

# Detekce středu rotace

Ondrej Valo <xvaloo00@stud.fit.vutbr.cz>

23. júla 2024

## 1 Zadanie

Implementácia funkcie, ktorá dokáže detekovať stred rotácie pozorovaného objektu, sa zameriava na identifikáciu stredu otáčania objektu v obraze. Tento proces sa vykonáva na základe snímky, na ktorej sú viditeľné kruhy spôsobené rotáciou objektu okolo osi stolíka.

Vzhľadom na to, že vstupné obrázky môžu byť rôzne, musí funkcia byť schopná riešiť aj neideálne situácie, ako je napríklad prípad, keď sa stred kruhov nachádza mimo obrázka. Toto zadanie môžeme rozdeliť na 4 pod úlohy:

- Predspracovanie obrázka
- Detekcia kruhov.
- Výpočet stredu rotácie
- Riešenie neideálnych situácií

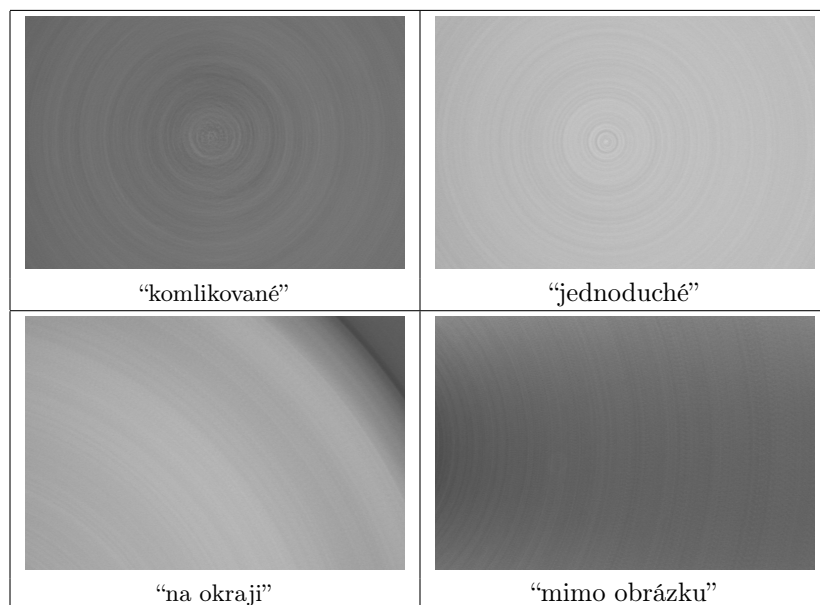
## 2 Existujúce metódy

- Hough Transform: Ide o populárnu metódu detekcie kruhov na obrázkoch. Funguje tak, že sa priestor obrazu prevedie na priestor parametrov, kde každý kruh na obrázku zodpovedá bodu v priestore parametrov. Houghova transformácia potom hľadá vrcholy v priestore parametrov, ktoré zodpovedajú kruhom na obrázku. [Ped07]
- Fourier Descriptor: V prípade hľadania stredu kruhu sa Fourierov deskriptor používa na vypočítanie frekvenčného spektra obrysu kruhu. Tento spektrum obsahuje informácie o polomere a polohe stredu kruhu. Na základe týchto informácií je potom možné vypočítať polomer a polohu stredu kruhu.. [SW11]
- Template Matching: Táto metóda zahŕňa vytvorenie šablóny kruhu, ktorý chceme zistiť, a jej porovnanie s obrázkom pomocou miery podobnosti. Táto metóda je výpočtovo nákladná a nemusí fungovať dobre, ak kruh nie je dokonale kruhový alebo ak sa menia svetelné podmienky. [Yan10]
- Edge-based methods: Ďalším prístupom k detekcii kruhov je najprv detekcia hrán kruhov pomocou algoritmov detekcie hrán, ako je Cannyho detektor hrán alebo Marr-Hildreth detektor hrán. Po detekcii hrán je možné použiť algoritmus na nájdenie stredu kruhu.[NSH08]

Existuje mnoho ďalších metód ktoré niesu vymenované a taktiež ich možné kombinácie keďže sa úloha dá rozdeliť na pod úlohy pre ktoré existujú rôzne riešenia.

## 3 Dátové sady

Je poskytnutá v rámci zadania, delí sa na 4 časti ktoré sú:



## 4 Návrh riešenia

- Predspracovanie obrázka: Predtým, ako sa budú hľadať kruhy, je vhodné obrázok pred spracovať. Prvým krokom môže byť aplikácia filtra na zníženie šumu a vylepšenie kvality obrázka. Následne by sa mohli použiť rôzne techniky aby sa kruhy stali ešte viac viditeľnými.
- Detekcia kruhov: Po pred spracovaní obrázka sa môžu hľadať kruhy pomocou algoritmu Houghovej transformácie. Tento algoritmus dokáže identifikovať kruhové tvary v obraze a určiť ich stredy a polomery. V prípade, že nie sú kruhy jasne viditeľné, môžu sa použiť aj iné techniky, ako napríklad detekcia okrajov alebo segmentácia objektov.
- Výpočet stredy rotácie: Keďže objekt sa otáča okolo osi stolíka, stred rotácie by mal byť v jeho strede. Po detekcii kruhov môžeme použiť ich stredy na určenie približnej polohy streda rotácie. Tento stred by sa potom mohol upresniť pomocou ďalších techník, ako napríklad výpočet centroidu alebo hľadanie stredov kruhov s najväčším polomerom, ktoré by mali byť bližšie k stredovému bodu.
- Riešenie neideálnych situácií: V prípade, že sa stred kruhov nachádza mimo obrázka, môže byť nutné aplikovať iné techniky na určenie streda rotácie. Jednou z možností môže byť +ze na výpočet stredy kružnice nepotrebuje celú kružnicu ale len jej časť.

## 5 Využité technológie

V tejto sekcii sú uvedené metódy a technológie, ktoré boli využité pri riešení tohto projektu, vrátane popisu ich funkcií, vlastností a spôsobu využitia pri rôznych výpočtoch a operáciách.

### 5.1 Histogram orientovaných gradientov

Histogram of Oriented Gradients (HOG) je metóda na extrahovanie príznakov z obrázkov. HOG pracuje tak, že najprv sa obrázok rozdelí na malé časti nazývané bunky. Potom sa pre každú bunku vypočíta gradient v každom pixli v bunke pomocou konvolučného jadra. Gradient sa potom vektorizuje a vektor sa normalizuje.

Keď máme normalizované gradienty v každej bunke, môžeme vypočítať histogram orientovaných gradientov. Histogram pozostáva z niekoľkých binov, pričom každý bin zodpovedá určitému rozsahu orientácie.

Ak označíme  $G_{x,y}$  a  $G_{y,x}$  gradienty v  $x$  a  $y$  smeroch pre pixel  $(x, y)$ , potom môžeme definovať gradient  $G$  a orientáciu  $\theta$  ako:

$$G = \sqrt{G_{x,y}^2 + G_{y,x}^2} \theta = \tan^{-1} \frac{G_{y,x}}{G_{x,y}}$$

Potom pre každú bunku môžeme vytvoriť histogram orientovaných gradientov ako:

$$H_i = \sum_{j \in B_i} w_j G_j(\theta_j - \theta_i)$$

kde  $B_i$  je množina pixelov v bunke  $i$ ,  $w_j$  je váha pre pixel  $j$ ,  $G_j$  je normalizovaný gradient pre pixel  $j$  a  $\theta_j$  je orientácia pre pixel  $j$ .

Výsledné histogramy z každej bunky sa potom spoja do jedného vektora príznakov.

## 5.2 Canny edge detection

Canny Edge Detection je algoritmus na detekciu hrán v obraze. Algoritmus pozostáva z niekoľkých krokov.

### 5.2.1 Gaussian Blur

Prvým krokom algoritmu je aplikácia Gaussovho filtra na obrázok. Gaussov filter znižuje šum v obraze a vyhladzuje hrany. Gaussov filter môžeme definovať ako:

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

### 5.2.2 Gradientsy a Magnitúda

Po aplikácii Gaussovho filtra sa vypočítajú gradienty v každom pixeli obrázka. Pre výpočet gradientov sa zvyčajne používa Sobelov operátor, ktorý sa dá definovať ako:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * IG_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I$$

kde  $*$  predstavuje konvolučný operátor a  $I$  je vstupný obrázok.

Potom sa vypočíta magnitúda gradientu  $M$  ako:

$$M(x, y) = \sqrt{G_x^2(x, y) + G_y^2(x, y)}$$

### 5.2.3 Non-maximum Suppression

Po výpočte magnitúdy sa použije algoritmus non-maximum suppression, ktorý slúži na zúženie hrán na jednopixelovú hrúbku a na elimináciu falošných hrán. Algoritmus prechádza každý pixel v obraze a porovnáva magnitúdy susedných pixelov v smere gradientu. Ak je magnitúda aktuálneho pixelu najväčšia zo susedov v smere gradientu, pixel sa ponechá, inak sa nastaví na hodnotu nula.

### 5.2.4 Hysteresis Thresholding

Posledným krokom algoritmu je aplikácia hysteresis thresholding. Tento krok sa používa na odstránenie slabých hrán a ponechanie iba silných hrán. Hysteresis thresholding pracuje tak, že vstupný obrázok sa rozdelí na dve kategórie hrán: silné a slabé hrany. Silné hrany sú tie, ktoré majú magnitúdu gradientu vyššiu ako určitý práh. Slabé hrany majú magnitúdu gradientu nižšiu ako určitý práh, ale sú susedné so silnými hranami.

Ak sú susedné hrany silné, ponechajú sa. Ak sú susedné hrany slabé, považujú sa za súčasť hrany iba v prípade, že sú spojené so silnými hranami.

## 5.3 Hough Gradient Method

Houghova transformácia s gradientovou metódou je algoritmus na detekciu priamok v obraze. Algoritmus pozostáva z niekoľkých krokov.

### 5.3.1 Gradienty a Magnitúda

Prvým krokom algoritmu je výpočet gradientov v každom pixeli obrázka. Pre výpočet gradientov sa zvyčajne používa Sobelov operátor, ktorý sme už uviedli pri Cannyho detekcii hrán.

Potom sa vypočíta magnitúda gradientu  $M$  ako:

$$M(x, y) = \sqrt{G_x^2(x, y) + G_y^2(x, y)}$$

### 5.3.2 Houghova Transformácia

Po výpočte magnitúdy sa využije Houghova transformácia na hľadanie priamok v obraze. Houghova transformácia pre každý pixel  $x, y$  spočíva v tom, že sa vygeneruje množina parametrov  $(r, \theta)$ , ktoré určujú priamku prechádzajúcu pixelom  $(x, y)$ . Parameter  $r$  určuje vzdialenosť priamky od počiatku súradnicového systému a parameter  $\theta$  určuje uhol medzi priamkou a osou  $x$ .

Výpočet parametrov  $r$  a  $\theta$  pre pixel  $(x, y)$  sa dá definovať ako:

$$r = x \cos \theta + y \sin \theta \quad \theta = \tan^{-1} \frac{G_{y,x}}{G_{x,y}}$$

Po vypočítaní parametrov  $(r, \theta)$  sa zvýši hlasovanie pre danú priamku v Houghovej transformácii.

### 5.3.3 Hlasovanie a Detekcia Priamok

Po dokončení Houghovej transformácie sa vyhľadajú priamky s najvyšším počtom hlasov. Tieto priamky predstavujú detekované priamky v obraze.

## 5.4 Erosion and Dilation

Erosion a Dilation sú základné operácie v spracovaní obrazu. Tieto operácie slúžia na úpravu tvaru a veľkosti objektov v obraze.

### 5.4.1 Erosion

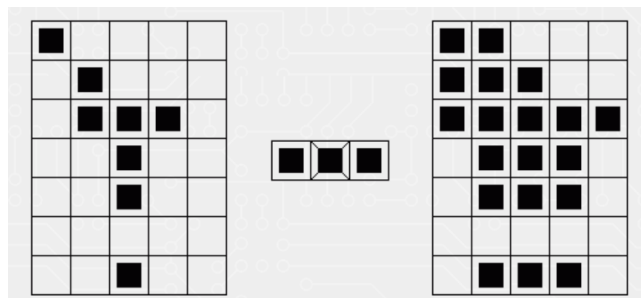
Erosion je operácia, ktorá znižuje objekty v obraze. Erosion sa dá definovať ako binárna operácia, ktorá pre každý pixel v obraze porovnáva hodnoty susedných pixelov v okolí definovanom šablónou. Ak sú hodnoty všetkých susedných pixelov vyššie ako hodnota aktuálneho pixelu, pixel sa ponechá. Inak sa nastaví na hodnotu nula.



Obr. 1: Erosion

### 5.4.2 Dilation

Dilation je operácia, ktorá zväčšuje objekty v obraze. Dilation sa dá definovať ako binárna operácia, ktorá pre každý pixel v obraze porovnáva hodnoty susedných pixelov v okolí definovanom šablónou. Ak sú hodnoty aspoň jedného susedného pixelu vyššie ako hodnota aktuálneho pixelu, pixel sa ponechá. Inak sa nastaví na hodnotu nula.



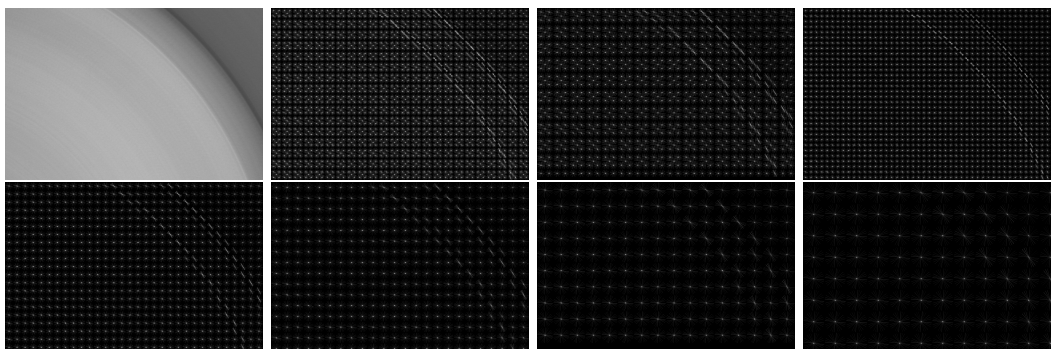
Obr. 2: Dilation

## 6 Vývoj

Vývoj aplikácie bol zameraný na voľbu a implementáciu najvhodnejších techník a metód pre úlohu hľadania stredu rotácie.

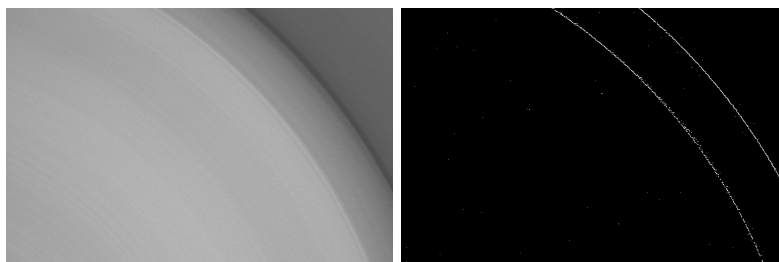
Po analýze dát bolo rozhodnuté otestovať 3 metódy a to sú *Hough-ove transformácie* 5.3, takzvaný *template matching* a *Edge-based* metódu *Canny edge detector* 5.2. Nakoniec ako hlavná metóda pre samotnú detekciu pre obrázky v ktorých sa nachádzal center kruhu bol zvolená technika *Hough-ových transformácií*, pretože sa ukázala ako najvhodnejšia zo zvolených metód. No jej aplikovanie bez pred spracovania dát bolo veľmi chabé.

Následne boli prevedené experimenty s pred spracovaním dát na zlepšenie viditeľnosti hrán ako prvé boli testovaný *Histogram orientovaných gradientov* 5.1 a experimentovanie s jeho parametrami, vo finálnej verzii je použitých viacero histogramov s rôznym nastavením veľkosti plochy v pixloch ako je ukázané na nasledujúcich obrázkoch 3.



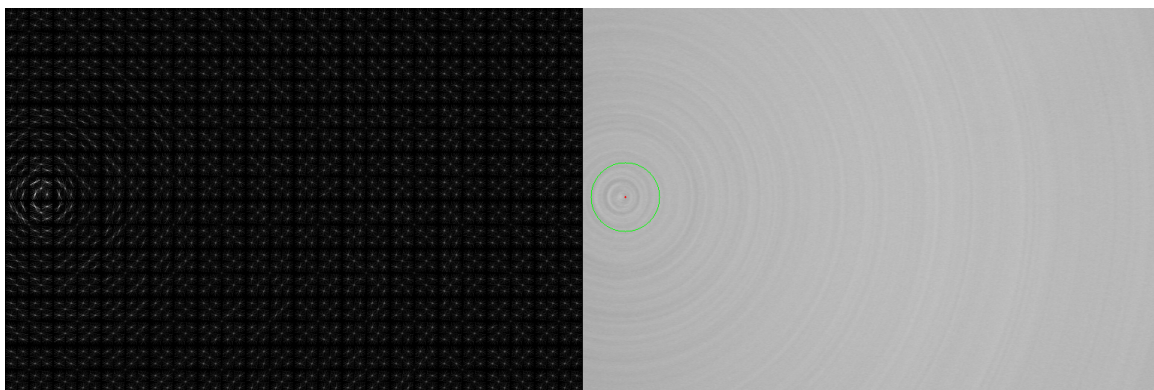
Obr. 3: histogramy orientovaných gradientov s pôvodným obrázkom

ďalej bolo experimentované s možnou aplikáciou techník *dilatation* a *erosion* opísaných v sekcii , a využitia možných filtrov pre spojenie najvýraznejších hrán ukázaných v histogramoch. Avšak tento prístup sa ukázal že len znižuje presnosť následného využitia *Hough-ovej* detekcie oproti iba využitiu samotných histogramov pre nájdenie kruhov. Následne bol opetovne skúšaná technika *Canny edge detector*, pre zvýraznenie hrán, a po ladení parametrov pre túto techniku boli dosiahnuté prekvapivo dobré výsledky uvedené v obrázku 4, avšak treba poukázať že príkladový obrázok ma v porovnaní s dátovou sadou veľmi dobrú viditeľnosť hrany kruhu.

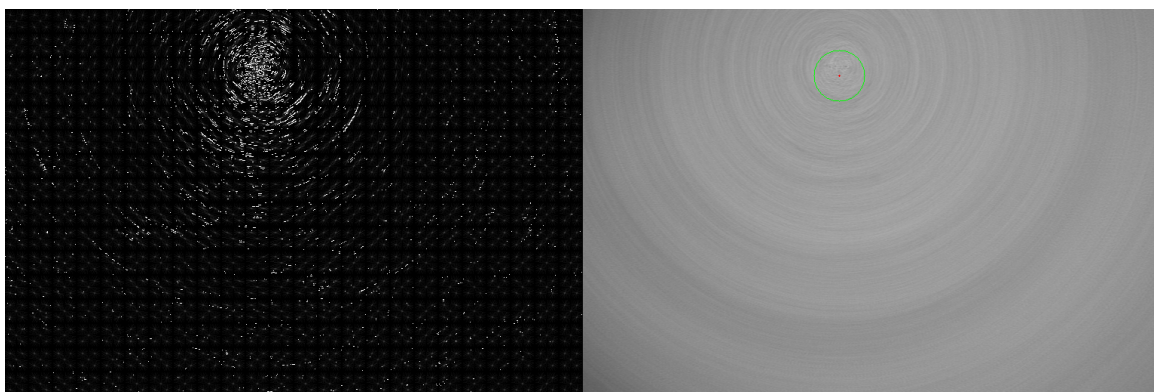


Obr. 4: canny edges s pôvodným obrázkom

Nakoniec pre prípady kedy stred je v obrázku bolo rozhodnuté využitie kombinácie týchto extrakcií, kde finálne riešenie je postupná kombinácia histogramov a hrán, kde pokiaľ nieje kruh nájdený v prvých histogramoch tak sa pokúsi ho nájsť v iných, nakoniec vznikli 4 vrstvy kde prvá používa používa histogramy 8,16,32 druhá histogramy 12,24,48 tretia histogramy 8,16,32,64 so spojením hrán z **Canny**-ho funkcie, a posledná sú len hrany z **Canny**-ho funkcie. Kde výsledky je možno vidieť v obrázku 5 a 6, kde väčšina dát zapadne do prvej alebo tretej vrstvy.

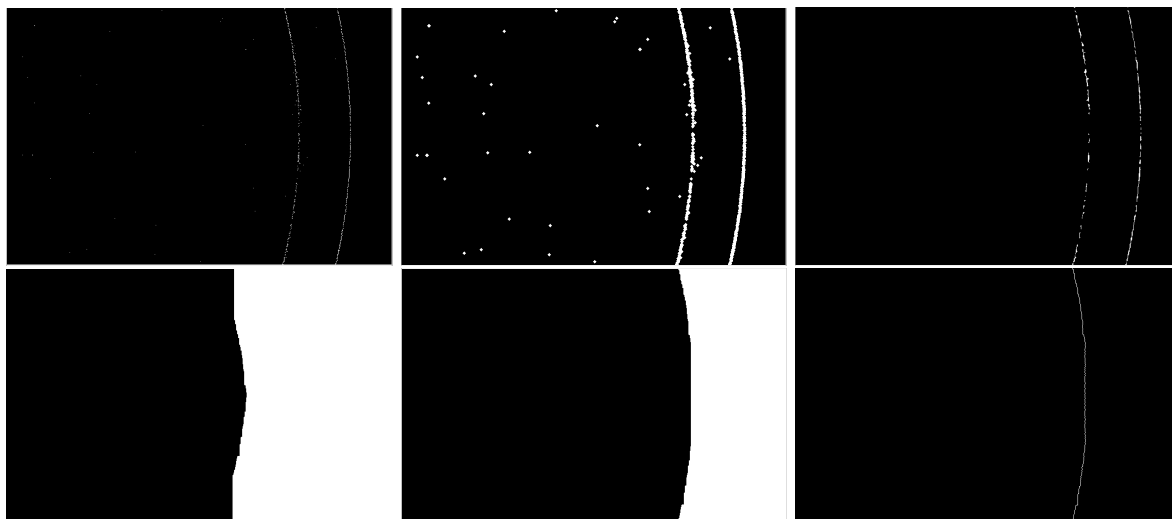


Obr. 5: výsledky detekcie pre očakávaný vstup



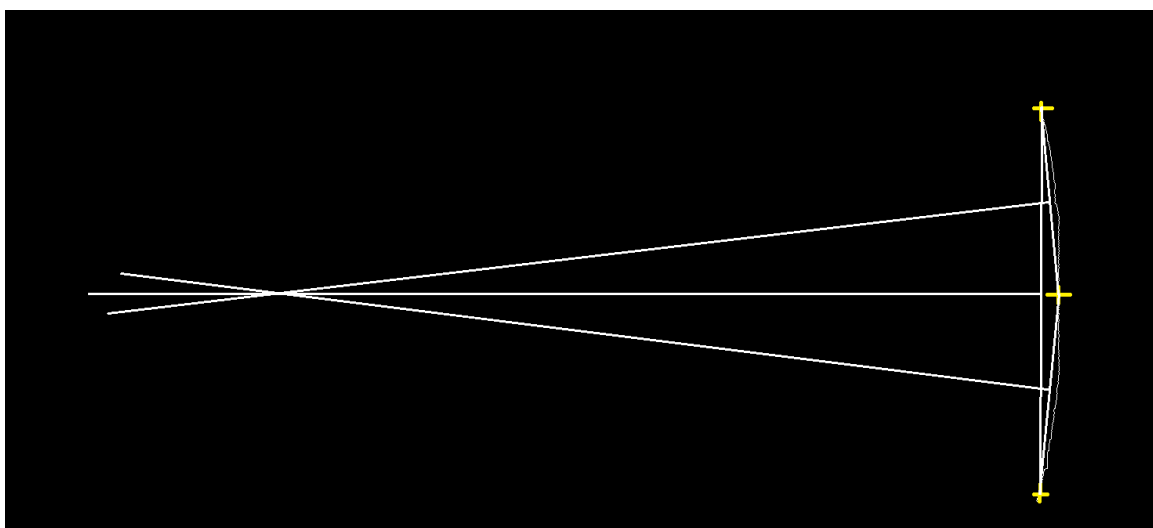
Obr. 6: výsledky detekcie pre očakávaný vstup

Následne pokiaľ nieje nájdený kruh tak sa jedná o obrázok ktorý ukazuje okraj kruhu alebo jeho časť ktorá neobsahuje stred kružnice, v takomto prípade je na extrahované kraje pomocou **Canny**-ho funkcie aplikuje kombinácia Dilation a erosion funkcií pre odstránenie šumu a spojenie prípadných dier v hrane postup je ukázaný na obrázkoch 7. kde ako prvé sa odstráni okolitý šumu a následne sa zvýrazní jediná hranu.



Obr. 7: postup úpravy okrajového obrázka

S takýmto výsledkom je následne triviálne odhadnutie stredu kružnice, kde sa určia dva od seba najvzdialenejšie body na hrne a tretý náhodný bod niekde v strede medzi nimi. Po spojení týchto bodov vzniknú tri úsečky ku ktorým sa postavia 3 kolmice, a v bode kde by sa tieto kolmice pretli je náš stred kruhu. Ilustračne zobrazené na obrázku 8. Avšak pokiaľ sa jedná o veľmi okrajovú úsečku kruhu táto technika nemusí byť stopercentne presná.



Obr. 8: ilustračný obrázok výpočtu stredu kružnice

## 7 Záver a zhodnotenie

Na základe experimentov a testovania algoritmu na datasete obrázkov možno zhodnotiť, že implementovaný algoritmus má úspešnosť v rozsahu 80-90%. Pri niektorých obrázkoch však mohlo dôjsť k neúspešnému nájdeniu kruhu aj napriek tomu, že sa na obrázku nachádza. Tento fakt môže byť spôsobený rôznymi faktormi, ako napríklad nízkou kvalitou obrazu alebo nesprávnym nastavením parametrov algoritmu.

Pri práci na tomto projekte som sa naučil a osvojil si rôzne techniky a metódy na spracovanie a zvýraznenie potrebných vlastností pri spracovaní obrazu. Taktiež som sa oboznámil s rôznymi algoritmami na detekciu kruhov, ktoré sú využívané v praxi.

Možné rozšírenie a vylepšenie algoritmu by mohlo spočívať v dôkladnejšom spracovaní okrajových prípadov a využití hlbokého učenia na extrahovanie príznakov pri vyhotovení tohoto projektu. Tieto vylepšenia by mohli prispieť k ešte presnejším a spoľahlivejším výsledkom detekcie kruhov na obrazoch.

## Literatúra

- [NSH08] Ehsan Nadernejad, Sara Sharifzadeh, and Hamid Hassanpour. Edge detection techniques: evaluations and comparisons. *Applied Mathematical Sciences*, 2(31):1507–1520, 2008.
- [Ped07] Simon Just Kjeldgaard Pedersen. Circular hough transform. *Aalborg University, Vision, Graphics, and Interactive Systems*, 123(6):2–3, 2007.
- [SW11] Xin Shu and Xiao-Jun Wu. A novel contour descriptor for 2d shape matching and its application to image retrieval. *Image and Vision Computing*, 29(4):286–294, 2011.
- [Yan10] Zhuo Yang. Fast template matching based on normalized cross correlation with centroid bounding. In *2010 International Conference on Measuring Technology and Mechatronics Automation*, volume 2, pages 224–227, 2010.