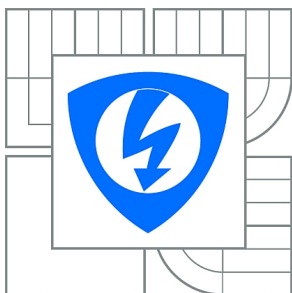


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

REKONSTRUKCE OBRAZŮ Z PROJEKCÍ

IMAGE RECONSTRUCTION FORM PROJECTIONS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

DUŠAN GARGULÁK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. RADOVAN JIŘÍK, Ph.D.

BRNO 2011

ABSTRAKT

Tato bakalářská práce pojednává o principu rekonstrukce obrazů z projekcí, na kterém jsou založeny moderní metody RTG výpočetní tomografie (CT RTG). Text se zaměřuje na rekonstrukci obrazu prostou a filtrovanou zpětnou projekcí.

V první části je popsán teoretický základ práce, v němž se vysvětluje princip Radonovy transformace a zpětné Radonovy transformace. Stejně tak se zde objasňuje postup prosté zpětné projekce a filtrované zpětné projekce.

Další část práce obsahuje praktické aplikace Radonovy transformace, pojednává o rekonstrukčních metodách a návrzích číslicových filtrů užívaných při rekonstrukci projekcí. Jako poslední je komentováno grafické rozhraní, které bylo vytvořeno k předchozím aplikacím a je také vytvořena laboratorní úloha pro počítačová cvičení.

KLÍČOVÁ SLOVA

Radonova transformace, inverzní Radonova transformace, projekční teorém, rekonstrukce obrazů z projekcí, prostá zpětná projekce, filtrovaná zpětná projekce, filtrovaná zpětná projekce, MATLAB, číslicový filtr

ABSTRACT

This bachelor work deals with the principle of the image reconstruction from projections which is the basis for the modern method of the RTG computed tomography (CT RTG). The work is above all focused on the image reconstruction by the Simple BackProjection and Filtered BackProjection.

In the first part there are outlined theoretical principles, which explain Radon Transform and inverse Radon Transform. There is also cleared up the process of the Simple BackProjection.

The next part of the work contains the practical applications of Radon Transform and deals with the reconstruction methods and design of the digital filters which are used for the reconstruction of the projections. As the last item there is annotated the graphic interface for the mentioned applications and also the laboratory task is added there.

THE KEY WORDS

Radon Transform, inverse Radon Transform, projection Theorem, image reconstruction from projections, Simple BackProjection, Filtered BackProjection, MATLAB, digital filter

GARGULÁK, D. *Rekonstrukce obrazů z projekcí*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2011. 26 s. Bakalářská práce. Vedoucí práce: Ing. Radovan Jiřík, Ph. D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma Rekonstrukce obrazů z projekcí jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

Dušan Gargulák

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce Ing. Radovanu Jiříkovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne

.....

Dušan Gargulák

Obsah

Seznam obrázků	2
1 Úvod	3
2 Teoretický základ	4
2.1 Radonova transformace	4
2.1.1 Projekční teorém	6
2.2 Rekonstrukce obrazů z projekcí	7
2.2.1 Prostá zpětná projekce	7
2.2.2 Filtrovaná zpětná projekce	8
2.2.3 Nejčastěji užívané filtry pro filtrovanou zpětnou projekci	10
3 Realizace zadání	11
3.1 Popis vytvořeného uživatelského rozhraní	11
3.1.1 Zdrojový kód pro tvorbu číslicových filtrů	14
3.1.2 Zdrojový kód pro generování souboru projekcí	15
3.1.3 Popis algoritmu rekonstrukce projekcí prostou i filtrovanou zpětnou projekcí	17
3.2 Srovnání zkonstruovaných filtrů	20
3.2.1 Kvalitativní porovnání	20
3.2.2 Kvantitativní porovnání	21
3.3 Zadání laboratorní úlohy	24
4 Závěr	25
Literatura	26

Seznam obrázků

2.1	Popis vzniku projekcí (převzato z [1], str. 389).	4
2.2	Vliv vzdálenosti bodu od středu obrazu na velikost amplitudy sinusoidy. Levý obrázek znázorňuje dva body v obrazu, pravý je pak příslušný sinogram (osy: $x - \vartheta[\text{deg}]$, $y - \text{detektory}$).	5
2.3	Hvězdicový artefakt vzniklý při prosté zpětné projekci. Obraz je rekonstruován z 10-ti sejmutých projekcí.	8
3.4	Vzhled uživatelského rozhraní: Zleva <i>Program1</i> , <i>Program2</i>	12
3.5	Uživatelské rozhraní funkce <i>Program3</i>	13
3.6	zdrojový kód frekvenční (nahore) a impulsní (dole) charakteristiky filtrů. Vlevo: Ram-Lak filtr, vpravo: Hannův filtr.	15
3.7	Funkce aplikace MATLAB pro generování souboru projekcí.	16
3.8	Na levém obrázku je znázorněn vstupní obraz před transformací. Na pravém obrázku nahore je snímek uložený v matici I_{rot} , pootočený o 180° a na snímku pod ním jemu odpovídající projekce (osa $x - \text{detektor}$, osa $y - \text{hodnota paprskového integrálu}$).	17
3.9	Funkce aplikace MATLAB pro filtrovanou rekonstrukci obrazů z projekcí. .	18
3.10	Vliv filtru na podobu rekonstruovaného obrazu. Vlevo je původní snímek. Uprostřed je rekonstruovaný snímek pomocí prosté zp. projekce. Snímek vpravo je pak rekonstruovaný obraz po aplikaci Ram-Lak filtru. (Počet sejmutých projekcí byl 180)	19
3.11	Frekvenční a impulsní charakteristiky vytvořených filtrů. Shora: Ram-Lak, Shepp-Logan, Hammingův, Hannův, kosinový.	19
3.12	Frekvenční přenosové charakteristiky použitých filtrů	20
3.13	Porovnání výstupních obrazů za použití různých filtrů. Zleva: originální obraz, nefiltrovaný, filtrovaný Ram-Lakenovým, Shepp-Loganovým, Hannovým, Hammingovým a kosinovým filtrem	20
3.14	17. Frekvenční přenosové charakteristiky použitých filtrů	23

1 Úvod

V matematice je definována *Radonova transformace* funkce, definovaná je také *zpětná Radonova transformace*. Aplikací Radonovy transformace na funkci dvou proměnných získáme soubor funkcí jedné proměnné – tzv. *soubor projekcí*. Aplikací zpětné Radonovy transformace na soubor těchto projekcí získáme zpět původní funkci dvou proměnných.

Předchozí poznatky tvoří základní princip tomografického¹ zobrazování. Tímto způsobem lze pozorovat vnitřní strukturu látek, aniž by muselo dojít k jejich fyzickému narušení. Existuje více aplikací tomografického zobrazování. Své uplatnění našly nejvíce v oblasti průmyslu, především ve strojírenství a farmacii – zde se jedná konkrétně o rentgenovou výpočetní tomografii (CT RTG) a tomografii založenou na nukleární magnetické rezonanci (NMR). Tato práce se zabývá aplikací Radonovy transformace právě v RTG výpočetní tomografii.

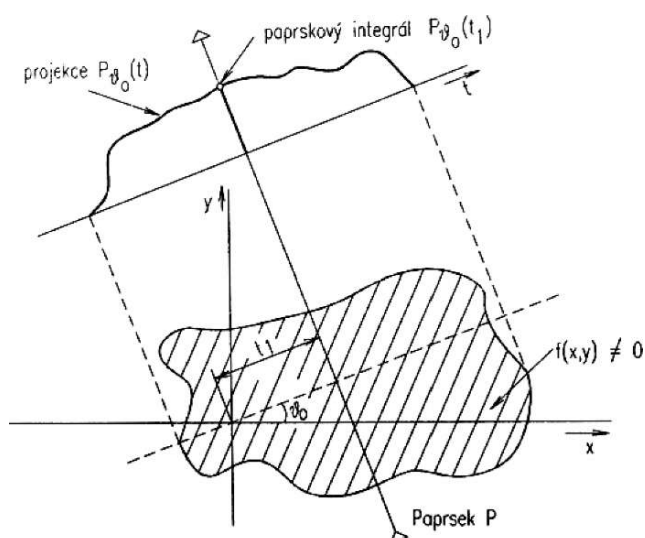
Cílem bakalářské práce je vytvoření výukového programu simulujícího procesy v rentgenové výpočetní tomografii. Práce je rozdělena do tří hlavních kapitol – v první jsou uvedeny teoretické základy, ve druhé je popsána realizace zadání, v poslední části jsou pak diskutovány výsledky.

¹Tomografické zobrazování znamená zobrazování v řezech.

2 Teoretický základ

2.1 Radonova transformace

Radonova transformace nese název po rakouském matematikovi Johannu Radonovi, jenž ji zformuloval. Jedná se o integrální transformaci, kterou lze provádět obecně pro funkci n proměnných, jejíž podstata spočívá v integraci dané funkce přes její nadroviny (tzn. $(n-1)$ -dimenzionální prostor). V našem případě budeme pracovat s funkcemi dvou proměnných, nadrovinou pro nás tedy bude přímka jakožto graf funkce jedné proměnné.



Obr. 2.1: Popis vzniku projekcí (převzato z [1], str. 389).

Na Obr. 2.1 je znázorněn tomografický řez tělesem, který leží v rovině xy . Charakteristický parametr tomografické vrstvy v každém bodě řezu je popsán funkcí $f(x,y)$, kterou budeme dále nazývat *předmětová funkce*. Zobrazíme-li na osu t integrál předmětové funkce po přímce (papřsku) p , získáme bod, který nazýváme papřskový integrál. Hodnota papřskového integrálu je ovlivňována parametry ϑ_0 a t_1 (které určují polohu přímky p) a je dána vztahem [1]

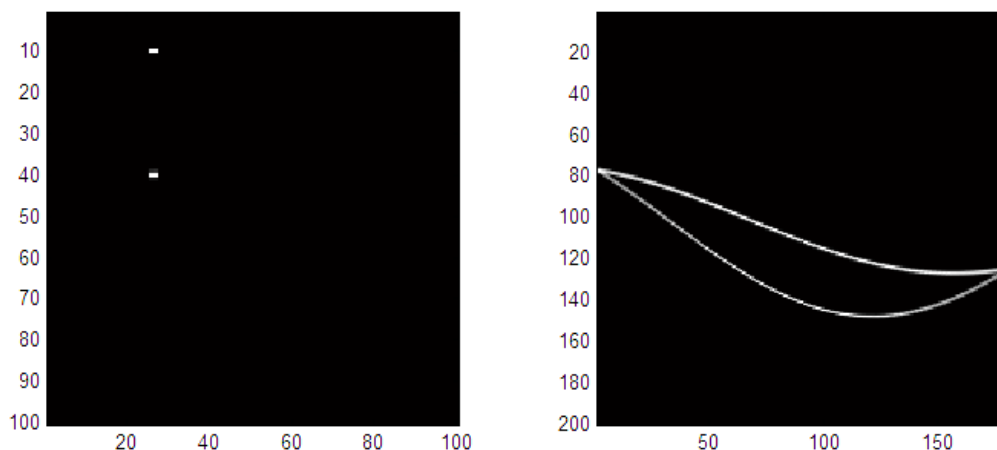
$$P_{\vartheta_0}(t_1) = \int_p f(x, y) ds, \quad (2.1)$$

kde ϑ_0 značí úhel, který svírá normála přímky p s osou x , a t_1 vzdálenost přímky p od počátku soustavy souřadnic.

Bude-li se při daném ϑ_0 spojitě měnit parametr t (tj. vzdálenost přímky p od počátku souřadného systému), dojde promítnutím jednotlivých paprskových integrálů na osu t k vykreslení tzv. *projekce* (viz Obr. 2.1) příslušející úhlu ϑ_0 . Bude-li se spojitě měnit také úhel ϑ , přejde integrál (2.1) do tvaru [1]

$$P_{\vartheta}(t) = P(t, \vartheta) = \int_{P_{t, \vartheta}} f(x, y) ds = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \vartheta + y \sin \vartheta - t) dx dy. \quad (2.2)$$

Tento integrál nazýváme Radonovou transformací. Podle vztahu (2.2) se jedná o dvou-rozměrnou funkci, kterou lze chápat jako spojitý *soubor projekcí*, z nichž každá přísluší spojitě se měnícímu úhlu ϑ . Prostorová reprezentace tohoto souboru projekcí je tzv. *sino-gram*. Tento název vychází ze skutečnosti, že zobrazením Radonovy transformace Diracovy delta funkce je sinusoida. Řečeno jinými slovy, každý bod (funkční hodnota) původní předmětové funkce se po transformaci zobrazí v sinogramu jako sinusoida. Amplituda této sinusoidy je dána polohou bodu v původní funkci – čím blíže ke středu se bod nachází, tím je amplituda příslušející sinusoidy menší (viz Obr. 2.2).



Obr. 2.2: Vliv vzdálenosti bodu od středu obrazu na velikost amplitudy sinusoidy. Levý obrázek znázorňuje dva body v obrazu, pravý je pak příslušný sinogram (osy: $x - \vartheta[\text{deg}]$, $y - \text{detektory}$).

Vyjádříme-li spojitou změnu úhlu ϑ spojitou změnou souřadnic souřadného systému v závislosti na ϑ následovně

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \vartheta & \sin \vartheta \\ -\sin \vartheta & \cos \vartheta \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad (2.3)$$

přičemž provedeme přeznačení $t = x'$, lze vyjádření Radonovy transformace psát ve tvaru

$$P(x', \vartheta) = \int_{-\infty}^{\infty} (x' \cos \vartheta - y' \sin \vartheta, x' \sin \vartheta + y' \cos \vartheta) dy'. \quad (2.4)$$

Na závěr uvedme dvě vlastnosti Radonovy transformace, které lze v praxi s výhodou využít. První je periodicitu funkce vzhledem k úhlu ϑ s periodou 2π druhou lze vyjádřit rovností [2]

$$P(\vartheta + \pi, x') = P(\vartheta, x'). \quad (2.5)$$

Celkem to tedy znamená, že projekce jsou symetrické a periodické s úhlem ϑ a periodou 2π , tj. $P_{\vartheta}(x') = P_{\vartheta \pm \pi}(x')$.

2.1.1 Projekční teorém

Výše popsaná Radonova transformace úzce souvisí s transformací Fourierovou a ve svém důsledku vede k formulaci tzv. *projekčního* či *centrálně řezového teorému*. Jedná se o jeden ze základních fenoménů rekonstrukčních metod obrazů zpětnou projekcí.

Projekční teorém říká, že jednorozměrné spektrum jedné projekce pořízené z určitého směru je totéž, jako centrální řez dvojrozměrným spektrem hledaného obrazu, který provedeme ve směru kolmém na směr projekce. Podrobné odvození výše zmíněného faktu lze najít např. v [3].

2.2 Rekonstrukce obrazů z projekcí

Cílem rekonstrukce obrazu z projekcí je nalezení původní předmětové funkce, na kterou byla aplikována Radonova transformace. Jde tedy o nalezení inverzní Radonovy transformace dané předmětové funkce.

Vzhledem k tomu, že v technické praxi není možné sejmout nekonečně mnoho projekcí, ale jen určitý omezený počet, spočívá formulace inverzní Radonovy transformace v nalezení vhodné aproximativní diskrétní inverzní transformace. Touto 2D rekonstrukcí získáme odhad původní předmětové funkce. Proces rekonstrukce původní funkce je tedy možno zařadit do skupiny restauračních postupů, neboť jde o získání původní funkce z 2D souboru 1D vektorů (projekcí), přičemž tento soubor lze chápat jako *zkreslený obraz*.

V současné době existují tři různé způsoby, jak numericky řešit rekonstrukci obrazů z projekcí. Jedná se o algebraickou rekonstrukci, Fourierovu rekonstrukci založenou na centrálním řezovém teorému (viz 2.1.1) a tzv. zpětnou projekci. Zpětnou projekci můžeme provádět buď jako *prostou zpětnou projekci* (nefiltrovanou zp. proj.) nebo jako *filtrovanou zpětnou projekci*. Tato práce se bude zabývat právě posledními dvěma zmíněnými metodami.

2.2.1 Prostá zpětná projekce

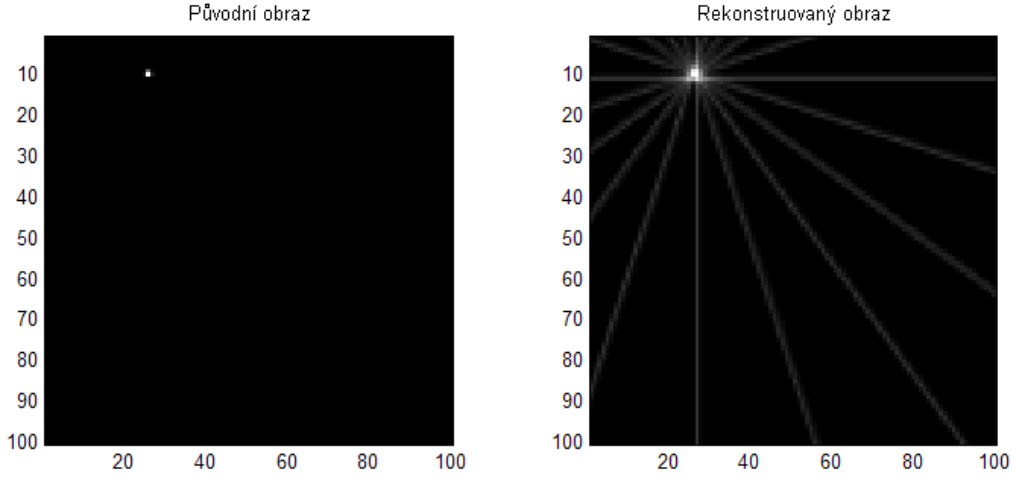
Prostá zpětná projekce (SBP - Simple BackProjection) je algoritmus, který původní předmětovou funkci hledá podle vztahu [3]

$$f'(x, y) = \frac{1}{\pi} p_{\vartheta}(x \cos \vartheta + z \sin \vartheta) d\vartheta. \quad (2.6)$$

Algoritmus vychází z myšlenky, že původní předmětovou funkci lze znovu získat přesně opačným způsobem, než jakým byl získán soubor všech projekcí. Úvaha je následující: prázdnou matici naplníme postupně všemi jednotlivými projekcemi P_{ϑ} a to ve směru původní akvizice dané projekce. Následným sečtením všech takto vytvořených obrazových matic dostáváme rekonstruovaný obraz.

Výše popsaným algoritmem prosté zpětné projekce ovšem nelze dospět přesně k původní předmětové funkci, rekonstruovaný obraz je totiž velmi zkreslený. Charakter této neostrosti lze dobře pozorovat na modelu Diracova impulsu, výsledný obraz se zobrazuje jako hvězdice. Tento jev bývá označován jako *hvězdicový artefakt* (viz Obr. 2.3).

Korigovat zkreslení vznikající při prosté zpětné projekci je možné *aplikací vhodného filtru* na jednotlivé projekce ještě před provedením rekonstrukce. Tomuto postupu říkáme filtrovaná zpětná projekce.



Obr. 2.3: Hvězdicový artefakt vzniklý při prosté zpětné projekci. Obraz je rekonstruován z 10-ti sejmutých projekcí.

2.2.2 Filtrovaná zpětná projekce

Jak již bylo zmíněno výše, nevýhody prosté zpětné projekce lze odstranit vhodným filtrováním jednotlivých projekcí. Filtrovaná zpětná projekce (FBP - Filtered BackProjection) hledá rekonstruovanou funkci $f'(x, y)$ podle vztahu [3]

$$f'(x, y) = \int_0^\pi \int_{-\infty}^{\infty} P_\vartheta(x') h(x \cos \vartheta + y \sin \vartheta - x') dx' d\vartheta, \quad (2.7)$$

ve kterém funkce $h(x, y)$ je funkce potlačující zkreslení při prosté zpětné projekci. Ze srovnání vztahu (2.6) pro prostou zpětnou projekci a výrazu (2.7) je patrné, že vnitřní integrál

$$\int_{-\infty}^{\infty} P_\vartheta(x') h(x \cos \vartheta + y \sin \vartheta - x') dx' d\vartheta, \quad (2.8)$$

je právě konvoluční integrál, který provádí danou filtraci. Pro návrh filtru je třeba určit jeho impulzní charakteristiku, to je funkci $h(x)$. K jejímu odvození vyjdeme z Fourierovy rekonstrukce, v níž rekonstruovanou funkci hledáme s využitím platnosti projekčního teoremu (viz 2.1.1) jako inverzní Fourierovu transformaci 2D spektra předmětové funkce, tj. $F_{2D}^{-1}[F(\omega_x, \omega_y)]$.

Pro určení impulzní charakteristiky $h(x)$ uijeme vztahu pro Fourierovu rekonstrukci předmětové funkce v polárních souřadnicích [3]

$$f'(r, \beta) = \int_0^\pi \int_{-\infty}^\infty F(\omega, \vartheta) e^{j\omega(x \cos \vartheta + z \sin \vartheta)} |J| d\omega d\vartheta, \quad (2.9)$$

kde pro Jakobián J platí

$$J = \begin{bmatrix} \frac{\partial \omega_x}{\partial \omega} & \frac{\partial \omega_y}{\partial \omega} \\ \frac{\partial \omega_x}{\partial \vartheta} & \frac{\partial \omega_y}{\partial \vartheta} \end{bmatrix} = \begin{bmatrix} \cos \vartheta & \sin \vartheta \\ -\omega \sin \vartheta & \omega \cos \vartheta \end{bmatrix} \quad (2.10)$$

a

$$\omega = \sqrt{\omega_x^2 + \omega_y^2}, \vartheta = \arctan \frac{\omega_x}{\omega_y}, r = \sqrt{x^2 + y^2}, \beta = \arctan \frac{y}{x}. \quad (2.11)$$

Abychom mohli následně porovnat vztahy (2.7) a (2.9), neboť oba popisují tutéž rekonstruovanou funkci, vyjádříme nejprve výraz (2.7) pomocí vztahů (2.11) v polárních souřadnicích

$$f'(r, \beta) = \int_0^\pi \int_{-\infty}^\infty P_\vartheta(x') h(r - x') dx' d\vartheta. \quad (2.12)$$

Nyní již lze psát

$$\int F(\omega, \vartheta) e^{j\omega r} |J| d\omega = \int_{-\infty}^\infty P_\vartheta(x') h(r - x') dx' \quad (2.13)$$

Jelikož výraz na levé straně rovnosti je zpětná Fourierova transformace a výraz na pravé straně je konvoluce dvou funkcí v prostoru, aplikací 2D Fourierovy transformace na tuto rovnost dostáváme

$$F(\omega, \vartheta) |J| = \mathbf{F} [P_\vartheta(x') * h(x')] = \mathbf{F} [P_\vartheta(x')] \cdot \mathbf{F} [h(x')] \quad (2.14)$$

Z projekčního teorému, tj. z platnosti rovnosti $F(\omega, \vartheta) = \mathbf{F} [P(x')]$ (viz. [3]) a ze vztahu $J = \omega$ musí platit

$$\mathbf{F} [h(x')] = [\omega]. \quad (2.15)$$

Tento vztah nám umožňuje vypočítat požadovanou impulzní charakteristiku filtru jako zpětnou transformaci známého Jakobiánu. Filtry s přenosovou charakteristikou tohoto typu se nazývají ramp-filtry.

Ramp-filtr však ve skutečnosti není realizovatelný, proto se v praxi musíme spokojit jen s jeho aproximací. Takový filtr má však velké zesílení na vysokých kmitočtech, čímž dochází k nárůstu šumu a ke snížení poměru signál-šum. Zavádí se proto filtry, které v sobě kombinují vlastnosti filtru ramp s filtry na potlačení šumu.

2.2.3 Nejčastěji užívané filtry pro filtrovanou zpětnou projekci

Aplikací ramp-filtru lze sice dosáhnout korekci zkreslení vznikající při prosté zpětné projekci vlivem hvězdicového artefaktu, na druhou stranu však dochází k velkému přenosu na vyšších kmitočtech, což s sebou v důsledku nese zvýraznění takových informací v projekcích, které jsou pro uživatele často nepotřebné. Tento problém se řeší tak, že se původní přenosová funkce ramp-filtru násobí funkcí, která vhodně ovlivní přenos ramp-filtru na vyšších kmitočtech – násobí se vhodným dolnoproputstním filtrem. Takto vzniklé filtry pak většinou nazýváme podle funkce, kterou jsme násobili původní ramp-filtr. Nejčastěji používané filtry jsou tyto [7]:

Ram-Lakenův filtr:

$$\mathbf{F} \left[h(x') \right] = |w|, \quad |\omega| \leq 2B, 0 \dots jinde; \quad (2.16)$$

Shepp-Loganův filtr:

$$\mathbf{F} \left[h(x') \right] = |w| \cdot \sin \left| \frac{\omega}{4B} \right|, \quad |\omega| \leq 2\pi B, 0 \dots jinde; \quad (2.17)$$

Hammingův filtr:

$$\mathbf{F} \left[h(x') \right] = |w| \cdot \left[0,54 + 0,46 \cos \left(\frac{\omega}{2B} \right) \right], \quad |\omega| \leq 2\pi B, 0 \dots jinde; \quad (2.18)$$

Hannův filtr:

$$\mathbf{F} \left[h(x') \right] = |w| \cdot 0,5 \left(1 + \cos \frac{\omega}{2B} \right), \quad |\omega| \leq 2\pi B, 0 \dots jinde; \quad (2.19)$$

Kosinový filtr:

$$\mathbf{F} \left[h(x') \right] = |w| \cdot \cos \left(\frac{\omega}{4B} - \frac{\pi}{2} \right), \quad |\omega| \leq 2\pi B, 0 \dots jinde; \quad (2.20)$$

Tvary a vlastnosti výsledných frekvenčních charakteristik popsaných filtrů prozkoumáme blíže v části. 3.1.1, kde se budeme zabývat tvorbou číslicových filtrů.

3 Realizace zadání

Cílem této práce bylo vytvořit v grafickém prostředí MATLAB uživatelské rozhraní simulačního programu, které by mohlo posloužit k snadnějšímu pochopení problematiky při výuce. Konkrétně mělo být dosaženo těchto pěti cílů: 1) měl být vytvořen demonstrační program pro generování projekcí ze zadaného obrazu, 2) program pro rekonstrukci obrazů z projekcí, a to pomocí 3) nefiltrované a 4) filtrované zpětné projekce s jednoduchou volbou metod a parametrů zpracování. Součástí práce bylo také 5) vytvořit zadání laboratorní úlohy.

Zadání práce bylo realizováno aplikací vytvořenou v prostředí programu MATLAB, ve kterém bylo využito systému Handle Graphics pro tvorbu grafického prostředí. Aplikace je rozložena do tří hlavních programů nazvaných postupně *Program1*, *Program2* a *Program3*, z nichž každý realizuje jednotlivé kroky zadání, tedy: **1.** tvorbu souboru projekcí ze zadaného obrazu, **2.** rekonstrukci původního obrazu zpětnou projekcí s výběrem nejčastěji používaných filtrů a **3.** porovnání jednotlivých filtrů výpočtem kvadratické odchylky rekonstruovaných obrazů.

V následujících podkapitolách bude nejdříve stručně popsán vzhled a celková funkčnost jednotlivých programů. Dále pak, vzhledem k tomu, že jádro vytvořené aplikace spočívá především v algoritmech, kterými vytváříme požadované číslíkové filtry, získáváme soubor projekcí a rekonstruujeme původní obraz ze souboru projekcí, budou tyto algoritmy podrobně vysvětleny. Naposledy bude zadána úloha laboratorního cvičení, která je koncipována tak, aby student mohl na jednotlivých cvičeních látku co nejlépe pochopit.

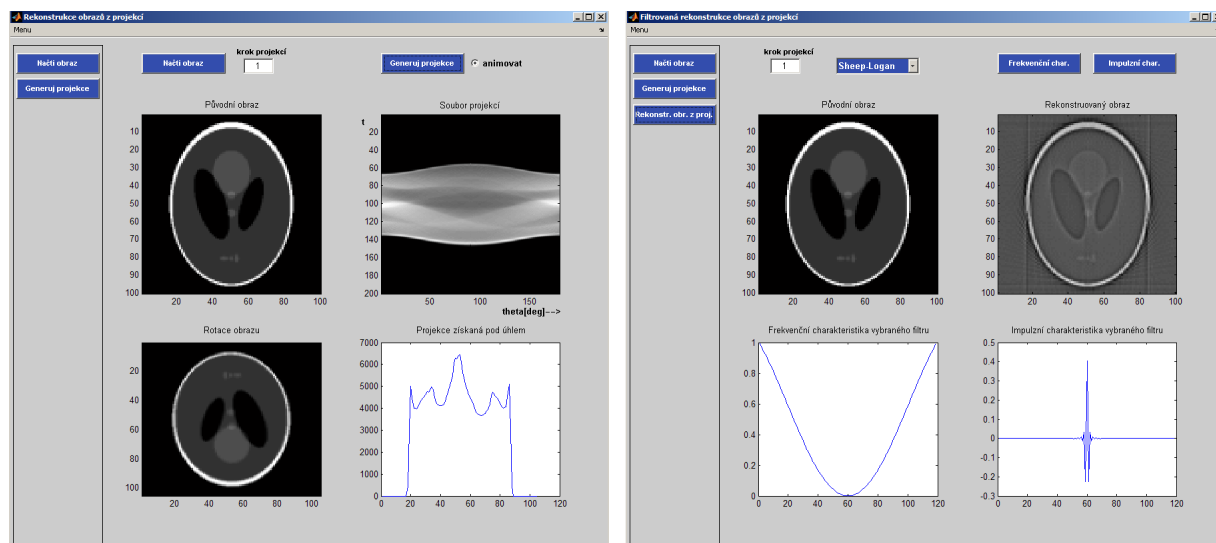
3.1 Popis vytvořeného uživatelského rozhraní

Zde bude ve stručnosti popsáno prostředí a funkčnost jednotlivých programů *Program1*, *Program2* a *Program3*.

Program1

Tento program (viz Obr. 3.4 vlevo) má za úkol vytvořit z předmětové funkce (vstupního obrazu) soubor projekcí. Je zde možno nastavit krok snímání jednotlivých projekcí, tj. z kolika projekcí se celkový soubor všech projekcí vytvoří. V programu je také mož-

nost zatržením přepínače „animovat“ spustit animaci generování projekcí, díky které lze mnohem lépe porozumět principu, jakým se soubor projekcí získává. V adresáři je spolu s programem uloženo i několik zkušebních obrázků, na kterých si uživatel může ověřit teoretické poznatky. Je mezi nimi i model Diracova impulsu v různých polohách prostoru obrazu, pomocí něhož lze dobře pozorovat vztah polohy bodu předmětové funkce a obsahu výsledného sinogramu.



Obr. 3.4: Vzhled uživatelského rozhraní: Zleva *Program1*, *Program2*.

Funkčnost je následující: Uživatel stiskem tlačítka „Načti obraz“ vybere přes dialogové okno umístění souboru s obrazovými daty. Načtený obraz se zobrazí v levém horním okně. Nyní může uživatel nastavit počet sejmutých projekcí vložení číselné hodnoty do pole „krok projekcí“. Chce-li, může také zapnout přepínačem „animovat“ možnost animace generování projekcí. Poté stisknutím tlačítka „Generuj projekce“ spustí generování souborů projekcí. Pokud byla zvolena možnost „animovat“, bude se průběh generování zobrazovat na ploše (v levém spodním rohu bude viditelná aktuální poloha natočení obrazu, v pravém spodním rohu jednotlivá projekce získaná pro daný úhel natočení). V pravém horním rohu se pak vykreslí (popř. průběžně vykresluje) soubor nasnímaných projekcí. (Na Obr. 3.4 vlevo je zachycen konečný stav programu po vykonání 180-ti nasnímaných projekcí. Ve spodních polích jsou zachyceny situace po sejmutí 180-té projekce.)

Jádro programu *Program1* spočívá ve funkci *generuj*, která obstarává generování souboru projekcí. Podrobněji bude tato funkce popsána v části 3.1.2.

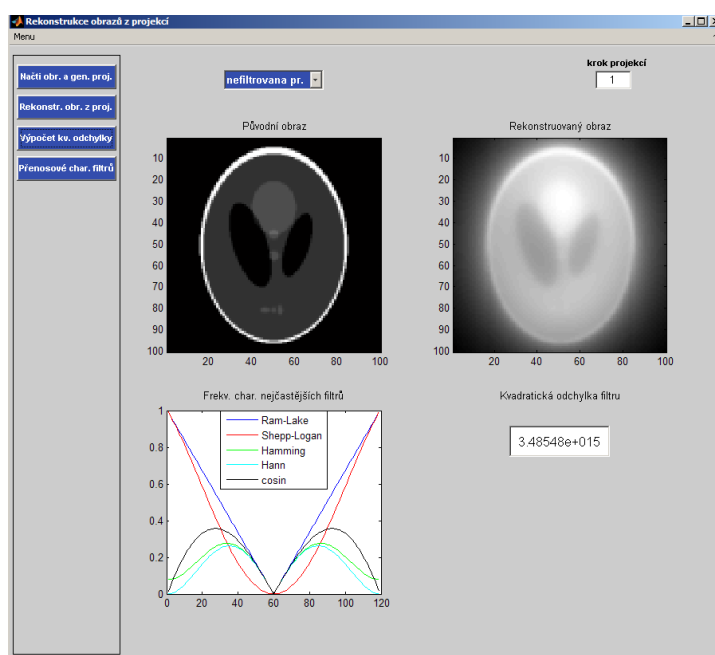
Program2

Tento program (viz Obr. 3.4 vpravo) má za úkol ze souboru projekcí rekonstruovat

zpět původní obraz. Rekonstrukci obrazu z projekcí je možno volit buď jako prostou zpětnou projekci, nebo filtrovanou zpětnou projekci s možností výběru nejpoužívanějších filtrů. Dále je možné zobrazit přenosovou a impulzní charakteristiku zvoleného filtru.

Uživatel stiskem tlačítka „Načti obr.“ vyvolá dialogové okno pro výběr požadovaného obrázku. Zvolený obraz se zobrazí v levém horním rohu. Dále je potřeba vytvořit soubor projekcí stisknutím tlačítka „Generuj projekce“. Až potom je funkčnost programu *Program2* shodná s programem *Program1*. Po vytvoření souboru projekcí je možno původní obraz rekonstruovat, a to s možností výběru z několika nabízených filtrů. Stisknutím tlačítka „Rekonstr. obr. z proj.“ se na pozadí programu vyvolá buď funkce *zp proj* pro prostou zpětnou proj. nebo funkce *filtravana* pro projekci filtrovanou (záleží na tom, zda uživatel použije některý z nabízených filtrů). Tyto funkce provádí výpočet rekonstrukce původního obrazu ze souboru projekcí a budou podrobněji popsány v části 3.1.3. Po výpočtu se rekonstruovaný obraz zobrazí v pravém horním rohu. Zobrazení frekvenční nebo impulzní charakteristiky se vyvolá stisknutím tlačítek „Frekvenční char.“ a „Impulzní char.“.

(Na Obr. 3.4 vpravo je v levém horním rohu zobrazen původní obraz, v levém pak obraz rekonstruovaný filtrem Shepp-Logan. Ve spodní části je zobrazena frekvenční a impulsová charakteristika použitého filtru.)



Obr. 3.5: Uživatelské rozhraní funkce *Program3*.

Program3

Tento program (viz Obr. 3.5) má za úkol zobrazit přenosové charakteristiky všech filtrů

v jednom grafu, získat soubor projekcí, provést rekonstrukci obrazu a pomocí kvadratické odchylky dále kvantitativně popsat vliv jednotlivých parametrů (počet sejmutých projekcí a zvolený filtr) na kvalitu rekonstruovaného obrazu.

Uživatel nejdříve načte vstupní obraz, který je ihned transformován na soubor projekcí. Poté může provést rekonstrukci původního obrazu buď pomocí konkrétního filtru, nebo prostou zpětnou projekcí. Výsledný obraz se poté zobrazí v levém horním rohu.

3.1.1 Zdrojový kód pro tvorbu číslicových filtrů

Číslicové filtry tvoří základ filtrované rekonstrukce obrazu z projekcí. Návrh nejčastěji používaných filtrů byl také součástí této práce. Způsobů, kterými lze filtry zkonstruovat, je více. V této práci jsem zvolil metodu vzorkování frekvenční charakteristiky, a to pro její jednoduchost.

Metoda spočívá v návrhu vhodné přenosové frekvenční charakteristiky ve spojitě oblasti a v její následné diskretizaci vzorkováním. Vzorkováním získáme diskrétní frekvenční charakteristiku, odkud lze velmi snadno pomocí zpětné diskrétní Fourierovy transformace obdržet impulsní charakteristiku požadovaného filtru. Tuto pak aplikujeme na jednotlivé projekce v prostorové oblasti při zpětné filtrované rekonstrukci obrazu (viz 2.2.1).

Průběhy frekvenčních charakteristik jsou zadány předpisy v části 2.2.3. V tomto programu je vzorkování frekvenční charakteristiky realizováno funkcemi `RAMLAKE_F`, `COSINE_F`, `SHEPPLOGAN_F`, `HANN_F` a `HAMMING_F`. Jako ukázka zdrojového textu (viz Obr. 3.6 nahoře) je zvolena přenosová charakteristika filtru Ram-Lake a přen. charakteristika Hannova filtru – která je násobkem filtru Ram-Lake a Hannova okna. Zápis funkce na řádce č. 9 je normován tak, aby maximální přenos funkce byl roven jedné. Funkce *osa* v podstatě vytváří „diskretizaci“ přenosové funkce neboť každému bodu na ose x je přiřazena hodnota funkce H .

Získání impulsní charakteristiky obstarávají funkce `ramlake.h`, `cosine.h`, `shepplogan.h` a `hamming.h`. Pro získání impulsní charakteristiky z přenosové funkce slouží v matlabu příkaz `ifft(H)`, kde H je definovaná frekvenční přenosová charakteristika. Na Obr. 3.6 dole je zdrojový kód funkcí `ramlake.h` a `shepplogan.h`. Na řádce č. 5 je zavolána předchozí funkce, která vytváří frekvenční charakteristiku a na řádce č. 8 je z této funkce zpětnou Fourierovou transformací získána impulsní charakteristika.

Frekvenční a impulsní charakteristiky zkonstruovaných filtrů jsou zobrazeny na obr. 3.11.

```

1  function H = RAMLAKE_F(vstpar)
2
3
4 -    [N,xmin,xmax,B] = osa(vstpar);
5
6 -    H = [];
7 -    for i = xmin:xmax
8 -        w = (2*pi*i);
9 -        H = [H, abs(w) / abs(2*pi*B)];
10 -    end
11
12 -    RAMLAKE = H;
13
14 -    save matlab H RAMLAKE -append;

1  function H = HANN_F(vstpar)
2
3 -    [N,xmin,xmax,B] = osa(vstpar);
4
5
6 -    A = [];
7 -    for i = xmin:xmax
8 -        w = abs(2*pi*i);
9 -        A = [A, 0.5*(1+cos(w/N))];
10 -    end
11
12 -    H = RAMLAKE_F(vstpar).*A;
13 -    HANN = H;
14 -    save matlab H HANN -append;

1  function hh = ramlake_h(vstpar)
2
3
4 -    N = osa(vstpar);
5 -    H = RAMLAKE_F(vstpar);
6
7 -    H = [H((N+3)/2:N) H(1:(N+1)/2)];
8 -    h = real(ifft(H));
9 -    ramlake = fftshift(h);
10
11 -    hh = ramlake;
12
13 -    save matlab hh ramlake -append;

1  function hh = hann_h(vstpar)
2
3
4 -    N = osa(vstpar);
5 -    H = HANN_F(vstpar);
6
7 -    H = [H((N+3)/2:N) H(1:(N+1)/2)];
8 -    h = real(ifft(H));
9 -    hann = fftshift(h);
10
11 -    hh = hann;
12
13 -    save matlab hh hann -append;

```

Obr. 3.6: zdrojový kód frekvenční (nahore) a impulsní (dole) charakteristiky filtrů. Vlevo: Ram-Lak filtr, vpravo: Hannův filtr.

3.1.2 Zdrojový kód pro generování souboru projekcí

Jádro programu *Program1* spočívá ve funkci *generuj*. Ta obstarává generování souboru projekcí. Jak bylo popsáno v teorii, soubor projekcí vzniká na základě měření spojitě změny paprskových integrálů pod spojitě se měnícím úhlem ϑ . Následující zdrojový kód aplikace MATLAB tuto operaci simuluje (viz obr. 3.7).

Důležitá vstupní proměnná, kterou bude uživatel aplikaci předávat, je krok ve stupních (krok), která, jak už bylo zmíněno, přímo ovlivňuje kolik projekcí se bude generovat (pro hodnotu proměnné $\text{krok} = 1$ bude sejmuto 180 projekcí, pro $\text{krok} = 90$ budou sejmuty pouze dvě projekce). Proměnnou krok zadává uživatel vložení hodnoty do editačního panelu v hlavním dialogovém okně. Její načtení je uskutečněno na řádcích 4-5.

```

1  function generuj(x)
2
3  -   load matlab;
4  -   krok = str2num(get(findobj('Tag','edit'),'String'));
5  -   theta = 0:krok:179.999;
6
7  -   [sy,sx] = size(I);
8  -   M = sx+sy;
9  -   projekci = length(theta);
10
11 -   scan=zeros(M,180);
12 -   for i = 1:projekci
13 -       I_rot = imrotate(I,360-theta(i),'bilinear');
14 -       projekce = sum(I_rot);
15 -       size_proj = max(size(projekce));
16 -       posun = round((M-size_proj)/2);
17 -       scan(posun+1:posun+size_proj,theta(i)+1) = projekce';
18 -   end
19
20 -   save matlab theta M sx sy scan -append;

```

Obr. 3.7: Funkce aplikace MATLAB pro generování souboru projekcí.

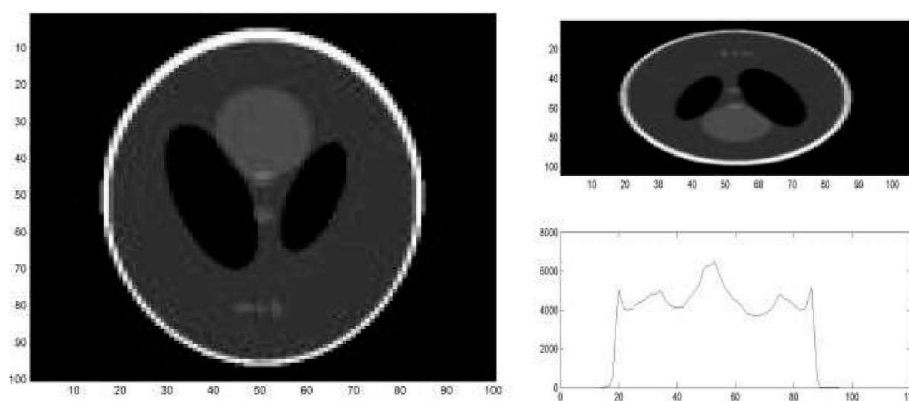
Poté se načte vstupní obraz, na který bude aplikována Radonova transformace, ten se uloží do proměnné I , která byla uložena voláním funkce *nacti* do souboru *matlab.mat*. Na řádcích 7-8 byly vytvořeny proměnné sy , sx a M , které budou později sloužit pro správné uložení jednotlivých projekcí do matice *scan* a také při procesu rekonstrukce. Na řádce 9 je zjišťována velikost matice θ . Tento údaj je důležitý, protože udává, kolik projekcí bude třeba vykonat, tj. kolikrát se vykoná cyklus *for*.

Dále se nachází jádro celého algoritmu. Do proměnné *Irot* je ukládán transformovaný obraz, který je ovšem v každém běhu cyklu natočen o úhel $360 - \theta(i)$. Jde zde vlastně o změnu úhlu ϑ , jak bylo popisováno v teorii ve vztahu (2.3) a (2.4). Po natočení obrázku dojde příkazem *sum(I_rot)* na řádce 14 k sejmutí množiny paprskových integrálů pro daný úhel ϑ a dojde tak k vytvoření jedné projekce, která je uložena do proměnné *projekce*. Příkaz *sum(I_rot)* provádí součet všech prvků ve sloupci matice. Operace se postupně provede pro všechny sloupce. Tato část procesu v podstatě vyjadřuje změnu vzdálenosti přímky p od středu souřadné soustavy.

Uvedeným postupem byl získán vektor *projekce*, který chceme uložit do matice *scan*. Postupným ukládáním takových vektorů každé sejmuté projekce do této matice dojde k vytvoření výsledného sinogramu. Matice *scan* má pro daný obrázek velikost 200×200 , vektor *projekce*, který chceme uložit do matice *scan* má však proměnnou velikost v závislosti na úhlu natočení matice *I_rot*. Je tedy třeba nějakým způsobem ošetřit to, aby se vektor projekce uložil na správné místo do matice *scan*. Na řádce 16 si proto spočítáme, jak velký je rozdíl mezi rozměry matice *scan* a *projekce*. Po vypočítání těchto hodnot můžeme uložit vektor *projekce* do matice *scan* pomocí indexování matice. Jelikož požadujeme, aby se vektor *projekce* do matice *scan* uložil jako sloupec, provedeme jeho transpozici. Cyklus *for* se takto vykoná několikrát po sobě, nejvíce 180krát, a program se poté ukončí. Po skončení

běhu cyklu *for* se na obrazovku zobrazí soubor všech sejmutých projekcí. (viz obr 3.4)

Naprosto stejně funguje i program *nacti_anim* (který slouží k animaci celého procesu), jediný rozdíl je pouze v tom, že vektor *projekce* stejně jako matice *scan* a také pootočený obrázek *I_rot* je zobrazen na obrazovku pro každý úhel natočení (každý běh cyklu *for*). Na monitor se data vykreslují postupně, lze tedy pozorovat souvislosti mezi celkovým souborem projekcí, jednotlivými projekcemi, počtem projekcí a úhlem natočení.



Obr. 3.8: Na levém obrázku je znázorněn vstupní obraz před transformací. Na pravém obrázku nahoře je snímek uložený v matici *I_rot*, pootočený o 180° a snímku pod ním jemu odpovídající projekce (osa x – detektor, osa y – hodnota paprskového integrálu).

3.1.3 Popis algoritmu rekonstrukce projekcí prostou i filtrovanou zpětnou projekcí

I zde (vzhledem k tomu, že se jedná o podstatu simulace zpětné rekonstrukce), bude popsán zdrojový kód. V teoretickém úvodu byla vysvětlena podstata algoritmu prosté zpětné projekce. Její princip spočívá v naplnění prázdné matice pod určitým úhlem jednou projekcí. Úhel, pod kterým vkládáme projekci do matice, je definován úhlem, pod kterým byla daná projekce získána. V případě prosté zpětné projekce jsou jednotlivé projekce vkládány do této prázdné matice tak, jak byly původně získány, tj. nedochází k jejich filtraci žádným filtrem. V případě filtrované zp. projekce dochází ještě před uložením jednotlivé projekce do prázdné matice k její filtraci. Vzhledem k tomu, že v tomto kroku spočívá jediný rozdíl mezi funkcí *zp_proj* (realizující prostou zp. proj.) a *filtrovana* (realizující filtrovanou zp. proj.), zde bude dále popsána jen funkce realizující filtrovanou zp. projekci. Zdrojový kód je na obr. 3.9.

Na začátku je třeba nejdříve načíst proměnné, které byly definovány ve skriptu pro Radonovu transformaci. Jedná se především o soubor projekcí *scan* a proměnné *M* a *theta*. Ty

vyvoláme načtením souboru `matlab.mat`, do kterých byly uloženy při generování souboru projekcí. Vzhledem k tomu, že budeme provádět filtrování projekcí v prostorové oblasti, je nutné nejprve si načíst impulsní charakteristiku zvoleného filtru, kterou budeme konvolovat s jednotlivými projekcemi. To se děje na řádce 5, kde voláme funkci `filtry`. Jak bylo řečeno v úvodu, budeme postupně naplňovat prázdnou matici jednotlivými projekcemi. Příkazem na 12. řádce si vytvoříme prázdnou matici o velikosti M .

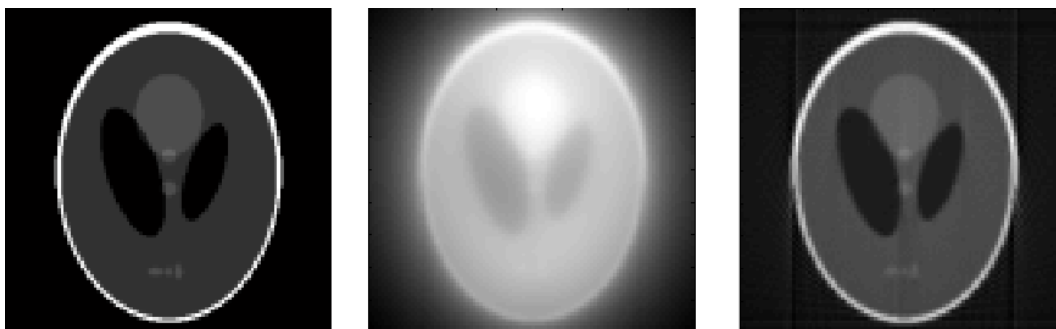
```

1  function filtrovana(vstpar)
2
3
4 -     load matlab;
5 -     projekci = length(theta);
6 -     [H,hh] = filtry (vstpar);
7
8 -     if hh==0
9 -         zp_proj(vstpar);
10 -    else
11
12 -        I_rek = zeros(M,M);
13 -        for i = 1:projekci
14 -            proj = scan(:,theta(i)+1)';
15 -            filtr_proj = conv(hh, proj);
16 -            filtr_proj = filtr_proj((length(hh)-1)/2+1:end-(length(hh)-1)/2);
17 -            rozmaz = repmat(filtr_proj,M,1);
18 -            rozmaz = imrotate(rozmaz,theta(i),'bilinear');
19 -            dy = round((size(rozmaz,1) - M) / 2);
20 -            dx = round((size(rozmaz,2) - M) / 2);
21
22 -            I_rek = I_rek + rozmaz(1+dy:dy+M,1+dx:dx+M);
23 -        end
24
25 -        I_rekk=I_rek(sx/2:sx/2+sy-1,sy/2:sy/2+sx-1);
26 -        set(h9,'Visible','on');
27 -        axes(a2); imagesc(I_rekk); colormap(gray);
28
29 -        save matlab I_rekk -append;
30
31 -    end

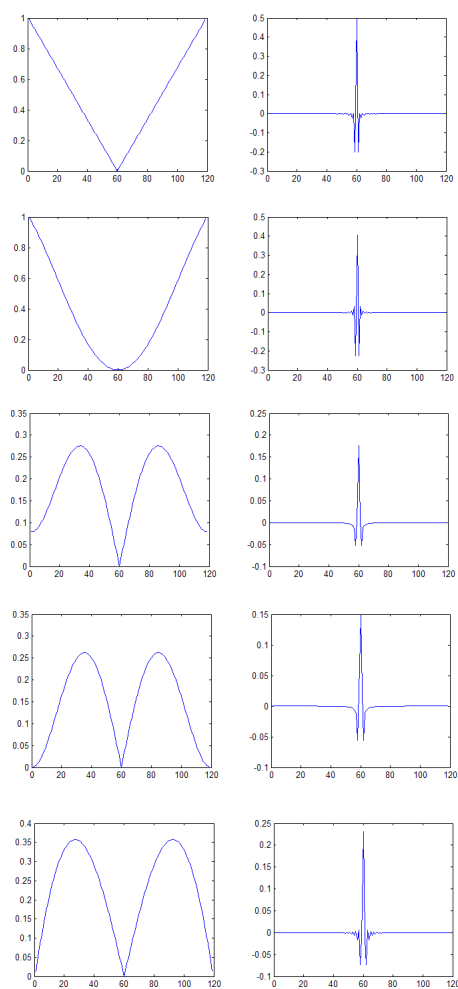
```

Obr. 3.9: Funkce aplikace MATLAB pro filtrovanou rekonstrukci obrazů z projekcí.

V těle funkce `for` pak bude probíhat samotný výpočet. Příkaz na řádce 15 provádí konvoluci projekce s impulsní charakteristikou filtru, příkaz na řádce 17 způsobí, že se vytvoří matice `rozmaz`, která se pomocí funkce `repmat` naplní tak, že v každém z M řádků matice `rozmaz` bude jeden sloupec (určen hodnotou $\theta(i)+1$) z matice `scan`. Následuje natočení matice `rozmaz` o jistý uhel, který odpovídá úhlu, pod kterým byla daná projekce získána (řádek 18). Na řádce 19-20 je znovu proveden výpočet, který srovná rozměry matice `rozmaz` a I_{rek} (M). Tento postup je nutný k tomu, abychom mohli v následujícím kroku provést bezproblémové sečtení matice `rozmaz` a I_{rek} . Na řádce 22 je pak proveden součet matice `rozmaz` s maticí I_{rek} , což je vlastně naplnění prázdné matice jednou projekcí pod daným úhlem. V průběhu cyklu `for` se takto naplní matice I_{rek} všemi projekcemi, které byly sejmuty a dojde k vytvoření rekonstruovaného obrazu. Na řádce 25 je pak už jen výsledný obraz „vystřižen“, aby nabýval svých původních rozměrů.



Obr. 3.10: Vliv filtru na podobu rekonstruovaného obrazu. Vlevo je původní snímek. Uprostřed je rekonstruovaný snímek pomocí prosté zp. projekce. Snímek vpravo je pak rekonstruovaný obraz po aplikaci Ram-Lak filtru. (Počet sejmutých projekcí byl 180)

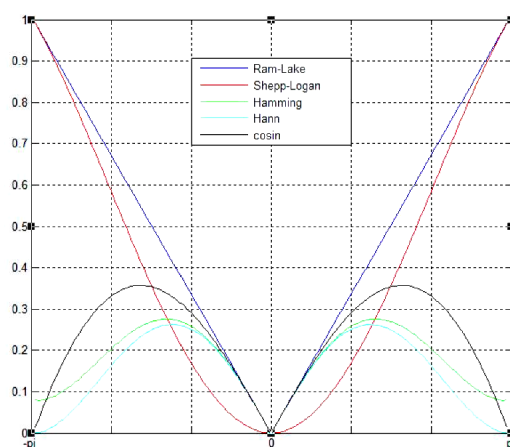


Obr. 3.11: Frekvenční a impulsní charakteristiky vytvořených filtrů.
Shora: Ram-Lak, Shepp-Logan, Hammingův, Hannův, kosinový.

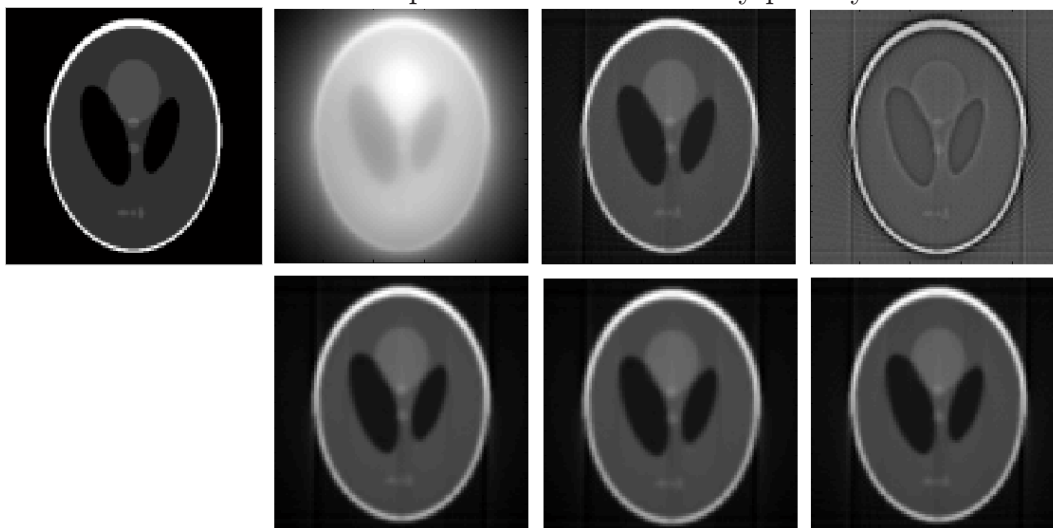
3.2 Srovnání zkonstruovaných filtrů

3.2.1 Kvalitativní porovnání

Frekvenční charakteristiky všech filtrů použitých v této práci je možné porovnat podle obr. 3.12. Hammingův, Hannův a kosinový filtr mají velmi podobné charakteristiky. Dá se tedy očekávat, že zpětně rekonstruované obrazy si budou na první pohled velmi podobné. Rozdíly budeme vnímat především mezi těmito filtry a filtry Ram-Lak a Shepp-Logan, které mají velký přenos na vysokých frekvencích.



Obr. 3.12: Frekvenční přenosové charakteristiky použitých filtrů



Obr. 3.13: Porovnání výstupních obrazů za použití různých filtrů. Zleva: originální obraz, nefiltrovaný, filtrovaný Ram-Lakenovým, Shepp-Loganovým, Hannovým, Hammingovým a kosinovým filtrem

Účinek jednotlivých filtrů na výstupní obraz lze pozorovat na obr. 3.13. Z něho je patrné, že opravdu filtry Ram-Lak a Shepp-Logan mají větší přenos na vyšších frekvencích, což se v obraze projevuje zobrazením jemnější struktury, jsou na nich však vidět i ty složky signálu, které nejsou pro uživatele žádoucí a mohou být naopak rušivé.

Naproti tomu Hannův, Hammingův a kosinový filtr vysokofrekvenční složky výrazně utlumil, zbavil tím rekonstruovaný obraz nežádoucích artefaktů, avšak za cenu nižší ostroty obrazu. Celkově se od sebe (subjektivně vnímáno) tyto tři filtry ve svém účinku příliš neliší.

Na obr. 3.13 můžeme také vidět výstupní obraz nefiltrované zpětné projekce. Vliv hvězdového artefaktu obraz znehodnocuje natolik, že rekonstruovaný obraz je téměř nepoužitelný k jakémukoliv účelu. Lze z toho usoudit, že rekonstrukce obrazů z projekcí se bez účinné filtrace zřejmě neobejde.

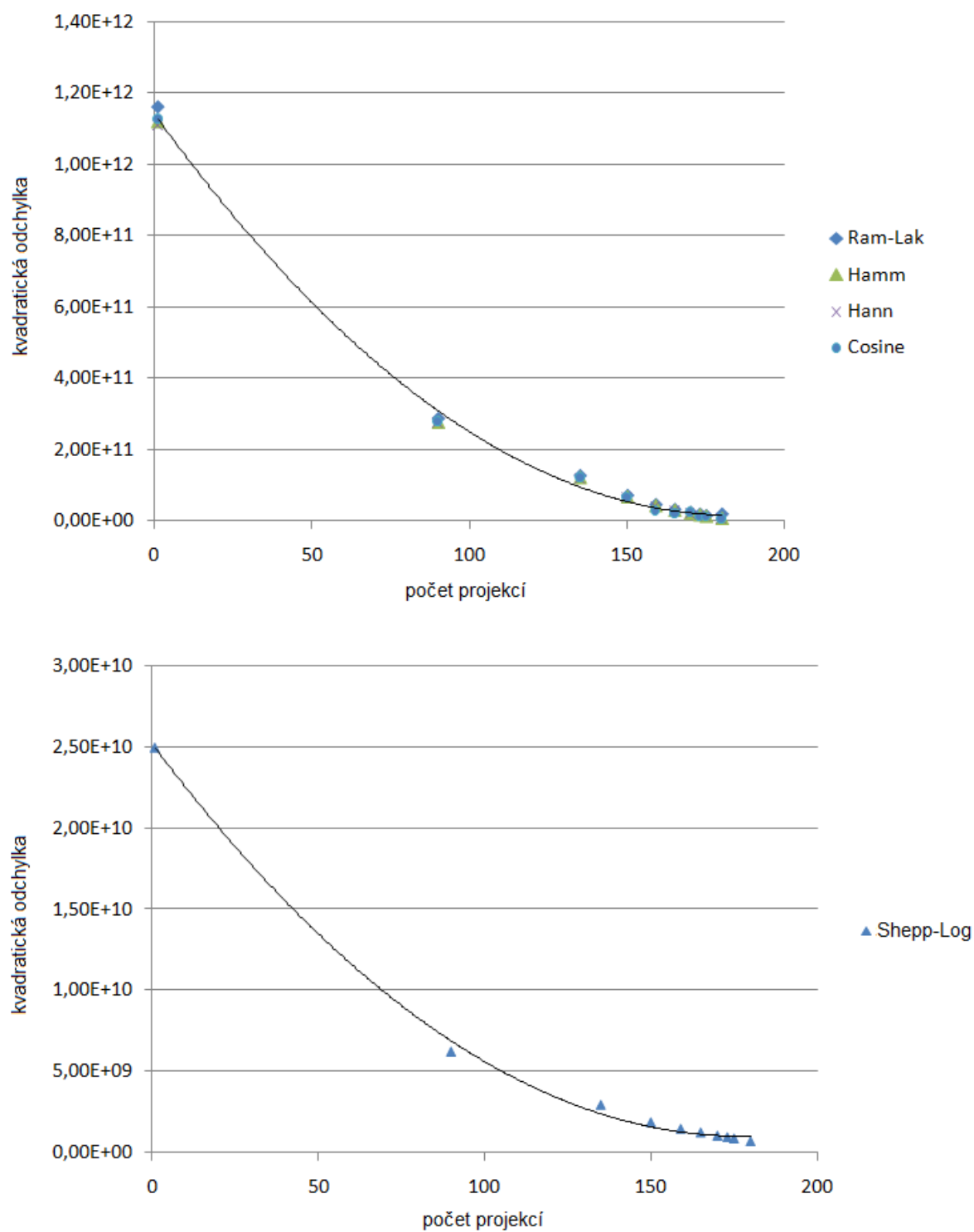
3.2.2 Kvantitativní porovnání

Dosud pozorované kvalitativní rozdíly v obrazech, které dostáváme při použití jednotlivých způsobů filtrace, je možné vyjádřit kvantitativně pomocí kvadratické odchylky - čím je hodnota tohoto ukazatele nižší, tím je možné rekonstruovaný obraz považovat za kvalitnější. Je zřejmé, že takové srovnání filtrů je relevantní za předpokladu, že je při filtraci zvolena tatáž obrazová předloha a stejný počet prováděných projekcí. Jak je patrné z dat uvedených v Tab. 1, v tomto smyslu se ze sestavených filtrů jako nejúčinnější projevoval filtr Shepp-Loganův, zatímco zbývající čtyři vykazovaly pro zvolený počet projekcí téměř totožné hodnoty kvadratické odchylky. Kvadratické odchylky pro nefiltrovanou zpětnou projekci pak podle očekávání byly ještě o dva až tři řády vyšší.

Vedle vzájemného srovnání filtrů bylo taktéž hodnot kvadratické odchylky použito ke kvantitativnímu popisu závislosti kvality obrazu na počtu provedených projekcí u jednotlivých filtrů. Podle očekávání hodnoty této charakteristiky pro zvolené filtry klesaly se vzrůstajícím počtem projekcí, z nichž byl rekonstruovaný obraz vytvořen. Obdržené hodnoty jsou pro jednotlivé filtry vyneseny v grafech na Obr. 3.15 (vyznačené proložení dat je jen orientační).

Tab. 1: hodnoty kvadratických odchylek pro jednotlivé filtry.

počet projekcí	KV(Ram-Lak)	KV(Shepp-Log)	KV(Hamm)	KV(Hann)	KV(Cosine)
1	1,16E+12	2,50E+10	1,12E+12	1,11E+12	1,13E+12
90	2,87E+11	6,24E+09	2,77E+11	2,76E+11	2,81E+11
135	1,27E+11	2,96E+09	1,22E+11	1,21E+11	1,24E+11
150	7,12E+10	1,88E+09	6,81E+10	6,79E+10	6,91E+10
159	4,57E+10	1,48E+09	4,33E+10	4,32E+10	3,06E+10
165	3,20E+10	1,26E+09	3,00E+10	2,99E+10	2,30E+10
170	2,42E+10	1,06E+09	2,25E+10	2,24E+10	2,81E+10
173	1,92E+10	9,64E+08	1,76E+10	1,76E+10	1,37E+10
175	1,47E+10	8,88E+08	1,33E+10	1,33E+10	1,37E+10
180	1,92E+10	7,12E+08	7,58E+09	7,52E+09	7,86E+09



Obr. 3.14: 17. Frekvenční přenosové charakteristiky použitých filtrů

3.3 Zadání laboratorní úlohy

1. V aplikaci MATLAB spusťte program s názvem *Program1*. Ovládacími tlačítky načtěte z disku soubor „dva impulsy.bmp“, nechte krok projekcí nastaveno na hodnotě 1 a stiskněte tlačítko „generuj projekce“. V pravém horním okně se zobrazí soubor projekcí. Jak tyto souvisí s původně načteným obrázkem?
2. Pro snazší pochopení nebo pro ověření svých předpokladů nyní označte políčko „animuj“ a spusťte znovu generování projekcí. Animace ukazuje jak se spojitě mění natočení původního obrázku, ukazuje také spojitý „sběr“ jednotlivých projekcí do souboru projekcí. Vznik projekcí si je možno představit tak, že podél celé spodní hrany rotovaného obrázku jsou vysílány kolmo na obrázek „paprsky“, jejichž průmět tvoří jednu projekci. Všimněte si, že průmět je pouze tam, kde je nějaký bod. (V praxi rotuje snímač kolem předmětu).
3. Načtěte soubor „phantom.bmp“. Pro úhly snímání 20° , 40° , ..., 180° vypočtěte kvadratickou odchylku, která vypovídá o kvalitě rekonstruovaného obrazu. Nejdříve bez použití filtrace a po té s použitím filtru Shepp-Logan. Která z rekonstrukcí by měla mít menší kvadratickou výchylku oproti originálu?

4 Závěr

Podle zadání práce byla prostudována teorie počítačové tomografie, na základě níž pak bylo s využitím programu MATLAB zrealizováno uživatelské prostředí simulující výpočetní rentgenovou tomografií. Vytvořený program, určený k podpoře výuky umožňuje provádět jak prostou, tak i filtrovanou zpětnou projekci. Celé prostředí programu bylo se záměrem jeho užití jako pedagogické pomůcky voleno tak, aby jej bylo možné ovládat co nejintuitivněji. K programu bylo též vytvořeno zadání laboratorní úlohy. Bylo také provedeno kvantitativní porovnání vlivu použitého filtru na výsledný rekonstruovaný obraz.

Literatura

- [1] JAN, J. Číslicová filtrace, analýza a restaurace signálů. VUTIUM, Brno, 2002. 427s. ISBN 80-214-2911-9
- [2] KAK, Avinash C., SLANEY, M. Principles of computerized tomographic imaging. IEEE PRESS, New York, 1987. 322s. ISBN 0-87942-198-3
- [3] DRASTICH, A. Tomografické zobrazovací systémy. Skriptum. Brno: FEKT VUT v Brně, 2004. 208s. ISBN 80-214-2788-4
- [4] ZAPLATÍLEK, K., DOŇAR, B. MATLAB pro začátečníky. BEN – technická literatura, Praha, 2005. 152s. ISBN 80-7300-175-6
- [5] ZAPLATÍLEK, K., DOŇAR, B. MATLAB tvorba uživatelských aplikací. BEN – technická literatura, Praha, 2005. 216s. ISBN 80-7300-133-0
- [6] WIKIPEDIE, otevřená encyklopedie. Radonova transformace [online]. Převzato: 22. 3. 2011. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Radonovatransformace>.
- [7] WIKIPEDIA, the free encyclopedia. Windows function [online]. Převzato: 18. 4. 2011. Dostupný z WWW: http://en.wikipedia.org/wiki/Window_function