

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF BIOMEDICAL ENGINEERING

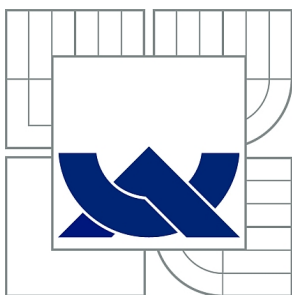
ODSTRANĚNÍ ANESTEZIOLOGICKÉ TRUBICE Z PEDIATRICKÝCH CT  
DAT

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

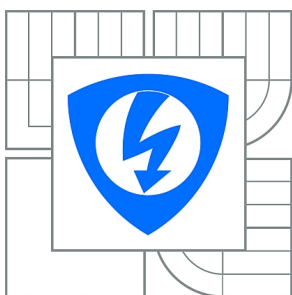
MARIE ŠEJNOHOVÁ

BRNO 2013



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ**

**ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF BIOMEDICAL ENGINEERING

## **ODSTRANĚNÍ ANESTEZIOLOGICKÉ TRUBICE Z PEDIATRICKÝCH CT DAT**

REMOVAL OF ANESTHESIA TUBE FROM PEDIATRIC CT DATA

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MARIE ŠEJNOHOVÁ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. PETR WALEK**

BRNO 2013



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav biomedicínského inženýrství

# Bakalářská práce

bakalářský studijní obor

**Biomedicínská technika a bioinformatika**

**Studentka:** Marie Šejnohová

**ID:** 137261

**Ročník:** 3

**Akademický rok:** 2012/2013

## NÁZEV TÉMATU:

### Odstranění anesteziologické trubice z pediatrických CT dat

#### POKYNY PRO VYPRACOVÁNÍ:

1) Seznamte se se zpracováním obrazů v programovém prostředí MATLAB. Popište koncept algoritmu pro odstranění anesteziologické trubice z CT dat s využitím Houghovy transformace a teoreticky popište metody zpracování obrazů, které budete používat. 2) Navrhněte a realizujte předzpracování obrazů, které je nezbytné pro realizaci Houghovy transformace. 3) Naprogramujte algoritmus Houghovy transformace pro vyhledávání kružnic v jednom dvourozměrném řezu daty. 4) Proveďte detekci pozice anesteziologické trubice vyhledáváním soustředných kružnic v Houghově prostoru a nalezenou trubici vhodně odstraňte. 5) Algoritmus rozšířte na celý objem dat a přizpůsobte jej pro zpracování na PC, které umožňuje paralelní zpracování dat v prostředí MATLAB. Dbejte na to, aby celý proces odstranění trubice byl co nejméně časově náročný. 6) Proveďte diskuzi a zhodnocení dosažených výsledků.

#### DOPORUČENÁ LITERATURA:

[1] JAN, Jiří. Medical Image Processing, Reconstruction and Restoration: Concepts and Methods. Boca Raton: CRC Press, 2005, ISBN 0-8247-5849-8

[2] KHAIROSFAIZAL, W. a NORAINI, A. Eyes Detection in Facial Images using Circular Hough Transform. Signal Processing & Its Applications. 2009, 5th International Colloquium, s. 238-242.

**Termín zadání:** 11.2.2013

**Termín odevzdání:** 31.5.2013

**Vedoucí práce:** Ing. Petr Walek

**Konzultanti bakalářské práce:**

**prof. Ing. Ivo Provazník, Ph.D.**

*Předseda oborové rady*

#### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Cílem této semestrální práce je detekovat a následně odstranit anesteziologickou trubici z pediatrických CT dat pomocí Houghovy transformace. V první části se věnuje předzpracování obrazů s využitím hranové detekce. Popisuje zde jednotlivé hranové detektory založené na výpočtech 1. a 2. diferencí. Dále vysvětluje princip hranové detekce pomocí těchto detektorů. Dále pojednává o Houghově transformaci, kde popisuje parametrický prostor přímky a kružnice pro jejich detekci. Následuje popis transformace obrazu do parametrického. V praktické části se realizuje hranová detekce a Houghova transformace. Houghův prostor se musí normalizovat, aby ho šlo lépe vyhodnotit. Při vyhodnocování se hledají soustředné kružnice odpovídající hranám anesteziologické trubice, kterou je potom možné odstranit.

## **KLÍČOVÁ SLOVA**

Houghova transformace, hranová detekce, akumulátor, detektor, anesteziologická trubice, kružnice, odstranění

## **ABSTRACT**

The goal of this semestral thesis is to detect and subsequently to eliminate an anaesthesiological tube from a pediatric CT data using a Hough transform. It describes particular edge detectors based on the first and the second differences. In addition, it explains a principle of the edge detection using to this detectors. Then there is talked about the Hough Transform where is described a parametric space for lines and circles detection. Here follows a description of image to the parametric space. In a practical part there is implementation of the edge detection and the Hough transform. The Hough space must be normalized to better evaluation. At the evaluation there are found a concentric circles corresponding with edges of an anaesthesiological tube which is then possible to remove.

## **KEYWORDS**

Hough transform, edge detection, accumulator, detector, anaesthesiological tube, circles, removing

ŠEJNOHOVÁ, M. Odstranění anesteziologické trubice z pediatrických CT dat.  
Brno: FEKT VUT v Brně, Ústav Biomedicínského inženýrství, 2012. 20 s., 1 příl.  
Vedoucí práce Ing. Petr Walek.

## PROHLÁŠENÍ

Prohlašuji, že svoji bakalářskou práci na téma Odstranění anesteziologické trubice z pediatrických CT dat jsem vypracovala samostatně pod vedením vedoucího semestrálního projektu 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 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 části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009Sb.

V Brně dne

podpis autora .....

## PODĚKOVÁNÍ

Ráda bych poděkovala vedoucímu bakalářské práce panu Ing. Petru Walkovi za odborné vedení, trpělivost, konzultace a jeho cenné rady.

Brno .....

podpis .....

# OBSAH

<b>Úvod</b> .....	<b>6</b>
<b>1. Hranová detekce</b> .....	<b>7</b>
1.1 Diskrétní diferenční operátory .....	8
1.1.1 Hranové detektory založené na prvních a druhých diferencích.....	8
1.2 Hranové detektory .....	11
1.2.1 Zero-crossing .....	12
1.2.2 Operátor Laplaciá gaussianu (LoG) .....	12
1.3 Další postupy hranové detekce .....	14
1.4 Šum v obrazu .....	14
1.5 Využití hranové detekce .....	15
<b>2. Houghova transformace</b> .....	<b>16</b>
2.1 Parametrický prostor přímky.....	16
2.2 Houghova transformace pro detekci kružnic .....	18
2.3 Akumulátor .....	19
2.5 Provedení.....	20
2.5.1 Jak skladovat data .....	20
2.5.2 Jak nakreslit kružnici v diskretním prostoru .....	20
2.6 Další rozšíření Houghovy transformace .....	20
<b>3. Realizace hranové detekce a Houghovy transformace</b> .....	<b>21</b>
3.1 Načtení CT snímků .....	21
3.2 Hranová detekce .....	21
3.3 Houghova transformace .....	24
<b>4. Normalizace Houghova prostoru</b> .....	<b>26</b>
<b>5. Vyhodnocení Houghova prostoru</b> .....	<b>27</b>
5.1 Blokované schéma vyhodnocení Houghova prostoru.....	27
5.2 Nalezení vhodných kružnic a vykreslení do původního obrazu .....	28
5.3 Hledání soustředných kružnic .....	29
<b>6. Odstranění anesteziologické trubice</b> .....	<b>33</b>

6.1 Výběr vzorku pozadí .....	33
6.2 Vyříznutí vzorku pozadí a jeho zvětšení .....	34
6.3 Vytvoření nulové intenzity pro trubici a její vnitřek .....	35
6.3 Přičtení vzorku .....	36
<b>7. Zjednodušení Houghovy transformace .....</b>	<b>38</b>
<b>8. Rozšíření algoritmu na celý objem dat .....</b>	<b>39</b>
<b>9. Vytvoření paralelního prostředí .....</b>	<b>43</b>
9.1 Paralelní prostředí .....	43
9.2 Paralelní for cykly- PARFOR .....	43
9.3 Realizace paralelního prostředí .....	43
<b>10. Zobrazení objemu dat ve 3D.....</b>	<b>44</b>
<b>11. Závěr .....</b>	<b>45</b>
<b>Seznam literatury .....</b>	<b>46</b>
<b>Seznam symbolů, veličin a zkratk .....</b>	<b>47</b>

# Seznam obrázků

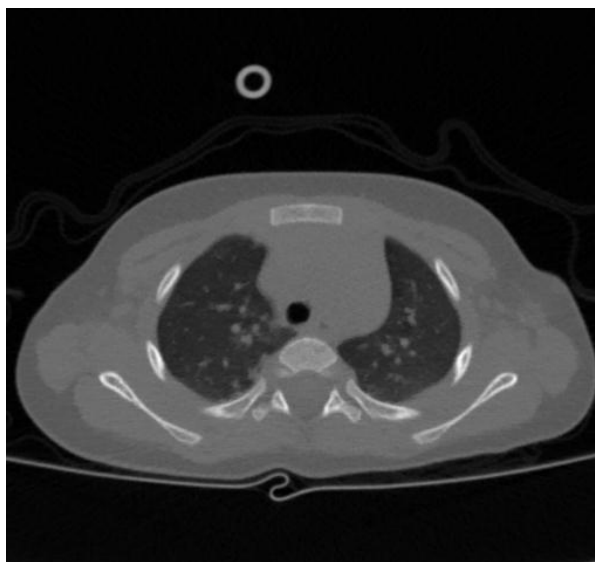
Obr.1.1 Profil hrany.....	7
Obr.1. 2 Profily odpovědí na operátory první a druhé derivace .....	9
Obr.1.3 Obrazy odvozené z detektorů první diference.....	13
Obr.1.4 Obrazy odvozené z detektorů druhé diference .....	14
Obr.2.1 Normálová parametrizace přímky [4].....	17
Obr.2.2 Parametrický prostor HT pro kružnici [3].....	18
Obr.2.3 Transformace z obrazové roviny do parametrického prostoru [3] .....	19
Obr.2.4 HT pro kružnici z řezu CT dat z Obr. 1 .....	19
Obr.3.1 Hranová detekce.....	24
Obr.3.2 Houghova transformace .....	26
Obr.5.1 Vykreslení vhodných kružnic.....	29
Obr.5.2 Nalezené soustředné kružnice .....	32
Obr.6. 1 Vybraný vzorek pozadí.....	34
Obr.6. 2 Vyříznutý a zvětšený .....	34
Obr.6. 3 Trubice a její vnitřek s hodnotou.....	36
Obr.6. 4 Přičtení vzorku k trubici a jejího .....	37
Obr.8. 1 1. snímek.....	39
Obr.8. 2 100. snímek.....	40
Obr.8. 3 200. snímek.....	40
Obr.8. 4 300. snímek .....	41
Obr.8. 5 357. snímek.....	41
Obr.8. 6 Odstranění anesteziologické trubice z vybraných CT snímků .....	42

# Úvod

Práce se zabývá detekcí a odstraněním anesteziologické trubice z CT dat s využitím Houghovy transformace (HT). Pediatrický pacient je připojen na dýchací přístroj, tzv. ventilátor pomocí umělé trubice. Tato trubice se na snímcích zobrazuje, a proto ji je nutné odstranit, viz Obr. 1. Prvním postupem je předzpracování obrazů pomocí hranové detekce, aby bylo možné zrealizovat algoritmus Houghovy transformace.

V roce 1959 publikoval Paul Hough matematickou operaci pro detekování přímek v obrazech, se kterou lze najít skupiny bodů ležících na stejné přímce [5]. Rosenfeld se zabýval podobným problémem, pouze místo bodů hledal protínající se přímky. Tato metoda vyžadovala transformaci každého bodu z obrazu do parametrického prostoru, tzv. Houghova prostoru. Klasická HT dokáže objevit pouze přímku. V roce 1972 publikovali R.O. Duda a P.E.Hart zobecněnou metodu, pomocí které lze nalézt další parametricky popsatelné křivky jako kružnici, parabolu, atd. Významnou úlohu hraje tato transformace při analýze textu, kdy se s její pomocí hledá řádek textu v obraze [1] [4].

Pro detekování anesteziologické trubice využijeme HT pro detekci kružnic. V Houghově prostoru se budou vyskytovat tři parametry, tzn. že vznikne trojrozměrný parametrický prostor. Takový prostor se také označuje jako akumulátor. Nejdříve však musíme obraz předzpracovat pomocí hranové detekce, aby anesteziologickou trubicí tvořily dvě soustředné kružnice. Hranovou detekcí se podrobně zabývá následující kapitola [1].



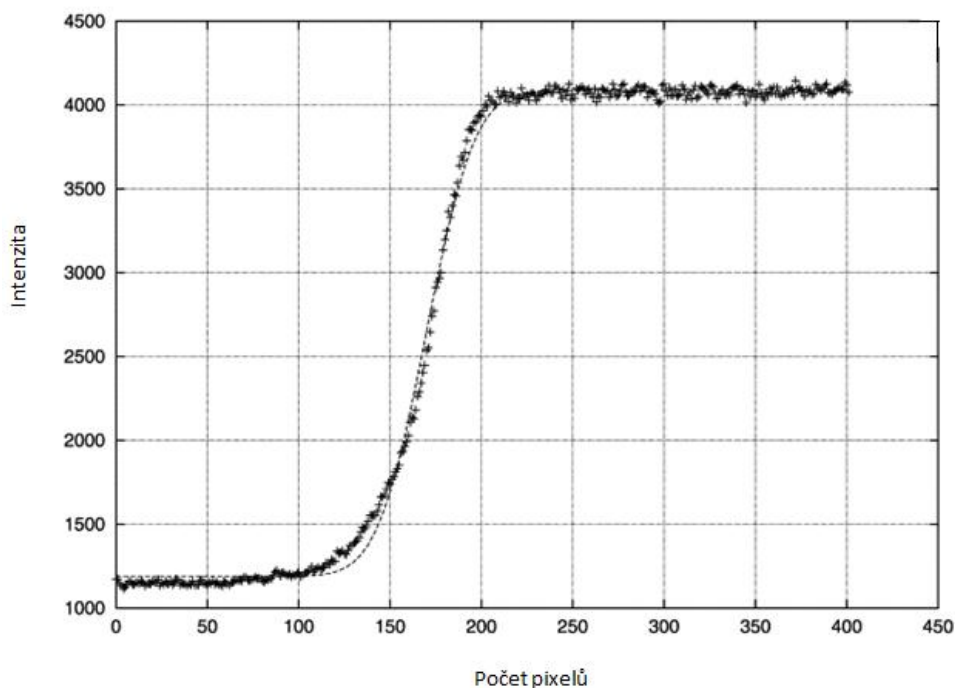
**Obr. 1** Jeden řez CT daty se zřetelně viditelnou anesteziologickou trubicí

# 1. Hranová detekce

Subjektivní kvalita obrazu závisí ve velké míře na ostrosti a zřetelnosti hran. Záleží na tom, kolik člověk dokáže vidět detailů při mezním zrakovém rozlišení. Rozpoznání tvarů nebo předmětů začíná určením hran. V praxi sem patří např. dostatečně velká a rychlá změna intenzity v malém okolí. Hranou může být hranice většího objektu v obrazu, rys předmětu nebo samostatně stojící čára či bod. Často však bývají detekovány jako hrany i lokální změny intenzity vzniklé na základě šumu. Měly by být vyloučeny následným zpracováním obrazu. Při jejich nedostatečné identifikaci se může zdát, že je v obrazu málo detailů.

Nízká ostrost lze v prostorové oblasti vyjádřit jako pomalá změna intenzity při průchodu přes okraje předmětu. Také se dá popsat ve frekvenční oblasti jako nízký podíl vysokofrekvenčních složek.

Existuje však významný rozdíl mezi rozlišením a ostrostití. Žádné zaostření nemůže zlepšit opravdové rozlišení obrazu. Ke zdokonalení rozlišení by se měly přidat právě chybějící vysokofrekvenční složky. Zaostřením se mohou vylepšit pouze již přítomné detaily, tzn. zvýší se strmost hran. Takto upravené okraje a hrany jsou potom lépe viditelné. Kvantitativní zhodnocení zaostření je založené na porovnání šířky profilu hrany, mezi 10 a 90 % amplitudy, před a po zpracování obrazu. Na Obr.1.1 vidíme příklad profilu, kde se vyskytuje hrana.



Obr.1.1 Profil hrany

Hranová detekce využívá analýzy obrazů pomocí lokálních parametrů, které počítají se skupinou pixelů. V praxi bývá analyzováno okolí konkrétního pixelu. Jako příklad může sloužit velikost lokálního gradientu určeného pro každý pixel, která je založená na intenzitách sousedních pixelů. Pomocí hodnot takových parametrů můžeme vytvořit nový, tzv. parametrický obraz. Další krok analýzy se nazývá segmentace nebo-li začlenění hodnoty parametru konkrétního pixelu nebo oblasti podle určitých kritérií. Segmentace využívá již zmiňovaných parametrických obrazů.

Většina znaků jsou definované pomocí lokálních parametrů buď v původní nebo transformované oblasti. Termín lokální znamená, že se zabýváme malou částí, která obklopuje pixel nesoucí určitý znak. Jeden z problémů představuje zvolit velikost a tvar okolí. Využitím statistických parametrů získáme menší odchylku při zvolení většího množství dat, a proto se dá považovat za více spolehlivou. Na druhou stranu, čím větší oblast analyzujeme, tím hrubší dostaneme prostorové rozlišení parametrického obrazu. Volba analyzované oblasti proto zůstává kompromisem závislejícím na charakteru obrazu a účelu konkrétní analýzy. V praxi můžeme začít z minima jako 2x1 nebo 1x2 pixely a skončit několika desítkami pixelů v průměru [1] [6] [8].

## 1.1 Diskrétní diferenční operátory

Rozmazání si můžeme v prostorové oblasti vyložit jako výsledek určité integrace nebo zprůměrování. Dá se proto očekávat, že pomocí opačných operací jako výpočet diferencí lze stanovit pozice hran. Detekce hran vyžaduje výpočet diferencí prvního nebo druhého řádu.

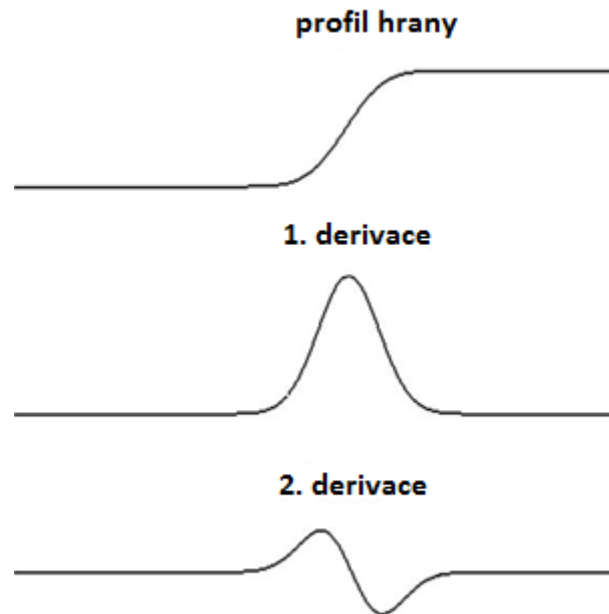
### 1.1.1 Hranové detektory založené na prvních a druhých diferencích

Parciální derivace  $\partial f(x,y)/\partial x$ ,  $\partial f(x,y)/\partial y$  vyjadřují směrové nebo-li anizotropní operátory. Reagují převážně na změny intenzity podél příslušné osy. Jinak nasměřované hrany vedou k odpovědi klesající se změnou úhlu od optima. Diferenční operátory nejvíce citlivé k hranám kolmým na jednotkový vektor  $s$ , jsou směrové parciální derivace:

$$\frac{\partial f}{\partial s} = \frac{\partial f}{\partial x} \cos \vartheta + \frac{\partial f}{\partial y} \sin \vartheta, \quad (1.1)$$

kde  $\vartheta$  představuje úhel mezi  $s$  a osou  $x$ . Profily výstupu obrazů odvozené od operátorů první a druhé derivace, můžeme vidět na Obr.1. 2.

Všechny směrové operátory a operátory k nim příbuzné jsou vhodné, když upřednostňujeme detekci určitého směru. Potom budou hrany k nim kolmé vynechány [1].



**Obr.1. 2** Profily odpovědí na operátory první a druhé derivace

Více často bývá však vyžadována detekce hran nezávislá na směru. Izotropní operátory využívají také parciální derivace. Mezi nejčastější patří velikost gradientu:

$$|f(x, y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}, \quad (1.2)$$

obvykle nesprávně nazývaní jako „gradient“ a další, Laplaceův operátor:

$$\mathcal{E}(f(x, y)) = f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}. \quad (1.3)$$

Hodnota gradientu obvykle poskytuje poměrně tlusté hrany, které by měly být dále zpracovány postupným ztenčováním.

V diskrétním prostředí digitálních obrazů musíme využít diferenčních operátorů založených na rozdílech mezi pixely. První diference jsou popsány vzorcem:

$$\Delta_x f_{i,k} = f_{i,k} - f_{i-1,k} \quad \Delta_y f_{i,k} = f_{i,k} - f_{i,k-1} . \quad (1.4)$$

Předpokládáme stejnou vzdálenost mezi jednotlivými vzorky při vzorkování. Potom jsou první derivace úměrné odpovídajícím diferencím mezi hodnotami pixelů. Existuje možnost nahradit druhé parciální derivace druhými diferencemi. Takové operátory jsou lineární a obvykle zde vynecháváme rozdělení na  $\Delta x$  nebo  $\Delta y$ . Nejvíce hranových detektorů je

založených na diferenčních operátorech. Patří mezi ně detektory vytvořené na základě gradientu. Zde dochází k vyhodnocení rychlosti změny intenzity v okolí určitého pixelu. Parciální derivace lze aproximovat směrovými diferencemi:

$$g_{i,k} = \sqrt{(\Delta_x f_{i,k})^2 + (\Delta_y f_{i,k})^2}. \quad (1.5)$$

Takový operátor je přibližně izotropní. Umocňování a odmocňování však patří mezi výpočetně náročnější operace, takže můžeme formulaci ještě více aproximovat použitím gradientu a vznikne:

$$g_{i,k} \approx \max(|\Delta_x f_{i,k}|, |\Delta_y f_{i,k}|). \quad (1.6)$$

Operátory využívající hodnotu gradientu už můžeme považovat za částečně anizotropní. Podle uvedených vzorců (1.5 a 1.6) lze skládat gradientní obrazy ve směru os  $x$  a  $y$ . Rozhodnutí, zda konkrétní pixel patří k hraně, uskutečňujeme porovnáním hodnoty lokálního gradientu se zvoleným prahem, který představuje parametr hranové detekce. Prah určuje kompromis mezi falešně pozitivní a falešně negativní odpovědí. Vyšší prah zmenšuje falešné hrany za cenu možných chybějících hran.

Diferenční operátory mohou být realizovány jako lokální operátory nebo-li masky. Pomocí nich jsou vytvořeny obrazy popisující difference jako funkce pozice. Výsledek se potom vypočítá pomocí konvoluce. Masky pro první a druhé difference mohou vypadat následovně:

$$\Delta_x f_{i,k}: \begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ nebo } \frac{1}{2} \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \Delta_y f_{i,k}: \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \text{ atd.}$$

$$\Delta_x^2 f_{i,k}: \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \Delta_y^2 f_{i,k}: \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix}. \quad (1.7)$$

Hodnoty diferencí mají obecně hodnoty v rozmezí  $\langle -f_{\max}, f_{\max} \rangle$ , zatímco dostupné hodnoty reprezentující obraz jsou v interval  $\langle 0, f_{\max} \rangle$ . Nezbytnou se zde stává normalizace obrazové matice. Také směrové difference pod úhlem  $\pm 45^\circ$  můžeme vyjádřit touto cestou:

$$\Delta_s f_{i,k}: \frac{1}{2\sqrt{2}} \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \text{a} \quad \frac{1}{2\sqrt{2}} \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (1.8)$$

Přírůstek v prostorové diagonále by měl být  $\sqrt{2}$  krát větší, když srovnáváme výsledky maskového operátoru. Nicméně však bývá zanedbaný ve prospěch výpočetní jednoduchosti. Součet posledních dvou masek v rovnici 4 vyjadřuje diskrétní Laplaceův operátor [1] [6].

$$\mathfrak{L}\{f_{i,k}\}: \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \quad (1.9)$$

## 1.2 Hranové detektory

Důležitou vlastnost charakterizující pixel představuje skutečnost, jestli se zde vyskytuje hrana. Globální hranová detekce aplikovaná na obraz poskytuje binární parametrický obraz. Využitím hranových detektorů vznikne obraz, kde pixely obsahující hrana se vykreslí bílou barvou s hodnotou jedna na černém pozadí hodnoty nula.

Pomocí Robertsova detektoru vypočítáme směrové diference konvolucí s maskami

$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \quad (1.10)$$

Výsledné hodnoty se potom sloučí do jednoho obrazu podle vzorců (1.5 a 1.6), kterými si, jak už jsem se jednou zmínila, vyjádříme gradienty ve směru os  $x$  a  $y$ .

Pokud máme dostupné částečné diference, dají se určit také lokální směry hran. Určíme je pomocí následujícího vztahu:

$$\theta_{i,k} = \arctan \frac{\Delta_y f_{i,k}}{\Delta_x f_{i,k}}. \quad (1.11)$$

Náročný výpočet funkce  $\arctan(\dots)$  můžeme aproximovat hledáním v dvojrozměrné vyhledávací tabulce. Takto získáme parametrický obraz složený z vektorových hodnot, kde pro každou pozitivně detekovanou hrana v pixelu určíme lokální směr, který může být výhodný u vyššího stupně analýzy obrazů.

Takové detektory jsou založené na opakující se konvoluci obrazu s 8 směrovými maskami. Přiblíží nám směr pomocí vážených sumací. Jako příklad slouží následující masky:

$$h_0 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad h_1 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix},$$

$$h_2 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad h_3 = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}, \text{ atd.} \quad (1.12)$$

Každá maska samozřejmě poskytuje různý parametrický obraz  $\{g_{i,k}\}$ , který zdůrazňuje jinak orientované hrany. Za zmínku stojí, že tyto čtyři masky jsou postačující a zbývající čtyři

pouze vytváří jejich negativy. Hranová detekce a odhad směru pro každý pixel je potom proveden nalezením maxima a následně prahováním:

$$g_{i,k} = (\max(|g_{i,k}|) \geq T), \quad (1.13)$$

kde  $j$  je stupeň natočení a  $T$  představuje zvolený práh. Předchozí čtyři masky reprezentují tzv. Sobelův operátor. Za podmínky, že nahradíme všechny nenulové hodnoty  $\pm 1$ , dostaneme operátor zvaný Prewittův. Srovnatelnou funkci má Kirschův kompasový operátor:

$$h_0 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix}, h_1 = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix}, h_2 = \begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix}, \text{ atd.} \quad (1.14)$$

Výhody kompasových masek spočívají ve vynechání nelineární operace funkce  $\arctan(\dots)$  při odhadu směru. Cíle však dosáhneme konvolucí za použití osmi masek místo dvou.

Při programování potřebujeme binární obraz nebo stupně šedi, které využijeme jako vstup. Vrací se binární obraz stejné velikosti, kde funkce nalezne hrany a označí je hodnotou jedna a nulou zbytek. Nejčastěji se využívá Sobelův operátor. V obrazu se jako hrana projeví místo, kde gradient dosahuje maxima. Na Obr.1.3 vidíme rozdíly mezi realizací jednotlivých detektorů [1] [6].

### 1.2.1 Zero-crossing

Laplaceův operátor počítá druhé diference obrazu, které v místě hran prochází nulou. Průchody nulou potom detekuje příslušný detektor tzv. zero-crossing. Tento postup poskytuje úzké a obvykle více přesné umístění hrany než detektor založený na gradientu, ačkoli výsledek bývá více znehodnocený šumem.

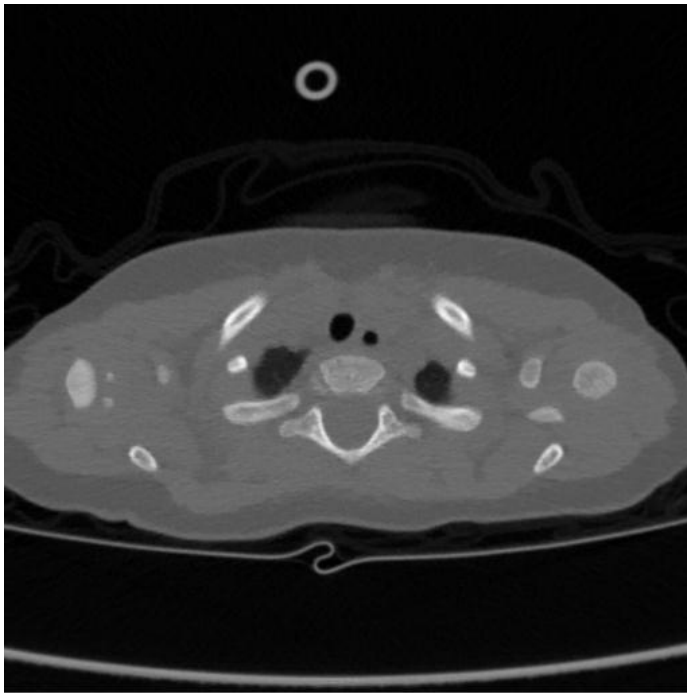
Hledáme umístění okolních pixelů, které mají opačnou intenzitu. Můžeme proto využít následující masku:

$$m = \begin{bmatrix} 0 & 1 & 0 \\ 1 & x & 1 \\ 0 & 1 & 0 \end{bmatrix}. \quad (1.15)$$

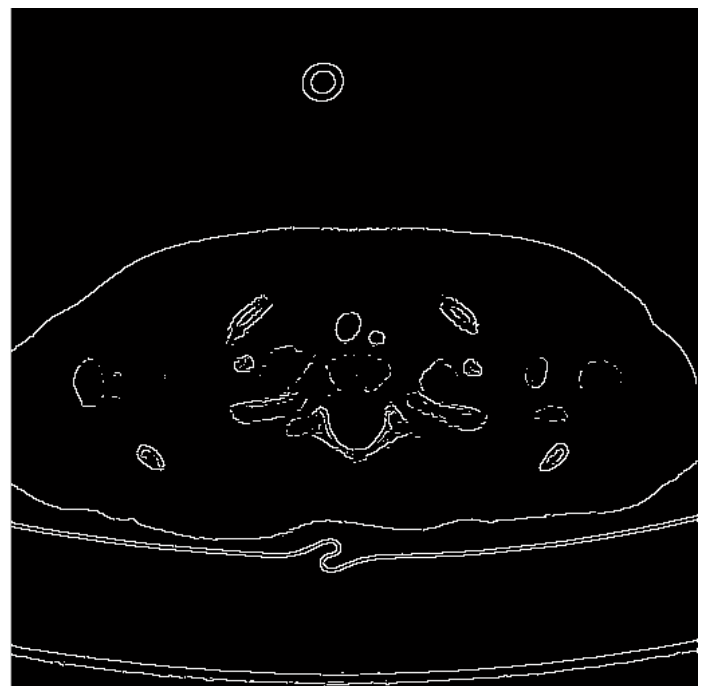
Jako výstup vznikne obraz s tloušťkou hran jednoho pixelu. Přejechy nulou se nevyskytují jen na místech, kde se rychle mění hodnota intenzity. Může se objevit i v začínajícím nárůstu nebo poklesu intenzity. Proto se ne vždy detekují hrany. Takový detektor patří mezi kvalitnější než např. Sobelův nebo Robertsův, počítající první diference [1][6].

### 1.2.2 Operátor Laplaciá gaussianu (LoG)

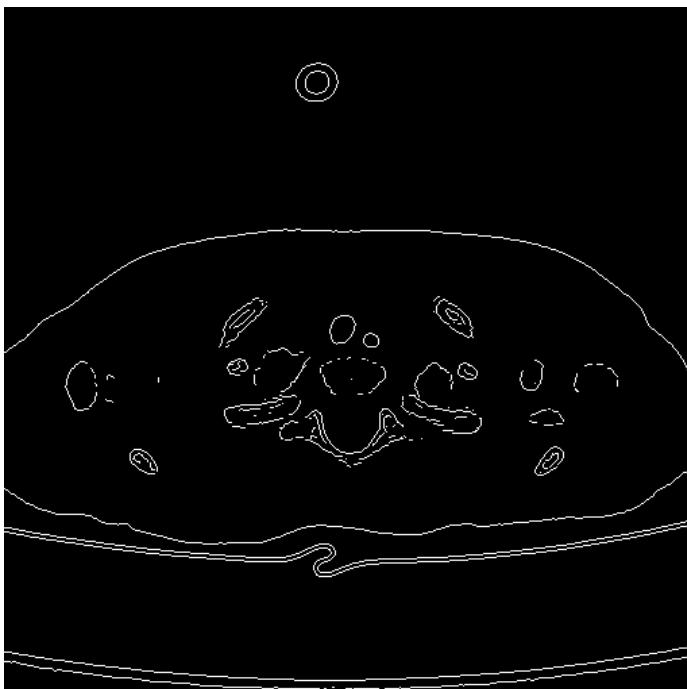
Předchozí operátor trpí vysokou citlivostí k šumu. Proto se dá očekávat, že bude třeba obraz předzpracovat nějakým vyhlazovacím operátorem. LoG spojuje masky Gaussianu (vyhlazovacího filtru) a Laplacianu [1].



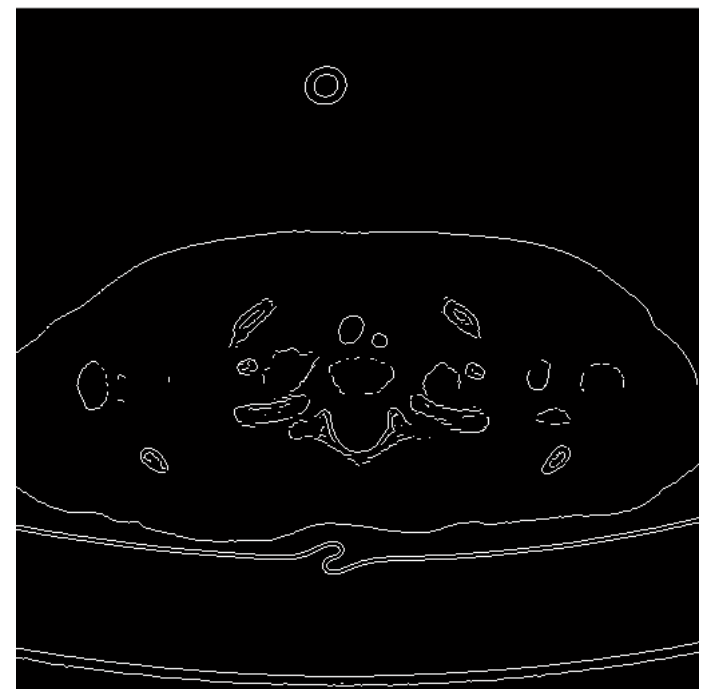
a) originální obraz



b) Robertsův operátor



c) Sobelův operátor



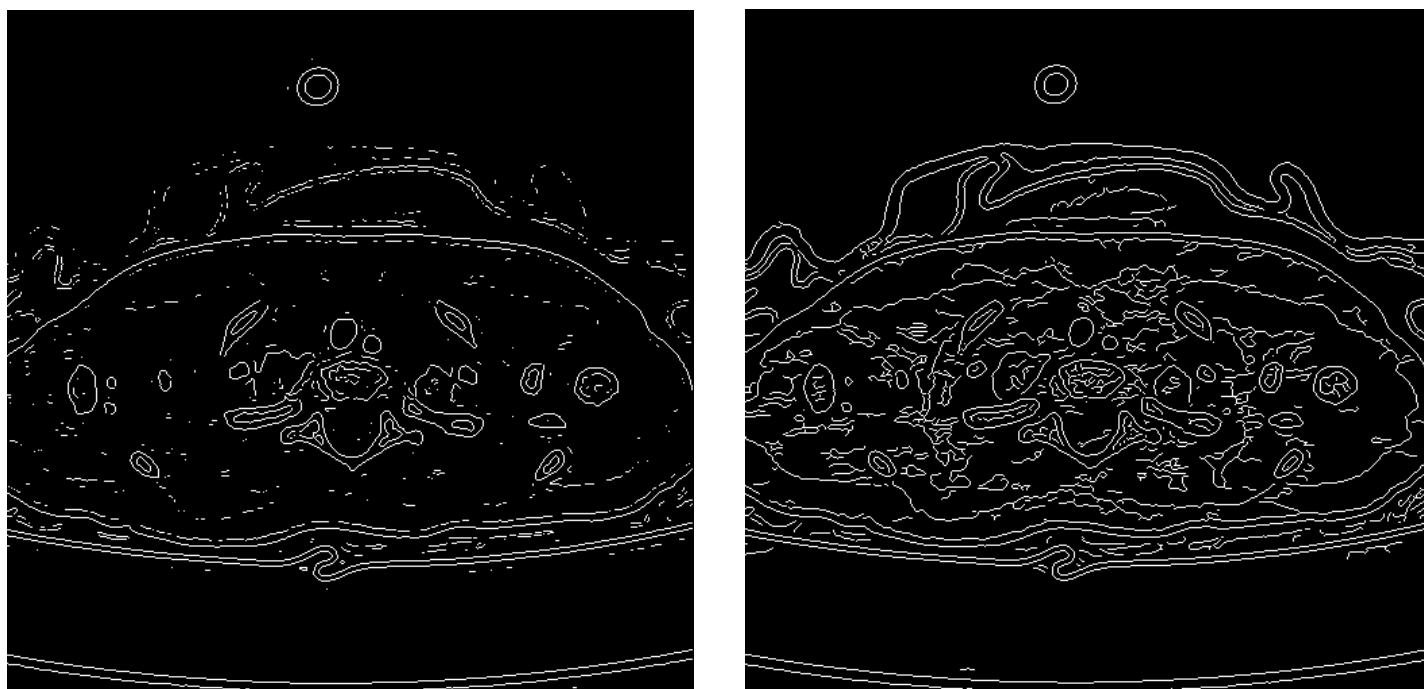
d) Prewittův operátor

**Obr.1.3** Obrazy odvozené z detektorů první diference

### 1.3 Další postupy hranové detekce

Cannyho detektor se často využívá jako spolehlivý prostředek založený na vyhlazování Gaussovým operátorem. Vylepšuje detekci pomocí směrových derivací a využití orientace gradientu. Počítá s druhými diferencemi. Má ještě některá další omezení, která zkvalitní obraz. Také připojuje slabší hrany, pokud jsou spojené s již detekovanou silnější hranou. Lze tak potlačit falešné hrany a vylepšit spojitost. Na Obr.1.4 vidíme, že Cannyho detektor detekuje hrany lépe než Zero-crossing nebo jiné detektory.

Důležitou roli hrají rohové detektory, které umožňují nalézt rysy obrazu obdélníkových nebo podobných předmětů. Těmi se však nebudeme více zabývat, protože potřebujeme detekovat kružnice [1].



a) Zero-crossing

b) Cannyho detektor

**Obr.1.4** Obrazy odvozené z detektorů druhé diference

### 1.4 Šum v obrazu

Diferenční operátory jsou citlivé na šum ve zpracovaném obrazu. Vliv šumu můžeme částečně potlačit sumací více vypočtených gradientů. Mezi takové patří např.:

$$\Delta_x f_{i,k}: \frac{1}{6} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \text{ nebo } \Delta_y f_{i,k}: \frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ atd.}, \quad (1.16)$$

kde druhá matice je příkladem nerovnoměrně váženého průměrování. Laplaceův operátor může být sečten ze dvou o 45° otočených operátorů buď jako maska:

$$\mathcal{L}\{f_{i,k}\}: \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \text{ nebo } \begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (1.17)$$

Ve druhé masce nepatrně převažuje operátor ve směru vertikálním a horizontálním, zatímco v první masce naopak [1][2].

## 1.5 Využití hranové detekce

Hranová detekce má mnoho oblastí využití. Patří mezi ně např. automatická hranová detekce pro Vlnkovou transformaci v Satelitních SAR snímcích. Slouží pro pozorování širokých ploch během dne i noci nezávisle na počasí. Adaptivní hranová detekce se zaměřuje na ultrazvukové snímky. Na těchto snímcích se objevuje tzv. speklový šum nebo-li speckles. Novější, přímá hranová detekce prostorových fázových filtrů, je založená na Hilbertově transformaci. Takové filtry jsou více efektivní a přizpůsobivé. Další metoda představuje myšlenku elektronického obvodu. Zde se mapuje intenzita hran do rezistorů a kalkuluje se rozdělení proudu. Aplikovat takovou techniku můžeme např. pro kódování videa, rozpoznávání snímků, jejich vyhledávání, atd. Pro naši práci budeme potřebovat Houghovu transformaci, které se bude věnovat následující kapitola podrobně.

## 2. Houghova transformace

Běžně řešený problém při zpracování obrazů bývá určit polohu, počet nebo orientaci konkrétního předmětu v obrazu. Za příklad můžeme považovat určení rovné cesty na letecké fotce. Takový případ můžeme vyřešit pomocí Houghovy transformace pro vyhledávání přímek. Častěji však řešíme jiné tvary než přímky jako parabolu, kruh, elipsu nebo další. Obecná Houghova transformace se využívá pro určení různých druhů tvarů, ačkoli její složitost roste s počtem parametrů potřebných k popisu. Metoda je založena na transformaci každého bodu obrazu do přímky v parametrickém prostoru. Ten je definován pomocí zastoupení určitých parametrů využitých k popisu přímek v obrazové rovině. Hough k popisu zvolil dobře známé parametry, které popisuje tzv. směrnicová rovnice přímky. Získal tak dvourozměrnou rovinu. Tyto parametry jsou však neohraničené, což komplikuje aplikování techniky. Proto se v praxi počítá s jinou rovnicí přímky, čímž se tento problém eliminuje [1] [4] [7].

### 2.1 Parametrický prostor přímky

Houghova transformace může být popsána jako transformace bodu z roviny  $x,y$  do parametrického prostoru. Ten je definován podle tvaru objektu, který nás zajímá. Přímku procházející skrz body  $(x_1,y_1)$  a  $(x_2,y_2)$  popisuje rovnice:

$$y = ax + b . \quad (2.1)$$

Uvedená rovnice představuje popis přímky v Kartézském souřadnicovém systému, kde  $a$  reprezentuje směrnici přímky a  $b$  úsek na ose  $y$ . Houghova transformace neuzivá tento tvar, protože přímka kolmá k ose  $x$  má hodnotu  $a$  rovnou nekonečnu. Proto bychom museli mít parametrický prostor nekonečné velikosti. Z tohoto důvodu je přímka popsána přes její normálu.

Soubor všech přímek v obrazu představuje dvouparametrickou rodinu. Libovolná přímka může být reprezentována jediným bodem ve dvourozměrném parametrickém prostoru. V následujícím obrázku přímku přesně určuje úhel  $\theta$ , který svírá její normála s osou  $x$  a velikost úseku normály, který je omezený počátkem souřadného systému a průsečíkem s úsečkou. Rovnice charakterizující tento případ má tvar:

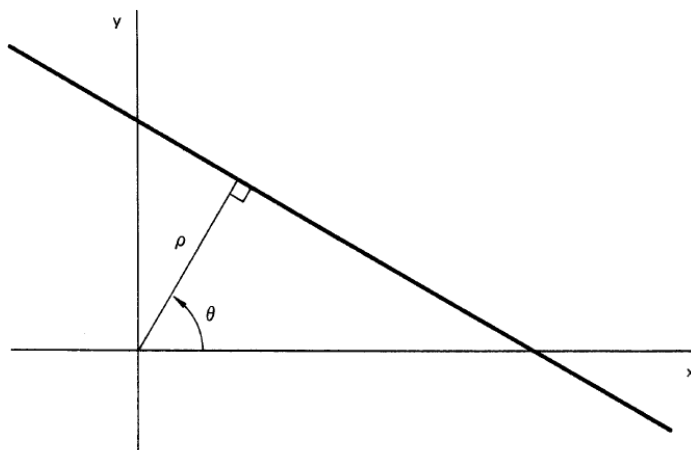
$$x \cos \theta + y \sin \theta = \rho . \quad (2.2)$$

Počet možných úhlů  $\theta$  bude mít konečnou velikost, závisující na užitém rozlišení. Pokud se omezíme pouze na interval  $\langle 0, \pi \rangle$ , potom každá přímky roviny  $x-y$  odpovídá jedinému bodu roviny  $\theta-\rho$ . Obr.2.1 ukazuje tzv. normálovou parametrizaci přímky.

Za předpokladu, že máme řadu  $\{ (x_1, y_1), \dots, (x_n, y_n) \}$   $n$  bodů, chceme najít soubor odpovídajících přímek, a proto musíme body transformovat do křivek v rovině  $\theta$ - $\rho$ . Odpovídající rovnice je potom definovaná jako:

$$\rho = x_i \cos \theta + y_i \sin \theta . \quad (2.3)$$

Body, které leží na stejné přímce, jsou v parametrickém prostoru reprezentovány křivkami, které se protínají v jednom bodě a souřadnice tohoto bodu reprezentují parametry přímky v originálním prostoru. Takový bod v parametrickém prostoru definující přímku se značí  $(\theta_0, \rho_0)$ . Nyní je potřeba vyřešit problém, jak nalézt protínající se křivky. Když si představíme řadu  $\{ (\theta_1, \rho_1), \dots, (\theta_k, \rho_k) \}$  bodů ležících na sinusovce  $\rho = x_0 \cos \theta + y_0 \sin \theta$ , pak tyto body odpovídají velkému množství přímek v obrazové rovině, které prochází body  $(x_0, y_0)$ .



**Obr.2.1** Normálová parametrizace přímky [4]

Pokud zmapujeme všechny body v obrazu do jejich odpovídajících křivek na parametrickém prostoru, dostaneme  $n(n-1)/2$  průsečíků. Takový postup je výpočetně náročný, ale požadovaný výpočet kvadraticky stoupá s počtem bodů v obrazu. Podle základního Houghova postupu přesně určíme přípustnou chybu úhlu  $\theta$  a vzdálenosti  $\rho$ . Následně vyčíslíme rovinu  $\theta$ - $\rho$  do mřížky odpovídající oblasti  $0 \leq \theta \leq \pi$ ,  $-R \leq \rho \leq R$ , kde  $R$  určuje velikost sítě. Tuto oblast považuje za dvojrozměrnou řadu akumulátoru. Pro každý bod  $(x_i, y_i)$  má odpovídající křivka danou vstupní hodnotu v řadě a narůstá procházením každé buňky podél křivky. A tak daná buňka dvojrozměrného akumulátoru nakonec udává celkové číslo všech křivek procházejících skrz ni. Po upravení všech bodů se parametrický prostor prozkoumá a najdou se buňky s největšími hodnotami. Pokud má buňka hodnotu akumulátoru  $k$ , potom přesně k obrazovým bodům leží na přímce, jejíž normálové parametry jsou  $(\theta_i, \rho_i)$  [4].

## 2.2 Houghova transformace pro detekci kružnic

Je zřejmé, že obecná Houghova transformace může být rozšířená na jiné objekty než přímka. Například chceme detekovat kružnicové rozmístění obrazových bodů, zvolíme si parametrickou reprezentaci pro všechny možné kružnice a transformujeme každý bod již zmíněnou cestou. Kružnice se dá popsat následující rovnicí:

$$(x - a)^2 + (y - b)^2 = r^2 . \quad (2.4)$$

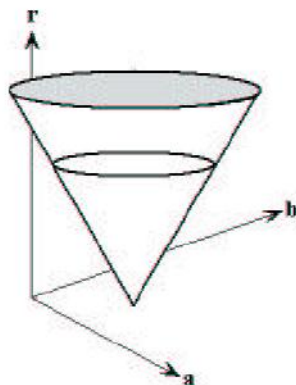
Potom náhodný bod  $(x_i, y_i)$  bude transformován do plochy v parametrickém prostoru, který popisují parametry  $a$ - $b$  a je definovaný jako:

$$(x_i - a)^2 + (y_i - b)^2 = r^2 . \quad (2.5)$$

Jak vidíme, kruh má tři parametry:  $r$ ,  $a$ ,  $b$ .  $A$  a  $b$  představují souřadnice středu kružnice ve směrech os  $x$  a  $y$  a  $r$  určuje souřadnice středu kružnice. Parametrickou reprezentaci tvoří následující tvary rovnic:

$$\begin{aligned} x &= a + r \cos \theta \\ y &= b + r \sin \theta \end{aligned} \quad (2.6)$$

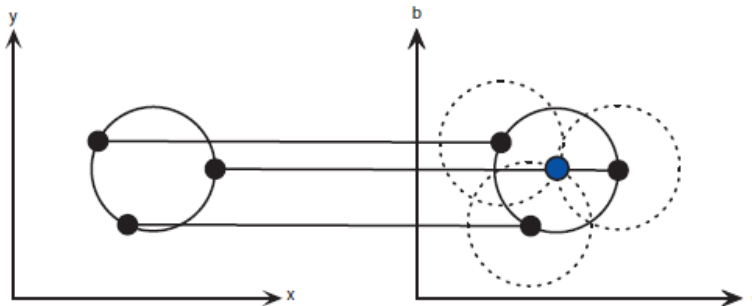
V tomto případě transformujeme každý bod do kužele trojrozměrného parametrického prostoru, viz Obr.2.2. Počet parametrů potřebných k popisu tvaru určuje rozměr parametrického prostoru. Jeden kužel odpovídá jednomu bodu v originálním prostoru. Takový bod určuje parametr  $a_0$ ,  $b_0$ ,  $r_0$ . Postup lze efektivně zrealizovat využitím trojrozměrné řady akumulátoru reprezentované trojrozměrným parametrickým prostorem. Proto, abychom zjednodušili parametrickou reprezentaci kružnice, poloměr můžeme považovat za konstantu nebo libovolný interval známých poloměrů [1] [9] [10].



**Obr.2.2** Parametrický prostor HT pro kružnici [1]

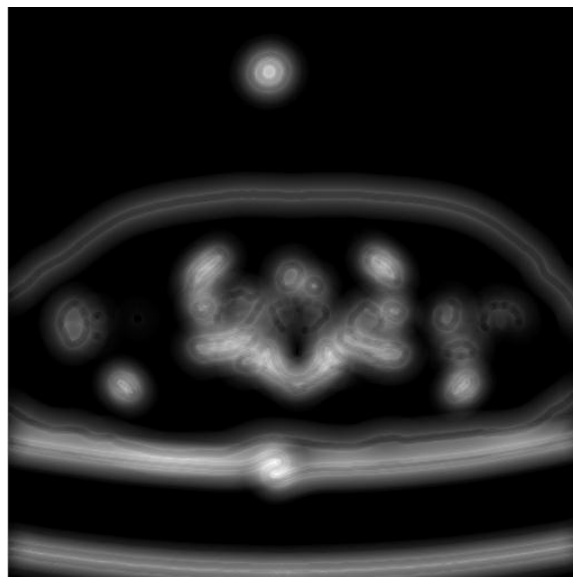
## 2.3 Akumulátor

Nyní si ukážeme postup pro nalezení kruhu v obrazu s využitím Houghovy transformace. Nejdříve nalezneme hrany objektů v obrazu metodou hranové detekce pomocí Cannyho, Sobelova nebo jiného operátoru.



**Obr.2.3** Transformace z obrazové roviny do parametrického prostoru [1]

V každém bodě detekované hrany nakreslíme kružnici se středem v bodě se zadaným poloměrem. Taková kružnice je vykreslena v parametrickém prostoru tak, že souřadnice hranového bodu představují parametry  $a$ ,  $b$  a  $r$ . V souřadnicích, které patří k obvodu nakreslené kružnice, zvýšíme hodnotu matice akumulátoru. Akumulátor má v podstatě stejnou velikost jako parametrický prostor. Tímto způsobem vezmeme každý okrajový bod nakreslené kružnice ve vstupním obrazu s požadovanými poloměry a zvýší se hodnoty akumulátoru. Nyní bude obsahovat čísla odpovídající počtu kružnic procházející skrz jednotlivé souřadnice. Tak se vybere největší hodnota představující střed kružnice obrazu. Na Obr.2.3 je znázorněna transformace z obrazové roviny do parametrického prostoru [1] [4] [5] [10].



**Obr.2.4** HT pro kružnici z řezu CT dat z Obr. 1

## 2.5 Provedení

S algoritmem nastíněným v předchozí části můžeme zrealizovat pracovní verzi. Musíme však ještě zvážit několik aspektů.

### 2.5.1 Jak skladovat data

Prostor akumulátoru je nekonstantní. Jeho velikost závisí na počtu různých poloměrů a zvláště velikosti obrazu. Výpočetní náročnost všech kružnic pro každý okrajový bod roste s počtem bodů v původním obrazu, které jsou obvykle funkcí velikosti obrazu. Celkový čas potřebný pro výpočet může být veliký, když použijeme obrovský obraz s mnoha hranovými body.

### 2.5.2 Jak nakreslit kružnici v diskretním prostoru

Kružnici můžeme v diskretním prostoru nakreslit přímo z rovnic  $x = a + r \cos \theta$ ;  $y = b + r \sin \theta$ , ale nastává zde jeden problém. Jak zvolit diskretní hodnoty nebo rozlišení? Jedno řešení představuje využít vysokého rozlišení úhlu  $\theta$ . Takový případ však pravděpodobně povede k obrovskému přečerpání, kdy se okrajový pixel vykreslí více než jedenkrát. Následně se může vytvořit i nedostatek pixelů při vysokém poloměru. Vzorkováním úhlu při vykreslování kružnice bychom se měli zabývat až po hodnotách vynásobených poloměrem.

Potřebujeme zajistit, aby se zakreslily všechny okrajové pixely. Zvolit vhodné rozlišení proto nepředstavuje lehký úkol a také náročnost výpočtu se zvyšuje. Takové nároky se dají obejít, když předem spočítáme hodnoty pro sinus, cosinus a využijeme vyhledávací tabulku [1] [5].

## 2.6 Další rozšíření Houghovy transformace

V principu se dá transformační metoda rozšířit na libovolné křivky. Pouze si potřebujeme zvolit praktickou parametrizaci pro rodinu křivek, které nás zajímají a potom postupovat podobnou cestou. Upřednostňují se parametry s omezenou množinou, ačkoli se tak za každou cenu postupovat nemusí. Nejdůležitější je mít malý počet parametrů, protože výpočetní náročnost roste exponenciálně s počtem parametrů [1].

# 3. Realizace hranové detekce a Houghovy transformace

V následujících podkapitolách bude popsána realizace hranové detekce a Houghovy transformace v programovém prostředí Matlab.

## 3.1 Načtení CT snímků

```
1. N=357;
2. format='DICOM';
3. X= zeros(512,512,357);
4. for i=1:N
5.     X(:,:,i)=dicomread(['Data_projekt01\' 'I' num2str(i) 'MR0']);
6. end
```

První řádek představuje počet CT snímků, které máme k dispozici. Data jsou ve formátu DICOM (2. řádek), takže se dají načítat jen pomocí funkce *dicomread* (5. řádek). Nejdříve se předpřipraví 3D matice (3. řádek) pro následné ukládání dat (5. řádek). Funkce *dicomread* dokáže načíst pouze jeden dvourozměrný řez, a proto se použije *for* cyklus k postupnému načítání celého objemu dat (4. – 6. řádek). Při vytváření cesty k souboru s uloženými CT snímky se musí brát v úvahu, že proměnná *i* představuje číslo, a tak ji je nutné převést na řetězec pomocí funkce *num2str* (5. řádek). Následně již lze realizovat hranovou detekci.

## 3.2 Hranová detekce

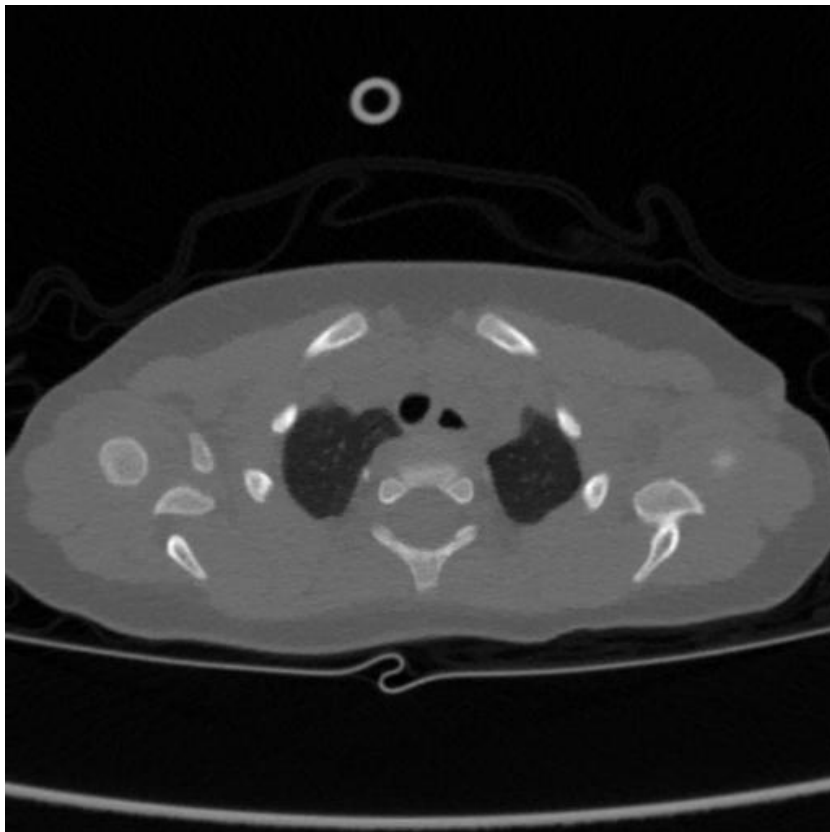
```
7. figure
8. imshow(X(:,:,135), []);
9.
10. X(:,:,135);
11. maska_y = [1  2  1
12.           0  0  0
13.          -1 -2 -1];
14. ct_y = conv2(X(:,:,135),maska_y, 'same');
15. maska_x = [-1 0 1
16.           -2 0 2
17.           -1 0 1];
18. ct_x = conv2(X(:,:,135),maska_x, 'same');
19. gradient = sqrt(ct_x.^2 + ct_y.^2);
20.
21. [M,N]= size(X(:,:,135));
22. prah_snimek= zeros(M,N);
23. prahovani = 2500+-50;
24.
```

```

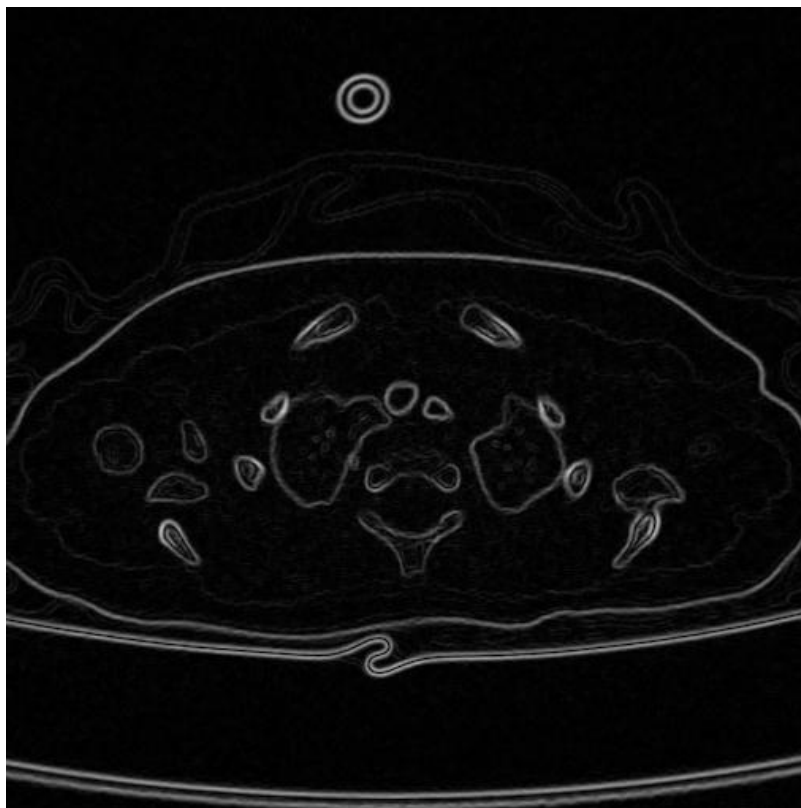
25. prah_snimek(gradient>=prahovani)=1;
26. prah_snimek(gradient<prahovani)=0;
27.
28. figure(2)
29. imshow(gradient, []);
30.
31. figure(3)
32. imshow(prah_snimek, []);

```

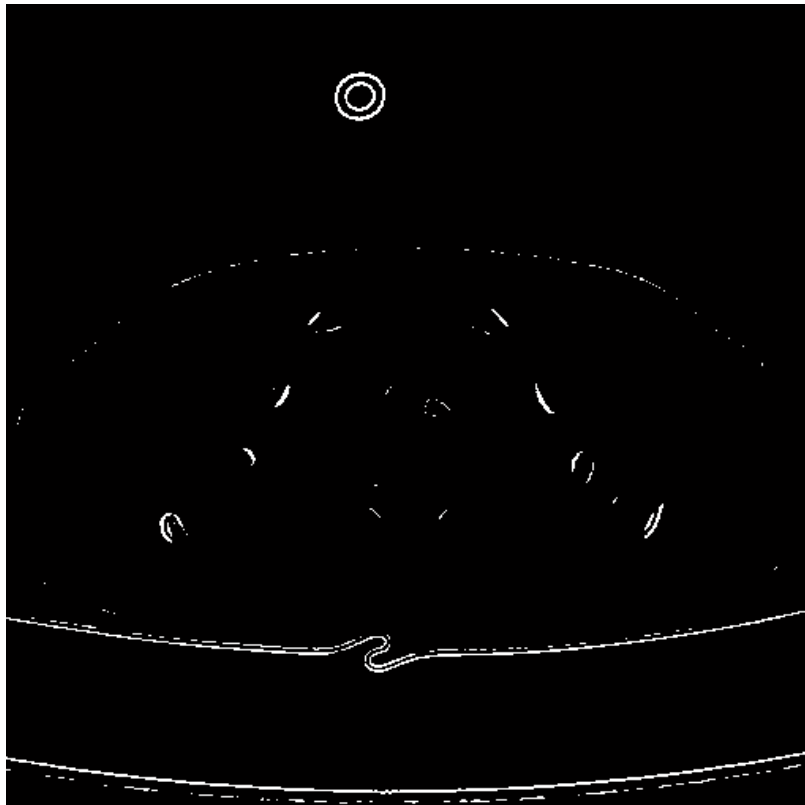
Obraz se nejdříve musí zpracovat vhodně zvoleným hranovým operátorem. Pro detekci anesteziologické trubice postačí Sobelův operátor, protože její hrany jsou dostatečně výrazné a není tak potřeba vykreslit detailně všechny hrany v obrazu. Pro zpracování hran v obrazu stačí použít pouze 2 masky z celkových 8. *Maska\_x* představuje masku pro hledání hran ve vertikálním směru a *maska\_y* pro směr horizontální. (11. -13. řádek a 15. -17. řádek). Následně se s těmito maskami provede konvoluce, která se dá v Matlabu uskutečnit příkazem *conv2* (14. a 18. řádek). Takovým maskám se také říká konvoluční masky nebo jádra. Pro masku o zvolené velikosti probíhá konvoluce následovně. Každý pixel překrytý maskou se vynásobí příslušným koeficientem v obrazu. Výsledné hodnoty se potom sečtou a konečné číslo uloží do aktuálního pixelu. Postupně se tak zpracují jednotlivé pixely v obrazu. Vzniknou tak dva výstupní obrazy s detekovanými hranami v určitém směru, z kterých je třeba vytvořit gradientní obraz podle vzorce 1.5, viz.obr.3.1a. Tak se zjistí, jak rychle se mění intenzita v okolí určitého pixelu a lze si vykreslit snímek s nalezenými hranami. Gradientní obraz tvoří maska s různými hodnotami intenzit, ale pro Houghovu transformaci je nutné získat obraz binární. Pro CT snímky se intenzita vyjadřuje v Hounsfieldových jednotkách nebo-li stupních šedi. Aby se získal binární obraz, musí se nejprve nastavit vhodný práh, podle kterého lze zvýraznit důležité hrany a potlačit méně důležité. Z hodnot pod zvolený práh vzniknou nulové intenzity a z ostatních číslo 1. Prah lze zjistit tak, že se nalezne rozdíl mezi intenzitami trubice a jejím pozadím s tolerancí +- 50. Tímto způsobem byl vhodný práh nastaven jako 2500+-50 viz Obr.3.1b



a) originální obraz



b) hranová detekce pomocí Sobelova operátoru- gradientní obraz



c) prahování – zvolený práh  $2\ 500 \pm 50$   
(vzniklý obraz je binární)

**Obr.3.1** Hranová detekce

### 3.3 Houghova transformace

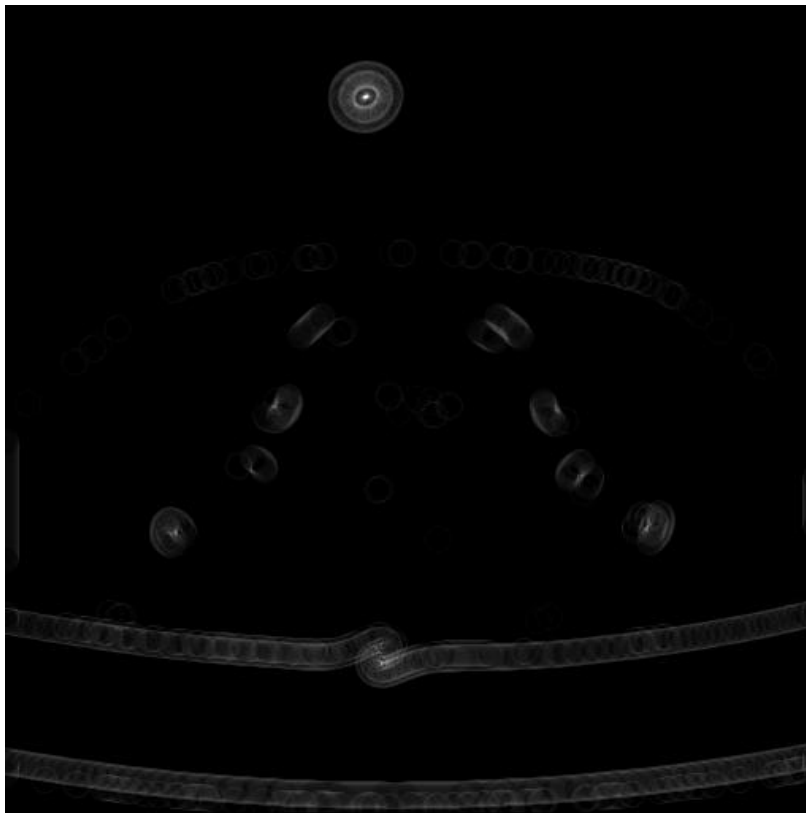
```

38. [M,N,K]= size(X);
39. polomer=4:20;
40. akumulator=zeros(M,N);
41. houghuv_prostor=zeros(M,N,polomer);
42. [x y] = find(XBW);
43.
44. for r=polomer
45.     for i=1:length(x)
46.         for k=0:2*pi/500:2*pi
47.             a=round(x(i)-r*cos(k));
48.             b=round(y(i)-r*sin(k));
49.             if (a>0) && (a<M) && (b>0) && (b<N)
50.                 akumulator(a,b)=akumulator(a,b)+1;
51.             end
52.         end
53.     end
54.     houghuv_prostor(:,:,r)=akumulator(:,:,r);
55.     akumulator=zeros(M,N);
56.     figure(4)
57.     imshow(houghuv_prostor(:,:,r),[]);
58. end

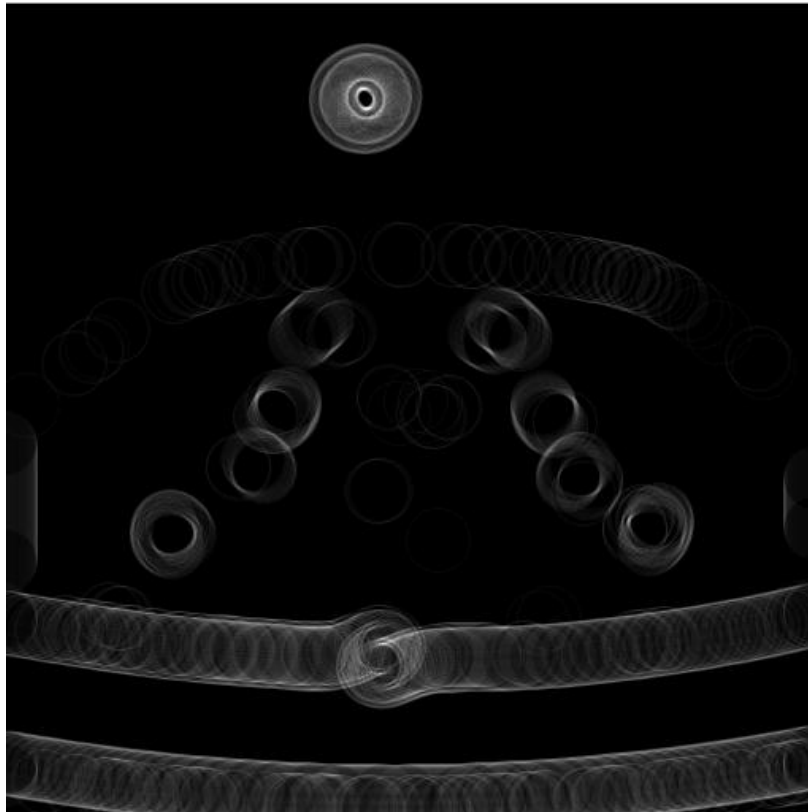
```

Obraz předzpracovaný hranovým operátorem se dá nyní využít k realizaci algoritmu Houghovy transformace. Nejdříve si připravíme proměnné, které budeme potřebovat pro HT. Do proměnných  $M$ ,  $N$ ,  $K$  se musí uložit velikost 3D matice (38. řádek). Dále si zvolíme interval poloměrů od 4 do 20 pixelů (39. řádek). Vytvoříme matici *akumulator*, která představuje rovinu pro výpočet konkrétního poloměru (40. řádek) a trojrozměrný *houghuv\_prostor*(41. řádek). Funkce *find* vrací indexy všech nenulových prvků pro jednotlivé souřadnice  $x$  a  $y$  obrazu upraveného hranovou detekcí. Dají se tak zjistit středy všech kružnic, které se vykreslují do Houghova prostoru (42. řádek). Nemusíme potom procházet celou matici, ale pouze vektor  $x$  (45. řádek). Samotný algoritmus probíhá tak, že se postupně prochází jednotlivé poloměry a počítají a následně zaokrouhlí indexy  $a$  a  $b$  podle vzorců z rovnice (2.6). Pokud vypočtené indexy splňují podmínku nacházet se uvnitř obrazu, zvýší se hodnota v akumulátoru na příslušných pozicích. Akumulátor se postupně ukládá do Houghova prostoru pro konkrétní poloměr (44.-54. řádek).

Po projítí Houghovy transformace jedním konkrétním poloměrem musí být akumulátor vymazán a nachystán tak pro následující poloměr (55. řádek). Funkcí *imshow* lze zobrazit libovolný obraz, v našem případě originální a upravený pomocí hranové detekce a Houghovy transformace (10., 13. a 33. řádek ve stejném pořadí). Můžeme je vidět i na Obr.3.2.



a) akumulátor pro poloměr  $r=8$  pixelů



b) akumulátor pro poloměr  $r=20$  pixelů

**Obr.3.2** Houghova transformace

## 4. Normalizace Houghova prostoru

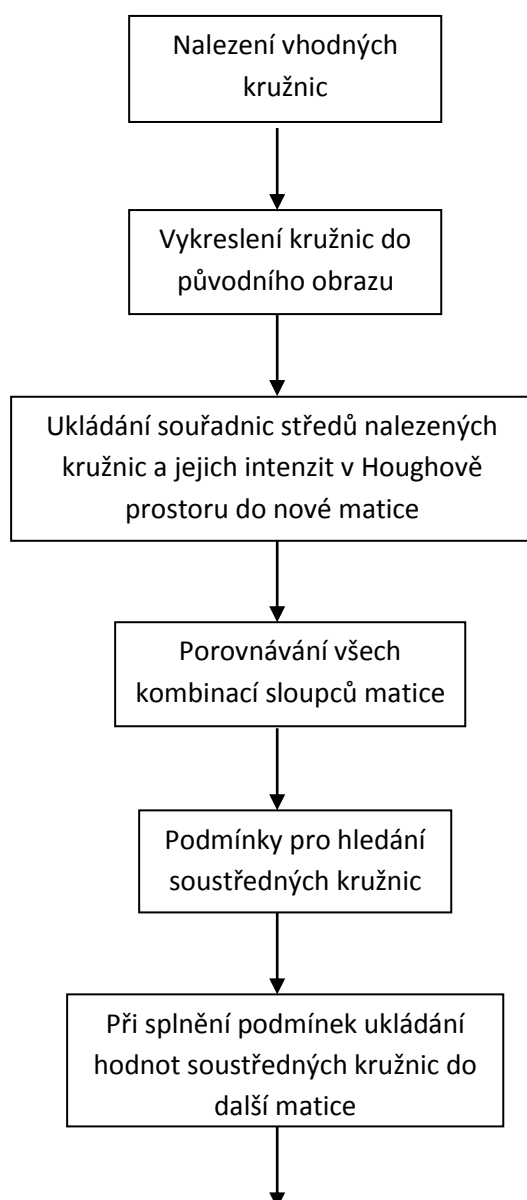
1.  $B = \max(\max(\text{houghuv\_prostor}))$ ;
2.  $\text{normalizace} = \text{houghuv\_prostor} / \max(B)$ ;

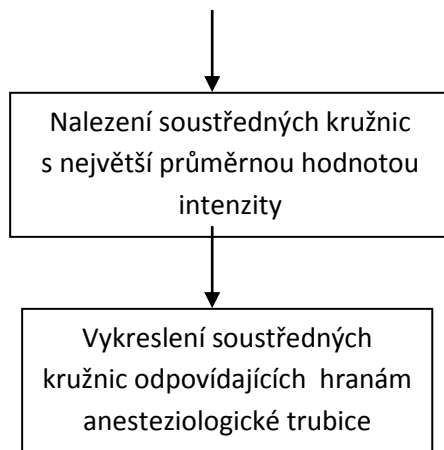
Pro další zpracování obrazu je třeba Houghův prostor normalizovat tzn. získat obrazovou matici v rozmezí intenzit  $\langle 0,1 \rangle$ . Nejdříve se musí nalézt nejvyšší hodnota intenzity a pomocí ní podělit všechny hodnoty intenzit nacházející se v Houghově prostoru.

# 5. Vyhodnocení Houghova prostoru

## 5.1 Blokové schéma vyhodnocení Houghova prostoru

Následující blokové schéma znázorňuje, jak se postupuje při vyhodnocení Houghova prostoru. Nejdříve se naleznou vhodné kružnice, které se vykreslí do původního CT snímku. Dále se uloží souřadnice středů nalezených kružnic a jejich intenzit v Houghově prostoru do nové matice. U této matice se porovnávají postupně všechny kombinace sloupců a pokud jsou splněny podmínky pro hledání soustředných kružnic, uloží se do nové matice. Z nalezených soustředných kružnic se vybere jenom jedna dvojice, která nejlépe odpovídá hranám anesteziologické trubice. V závěru se tyto kružnice vykreslí opět do původního CT snímku a snímku hranové detekce.



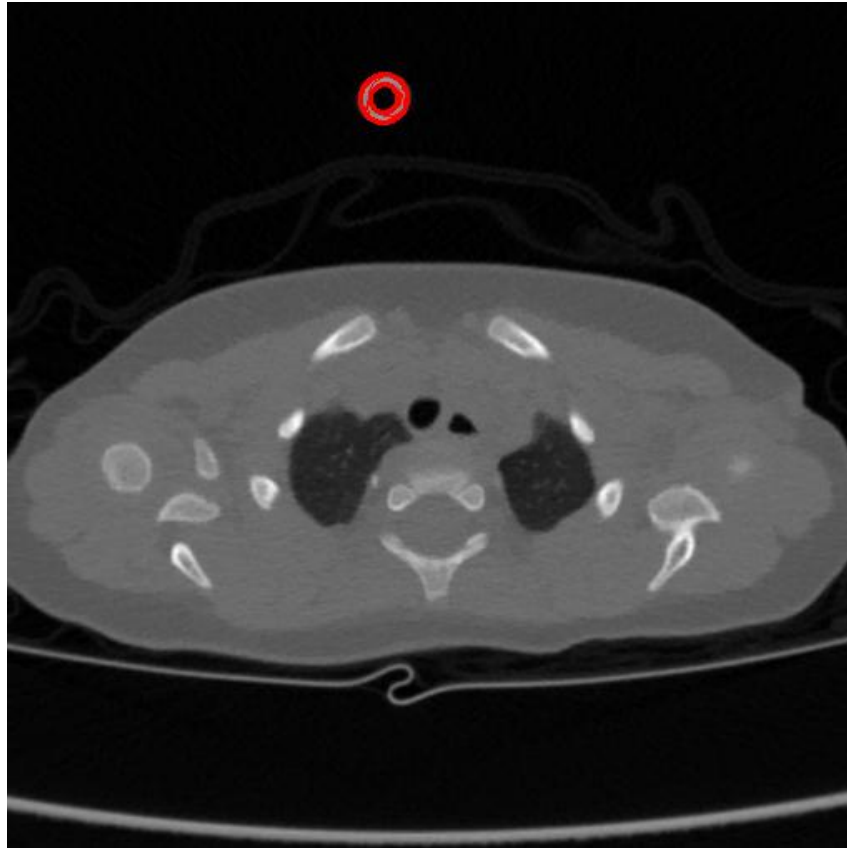


## 5.2 Nalezení vhodných kružnic a vykreslení do původního obrazu

```

1. [D E F]= ind2sub(size(normalizace),find(normalizace>0.72));
2. figure(5)
3. kruznicel=zeros(M,N);
4. imshow(X(:,:,135),[]);
5. hold on
6. k = 0:2*pi/500:2*pi;
7. for i= 1:length(D)
8. a=round(D(i)-F(i)*cos(k));
9. b=round(E(i)-F(i)*sin(k));
10. hold on
11. plot(b,a,'r');
12. hold off
13. end
14. hold off
  
```

Funkce `find` slouží k hledání hodnot nad nebo pod určitým prahem v konkrétním obrazu. Pomocí funkce `ind2sub` se těmito hodnoty z normalizovaného Houghova prostoru přiřadí příslušné souřadnice a uloží do matice. V 1. řádku lze vidět, že pro normalizovaný Houghův prostor se hledají body o intenzitě vyšší než 0,72. Souřadnice  $x$ ,  $y$  a  $z$  se následně ukládají do proměnných  $D$ ,  $E$ ,  $F$ . Osy  $x, y$  představují souřadnice středů nalezených kružnic a osa  $z$  velikosti jejich poloměrů, které lze nyní vykreslit do původního obrazu. Vytváření kružnic se uskutečňuje ve *for* cyklu pomocí parametrických rovnic s nalezenými souřadnicemi  $D(i)$ ,  $E(i)$  a  $F(i)$  (7. -12. řádek). Nejdříve se však zobrazí originální snímek (4. řádek) a pomocí funkce `hold on` (5. A 10. řádek) se do něj vykreslí načtené kružnice, které se graficky zobrazí funkcí `plot` (11. řádek). Výhodou grafického zobrazení je, že se dá libovolně měnit barva. Pro parametr 'r' se přiřadí červená barva. Vykreslování do stejného obrazu se ukončí příkazem `hold off`. (12. a 14. řádek). Na Obr.5.1 je vidět vykreslení těchto kružnic ze snímku 135.



**Obr.5.1** Vykreslení vhodných kružnic

### 5.3 Hledání soustředných kružnic

1. `for l=1:length(D)`
2. `maximum(4,l)=normalizace(D(l),E(l),F(l));`
3. `end`

Aby se lépe hledaly soustředné kružnice, je třeba nejdříve vytvořit matici, která bude mít 4 řádky a počet sloupců odpovídající počtu nalezených kružnic. Do 1.-3. řádku se postupně pomocí cyklu *for* ukládají souřadnice *x*, *y*, *z* pro středy nalezených kružnic. Do 4. řádku se potom uloží intenzita tohoto středu v Houghově prostoru jako velikost čítače v příslušném bodě.

```

4. p=zeros(4,2);
5. r=0;
6. for l=1:length(maximum(1,:))
7.     for m=l+1:length(maximum(1,:))
8.         if abs(maximum(1,l)-maximum(1,m))<2 && abs(maximum(2,l)-
9.             maximum(2,m))<2 && abs(maximum(3,l)-maximum(3,m))>5 &&
10.            abs(maximum(4,l)-maximum(4,m))<8
11.            p(:,r+1)=maximum(:,l);
12.            p(:,r+2)=maximum(:,m);
13.            r=r+2;
14.        end
15.    end
16. end

```

Při hledání soustředných kružnic odpovídajících hranám anesteziologické trubice je třeba stanovit vhodné podmínky. Pro soustředné kružnice platí, že mají společný střed. Navíc je známá přibližná vzdálenost těchto kružnic od sebe navzájem. Ve výše uvedeném algoritmu je nejdříve předpřipravená matice, do které se budou později ukládat nalezené kružnice. Dále se porovnávají všechny kombinace dvojic sloupců matice označené jako *maximum*, viz. začátek této kapitoly. Vytvořené podmínky (8. – 10. řádek) odpovídají faktům, že souřadnice  $x$ ,  $y$  středů kružnic musí mít stejnou hodnotu. Pro správnost vyhodnocení se toleruje  $\pm 1$  pixel. Souřadnice  $z$ , tedy poloměry kružnic, jsou od sebe ve vzdálenosti přibližně 5-7 pixelů. Pokud jednotlivé dvojice porovnávaných kružnic splňují tyto podmínky, uloží se jejich hodnoty  $z$  matice *maximum* do nové matice  $p$  (11.-12. řádek). Ve výsledku jsou v této matici vedle sebe uloženy všechny možné dvojice soustředných kružnic, ze kterých je třeba vybrat nejvíce odpovídající hranám anesteziologické trubice.

```

17. p1=zeros(1,r/2);
18. s=1;
19. for l=1:r/2
20.     p1(l)=(p(4,s)+p(4,s+1))/2;
21.     s=s+2;
22. end
23. x1=max(p1);

```

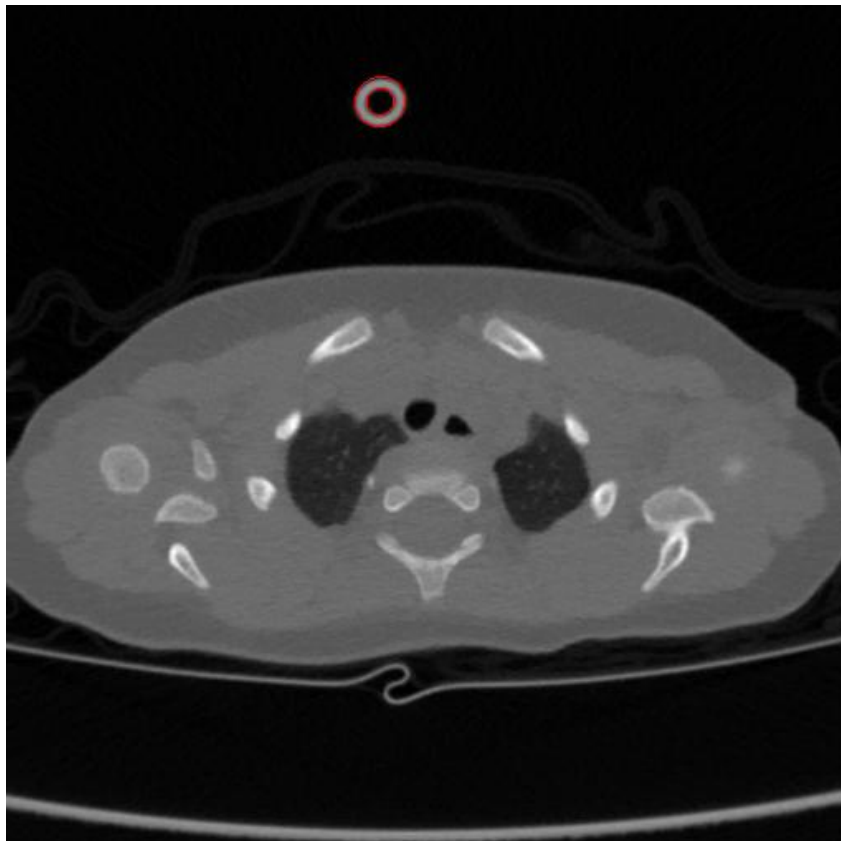
V následujícím algoritmu se jako nejvhodnější soustředné kružnice vyberou ty, které mají v průměru největší hodnotu intenzity. Jako první krok se předpřipraví matice  $p1$  a parametr  $s$ . Ve for cyklu se průměrují hodnoty intenzit nalezených soustředných kružnic a ty se dále ukládají do matice  $p1$ . Z této matice je třeba vybrat maximum (23. řádek).

```

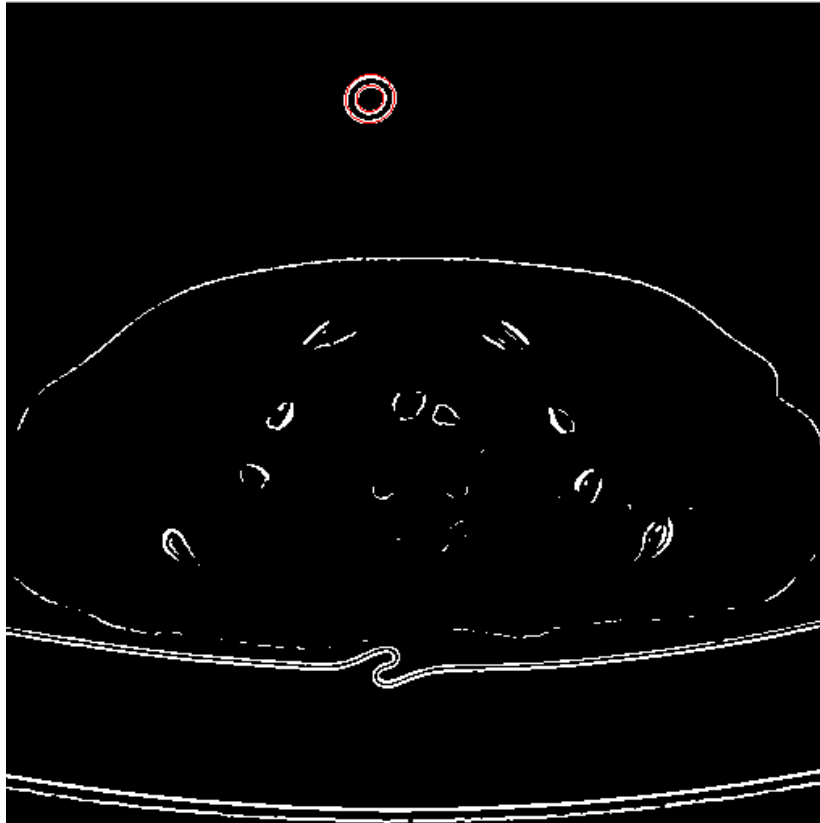
24. soustredne_kruznice=zeros(4,2);
25. for l=1:length(p1)
26.     if x1==p1(l)
27.         soustredne_kruznice(:,1)=p(:,l*2);
28.         soustredne_kruznice(:,2)=p(:,(l*2)-1);
29.     end
30. end

```

Dále se musí nalézt pravé hodnoty intenzit pro nalezenou nejvyšší zprůměrovanou hodnotu a odpovídající souřadnice  $x, y, z$ . Postupně se prochází se matice  $p1$  a  $z$  místa, kde její některá hodnota odpovídá  $x1$ , lze jednoduše dohledat potřebné informace. Stačí jen tuto pozici sloupce  $l$  vynásobit 2 a získají se hodnoty pro 1. kružnici. Pokud se od této pozice navíc odečte číslo 1, nalezne se 2. kružnice. Hodnoty intenzit a souřadnic těchto 2 kružnic se uloží do matice pojmenované jako *soustredne\_kruznice*. Tato matice obsahuje informace pro nejvhodnější soustředné kružnice odpovídající hranám anesteziologické trubice. Na Obr.5.2 jsou vykreslené tyto nalezené kružnice.



a) Soustředné kružnice v originálním obrazu



b) Soustředné kružnice v obrazu hranové detekce

**Obr.5.2** Nalezené soustředné kružnice

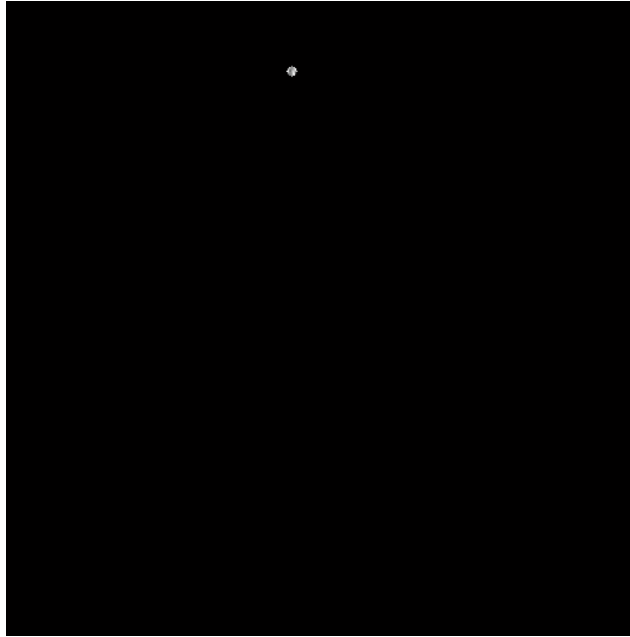
## 6. Odstranění anesteziologické trubice

Pokud by se anesteziologická trubice odstranila způsobem, že by se za ni dosadily nuly. Vznikla by oblast nul, avšak okolní vzduch má díky šumu v CT datech nenulové hodnoty. V takovém případě by byl zřetelný zbytek po odstranění trubice. Pro odstranění anesteziologické trubice se proto nejdříve musí získat vzorek pozadí, kterým ji lze překrýt. V následujících podkapitolách je podrobně popsán postup, jak anesteziologickou trubicí odstranit.

### 6.1 Výběr vzorku pozadí

```
1. AxeX=1:512;
2. AxeY=AxeX';
3. CenterY=soustredne_kruznice(1,1); CenterX=soustredne_kruznice(2,1);
4. f=@(xx,yy) (xx-CenterX).^2+(yy-CenterY).^2 <=4^2;
5. BinMask=bsxfun(f,AxeX,AxeY);
6. figure
7. Vzorek = BinMask.*X(:, :, i);
8. figure(9)
9. imshow(Vzorek, []);
```

Pro výběr vzorku pozadí je nutné vytvořit binární masku, s kterou se následně vynásobí originální obraz a zůstanou tak zachovaná taková místa obrazu, kde měla maska hodnotu intenzity 1. Vzhledem k tomu, že je potřeba získat vzorek s vnitřní částí anesteziologické trubice, vytvoří se binární maska *BinMask* o velikosti originálního CT snímku s intenzitou 1 právě v místě vnitřku trubice (1.-5. řádek). Takovou masku lze vytvořit pomocí funkce *bsxfun*, kde se jako první vstup použije funkce *f* a další vstupy osa *x* a *y*. Funkce *f* představuje parametrickou rovnici kruhu. Vynásobením masky s originálním obrazem se získá vzorek, který představuje vnitřní část anesteziologické trubice viz Obr.6. 1.



**Obr.6. 1** Vybraný vzorek pozadí  
z originálního obrazu

## 6.2 Vyříznutí vzorku pozadí a jeho zvětšení

```
1.  vyriznuti=regionprops (BinMask, 'ConvexHull');
2.  convexies=cat (1,vyriznuti.ConvexHull);
3.  x1 = min(convexies(:,1));
4.  x2 = max(convexies(:,1));
5.
6.  y1 = min(convexies(:,2));
7.  y2 = max(convexies(:,2));

8.  vyrez = BinMask(y1:y2,x1:x2);
9.  figure(10),
10. imshow(vyrez)
11. figure(11)
12. obraz=X(:, :, i);
13. vzorek1=obraz (y1:y2,x1:x2);
14. nahrazeni=vyrez.*vzorek1;
15. imshow(nahrazeni, []);
16. J=imresize (nahrazeni, 5);
17. figure(12)
18. imshow(J, []);
```



**Obr.6. 2** Vyříznutý a zvětšený  
vzorek pozadí

V následující části algoritmu se využívají dvě důležité funkce: funkce *regionprops* a *imresize*. Funkce *regionprops* měří vlastnosti binárních objektů v obraze. Její obecný zápis je `STATS = REGIONPROPS(BW,PROPERTIES)`. Vlastnosti- `PROPERTIES`- se změří pro každý souvislý objekt v binárním obraze `BW`. Vlastnost, která se měří, se musí zadat do funkce. `STATS` je strukturní pole s délkou rovnou počtu oblastí v obraze. Toto pole značí různé vlastnosti pro každou určenou oblast. `PROPERTIES` se oddělují uvozovkami. Mezi oblasti, které lze měřit, patří např. 'Centroid', 'BoundingBox', 'ConvexArea', 'ConvexHull', a další. Pomocí příkazu *cat* se zjistí souřadnice zvolené oblasti funkce *regionprops*. Příkaz *imresize* mění velikost obrazu. Obecný zápis vypadá následovně: `B = IMRESIZE(A, SCALE)`. Znamená, že se vrátí obraz, který je násobkem obrazu `A`. Obraz může být RGB, binární i ve stupních šedi. Násobek se vkládá jako vstupní parametr `SCALE`. Nelze měnit velikosti částí obrazu, ale jen obrazu jako celku.

Aby bylo možné vyříznout vzorek pozadí, musí se nejprve v binární masce *BinMask* funkcí *regionprops* změřit vlastnosti kruhu. Na základě těchto hodnot lze získat souřadnice čtverce, do kterého je vepsán kruh binární masky (1.-2. řádek). Z nich se dále vyberou jen souřadnice rohů tohoto čtverce, tzn. maximální a minimální souřadnice na ose `x` a `y` (3.-7. řádek). V intervalu nalezených souřadnic `<x1:x2>` a `<y1:y2>` se z masky vyřízne čtverec vyřez obsahující jen její kruh (9. řádek). Ve stejném intervalu je potřeba získat čtverec z originálního CT snímku *vzorek1*, který se následně vynásobí se maticí *vyrez*. Vznikne tak čtverec s vnitřní částí anesteziologické trubice s černým okolím. Takto odebraný vzorek je už jen potřeba zvětšit na velikost anesteziologické trubice, aby se s ním mohla nahradit. Výsledný vzorek sloužící k nahrazení trubice je vidět na Obr.6. 2.

### 6.3 Vytvoření nulové intenzity pro trubici a její vnitřek

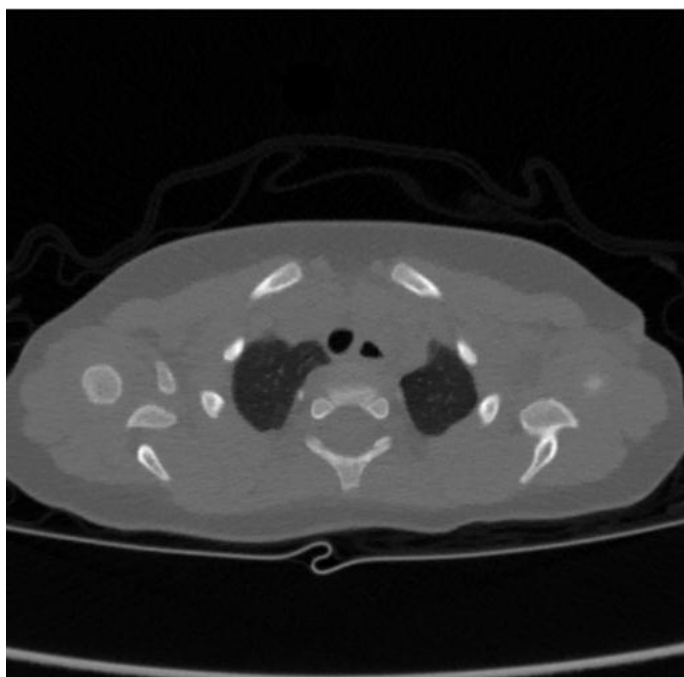
```

1. AxeX=1:512;
2. AxeY=AxeX';
3. CenterY=q(1,1); CenterX=q(2,1);
4. f=@(xx,yy) (xx-CenterX).^2+(yy-CenterY).^2 <= 20^2;
5. Mask=bsxfun(f,AxeX,AxeY);
6. obraz(Mask)=0;
7. figure(13)
8. imshow(obraz,[]);

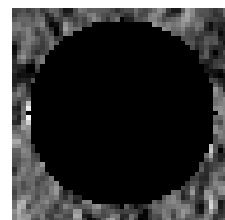
```

Stejným způsobem jako se získala binární maska s intenzitou jedna v místě vnitřku anesteziologické trubice, lze vytvořit masku s kruhem velikosti anesteziologické trubice a jejího vnitřku *Mask* (1.-5. řádek). Indexováním CT snímku vznikne v místě masky černá

plocha o intenzitě 0 (6. řádek), jak je vidět na Obr.6. 3 a.Obr.6. 3 b detailně ukazuje, že se kolem trubice nachází zašuměné pozadí, takže trubice ještě není dostatečně odstraněná.



a) celý snímek



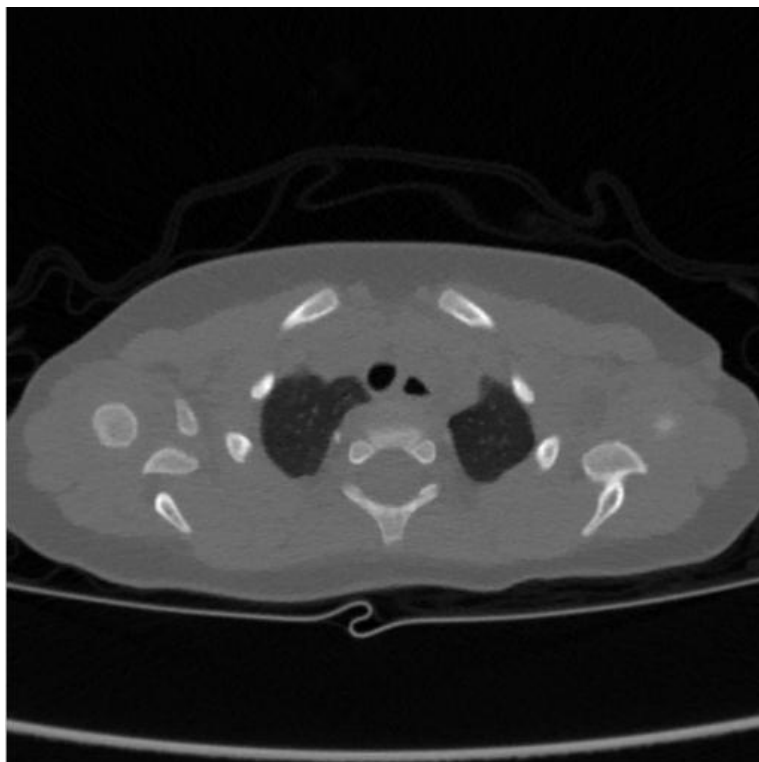
b) část snímku-oblast trubice

**Obr.6. 3** Trubice a její vnitřek s hodnotou intenzity 0

### 6.3 Přičtení vzorku

1.  $\text{obraz}(y1-20:y2+20, x1-20:x2+20) = \text{obraz}(y1-20:y2+20, x1-20:x2+20) + J;$

Porovnáním matice  $J$  (zvětšeným vzorkem) s maticí *nahrazení* (původním vzorkem) lze zjistit souřadnice, kde se má přičíst vzorek  $J$  k trubici a jejího vnitřku s nulovou intenzitou. Vzorek  $J$  se zvětšil o 40 pixelů celkem v ose  $x$  a  $y$ , tzn. z každé strany o 20 pixelů. Na Obr.6. 4 je vidět, jak se přičtením vzorku  $J$  odstranila anesteziologická trubice.



**Obr.6. 4** Příklad vzorku k trubici a jejího vnitřku s nulovou intenzitou

## 7. Zjednodušení Houghovy transformace

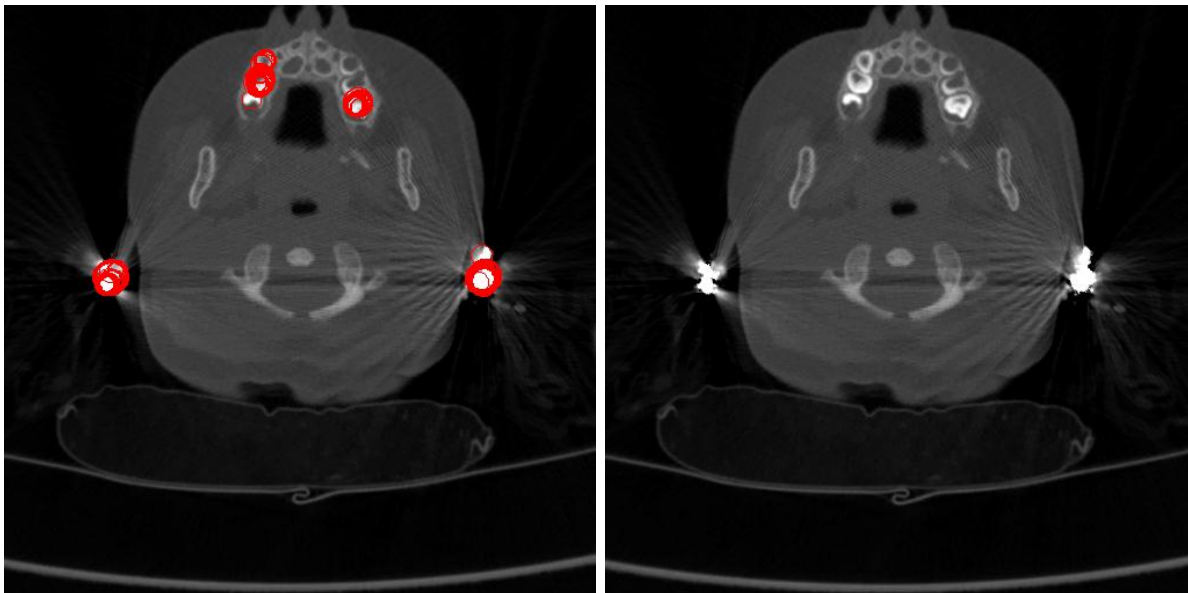
```
1. k=0:2*pi/500:2*pi;
2. for r=polomer
3.     for l=1:length(x)
4.         a=round(x(l)-r*cos(k));
5.         b=round(y(l)-r*sin(k));
6.         for m=1:length(a)
7.             if (a(m)>0) && (a(m)<M) && (b(m)>0) && (b(m)<N)
8.                 akumulator(a(m),b(m))=akumulator(a(m),b(m))+1;
9.             end
10.        end
11.    end
12. houghuv_prostor(:, :, r)=akumulator(:, :);
13. akumulator=zeros(M,N);
14. figure(4)
15. imshow(houghuv_prostor(:, :, r), []);
16. end
```

Houghovu transformaci je potřeba zjednodušit, protože trvá delší dobu, než se celý algoritmus při spuštění provede. Lze odebrat jeden for cyklus, kde se nachází vektor  $k=0:2\pi/500:2\pi$ . Z tohoto vektoru se následně vytvoří proměnná stejné velikosti a umístí před všechny for cykly (1. řádek). Takto se mohou načítat jednotlivé kružnice naráz a ne bod po bodu. Podmínka if se musí upravit indexováním, aby se zkontrolovaly postupně všechny body, zda leží v původním obrazu.

## 8. Rozšíření algoritmu na celý objem dat

Algoritmus se postupně realizuje pro každý CT snímek pomocí cyklu `for i=1:N`, kde  $N$  představuje počet snímků. Příkaz `end` tohoto prvního cyklu se nachází na konci programu, tzn. že ostatní cykly jsou vnořené a musí se u nich nahradit index  $i$  jiným např.  $l$ . Všude, kde se dříve načítal jeden konkrétní řez, např.  $X(:, :, l35)$ , nahradí se číslo indexem  $i$ :  $X(:, :, i)$ .

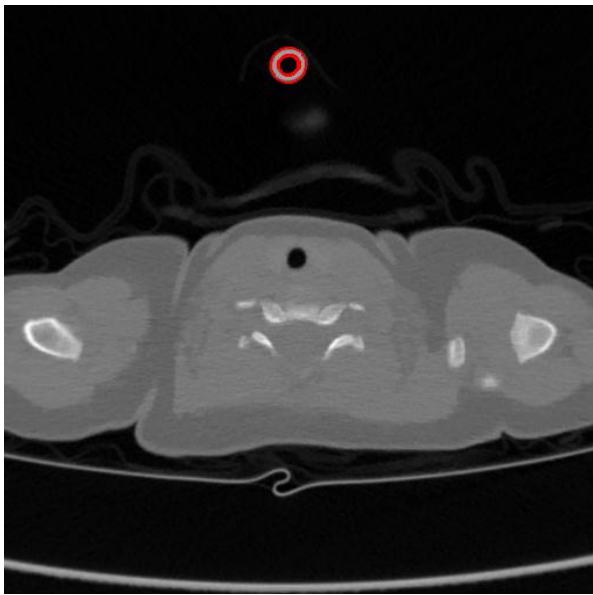
Na následujících obrázcích Obr.8. 1, Obr.8. 2, Obr.8. 3, Obr.8. 4, Obr.8. 5 je vidět srovnání vybraných CT snímků. Pro konkrétní snímek jsou vždy ukázány 2 druhy. První druh zobrazuje nalezené kružnice pomocí Houghovy transformace a druhý představuje stav po provedení algoritmu hledání soustředných kružnic.



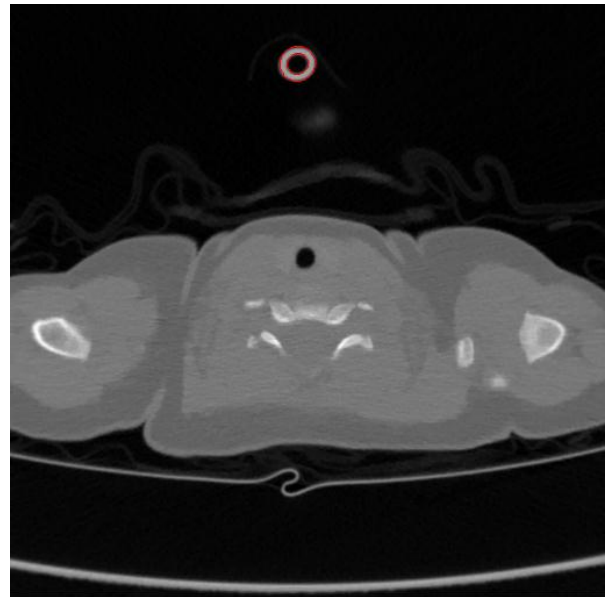
a) nalezené kružnice pomocí HT

b) nalezené soustředné kružnice

**Obr.8. 1** 1. snímek

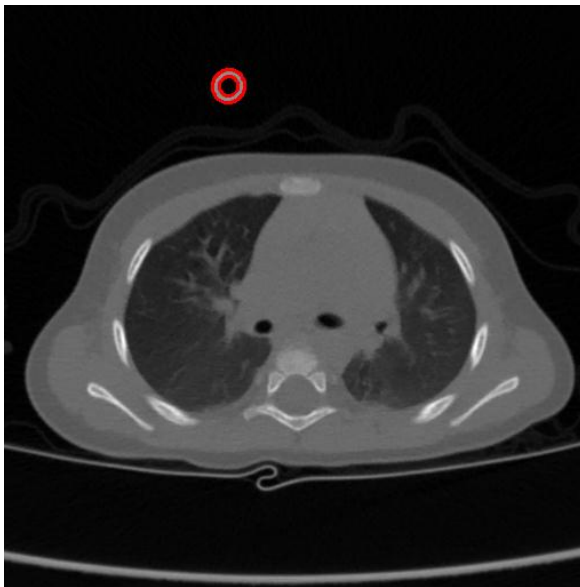


a) nalezené kružnice pomocí HT

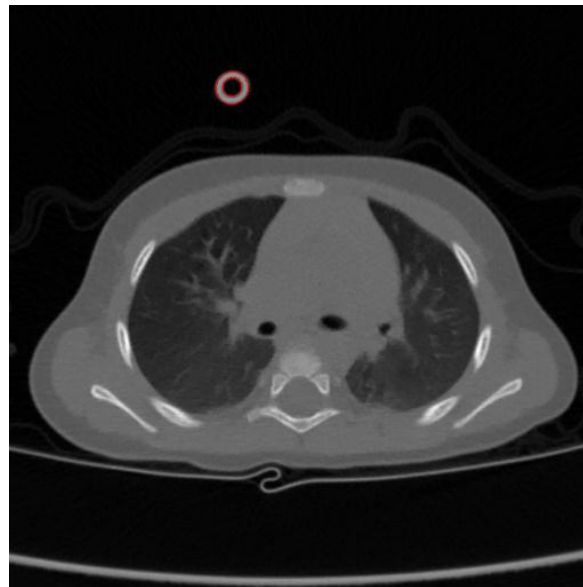


b) nalezené soustředné kružnice

**Obr.8. 2** 100. snímek

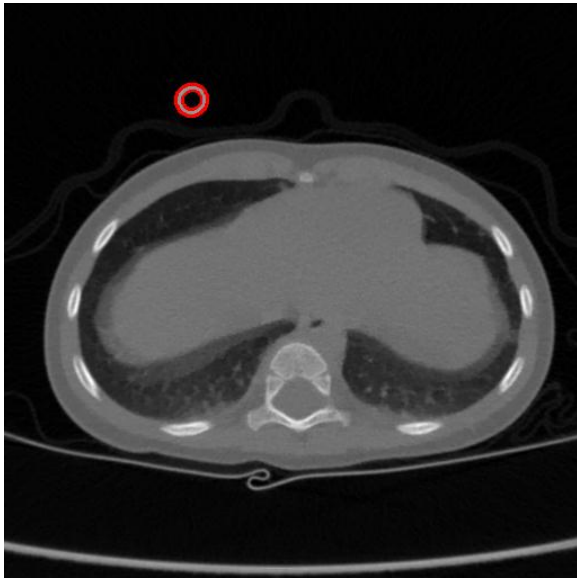


a) nalezené kružnice pomocí HT

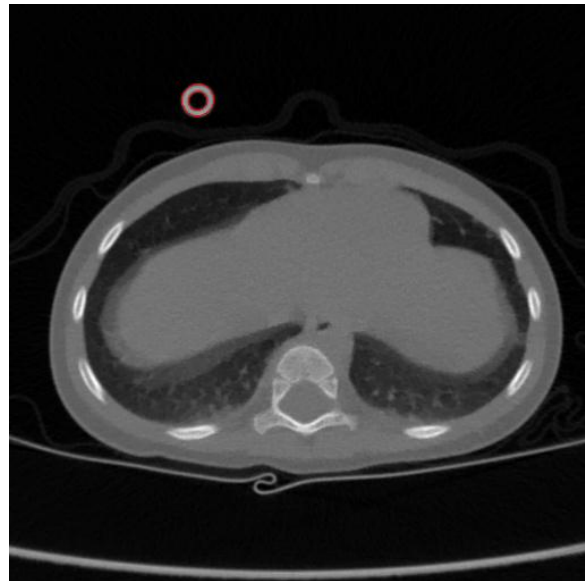


b) nalezené soustředné kružnice

**Obr.8. 3** 200. snímek

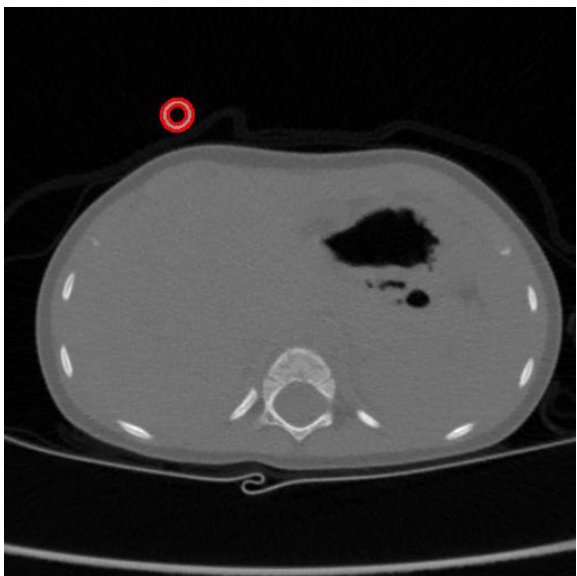


a) nalezené kružnice pomocí HT

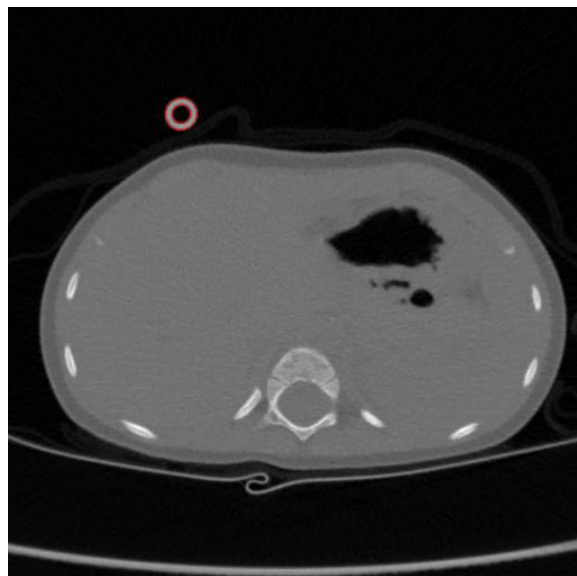


b) nalezené soustředné kružnice

**Obr.8. 4** 300. snímek



a) nalezené kružnice pomocí HT



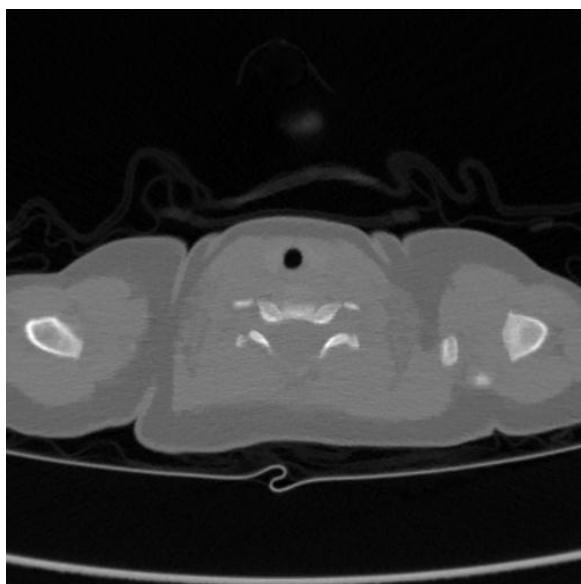
b) nalezené soustředné kružnice

**Obr.8. 5** 357. snímek

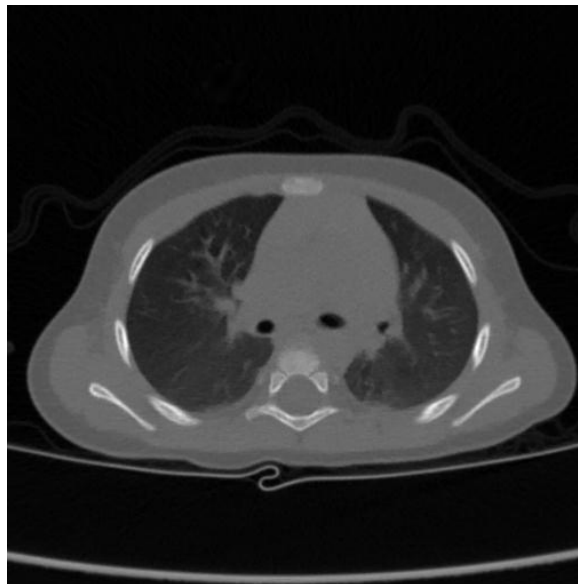
Na 1. snímku se díky Houghově transformaci našlo hodně kružnic viz.Obr.8.4 a). Žádná dvojice však nepředstavuje soustředné kružnice, jak ukazuje Obr.8.4 b). Na dalších snímcích se při hledání soustředných kružnic vždy našly jen kružnice odpovídající hranám anesteziologické trubice. Na snímcích, kde se nachází jen část trubice, kód nelze realizovat. V místě středu trubice totiž vznikne v parametrickém prostoru bod o příliš nízké intenzitě.

Dále se ukázalo, že na snímcích 24 a 25 se našly soustředné kružnice v místě jednoho oka, které má stejné rozměry jako trubice. Proto se musela přidat podmínka pro upřesnění pozice anesteziologické trubice: `if soustredne_kruznice(2,1)<330.`

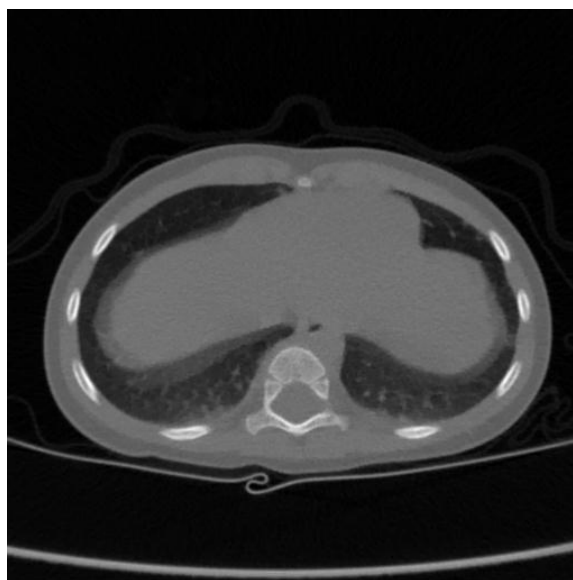
Na dalších vybraných snímcích Obr.8. 6 lze vidět, jak se na nich odstranila anesteziologická trubice.



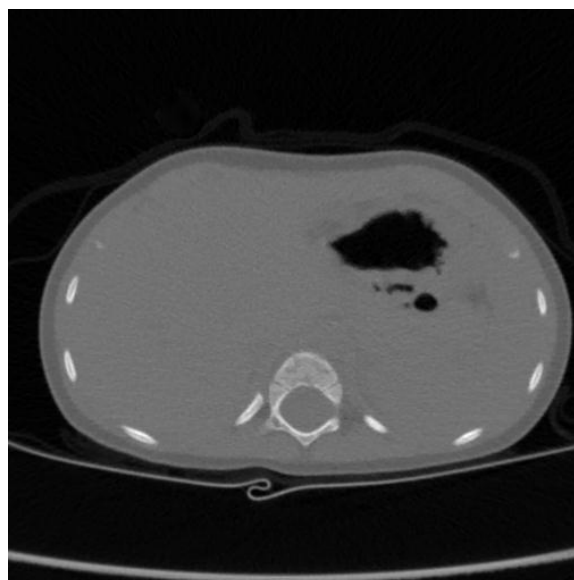
a) 100. snímek



b)200. snímek



c) 300. snímek



d)357. snímek

**Obr.8. 6** Odstranění anesteziologické trubice z vybraných CT snímků

# 9. Vytvoření paralelního prostředí

## 9.1 Paralelní prostředí

Matlab obsahuje jako jednu z možných knihoven funkcí Paralel Computing Toolbox pro vytvoření paralelního prostředí. Pomocí něj lze zpracovávat objemná data nebo složité matematické operace efektivněji a rozdělit úkoly mezi více procesorů. Toto prostředí lze využít pro vytvoření paralelních smyček, paralelních algoritmů, funkcí a další. Paralel Computing Toolbox tak podporuje vývoj a testování různých paralelních aplikací.

## 9.2 Paralelní for cykly- PARFOR

Paralelní *for* smyčky umožňují rozdělení úloh mezi více procesů. Zdrojový kód se tak může provádět několika jádry souběžně. Vytvoření algoritmu pomocí *parfor* cyklů je podobné cyklům *for*. Jediné omezení zde je, že na sobě nesmí být zpracovávány úlohy závislé. Aby bylo možné toto prostředí použít, nejdříve se musí použít příkaz *matlabpool*, který dovoluje vzájemné působení mezi příkazovým oknem a zpracovávanými procesy. Zjistí tak přítomnost těchto procesů.

## 9.3 Realizace paralelního prostředí

```
1. k=0:2*pi/500:2*pi;
2. parfor r=8:20
3.     akumulator=zeros(M,N);
4.     [x y] = find(prah_snimek);
5.     for l=1:length(x)
6.         a=round(x(l)-r*cos(k));
7.         b=round(y(l)-r*sin(k));
8.         for m=1:length(a)
9.             if (a(m)>0) && (a(m)<M) && (b(m)>0) && (b(m)<N)
10.                akumulator(a(m),b(m))=akumulator(a(m),b(m))+1;
11.            end
12.        end
13.    end
14. houghuv_prostor(:, :, r)=akumulator(:, :);
15. akumulator=zeros(M,N);
16. figure(4)
17. imshow(houghuv_prostor(:, :, r), []);
18. end
```

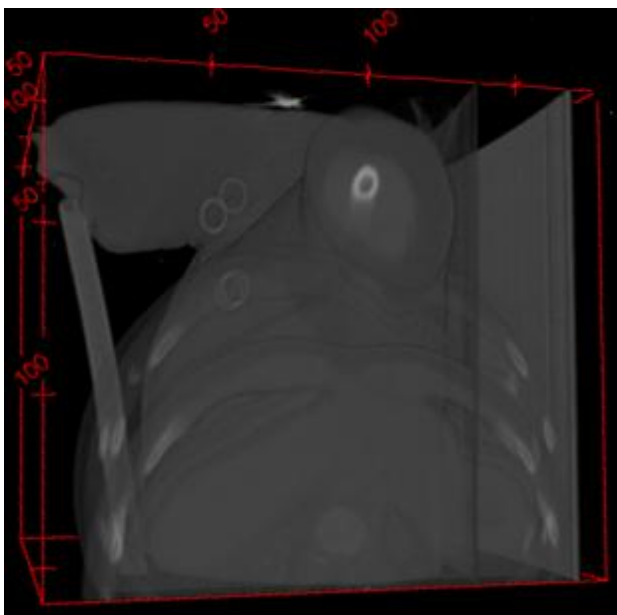
Výpočet byl realizován na dvanáctijádrovém počítači s 48 GB RAM s Intel Xeon CPU, 2.53 GHz. Na začátek celého algoritmu se napíše příkaz *matlabpool open* pro připojení k více procesorům a na konec *matlabpool close*. Je třeba vybrat vhodné *for* cykly, které jsou na sobě nezávislé a šlo by je tak nahradit smyčkami *parfor*. Pro realizaci paralelního prostředí

lze použít algoritmus Houghovy transformace a cyklus, kde se vykreslují kružnice do původního obrazu. U vykreslování kružnic postačí nahradit for příkazem `parfor`. Houghova transformace se ještě musí dopravit tak, že se některé proměnné vloží dovnitř `parfor` cyklu (3. a 4. řádek). Takto se zajistí nezávislost na ostatních úlohách. Ve zbylých cyklech nelze `parfor` jednoduše použít, protože jsou na sobě používané proměnné závislé.

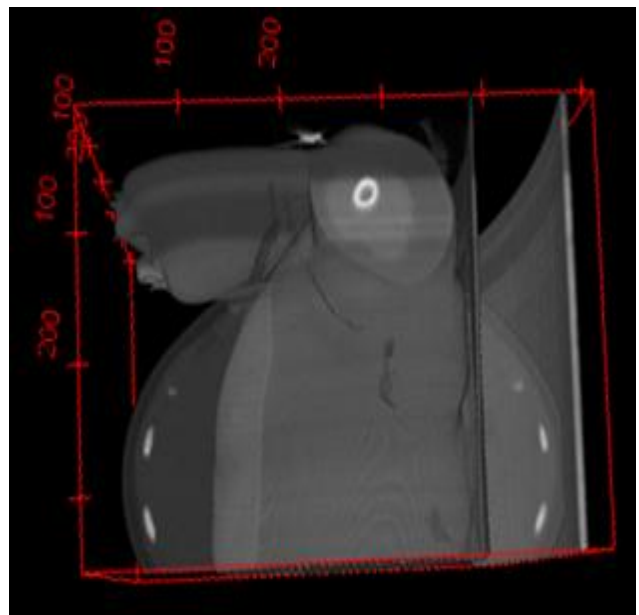
## 10. Zobrazení objemu dat ve 3D

```
1. CC=max(max(obraz));
2. normalizace1 =obraz/max(CC);
3. filename=sprintf('obraz_vysledny%d.dcm',i);
4. dicomwrite(normalizace1,filename);
```

Na rekonstrukci 3D dat lze použít některé speciální programy, jako např. Fiji nebo VG Studio. Program Fiji je dobře dostupný, takže následující snímky jsou zrekonstruovány v něm. Nejdříve se však musí uložit snímky, které zpracoval program vytvořený v Matlabu. Pro ukládání je třeba zpracované snímky normalizovat (1. a 2. řádek) a následně pomocí příkazů `sprintf` a `dicomwrite` ukládat každý snímek do jiného souboru (3. a 4. řádek). Na prvním obrázku Obr.10. 1a je vidět anesteziologická trubice a na druhém snímku s odstraněnou trubicí, jak byly zrekonstruovány v programu Fiji. Pomocí `File\Import\ImageSequence` se načetly obrázky a pomocí `Plugins\3DViewer` se zobrazily 3D data.



a) s anesteziologickou trubicí



b) bez anesteziologické trubice

**Obr.10. 1** 3D data

# 11. Závěr

Tato práce měla za cíl popsat princip Houghovy transformace pro vyhledávání přímky a hlavně kružnice v obrazu a dále se zaměřit na metody zpracování obrazů potřebné k realizaci jeho algoritmu. V práci je popsáno, jak se využívá hranové detekce pomocí různých detektorů, např. Sobelův, Robertsův, Cannyho, atd. Pro naši práci byl dostačující Sobelův operátor, který patří mezi detektory první diference. Dalším úkolem byla realizace předzpracování obrazů a naprogramování algoritmu Houghovy transformace pro vyhledávání kružnic v jednom dvourozměrném řezu daty. Při realizaci hranové detekce se vytvořil gradientní obraz. Aby se mohl použít algoritmus HT, bylo potřeba, aby vznikl binární obraz vhodným prahováním. Následně se realizovala Houghova transformace. Jelikož nejsou známe poloměry hledaných kružnic, muselo se dosadit vhodné rozmezí hodnot, aby šel algoritmus zrealizovat.

V další části se bakalářská práce se zabývala detekcí pozice anesteziologické trubice vyhledáváním soustředných kružnic v Houghově prostoru a odstraněním trubice. Vytvořený algoritmus se rozšířil na celý objem dat a využilo se paralelního prostředí, které slouží pro zpracovávání velkého objemu dat. Soustředné kružnice skutečně odpovídají hranám anesteziologické trubice. Pouze ve snímcích, kde se nachází část trubice, se hrany nedetekovaly a následně ani neodstranily, protože v parametrickém prostoru vznikl v požadovaném místě bod o příliš nízké intenzitě. Dále se u snímku 24 a 25 po použití algoritmu pro hledání soustředných kružnic našly jako soustředné kružnice hrany jednoho oka, které má téměř stejné rozměry jako anesteziologická trubice. Proto se do algoritmu musela přidat podmínka, která upřesnila polohu trubice. Anesteziologická trubice nešla odstranit pouze dosazením intenzit s nulovou hodnotou, protože okolní vzduch má díky šumu v CT datech nenulové hodnoty. Musel se odebrat vzorek pozadí vnitřku trubice a zvětšeným vzorkem trubicí překrýt. Následně se realizovalo paralelní prostředí u vhodných nezávislých for cyklů použitím cyklů parfor. Toto prostředí dokáže jednotlivé části zpracovávat současně na více jádrech počítače a tím zefektivnit zpracování velkého objemu CT dat. Pro názornost byl celý objem snímků zrekonstruován v programu Fiji a ukázán rozdíl mezi daty před a po odstranění anesteziologické trubice.

# Seznam literatury

- [1] JAN, J. Medical Image Processing, Reconstruction and Restoration: Concepts and Methods, CRC, Press, 2005. ISBN 0-8247-5849-8
- [2] KHAIROSFAILZAL, W., NORAINI, A. Eyes Detection in Facial Images using Circular Hough Transform. Signal Processing & Its Applications. CSPA 2009. 5th International Colloquium, p. 238-242, March 2009.
- [3] PEDERSEN S. J. K., Circular Hough Transform. Aalborg University, Vision, Graphics, and Interactive Systems, November 2007.
- [4] DUDA, R. O., HART, P. E. Use of the Hough transformation to Detect Lines and Curves in Pictures. In: Comm. ACM, January, 1972, Vol. 15, s. 1-15. Dostupné z URL: <<http://www.ai.sri.com/pubs/files/tn036-duda71.pdf>>.
- [5] VLACH, J. Hledání úseček a kružnic s využitím Houghovy transformace při zpracování obrazu v LabView [online]. Automa, Únor 2011, s. 42-44. [cit. 7. 10.2011] Dostupné z: <<http://www.odbornecasopisy.cz/res/pdf/42983.pdf>>.
- [6] FISHER R., PERKINS S., WALKER A., WOLFART E. Image Processing Operator Worksheets. The HIPR Copyright 2003. Dostupné z: <<http://homepages.inf.ed.ac.uk/rbf/HIPR2/wksheets.htm>>.
- [7] BALLARD D. H. Generalizing the Hough Transform to detect arbitrary shapes. Computer Science Department, University of Rochester, Rochester, NY 14627, U.S.A . Dostupné z: <<http://comp-eng.binus.ac.id/files/2012/04/D.H.-Ballard-Generalizing-the-Hough-Transform-to-Detect-Arbitrary-Shapes1.pdf>>.
- [8] ŠPANĚL, M., BERAN, V. Obrazové segmentační techniky [online]. Vysoké učení technické v Brně, Říjen 12, 2005 [cit. 5. 11. 2011] Dostupné z: <<http://www.fit.vutbr.cz/spanel/segmentace/>>.
- [9] ATHERTON T. J., KERBYSON D. J. The Coherent Circle Hough Transform. Department of Computer Science, University of Warwick, Coventry, UK. Dostupné z: <<http://www.bmva.org/bmvc/1993/bmvc-93-027.pdf>>
- [10] XING CH., LING L., YANG G. A new concentric circle detection method based on Hough Transform. College of Electrical Engineering and Renewable Energy, China Three Gorges University Yichang, China, Department of Automation Xiamen University Xiamen, China.

# Seznam symbolů, veličin a zkratk

HT Houghova transformace

LoG Laplacian of Gaussian