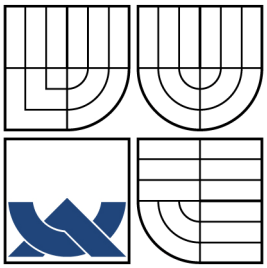
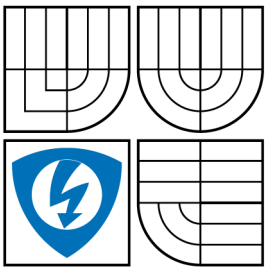


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

HOGHOVA TRANSFORMACE PRO DETEKCI KRUŽNIC

HOUG TRANSFORM FR CIRCLE RECOGNITION

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE — MARTIN KAZÍK
AUTHOR

VEDOUCÍ PRÁCE — ING. KAMIL ŘÍHA, PH.D.
SUPERVISOR

BRNO 2009

VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Martin Kazík ID:98167
Ročník: 3 Akademický rok: 2008/2009

NÁZEV TÉMATU:

Houghova transformace pro detekci kružnic

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte a implementujte algoritmus Houghovy transformace pro detekci kružnic v libovolném obraze formou samostatné aplikace a bez použití zapouzdřené knihovní funkce. Aplikace by měla umožnit načtení libovolného obrazu a přehlednou volbu parametrů algoritmu. Zaměřte se také na možnosti zefektivnění (zrychlení) výpočtu.

DOPORUČENÁ LITERATURA:

[1] GONZALEZ R. C., WOODS R. E.: Digital Image Processing, Prentice Hall, New Jersey, 2002
[2] FISHER R. B.: CVonline: The Evolving, Distributed, Non-Proprietary, On-Line Compendium of Computer Vision, http://homepages.inf.ed.ac.uk/rbf/CVonline/
[3] IOANNOU, D.; HUDA, W.; LAINE, F. A.: Circle Recognition Through A 2D Hough Transform And Radius Histogramming. In Image and Vision Computing Vol-17, Elsevier Science B. V., DOI: 10.1016/S0262-8856(98)00090-0.

Termín zadání: 9.2.2009 Termín odevzdání: 2.6.2009

Vedoucí práce: Ing. Kamil Říha, Ph.D.

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

ANOTACE

Práce je zaměřena na implementaci algoritmu Houghovy transformace pro detekci kružnic. Algoritmus je implementován v jazyce C++ za použití volně dostupné knihovny OpenCv [5]. Jako implementační prostředí bylo zvoleno Microsoft Visual Studio 2008.

V první kapitole je obecně popsána klasická Houghova transformace pro detekci přímek a kružnic. Dále práce obsahuje popis jednotlivých kroků algoritmu Houghovy transformace a popis funkcí OpenCv, které jsou v těchto krocích použity. Podrobně jsou popsány funkce pro převedení obrazu do stupňů šedi, vyhlazení obrazu Gaussovým filtrem a Cannyho hranový detektor pro nalezení hran ve vyhlazeném obraze.

Efektivita a rychlost algoritmu je zvýšena zavedením funkce pro vyhledání potenciálních středů. Princip hledání potenciálních středů je založen na faktu, že přímka kolmá na sečnu kružnice a zároveň procházející středem sečny vždy prochází také středem kružnice samotné.

Výsledky jednotlivých fází algoritmu (převedení do stupňů šedi, vyhlazení Gaussovým filtrem, detekce hran, vytvoření akumulátoru potenciálních středů a vykreslení kružnic) jsou prezentovány na ultrazvukovém snímku kolagenové tepenní náhrady.

V druhé části práce je algoritmus Houghovy transformace využit pro detekci tepny ve snímcích videosekvence zachycené ultrazvukem. Je zde popsána automatizovaná metoda vyhodnocování úspěšnosti detekce tepny. Úspěšnost detekce je testována při změně důležitých parametrů algoritmu. Ze série testů jsou určeny ideální parametry algoritmu pro detekci tepny v dané videosekvenci.

Klíčová slova: Houghova transformace, Cannyho hranový detektor, OpenCv, Rozpoznávání objektů, Detekce kružnic, Počítačové vidění

ABSTRACT

The thesis is focused on the implementation of Hough transform algorithm for circle recognition. Algorithm is implemented in C++ language using open source library OpenCv. As a development environment was chosen Microsoft Visual Studio 2008.

At first there is general description of classical Hough transform for line and circle recognition. Then thesis contains description of particular steps of Hough transform algorithm and description of OpenCv functions witch are used in these steps. There is a detail description of functions for converting image to grayscale, smoothing image by Gaussian filter and Canny edge detector for edge detecting in smoothed image.

Efficiency and speed of algorithm is increased by using function for finding possible centers. This function using the fact that line perpendicular to the chord of circle and going thought its middle point at the same time have to cross the center of the circle.

Results of particular stages of algorithm (converting to grayscale, smoothing by Gaussian filter, edge detection, creating of possible centers accumulator and drawing circles) are presented on ultrasonic image of collagen arterial substitute.

In the second part of the thesis the algorithm is used for detection of artery in frames of video captured by ultrasound. There is a description of automatic method for evaluating of success of artery detection. Success of detection is analyzed by changing values of important algorithm parameters. From series of tests there are defined ideal parameters of algorithm for artery detection in the video.

Keywords: Hough transform, Canny edge detector, OpenCv, Object recognition, Circle recognition, Computer vision.

KAZÍK, M. *Houghova transformace pro detekci kružnic.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 53 s. Vedoucí bakalářské práce Ing. Kamil Říha, Ph.D.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Houghova transformace pro detekci kružnic jsem vypracoval 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 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 ....

podpis autora

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Kamilu Říhovi, Ph.D. za konstruktivní kritiku a připomínky při vývoji programu i při psaní této práce.

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

1 ÚVOD

Rozpoznávání objektů v digitálním obraze je velmi důležitá oblast digitálního zpracování obrazu a nachází rozsáhlé využití ve vědecké činnosti ale i v praxi. Mnou implementovaná metoda rozpoznávání kružnic pomocí Houghovy transformace byla poprvé představena Paulem Hough v roce 1962 [6]. Tato metoda slouží k detekci geometrických tvarů, jako jsou přímka, kružnice nebo elipsa, v digitálním obraze. V praxi se používá například ve sledování dopravy nebo v robotice. V tomto konkrétním případě je Houghova transformace implementována pro detekci kružnic v ultrazvukových snímcích tepenních náhrad a měření jejich poloměrů. Silnou stránkou algoritmu Houghovy transformace je robustnost algoritmu. To znamená, že algoritmus dobře zvládá rozpoznávání tvarů i v zašuměném obraze, dokáže si poradit i s tvarovými nepřesnostmi a nesouvislostí hledaných tvarů. Tato vlastnost algoritmu je velice důležitá, protože snímky zachycené ultrazvukem obsahují mnoho šumu. Hlavní nevýhodou je poměrně velká výpočetní náročnost, která značně omezuje praktické využití. K implementaci je použital knihovna OpenCV, která byla vytvořena firmou Intel a je volně ke stažení. Tato knihovna poskytuje řadu funkcí a datových struktur pro práci s digitálním obrazem [5]. Metody implementované v OpenCV byly použity pro úpravu obrazu a nalezení hran. Na takto upravený obraz byla pak aplikována moje vlastní funkce pro Houghovu transformaci. Implementace algoritmu je provedena v jazyce C++ v prostředí Microsoft Visual Studio 2008, které mám jako student VUT zdarma k dispozici.

Během vypracovávání této práce byla pozornost zaměřena spíše na efektivitu algoritmu a přesnost detekce kružnic, od vytváření grafického uživatelského rozhraní bylo upuštěno. Spuštění programu a nastavování parametrů proto vyžaduje vývojové prostředí a instalaci knihovny OpenCv. Návod k nastavení Microsoft Visual Studia 2008 pro použití knihovny OpenCv je uveden v příloze. Na úkor grafického uživatelského rozhraní bylo zadání rozšířeno o úpravu algoritmu pro využití k detekci tepny v ultrazvukové videosekvenci.

2 KLASICKÁ HOUGHOVA TRANSFORMACE

Klasická Houghova transformace se používá k detekci přímek a kružnic v digitálním obraze. Metoda spočívá v nalezení parametrického popisu objektů v obraze a transformaci kartézského souřadnicového systému do systému polárního [1].

2.1 Detekce přímek

Nejjednodušší Houghova transformace pro nalezení přímek využívá parametrického popisu přímky pomocí rovnice

$$y = mx + b \tag{2.1}$$

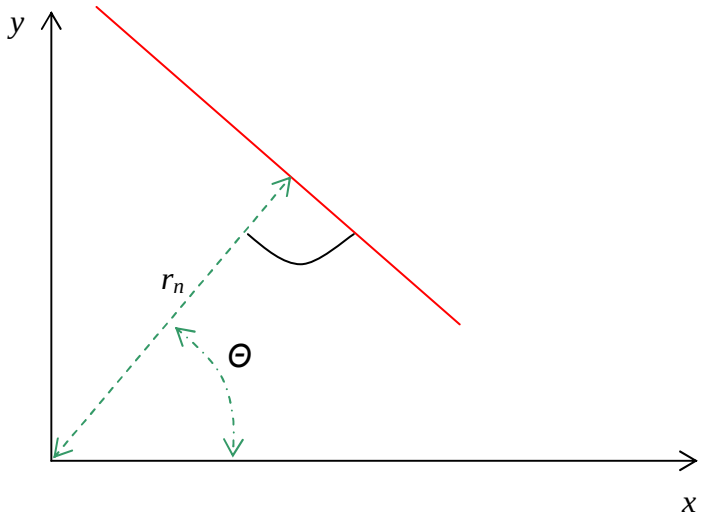
Pro každý bod (*x*,*y*) můžeme změnou parametru *m* a dopočítáváním *b* vykreslit přímku. Každá přímka se tudíž zobrazí v parametrickém prostoru (*m*,*b*) jako bod (*x*,*y*). Kvůli neohraničenému parametru *m* ve výrazu (2.1) (může nabývat hodnot $(-\infty, \infty)$) je výhodnější metodou detekce přímek použití parametrického popisu pomocí výrazu

$$y = \left(-\frac{\cos\Theta}{\sin\Theta}\right)x + \left(\frac{r}{\sin\Theta}\right). \tag{2.2}$$

Tento výraz můžeme upravit na

$$r(\Theta) = x \cdot \cos\Theta + y \cdot \sin\Theta. \tag{2.3}$$

Takto je možné vyjádřit každou přímku v obraze párem (*r*, Θ) , který je jedinečný pokud $\Theta \in [0, \pi]$ a $r \in R$ nebo $\Theta \in [0, 2\pi]$ a $r \geq 0$. Na obrázku 2.1 je vidět jak tento pár (*r*, Θ) vzniká.



Obr. 2.1: Parametrický popis přímky.

Postupnou změnou parametru Θ od *0* do π a dopočítáváním parametru *r* získáme sinusoidu, která je jedinečná pro daný bod. Bod, kde se sinusoidy protínají v parametrickém prostoru určuje přímku v prostoru obrazu.

Prostor, do kterého jsou vykreslovány jednotlivé křivky, se nazývá akumulátor. Při implementaci je akumulátor nejprve vynulován, poté je každý bod (*x*,*y*) transformován na křivku a body této křivky jsou v akumulátoru inkrementovány. Nalezením maxima v akumulátoru lze získat parametry přímky nalezené v obraze.

Při hledání přímek má akumulátor pouze dva rozměry (*r*, Θ), jako akumulátor lze tedy použít dvourozměrný obraz a inkrementovat jas obrazových bodů. Toto je výhodné jelikož lze výsledky transformace zobrazit.

2.2 Detekce kružnic

Houghova transformace pro detekci kružnic funguje analogicky k metodě pro detekci přímek.

Kružnice je parametricky popsána vztahem

$$r^2 = (x - x_0) + (y - y_0), \tag{2.4}$$

kde *x*,*y* jsou souřadnice bodu v prostoru obrazu, body x_0,y_0 jsou souřadnice středu kružnice a *r* je poloměr kružnice. Parametrický prostor bude mít v tomto případě tři souřadnice (x_0,y_0,*r*).

Pro každý hranový (obvodový) bod (tj. bílý bod v binárním obraze) měníme x_0,y_0 a dopočítáváme r. To znamená, že každý nehranový bod je považována za potenciální střed kružnice. Pokud se nějaký potenciální střed (x_0,y_0) vyskytuje v parametrickém prostoru pro konkrétní rádius r často, je velice pravděpodobné, že se v obraze nachází kružnice s odpovídající souřadnicí středu (x_0,y_0) a velikostí poloměru r.

Při detekci kružnic má akumulátor tři rozměry, proto již nelze jako akumulátor použít dvourozměrný obraz. Je nutné vytvořit datovou strukturu ve které bude inkrementována hodnota odpovídající trojici hodnot (x_0,y_0,r).

3 PŘEDZPRACOVÁNÍ OBRAZU

Před aplikací Houghovy transformace je zapotřebí vstupní obraz upravit tak, aby všechny body kromě hranových měly hodnotu nula. Jednotlivé fáze této úpravy budou popsány v této kapitole. Na všechny níže popsané úpravy používám již definované funkce implementované v knihovně OpenCV.

3.1 Převedení obrazu do stupňů šedi

V obraze ve stupních šedi je každý bod obrazu vyjádřen hodnotou intenzity bílé (0-255 pro osmibitový obraz). Při převádění obrazu do stupňů šedi se jasová složka (tedy hodnota intenzity bílé) určí ze vztahu

$$Y = 0{,}299R + 0{,}587G + 0{,}114B\,, \tag{3.1}$$

kde Y je jasová složka a R,G,B jsou jednotlivé barevné složky obrazu.
V algoritmu je použita funkce `cvCvtColor` s parametrem `CV_BGR2GRAY`.

3.2 Vyhlazení obrazu Gaussovým filtrem

Jedná se o odstranění šumu pomocí konvoluce s Gaussovoým konvolučním jádrem, jehož prvky se vypočítají z následující funkce [6]

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{x^2+y^2}{2\sigma^2}}\,, \tag{3.2}$$

kde x je vzdálenost od počátku na horizontální ose, y je vzdálenost od počátku na vertikální ose a σ je standardní odchylka Gaussovy distribuce. Na obrázku 3.1 je příklad konvolučního jádra pro $\sigma = 1$.

0,000007225	**0,000394505**	**0,004806062**	**0,000394505**	**0,000007225**
0,000394505	**0,035512267**	**0,096532352**	**0,035512267**	**0,000394505**
0,004806062	**0,096532352**	**0,159154943**	**0,096532352**	**0,004806062**
0,000394505	**0,035512267**	**0,096532352**	**0,035512267**	**0,000394505**
0,000007225	**0,000394505**	**0,004806062**	**0,000394505**	**0,000007225**

Obr. 3.1: Gaussovo konvoluční jádro velikosti 5x5 pro $\sigma = 1$.

OpenCv poskytuje funkci `cvSmooth`, která nabízí více metod potlačení šumu. Parametrem `CV_GAUSSIAN` je v programu zvolen Gaussův filtr. Dalšími parametry je možné zvolit také rozměr konvolučního jádra. Funkce je pro zadanou velikost konvolučního jádra v algoritmu volána dvakrát za sebou.

3.3 Detekce hran (Cannyho hranový detektor)

Pro detekci hran je použit Cannyho hranový detektor [4]. Je to vyspělý algoritmus, který je schopen detekovat hrany s vysokou přesností a nízkou chybovostí. V OpenCV je implementován ve funkci `cvCanny`.

Postup detekce hran je rozdělen do několika kroků. Na vyhlazený obraz je nejprve aplikován **Sobelovův operátor** pro nalezení gradientů a jejich směrů pro každý bod obrazu. Matematicky je gradient určen jako 2D vektor s komponenty určenými derivacemi v horizontálním a vertikálním směru. Výsledek ukazuje kvantitativně vyjádřenou míru, s jakou se mění jas v daném obraze v daných směrech. V bodech s největší změnou se nejpravděpodobněji nacházejí hrany.

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \qquad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Obr . 3.2: Sobelovy konvoluční operátory pro horizontální a vertikální směr gradientu.

Další fází Cannyho algoritmu je tzv. „**ztenčení“.** Z hodnot gradientů vrácených v předchozím kroku se vyberou pouze lokální maxima. Respektive odeberou se body, které lokálními maximy nejsou.

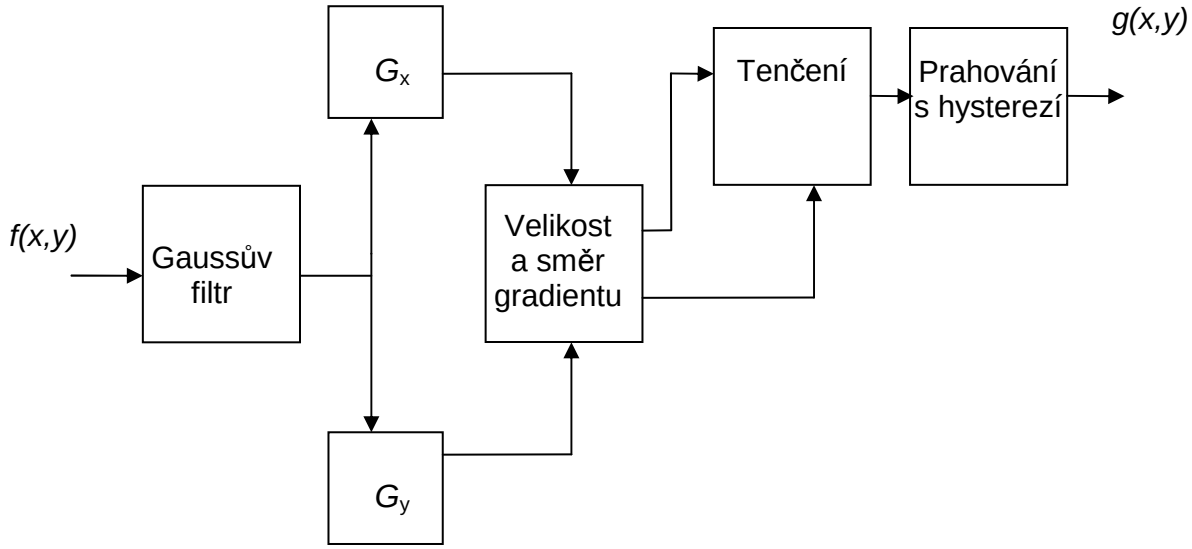
V praxi to znamená, že se porovnávají pixely ve směru a proti směru gradientu. Pokud je nalezen pixel, který má větší hodnotu (gradientu) než jeho dva sousedé ve směru a proti směru gradientu, je tento pixel označen za hranový.

Pokud pixel tuto podmínku nesplňuje, je označen jako nehranový. Směry gradientů, ve kterých porovnáváme jednotlivé pixely a jejich velikosti, jsou stanoveny v předchozím kroku.

Posledním krokem detekce je **prahování s hysterezí.** Většinou se při prahování používá jediný prahovací limit, což znamená, že pokud hodnoty hranových bodů kolísají pod a nad tímto limitem, hrana je detekována přerušovaná. Proto se při prahování s hysterezí zvolí dva prahové limity. Pokud, je hodnota bodu nad vyšším limitem, je bod automaticky přijat jako hranový. Pokud, je však hodnota bodu pod hodnotou nižšího limitu, je bod automaticky považován za nehranový. Bod s hodnotou ležící uprostřed dvou limitů je považován za hranový, pouze, pokud jemu sousedící bod už byl za hranový bod prohlášen. Doporučuje se volit rozmezí mezi horním a dolním prahovacím limitem v poměru dvě nebo tři ku jedné [5].

Funkce `cvCanny` nabízí možnost zvolit velikost Sobelových operátorů a také limitů pro prahování s hysterezí.

Blokové schéma Cannyho hranového detektoru je na obrázku 3.3.



Obr. 3.3: Blokové schéma Cannyho hranového detektoru.

4 IMPLEMENTACE

V následující kapitole bude popsán postup implementace Houghovy transformace pro detekci kružnic. Budou zde diskutované některé matematické vztahy, které byly pro implementaci použity.

4.1 Obecná část

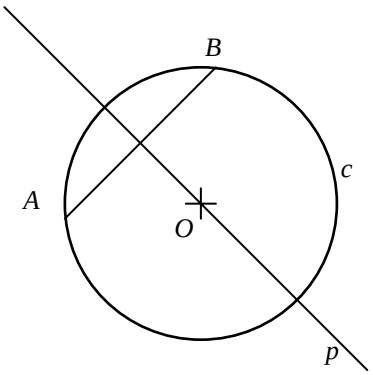
V této kapitole popíšu postup při implementaci algoritmu pro detekci kružnic v obecném barevném obraze.

4.1.1 Předzpracování obrazu

V této fázi algoritmu je obraz nejprve převeden do stupňů šedi. Následně je obraz vyhlazen Gaussovým filtrem a ve vyhlazeném obraze jsou pomocí Cannyho detektoru nalezeny hrany. Tato fáze implementace je popsána v kapitole 3.

4.1.2 Naplnění akumulátoru potenciálními středy

Protože považovat každý bod hranového obrazu za potenciální střed je příliš výpočetně náročné, byla implementována metoda, která počet potenciálních středů podstatně sníží. Nejprve jsou sestrojeny tětivy kružnice a poté se v jejich středech vynesou přímky na ně kolmé. Takto vzniklá přímka musí vždy procházet středem kružnice, viz obr. 4.1.

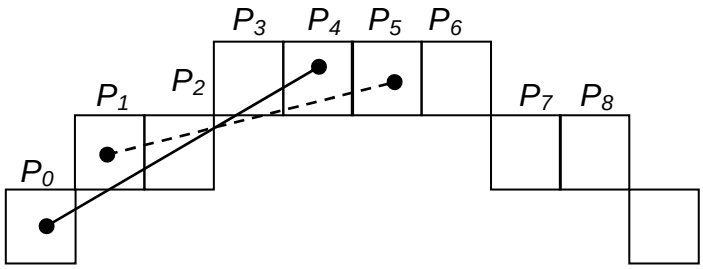


Obr. 4.1: Metoda nalezení potenciálních středů.

Obrázek 4.1 ilustruje, že střed *O* kružnice *c* leží na přímce *p*, která je kolmá na úsečku *AB*. Přitom přímka *AB* je tětivou kružnice *c*.

Při implementaci metody je nejprve třeba v hranovém obraze najít spojené komponenty, tzn. řetězce pixelů, které jsou navzájem spojené v jednom ze svých prvků v osmiokolí. K nalezení spojených komponentů je použita funkce `cvFindContours`. Funkce uloží spojené komponenty do speciálního datového pole typu `CvSeq`.

K takto uloženým komponentům lze jednoduše přistupovat a získávat souřadnice jednotlivých bodů. Elementy jednotlivých spojených komponent jsou spojovány přímkami pomocí tzv. plovoucího okénka. Princip plovoucího okénka je zobrazen na obrázku 4.2.



Obr. 4.2: Princip plovoucího okénka.

Postupně jsou vyčítány body ze spojených objektů, vzdálenost mezi vyčítanými body určuje velikost plovoucího okénka, tedy pro body P_0-P_4, P_1-P_5 je velikost okénka čtyři. Pokud jsou známy dva body $A[x_A,y_A]$ a $B[x_B,y_B]$, je možné vypočítat směrový vektor přímky, která těmito body prochází.

$$u = B - A. \tag{4.1}$$

$$u[x_u, y_u] = (x_B - x_A, y_B - y_A). \tag{4.2}$$

Vztah (4.1) je obecný výpočet směrového vektoru přímky procházející body *A,B*. Vztah (4.2) znázorňuje výpočet jednotlivých souřadnic vektoru.

Dále je nutné vypočítat středový bod úsečky dané směrovým vektorem a dvojící bodů *A,B*.

Souřadnice středového bodu jsou vypočteny ze vztahu

$$p[x_p, y_p] = \left(\frac{x_A + x_B}{2}, \frac{y_A + y_B}{2} \right). \tag{4.3}$$

Nyní je znám směrový vektor i středový bod úsečky, pomocí těchto údajů je možné vyjádřit přímku, která prochází středem úsečky a zároveň je na ni kolmá. Taková přímka je popsána vztahem

$$y = \left(\frac{-x_u \cdot \left(x - x_p\right)}{y_u} \right) + y_p \,. \tag{4.4}$$

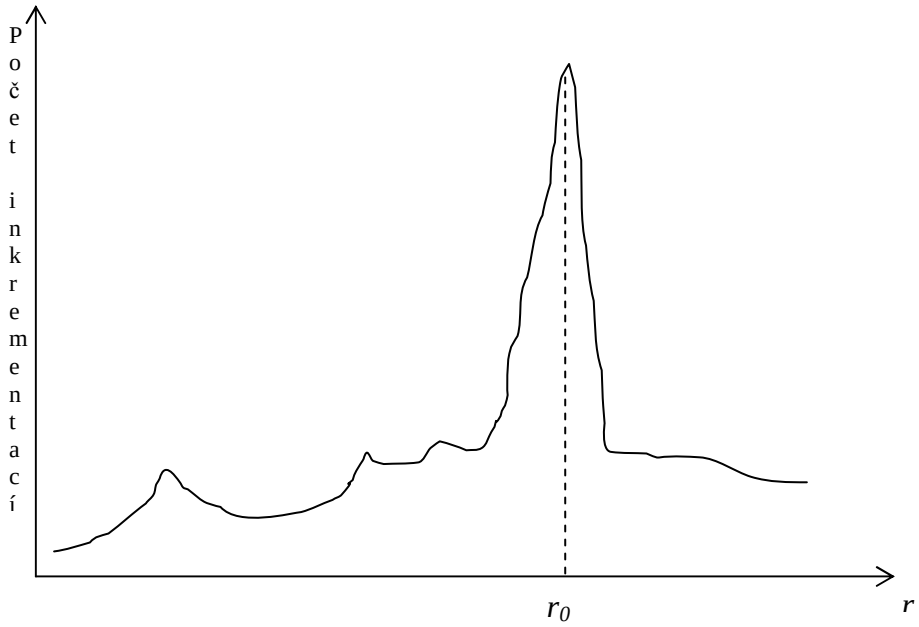
Nyní je nutné vytvořit akumulátor jedná se o dvourozměrný obraz, jehož všechny body jsou nastaveny na nulu. Postupnou změnou *x* a dopočítáváním *y* jsou ze vztahu (4.4) získávány body ležící na přímce. Body náležící přímce kolmé na tětivu kružnice jsou v akumulátoru inkrementovány. Pokud má spojený objekt tvar kružnice, prochází všechny takto vytvořené přímky středem této kružnice, proto je středový bod nejčastěji inkrementován (má největší hodnotu). Pokud je tvar spojeného objektu jiný než kruhový, nebudou stejné body inkrementovány tolikrát. Nalezením určitého počtu nejvyšších hodnot v akumulátorovém obraze jsou získány souřadnice potenciálních středů kružnic.

4.1.3 Zjištění poloměrů kružnic

Pro zjištění poloměrů kružnic jsou postupně procházeny všechny hranové body a ze vztahu (4.5) jsou dopočítávány vzdálenosti těchto bodů od daného potenciálního středu. Vzdálenost bodu od potenciálního středu se vypočte ze vztahu

$$r = \sqrt{\left(x - x_0\right)^2 + \left(y - y_0\right)^2} \,. \tag{4.5}$$

V datovém akumulátoru je inkrementována položka odpovídající dané vzdálenosti. Pokud je v obraze kružnice objeví se velké množství bodů se stejnou vzdáleností od středu. Tato vzdálenost je pak s velkou pravděpodobností poloměrem kružnice. Na obrázku 4.3 je znázorněn akumulátor pro jeden potenciální střed. Výsledek průchodu hranových bodů ukazuje, že nejvíce bodů se nachází ve vzdálenosti r_0 od zjištěného středu. V ostatních vzdálenostech se nachází pouze (poměrně) malé množství hranových bodů. Z toho vyplývá, že r_0 je poloměrem nejpravděpodobnější kružnice, která se v obraze nachází.



Obr. 4.3 Nalezení poloměrů kružnic.

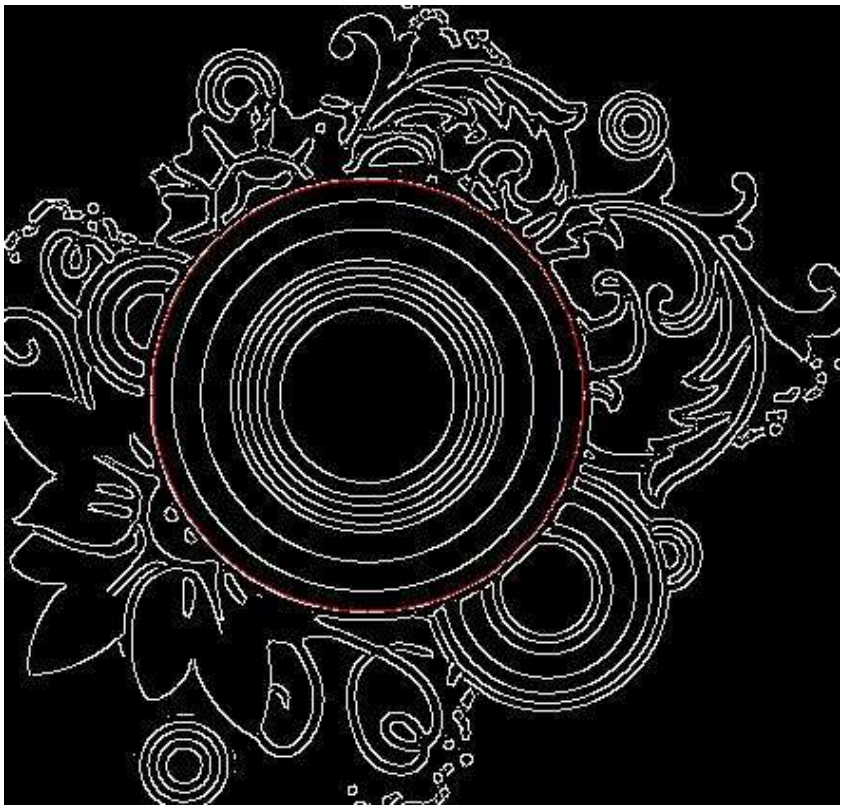
4.1.4 Seřazení kružnic podle pravděpodobnosti

Informace o nalezených kružnicích jsou po vyhledání uloženy v akumulátoru. Za nejpravděpodobnější je považována ta kružnice, u které program nalezne nejvíce obvodových bodů (tedy bodů se stejnou vzdáleností od středu) pro daný střed. Informace o kružnicích jsou vyčítány z akumulátoru a řazeny při vkládání do nového datového pole podle počtu obvodových bodů. Tato metoda řazení se nazývá Insert sort.

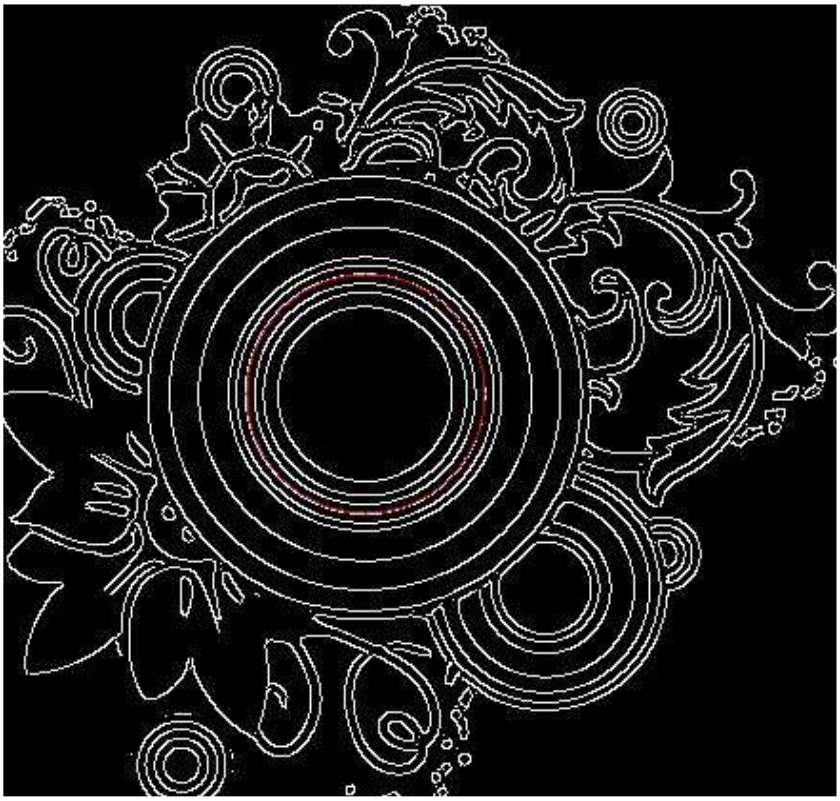
Pokud jsou ovšem kružnice řazeny podle počtu obvodových bodů, vyvstává zde problém zvýhodnění větších kružnic před menšími. Kružnice o větším obvodu má logicky více obvodových bodů než kružnice o obvodu menším.

Na obrázku 4.4 je znázorněna nejpravděpodobnější kružnice nalezená v obraze s větším počtem kružnic různých poloměrů. Výsledek detekce dokazuje, že větší kružnice jsou zvýhodněny. Tento problém je v programu vyřešen normalizací hodnoty počtu nalezených obvodových bodů. Normalizace je provedena dělením počtu nalezených obvodových bodů hodnotou poloměru kružnice. Nejpravděpodobnější kružnice, která je programem detekována po zavedení normalizace, je znázorněna na obrázku 4.5. O pravděpodobnosti kružnice po zavedení normalizaci již nerozhoduje velikost poloměru, ale přesnost („kulatost“) a plnost jejího tvaru.

Normalizace se provádí při řazení kružnic ve funkci `insertSort`.



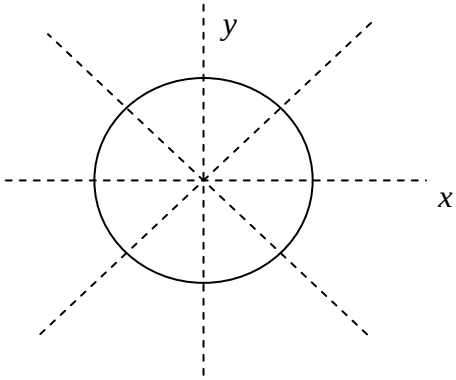
Obr. 4.4: Důkaz zvýhodnění kružnic s větším poloměrem.



Obr. 4.5: Nejpravděpodobnější kružnice po zavedení normalizace.
(červeně vykreslená kružnice je detekována jako nejpravděpodobnější)

4.1.5 Vykreslení kružnic

K vykreslování kružnic je použita funkce `cvCircle`, která je součástí knihovny OpenCV. Tato funkce vykresluje kružnice s použitím Bresenhamova algoritmu. Tento algoritmus využívá symetrie kružnice, jak je znázorněna na obrázku 4.6.



Obr. 4.6: Osy symetrie kružnice.

Jak je vidět na obrázku, kružnice je symetrická podle os *x*,*y* a podle dvou diagonálních os. Z toho plyne, že je nutné vykreslit pouze jednu osminu kružnice (45^{o}) zbytek je symetrický[6].

4.2 Implementace pro detekci tepny v ultrazvukové videosekvenci

V této části je popsáno použití algoritmu Houghovy transformace pro detekci tepny v ultrazvukové videosekvenci. Detekovaná kružnice není v tomto případě vykreslována do výsledného obrazu. Detekce se provádí pro všechny snímky testovacího videa *19_tepu.avi*. Jedná se o ultrazvukovou videosekvenci, na které je zachycena krční tepna vedoucího této práce Ing. Kamila Říhy, Ph.d..

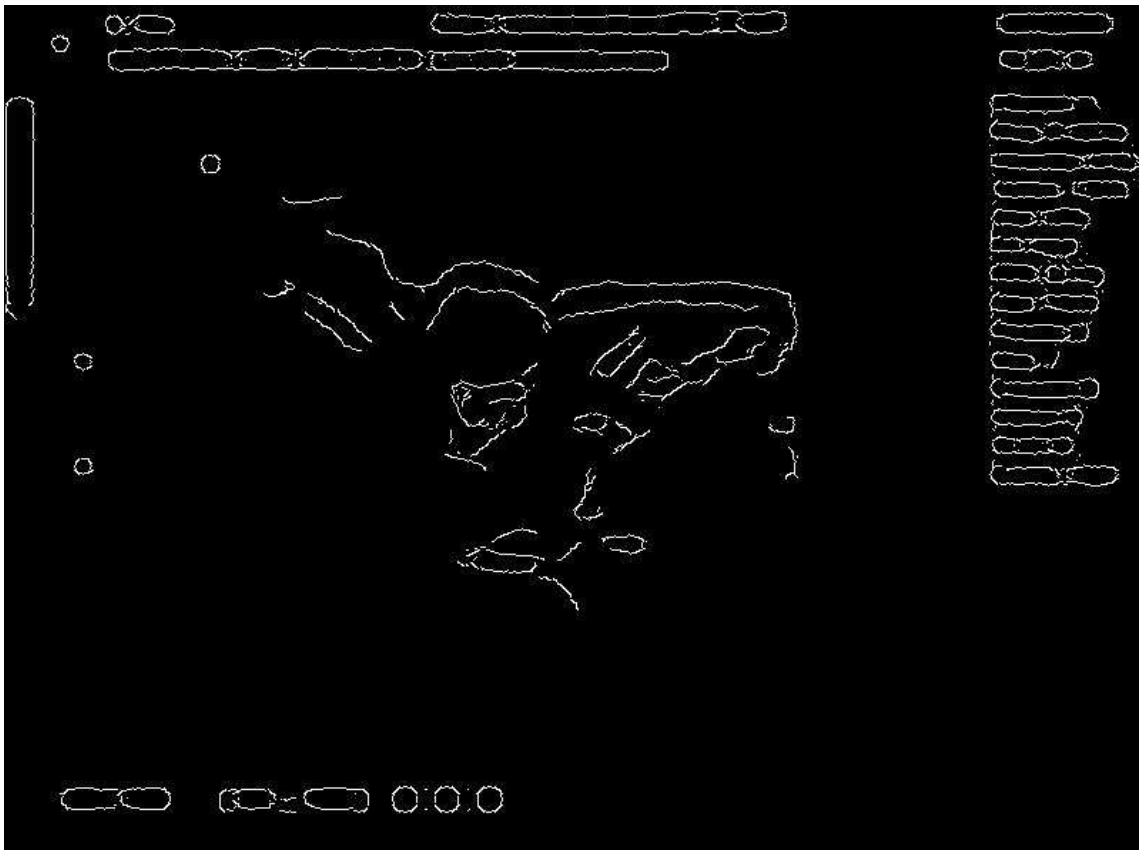
Videosekvence je dlouhá 13 sekund a obsahuje 421 snímků. První snímek videosekvence *19_tepu.avi* je zobrazen na obrázku 4.7.

Obr. 4.7: První snímek videosekvence *19_tepu.avi*.

4.2.1 Úprava algoritmu

Algoritmus pro detekci kružnic je upraven tak, aby ho bylo možno použít jako knihovnu funkcí v jakémkoli programu. Funkce hough které je jako parametr předán jeden ze snímků videa, vrátí souřadnice středu a také poloměr nejpravděpodobnější kružnice detekované v daném snímku.

Do funkce hough je také implementován kód, který vyřízne ze vstupního obrazu střední část. Tato úprava znamená výrazné zrychlení vykonání algoritmu a také velmi znatelně zvýší pravděpodobnost úspěšného nalezení tepny ve snímku (viz kapitola výsledky). Jak je vidět z obrázku 4.7, na okrajích snímku jsou technické údaje ve formě textu algoritmus by bez oříznutí v této oblasti detekoval velké množství hranových bodů. Hrany detekované v prvním snímku videosekvence *19_tepu.avi* jsou zobrazeny na obrázku 4.8.



Obr. 4.8 Výsledek detekce hran pro první snímek videa *19_tepu.avi*.

(hrany byly detekovány s parametry: Cannyho prahy T_1=135 a T_2=405, velikost Gaussova konvolučního jádra=30x30)

Pro každý hranový bod, který je v obraze nalezen, musí algoritmus vyhodnotit, zda se nejedná o bod náležící k nějaké kružnici proto je bez oříznutí délka běhu algoritmu výrazně delší než při zavedení oříznutí. Vzhledem k nepřesnosti tvaru a nevýrazné hraně hledané kružnice (viz obr. 4.8), dochází v oblasti písma k velkému množství chybných detekcí. Hustota hranových bodů detekovaných v oblasti písma je v porovnání s hustotou hranových bodů detekovaných ve střední části obrazu velmi vysoká. To je důvod, proč jsem při testování zaznamenal nejvíce chybných detekcí v pravém okraji snímku.

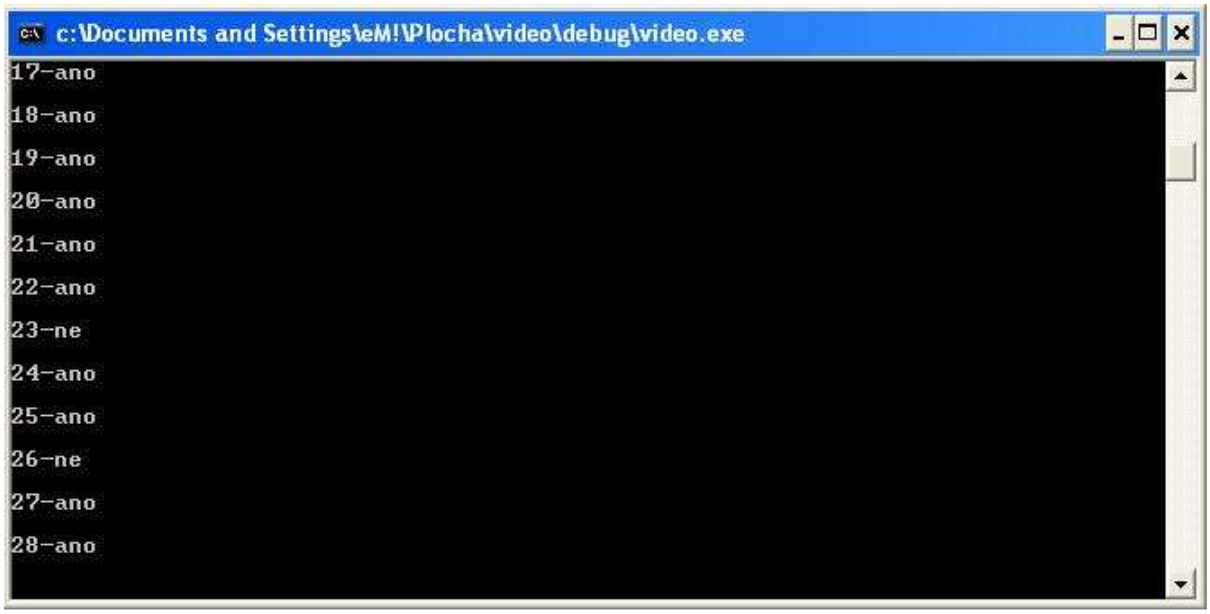
V algoritmu je také implementováno omezení poloměru hledané kružnice a to od 25 do 55 pixelů. Poloměr hledané tepny se během srdečního rytmu mění přibližně od 30 do 35 pixelů.

4.2.3 Testování úspěšnosti detekce

Jak je již výše uvedeno videosekvence *19_tepu.avi* obsahuje 421 snímků nebylo by proto z časových důvodů možné spouštět detekci tepny pro každý snímek zvlášť. Proto

je k tomuto účelům vytvořen jednoduchý program, který v cyklu prochází snímky videosekvence a u každého snímku spustí detekci.

Program je schopen automaticky rozpoznat, zda byla detekce tepny v daném snímku úspěšná či nikoliv. Výsledek detekce pro jednotlivé snímky je vypsán na standardní výstup (do konzole) spolu s číslem odpovídajícího snímku, viz obr. 4.10.



Obr. 4.10: Zobrazení úspěšnosti detekce na standardním výstupu.

Zároveň je údaj o úspěšnosti zapisován do textového souboru.

Jak je již uvedeno výše, funkce `hough` která je na snímek aplikována vrací pouze střed nejpravděpodobnější kružnice nalezené ve snímku. Aby bylo možno automaticky rozpoznat, zda detekce proběhla úspěšně je tento bod využit jako „semínko“ pro vyplňovaní funkci `cvFloodFill`. Podle počtu vyplněných pixelů program rozpozná zda byla vyplněna oblast tepny, nebo jiná oblast v obraze.

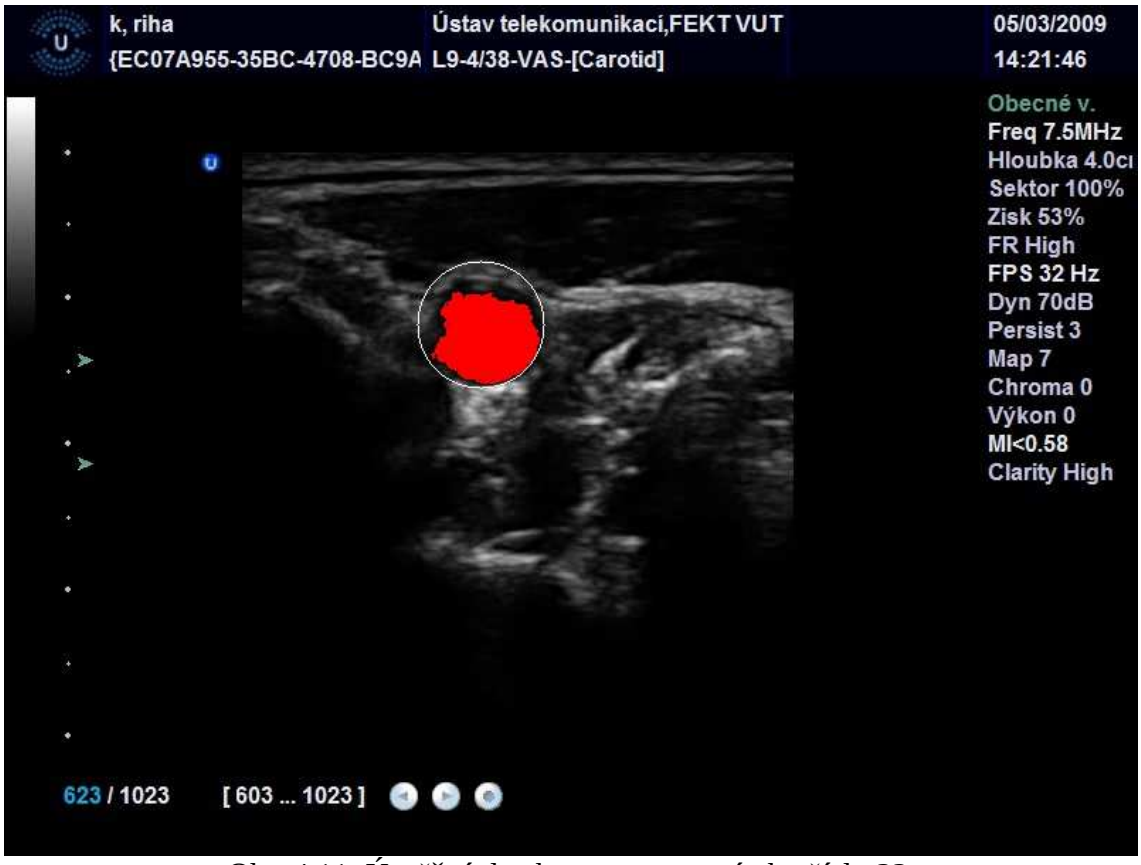
Pokud počet vyplněných bodů náleží do určitého intervalu, je detekce považována za správnou pokud do intervalu nenáleží, je detekce vyhodnocena jako nesprávná. Z toho jak funguje vyplňovaní funkce `cvFloodFill` (viz následující kapitola) plyne, že program vyhodnotí detekci jako správnou, pokud funkce `hough` vrátí bod, který se nachází uvnitř oblasti tepny. Interval počtu vyplněných bodů byl pro tepnu ve videosekvenci *19_tepu.avi* zjištěn experimentálně pro úspěšnou detekci je nutné, aby byl počet vyplněných bodů větší než 3000 a zároveň menší než 4500.

Jelikož princip metody nezaručuje, že bude vyhodnocení vždy správné, umožňuje program vizuální kontrolu výsledků. Snímky jsou postupně zobrazovány i s vyplněnou

oblastí. Je možné porovnat výsledek zobrazený na standardním výstupu s polohou vyplněné oblasti. Pokud se vyplněná oblast nenachází uvnitř tepny a přesto je na standardním výstupu zobrazena úspěšná detekce, došlo k chybě.

Na obrázku 4.11 je snímek číslo 22, ve kterém byla detekce úspěšná (oblast tepny je vyplněna). Obrázek 4.12 ukazuje chybnou detekci ve snímku 23 (je vyplněna mnohem menší oblast nalevo od tepny). Do každého snímku je mimo vyplněné oblasti vykreslena také detekovaná kružnice.

Červeně vykreslený obdélník ve snímcích znázorňuje oblast výřezu, se kterou program při detekci pracuje.



Obr. 4.11: Úspěšná detekce tepny ve snímku číslo 22.

Obr. 4.12: Chybná detekce ve snímku číslo 23.

4.2.2 Vyplňování oblasti tepny

Pro vyplňování oblasti tepny je použita funkce `cvFloodFill`. Tato funkce je obdobou vyplňovacích funkcí známých z běžných grafických programů.

Rozdíl u `cvFloodFill` je ten, že všechny sousední body nemusí být barevně stejné, aby byly vyplněny. Funkce `cvFloodFill` přijímá mimo jiné jako argumenty tzv. `upDiff` a `loDiff`. Pixel je obarven, pokud je jeho intenzita není menší než intenzita jeho obarveného souseda mínus `loDiff` a pokud jeho intenzita není větší než intenzita jeho obarveného souseda plus `upDiff`.

Pakliže je v argumentu `flags` použita konstanta `CV_FLOODFILL_FIXED_RANGE` potom pixel není porovnáván se svým sousedním bodem, ale se základním bodem pro vyplňování („semínkem“). Výsledkem vyplňování je vždy souvislá oblast. Jako další argument lze funkci předat datovou strukturu `cvConnectedComp`, do které funkce ukládá informace vyplněné oblasti (např. počet obarvených pixelů atd.)[6].

5 VÝSLEDKY

5.1 Výsledky obecné části

V této kapitole jsou demonstrovány jednotlivé fáze algoritmu na jeho výstupech. Jako vstupní obraz je použit ultrazvukový snímek *100mmHg.png*, který je na obrázku 5.1. Jedná se o ultrazvukový snímek kolagenové tepenní náhrady.

Těchto výsledků bylo dosaženo pro parametry:
velikost plovoucího okénka 40,
velikost Gaussovy matice 70x70,
Cannyho prahy 130,390,
velikost Sobelova konvolučního jádra 5x5,
Počet hledaných kružnic 20,
Minimální průměr kružnice 20 pixelů,
Maximální průměr kružnice 200 pixelů.

Na obrázku 5.2 je vstupní obraz převedený do stupňů šedi pomocí funkce `cvConvertColor`. Z obrázku je patrné, že upravený snímek se od vstupního příliš neliší, jelikož snímek vytvořený ultrazvukem již ve stupních šedi je. Upravena je tedy pouze oblast textu na okrajích snímku. I přesto je převedení do stupňů šedi nezbytné jelikož funkce `cvCanny`, která nalezne v obraze hrany, očekává jako vstup takto upravený obraz.
Další fází algoritmu je vyhlazení obrazu, to je provedeno pomocí funkce `cvSmooth`, která je, jak je již výše zmíněno, volána dvakrát za sebou. Tato funkce je použita pro odstranění šumu v obraze. Z obrázku 5.3 je vidět, že oblasti na okrajích snímku jsou oproti užitečnému obrazu průřezu tepenní náhradou výrazně potlačeny, to je velmi výhodné pro hranovou detekci. Funkce `cvCanny` nalezne hrany téměř výhradně ve střední části snímku. Výsledek detekce hran je znázorněn na obrázku 5.4, hranové body nalezené na okrajích snímku, nepředstavují výrazné rušení.
V této fázi algoritmu je možné aplikovat na hranový obraz funkci pro vytvoření akumulátoru potenciálních středů. Na obrázku 5.5 je tento akumulátor zobrazen. Na

snímku je patrné maximum uprostřed průřezu tepenní náhrady, což dokazuje, že metoda funguje správně.

Poslední fází algoritmu je vykreslení kružnic na obrázku 5.6 je vykresleno prvních pět nejpravděpodobnějších kružnic. Kružnice jsou vykresleny do vstupního obrazu. Zavedení normalizace (viz výše) zaručuje, že za nejpravděpodobnější nejsou považovány pouze kružnice detekované na vnějším okraji průřezu tepenní náhradou. Z výsledku lze konstatovat, že potenciální středy byly nalezeny s velkou přesností.

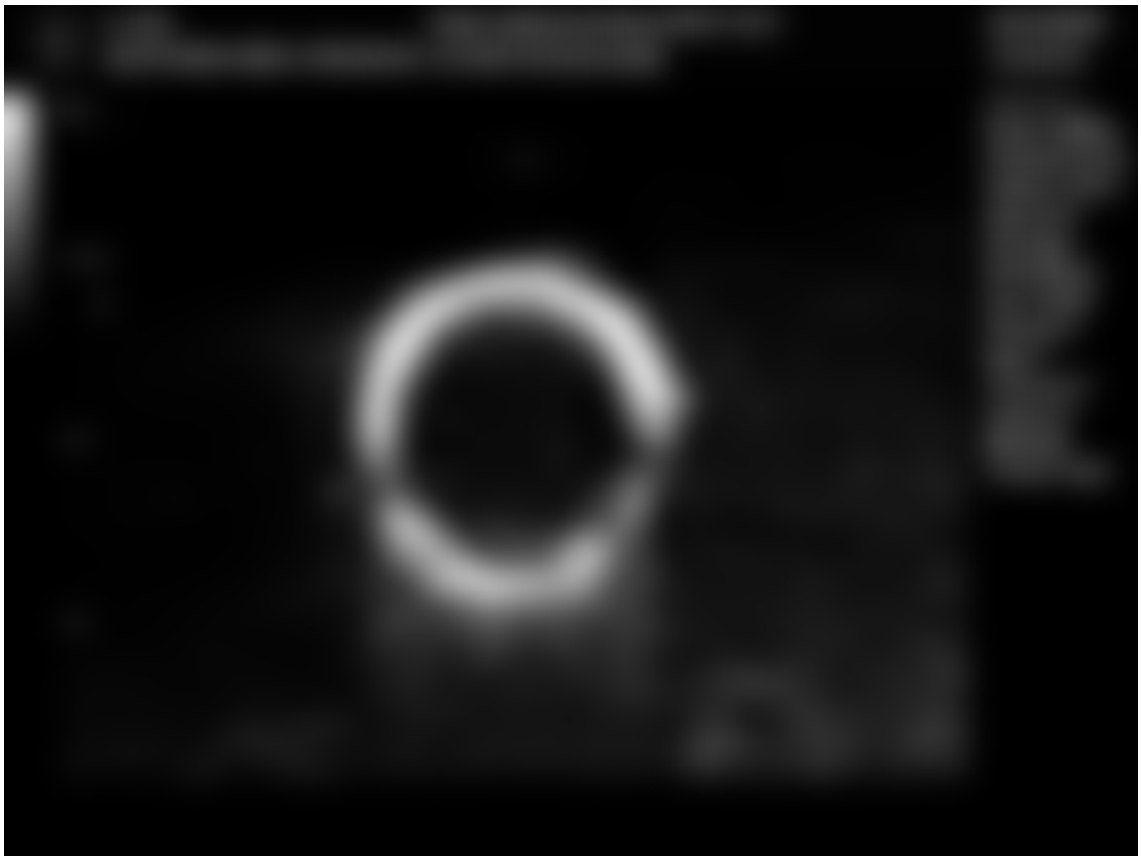
Na obrázku 5.7 je znovu vykresleno prvních pět nejpravděpodobnějších kružnic, tentokrát jsou ovšem vykresleny do hranového obrazu. Z tohoto zobrazení je patrné, že o pravděpodobnosti kružnice skutečně rozhoduje přesnost jejího tvaru. Zejména u větších kružnic je patrné, že se „chytily“ horní „kulatější“ části detekované hrany. Tato část obsahuje velké množství hranových bodů, které se nachází ve stejné vzdálenosti od nalezeného středu. Ve spodní části průřezu tepenní náhradou neodpovídá tvar detekované hrany kružnici, tudíž hranové body mají různou vzdálenost od detekovaného středu.

Obrázek 5.8 zobrazuje pět nejpravděpodobnějších kružnic při deaktivované normalizaci vykreslených do vstupního obrazu. Deaktivace normalizace způsobuje, že prvních pět nejpravděpodobnějších kružnic je detekováno na vnější hraně průřezu tepenní náhradou. Kružnice detekované na vnitřní hraně tepenní náhrady, jsou bez normalizace považovány za méně pravděpodobné.

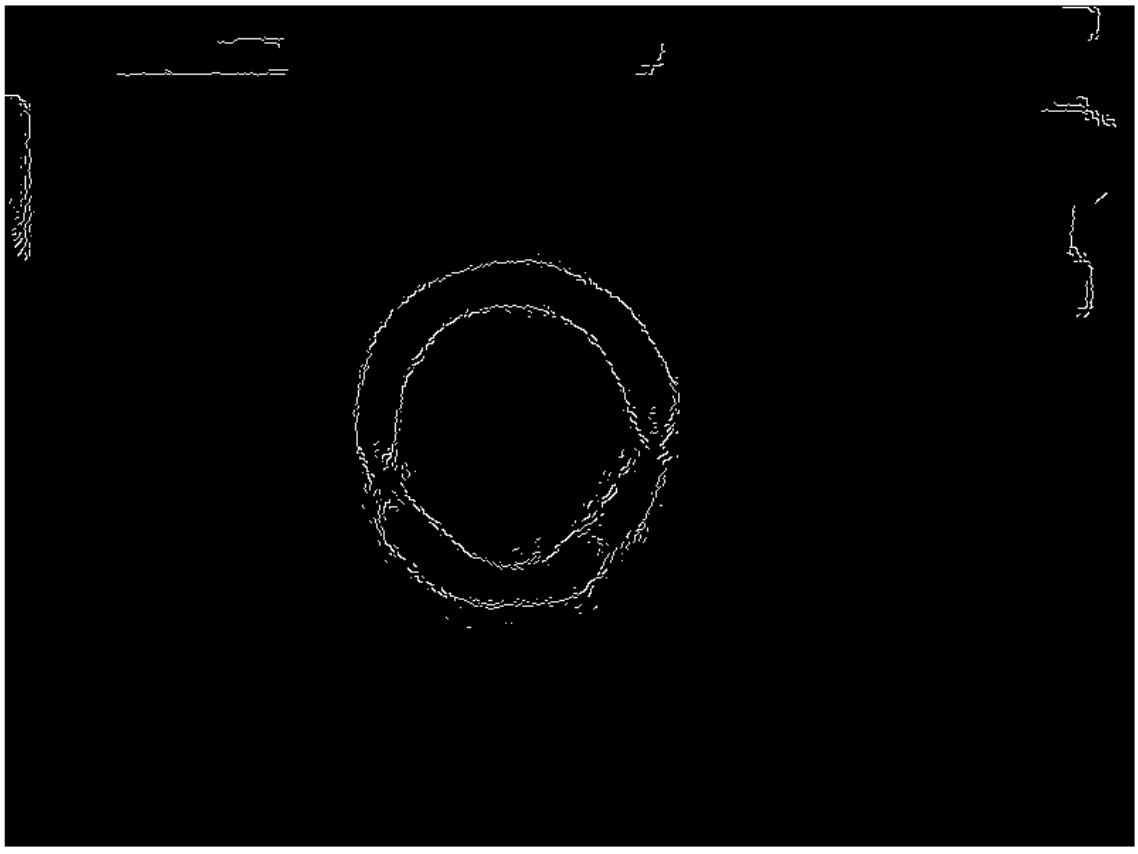
Na obrázku 5.9 je opět prvních pět nejpravděpodobnějších kružnic bez normalizace, tentokrát jsou však vykresleny do hranového obrazu.

Obr. 5.1: Vstupní obrázek.

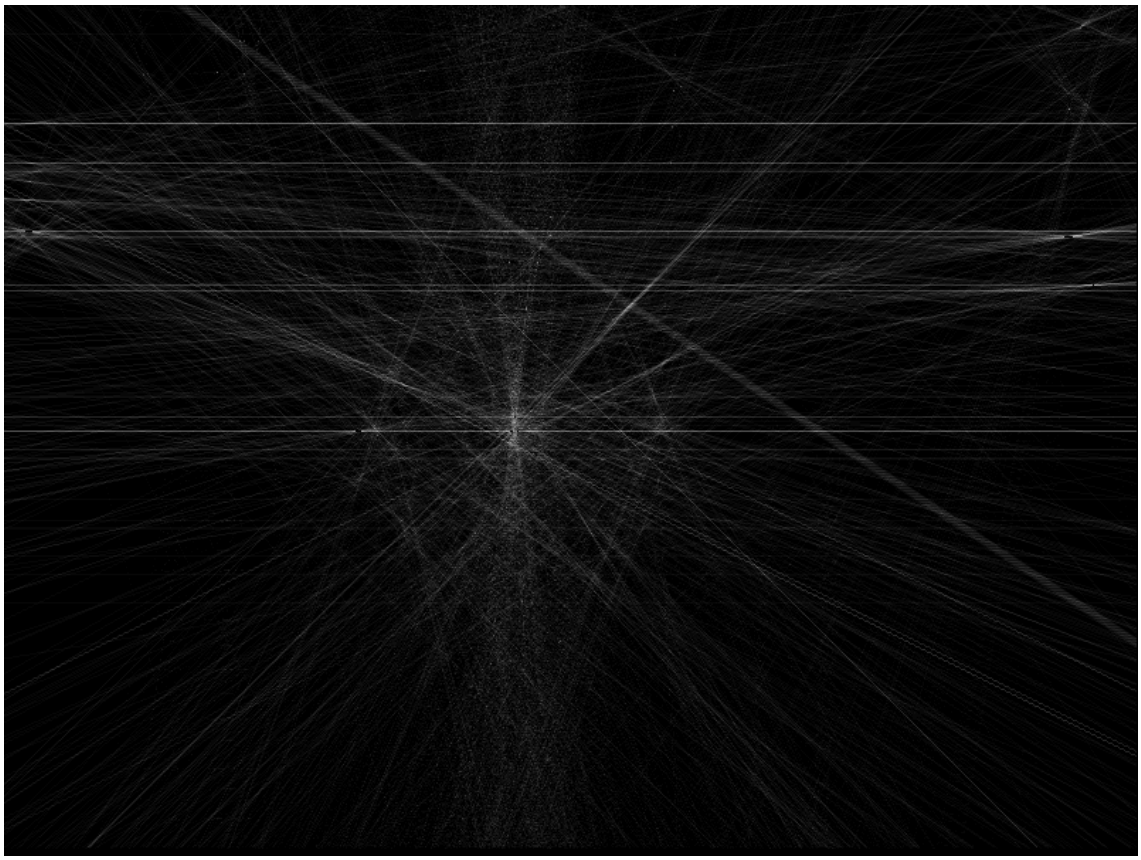
Obr. 5.2 Obrázek ve stupních šedi.



Obr. 5.3: Vyhlazený obrázek po Gaussově filtru.

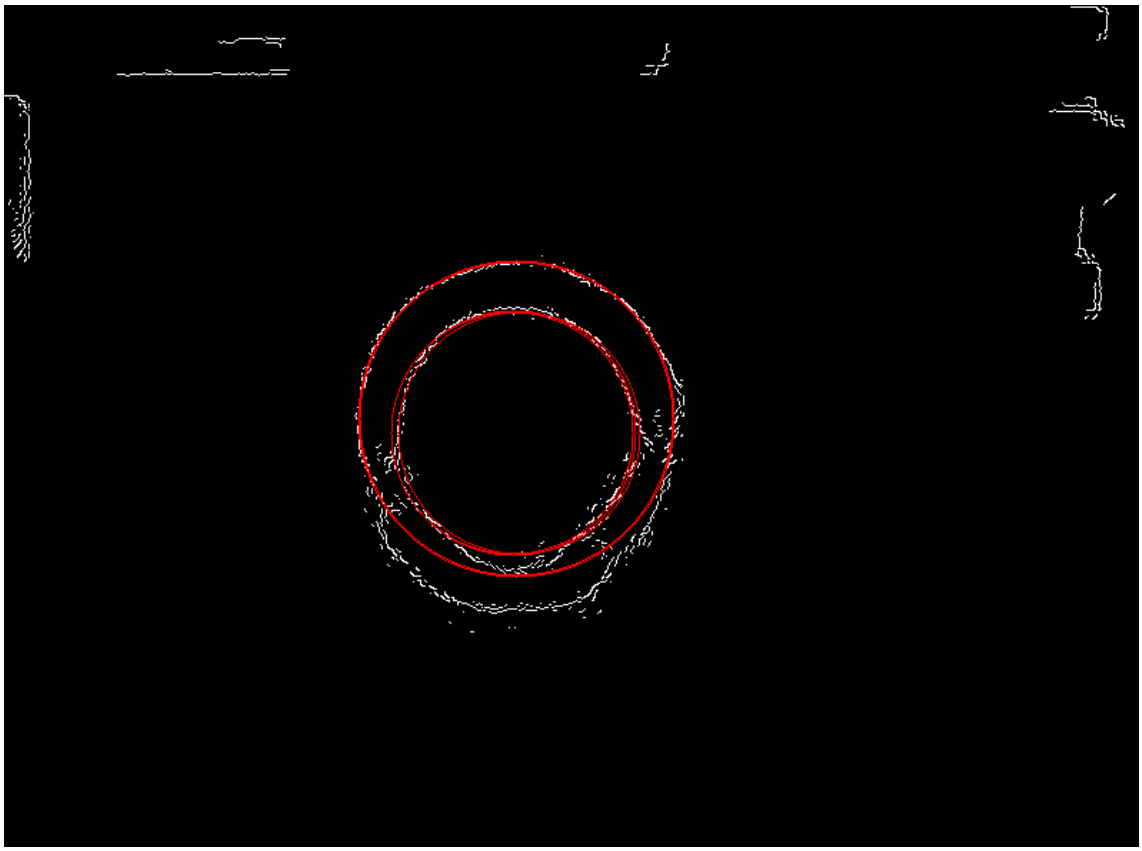


Obr. 5.4: Hranový obrázek výsledek Cannyho detektoru.



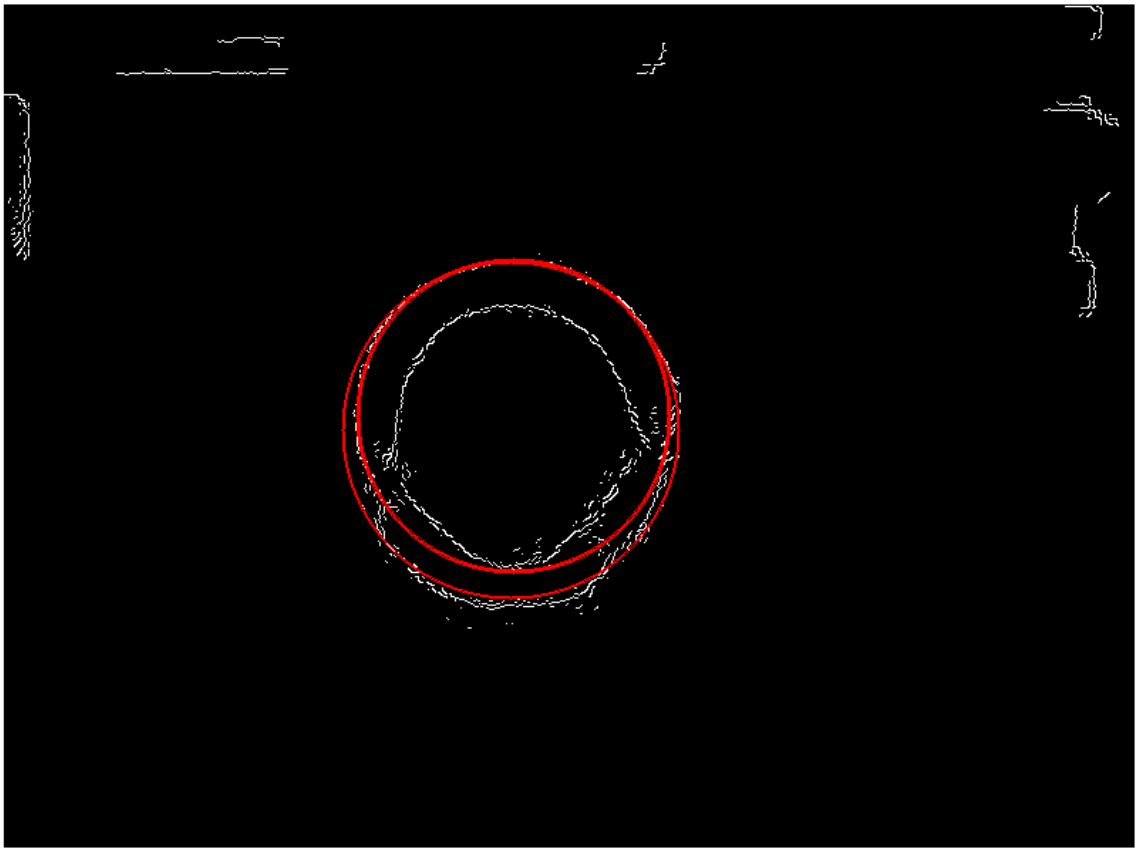
Obr. 5.5: Akumulátor pro nalezení potenciálních středů.

Obr. 5.6: Detekované kružnice ve výsledném obraze s normalizací.



Obr. 5.7: Detekované kružnice v hranovém obraze s normalizací.

Obr. 5.8: Detekované kružnice ve výsledném obraze bez normalizace.



Obr. 5.9: Detekované kružnice v hranovém obraze bez normalizace.

5.2 Výsledky části pro detekci tepny v ultrazvukové videosekvenci

V této části jsou prezentovány výsledky detekce tepny v ultrazvukové videosekvenci *19_tepu.avi*. Změnou parametrů algoritmu postupně dosahováno co nejlepší procentuální úspěšnosti detekce. Všechny snímky videa jsou testovány pro každou kombinaci parametrů.

Při testování je vždy jeden parametr měněn a ostatní zůstávají konstantní. Rozsah změny a velikost kroku je zvolena pro každý parametr zvlášť, vzhledem ke druhu měněného parametru. Po vyhodnocení testu je hodnota měněného parametru, pro kterou byla úspěšnost detekce největší, použita při testu dalších parametrů.

Testy jsou prováděny posupně pro změnu následujících parametrů algoritmu:

velikost plovoucího okénka,

počet porovnávaných kružnic,

velikost Cannyho prahů,

velikost Gaussovy matice.

Ostatní důležité parametry zůstaly po celou dobu konstantní a byly nastaveny na hodnoty uvedené v tabulce 5.1.

Tab. 5.1: Hodnoty parametrů, které zůstaly během testování konstantní.

Název parametru	Hodnota
Minimální poloměr hledané kružnice	25 pixelů
Maximální poloměr hledané kružnice	55 pixelů
Velikost Sobelovy konvoluční matice	5x5

Hodnoty všech měněných parametrů budou uvedeny v tabulce vždy u každého testu zvlášť.

5.2.1 Test parametru velikost plovoucího okénka

V této kapitole jsou prezentovány výsledky testu při změně parametru velikost plovoucího okénka. Funkce tohoto parametru je blíže popsána v kapitole 4.1.2.

Z testu vyplývá, že optimální hodnota velikosti plovoucího okénka pro snímky dané videosekvence je **10 pixelů** při této velikosti okénka se podařilo dosáhnout úspěšnosti detekce **80%**.

Sloupec s označením „norm“ v tabulce obsahuje hodnoty úspěšnosti detekce při zapnuté normalizaci. Princip normalizace je popsán v kapitole 4.2.4. Jak je vidět z tabulky 5.3 a grafu na obrázku 5.10, normalizace výrazně snižuje procento úspěšných detekcí. Kružnice detekovaná na hraně tepny je v poměru ke kružnicím detekovaným v zašumělých oblastech větší, proto je výhodné normalizaci deaktivovat a tím větší kružnici upřednostnit. Z těchto důvodů je od normalizace při dalším testování upuštěno.

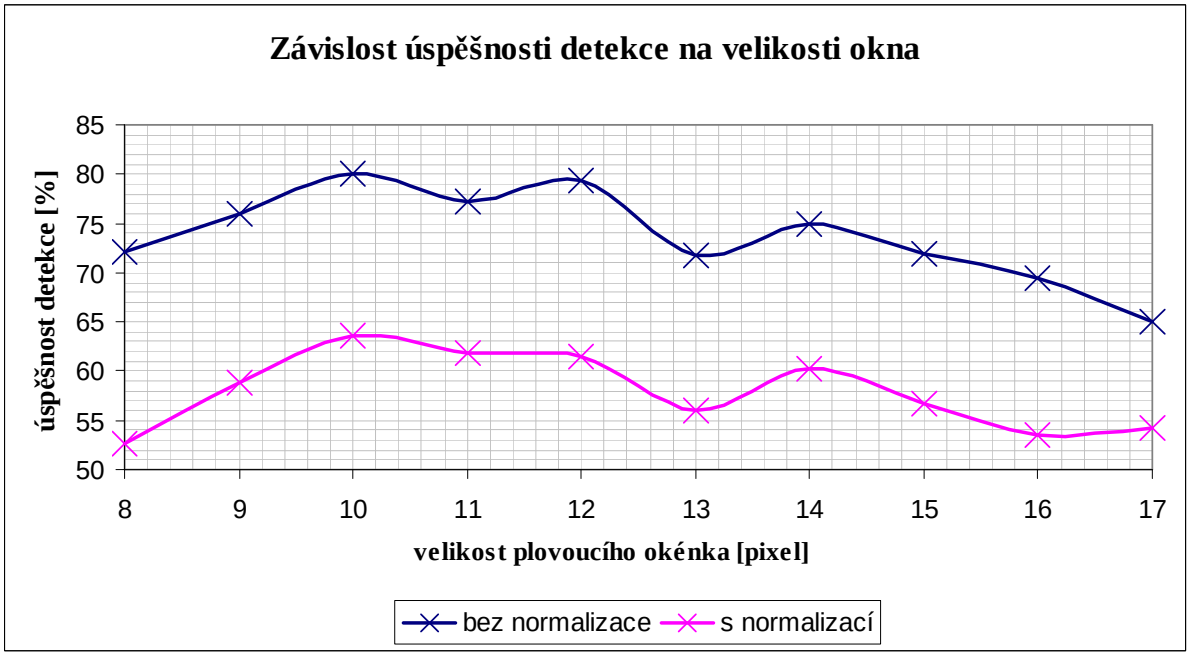
Byl proveden také test, při kterém byla z algoritmu odstraněna funkce pro ořezávání okrajových oblastí snímku. V tomto případě byla detekce tepny úspěšná na pouhých 15%, proto je od dalšího testování s takto upraveným algoritmem také upuštěno.

Tab. 5.2: Hodnoty ostatních parametrů při změně velikosti plovoucího okénka.

Název parametru	**Hodnota**
Počet porovnávaných kružnic	35
Cannyho prahy	T_1=150, T_2=450
Velikost Gaussovy matice	30x30

Tab. 5.3: Výsledky testů při změně hodnoty plovoucího okénka.

Velikost pl.okénka [pixel]	**Úspěšnost detekce [%]**	**Úspěšnost detekce (norm) [%]**
8	72,14286	52,61905
9	75,95238	58,80952
10	80	63,57143
11	77,14286	61,90476
12	79,28571	61,42857
13	71,66667	55,95238
14	75	60,2381
15	71,90476	56,66667
16	69,52381	53,57143
17	65	54,28571



Obr. 5.10: Závislost úspěšnosti detekce na velikosti plovoucího okénka.

5.2.2 Test parametru počet porovnávaných kružnic

Parametr počet porovnávaných kružnic není v textu blíže specifikován, proto je stručně popsán zde. Tento parametr ve skutečnosti určuje, kolik potenciálních středů bude vyčteno z akumulátoru pro potenciální středy. Program postupně prochází jednotlivé potenciální středy a pro každý hranový bod vypočítá vzdálenost od daného středu, tedy poloměr potenciální kružnice. Pokud tento poloměr náleží do rozsahu povolených poloměrů, je v datovém akumulátoru inkrementováno pole odpovídající danému středu a poloměru. Střed a poloměr s největším počtem inkrementací, tedy nejvyšší hodnotou akumulátoru, popisuje nejpravděpodobnější kružnici.

Velikost tohoto parametru výrazně ovlivňuje délku běhu programu.

Z grafu na obrázku 5.11 je patrné, že procentuální úspěšnost detekce s počtem porovnávaných kružnic stále stoupá až k hodnotě 50 porovnávaných kružnic, poté opět klesne.

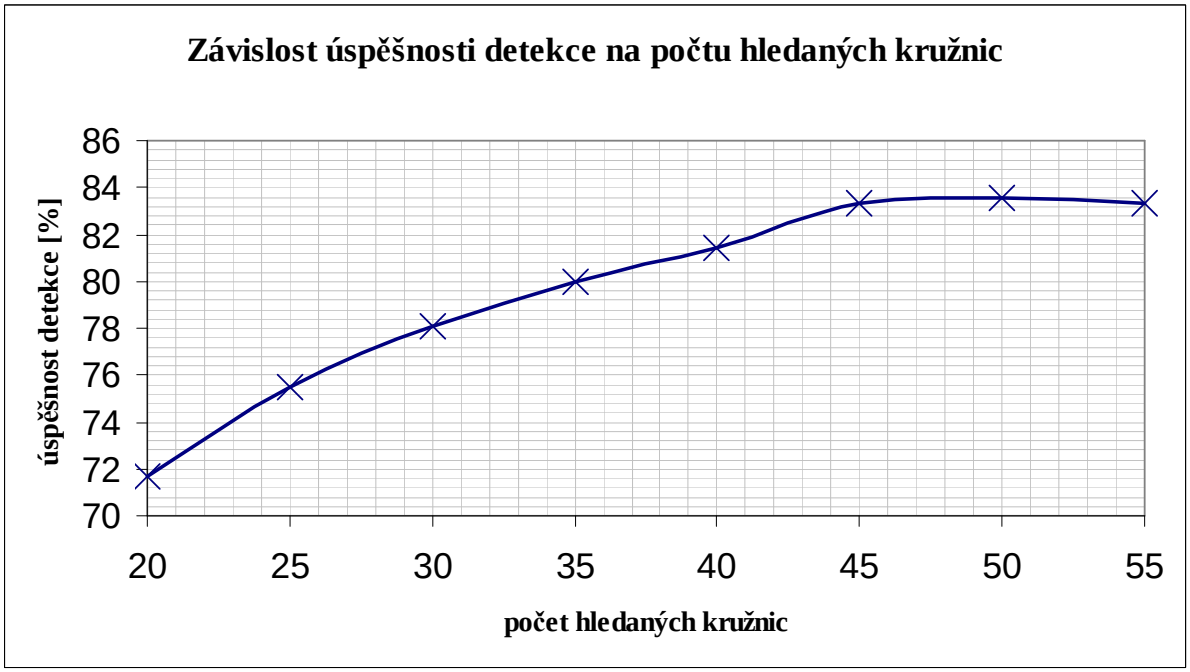
Pro hodnotu **50 porovnávaných kružnic** byla tepna úspěšně detekována u **83,57143 %** snímků.

Tab. 5.4: Hodnoty ostatních parametrů při změně počtu porovnávaných kružnic.

Název parametru	Hodnota
Velikost plovoucího okénka	10 pixelů
Cannyho prahy	T_1=150, T_2=450
Velikost Gaussovy matice	30x30

Tab. 5.5: Výsledky testů při změně počtu porovnávaných kružnic.

Počet pr. kružnic	Úspěšnost detekce [%]
20	71,66667
25	75,47619
30	78,09524
35	80
40	81,42857
45	83,33333
50	83,57143
55	83,33333



Obr. 5.11: Závislost úspěšnosti detekce na počtu porovnávaných kružnic.

5.2.3 Test parametru Cannyho práh

Parametr Cannyho práh představuje nižší z dvojice prahů (viz kapitola 3.3), druhý práh je algoritmem automaticky nastaven na hodnotu třikrát vyšší.

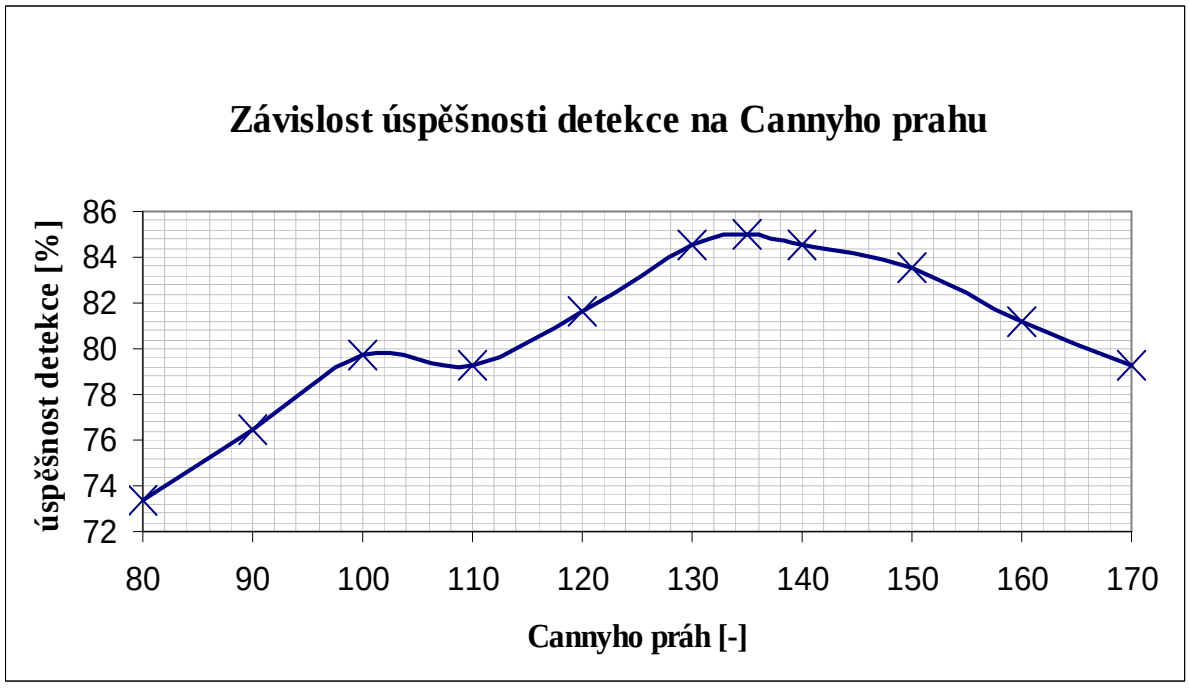
Jako optimální hodnotu **Cannyho prahu** lze na základě tohoto testu zvolit **135**, tudíž hodnota vyššího z prahů je optimálně nastavena na hodnotu **405** (třikrát vyšší). Při tomto nastavení bylo dosaženo úspěšnosti detekce **85 %**.

Tab. 5.6: Hodnoty ostatních parametrů při změně velikosti Cannyho prahů.

Název parametru	Hodnota
Velikost plovoucího okénka	10 pixelů
Počet porovnávaných kružnic	50
Velikost Gaussovy matice	30x30

Tab. 5.7: Výsledky testů při změně velikosti Cannyho prahů.

Velikost Cannyho prahu T_1 [-]	Úspěšnost detekce [%]
80	73,33333
90	76,42857
100	79,7619
110	79,28571
120	81,66667
130	84,52381
135	85
140	84,52381
150	83,57143
160	81,19048
170	79,28517



Obr. 5.12: Závislost úspěšnosti detekce na Cannyho prazích.

5.2.4 Test parametru velikost Gaussovy matice

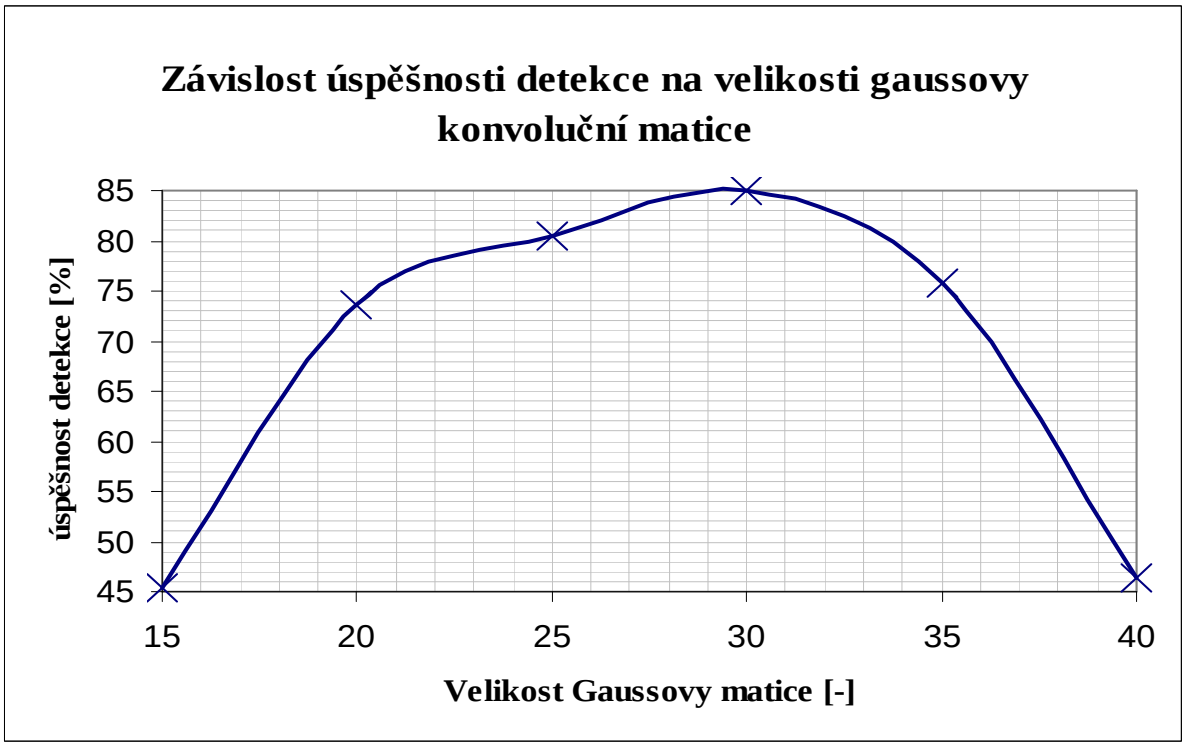
Význam parametru, který je testován v této kapitole, je vysvětlen v kapitole 3.2. V praxi platí, že čím vetší je Gausova matice, tím méně ostrých hran je v obraze detekováno. Při nastavení příliš velké hodnoty, jsou rozmazány i ostré hrany.
Z testu vyplývá, že ideální velikost Gaussovy matice je **30x30**. Pro tuto hodnotu byla tepna ve snímcích videa detekována nejčastěji, a to v **85 %**.

Tab. 5.8: Hodnoty ostatních parametrů při změně velikosti Gaussovy matice.

Název parametru	**Hodnota**
Velikost plovoucího okénka	10 pixelů
Počet porovnávaných kružnic	50
Cannyho práhy	T_1=135, T_2=405

Tab. 5.9: Výsledky testů při změně velikosti Gaussovy matice.

Velikost Gaussovy matice [-]	**Úspěšnost detekce [%]**
15	45,47619
20	73,57143
25	80,47619
30	85
35	75,71429
40	46,42857



Obr. 5.13: Závislost úspěšnosti detekce na velikosti Gaussovy matice.

5.2.5 Zhodnocení výsledků

Ze série výše popsaných testů vyplývá, že největší úspěšnosti detekce bylo dosaženo pro hodnoty parametrů uvedené v tabulce 5.10.

Tab. 5.10: Optimální parametry detekce tepny ve snímcích videosekvence *19_tepu.avi*.

Název parametru	**Hodnota**
Velikost plovoucího okénka	10 pixelů
Počet porovnávaných kružnic	50
Cannyho prahy	T_1=135, T_2=405
Velikost Gaussovy matice	30x30

Pro tyto parametry byla detekce tepny úspěšná u **85 %** snímků videosekvence.

6 ZÁVĚR

Nastudoval jsem princip Houghovy transformace a seznámil se s možnostmi zpracování obrazu v knihovně OpenCV. Jedním z bodů zadání této práce je implementace algoritmu se zaměřením na zvýšení jeho efektivity, toho je dosaženo zavedením metody pro nalezení potenciálních středů (viz kapitola 4.2) [2]. Na základě získaných teoretických poznatků byl vytvořen program, který je schopen detekovat kružnice v libovolném dvourozměrném obraze. Od vytvoření grafického uživatelského rozhraní bylo po dohodě s vedoucím práce upuštěno ve prospěch rozšíření zadání. Program tedy nabízí možnost volby parametrů pouze přes vývojové prostředí.

Zadání bylo rozšířeno o část týkající se využití algoritmu Houghovy transformace pro detekci tepny ve snímcích ultrazvukové videosekvence. Testy byly prováděny na videosekvenci *19_tepu.avi*, která obsahuje 421 snímků. Program dokázal tepnu správně detekovat u 85% z nich, což odpovídá 357 úspěšným detekcím.

LITERATURA

[1] KYEWOOK, L. *Application of the Hough Transform.* Lowell: University of Massachusetts, 2006. Dostupné na URL: http://www.cs.uml.edu/~lkyewook/hough/APPLICATION%20OF%20THE%20HOUGH%20TRANSFORM%20V1.9.pdf

[2] IOANNOU, D.; HUDA, W.; LAINE, F. A.: *Circle Recognition Through A 2D Hough Transform And Radius Histogramming. In Image and Vision Computing Vol-17*, Elsevier Science B. V., DOI: 10.1016/S0262-8856(98)00090-0.

[3] MATTHEWS, J. *An introduction to edge detection: The Sobel edge detektor*. Gebration5 [online]. 2002. [cit. 2008-12-18].
Dostupný na URL: http://www.generation5.org/content/2002/im01.asp

[4] GREEN, B. *Canny Edge Detection Tutorial*. Drexel University [online]. 2002 [cit. 2008-12-18].
Dostupný z WWW:<http://www.pages.drexel.edu/~weg22/can_tut.html >.

[5] INTEL COSPORATION. *Open Source Computer Vision Library : Reference Manual*. 2000. Dostupný na URL: www.cs.unc.edu/Research/stc/FAQs/OpenCV/OpenCVReferenceManual.pdf

[6] BRADSKY, G.; KAEBLER, A.: Learning *OpenCv Computer Vision with OpenCv Library*. 2008. Dostupné na URL: http://my.safaribooksonline.com/9780596516130

SEZNAM POUŽITÝCH SYMBOLŮ

x	vzdálenost od počátku na horizontální ose
y	vzdálenost od počátku na vertikální ose
m	směrnice přímky
b	úsek vyťatý přímkou na ose y
r_n	délka normály
Θ	úhel mezi normálou a osou x
r	poloměr kružnice
x_0	x-ová souřadnice potenciálního středu
y_0	y-ová souřadnice potenciálního středu
Y	jasová složka digitálního obrazu
R	červená složka digitálního obrazu
G	zelená složka digitálního obrazu
B	modrá složka digitálního obrazu
$G(x,y)$	prvek Gausovy konvouluční matice o souřadnicích x,y
G(x)	Sobelův konvoluční operátor pro horizontální směr gradientu
G(y)	Sobelův konvoluční operátor pro vertikální směr gradientu
σ	standardní odchylka Gaussovy distribuce
O	střed kružnice
c	kružnice
p	přímka kolmá na tětivu kružnice procházející jejím středem
A	bod v dvourozměrném prostoru
x_A	x-ová souřadnice bodu A
y_A	y-ová souřadnice bodu A
B	bod v dvourozměrném prostoru
x_B	x-ová souřadnice bodu B
y_B	y-ová souřadnice bodu B
u	směrový vektor úsečky AB
x_u	x-ová souřadnice směrového vektoru u
y_u	y-ová souřadnice směrového vektoru u
p	středový bod úsečky AB
x_p	x-ová souřadnice bodu p
y_p	y-ová souřadnice bodu p

SEZNAM PŘÍLOH

A Obsah přiloženého CD

Adresář:	**Popis obsahu:**
Bakalářská práce	V adresáři se nachází dokument *Bakalářská práce.pdf*, který obsahuje text bakalářské práce.
Houghova transformace	Adresář obsahuje soubory a složky projektu konsolové aplikace pro detekci kružnic ve statickém obraze, vytvořeného ve Visual Studiu 2008.
Analýza videosekvence	Obsahuje projekt konsolové aplikace pro testování úspěšnosti detekce tepny v ultrazvukové videosekvenci. Projekt je vytvořen ve Visual Studiu 2008.
OpenCv	Obsahuje instalační soubor knihovny OpenCv, *OpenCV_1.0.exe*.

B Návod na použití programu Houghova transformace

Ve Visual Studiu 2008 se projekt otevírá souborem *Hough.sln*, který se nachází v adresáři *Houghova transformace* na CD. Zdrojový soubor programu se jmenuje *Hough.cpp* a nachází se v adresáři *Hough*. Adresář *Hough* se také nachází ve složce *Houghova transformace*. Po otevření projektu respektive zdrojového souboru *Hough.cpp* ve vývojovém prostředí je nutné nejprve nastavit cestu ke vstupnímu souboru, to se provádí inicializací proměnné `inputFile`, která je uvedena na prvním řádku programu pod „includy“ standardních knihoven. Cesta je přednastavena na soubor *100_mmHg.png*, který se nachází ve adresáři *srovnani*.

Na dalším řádku programu lze nastavit přepínač pro zobrazování jednotlivých fází algoritmu. Přepínač ve formě logické proměnné s názvem `zobrazuj` je přednastaven na hodnotu `true`, což znamená, že program zobrazí grafické výstupy všech fází algoritmu.

Další logická proměnná s názvem `norm` slouží k vypínání a zapínání normalizace.

Na dalších řádcích zdrojového kódu se nacházejí konstanty, které představují důležité parametry algoritmu. Významy jednotlivých konstant jsou uvedeny v komentářích ve zdrojovém kódu.

Po nastavení cesty ke vstupnímu obrazu a všech důležitých parametrů algoritmu je možné program spustit. Po spuštění se zobrazí zdrojový obraz (Source image) a také hranový obraz (Edges). Pomocí posuvníků nad hranovým obrazem lze upravit rozměr Gaussovy matice a velikost nižšího z Cannyho prahů (vyšší práh je vždy třikrát větší). Změna těchto parametrů ovlivní počet a podobu hran detekovaných ve vstupním obraze. Jakmile je hranový obraz podle našich představ, můžeme provést výběr oblasti zájmu. Při stisku klávesy Enter bez provedení výběru je automaticky vybrán celý vstupní obraz. Výběr části obrazu se provádí klasickým způsobem, stisknutím, tažením a uvolněním levého tlačítka myši. Vybraný obdélník je možné zrušit stisknutím klávesy Escape. Stisknutím klávesy Enter je vybraná oblast odeslána funkci pro nalezení kružnic.

Po provedení výpočtů program zobrazí nejpravděpodobnější kružnici nalezenou ve vybrané oblasti a vykreslí ji, jak do hranového obrazu (Result_edge), tak do kopie původního obrazu (Result). Nyní je možné posuvníkem s označením Index zobrazovat kružnice, v pořadí podle jejich pravděpodobnosti.

Poloměr a souřadnice středu aktuálně zobrazené kružnice se vypisují na standardní výstup.

Stisknutím klávesy „p“ lze uložit informace o aktuálně zobrazené kružnici do souboru, který je definován proměnnou `saveFile`. Inicializace této proměnné je v kódu umístěna pod konstantami pro nastavení parametrů algoritmu. Pokud soubor pro ukládání neexistuje, program ho vytvoří. Informace o kružnici se do souboru uloží ve formátu:

Název Vstupního Souboru

index.stred(x,y),radius:hodnota poloměru.

Program lze ukončit stisknutím klávesy Escape.

C Návod na použití programu Analýza videosekvence

Ve Visual Studiu 2008 se projekt otevírá souborem *video.sln*, který se nachází v adresáři *Analýza videosekvence* na CD. Zdrojový soubor programu se jmenuje *video.cpp* a nachází se v adresáři *video*. Adresář *video* se také nachází ve složce *Analýza videsekvence*. Po otevření projektu respektive zdrojového souboru *video.cpp* ve vývojovém prostředí je nutné nejprve nastavit cestu ke vstupnímu souboru to se provádí inicializací proměnné `sourceFile`, která je uvedena na prvním řádku programu pod „includem“ knihovny `hough.h`. Cesta je přednastavena na soubor *19_tepu.avi*.

Na dalším řádku lze nastavit cestu k souboru, do kterého se budou ukládat výsledky testů pro jednotlivé snímky. Cesta k souboru se nastavuje inicializací proměnné `saveImage`.

Parametry algoritmu lze nastavovat v souboru *funkce.h*, který se nachází ve složce *video*. Konstanty představující hodnoty parametrů jsou opět uvedeny na začátku souboru pod „includy“ standardních knihoven. Pod nastavením parametrů je opět logická proměnná `norm`, která slouží pro aktivaci, či deaktivaci normalizace.

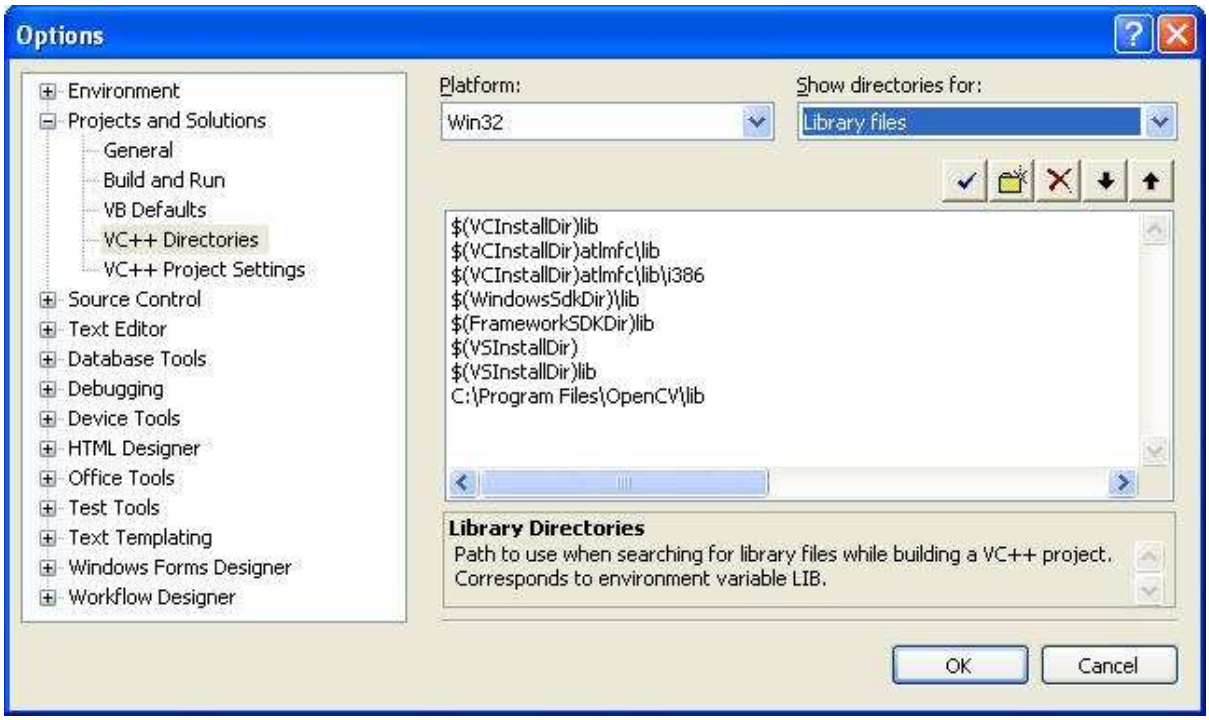
Zdrojový soubor obsahující definice funkcí má název *funkce.cpp* a nachází se také ve složce video.

Po nastavení parametrů algoritmu je možné program spustit. Program lze ukončit stisknutím klávesy Escape.

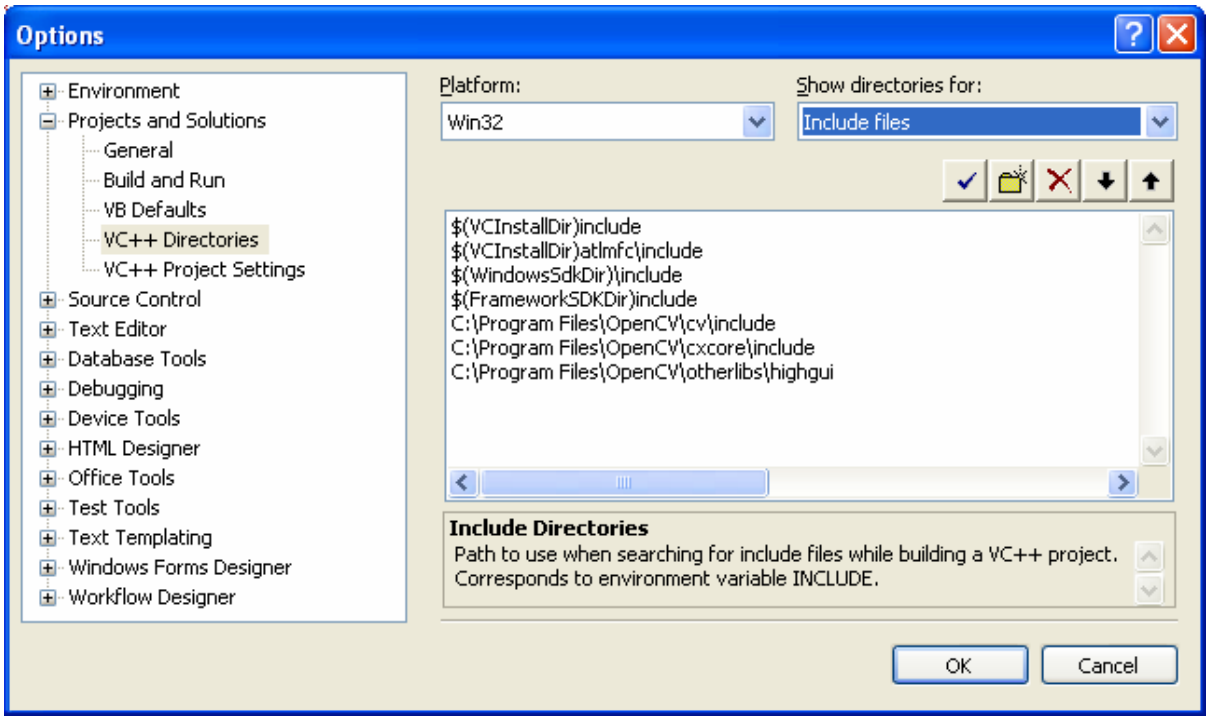
D Nastavení Visual Studia 2008 pro použití knihoven OpenCv

Instalační soubor je uložen na přiloženém CD ve složce *OpenCv*.
V menu *Tools\Options\Projects and Solutions\VC++ Directories* je nutné nastavit cestu k nainstalovaným knihovním souborům. Vpravo nahoře je nutné zvolit *Library files*. Nastavení pro standardní instalační adresář je na obrázku D.1.
Obdobně pro *Include files* viz obrázek D.2.

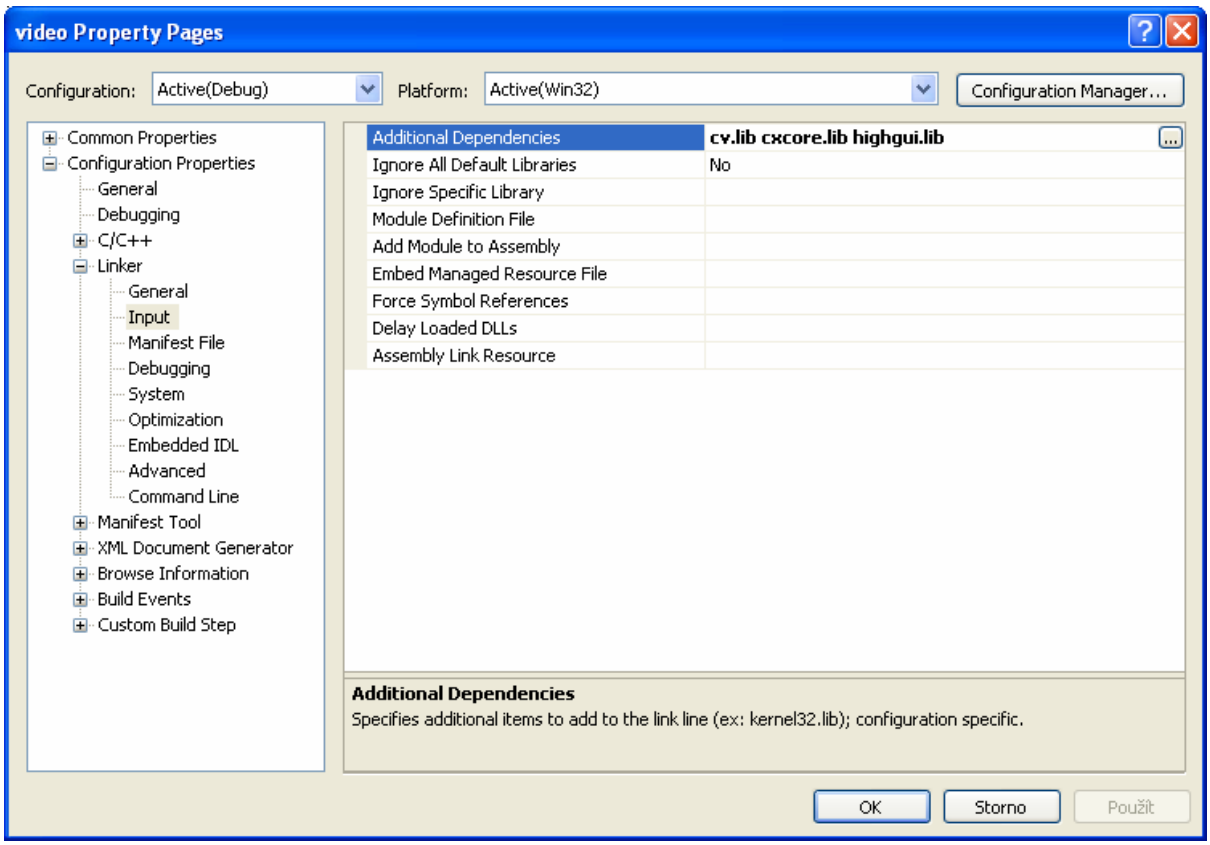
Dále je nutné **pro každý projekt** do menu *Projects\project Properties\Configuration Properties\Linker\Input\Additional Dependencies* přidat položky **cv.lib, cxcore.lib** a **highgui.lib**, jak je znázorněno na obrázku D.3 (v mých projektech již nastaveno).



Obr. D.1:Nastavení cesty ke knihovním souborům



Obr. D.2:Nastavení *include files*



Obr. D.3:Nastavení *Aditional Dependencies* ve Visual Studiu 2008