(5.5)} \lambda_k^{(-)i+i} = \lambda_k^{(-)i} = 0$

Beim finalen Algorithmus kann daher $\lambda^{(+)i}$ und $\lambda^{(-)i}$ zu einem Vektor $\lambda^i$ zusammen gefasst werden. Es muss beachtet werden, dass durch die Annahme

$\lambda_k^{(+)i} \geq \lambda_k^{(-)i}$ immer $\lambda_k^{(-)i} = 0$ gilt. Außerdem kann die Anpassung $(c' - c'')$ aus (5.4) weiter umgewandelt werden:

$$\begin{aligned} c' - c'' &= c' - \min\left(\lambda_k^{(-)i}, c' - \frac{(b_k - \epsilon) - \langle a_k, x^i \rangle}{\|a_k\|^2}\right) \\ &= c' + \max\left(0, -c' + \frac{(b_k - \epsilon) - \langle a_k, x^i \rangle}{\|a_k\|^2}\right) \\ &= \max\left(c', \frac{(b_k - \epsilon) - \langle a_k, x^i \rangle}{\|a_k\|^2}\right) \\ &= \max\left(\min\left(\lambda_k^{(+)i}, \frac{(b_k + \epsilon) - \langle a_k, x^i \rangle}{\|a_k\|^2}\right), \frac{(b_k - \epsilon) - \langle a_k, x^i \rangle}{\|a_k\|^2}\right) \end{aligned}$$

Abschließend lautet der Algorithmus, um die Lösung mit der kleinsten Norm vom Problem $b - \epsilon \leq A \cdot x \leq b + \epsilon$ zu finden, unter Beachtung, dass $(b_{k(i)} + \epsilon) > (b_{k(i)} - \epsilon)$:

$$\begin{aligned} x^0 &= 0 \in \mathbb{R}^n \\ \lambda^0 &= 0 \in \mathbb{R}^m \\ c^i &= \text{mid}\left(\lambda_{k(i)}^i, \frac{(b_{k(i)} + \epsilon) - \langle a_{k(i)}, x^i \rangle}{\left\|a_{k(i)}\right\|^2}, \frac{b_{k(i)} - \epsilon) - \langle a_{k(i)}, x^i \rangle}{\left\|a_{k(i)}\right\|^2}\right) \\ x^{i+1} &= x^i + c^i \cdot a_{k(i)} \\ \lambda_{k(i)}^{i+1} &= \lambda_{k(i)}^i - c^i \cdot e_{k(i)} \end{aligned}$$

Dieser Algorithmus wird mit ART4 bezeichnet.

### 5.1.1 Alternativer Algorithmus ohne Hildreths Prozess

Wird anstatt des Algorithmus für Ungleichungen aus Kapitel 4.2 jener aus Kapitel 4.1 genutzt, so entsteht ein weiterer Algorithmus um $b - \epsilon \leq A \cdot x \leq b + \epsilon$ zu lösen:

$$x^{i+1} = x^i + \begin{cases} 0, \text{ falls } (b_{k(i)} - \epsilon) \leq (a_{k(i)} \cdot x) \text{ und } (a_{k(i)} \cdot x) \leq (b_{k(i)} + \epsilon) \\ a_{k(i)} \frac{(b_{k(i)} - \langle a_{k(i)}, x^i \rangle}{\left\|a_{k(i)}\right\|^2}, \text{ sonst.} \end{cases}$$

Hier wird die Anpassung nur angewandt, falls eine der Ungleichungen nicht erfüllt ist. Sonst unterscheidet sich dieser Algorithmus nicht vom ART Algorithmus. Fortlaufend wird dieser Algorithmus als ART4′ bezeichnet und er wird nachfolgend verwendet, um ihn mit dem ART4 Algorithmus zu vergleichen.

## 5.2 Optimale Toleranz

Um die optimale Toleranz zu ermitteln, wurden die Ergebnisvektoren von zwei verschiedenen Datensätzen mit 3%, 5% und 10%igem Rauschen versehen. Der erste Datensatz [AMJ+17] besteht aus 14 835 Strahlen und liefert dabei ein Bild mit Dimension $128 \times 128$. Der zweite Datensatz [KLA+15] beinhaltet 19 680 und ergibt ein $164 \times 164$ Bild. Anschließend wurden die Ergebnisse von ART4 und ART4′ mit verschiedenen Toleranzen berechnet, sowie die Fehler zum rauschfreien Ergebnis. Als Fehler wurde die euklidische Norm verwendet. Zu sehen ist außerdem das Ergebnis, welches der normale ART Algorithmus bei den verrauschten Daten liefert. Für alle Berechnungen wurde die Systemmatrix zyklisch durchlaufen. Allerdings haben die im Zuge der Arbeit gemachten Tests ergeben, dass für verrauschte Datensätze ein zufälliges Durchlaufen in der Regel zu einem niedrigeren Fehler konvergiert. Da es in diesem Abschnitt jedoch nur um die Ermittlung der optimalen Toleranzen geht, wurde aufgrund von Reproduzierbarkeit erneut die zyklische Variante gewählt.
Zur besseren Erkennbarkeit wurde bei allen Bildern der Kontrast erhöht.

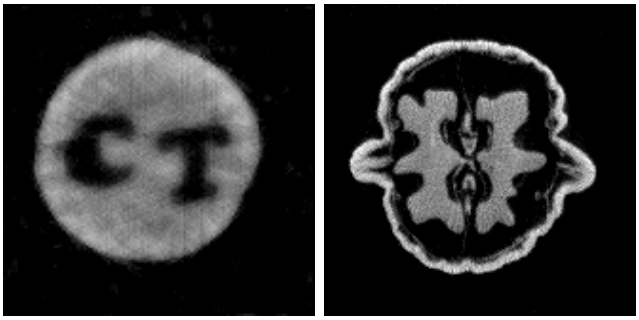


Abbildung 5.1: Referenzbilder: Erster Datensatz (links), zweiter Datensatz (rechts)

Das Rauschen wurde in allen Fällen mittels des folgenden Codes in MATLAB auf den Ergebnisvektor $b$ gelegt:

```
b = b + GradDesRauschensInProzent*randn(size(b));
```

### 5.2.1 3% Rauschen

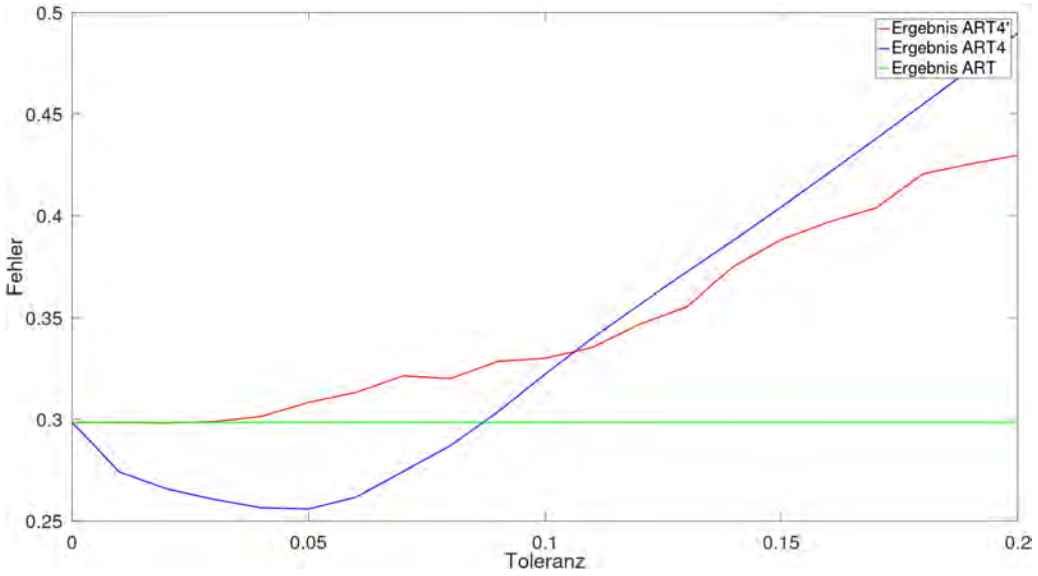



Abbildung 5.2: Fall: 3% Rauschen auf dem Datensatz [AMJ+17]. ART liefert einen Fehler von $\approx 0.299$. Optimales Ergebnis von ART4: Toleranz $= 0.05$, Fehler $\approx 0.256$; Optimales Ergebnis von ART4′: Toleranz $= 0.02$, Fehler $\approx 0.298$.

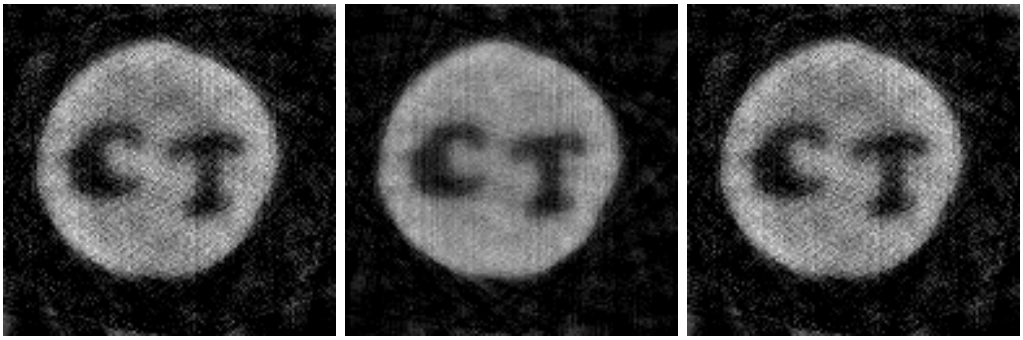


Abbildung 5.3: Ergebnis des ART Algorithmus (links), optimales Ergebnis des ART4 Algorithmus (mitte) und des ART4′ Algorithmus (rechts).

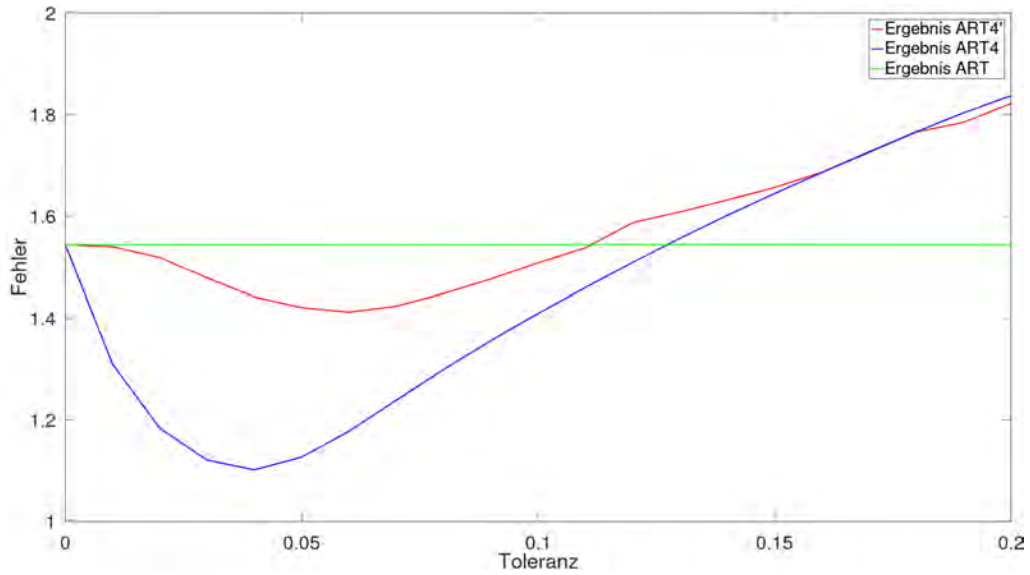



Abbildung 5.4: Fall: 3% Rauschen auf dem Datensatz [KLA+15]. ART liefert einen Fehler von $\approx 1.54$. Optimales Ergebnis von ART4: Toleranz $= 0.04$, Fehler $\approx 1.10$; Optimales Ergebnis von ART4′: Toleranz $= 0.06$, Fehler $\approx 1.41$.

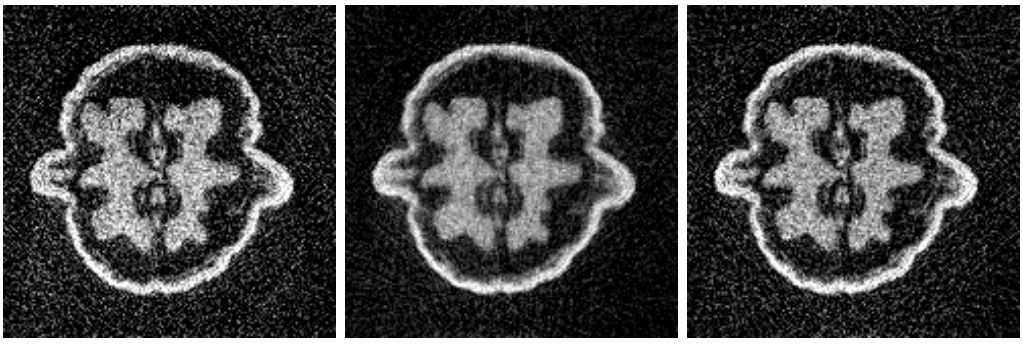


Abbildung 5.5: Ergebnis des ART Algorithmus (links), optimales Ergebnis des ART4 Algorithmus (mitte) und des ART4′ Algorithmus (rechts).

### 5.2.2 5% Rauschen

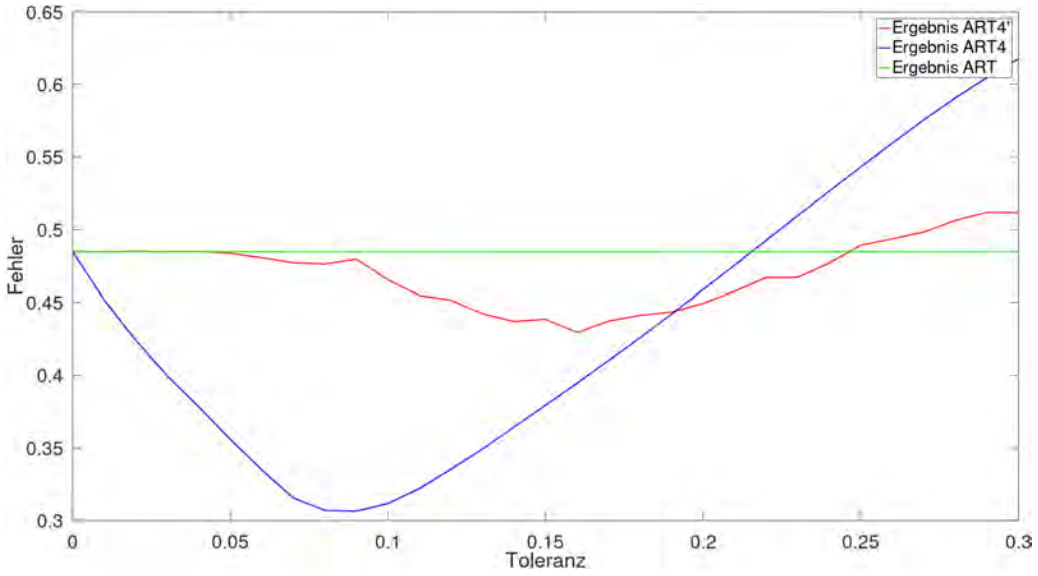



Abbildung 5.6: Fall: 5% Rauschen auf dem Datensatz [AMJ+17]. ART liefert einen Fehler von $\approx 0.48$. Optimales Ergebnis von ART4: Toleranz $= 0.09$, Fehler $\approx 0.31$; Optimales Ergebnis von ART4′: Toleranz $= 0.16$, Fehler $\approx 0.43$.

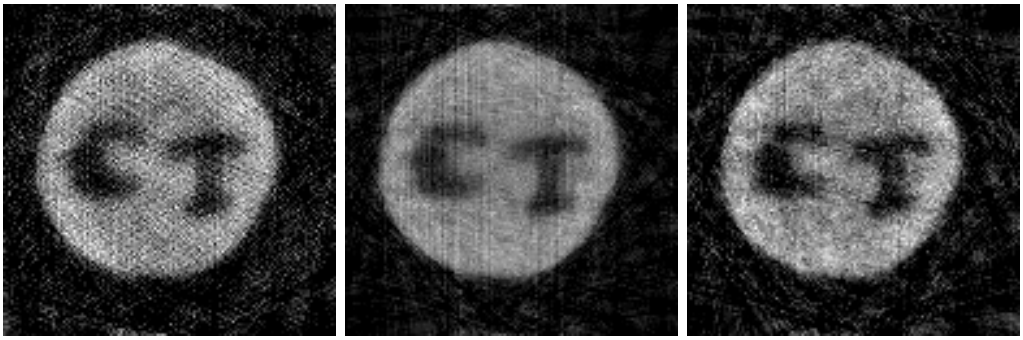


Abbildung 5.7: Ergebnis des ART Algorithmus (links), optimales Ergebnis des ART4 Algorithmus (mitte) und des ART4′ Algorithmus (rechts).

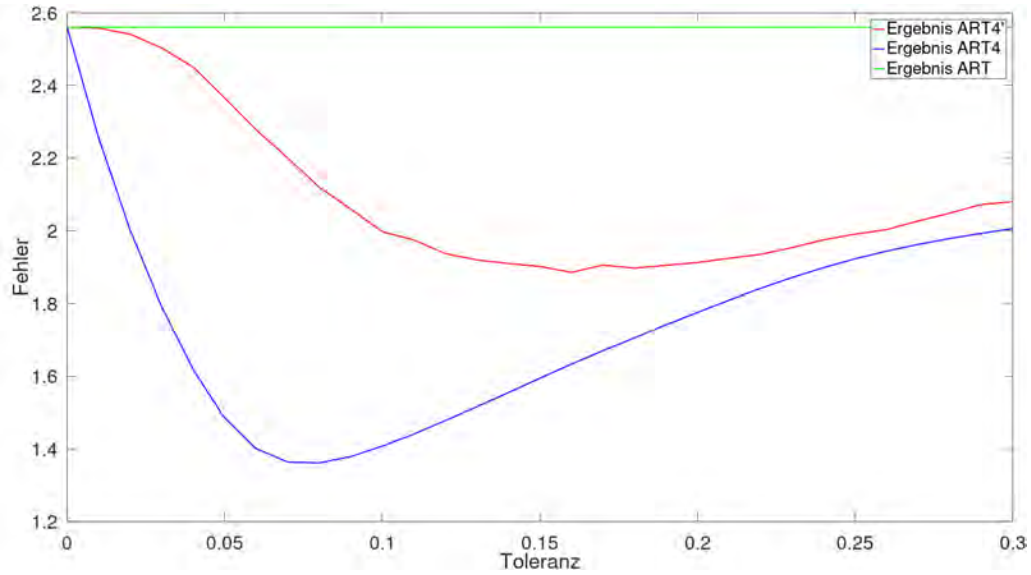



Abbildung 5.8: Fall: 5% Rauschen auf dem Datensatz [KLA$^+$15]. ART liefert einen Fehler von $\approx 2.56$. Optimales Ergebnis von ART4: Toleranz $= 0.08$, Fehler $\approx 1.36$; Optimales Ergebnis von ART4$'$: Toleranz $= 0.16$, Fehler $\approx 1.88$.

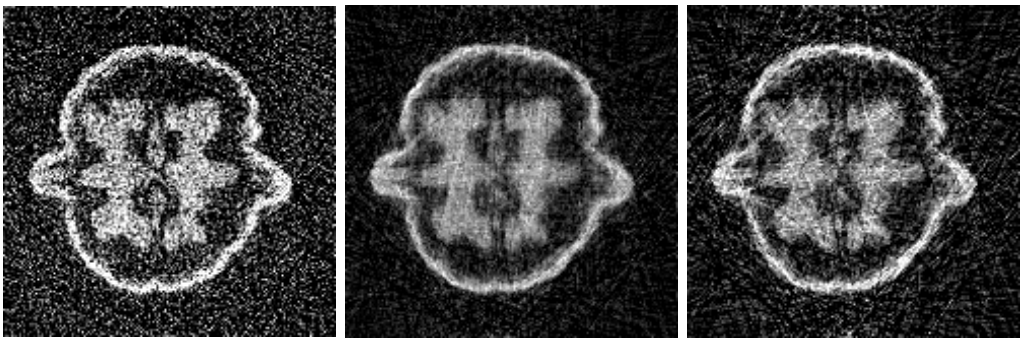


Abbildung 5.9: Ergebnis des ART Algorithmus (links), optimales Ergebnis des ART4 Algorithmus (mitte) und des ART4$'$ Algorithmus (rechts).

### 5.2.3 10% Rauschen

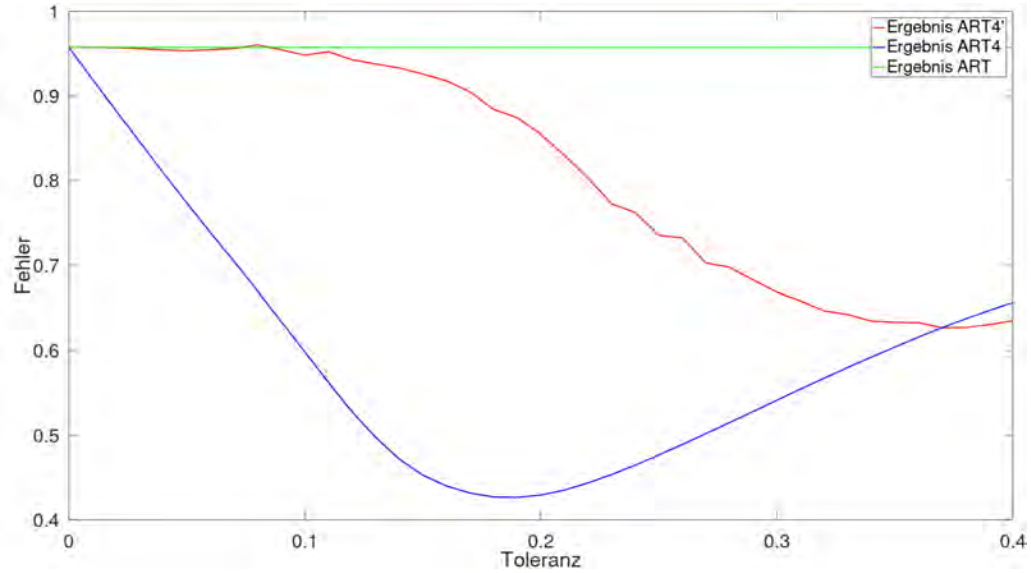



Abbildung 5.10: Fall: 10% Rauschen auf dem Datensatz [AMJ+17]. ART liefert einen Fehler von $\approx 0.96$. Optimales Ergebnis von ART4: Toleranz $= 0.19$, Fehler $\approx 0.43$; Optimales Ergebnis von ART4′: Toleranz $= 0.37$, Fehler $\approx 0.63$.

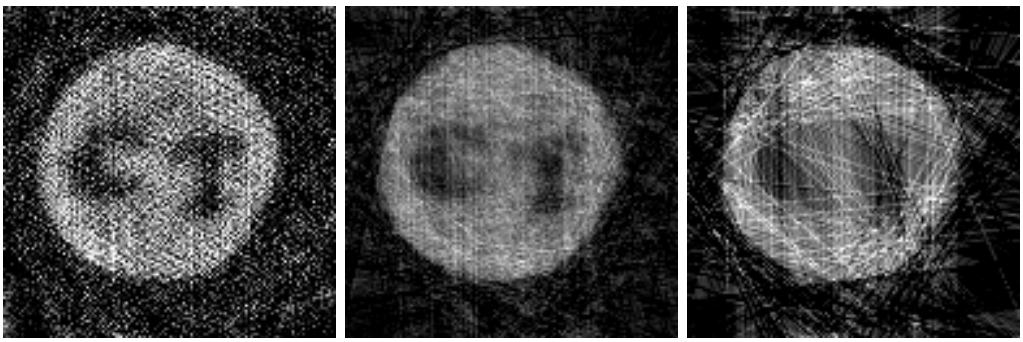


Abbildung 5.11: Ergebnis des ART Algorithmus (links), optimales Ergebnis des ART4 Algorithmus (mitte) und des ART4′ Algorithmus (rechts).

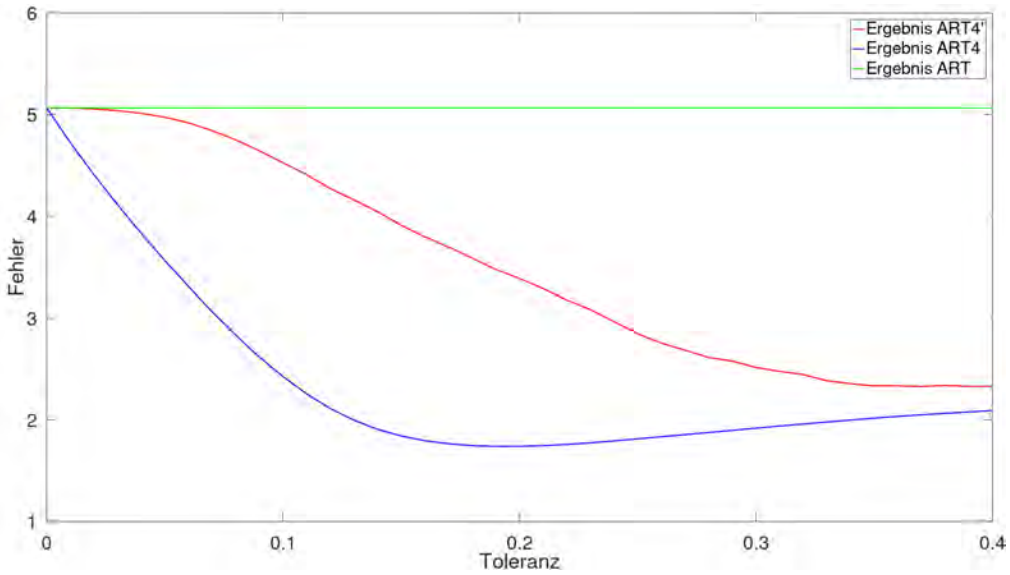



Abbildung 5.12: Fall: 10% Rauschen auf dem Datensatz [KLA+15]. ART liefert einen Fehler von $\approx 5.07$. Optimales Ergebnis von ART4: Toleranz $= 0.19$, Fehler $\approx 1.74$; Optimales Ergebnis von ART4′: Toleranz $= 0.37$, Fehler $\approx 2.33$.

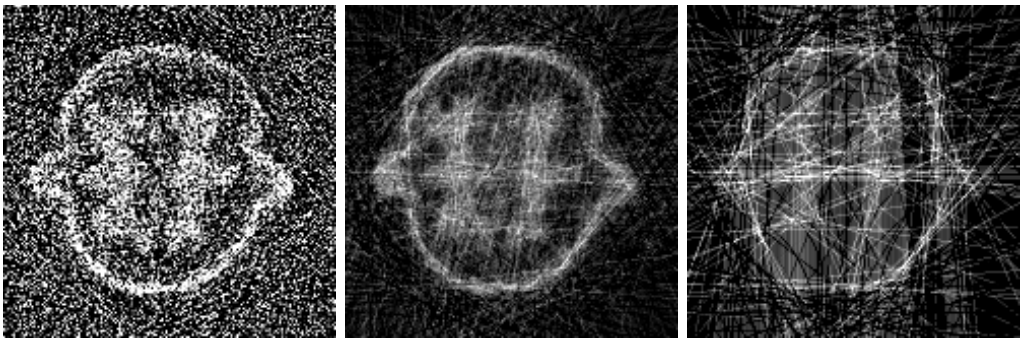


Abbildung 5.13: Ergebnis des ART Algorithmus (links), optimales Ergebnis des ART4 Algorithmus (mitte) und des ART4′ Algorithmus (rechts).

### 5.2.4 Zwischenfazit

Der ART4 Algorithmus liefert in allen Beispielen einen signifikant niedrigeren Fehler als der ART, sowie der ART4′ Algorithmus, was sich auch visuell wiederspiegelt. Außerdem wird eine geringere Toleranz als bei ART4' benötigt, um den minimalen Fehler zu erreichen. Bei 3%igem Rauschen ist in beiden Beispielen eine Toleranz von etwa 0.05 optimal, bei 5%igem Rauschen eine

Toleranz von 0.08 und bei 10%igem Rauschen liefert eine Toleranz von circa 0.19 das optimale Ergebnis. Es scheint also als wäre der Grad des Rauschens weitestgehend ausschlaggebend für die optimale Toleranz.

## 5.3 Test mittels Peak Signal to Noise Ratio (PSNR)

Die Überlegenheit des ART4 Algorithmus gegenüber des ART4' Algorithmus wird desweiteren durch einen Test mittels PSNR gezeigt. Dazu wurde der zweite Datensatz [KLA+15] mit allen Graden des Rauschens zwischen 1% und 20% belegt und für jeden Grad das optimale Ergebnis des ART4, des ART4' und des ART Algorithmus berechnet und die jeweilige PSNR berechnet. Bis auf den 1%igen Fall liefert ART4 eine deutlich geringere PSNR als ART4' oder ART.

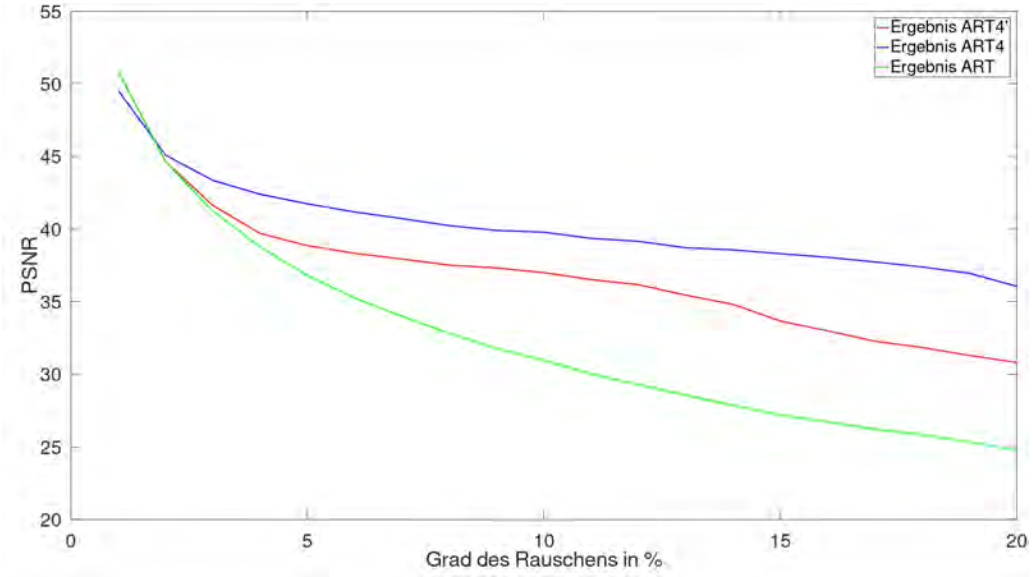



Abbildung 5.14: Vergleich PSNR der optimalen Ergebnise des ART4', ART4 und ART Algorithmus für 1% bis 20%iges Rauschen auf Datensatz [KLA+15].

## 5.4 Implementierung

Die Implementierung des ART4 Algorithmus erfolgt ähnlich zu der des ART Algorithmus. Auch hier gibt es eine Matrix, um eine Übersicht über die nie

besuchten Pixel zu haben. Aufgrund des erwarteten Rauschens der eingegebenen Daten werden hier allerdings bereits alle Strahlen ignoriert, bei denen mehr als 95% absorbiert werden.

```
function x = solveART4(A,b,tol)
  m = size(A)(1);
  n = size(A)(2);
  x = zeros(n,1);
  flags = zeros(size(x));
  lambda = zeros(m,1);
  for i = 0:(3*m)
      ki = mod(i,m) + 1;
      bki = b(ki);
      aki = A(ki,:);
      if(bki > 0.95)
          continue;
      end
      flags(aki!=0) = 1;
      c = median([lambda(ki)
                  (bki + tol) - aki*sol)/(aki*aki')
                  (bki - tol) - aki*sol)/(aki*aki')]);
    sol = sol + c*aki';
    lambda(ki) = lambda(ki) - c;
  end
  x(flags==0) = 1;
end
```

# Kapitel 6

# Beispiel: Lotusfrucht

In diesem Kapitel wird ein neuer Datensatz [AAS+16] verwendet. Das Ziel hierbei ist, die im Laufe der Arbeit gewonnenen Ergebnisse auf ihre Güte zu prüfen. Das gescannte Objekt ist eine aufgeschnittene Lotusfrucht, in deren Löcher ein Bleistift, Keramikstücke, Streichhölzer und Kreide gelegt wurden. All diese Gegenstände sind aus verschiedenen Materialien und unterschiedlich geformt.
Der dazugehörige Datensatz besteht aus 51 480 X-Ray Strahlen und liefert ein $256 \times 256$ großes Bild. Auch in allen nachfolgenden Bildern wurde zur besseren Erkennbarkeit der Kontrast erhöht. Anders als bei den vorherigen Tests zur Ergebnisgewinnung, in denen die Systemmatrix zyklisch durchlaufen wurde, wurde hier bei jeder Iteration eine zufällige Zeile der Systemmatrix mit Zurücklegen gewählt.
Der ART Algorithmus liefert mit diesen Daten ein Ergebnis mit einem Fehler von 75.62. Das Bild ist größtenteils unkenntlich und mit grauen Linien durchzogen und lediglich der Umriss der Lotusfrucht und die Kreide können erahnt werden.

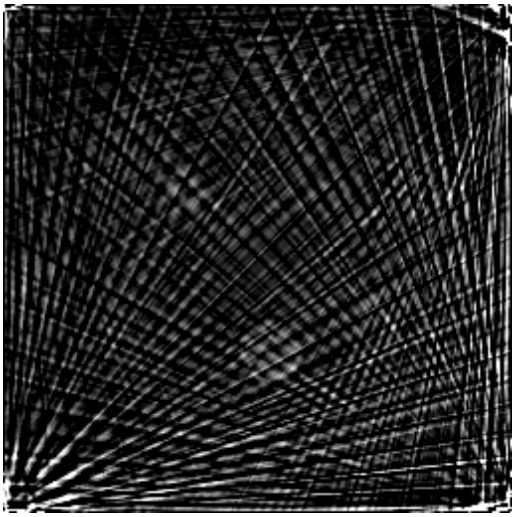


Abbildung 6.1: Ergebnis des ART Algorithmus.

Werden alle Strahlen i ignoriert, mit einem Eintrag im Ergebnisvektor $b$ von $b_i \leq 0.1$, was 21 600 Strahlen, also circa der Hälfte aller Strahlen, entspricht, besitzt das Resultat vom ART Algorithmus immerhin einen Fehler von 32.63 und die Lotusfrucht und die Objekte in ihren Löchern sind gut erkennbar.

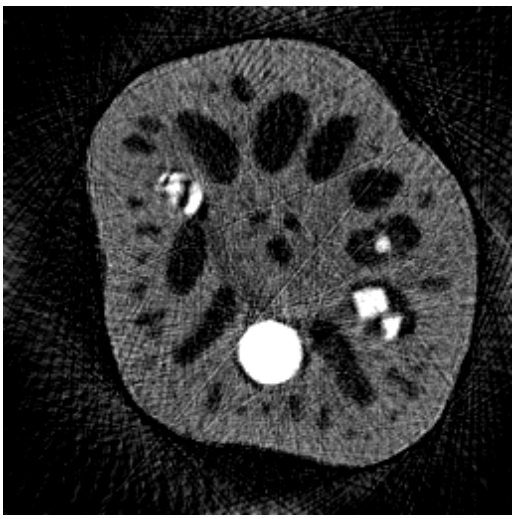


Abbildung 6.2: Ergebnis des ART Algorithmus ohne die Strahlen mit $b_i \leq 0.1$.

Das zyklische Durchlaufen der Systemmatrix liefert ähnliche Ergebnisse. Die Daten sind also in irgendeiner Weise verrauscht oder fehlerhaft und der

Vektor, der dem gescannten Objekt entsprechen würde, ist nicht die Lösung mit minimaler Norm von $A \cdot x = b$. Der ART4 Algorithmus hingegen liefert mit einer Toleranz von 0.02 einen Fehler von 3.60, ohne einzelne Strahlen zu ignorieren. Hier wurde die gescannte Lotusfrucht sehr gut rekonstruiert.

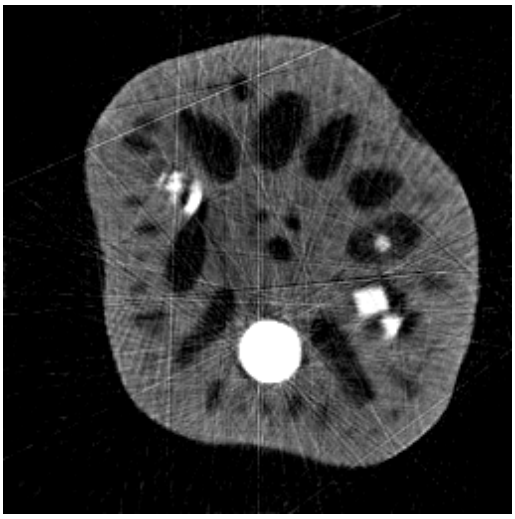


Abbildung 6.3: Ergebnis des ART4 Algorithmus mit 0.02 Toleranz.

Nach manuellem verrauschen der Daten von 3% beträgt der Fehler des ART4 Algorithmus 8.98. Als Toleranz wurde 0.07 gewählt. Dies entspricht der Summe aus den 0.02 für die nicht manuell verrauschten Daten und der optimalen Toleranz von 0.05, die in Kapitel 5.2.1 für 3%iges Rauschen ermittelt wurde. Eine Toleranz von lediglich 0.05 liefert jedoch ein ähnliches Ergebnis wie jenes aus Abbildung 6.1.

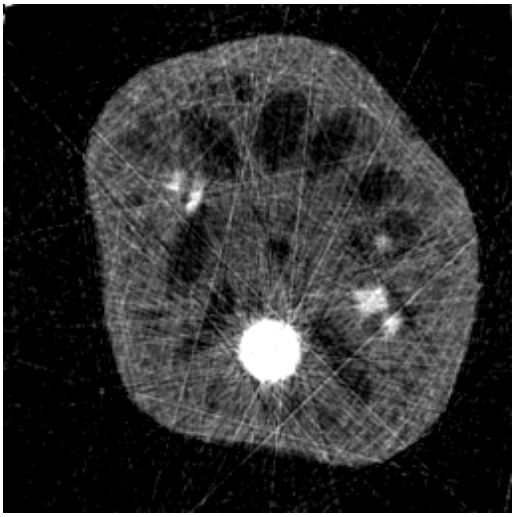


Abbildung 6.4: Ergebnis des ART4 Algorithmus mit 0.07 Toleranz. Die Daten wurden mit 3% verrauscht.

Mit dem selben Prinzip wurden die Toleranzen für 5%iges und 10%iges Rauschen ermittelt. Demnach beträgt die Toleranz für 5%iges Rauschen 0.11 mit einem Fehler von 13.4 und im 10%igen Fall resultiert mit einer Toleranz von 0.22 ein Fehler von 24.75. Es kann festgehalten werden, dass die Rekonstruktion des ART4 Algorithmus auch auf diesen Daten sehr gut funktioniert, sowie dass alle erhaltenen Bilder gut erkennbar sind.

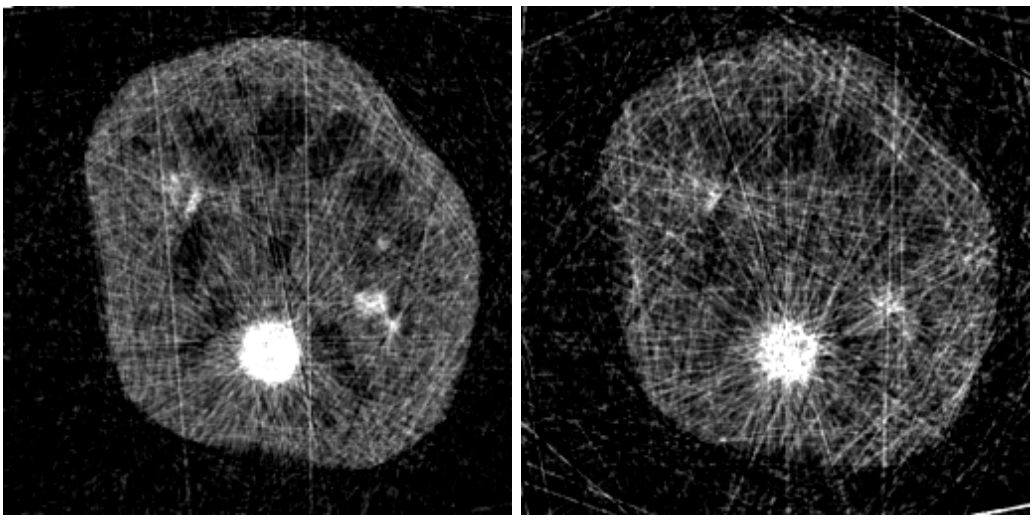


Abbildung 6.5: Ergebnis des ART4 Algorithmus für 5%ig (links) und 10%ig (rechts) verrauschte Daten.

# Kapitel 7

# Fazit

Mit dem in dieser Arbeit hergeleiteten ART Algorithmus können CT Scans gut rekonstruiert werden, wobei der Fehler zum eingescannten Objekt sehr gering ist. Der Algorithmus konvergiert dabei zur Lösung mit minimaler Norm vom Gleichungssystem $A \cdot x = b$, wobei $A$ die Systemmatrix und $b$ der Ergebnisvektor des Scans ist. Nach spätestens dreimaligem Betrachten aller X-Ray Strahlen kann der Algorithmus beendet werden. Bei zufälligem Durchlaufen der Strahlen genügen sogar ein Viertel aller Strahlen, um das Objekt bereits erkennen zu können. Es werden also nicht viele Iterationen zur Rekonstruktion benötigt.
Für verrauschte Scans eignet sich der ART4 Algorithmus. Dabei scheint die optimale Toleranz fast ausschließlich vom Grad des Rauschens abzuhängen. Diese kann somit im Vorhinein ermittelt werden. Der Algorithmus konvergiert ebenso gegen die Lösung mit minimaler Norm innerhalb dieser Toleranz. Die Konvegenzgeschwindigkeit ist ähnlich zu der des ART Algorithmus.
Als genereller Kritikpunkt muss allerdings aufgeführt werden, dass sowohl die Konvergenzgeschwindigkeit als auch die optimalen Toleranzen auf sehr wenigen Datensätzen ermittelt und getestet wurden. Der Grund dafür ist die fehlende Verfügbarkeit von genügend frei zugänglichen Datensätzen.
Als kleiner Ausblick kann das gewichtete ART genannt werden. Dieses Thema wurde in Kapitel 3.8 bereits kurz angeschnitten. Eine solche Gewichtung kann auch in den ART4 Algorithmus implementiert werden und eventuell ist mit Hilfe dieser die Laufzeit weiter optimierbar. Dazu muss allerdings vermutlich eine vom Index der aktuellen Iteration abhängige Gewichtung gewählt werden. Alle im Zuge vorliegender Arbeit gemachten Tests konnten nachweislich keine generelle Verbesserung der Konvergenzgeschwindigkeit für fixe Gewichtungen ausmachen.

# Literaturverzeichnis


[AAS+16] A., Bubba T. ; Andreas, Hauptmann ; Simo, Huotari ; Juho, Rimpeläinen ; Samuli, Siltanen: *Tomographic X-ray data of a lotus root filled with attenuating objects (Version 1.0.0)[Data set] Zenodo.* https://zenodo.org/record/1254204, 2016

[AMJ+17] A., Bubba T. ; Markus, Juvonen ; Jonatan, Lehtonen ; Maximilian, März ; Alexander, Meaney ; Zenith, Purisha ; Samuli, Siltanen: *Tomographic X-ray data of carved cheese (Version 1.0.0)[Data set] Zenodo.* https://zenodo.org/record/1254210, 2017

[Dan73] Daniel, James W.: *On perturbations of systems of linear inequalities.* SIAM Journal on Numerical Analysis 10, 1973

[Han] Hansen, Per C.: *Algebraic Methods for Computed Tomography.* http://people.compute.dtu.dk/pcha/HDtomo/Day1algebraic.pdf [zuletzt geprüft: 07.09.2020]

[Her73] Herman, Gabor T.: *A relaxation method for reconstructing objects from noisy x-rays.* Mathematic Programming Study 8, 1973

[Hil57] Hildreth, Clifford: *A quadratic programming procedure.* Naval Research Logistics Quarterly, 1957

[HL76] Herman, Gabor T. ; Lent, Arnold: *Iterative Reconstruction Algorithms*, 1976

[HL77] Herman, Gabor T. ; Lent, Arnold: *A Family of iterative quadratic optimization algorithms for pairs of inequalities, with appliation in diagnostic radiology.* Mathematic Programming Study 9, 1977

[HLR73] Herman, Gabor T. ; Lent, Arnold ; Rowland, Stuart W.: *ART: Mathematics and Applications A Report on the Mathematical Foundations and on the Applicability to Real Data of the Algebraic Reconstruction Techniques*, 1973

[Hof52] Hoffman, Alan J.: *On approximate solutions of systems of linear inequalities.* Journal of Research of the National Bureau of Standards 49, 1952

[Kac37] Kaczmarz, Stefan: *Angenäherte Auflösung von Systemen linearer Gleichungen*, 1937

[KLA+15] Keijo, Hämäläinen ; Lauri, Harhanen ; Aki, Kallonen ; Antti, Kujanpää ; Esa, Niemi ; Samuli, Siltanen: *Tomographic X-ray data of a walnut (Version 1.0.0)[Data set] Zenodo.* https://zenodo.org/record/1254206, 2015

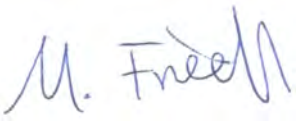
[LC80] Lent, Arnold ; Censor, Yair: *Extensions of Hildreth's Row-Action Method for quadratic Programming.* SIAM Jornal on Control and Optimization 18, 1980

[Lue69] Luenberger, David G.: *Optimization by Vector Space Methods*, 1969

# Erklärung

Ich erkläre hiermit eidesstattlich, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Passau, 10. September 2020



Maximilian Friedl