

PRAKTIKUM NUMERISCHE SIMULATION IN DER TECHNIK

Computertomographie (CT)

Stichworte: Medizinische Bildgebung, algebraische Rekonstruktion, gefilterte Rückprojektion

Betreuer: Richter

Termine: WT oder FT

Thema: Computertomographie (CT) rekonstruiert das Innere eines Objekts durch Messung der Abschwächung von haarfeinen Röntgenstrahlen, die in vielen Richtungen und Abständen durch das Objekt gesendet werden. Abbildung 1, links, zeigt einen ebenen Querschnitt durch einen Körper, unterschiedliche Gewebedichten sind durch unterschiedliche Graustufen farblich markiert – je dunkler die Farbe, desto dichter das Gewebe. Die Gewebedichte wird durch eine Funktion

$$f : D \rightarrow \mathbb{R}, \quad x \mapsto f(x),$$

modelliert, wobei $D := \{x \in \mathbb{R}^2; \|x\|_2 < 1\} \subseteq \mathbb{R}^2$ die Einheitskreisscheibe ist. Man darf sich, wo immer das im Folgenden günstig ist, f als eine auf ganz \mathbb{R}^2 definierte Funktion vorstellen, welche außerhalb von D gleich null ist. Ebenfalls gezeichnet ist ein Röntgenstrahl L , der den Körper längs einer Geraden von einer Röntgenquelle zum Photonendetektor durchdringt. Der gemessene gesamte Intensitätsverlust längs L entspricht einem Kurvenintegral (Wegintegral)

$$\int_L f(x) ds. \quad (1)$$

(Interpretation: Intensitätsverlust des Röntgenstrahls im Punkt $x \in D$ ist proportional zur Gewebedichte $f(x)$; Summierung der Intensitätsverluste längs des Strahls.) Parametrisiert man den Strahl L in der Form

$$L = L(\varphi, s) = \{s\theta + t\theta^\perp; t \in \mathbb{R}\} \quad \text{für } \varphi \in [0, \pi) \quad \text{und } s \in (-1, 1) \quad (2)$$

(Aufpunkt $s\theta \in D$, Richtungsvektor θ^\perp), wobei die Abkürzungen

$$\theta := \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix} \quad \text{sowie} \quad \theta^\perp := \begin{pmatrix} -\sin \varphi \\ \cos \varphi \end{pmatrix} \quad (3)$$

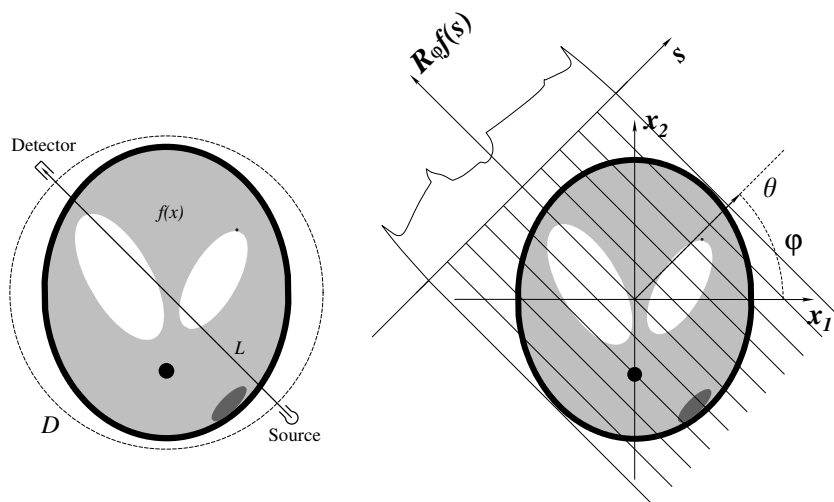


Abbildung 1: Prinzip der CT

fest vereinbart werden, dann lässt sich das Integral (1) in der Form

$$Rf(\varphi, s) := R_\varphi f(s) := \int_{-\infty}^{\infty} f(s\theta + t\theta^\perp) dt \quad (4)$$

schreiben. Weil die Funktion f außerhalb der Einheitskreisscheibe gleich null ist, ist das Integral nur scheinbar uneigentlich und könnte genauso mit endlichen Integrationsgrenzen in der Form

$$\int_{-\sqrt{1-s^2}}^{\sqrt{1-s^2}} f(s\theta + t\theta^\perp) dt$$

geschrieben werden. Rechts in Abbildung 1 sind alle Integralwerte (4) (Intensitätsverluste von Röntgenstrahlen), welche zu einem festen Winkel $\varphi \in [0, \pi)$ gehören, als Funktion in s angetragen. Dort, wo der Röntgenstrahl besonders dichtes (schwarz gezeichnetes) Gewebe durchdringt und dementsprechend viel Intensität verliert, sind lokale Maxima der Funktion $s \mapsto R_\varphi(s)$ zu verzeichnen. Die Funktion

$$Rf : [0, \pi) \times (-1, 1) \rightarrow \mathbb{R}, \quad (\varphi, s) \mapsto Rf(\varphi, s), \quad (5)$$

heißt **Radontransformierte** von f . Es ist die Aufgabe der Computertomographie, aus Kenntnis der Funktion Rf (aus der Kenntnis von Intensitätsverlusten) die Funktion f zu rekonstruieren.

Anwendungsfall: Rekonstruktion eines Querschnittsbilds. Ein vielfach benutztes Bild zum Testen von Verfahren der CT ist das sogenannten **Phantombild von Shepp und Logan**, siehe Abbildung 2. Wie in der CT üblich, ist hier die Farbgebung gegenüber Abbildung 1 gerade invertiert, das heißt das dargestellte Gewebe wird desto heller dargestellt, je dichter es ist. Ein CT-Scanner liefere für feste Parameter $p, q \in \mathbb{N}$ die $2pq$ Messwerte

$$g(\varphi_j, s_\ell) := Rf(\varphi_j, s_\ell) \quad (6)$$



Abbildung 2: Phantombild f (Shepp-Logan)

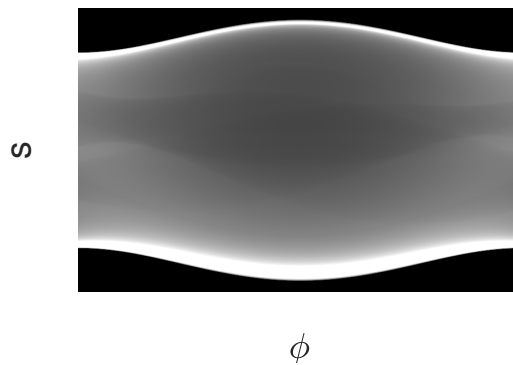


Abbildung 3: Radontransformierte Rf des Phantombilds

an den Stellen

$$\varphi_j = \frac{j}{p} \cdot \pi, \quad s_\ell = \frac{\ell}{q}, \quad j = 0, \dots, p-1, \quad \ell = -q, \dots, q-1, \quad (7)$$

welche in Abbildung 3 dargestellt sind. In den folgenden Aufgaben 1 bis 4 soll es Schritt für Schritt unternommen werden, das Phantombild f aus den gegebenen Messwerten (6)-(7) zu rekonstruieren. Dabei wird wie folgt vorgegangen:

- Es wird die exakte Radontransformierte $g = Rf$ des Phantombilds von Shepp und Logan berechnet.
- Die unbekannte Funktion f wird als eine stückweise konstante Funktion (ein „Pixelbild“) angesetzt. Zur Bestimmung der unbekanntenen Koeffizienten wird ein lineares Gleichungssystem aufgestellt.
- Es werden Verfahren zur Lösung des linearen Gleichungssystems ausprobiert und diskutiert.

Dieses Verfahren heißt in der Computertomographie **algebraische Rekonstruktionstechnik (ART)**. In den Aufgaben 5 bis 7 wird die in der heutigen Praxis bevorzugte Methode der **gefilterten Rückprojektion** besprochen.

1. Aufgabe. Sie bekommen zwei Matlab-Funktionen zur Verfügung gestellt: Die Funktion

```
function P = sheppLoganP(N)
```

generiert ein Bild $P \in \mathbb{R}^{N,N}$ der Auflösung $N \times N$ Pixel, welches dem Phantombild von Shepp und Logan entspricht. Die Funktion

```
function R = sheppLoganR(p,q)
```

berechnet ein Matrix $R \in \mathbb{R}^{2q,p}$, wobei

$$R(j+1, q+\ell+1) = Rf(\varphi_j, s_\ell), \quad j = 0, \dots, p-1, \quad \ell = -q, \dots, q-1, \quad \text{gemäß (6)}.$$

(Bei Interesse können Sie analytische Formeln für die Berechnung von Rf im 3. Kapitel des Buchs *Principles of Computerized Tomographic Imaging* von Kak und Slaney nachlesen, siehe dort die Seiten 53 ff.) Testen Sie diese beiden Funktionen für $N = 512$, $p = 800$ und $q = 256$ und generieren Sie Bilder wie in den Abbildungen 2 und 3. Benutzen Sie hierzu die Matlab-Funktion `imshow`. **Bemerkung:** Beim Aufruf `imshow(P)` wird P als Graustufenbild interpretiert, Werte 0 werden schwarz, maximale Werte – beim Sheppschen Phantom sind das Werte der Größe 2 (maximale Dichtewerte für die „Schädelwand“) – werden weiß dargestellt. Die Werte im Inneren des Phantoms haben Werte der Größenordnung 1 und unterscheiden sich nur um wenige Prozent voneinander – eine grundsätzliche Schwierigkeit der CT. Um die interessanten Werte im Inneren des Phantoms sichtbar zu machen, muss der Graustufenbereich durch einen Aufruf der Form `imshow(P, [low high])` sorgfältig gewählt werden (`low` muss etwas kleiner, `high` etwas größer als 1 gewählt werden). Eine ähnliche Anpassung muss auch bei der Darstellung von R vorgenommen werden. Die tatsächlichen Unterschiede der Werte von P können Sie sich besonders deutlich machen, wenn Sie einen geraden Schnitt durch P , etwa in Form der Werte $P(z, :)$ für eine feste Zeile z (zum Beispiel $z=235$) von P mithilfe der Funktion `plot` zeichnen.

2. Aufgabe. Betrachten Sie die charakteristische Funktion $f = \mathbf{1}_Q : \mathbb{R}^2 \rightarrow \mathbb{R}$ eines Quadrats Q . Diese nimmt „auf Q “ den Wert 1 an und den Wert 0 sonst, ist also definiert durch:

$$\mathbf{1}_Q(x_1, x_2) = \begin{cases} 1, & (x_1, x_2) \in Q \\ 0, & (x_1, x_2) \notin Q \end{cases} \quad (8)$$

Für feste Werte $j \in \{0, \dots, p-1\}$ und $\ell \in \{-q, \dots, q-1\}$ sei

$$L_{j,\ell} := \left\{ s_\ell \cdot \begin{pmatrix} \cos \varphi_j \\ \sin \varphi_j \end{pmatrix} + t \cdot \begin{pmatrix} -\sin \varphi_j \\ \cos \varphi_j \end{pmatrix}; \quad t \in \mathbb{R} \right\}$$

ein Strahl der Form (2). Setzen Sie

$$Q = [a_1, a_1 + \Delta) \times [a_2, a_2 + \Delta) \quad (9)$$

und berechnen Sie für die charakteristische Funktion $f = \mathbf{1}_Q$ den Wert der Radontransformierten $Rf(\varphi_j, s_\ell) = R\mathbf{1}_Q(\varphi_j, s_\ell)$ gemäß (4), wobei für die Parameter j und ℓ die Werte φ_j und s_ℓ gemäß (7) festgelegt sind.

Anleitung. Für die spezielle Funktion (8) ist der Wert (4) der Radontransformierten gerade die Länge des Schnittsegments von Q mit der Geraden $L_{j,\ell}$ (Wert 0, falls $L_{j,\ell}$ das Quadrat Q nicht trifft):

$$R\mathbf{1}_Q(\varphi_j, s_\ell) = |L_{j,\ell} \cap Q|. \quad (10)$$

Dieser Wert kann elementargeometrisch bestimmt werden, wenn man die Schnittpunkte der Geraden $x_1 = a$, $x_1 = a_1 + \Delta$, $x_2 = a_2$ und $x_2 = a_2 + \Delta$ mit der Geraden $L_{j,\ell}$ berechnet. Behandeln Sie dabei die horizontalen und die vertikalen Geraden $L_{j,\ell}$ (also die Fälle $j = 0$ und $j = p/2$) gesondert. Schreiben Sie eine Matlab-Funktion

```
function len = sekante(a1,a2,Delta,j,l,p,q)
```

zur Berechnung von (10). Hierbei ist $[a1; a2] = a$ der Koordinatenvektor der linken unteren Ecke von Q . **Bemerkung.** Die Berechnung erfordert keinerlei „Höhere Mathematik“, ist aber trotzdem fehleranfällig. Besprechen Sie deswegen die Lösung dieser Aufgabe mit dem Betreuer, der Ihnen auch ein Testprogramm zum Testen Ihrer Funktion `sekante` zur Verfügung stellen kann.

3. Aufgabe. Ein (zu rekonstruierendes) Bild f soll mit einer Auflösung von

$$N \cdot N = N^2 \quad \text{Pixeln}$$

dargestellt werden, wobei angenommen wird, dass N geradzahlig ist. Da das Bild im Quadrat $[-1, 1) \times [-1, 1)$ enthalten ist, hat ein Pixel die Seitenlänge

$$H = 2/N.$$

Die Pixel eines Bildes werden in Matlab mit einem Doppelindex durchnummeriert:

$$\alpha = (\alpha_1, \alpha_2) \in I := \{1, \dots, N\} \times \{1, \dots, N\},$$

hierbei erhält das Pixel in der linken oberen Bildecke die Nummer $(1, 1)$, dasjenige in der rechten oberen Ecke die Nummer $(1, N)$ und dasjenige in der rechten unteren Ecke die Nummer (N, N) – Sie können dies anhand des Ergebnisses der Funktion `sheppLoganP` der 1. Aufgabe verifizieren. Ein Pixel mit Seitenlänge H entspricht im x - y -Koordinatensystem einem Quadrat

$$Q_\alpha = [a_1, a_1 + H) \times [a_2, a_2 + H) \subset [-1, 1) \times [-1, 1) \quad (11)$$

mit

$$a_1 = (\alpha_2 - 1) \cdot H - 1 \quad \text{und} \quad a_2 = 1 - \alpha_1 \cdot H. \quad (12)$$

Eine Funktion $f : [-1, 1) \times [-1, 1) \rightarrow \mathbb{R}$ (und ebenso ein $f : D \rightarrow \mathbb{R}$) werde nun approximiert durch

$$f(x_1, x_2) \approx \tilde{f}(x_1, x_2) := \sum_{\alpha_1=1}^N \sum_{\alpha_2=1}^N c_\alpha \mathbf{1}_{Q_\alpha}(x_1, x_2), \quad (13)$$

das heißt durch eine stückweise konstante Funktion, welche auf dem Quadrat Q_α (dem Pixel α) den noch zu bestimmenden Funktionswert c_α annimmt. Ziel ist es, die Werte c_α so zu bestimmen, dass

$$R\tilde{f}(\varphi_j, s_\ell) = g(\varphi_j, s_\ell), \quad j = 0, \dots, p-1, \quad \ell = -q, \dots, q-1, \quad (14)$$

wobei g die Radontransformierte von Rf ist, siehe (6). Geht man mit dem Ansatz (13) direkt in (14), so ergibt sich das lineare Gleichungssystem

$$\sum_{\alpha_1=1}^N \sum_{\alpha_2=1}^N c_\alpha \cdot R\mathbf{1}_{Q_\alpha}(\varphi_j, s_\ell) = g(\varphi_j, s_\ell), \quad \forall j, \quad \forall \ell. \quad (15)$$

Dieses lineare Gleichungssystem aus $2pq$ Gleichungen für N^2 Koeffizienten c_α lässt sich kurz in der Form

$$A\mathbf{c} = \mathbf{g}, \quad A \in \mathbb{R}^{2pq, N^2}, \quad \mathbf{c} \in \mathbb{R}^{N^2}, \quad \mathbf{g} \in \mathbb{R}^{2pq} \quad (16)$$

schreiben. Dabei werden die Pixelindizes $\alpha = (\alpha_1, \alpha_2) \in \{1, \dots, N\}^2$ durch zeilenweise Abzählung zu eindimensionalen Indizes

$$s = (\alpha_1 - 1) \cdot N + \alpha_2 \in \{1, \dots, N^2\} \quad (17)$$

und die zweidimensionalen Indizes $(j, \ell) \in \{0, \dots, p-1\} \times \{-q, \dots, q-1\}$ zu eindimensionalen Indizes

$$z = (j-1) \cdot (2q) + q + 1 + \ell \in \{1, \dots, 2pq\} \quad (18)$$

geordnet und folgende Festlegungen für die komponentenweise Definition von A , \mathbf{c} und \mathbf{g} getroffen

$$A_{z,s} = R\mathbf{1}_{Q_\alpha}(\varphi_j, s_\ell), \quad \mathbf{c}_s = c_\alpha \quad \text{und} \quad \mathbf{g}_z = g(\varphi_j, s_\ell). \quad (19)$$

Schreiben Sie eine Matlab-Funktion

```
function g = getMesswerte(p,q),
```

welche die rechte Seite \mathbf{g} des Gleichungssystems (16) berechnet. Diese Funktion ruft im Wesentlichen die Funktion sheppLoganR aus der 1. Aufgabe auf und ordnet deren Rückgabewerte entsprechend der Vorgabe (18) in einem Vektor an. Benutzen

Sie zur Umordnung die Matlab-Funktion `reshape`. Schreiben Sie außerdem eine Matlab-Funktion

```
function A = getRadonMatrix(p,q,N),
```

welche die Matrix A des Gleichungssystems (16) berechnet. Diese Funktion sollte sich auf die Funktion `sekante` der 2. Aufgabe stützen. **Hinweise.** Mit anwachsenden Parametern p , q und N wird die Dimension von A sehr groß, es kommt sowohl zu Speicherplatzproblemen als auch zu erheblichen Laufzeitproblemen. Um letztere in den Griff zu bekommen, kann die Funktion `sekante` zu einer Funktion `sekante2` modifiziert werden, welche die Berechnungen (10) für alle Pixel $Q = Q_\alpha$ eines Bilds parallel ausführt, wodurch sich die Berechnungsgeschwindigkeit um ein Vielfaches steigern lässt. Sie erhalten vom Betreuer ein vorgefertigtes Gerüst der Funktion `sekante2`, in welchem Sie nach dem dort programmierten (und kommentierten) Muster die fehlenden Bausteine einer parallelen Berechnung noch selbst ausführen. Zur Vermeidung der Speicherplatzprobleme dürfen nur jene (relativ wenigen) Elemente von A abgespeichert werden, welche ungleich null sind. Informieren Sie sich hierzu über den Typ `sparse` in Matlab. Auch zur Funktion `getRadonMatrix` erhalten Sie vom Betreuer ein vorgefertigtes Gerüst, in dem die noch fehlenden Programmteile selbst zu ergänzen sind. Schließlich sollte die einmal berechnete Matrix A mittels des Matlab-Befehls `save` in einer Datei gespeichert werden. Bei späteren Simulationsläufen (4. Aufgabe) kann dann A mittels eines `load`-Befehls geladen und muss nicht jedesmal neu berechnet werden.

4. Aufgabe. Geben Sie Gründe an, warum das lineare Gleichungssystem (16) im Allgemeinen keine Lösung hat. Ersatzweise kann man versuchen, den Vektor \mathbf{c} wenigstens so zu wählen, dass $A\mathbf{c}$ „möglichst gut“ zu \mathbf{g} passt. Eine Möglichkeit ist der Übergang zum **linearen Ausgleichsproblem**

$$\|A\mathbf{c} - \mathbf{g}\|_2 \stackrel{!}{=} \text{Min} \quad (20)$$

Dies bedeutet: \mathbf{c} soll so gewählt werden, dass der Abstand $A\mathbf{c} - \mathbf{g}$, gemessen in der Euklidischen Norm, minimal wird. Verifizieren Sie, dass in dem Fall, dass (16) eine Lösung besitzt, diese auch eine Lösung von (20) ist. Aus diesem Grund kann (20) als Verallgemeinerung von (16) angesehen werden. Wegen der Größe von A kommen nur iterative Lösungsverfahren in Betracht. In Matlab ist das sogenannte „LSQR-Verfahren von Paige und Saunders“ als Funktion `lsqr` implementiert, mit dem lineare Ausgleichsprobleme der Form (20) gelöst werden¹. Informieren Sie sich in der Matlab-Dokumentation über die Funktion `lsqr` und benutzen Sie diese zur Lösung von (20). Schreiben Sie ein Matlab-Skript `RekonstruktionART.m`, in welchem Sie

- Die Parameter $p = 400$, $q = 128$ und $N = 256$ wählen, die zu rekonstruierende Dichtefunktion f des Bilds von Shepp generieren und zeichnen (siehe 1. Aufgabe)

¹Das LSQR-Verfahren ist mathematisch äquivalent zum CG-Verfahren, angewandt auf die Normalgleichungen des linearen Ausgleichsproblems.

- Mit der Funktion `getMesswerte` die Radontransformierte $g = Rf$ von f erzeugen.
- Die mit der Funktion `getRadonMatrix` in Aufgabe 3 erzeugte Matrix A einlesen (Befehl `load`).
- Das Ausgleichsproblem (20) mittels der Funktion `lsqr` lösen.
- Den errechneten Lösungsvektor \mathbf{c} in ein Bild (eine Matrix) der Dimension $N \times N$ umordnen (Funktion `reshape`) und dieses graphisch darstellen neben dem durch f repräsentierten Originalbild des ersten Schritts.

Experimentieren Sie dabei mit dem Parameter `maxit` von `lsqr`: Dieser begrenzt die maximale Anzahl von Iterationsschritten, welche `lsqr` ausführt. Für welche Werte von `maxit` erhalten Sie die (optisch) besten Rekonstruktionsergebnisse? Der optimale Wert von `maxit` sollte in der Nähe von 20 liegen, bei einer Steigerung der Iterationsschritte wird das Ergebnis (optisch) wieder schlechter. Dieser Effekt ist als **Semikonvergenz** des LSQR-Verfahrens bekannt, er tritt auch bei anderen iterativen Verfahren auf und lässt sich durch eine detaillierte Konvergenzanalyse erklären. Der Grund ist letztlich in der schlechten Kondition der Matrix A zu suchen.

In der Praxis wird zur Rekonstruktion von f aus Rf nicht die algebraische Rekonstruktionstechnik, sondern die sogenannten **gefilterte Rückprojektion** benutzt. In der folgenden 5. Aufgabe soll zunächst die Rückprojektion programmiert werden, in der 6. Aufgabe die gefilterte Rückprojektion. Einige benötigte technische Festlegungen werden vorab getroffen.

Erweiterte Radontransformation. Das Integral (4) ist auch für $\varphi \in [\pi, 2\pi)$ definiert und ebenso für alle $s \in \mathbb{R}$ – die so definierte Radontransformierte genügt dann offenbar der Identität

$$Rf(\varphi, s) = Rf(\varphi + \pi, -s). \quad (21)$$

Mit den Festlegungen (3) können

$$\varphi \in [0, 2\pi) \quad \text{und} \quad \theta = \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix} \in S^1 := \{x \in \mathbb{R}^2; \|x\|_2 = 1\}$$

gleichwertig zueinander benutzt werden. Analog zur Identifikation von φ mit θ schreiben wir auch

$$g(\varphi, s) = g(\theta, s), \quad \varphi \in [0, 2\pi), \quad \theta \in S^1,$$

und die Identität (21) lässt sich damit auch in der Form

$$Rf(\theta, s) = Rf(-\theta, -s) \quad (22)$$

schreiben, das heißt Rf ist eine gerade Funktion auf dem Zylindermantel

$$Z := S^1 \times \mathbb{R} = \{(\theta, s); \theta \in S^1, s \in \mathbb{R}\} \subset \mathbb{R}^3.$$

Ebenso schreiben wir gleichwertig zueinander

$$\int_{S^1} g(\theta, s) d\theta = \int_0^{2\pi} g(\varphi, s) d\varphi,$$

was sich auch durch Parametrisierung des links stehenden Kurvenintegrals 1. Art formal nachrechnen lässt.

Adjungierte Radontransformation. Für eine auf dem Zylindermantel Z definierte Funktion $g = g(\theta, s) = g(\varphi, s)$ wird durch

$$R^*g(x) = \int_{S^1} g(\theta, \theta \cdot x) d\theta = \int_0^{2\pi} \underbrace{g(\varphi, x_1 \cos \varphi + x_2 \sin \varphi)}_{=: v(\varphi)} d\varphi \quad (23)$$

eine Funktion $R^*g : \mathbb{R}^2 \rightarrow \mathbb{R}$ definiert². Hier ist $x = (x_1, x_2)$ und $\theta \cdot x$ ist das Skalarprodukt der Vektoren θ und x . Die Abbildung R^* heißt **adjungierte Radontransformation**.

Faltung. Die **Faltung** zweier Funktionen $u, v : \mathbb{R} \rightarrow \mathbb{R}$ ist eine Funktion $u * v : \mathbb{R} \rightarrow \mathbb{R}$, die durch

$$u * v(x) := \int_{-\infty}^{\infty} u(x-t) \cdot v(t) dt$$

definiert ist. Die **Faltung** zweier Funktionen $u, v : \mathbb{R}^2 \rightarrow \mathbb{R}$ ist eine Funktion $u * v : \mathbb{R}^2 \rightarrow \mathbb{R}$, die durch

$$u * v(x_1, x_2) := \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(x_1 - t_1, x_2 - t_2) \cdot v(t_1, t_2) dt_2 dt_1$$

definiert ist.

Faltungsformel für Radontransformation Es sei $v = v(\theta, s)$ eine auf dem Zylinder Z definierte und f eine auf D definierte Funktion. Weiterhin sei $V := R^*v$ gemäß (25) definiert. Für die Funktionen v und V und f gilt dann die Beziehung

$$V * f = R^*(v * Rf) \quad (24)$$

(was sich mittels der Transformationsformel für zweidimensionale Integration nachweisen lässt). Die Faltung auf der rechten Seite von (24) *bezieht sich ausschließlich auf die Variable s* , ist also eindimensional, so dass $v * Rf$ erneut eine auf Z erklärte Funktion ist. Auf der linken Seite ist ein zweidimensionale Faltung gemeint.

²Falls $g = Rf$, dann ist $g(\theta, s) = 0$ für $|s| \geq 1$ und somit ist dann $R^*g(x) = 0$ für $\|x\|_2 \geq 1$.

5. Aufgabe. Verifizieren Sie, dass eine Gerade

$$L = \{s\theta + t\theta^\perp; t \in \mathbb{R}\}, \quad \theta \in S^1, \quad s \in \mathbb{R},$$

wie sie bei der Integration (4) auftritt, genau dann durch einen Punkt $x = (x_1, x_2) \in \mathbb{R}^2$ geht, wenn sie die Eigenschaft $s = x_1 \cos \varphi + x_2 \sin \varphi = \theta \cdot x$ hat. Warum lässt sich deswegen erwarten, dass für $g(\theta, s) = Rf(\theta, s)$ die Funktion $R^*g(x)$ etwas mit $f = f(x)$ zu tun hat? Zur Überprüfung der Vermutung sollen für das Phantombild f von Shepp-Logan die Werte

$$(R^*(Rf))(a) \text{ für } a = (a_1, a_2) \text{ aus (12)} \quad (25)$$

berechnet werden, das heißt zu gegebenem $g = Rf$ soll das Integral (23) an den Stellen $x = a$ berechnet werden. Schreiben Sie ein Matlab-Skript `backProjection`, welches die Berechnung (25) numerisch durchführt. Wenn Sie die Werte $R^*(Rf)(a)$ zeichnen, sollten Sie eine „Ähnlichkeit“ zu den Werten $f(a)$ feststellen können.

Anleitung. Wegen der Symmetriebeziehung (22) ist (23) gleichwertig zu

$$R^*g(x) = 2 \int_0^\pi v(\varphi) d\varphi \quad (26)$$

mit dem π -periodischen Integranden $v(\varphi) = g(\varphi, x_1 \cos \varphi + x_2 \sin \varphi)$. Wenn der Integrand v von (23) an äquidistanten Stellen ausgewertet als Vektor vorliegt, zum Beispiel

$$\mathbf{v} = (v_0, \dots, v_{p-1}) \quad \text{mit} \quad v_j = v\left(\frac{j \cdot \pi}{p}\right), \quad j = 0, \dots, p-1,$$

für $v(\varphi) = g(\varphi, x_1 \cos \varphi + x_2 \sin \varphi)$, dann kann das Integral in (23) numerisch durch die Trapezregel ausgewertet werden als

$$R^*g(x) \approx \frac{2\pi}{p} \sum_{j=0}^{p-1} v_j. \quad (27)$$

Diese Berechnung muss für alle Punkte $x = a = (a_1, a_2)$ der Form (12), also für alle $\alpha_1 = 1, \dots, N$ und $\alpha_2 = 1, \dots, N$ vorgenommen werden, somit sind N^2 Integralnäherungen der Form (27) zu berechnen. Für gegebenes (x_1, x_2) und gegebenes $\varphi_j = j \cdot \pi/p$, $j = 0, \dots, p-1$, sei

$$s_{j,x} := x_1 \cos \varphi_j + x_2 \sin \varphi_j.$$

Nun wären zur Berechnung von (27) die Werte $v_j = g(\varphi_j, s_{j,x})$ von g nötig, jedoch liegen nur die Werte $g(\varphi_j, s_\ell)$ (als Ergebnis des Aufrufs der Funktion `sheppLoganR` aus Aufgabe 1) vor. Aus letzteren gewinnt man $g(\varphi_j, s_{j,x})$ näherungsweise durch lineare Interpolation, das heißt man bestimmt k und γ so, dass

$$k \in \{-q, \dots, q-1\}, \quad kh \leq s_{j,x} < (k+1)h, \quad \gamma = (s_{j,x} - kh)/h \in [0, 1], \quad (28)$$

setzt dann

$$\tilde{v}_j := (1 - \gamma) \cdot g(\varphi_j, s_k) + \gamma \cdot g(\varphi_j, s_{k+1}) \quad (29)$$

und berechnet

$$R^*g(x) \approx \frac{2\pi}{p} \sum_{j=0}^{p-1} \tilde{v}_j. \quad (30)$$

anstelle von (27). Damit $g(\varphi_j, s_{k+1})$ in jedem Fall definiert ist, sollte $g(\varphi, 1) := 0$ gesetzt werden (man fügt dazu der von sheppLoganR berechneten Matrix R eine Nullzeile hinzu).

Der **Algorithmus der gefilterten Rückprojektion** (*algorithm of filtered backprojection*) beruht auf einer Implementierung von (24), wobei V so gewählt wird, dass in guter Näherung

$$f \approx V * f \quad (2D\text{-Faltung})$$

gilt. Dies bedeutet, dass Berechnung der rechten Seite von (24) eine Näherung von f berechnet. Von Shepp und Logan stammt folgender Vorschlag³. Man setze für $b := \pi/h$ mit $h := 1/q$

$$v(\theta, s) := \frac{b^2}{2\pi^3} u(bs) \quad (31)$$

– diese Funktion hängt nicht von θ ab, das heißt sie ist radialsymmetrisch – mit

$$u(s) := \begin{cases} \frac{\pi/2 - s \sin(s)}{(\pi/2)^2 - s^2}, & s \neq \pm\pi/2, \\ \frac{1}{\pi}, & s = \pm\pi/2 \end{cases}.$$

Daraus ergeben sich speziell für $s = s_\ell$, $\ell = -q, \dots, q-1$ die Werte

$$v(s_\ell) = \frac{b^2}{\pi^4} \cdot \frac{1}{1 - 4\ell^2}. \quad (32)$$

6. Aufgabe. Programmieren Sie ein Matlab-Skript `filteredBackProjection`, welches die rechte Seite von (24) für die Funktion v aus (31) numerisch berechnet. Gehen Sie dabei wie folgt vor.

1. Berechnen Sie die Werte $R(\varphi_j, s_\ell)$ mit der Funktion `sheppLoganR` aus der 1. Aufgabe.
2. Approximieren Sie die Berechnung der Faltungsprodukte $v * Rf$ für die Funktion v aus (31) durch numerische Integration mittels der Trapezsumme:

$$(v * Rf)(\varphi_j, s_k) \approx v_{j,k} := h \sum_{\ell=-q}^{q-1} v(s_k - s_\ell) g(\varphi_j, s_\ell) \quad (33)$$

für $j = 0, \dots, p-1$ und $k = -q, \dots, q-1$.

³Dieser Vorschlag fällt nicht vom Himmel. Die Bedingung $f \approx f * V$ erfüllt man, indem man einen geeigneten Ansatz für die Fouriertransformierte \hat{V} von V macht. Weiterhin lässt sich für $V = R^*v$ eine direkte Beziehung der Fouriertransformierten \hat{V} und \hat{v} herleiten, so dass man aus \hat{V} zunächst \hat{v} und daraus die Funktion v durch inverse Fouriertransformation finden kann.

3. Berechnen Sie $R^*g(a)$ für die Stellen $a = (a_1, a_2)$ aus (12) näherungsweise wie in der 5. Aufgabe, das heißt durch Auswertung der Formel (30), wobei die Werte \tilde{v}_j wie in (29), nun aber mit den Werten $v_{j,k}$ bzw. $v_{j,k+1}$ aus (33) anstelle der Werte $g(\varphi_j, s_k)$ bzw. $g(\varphi_j, s_{k+1})$ zu berechnen sind.

Testen Sie das Verfahren für $p = 800$, $q = 256$ und $N = 512$ und vergleichen Sie Ihr Ergebnis optisch mit dem Originalbild f .

7. Aufgabe. Was ist die arithmetische Komplexität des Algorithmus der gefilterten Rückprojektion? Implementiert man die Berechnung der Faltungsprodukte in (33) effizient mittels der sogenannten *Fast Fourier Transform*, dann lassen sich alle Werte $v_{j,k}$ zusammen in $\mathcal{O}(pq \log q)$ Operationen berechnen. Wie groß ist dann die Gesamtkomplexität der gefilterten Rückprojektion?