

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## DETEKCE ELIPSY V OBRAZE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. PETR HŘÍBEK

BRNO 2008



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

**FACULTY OF INFORMATION TECHNOLOGY**  
**DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA**

## **DETEKCE ELIPSY V OBRAZE**

ELLIPSE DETECTION

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. PETR HŘÍBEK**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. MICHAL HRADIŠ**

BRNO 2008

## Abstrakt

Práce představuje algoritmy používané pro detekci elipsy v obraze. Každá metoda je teoreticky popsána ve vlastní podkapitole. Mezi popisované metody patří mimo jiné Houghova transformace, Náhodná Houghova transformace, RANSAC, genetické algoritmy a jejich různé úpravy a optimalizace. Dále jsou popsány modifikace stávajících postupů s cílem dosažení lepších výsledků, jedná se o Obloukovou MHT, Kombinaci NMHT a Detekci metodou nejmenších čtverců. Předposlední kapitolou je popis testování parametrů rychlosti, úspěšnosti a přesnosti implementovaných algoritmů. Závěr práce shrnuje výsledky dosažené testováním a jednotlivé metody komentuje.

## Klíčová slova

detekce elipsy, analýza scény, zpracování obrazu, rozpoznání vzoru, Houghova transformace, detekce křivek, RANSAC, náhodná Houghova transformace, genetický algoritmus

## Abstract

The thesis introduces methods used for an ellipse detection. Each method is theoretically described in current subsection. The description includes methods like Hough transform, Random Hough transform, RANSAC, Genetic Algorithm and improvements with optimization. Further there are described modifications of current procedures in the thesis to reach better results. Next to the last chapter represents testing parameters of speed, quality and accuracy of implemented algorithms. There is a conclusion of testing and a result discussion at the end.

## Keywords

ellipse detection, scene analysis, picture processing, pattern recognition, Hough Transformation, curve detection, RANSAC, Random Hough Transform, Genetic Algorithm

## Citace

Petr Hříbek: Detekce elipsy v obraze, diplomová práce, Brno, FIT VUT v Brně, 2008

# Detekce elipsy v obraze

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Michala Hradiše. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Petr Hříbek  
5. května 2008

## Poděkování

Na tomto místě bych chtěl poděkovat vedoucímu mé diplomové práce Ing. Michalu Hradišovi za zájem, připomínky a čas, který věnoval mé práci.

© Petr Hříbek, 2008.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Úvod</b>  | <b>2</b>  |
| <b>2</b> | <b>Přehled</b>   | <b>3</b>  |
| 2.1      | Houghova transformace . . . . .                                  | 3         |
| 2.2      | Modifikovaná Houghova transformace . . . . .                     | 4         |
| 2.3      | Nová Houghova transformace . . . . .                             | 6         |
| 2.4      | Náhodná Houghova transformace . . . . .                          | 8         |
| 2.5      | Omezená náhodná Houghova transformace . . . . .                  | 9         |
| 2.6      | RANSAC . . . . .   | 10        |
| 2.7      | Genetický algoritmus . . . . .                                   | 13        |
| 2.8      | Rychlý robustní genetický algoritmus . . . . .                   | 15        |
| 2.9      | Jednoduchá metoda detekce elipsy . . . . .                       | 17        |
| 2.10     | Účinná detekce . . . . .   | 18        |
| 2.11     | Statisticky účinná metoda . . . . .                              | 20        |
| 2.12     | Složený robustní algoritmus . . . . .                            | 22        |
| <b>3</b> | <b>Modifikace</b>  | <b>25</b> |
| 3.1      | Oblouková Modifikovaná Houghova transformace . . . . .           | 25        |
| 3.2      | Kombinace Náhodné a Modifikované Houghovy transformace . . . . . | 26        |
| 3.3      | Detekce metodou nejmenších čtverců . . . . .                     | 28        |
| <b>4</b> | <b>Testování</b>   | <b>30</b> |
| 4.1      | Typ dat . . . . .  | 30        |
| 4.2      | System vyhodnocování . . . . .                                   | 31        |
| 4.3      | Úspěšnost detekce . . . . .                                      | 31        |
| 4.4      | Přesnost detekce . . . . .                                       | 36        |
| 4.5      | Rychlost detekce . . . . .                                       | 37        |
| 4.6      | Shrnutí . . . . .  | 40        |
| <b>5</b> | <b>Závěr</b>   | <b>43</b> |
|          | <b>Literatura</b>  | <b>44</b> |

# Kapitola 1

## Úvod

Jedním z klíčových problémů počítačového vidění je hledání objektu zájmu. Velkou část informací o tvaru objektů obsahují hrany v obraze. Pátrání po cílovém objektu bývá z tohoto důvodu orientováno na identifikaci tvarů, které poskytují důležité rysy pro detekci.

Mezi významné objekty dvojrozměrného vidění patří elipsy. Objevují se na obrazech díky perspektivní projekci trojrozměrných kruhových a eliptických útvarů. Četnost výskytu a snaha o sémantické vyjádření informace z obrazu vede ke snaze získat prostředek, kterým by bylo možné tyto cíle detekovat.

Analýza tvaru elipsy nachází uplatnění v nejrůznějších oborech. S její pomocí lze detekovat a počítat součástky na páse při průmyslovém výrobním procesu nebo zjišťovat pozici buněk, jež mají v trojrozměrném světě tvar tvar elipsoidu (ve dvou rozměrech jde o elipsu). Konkrétní praktické využití bude v této práci testováno při detekci kruhových dopravních značek.

Druhá kapitola popisuje teorii algoritmů, které se na detekování elips používají. Jde konkrétně o dvanáct postupů, které využívají různé geometrické vlastnosti elipsy k její přesné detekci. Součástí metod jsou i rady k efektivnějšímu předzpracování obrazu před samotným vyhledáváním.

Ve střední části práce se nachází optimalizace původních metod. Jde o tři varianty, které by oproti původním algoritmům měly vykazovat kvalitnější výstupy. Při jejich kompozici byly použity optimální části a rysy originálních postupů.

Poslední část práce srovnává metody v testech rychlosti, kvality a přesnosti detekce. Poměrování probíhá navzájem mezi jednotlivými algoritmy i mezi původními a modifikovanými postupy. Výsledky testů jsou diskutovány, stejně jako vlastnosti algoritmů a důvody získání konkrétních hodnot testu.

Závěr se věnuje shrnutí a možnému pokračování v započatém směru. Výstupem práce by měl být ucelený přehled metod pro detekci elipsy, jejich objektivní srovnání a vyjádření jejich vlastností a schopností.

Diplomová práce navazuje na Semestrální projekt, ve kterém byl uveden popis algoritmů pro detekci elipsy v obraze. Výsledky jsou shrnuty ve druhé kapitole.

# Kapitola 2

## Přehled

V této kapitole uvádím přehled metod používaných k detekci elipsy v obraze. Každý algoritmus je teoreticky popsán, jsou vyjmenovány hlavní rysy a výkonný algoritmus. Hlavní rysy vlastní detekce jsou patrné napříč různými parametry a základní principy zpracování obrazu zůstávají zachovány. Autoři se snaží kombinovat různé techniky a vytvořit tak co nejefektivnější algoritmus.

### 2.1 Houghova transformace

Houghova transformace patří mezi nejčastěji používanou metodu na detekci geometrických primitiv. Představuje základní kámen pro další metody, které redukuje výpočetní čas a prostor.

Cílem metody je nalezení parametrů pomocí analytického popisu nacházených útvarů (přímek, kružnic, elips apod.). Jejich hodnoty se snažíme získat z parametrického (neboli Houghova) prostoru. Tato metoda je často využívána zejména díky její robustnosti - je odolná vůči šumům a nekvalitě původního obrazu, rovněž je schopna detekovat neúplné elipsy.

Primitiva jsou při Houghově transformaci reprezentovány parametrickými křivkami s určitým počtem volných parametrů. Základním principem je definice mapování mezi obrazovým a parametrickým prostorem. V obrázku je každý bod popředí mapováním transformován na buňky parametrického prostoru. Jejich parametry definují primitivum procházející skrz bod v původním obrázku. Těmto buňkám je akumulována hodnota a poté co jsme takto prošli všechny body v obraze, vybereme z akumulátoru lokální maxima odpovídající parametrům specifikovaného tvaru.

Pro každou hledanou hodnotu je nutné uvažovat jednu dimenzi Houghova prostoru. Pokud tedy uvažujeme parametrickou rovnici elipsy 2.1 je patrné, že naše pole bude mít pět rozměrů. Tato velikost se stává s větším prostorem značně problematická.

Parametrická rovnice elipsy

$$\begin{aligned}x &= c_x + a \cos(\theta) \cos(\eta) - b \sin(\theta) \sin(\eta) \\y &= c_y + b \sin(\theta) \cos(\eta) + a \cos(\theta) \sin(\eta)\end{aligned}\tag{2.1}$$

s pěti neznámými - souřadnicemi středu  $c_x, c_y$  (posunutí od počátku souřadného systému), velikostmi hlavní a vedlejší osy  $a, b$  a úhlem natočení  $\theta$  ( $\eta$  je excentrická anomálie nabývající

```

Vstup: Obrázek s pixely náležícími hranám.

Nuluj(Akumulátor[] [] []);

Pro (každý pixel popředí P) proved' {
  Pro (všechny možné Středy elipsy) proved' {
    Pro (všechny možné Velikosti elipsy) proved' {
      Pro (všechny možné Sklony elipsy) proved' {
        Pokud (Elipsa(Střed, Velikost, Sklon) prochází bodem P) {
          Akumulátor[Střed][Velikost][Sklon] ++;
        }
      }
    }
  }
}

Pro (každé lokální maximum Akumulátoru) proved' {
  Pokud (Akumulátor[střed][velikost][sklon] > Práh) {
    Přidej do seznamu Elipsa(Střed, Velikost, Sklon);
  }
}

Výstup: Parametry detekovaných elips.

```

Algoritmus 2.1: Algoritmus Houghovy transformace.

hodnot  $\eta \in (0, 2\pi)$ .

## 2.2 Modifikovaná Houghova transformace

V této podkapitole je uvedena modifikovaná metoda, která využívá dvou vlastností elipsy tak, že iterativně vyhledává vrcholy ve dvou různých parametrických protorech, aby našla body náležící elipse, jak je popsáno v [9]. Jejich parametry se hledají metodou nejmenších čtverců.

Vstupní obraz opět transformujeme hranovou funkcí, detekujeme dlouhé úsečky, u kterých se nepředpokládá, že by mohly být součástí elipsy a odstraníme je.

Detekce středu probíhá pomocí jednoduché vlastnosti - předpokládané rovnoběžné tečny elipsy  $P$  a  $Q$  (viz obrázek 2.1(a)) mají stejnou vzdálenost od středu elipsy  $O$ . Budeme hledat pozice elips v ohraněném obrázku  $E$  za použití dvourozměrného sčítacího pole  $\{a_{x,y}\}$ . Pole je indexováno dvěma parametry  $(x, y)$ , které určují střed elipsy. Pro každý pár dvou pixelů popředí  $(x_1, y_1)$  a  $(x_2, y_2)$  v seznamu hran  $e_\theta$  (seznam hran majících úhel  $\theta$ ) je sčítací pole na pozici  $((x_1 + x_2)/2, (y_1 + y_2)/2)$  inkrementováno o jedna. Poté co je tento krok proveden pro všechny páry množiny  $\{e_\theta\}$ , je pole lokálně zprůměrováno použitím okolí  $3 \times 3$ . Sčítací pole odpovídající středu úplné elipsy má tedy počet přibližně odpovídající délce jejího obvodu.

Poté hledáme v akumulátoru  $a_{m,n}$  s nejvyšší hodnoutou, která indikuje pozici středu elipsy (nebo soustředných elips). Jako kandidátní pixely elipsy vybereme všechny dvojice



hranových elementů, které v předchozím kroku zvyšovaly  $a_{m,n}$ .

Nechť je  $C$  množina vybraných kandidátů na elipsu. Jak bylo již zmíněno, hledání středu jednoduše bere hranové body jak leží na zakřivených nebo přímých úsecích symetrických k bodu  $(m, n)$  v  $E$ . Každý prvek  $C$  tedy nemusí ležet na elipse. Aplikace metody nejmenších čtverců na shodu elipsy s  $C$  by pravděpodobně dala nepřesné výsledky, zvláště pokud se v  $C$  nachází body soustředných elips. Kvalita množiny elementů  $C$  je tedy významně rušena symetrickými vzory neležícími na elipse, proto musíme prvky filtrovat další vlastností elipsy a vybrat hranové body, které určitě leží na elipse. Teprve poté se aplikuje metoda nejmenších čtverců k nalezení správných parametrů.



Obrázek 2.1: (a)  $OP = OQ$ , pokud jsou dvě tečny rovnoběžné. (b)  $1/|OP|^2 + 1/|OQ|^2 = 1/|R|^2 = konstanta$ , pro elipsu pokud  $\angle POQ = 90^\circ$ .

Máme-li takové dva body  $P$  a  $Q$  na elipse, že platí  $\angle POQ = 90^\circ$  kde  $O$  je střed elipsy (viz obrázek 2.1(b)). Potom je jednoduché dokázat, že pro elipsu platí

$$\begin{aligned} \frac{1}{|OP|^2} + \frac{1}{|OQ|^2} &= \frac{1}{|R|^2} \\ &= konstanta. \end{aligned} \quad (2.2)$$

Nyní je k testování kandidátů použito jednorozměrné sčítací pole  $\{a_i\}$ . Předpokládáme že  $O$  je umístěno na pozici  $(m, n)$ , což jsme zjistili v části hledání středu. Pokud některý z bodů  $P$  a  $Q$  v  $C$  splňuje  $\angle POQ = 90^\circ \pm \delta$  ( $\delta$  je míra chyby), potom je akumulátor odpovídající vypočítanému  $R$  ze vzorce 2.2 zvýšen o jedničku. Když tatko projdeme všechny členy v  $C$ , obdržíme histogram  $R$ , který nám dává hodnotnou informaci o elipsách v  $C$ . Více vrcholků v histogramu znamená existenci soustředných elips v  $C$  (i přesto, že je počet elips ve většině případů nižší než počet vrcholků) nebo předpokládáme (pokud pouze o nízké vrcholky), že v  $C$  není žádná viditelná elipsa.

Vzájemné ovlivňování mezi hranovými body soustředných elips způsobuje zřetelné falešné extrémy histogramu, někdy dokonce vyšší než vrcholky odpovídající správným elipsám. V  $C$  je tedy nejprve testována nejmenší elipsa, protože její hodnota v akumulátoru je k tomuto rušení méně citlivá. Vybereme všechny dvojice hranových bodů odpovídajících nejlevějšímu lokálnímu extrému v histogramu. Poté metodou nejmenších čtverců najdeme pět parametrů elipsy, která se s těmito hranovými body nejlépe shoduje. Pokud hranové body v  $E$  pokrývají pouze malou část této elipsy, je proces pozouzen jako chybný. V opačném případě jsou hranové body na nalezené elipse vyjmuty z  $E, C$ , a  $\{e_\theta\}$  kvůli eliminaci jejich interakce s dalšími elipsami. Proces testování kandidátů je zopakován vypočítáním nového histogramu  $R$  z aktualizovaného  $C$  a zkoušením nejlevějšího vrcholku, dokud histogram neobsahuje pouze nízké špičky.

Popsaná procedura pro testování kandidátů je iterativně aplikována na všechny významné shluky ve dvojrozměrném sčítacím poli. Nakonec testujeme, zdali každá krátká úsečka v seznamu přímých čar patří objevené elipse či nikoliv.

## 2.3 Nová Houghova transformace

Tato podkapitola ukazuje, jak je možné rozložit parametrický prostor pomocí polární definice elipsy podle [1]. Úhel počítáme pomocí dvou bodů a jejich sklonu.

Směr je použit k redukování výpočetních požadavků Houghovy transformace. Informaci o úhlu používáme k extrakci elipsy vztahem lokálních geometrických rysů parametrické reprezentace s lokálními rysy v obraze. Elipsu reprezentujeme pomocí ohniska definovaného diferenciovatelnou vektorovou funkcí  $z(\theta) = xU_x + yU_y$  pro  $U_x = [1, 0]$ ,  $U_y = [0, 1]$  a

$$\begin{aligned}x &= a_0 + a_x \cos(\theta) + b_x \sin(\theta) \\ y &= b_0 + a_y \cos(\theta) + b_y \sin(\theta)\end{aligned}\tag{2.3}$$

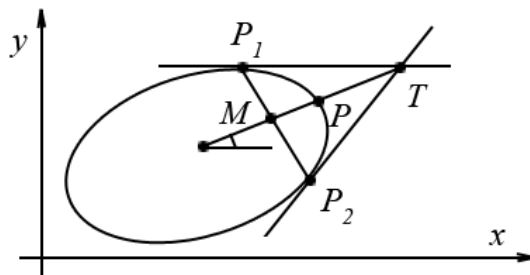
V tomto výrazu je pět parametrů definováno dvěma středovými parametry  $(a_0, b_0)$  a jen třemi z osových parametrů  $(a_x, a_y, b_x, b_y)$ ,  $\theta$  je index pozice. Zbývající parametr os může být vypočítán z ortonormálního vztahu definovaného  $a_x b_x + a_y b_y = 0$ . Tečny úhlu prvního a druhého řádu derivace vektoru pozic  $z(\theta)$  jsou

$$\frac{y'}{x'} = \frac{-a_y \sin(\theta) + b_y \cos(\theta)}{-a_x \sin(\theta) + b_x \cos(\theta)}\tag{2.4}$$

$$\frac{y''}{x''} = \frac{-a_y \cos(\theta) - b_y \sin(\theta)}{-a_x \sin(\theta) - b_x \cos(\theta)}\tag{2.5}$$

kde  $x', y', x''$  a  $y''$  jsou první a druhé derivace vzhledem k  $\theta$ . Kombinováním rovnic 2.4 a 2.5 obdržíme mapování které je vztahem mezi středem a úhlem elipsy.

$$\frac{y''}{x''} = \frac{(y - b_0)}{(x - a_0)}\tag{2.6}$$



Obrázek 2.2: Geometrie dvou bodů na elipse.

Výraz který definuje vztah osových parametrů a informaci o úhlu je formulován omezením vlastnosti ortonormality elipsových os

$$\tan(\phi_1 - \rho) \tan(\phi_2 - \rho) = N^2 \quad (2.7)$$

kde

$$\phi_1 = \arctan\left(\frac{y'}{x'}\right), \quad (2.8)$$

$$\phi_2 = \arctan\left(\frac{y''}{x''}\right), \quad (2.9)$$

$$\rho = \arctan(K), \quad K = \frac{a_y}{a_x}, \quad N = \frac{b_y}{a_x} \quad (2.10)$$

Definováním rovnic 2.6 a 2.7 jako transformace mezi obrazem a parametrickým prostorem je možné vypočítat čtyři parametry elipsy reprezentovné  $(a_0, b_0)$  a vztahy os  $(K, N)$ .

Abychom po výpočtu poměrů pozic os získali i jejich hodnoty, je nezbytné provést algoritmus histogramu akumulátoru, kterým získáme  $a_x$ . Výraz pro osovou část  $a_x$ , máme li hraniční bod a hodnoty  $a_0, b_0, K$  a  $N$ , je

$$a_x = \sqrt{\frac{y_0^2 - x_0^2 N^2}{N^2(1 + K^2)}} \quad (2.11)$$

pro  $x_0, y_0$  definované jako

$$x_0 = \frac{(x - a_0)}{\sqrt{K^2 + 1}} + \frac{(y - b_0)K}{\sqrt{K^2 + 1}} \quad (2.12)$$

$$y_0 = \frac{(x - a_0)K}{\sqrt{K^2 + 1}} + \frac{(y - b_0)}{\sqrt{K^2 + 1}} \quad (2.13)$$

Následně je možné užitím rovnic 2.6, 2.7 a 2.11 dekomponovat parametrický prostor pro extrakci elipsy do dvou nezávislých akumulátorů a histogramu akumulátoru. Tato dekompozice založená na lokální změně křivky definované úhlem tečných vektorových funkcí  $z'(\theta)$  a  $z''(\theta)$ .

Abychom získali úhel prvního a druhého řádu přímé derivace na hranovém bodu  $P$ , využíváme informaci o pozici a směru dvou libovolných bodů  $P_1(x_1, y_1)$  a  $P_2(x_2, y_2)$ . Obrázek 2.2 ukazuje geometrii dvou bodů elipsy a vztahu definovaného jejich derivací. Bod  $P$  může být získán vypočítáním průsečíku úsečky  $MT$  s elipsou. Pokud  $\theta_1$  a  $\theta_2$  definují pozice bodů  $P_1$  a  $P_2$ , potom hodnota  $\theta$ , která specifikuje bod  $P$  je střední hodnota  $\theta_1$  a  $\theta_2$ . Pokud  $a_y = b_x = 0$  a upravíme vzorec 2.4, obdržíme úhel první a druhé přímé derivace pro bod  $P$

$$\frac{y'}{x'} = \frac{Y}{X} \quad (2.14)$$

$$\frac{y''}{x''} = \frac{2M_1X - YM_2}{XM_2 - 2Y} \quad (2.15)$$

kde

$$Y = y_2 - y_1, \quad X = x_2 - x_1, \quad (2.16)$$

$$M_1 = m_1 m_2, \quad M_2 = M_1 + M_2 \quad (2.17)$$

a kde jsou  $m_1$  a  $m_2$  tečny úhlu sklonu tečné úsečky navzájem k bodům  $P_1$  a  $P_2$ . Tuto informaci lze získat rotací obrázku.

## 2.4 Náhodná Houghova transformace

Náhodná Houghova transformace [6] akumuluje body v parametrickém prostoru náhodným vybíráním  $n$ -tic pixelů z obrázku a počítáním parametrů objektu, který skrz tyto pixely prochází. Je to užitečné pro detekci objektů, které jsou definovány jako lineární funkce množiny parametrů.

Elipsa je definována jako řešení rovnice

$$a(x - p)^2 + 2b(x - p)(y - q) + c(y - q)^2 = 1 \quad (2.18)$$

s omezením  $ac - b^2 > 0$ . Tato rovnice je s ohledem na parametry  $(p, q, a, b, c)$  nelineární. Nalezení unikátní elipsy procházející skrz pěti pixelů v obrázku by vyžadovalo řešit rovnici pěti paralelních nelineárních rovnic. Řešit tento problém nelineárních rovnic pro každou pěti vybraných pixelů by zpomalilo Náhodnou Houghovu transformaci a učinilo ji to nepraktickou. Následující algoritmus redukuje tento problém na lineární.

Na začátku se z obrázku vyberou tři body a v každém pixelu se odhadnou tečny - kolem pixelu se definuje malé okolí a k těmto pixelům v okolí se (užitím metody nejmenších čtverců) hledá nejvhodnější úsečka. Vstupem do algoritmu jsou spolu s odhady tečny tři pixely  $x_1, x_2, x_3$ . Vypočítání parametrů odpovídajících elipse bude rozděleno do dvou kroků: hledání středu a shoda zbývajících tří parametrů.

K nalezení středu elipsy využíváme geometrické vlastnosti elipsy. Vezmeme dva body na elipse a najdeme jejich středový bod  $M$  a průsečík jejich tečen  $T$ . Střed elipsy bude ležet na polopřímce  $TM$  (viz obrázek 2.2). Pro jednoduchost mějme jednoduchý příklad kružnice.

Úsečku vypočítáme pomocí pixelů  $x_1$  a  $x_2$ , potom znova s pixely  $x_2$  a  $x_3$ . Průsečík dvou úseček nám dává odhad středu elipsy.

Dále hledáme zbývajících tři parametry elipsy. Nejprve posuneme elipsu na počátek soustavy. To upraví rovnici 2.18 na následující

$$ax^2 + 2bxy + cy^2 = 1 \quad (2.19)$$

Substitucí souřadnic tří pixelů ( $x_1 = (x_1, y_1), x_2 = (x_2, y_2), x_3 = (x_3, y_3)$ ) obdržíme následující systém tří paralelních lineárních rovnic

$$\begin{bmatrix} x_1^2 & 2x_1y_1 & y_1^2 \\ x_2^2 & 2x_2y_2 & y_2^2 \\ x_3^2 & 2x_3y_3 & y_3^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (2.20)$$

Řešení pro  $a, b, c$  dává zbývajících tři parametry elipsy.

Poté zkontrolujeme nerovnost  $ac - b^2 > 0$ , která platí, pokud parametry reprezentují platnou elipsu. Neplatnost implikuje, že buď ony tři pixely neleží na stejné elipse nebo že byly odhady tečen nepřesné. V tom případě výpočet ukončíme a vybereme nový triplet pixelů.

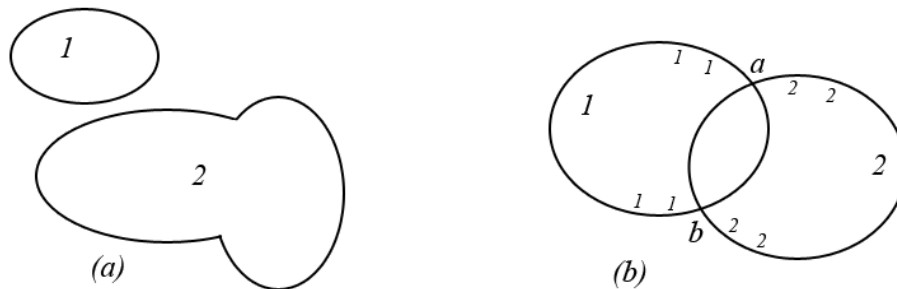
Nahonec převedeme parametry  $(p, q, a, b, c)$  na parametry  $(p, q, r_1, r_2, \theta)$ , kde  $r_1, r_2$  jsou hlavní a vedlejší poloměry elipsy a  $\theta$  je úhel hlavní osy. Tento parametrický prostor se zdá být lepší nežli  $(p, q, a, b, c)$ , neboť body reprezentující elipsy jsou parametrickým prostorem šířeny rovnoměrněji.

Histogram, typicky používaný v Houghových algoritmech, byl nahrazen více efektní strukturou řazeného spojovaného seznamu. Seznam obsahuje pouze oblasti parametrického prostoru, které se vyskytují v původním obrázku. V opačném případě by histogram obsahoval mnoho hodnot, které zůstanou nulové. Pokud bychom použili pole, bylo by dopředu nutné rozhodnout, které oblasti bude parametrický prostor pokrývat. Čím větší je tato oblast, tím větší je požadavek na paměť a naopak čím menší je tato oblast, tím menší je pravděpodobnost, že pokryje všechny elipsy v obraze.

## 2.5 Omezená náhodná Houghova transformace

V této podkapitole je popsáno vylepšení Náhodné Houghovy transformace pomocí redukce nevhodných bodů. Omezená náhodná Houghova transformace je založena na předchozím algoritmu. Aby učinila následné zpracování efektivnější používá navíc předzpracování obrazu, jak je popsáno v [2].

Provádí se ztenčování, jelikož ztenčené obrázky mohou být efektně reprezentovány grafy. Každý pixel ztenčeného obrázku je převeden na uzlu grafu. Uzel grafu sedm má atributů (k popisu jeho pozice a vztahu k ostatním uzlům). Tyto atributy jsou číslo uzlu, číslo křivky, pozice  $(x, y)$ , typ, seznam odkazů, smazání a odkazy na křivky. Číslo uzlu je identifikátor v seznamu uzlů. Číslo křivky uzlu je číslo spojené grafické komponenty ke které uzel patří. Atribut typ je počet odkazů, které spojují uzel s jeho sousedními uzly v grafu. Seznam odkazů popisuje topologické vztahy uzlu s dalšími sousedními uzly. Atribut mazání indikuje potřebnost uzlu. Pokud je nastaven na hodnotu 1, potom se o tento uzel nemusíme starat a můžeme ho vymazat z našeho grafu. Odkazy na křivky představují čísla křivek, které jej protínají, pokud je typ větší než dva.



Obrázek 2.3: (a) Diskrétní elipsy. (b) Překryté elipsy.

Nejprve probíhá skenování ztenčeného obrázku z levého horního rohu do pravého dolního rohu, když se narazí na pixel popředí, zaznamená se pixel jako uzel a typ uzlu je inicializován

na hodnotu nula a číslo křivky dostane nové označení jedna. Pokračujeme v procházení a pokud je nalezen pixel popředí, zkontroluje se levý, levý horní, horní a pravý horní soused tohoto pixelu. Když je některý z těchto sousedních pixelů bod popředí, hodnota typu se zvyšuje o jedničku a číslo uzlu sousedního pixelu popředí je přidáno do seznamu odkazů původního bodu. Sousedící uzly popředí také zvýší hodnoty svého typu a záznam tohoto uzlu ve svých seznamech odkazů, aby utvořily obousměrný graf. Pokud má jeden nebo více sousedů stejné číslo křivky, potom se to číslo uzlu kopíruje. Jestliže mají sousední uzly různá čísla křivek, přidej tyto hodnoty do tabulky ekvivalencí jako totžná čísla. Pokud není v okolí nalezen žádný bod popředí, uzel je považován za startovní uzel k vytvoření nového grafu a zároveň přidáme nové číslo křivky do ekvivalenční tabulky. Poté co takto projdeme všechny body obrázku, jsou spojené útvary v obrázku kódovány do topologických grafů. Algoritmus je ilustrován na Obrázku 2.3(a). Pokud jsou dvě elipsy překryty 2.3(b), musí být tento algoritmus pozměněn.

Když se dvě elipsy protínají v pozici  $a$  v obrázku 2.3(b) a sousedé pixelu popředí mají stejné číslo křivky, potom tento pixel tuto hodnotu přejímá. Jestliže mají různá čísla křivek, pixel popředí musí být průnikem několika křivek (viz  $a$  v Obrázku 2.3(b)) a atribut smazání se změní na hodnotu 1, abychom označili, že tento bod je v budoucím grafu nepotřebný. Odkazy křivek zaznamenávají čísla křivek, které se protínají v pixelu popředí po směru hodinových ručiček a slouží k odlišení různých křivek po setkání s dalším průsečíkem (viz  $b$  v Obrázku 2.3(b)). Pokud jsou křivky mezi dvěma průsečíky (křivky mezi  $a$  a  $b$  na Obrázku 2.3(b)) spojeny a narazí se na jejich pixel popředí, testuje se mazací atribut jeho sousedů. Pokud je mazání nastaveno na hodnotu 1, pak víme, že jde o jednu z protínajících křivek a kopíruje se atribut smazání do tohoto pixelu a další atributy jsou nastaveny podle zmíněné metody. Pokud se narazí na další průsečík ( $b$  v Obrázku 2.3(b)) křivek, pak víme z atributů počtu a mazání jeho sousedů, že to je další průnik a překrývající křivky zde končí. Proto z tohoto pixelu nebudeme dále kopírovat atribut mazání. Atribut odkazů na křivky je zde použit k oznámení, že jde o různé křivky.

Následně se z obrazu odstraní uzly, které mají být smazány a spojené komponenty, ať jsou překryty či nikoliv, jsou kódovány do topologických grafů a dostáváme tak kandidáty na elipsy.

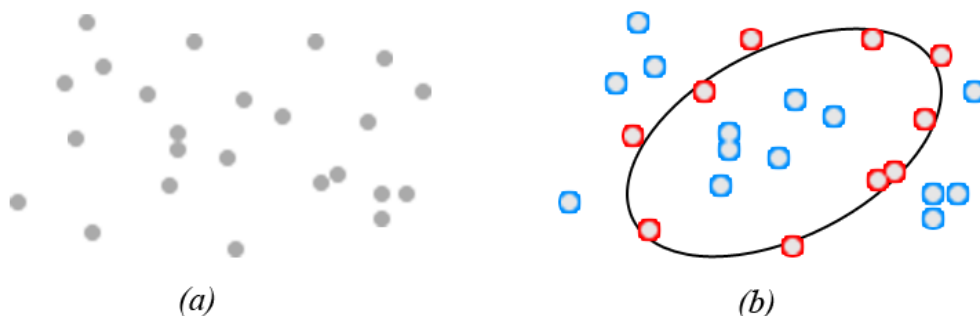
Po tomto zpracování máme detailní informace o obrázku (všechny kandidátní křivky, všechny uzly, uzly připadající křivce, informace o souřadnicích uzlů). Následuje Náhodná Houghova transformace s jistou modifikací. Poprvé mohou být jako kandidáti vybrány pouze tři body v první křivce. Nad nimi se provede Náhodná Houghova transformace. Pokud je nalezena elipsa, bude se pokračovat s vybranými třemi body druhé křivky. Pokud se tak nestalo, vyberou se další kandidáti první křivky a nastává druhá detekce. Všechny křivky jsou procházeny popořadě. Jelikož jsme omezili prohledávací prostor a neuvažujeme neplatné body, je ušetřen čas a zvýšena přesnost. Každá křivka je kandidátem a její vybrané body dokážou, zdali jde o elipsu či náhodně vybrané pixely.

## 2.6 RANSAC

RANSAC je zkratka pro RANdom Sample Consensus. Jedná se o algoritmus [8] na odhad parametrů matematického modelu z množiny pozorovaných dat, která obsahují tzv. outliers (hodnoty mimo naše pozorování).

Základní předpoklad je, že jsou data složena z tzv. inliers tj. datových bodů, které mohou být vyloženy nějakou množinou parametrů modelu a outliers, což jsou data, které se s modelem neshodují. Outliers mohou vzniknout například z extrémních hodnot šumu, nesprávným

měřením nebo chybnými hypotézami o interpretaci dat. RANSAC toto předpokládá a máme-li (obvykle malou) množinu inliers, pak existuje procedura odhadující parametry modelu, který tato data optimálně popisuje, popřípadě se s nimi shoduje.



Obrázek 2.4: (a) Množina dat s mnoha outliers ve kterých hledáme elipsu. (b) Elipsa vypočítaná RANSACem, outliers nemají žádný vliv na výsledek.

RANSAC dosahuje svého cíle itarativním výběrem náhodné množiny původních bodů. Tyto body jsou hypotetické inliers a tato hypotéza je následně testována. Model je porovnáván s hypotetickými inliers, to znamená, že všechny volné parametry modelu jsou rekonstruovány z množiny bodů. Všechny ostatní body jsou potom testovány proti porovnávanému modelu, to znamená, že algoritmus určuje pro každý bod zbývající množiny jak dobře se shoduje s odhadovaným modelem. Pokud se shoduje je také považován za hypotetický inlier. Pokud bylo klasifikováno dostatečně velké množství bodů jako hypotetických inliers, potom znova odhadneme model z celé množiny hypotetických bodů - inliers. Navíc počítáme chybu inliers vzhledem k modelu.

Tato procedura je poté opakována po určitý počet iterací. Pokaždé je výstupem model, který je buď odmítnut, pokud je jako inliers hodnoceno málo bodů, nebo je přijat výsledný model s odpovídající chybovou mírou. Tento model ukládáme, pakliže je chyba nižší než u posledního uloženého modelu (nebo vyšší než prahová hodnota).

Hodnoty parametrů  $t$  (prahová hodnota pro určení zdali bod spadá do modelu) a  $d$  (počet inliers požadovaných k tvrzení, že se model shoduje s daty) musí být určeny ze specifických požadavků spojených s aplikací a množinou dat (je možné experimentální vyhodnocení). Parametr  $k$  (počet iterací), může být určený z teoretického výsledku. Necht'  $p$  je pravděpodobnost, že algoritmus RANSAC v některé iteraci vybere ze vstupní množiny dat pouze inliers, když vybírá  $n$  bodů, ze kerých jsou odhadovány parametry modelu. Hodnota  $p$  udává pravděpodobnost, že algoritmus vyprodukuje správný výsledek. Necht'  $w$  je pravděpodobnost vybrání inlier bodu

$$w = \text{počet inliers} / \text{počet všech bodů}.$$

Obvyklý případ je, že  $w$  není předem známo, ale může být dána nějaká přibližná hodnota. Předpokládáme, že  $n$  bodů, potřebných k odhadu modelu, je vybráno nezávisle a  $w^n$  je pravděpodobnost, že všechny body  $n$  jsou inliers a  $1 - w^n$  je pravděpodobnost, že nejméně jeden z  $n$  bodů je outlier. Tedy případ, který označuje, že bude z této množiny odhadnutý špatný model. Hodnota  $k$  je pravděpodobnost, že algoritmus nikdy nevybere množinu  $n$  bodů, které jsou všechny inliers a to musí být stejné jako  $1 - p$ . Následně

Vstup: Obrázek s pixely náležícími hranám.

```
Pro (K opakování) proved' {
  Náhodně vyber N pixelů;
  Najdi k těmto pixelům Elipsu(Střed, Velikost, Sklon);
  Spočítej Inliers v blízkosti získané elipsy;
  Poměr := Inliers / všechny body elipsy;

  Pokud (Poměr > MaxPoměr) {
    MaxPoměr := Poměr;
    MaxElipsa(Střed, Velikost, Sklon) = Elipsa(Střed, Velikost, Sklon)
  }
}
```

Výstup: Parametry detekované elipsy

Algoritmus 2.2: Algoritmus RANSAC.

$$1 - p = (1 - w^n)^k \quad (2.21)$$

který po zlogaritmování vede k

$$k = \frac{\log(1 - p)}{\log(1 - w^n)} \quad (2.22)$$

Tento výsledek předpokládá, že  $n$  bodů bylo vybráno nezávisle, to je, že bod, který byl jednou vybrán je přemístěn a může být vybrán znovu ve stejné iteraci. To často není vhodný přístup a získaná hodnota  $k$  by měla být stejná jako horní limit v případě, že jsou body vybrány bez přemístění.

Ke zvýšení spolehlivosti může být přičtena ke  $k$  standardní odchylka. Standardní odchylka  $k$  je definována jako

$$SD(x) = \frac{\sqrt{1 - w^n}}{w^n} \quad (2.23)$$

Výhoda RANSACu je schopnost provádět robustní odhad parametrů modelu, to jest, že může odhadovat parametry s vysokým stupněm přesnosti dokonce když jsou v množině dat přítomny outliers. Nevýhodou RANSACu je fakt, že neexistuje horní časová hranice výpočtu těchto parametrů. Pokud je použita horní hranice, obdržené řešení nemusí být to nejoptimálnější.

Algoritmus RANSAC může hledat pouze jeden model pro jednotlivé množiny dat. Stejně jako i u jiných přístupů, které hledají pouze jeden model, platí, že pokud existují v množině dat modely dva (či více), metoda RANSAC nemusí odhalit ani jeden.



## 2.7 Genetický algoritmus

V této kapitole je popsána metoda detekce elipsy založená na genetickém algoritmu. Algoritmus [5] nejprve sdružuje přilehlé hranové pixely s podobnými směry do oblastí zvaných bodové segmenty. Dále jsou z těchto oblastí vybráni kandidáti na elipsy. V tomto algoritmu je každá elipsa definovaná trojicí bodových segmentů. Použitím genetického algoritmu vyhledává trojice, které mají s obrázkem největší shodu.

Popsaný algoritmus sestává z následujících tří kroků.

1. Po hranové detekci jsou přilehlé pixely s podobnými směry sklonu sdružovány do regionů zvaných bodové segmenty.
2. Z množin bodových segmentů jsou vybráni kandidáti na části elips.
3. Výsledné elipsy jsou genetickým algoritmem extrahovány z množin bodových segmentů vybraných v kroku (2). V této fázi je každá elipsa definovaná jako trojice bodových segmentů a ta, která po hledání genetickým algoritmem vykazuje nejvyšší míru shody, je vybrána jako výsledná elipsa. Když je detekována, oblasti které ji tvoří jsou vymazány a hledají se další elipsy.

Navržený algoritmus tedy nejprve následujícím způsobem extrahuje bodové segmenty:

1. Aplikuj hranový filtr na každý pixel  $(i, j)$  v původním obrázku. Nechť  $G_x(i, j)$  a  $G_y(i, j)$  jsou horizontální a vertikální části sklonu získané z masky aplikované na pixel  $(i, j)$ . Poté vypočítej velikost sklonu  $E(i, j)$  a směr  $\Phi(i, j)$  pomocí

$$E(i, j) = \sqrt{G_x(i, j)^2 + G_y(i, j)^2} \quad (2.24)$$

$$\Phi(i, j) = \tan^{-1} \frac{G_y(i, j)}{G_x(i, j)} \quad (2.25)$$

2. Vyber pixely se sklonem mocnosti větším než prahová hodnota  $TH_E$  jako hranové pixely
3. Rozděl hranové pixely obrázku do oblastí podle směru jejich sklonu. Tyto oblasti jsou nazvány bodové segmenty.

Algoritmus vybírá kandidáty na části elips z množiny bodových segmentů. Následující postup rozhoduje, zdali je bodový segment kandidátem na eliptický oblouk.

1. Spočítej excentricitu bodového segmentu. Pokud excentricita překročí prahovou hodnotu  $TH_c$ , přeskoč na krok (2). Jinak se segmentem dále nepracujeme.
2. Rozděl bodový segment do tří podoblastí  $R_k (1 \leq k \leq 3)$  se stejnými délkami.
3. Pro každou  $R_k (1 \leq k \leq 3)$  spočítej střední hodnotu  $\bar{\phi}_k$  ze směrů sklonu pixelů v ní obsažených. Pokud platí  $\bar{\phi}_1 < \bar{\phi}_2 < \bar{\phi}_3$  nebo  $\bar{\phi}_1 > \bar{\phi}_2 > \bar{\phi}_3$  a navíc  $|\bar{\phi}_1 - \bar{\phi}_3| > 10^\circ$ , vyber bodový segment jako kandidáta na část elipsy.

Nejprve vybereme trojici bodových segmentů a potom hledáme elipsu, která se s body z těchto segmentů nejvíce shoduje. Algoritmus počítá míru shody pro všechny generované elipsy a výslednou elipsu vybírá podle největší shody.

Při hledání elipsy k trojici bodových segmentů bereme v úvahu nejprve rovnici elipsy

$$Q(x, y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + 1 = 0 \quad (2.26)$$

a počítáme parametry  $A \sim E$  minimalizující chybovou funkci

$$S = \sum_{i=1}^n \{Q(x_i, y_i)\}^2 \quad (2.27)$$

kde  $(x_i, y_i)$  ( $1 \leq i \leq n$ ) označuje body uvnitř trojice bodových segmentů. Algebraická vzdálenost  $Q(x_i, y_i)$  je brána jako aproximace Eklidovské vzdálenosti mezi bodem  $(x_i, y_i)$  a elisou (potřebujeme aby elipsa co nejlépe vystihla množinu bodů).

Výpočet míry shody elipsy začínáme u nejhodnějších bodových segmentů. Následující postup zjišťuje zdali se pro elipsu hodí bodový segment  $R$ .

1. Spočítej vzdálenost od těžiště  $R$  k elipse. Pokud je vzdálenost menší než prahová hodnota  $TH_1$ , přeskoč na krok (2). V opačném případě již s  $R$  nepracuj.
2. Pro každý pixel  $(x_i, y_i)$  uvnitř  $R$  spočítej vzdálenost  $\varepsilon_i$  od  $(x_i, y_i)$  k elipse. Nechť  $\varepsilon$  označuje průměr  $\varepsilon_i$  všech pixelů uvnitř  $R$ . Pokud je  $\varepsilon$  menší než prahová hodnota  $TH_2$ , potom je  $R$  pro elipsu vhodný.

Jako vzdálenost od pixelu  $(x_i, y_i)$  k elipse použijeme

$$\varepsilon_i = \frac{Q(x_i, y_i)}{|\nabla Q(x_i, y_i)|} \quad (2.28)$$

kde  $\nabla Q(x_i, y_i)$  je sklon  $Q(x_i, y_i)$ . Algebraickou vzdálenost  $Q(x_i, y_i)$  nelze použít, pakliže porovnáváme vzdálenosti dvou bodů ke dvěma elipsám. Váhování inverzní hodnotou sklonu vede ke zlepšení algebraické vzdálenosti  $Q(x_i, y_i)$ .

Po výběru vhodných bodových segmentů počítáme míru shody jako

$$F = \frac{P}{L} \quad (2.29)$$

kde  $P$  je suma délek vybraných bodových segmentů a  $L$  je délka hranice elipsy.

Dále je popsán genetický algoritmus k hledání elips. Trojice bodových segmentů je nazvaná chromozom. Nejprve nastaví  $M$  jako počet elips extrahovaných z obrazu a  $M$ -krát opakuje následující kroky.

Vstup: Bodové segmenty.

```
Generuj 2 chromozomy X, Y;
```

```
Populace := X, Y;
```

```
Pro (T opakování) proved' {
```

```
  Náhodně vyber 2 chromozomy S1, S2;
```

```
  Populace := Populace - (S1, S2);
```

```
  Křížením (S1, S2) získej nové chromozomy (S3, S4);
```

```
  KvalitaA := kvalita chromozomů (S1, S2);
```

```
  KvalitaB := kvalita chromozomů (S3, S4);
```

```
  Pokud (KvalitaA > KvalitaB) {
```

```
    Populace := Populace + (S1, S2);
```

```
  } jinak {
```

```
    Populace := Populace + (S3, S4);
```

```
  }
```

```
}
```

Výstup: Parametry detekované elipsy.

Algoritmus 2.3: Genetický algoritmus.

## 2.8 Rychlý robustní genetický algoritmus

Tato podkapitola obsahuje popis techniky pro získávání elips ze zašuměných obrázků použitím Multipopulačního genetického algoritmu. Oproti tradičnímu Genetickému algoritmu tato metoda [4] paralelně vyvíjí určitý počet subpopulací, které zároveň prohledávají všechna možná optima a výsledkem je dramatický nárůst schopnosti detekovat elipsu.

Algoritmus začíná s jednou populací, u které jsou zaznamenávány dva rysy - podobnost a vzdálenost a jsou v ní hledáni vhodné kandidáty (pokud tam nějakí jsou). Poté je použita shlukovací technika, aby dělila chromozomy na shluky (nebo subpopulace). Všechny subpopulace jsou vyvíjeny paralelně. Pokud se jedna ze subpopulací přibližuje optimálnímu či suboptimálnímu chromozomu, potom je celá subpopulace a odpovídající elipsa (v obrázku) odstraněna. Algoritmus končí, když jsou nalezeny všechny (úplné i částečné) elipsy, nebo když je dosažen přednastavený počet generací.

Jelikož je elipsa reprezentována užitím minimálně pěti bodů, každý chromozom obsahuje pět genů, z nich každý je souřadnicí pixelu. Multipopulační genetický algoritmus vytvoří iniciační populaci 30 až 100 chromozomů podle složitosti cílového obrázku. Náhodně je z množiny popředí v cílovém obrázku vybráno pět bodů pro každý nový chromozom.

V Multipopulačním genetickém algoritmu má míra shoda dvě složky - Podobnost a Vzdálenost.

1. Podobnost  $S$  je vypočítána podílem sumy bodů blízkých vzoru a všech bodů na vzoru.

$$S = \frac{\sum_{(x,y)} \frac{E(x+i, y+j)}{d_{i,j}}}{\#total} \quad (2.30)$$

Pro daný bod  $(x, y)$  na ideální elipse vrací člen  $E(x+i, y+j)$  jedničku, pokud v obrázku existuje blízký nebo totožný bod  $(x, y)$ , jinak vrací nulu. Členy  $i$  a  $j$  reprezentují horizontální a vertikální posunutí mezi bodem na ideální elipse a odpovídajícím aktuálním bodem v obrázku.  $\#total$  je úplný počet bodů na ideálním vzoru.  $d_{i,j}$  je definováno jako

$$d_{i,j} = e^{\frac{|i|+|j|}{4}} \quad (2.31)$$

2. Vzdálenost  $D$  je definována jako podíl sumy vzdáleností bodů a počtem efektivních bodů (bodů do určité vzdálenosti).

$$S = \frac{\sum_{(x,y)} d_{i,j}}{\#eff} \quad (2.32)$$

$\#eff$  je úplný počet efektivních bodů. Tento člen je použitý k odhadu průměrné vzdálenosti mezi pixely aktuální části elipsy a jejího ideálního vzoru. V Multipopulačním genetickém algoritmu je míra shody vypočítávána následujícím způsobem. Ideální elipsa je spočítána z pěti genů každého chromozomu. Po srovnání této elipsy s obrázkem jsou k výpočtu použity rovnice 2.30 až 2.32 a výsledkem je shrnutí nakolik se tato elipsa shoduje s aktuálním rozložením pixelů podle podmínek Podobnosti a Vzdálenosti.

Ukončení vývoje některé z těchto podpopulací nastává nezávisle na ukončení ostatních. Subpopulace končí pokud existuje optimální chromozom a dále pokud existuje vhodný chromozom, jehož populace byla po dlouhou dobu stejná. Subpopulace je efektivně ukončena také když je sloučena s další populací.

V tomto algoritmu je se subpopulacemi manipulováno skrz shlukovací proces a každý shluk je soustředěn kolem chromozomu s nejvyšší Podobností. Tento proces zahrnuje Migraci, Dělení a Slučování. Je aplikován na všechny dobré chromozomy v každé populaci. Všechny špatné chromozomy jsou jednoduše ponechány v jejich vlastních shlucích aby neovlivnily směr vývoje.

1. Migrace - přesun chromozomu do jiného shluku.
2. Dělení - vytváření nového shluku.
3. Slučování - pokud se dva shluky vyvíjí ke stejnému optimu, jsou sloučeny.

Evoluce postupuje hlavně skrz Selekcí a Diverzifikaci. Selekcce se soustřed'uje na hledání v oblastech příznivých na shodu. Naopak diversita je dosažena křížením a mutací, což vede k vyhledávání nových a potenciálně slibných oblastí.

Během evoluce subpopulace je nejlepších 15% konkrétní generace kopírováno do další generace bez modifikace. Analogicky jsou z celé této populace vybrány dva chromozomy s pravděpodobností úměrnou jejich Podobnosti. Tyto dva chromozomy jsou poté kříženy, aby generovaly nového potomka.

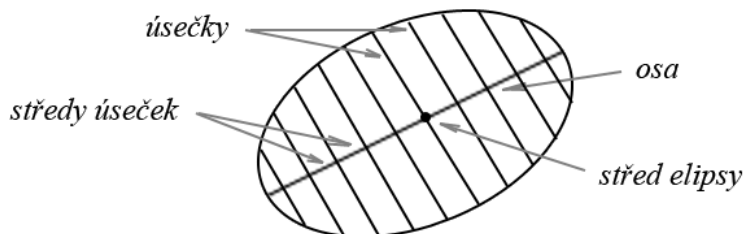
Pro mutaci definujeme nový mutační operátor nastavený speciálně pro naši aplikaci. Nejprve je z chromozomu, který jsme hodlali mutovat, náhodně vybrán gen (či bod) chovající se jako počáteční bod cesty, která překračuje obvod. Pokud je cesta dostatečně dlouhá, jsou z této cesty náhodně sbírány také zbývající geny (body). Je vysoce pravděpodobné, že ostatní body budou ležet na vhodné kandidátské elipse stejně tak dlouho jako počáteční bod. To velmi zvyšuje možnost mutování daného chromozomu na lepší.

Každý nový chromozom, zavedený do další generace, je testován na svoji vhodnost a později, pokud je dostatečně kvalitní, také na Migraci či Dělení.

## 2.9 Jednoduchá metoda detekce elipsy

Metoda detekce elipsy popsaná v této kapitole je podle autora [10] spolehlivější než předchozí metody, vede k významným úsporám výpočetního času včetně paměťových požadavků a eliminuje potřebu hledání směrů hran v obraze. Je potřeba pouze jejich poloha.

Základním principem je typická geometrická vlastnost (viz obrázek 2.5). Pokud v elipse nakreslíme množinu rovnoběžných úseček, nezávisle na jejich směru, potom poloviny těchto úseček leží na přímce, která prochází skrz střed elipsy.

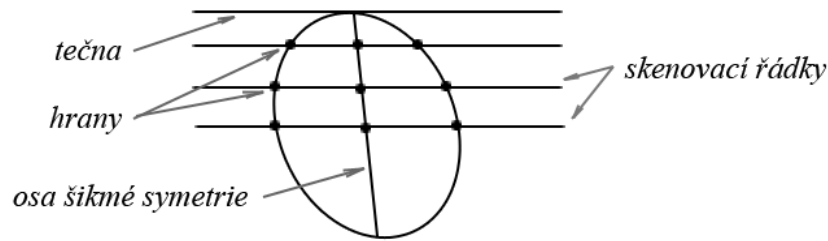


Obrázek 2.5: Středy množiny rovnoběžných úseček leží na téže přímce.

Obecně je tato středová vlastnost objektu nazývána šikmá symetrie. Pravá a levá polovina objektu nejsou ve stejné vzdálenosti od osy symetrie, ale vzdálenost není měřena kolmo k této ose. Pro elipsu je osa průměr, proto je elipsa šikmo symetrická pro jakýkoliv směr.

K aplikování této teorie při detekování elipsy procházíme obrázek a na každém skenovacím řádku detekujeme hrany. Souvislé páry opačných hran mohou určit úsečku procházející objektem. Proto budou tyto dvojice, nalezené na všech skenovaných řádcích, obsahovat rovnoběžné úsečky (viz obrázek 2.6).

Pro každou dvojici bodů spočítáme jejich střed. Množina středů je zpracována standardním úsečkovým detektorem jakým je Houghova transformace. Ta detekuje osy šikmé symetrie (pokud existují).



Obrázek 2.6: Množina středů rovnoběžných úseček ležící na přímce nazývané osa šikmé symetrie.

Proces může být opakován i pro další směry skenovacích řádků, aby byly nalezeny šikmé symetrie v ostatních směrech. Jak bude dále popsáno, výsledky mohou být pro detekci elipsy a odhad parametrů kombinovány.

Hranové body, jejichž středy leží na detekované úsečce (při procházení v daném směru), jsou označeny jako členy množiny šikmé symetrie. Poté srovnáváme množiny šikmé symetrie procházené v různých směrech, abychom zjistili zda některá z nich obsahuje totožné prvky. Pokud taková množina společných bodů existuje, znamená to možnou přítomnost elipsy nebo skupiny eliptických obrysů mající stejný střed, úhel a excentricitu. Tato množina bude nazývána eliptická množina. Středové souřadnice, úhel a excentricita elipsy (elips) odpovídající eliptické množině mohou být jednoduše spočítány z parametrů symetrických os dvou množin šikmé symetrie.

Každá eliptická množina by měla obsahovat jeden nebo více soustředných eliptických obrysů lišících se pouze ve velikosti. Abychom tyto velikosti oddělili (poté co známe úhel a excentricitu množiny), můžeme provést geometrickou (afinní) transformaci, která změní elipsy na kružnice. Množina potom obsahuje kruhové obrysy (místo eliptických). Pak stačí spočítat vzdálenosti bodů od středu a vytvořit histogram ukazující poloměry jednotlivých hran. Lze tedy určit podmnožinu hran odpovídajících každému poloměru. Tato podmnožina odpovídá elipse v původním obrázku.

Takto nalezené elipsy můžeme ověřit testováním každé eliptické podmnožiny hran. Pět parametrů elipsy, ke kterým tato podmnožina patří, je známo. Lepší odhady parametrů lze získat přímým shodováním elipsy s podmnožinou pomocí metody nejmenších čtverců. Není potřeba žádné další hledání, protože podmnožina reprezentuje právě jednu elipsu. Míra shody může být získána spočítáním průměrné vzdálenosti hran (v eliptické podmnožině) od elipsy. Pokud je vzdálenost malá, byla detekována elipsa.

## 2.10 Účinná detekce

V této kapitole je představen další přístup k detekci elipsy. Metoda využívá vlastností hlavní osy pro rychlé a efektivní hledání parametrů. K akumulování informace o délce vedlejší osy elipsy stačí pouze jednorozměrné pole. Požadované výpočetní i prostorové nároky jsou podle autora [11] ve srovnání s jinými metodami o mnoho nižší.

U libovolné elipsy hledáme pět neznámých parametrů,  $(x_0, y_0)$  pro pozici středu,  $\alpha$  pro úhel a  $a, b$  pro délku hlavní resp. vedlejší osy. Obvykle potřebujeme k vypočítání všech parametrů množinu pěti pixelů hrany. Pokud využijeme další informace bodů hrany, stačí ke zjištění pozice elipsy méně bodů. V následujícím textu je představen postup k dosažení

tohoto cíle.

U každého páru pixelů  $(x_1, y_1)$  a  $(x_2, y_2)$  předpokládáme, že jsou to dva body na hlavní ose elipsy. Poté můžeme vypočítat čtyři parametry předpokládané elipsy následujícím způsobem:

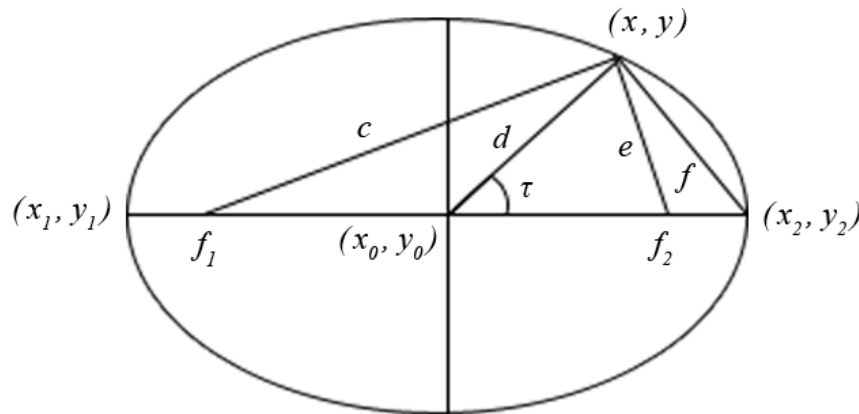
$$x_0 = \frac{(x_1 + x_2)}{2} \quad (2.33)$$

$$y_0 = \frac{(y_1 + y_2)}{2} \quad (2.34)$$

$$a = \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{2} \quad (2.35)$$

$$\alpha = \tan^{-1} \frac{y_2 - y_1}{x_2 - x_1} \quad (2.36)$$

Kde  $(x_0, y_0)$  je střed předpokládané elipsy,  $a$  je poloviční délka hlavní osy a  $\alpha$  je orientace elipsy.



Obrázek 2.7: Geometrie elipsy.

Obrázek 2.7 ukazuje geometrii elipsy, kde  $f_1$  a  $f_2$  jsou ohniska elipsy a  $(x, y)$  je třetí bod použitý k vypočítání pátého parametru. Vzdálenost mezi  $(x, y)$  a  $(x_0, y_0)$  by měla být menší než vzdálenost mezi  $(x_1, y_1)$  a  $(x_0, y_0)$  nebo mezi  $(x_2, y_2)$  a  $(x_0, y_0)$ . Takže polovina délky vedlejší osy může být vypočítána následující rovnicí

$$b^2 = \frac{a^2 d^2 \sin^2 \tau}{a^2 - d^2 \cos^2 \tau} \quad (2.37)$$

kde  $\cos \tau$  je

$$\cos \tau = \frac{a^2 + d^2 - f^2}{2ad} \quad (2.38)$$

a  $d$  je vzdálenost mezi  $(x, y)$  a  $(x_0, y_0)$ .

Vstup: Obrázek s pixely náležícími hranám.

```
Pro (každý pixel popředí P) proved' {
  Pro (každý pixel popředí Q) proved' {
    Pokud (Vzdálenost(P, Q) > MIN) {
      Pomocí (P, Q) vypočítej Střed, Sklon a Velikost hlavní osy;
      Vynuluj Akumulátor[];
      Pro (každý pixel popředí R) proved' {
        Pomocí R vypočítej Velikost vedlejší osy;
        Akumulátor[Velikost vedlejší osy] ++;
      }
      Velikost vedlejší osy := PoziceMaxima(Akumulátor);
      Pokud (Maximum(Akumulátor) > Práh) {
        Přidej do seznamu Elipsa(Střed, Velikost, Sklon);
      }
    }
  }
}
```

Výstup: Parametry detekovaných elips.

#### Algoritmus 2.4: Algoritmus Účinná detekce.

V důsledku lze tedy užitím rovnic 2.33 až 2.38 vypočítat všech pět parametrů elipsy. Stačí jen akumulovat hodnoty polovičních délek vedlejší osy. Využíváme tedy jednodimenzionální sčítací pole. Pokud počet hlasů dosáhne prahu, je nalezena elipsa a můžeme z obrázku odstranit všechny body, které na ní leží. Poté, co je takto testován pár bodů, vyprázdníme sčítací pole a pokračujeme dalším párem.

Jak jsme již výše zmínili, kroky algoritmu 2.4 vedou k detekci elipsy v obrázku. Vstupním obrázkem je hranově filtrovaný obraz, kde černé pixely reprezentují pozadí a bílé pixely jsou body popředí.

## 2.11 Statisticky účinná metoda

V této kapitole je popsána statisticky účinná metoda pro detekci elipsy v obraze. Máme množinu oblouků a jsou představena geometrická kritéria k vybrání možných párů obloukových částí patřících stejné elipse. Platnost obloukových dvojic je postupně ověřena nebo zamítnuta podle statistických kritérií - ověřování předpokladů. Mezi výhody této techniky patří odolnost vůči šumům.

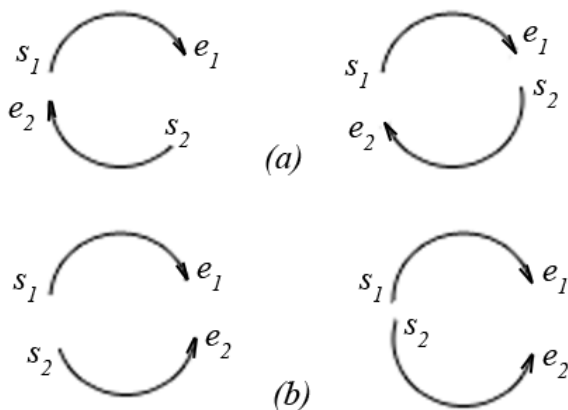
Sdružování oblouků je prováděno ve dvou krocích. První krok, uváděný jako bezseznamové shlukování, vykonává obloukové sdružování spojováním sousedních obloukových segmentů (nahrazování sousedících krátkých oblouků delšími). Spojené oblouky patří stejnému seznamu spojených hran. Druhý krok - meziseznamové shlukování zahrnuje spojování oblouků umístěných v různých seznamech spojených pixelů. Při obloukovém shlukování se musí rozhodnout, které dva oblouky by měly být vybrány jako kandidáti na možné spojení. Podmínka vzdálenosti určuje, zdali jsou pro spojení dostatečně blízko.



Posuzuje se mezi

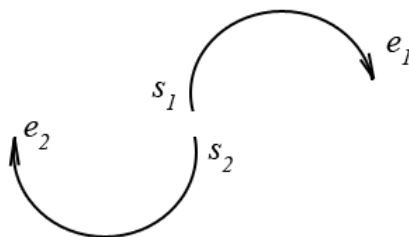
- počátkem jednoho oblouku a koncem druhého, pokud mají oblouky stejný směr.
- koncem (začátkem) jednoho oblouku a koncem (začátkem) druhého, pokud mají oblouky opačný směr.

To znamená čtyři možná uspořádání pro dva eliptické oblouky jak je graficky znázorněno na obrázku 2.8. Oblouky vybrané pro sloučení musí odpovídat jednomu z uvedených způsobů uspořádání.



Obrázek 2.8: Možné uspořádání mezi dvěma oblouky. (a) Dva oblouky se stejným směrem. (b) Dva oblouky s různými směry.

Oblouky s jiným uspořádáním bychom měli vyloučit. Příklad takového uspořádání mezi dvěma oblouky jež mají stejný směr ukazuje obrázek 2.9. V tomto uspořádání je vzdálenost vypočítána ze začátku prvního oblouku a začátku druhého oblouku. Ačkoli vzdálenost splňuje podmínku, nemohou být sloučeny, jelikož jejich spojení nemůže vytvořit elipsu.



Obrázek 2.9: Příklad nespojitelného uspořádání mezi dvěma oblouky.

Je potřeba získat rozhodovací mechanismus, zda k sobě oblouky patří. Principem, jak už z názvu vyplývá, je statistika. Stanovují se hypotézy a zjišťuje se nakolik jsou splněny. V praxi to znamená, že máme např. dvě elipsy a sledujeme, ke které se daný oblouk více statisticky hodí. Autor [7] uvádí tři možné statistické přístupy, z nichž žádný není optimální.

První test neumí odhadnout podobnost mezi parametry testovaných oblouků. Druhá metoda pracuje dobře, pokud mají oba oblouky stejnou délku. Pokud je jeden z nich delší, statistický test je ovlivněn širším útvarem. Třetí metoda trpí nedokonalostí testovací statistiky, ve které chybí kontrola rozptylu zbytkových chyb (velký rozptyl znamená nízkou míru shody). Proto se provádí další test, který tento nedostatek eliminuje.

Pro vysokou míru shody obou těles by měly mít zbytkové chyby obou oblouků stejné nebo blízké roptyly. To je ověřeno statistickým testováním rozložení podílu obou rozptylů. Dva oblouky jsou spojeny pouze pokud jejich příslušná statistika  $T$  a poměr rozptylu splňuje požadované rozložení  $F$ .

Statisticky účinná metoda tedy definuje techniku pro detekování elipsy ze seznamů spojených hraničních pixelů. Postup by měl zlepšovat stávající obloukové techniky přidáním statistických kritérií pro slučování oblouků patřících stejné elipse. Vylepšení vede k získání delších eliptických oblouků a eliminuje nevhodné části.

## 2.12 Složený robustní algoritmus

Algoritmus popsáný v [3] se zabývá převážně přesností vybírání bodů z obrázku, ze kterých se poté počítají parametry elipsy. Náhodně vybere pěti bodů a spočítá parametry elipsy pomocí následující obecné rovnice elipsy

$$Ax^2 + By^2 + Cxy + Dx + Ey - 1 = 0 \quad (2.39)$$

a zjišťuje, nakolik vypočítané parametry elipsy odpovídají vzorům z obrázku (viz dále). Rovnice, kde vystupuje pět bodů elipsy má tvar

$$\begin{bmatrix} x_1^2 & 2x_1y_1 & y_1^2 & x_1 & y_1 \\ x_2^2 & 2x_2y_2 & y_2^2 & x_2 & y_2 \\ x_3^2 & 2x_3y_3 & y_3^2 & x_3 & y_3 \\ x_4^2 & 2x_4y_4 & y_4^2 & x_4 & y_4 \\ x_5^2 & 2x_5y_5 & y_5^2 & x_5 & y_5 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} = 1 \quad (2.40)$$

kde  $(x_i, y_i)$  je souřadnice bodu vstupního obrázku. Potom bychom měli ověřit zdali se vyřešené parametry reprezentují elipsu.

$$I_3 = \begin{vmatrix} A & B/2 & D/2 \\ B/2 & C & E/2 \\ D/2 & E/2 & F \end{vmatrix} \quad I_2 = \begin{vmatrix} A & B/2 \\ B/2 & C \end{vmatrix} \quad I_1 = A + C \quad (2.41)$$

kde  $F = -1$ . Získané hodnoty nejsou parametry elipsy pokud platí  $I_2 \leq 0$  nebo  $I_1 I_3 > 0$ . Pokud jde o elipsu, symetrická matice může být transformována na diagonální matici pomocí ortogonální transformace a diagonální matice je standardní forma matice elipsy  $I_3$ . Vzorec ortogonální transformace je

$$P^T I_3 P = I_3^* \quad (2.42)$$

kde  $I_3^*$  je standardní forma matice elipsy  $I_3$  a  $P$  je ortogonální matice. Poté se řeší ohniska elipsy

$$\bar{Z}_1 = P[c_1^* \ 0 \ 1] \quad \bar{Z}_2 = P[-c_1^* \ 0 \ 1] \quad (2.43)$$

kde jsou  $\bar{Z}_1$  a  $\bar{Z}_2$  homogenní souřadnice ohisek a označíme odpovídající souřadnice jako  $Z_1, Z_2$ . Potom můžeme posoudit, zdali je bod  $Q$  na elipse, pokud splňuje následující nerovnici

$$|||Q - Z_1|| + ||Q - Z_2|| - 2a| < \xi \quad (2.44)$$

kde  $a$  je hlavní osa elipsy.

Ve srovnání s metodou nejmenších čtverců, je pravidlo 2.44 přijatelné, efektní a také je rychlejší.

Poté co bylo nastaveno  $\xi$  a body odpovídající přechozí rovnici dosáhly definované kvantity, získáváme kandidátní elipsu. Ze všech takto získaných elips najdeme tu, která se s body shoduje (obsahuje jich nejvíce).

U komplexních obrázků je příliš mnoho bodů, které neleží na elipsách a které přináší příliš mnoho neúčinných výpočtů. Víme, že se v obrázcích z reálného světa objevuje příliš mnoho rohových bodů a přímých čar, proto bychom měli tyto rohové body a rovné čary filtrovat jako první. Jejich rysy se od elips značně liší, což je výhodné. Měli bychom také eliminovat příliš krátké úseky a je vhodné testovat zbylé křivky, zda se shodují s elipsou metodou nejmenších čtverců. Tato předzpracování netrvalí příliš dlouho, ale velmi zlepšují rychlost algoritmu, neboť redukuje nadbytečné výpočty. Spojením všech předchozích metod vzniká algoritmus 2.5.

Vstup: Barevný obrázek.

```
Použij Cannyho hranový detektor;
Filtruj (šum, rohové body a přímé čáry);
Filtruj (délka(bodový segment) < Práh);

Pro (bodové segmenty seřazené podle délky) proved' {
  Hledej Elipsu(Střed, Velikost, Sklon) metodou nejmenších čtverců;

  Pokud (Elipsa nalezena) {
    Přidej do seznamu Elipsa(Střed, Velikost, Sklon);
    Odstraň pixely elipsy ze všech bodových segmentů;
  } jinak {
    Pro (K iterací) proved' {
      Náhodně vyber pětici pixelů P;
      Vypočítej parametry elipsy pomocí P;
      Pokud (elipsa je platná) {
        Přidej do seznamu Elipsa(Střed, Velikost, Sklon);
      }
    }
  }
}
```

Výstup: Parametry detekovaných elips.

Algoritmus 2.5: Složený robustní algoritmus.

## Kapitola 3

# Modifikace

Metody popsané v druhé kapitole využívají různých geometrických vlastností elipsy, aby z obrazu v ideálním případě rychle a přesně extrahovaly všechny objekty při co nejnižších paměťových nárocích. U zmíněných atributů jdou jednotlivé požadavky proti sobě, proto se hledá kompromis.

Některé z postupů jsem se pokusil vylepšit (zvýšit jejich úspěšnost, přesnost či rychlost). Hlavní změnou bylo provádění operací nad většími topologickými celky - bodovými segmenty (podobně jako u Genetického algoritmu v podkapitole 2.7). Algoritmy jsou tedy v úpravě prováděny na každém segmentu zvlášť. Ztrácíme tak informaci o bodech mimo segment, ale zároveň již tyto pixely nemusíme vyhodnocovat a šetříme výpočetní čas. Výsledek je poměrem mezi rychlostí (zjednodušili jsme obraz) a přesností (ztráta informace kvůli zjednodušení).

Následující modifikace vykrystalizovaly při zpracování a implementaci původních metod jako zajímavé úpravy hodné testování a porovnání s mateřskými metodami. Hlavní motivací k úpravě stávajících metod bylo zvýšení rychlosti detekce. Implementovány byly i další upravené algoritmy, ale jejich experimentální výsledky nepřekračovaly zde uvedené popisy.

### 3.1 Oblouková Modifikovaná Houghova transformace

Důvodem inspirace Modifikovanou Houghovou transformací byla její důmyslnost a přesnost. Cílem nové metody byla snaha o časovou optimalizaci. Klasická MHT jak byla popsána v podkapitole 2.2 určuje střed pomocí histogramu akumulátoru. Algoritmus je postupně opakován pro všechna lokální maxima. Ta zahrnují i vrcholky, které nereprezentují střed elipsy. Výpočet a hledání bodů v těchto případech neefektivně zabírá výpočetní čas.

Názorně lze popsanou situaci vidět na obrázku 3.1. Velké množství redundantních bodů tvoří v pravé části daleko vyšší pozice histogramu než značně méně početné body obvodu elipsy. V obraze je střed elipsy poměrně dobře patrný, okolo se vysoké hodnoty nevyskytují. Bohužel nemůžeme použít sklon tečné plochy, abychom tento prudký nárůst detekovali. Mohli bychom tak nechtěně získat střed mezi dvěma elipsami (viz obrázek 3.1). Stejně tak by mohla elipsa ležet v levé části obrázku, kde by byl nárůst hodnot nepatrný. Postupné procházení všech významných vrcholeků histogramu původní Modifikované Houghovy transformace bylo nutné eliminovat jiným způsobem.

Pakliže budeme operovat pouze nad seznamy spojených bodů, resp. bodovými segmenty, předpokládáme, že tento úsek bude obsahovat maximálně jednu elipsu. Zmíněné opakované hledání lokálních extrémů ve dvou akumulátorech se mění na unikátní získání globálních



Obrázek 3.1: Detekce středu elipsy dle původního algoritmu Modifikovaná Houghova transformace (podkapitola 2.2). (a) Původní obraz. (b) Hrany původního obrazu na výšce červené barvy a jejího okolí. (c) Normovaný histogram akumulovaných pozic středů úseček mezi body hran.

maxim. Zjednodušení přináší i větší nároky na souvislost kontury. Použití rychlého způsobu nalezení středu elipsy pomocí akumulátoru nezíská při významné absenci bodů hrany přesné informace o jeho pozici. Typický výsledek detekce zvláště pro každou polovinu elipsy je vidět na obrázku 3.2(c). Pokud bychom detekovali oba oblouky značky Zákaz stání, získáme střed zcela přesně, jelikož absence bodů na hranách je symetrická (viz obrázek 3.2(b)). Protože postupujeme po jednotlivých obloucích, nejsme schopni střed správně detekovat.

Ačkoliv tato varianta prochází pole jen jednou, je nutné ji opakovat pro všechny významné bodové segmenty. Jednotlivé kroky jsou popsány v algoritmu 3.1.

### 3.2 Kombinace Náhodné a Modifikované Houghovy transformace

V této podkapitole je popsán algoritmus rozšiřující postup z částí 2.2 a 2.4. Využívá kombinování Modifikované Houghovy transformace a Náhodné Houghovy transformace.

Hledání středu probíhá stejně jako u MHT pomocí akumulování pozice středu úsečky mezi pixely popředí. Tento způsob je při detekci na bodových segmentech poměrně výhodný (nevznikají falešná maxima). Pro neúplné kontury nejsou výsledky uspokojivé (problematiku podrobněji popisuje podkapitola 3.1 a obrázek 3.2(c)) a je možné využít jiný způsob nalezení středu. Původní způsob detekce středu Náhodné Houghovy transformace nenajde vždy správné řešení, je to dáno nepřesným odhadem tečen kvůli rozlišení 1 pixel.

Vstup: Obrázek s pixely náležícími hranám.

Najdi v obraze všechny bodové segmenty;

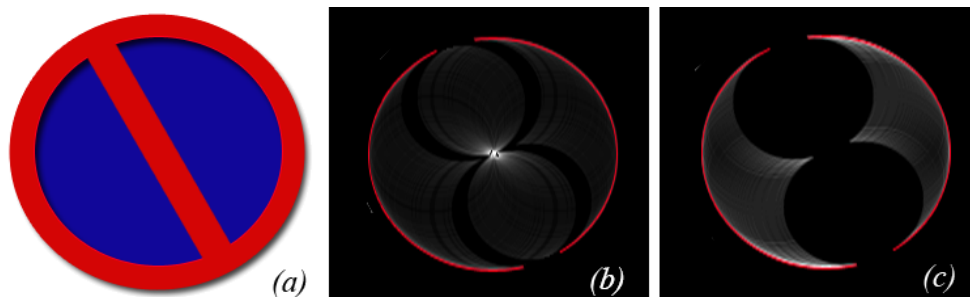
```
Pro (každý významný bodový segment S) proved' {
  Nuluj(AkumulátorStředu[] []);
  Nuluj(AkumulátorR[]);
  Pro (každé dva body (P, Q) z S) proved' {
    AkumulátorStředu[(P.x + Q.x) / 2][(P.y + Q.y) / 2] ++;
  }
  PoziceMaxima := PoziceMaxima(Akumulátor);

  Pro (každé dva body (P, Q) udávající PoziceMaxima) proved' {
    Pomocí (P a Q) vypočítej hodnotu R;
    AkumulátorR[R] ++;
  }
  PoziceMaximaR := PoziceMaxima(AkumulátorR);

  Pro (všechny, které se podílely na vzniku PoziceMaximaR) proved' {
    Získej parametry elipsy metodou nejmenších čtverců;
  }
}
```

Výstup: Parametry detekovaných elips.

Algoritmus 3.1: Oblouková MHT.



Obrázek 3.2: Hledání středu elipsy akumulováním středů úseček mezi body. (a) Dopravní značka Zákaz stání. (b) Histogram akumulace středu na kompletním obraze (rovné byly úsečky filtrovány). (c) Histogram akumulace středu na jednotlivých bodových segmentech.

Druhá fáze hledání zbývajících tří parametrů je již typická pro NHT. Z bodového segmentu náhodně vybíráme tři body a pomocí lineární rovnice 2.20 získáváme koeficienty elipsy. Jsou to hodnoty kovarianční matice, pomocí které získáme z jednotkové kružnice zbývající parametry elipsy a známe tak její kompletní popis. Existence řešení soustavy udává, zdali jsme vybrali body ležící na elipse. Pokud ne, vybíráme další body a výpočet opakujeme. Při dosažení maximálního počtu neúspěšných iterací přecházíme na další segment. Míra odchylky udává, nakolik jsou takto získané parametry přesné, pokud bychom práh chyby snížili, získali bychom přesnější výsledky (za delší čas).

Jednotlivé kroky jsou popsány v algoritmu 3.2.

### 3.3 Detekce metodou nejmenších čtverců

Při původním předpokladu (že každý významný bodový segment tvoří právě jednu elipsu) se nabízí možnost získat parametry elipsy, která má od těchto bodů minimální vzdálenost. Budeme tedy hledat pozici elipsy metodou nejmenších čtverců. Na rozdíl od algoritmu RANSAC budeme výpočet provádět na všech bodech (ne pouze na pěti). Pokud budou přítomny i body šumu, budou jimi výsledné parametry zkresleny. Kdychom hledali na méně pixelech, pravděpodobnost že vybereme body mimo elipsu je nižší, ale případná odchylka by byla podstatně větší. Pro každý segment ověříme výsledek pomocí poměru inliers vůči všem ostatním bodům (v tomto případě všem bodům původního obrazu - viz obrázek 2.4).

Jak vyplývá i z předchozích dvou popisů metod, ideální jsou pro nalezení elipsy úplně nepřerušované hrany. Výsledek totiž zkresluje nejen redundantní body, které k elipse nepatří, ale i chybějící pixely.

Tuto modifikaci je možné dále upravovat - vyřazovat body způsobující velkou chybu (od elipsy příliš vzdálené), nebo provést jiný typ výpočtu (stejně jako Složený rojbustní algoritmus v podkapitole 2.12). Tento krok vynecháváme, pokud klademe důraz především na rychlost. Bylo by nutné provádět výpočet i tam, kde se ve skutečnosti žádná elipsa nenachází. Výsledný postup je popsán algoritmem 3.3.



```

Vstup: Obrázek s pixely náležícími hranám.

Najdi v obraze všechny bodové segmenty;

Pro (každý významný bodový segment S) proved' {
  Získej parametry elipsy metodou nejmenších čtverců;
  Pro (každé dva body (P, Q) z S) proved' {
    AkumulátorStředu[(P.x + Q.x) / 2][(P.y + Q.y) / 2] ++;
  }
  PoziceMaxima := PoziceMaxima(Akumulátor);

  Opakuj {
    Pomocí rovnice vypočítej z T parametry elipsy;
  } dokud (nalezeny parametry elipsy)

  Spočítej Inliers v blízkosti získané elipsy;
  Pokud ((Inliers / všechny body elipsy) > Práh) {
    Přidej do seznamu parametry elipsy;
  }
}

Výstup: Parametry detekovaných elips.

```

Algoritmus 3.2: Kombinace NMHT.

```

Vstup: Obrázek s pixely náležícími hranám.

Najdi v obraze všechny bodové segmenty;

Pro (každý významný bodový segment S) proved' {
  Vypočítej z S parametry elipsy metodou nejmenších čtverců;

  Spočítej Inliers v blízkosti získané elipsy;
  Pokud ((Inliers / všechny body elipsy) > Práh) {
    Přidej do seznamu parametry elipsy;
  }
}

Výstup: Parametry detekovaných elips.

```

Algoritmus 3.3: Detekce MNČ.

## Kapitola 4

# Testování

Při testování metod jsem se zaměřil na hlavní atributy algoritmů detekce elips, kterými jsou úspěšnost, rychlost a přesnost. Je zastoupena rozmanitá škála rychlých, pomalých, přesných i méně robustních postupů, na jejichž výsledcích lze odhadnout směr možného vylepšení, případně se inspirovat z kvalitních partií.

Druhotně jde samozřejmě i o srovnání původních a modifikovaných algoritmů. Výsledky budou průběžně komentovány a v závěrečné podkapitole shrnuty.

Algoritmy byly implementovány s ohledem na přiměřenou časovou náročnost. Neznamená to, že by byla implementace šita na míru konkrétnímu testu. Konstanty byly nastaveny s přihlédnutím ke zdrojovým datům (je rozdíl hledat jednu součástku na továrním pásu či neomezené množství elips na běžných fotografiích).

Implementace proběhla v jazyce C a využil jsem při ní knihovnu grafických funkcí OpenCV (Open Source Computer Vision). Dovolila mi použití optimalizovaných knihovnických funkcí pro práci s obrazem a umožnila mi plně se soustředit na programování samotných algoritmů.

Mezi testované metody patří Účinná detekce, RANSAC, Jednoduchá metoda, Náhodná a Modifikovaná Houghova transformace včetně jejich modifikací (Obloukové MHT a Kombinace NMHT) a Detekce metodou nejmenších čtverců.

### 4.1 Typ dat

Metody jsou testovány na konkrétní úloze hledání dopravního značení. Jde o reálné obrázky, které byly náhodně získávány za různého počasí a jasů. Snažil jsem se získat co nejvyšší počet a nejširší varietu typů, velikostí a úhlů. V databázi je 200 obrázků, každý zachycuje až tři hledané dopravní značky. Jde konkrétně o kruhové typy s červeným lemováním (Zákaz vjezdu, Zákaz stání atd.).

Předzpracování fotek bylo pro všechny algoritmy z důvodu objektivity totožné. Převod barevného obrázku na hranové pixely je popsán v algoritmu 4.1.

Tento test simuluje možné nasazení metod v reálném světě. V našem případě využíváme znalosti o barvě - částečně zmenšíme prohledávaný prostor na barvy kolem červené (i oranžové a hnědé), ale červená auta, střechy domů apod. zůstávají (viz obrázek 4.1). V jiných případech můžeme znát třeba velikost či excentricitu.

Vstup: Barevný obrázek.

Převeď vstupní RGB obrázek na reprezentaci barevným modelem HSV;  
Filtruj výseč červené barvy a jejího okolí;  
Převeď výsledek na stupně šedi;  
Filtruj body šumu;  
Proved' Cannyho hranovou detekci;  
Odstraň drobné body a dlouhé úsečky;

Výstup: Obraz s bílými pixely hran a černým pozadím.

Algoritmus 4.1: Algoritmus předzpracování obrazu.

## 4.2 Systém vyhodnocování

Ruční vyhodnocování by bylo extrémně náročné (až nemožné), proto bylo nutné použít automatický systém. Cílem je co nejvěrněji posoudit, zdali se hledaná a nalezená elipsa shodují.

Na původních obrázcích byly předem anotovány polohy a velikosti elips pomocí tzv. boundingboxů. Tato obalová tělesa mají (v tomto konkrétním případě) tvar obdélníku a stěny rovnoběžné s okraji (viz obrázek 4.2).

Z uvedeného plyne, že v ideálním případě bude totožný levý horní a pravý dolní roh bounding boxů anotované a detekované elipsy (případně budou velmi blízko). Abychom mohli jasně vyjádřit hodnotu překrytí obalových těles, volíme následující vzorec

$$\text{míra shody} = \text{plocha překrytí} / (\text{plocha anotace} + \text{plocha detekce}),$$

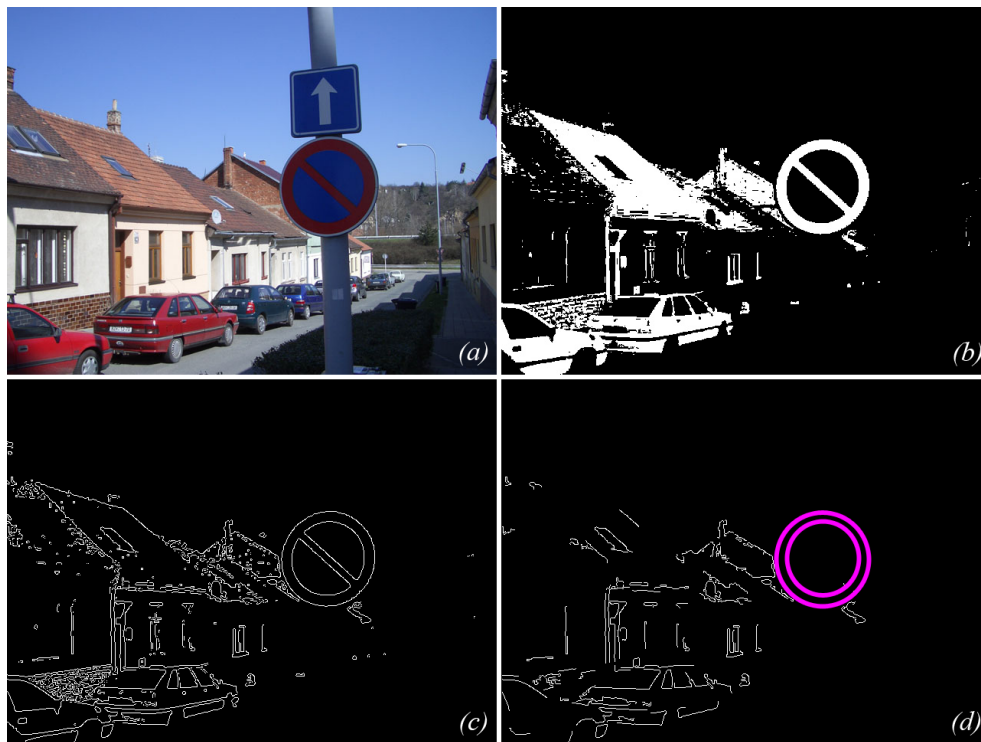
kde míra shody nabývá hodnot v intervalu  $\langle 0; 0.5 \rangle$ . Elipsa je prohlášena za detekovanou při překročení prahu 0.3.

Abychom prahem striktně nedělili elipsy na detekované a nenalezené, uvedeme v testu přesnosti (podkapitola 4.4) rozložení četnosti elips na intervalu  $\langle 0.3; 0.5 \rangle$ .

## 4.3 Úspěšnost detekce

Nezákladnější otázkou při hledání elips je úspěšnost detekce, tedy zdali jsme schopni nalézt elipsu přítomnou v obraze (správná detekce - True Positive). Protipólem je získání nadbytečných elips (chybná detekce - False Positive), které se v původním zdroji nenacházejí. Získáváním a zpracováním počtů těchto elips se bude zabývat i následující test, jelikož výsledek - úspěšnost detekování - je určujícím faktorem použitelnosti daných metod.

Test byl proveden postupně pro různé hladiny prahů od 0.0 (vrací obrovské množství nadbytečných elips) až po 1.0 (nenajde žádnou elipsu). U každého prahu se zaznamenaly správné a špatné detekce. Pokud tyto hodnoty zaneseme do grafu, a spojíme úsečkami, získáváme křivku vyjadřující poměr správně a špatně identifikovaných elips pro různé hodnoty prahů. Na každém konci této křivky je jeden extrém thresholdu. V bodě  $[0; 0]$  je práh nejvyšší (algoritmus nic nedetekuje), tedy 1.0. Na opačném konci je často extrémní výsledek detekce při hodnotě 0.0. Tato křivka se nazývá ROC (z anglického Receiver Operating Characteristic). Z uvedeného lze lehce usoudit, že čím více se křivka blíží levému hornímu rohu



Obrázek 4.1: Předzpracování vstupního obrazu. (a) Původní obraz. (b) Vyselektované okolí červené barvy. (c) Hranová detekce. (d) Ideální výsledek detekce elips umístěný na filtrovaném hranovém obraze.

grafu, tím lepší metodu reprezentuje (souřadnice  $[1;0]$  znamená, že byly nalezeny všechny elipsy a nedošlo ke špatné shodě). V našem případě je vertikální osa normovaná celkovým počtem elips. Výsledky testu jsou uvedeny v grafu na obrázku 4.3.

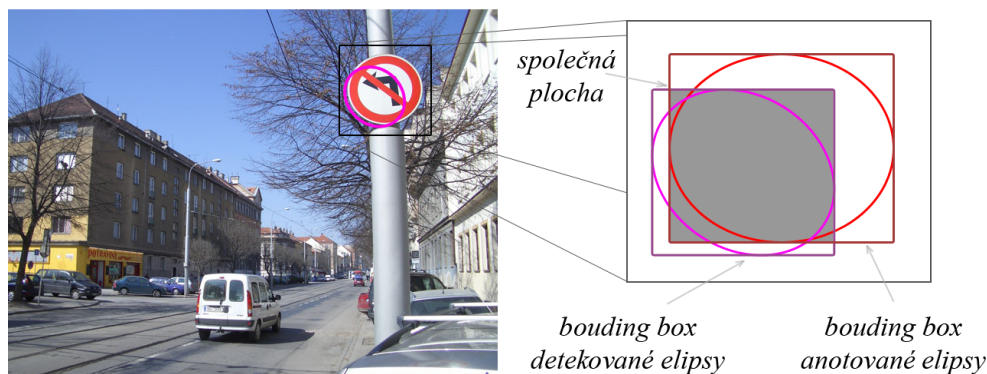
Každý algoritmus využívá pro hledání jiný princip a konkrétní výhody či nevýhody budou komentovány v sekcích vyhrazených pro jednotlivé metody.

## RANSAC

Z grafu na obrázku 4.3 se dá usuzovat, že metoda RANSAC dopadla v testu poměrně dobře. Již v podkapitole 2.6 bylo zmíněno, že čím déle algoritmus prostor prohledává, tím lepší řešení najde. Pokud bychom tedy nechali hledat elipsy déle, získali bychom lepší výsledky (při zrychlení snaopak horší).

Finální nastavení konstant metody je kompromisem mezi rychlostí a přesností. Stejně jako u některých dalších metod platí, že s časem bychom získali kvalitnější výstup. Obecnou pravdou je, že vhodné předzpracování přináší značné zlepšení (zvláště pro RANSAC). Provedeme-li jej na základě znalosti prostředí (například při úloze hledání středového kruhu na fotbalovém hřišti), snížíme počet hranových bodů a tím i výpočetní čas.

RANSAC tedy nebudeme s ostatními metodami srovnávat pouze z údajů o kvalitě jeho detekce. Je nutné vzít v potaz i délku trvání výpočtu, což bude provedeno ve shrnutí komplexních výsledků.



Obrázek 4.2: Automatické vyhodnocení detekce pomocí bounding boxů.

### Účinná detekce

Zpočátku velmi nízká křivka v testu úspěšnosti patří Účinné detekci. Obdobně jako u minulého algoritmu, i u této metody volíme mezi kvalitou a rychlostí. Tentokrát neomezujeme počet výběrů (iterací) náhodných čísel, ale rychlost průchodu hranového obrazu, ze kterého postupně vybíráme všechny (příp. danou část) trojice pixelů.

Čas nezabírá pouze velký počet variací které musíme získat. Značnou zátěží je i výpočet délky vedlejší osy, který musíme pro každý triplet povést. Metoda lépe vyhodnocuje elipsy s podobnými velikostmi os. U kružnic totiž stačí najít dva protilehlé body kdekoliv na obvodu (narozdíl od dvou konkrétních bodů ve vrcholech elipsy). Když algoritmus neprovádíme pro každé dva body, je u kruhu větší šance, že narazíme na dvojici, ze které jsme schopni parametry elipsy vypočítat (resp. že získání parametrů elipsy nebude záležet jen na dvou pixelech z celé elipsy).

### Náhodná Houghova transformace

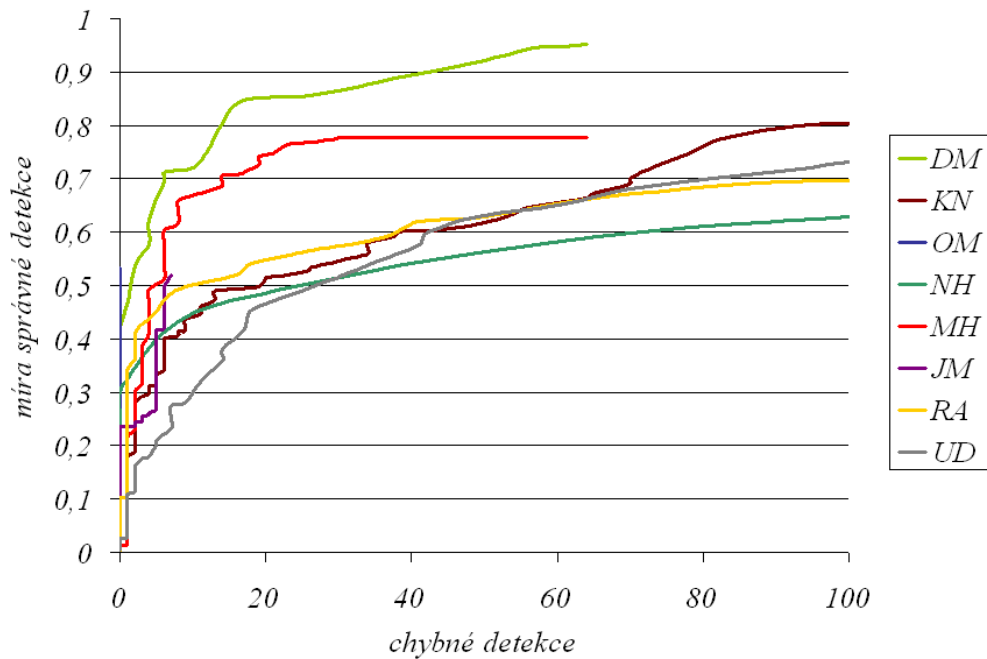
Třetí a poslední metodou jejíž kvalita se zlepšuje úměrně s počtem provedených iterací je Náhodná Houghova transformace. V každém cyklu algoritmu se náhodně vybírají tři body, pro které se určí pozice středu a následně rovnicí spočítají zbývající tři parametry.

Pravděpodobnost, že náhodně narazíme na tři pixely z hrany elipsy je vyšší než u metody RANSAC (pět bodů). Pozice středu je někdy díky velmi hrubému vzorkování (rozlišení 1 pixel) odhadnuta nepřesně. Tím pádem se výpočet pro danou elipsu stává neefektivním, protože jeho výsledek je předchozí chybou značně ovlivněn.

Samotná idea výpočtu je rychlá a jednoduchá. Vychází z ní i Varianta Náhodné Houghovy transformace, která má změněné získávání pozice středu (výsledek varianty bude dále diskutován).

### Modifikovaná Houghova transformace

Značně se levému hornímu rohu grafu na obrázku 4.3 (souřadnici  $[1 : 0]$ ) přiblížila Modifikovaná Houghova transformace. Detekce elipsy touto metodou probíhá za pomoci dvou akumulátorů a dále se pracuje již jen s body, které se podílely na vytvoření maxim v histogramu, proto je výsledkem je poměrně přesná detekce u které lze spatřit zřetelnou hranici, kdy již dále není schopna elipsy správně identifikovat.



Obrázek 4.3: ROC křivka. Zobrazuje poměr správných a špatných identifikací u jednotlivých metod.

Tento algoritmus bych si dovil označit, co se týče kvality detekce za nejlepší z původních testovaných metod. Jeho síla je především ve vyřazení neplatných bodů mimo elipsu, podobně jako je tomu u Jednoduché metody.

### Oblouková MHT

Při testování vznikla poměrně kuriózní ROC křivka, která pouze kopíruje vertikální osu. To znamená, že metoda neidentifikovala žádnou elipsu navíc. Samozřejmě existuje i horní hranice správně nalezených elips, která není nikterak vysoká, nicméně se jedná o unikátní výsledek zapříčiněný mechanismem vyřazování neplatných bodů.

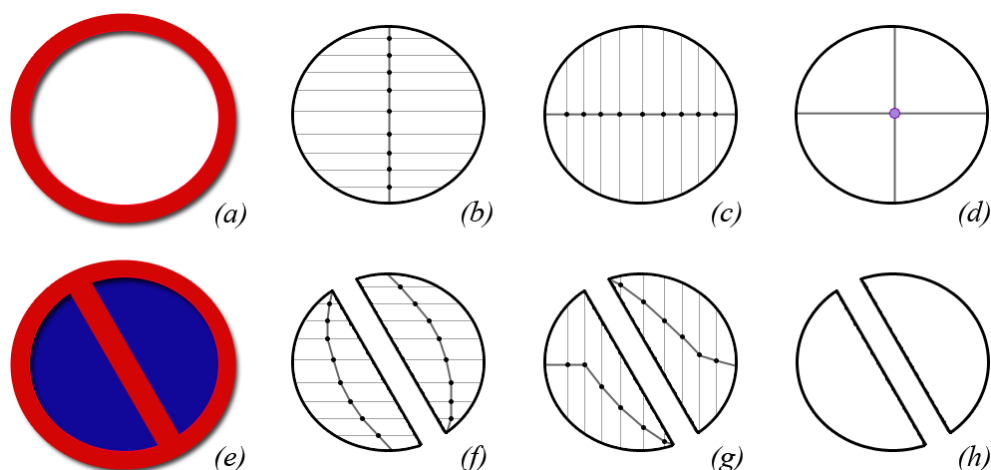
Při detekci elipsy na bodovém segmentu pomocí MHT vynecháváme okolní rušivé body a jak je vidět, dokáže poměrně přesně rozhodnout, že se o elipsu nejedná. Metodu lze dále upravovat ke zvýšení počtu správných identifikací i za cenu drobného zvýšení těch špatných.

Jak již bylo zmíněno, jedná se spíše o demonstraci, že modifikace algoritmu dokáže značně změnit jeho kvalitativní výsledky a jak bude dále popsáno, tak i rychlostní.

### Jednoduchá metoda

Původně jsem tento algoritmus implementoval s procházením ve čtyřech směrech (po 45 stupních). Tento postup ale vykazoval extrémní časovou náročnost, proto jsem jej zjednodušil na dva průchody v pravém úhlu. Toto zjednodušení se na výsledcích detekce projevilo minimálně (narozdíl od časové úspory).

ROC křiva Jednoduché metody na obrázku 4.3 jako dobrý výsledek nevypadá. Je to způsobeno charakterem testovaných dat. Jednoduchá metoda je účinná pouze pokud je elipsa v obraze prázdná (viz konkrétní příklad pro dopravní značku Zákaz vjezdu všech



Obrázek 4.4: Demonstrace algoritmu Jednoduchá metoda. (a) Dopravní značka Zákaz vjezdu. (b) Horizontální průchod obrazu. (c) Vertikální průchod obrazu. (d) Detekované osy šikmé souměrnosti. (e) Dopravní značka Zákaz stání. (f) Horizontální průchod obrazu. (g) Vertikální průchod obrazu. (h) Nedetekované osy šikmé souměrnosti.

vozidel na obrázku 3.2(a). V opačném případě nezískáme křivky, které tvoří osy šikmé symetrie elipsy a elipsu není možné detekovat (viz případ na obrázku 3.2(e)).

Ikdyž jsou při předzpracování odstraňovány dlouhé úsečky, nemusí být kvůli šumu či jiným nepřesnostem detekovány a takto pozůstalé hranové body působí rušivě (viz obrázek 3.2(f) a (g)). Podobný problém jako nadbytečné body uvnitř elipsy působí chybějící pixely na hranách. Pokud je přítomna pouze polovina elipsy, je jisté, že ji Jednoduchá metoda nenajde.

### Kombinace NMHT

Co se týče úspěšnosti detekce, algoritmus vykazuje téměř totožné parametry jako původní NHT. Cílem u této metody bylo zachovat (nebo vylepšit) původní ROC křivku a zvýšit rychlost (viz podkapitola Rychlost detekce 4.5).

### Detekce MNČ

Detekce metodou nejmenších čtverců se svojí ROC křivkou ze všech testovaných metod nejvíce přiblížila ideálu levého horního rohu grafu. Poměrně suverénně vede před ostatními algoritmy pro všechny prahové hodnoty.

Za tuto skutečnost vděčí faktu, že ikdyž se na bodovém segmentu vyskytují pixely, které elipse nenáleží, není jich tolik, aby významně zkreslily výsledek metody nejmenších čtverců. Ke drobné nepřesnosti dojde i při hledání parametrů u neúplné elipsy. Tyto skutečnosti jsou při celkové robustnosti a srovnání s ostatními algoritmy zanedbatelné.

V celkovém hodnocení úspěšnosti detekce všech metod vynikají dva algoritmy. Těmi jsou Detekce metodou nejmenších čtverců a Modifikovaná Houghova transformace. U některých metod by se lepšího výsledku dosáhlo prodloužením doby výpočtu, ale jak bylo zmíněno

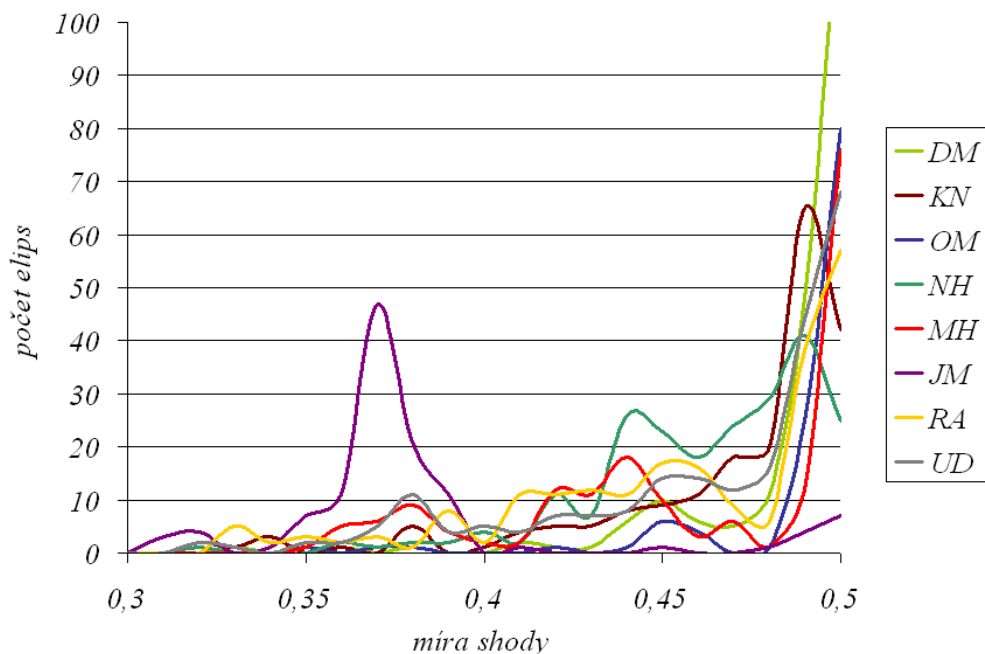
(a konkrétně bude uvedeno v podkapitole 4.3), časový příděl byl stejný, ne-li vyšší jako u ostatních metod.

## 4.4 Přesnost detekce

Tato podkapitola doplňuje výsledky testu úspěšnosti detekce. Ten striktně rozděluje nalezené elipsy na správně a špatně detekované. V části 4.2 byl popsán způsob vyhodnocování detekce, kde parametr shody nabývá hodnot z intervalu  $\langle 0; 0.5 \rangle$ . Čím je parametr vyšší, tím lépe obvod nalezené elipsy kopíruje původní objekt.

Na grafu 4.5 vidíme na horizontální ose výřez intervalu  $\langle 0.3; 0.5 \rangle$  míry shody, tedy jak přesné jsou elipsy prohlášené za shodné. Podle očekávání jsou maximální hodnoty kumulovány u hodnoty maximální shody.

Přesnost detekce je ovlivněna i kvalitou anotace. Získané hodnoty poukazují na poměrně úspěšné určení poloh elips v původních obrazech.



Obrázek 4.5: Přesnost detekce elips.

### Jednoduchá metoda

Výrazně v grafu 4.5 vyniká Jednoduchá metoda. Je to dáno způsobem detekce a konkrétním testem. Pokud se v obraze vyskytnou soustředné elipsy (náš případ), najde vždy pouze tu nejmenší.

Jelikož byly anotovány velikosti vnějších oblouků a Jednoduchá metoda detekuje vnitřní elipsy, je vidět jasný vrcholek poměrů ploch vnějších a vnitřních elips dopravních značek.

Vyhodnocování parametrů lineární rovnicí pomocí tří bodů přímo ovlivňuje (snižuje) přesnost detekce. Náhodná Houghova transformace a Kombinace NMHT proto mají totožně



zalomený konec křivky grafu. Maximum jejich přesnosti tedy neleží v místě úplné shody. Kombinace NMHT je výrazně přesnější než původní NHT.

Stejně jako u kombinace NMHT došlo i v tomto případě ke zlepšení přesnosti metody při implementaci na obloukových segmentech oproti původní MHT.

Detekce metodou nejmenších čtverců vévodí i testu přesnosti, je to díky počítání všech bodů obloukového segmentu oproti třem či pěti pixelům u metod NHT a RANSAC.

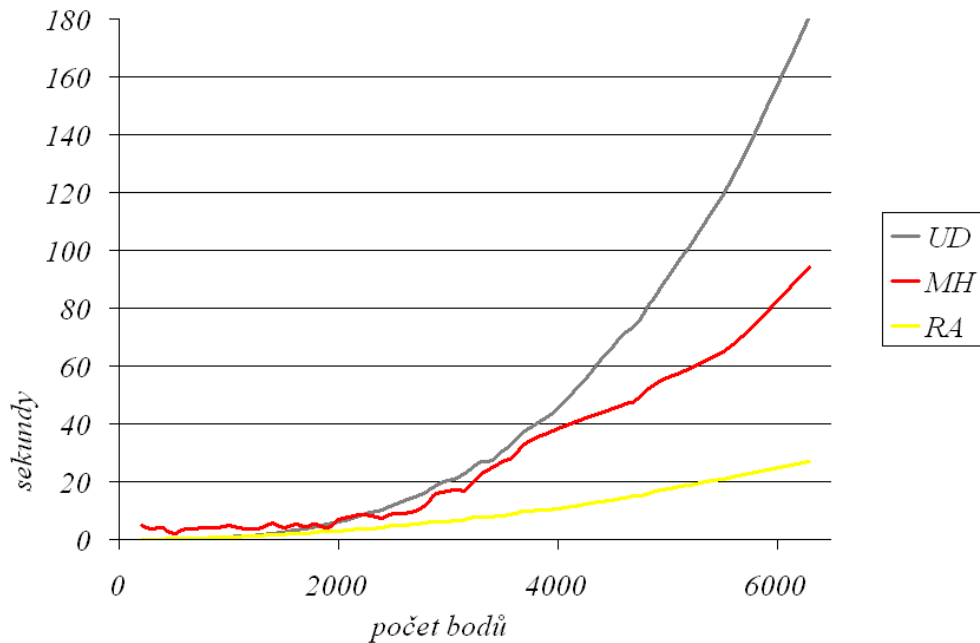
## 4.5 Rychlost detekce

Rychlost detekce byla vyhodnocována u každého obrázku. Měřil se čas hledání vzhledem k počtu bodů obrazu vstupujícího do algoritmu. Metody procházející navzájem všechny body obrazu reagují na zvýšený počet pixelů nelineárně.

Konkrétně budou představeny algoritmy, které svojí rychlostí vybočují od střední hodnoty oběma směry. Zaměříme se také na časový rozdíl výpočtu mezi původními a modifikovanými algoritmy.

### Účinná detekce

Jednou z metod, jejichž délka se zvyšujícím se počtem bodů hranového obrazu rostla, je Účinná detekce (viz graf na obrázku 4.6). V algoritmu se prakticky provádí výpočet nad všemi možnými kombinacemi trojic bodů. Pokud tedy máme obraz, jehož hrany jsou určeny  $N$  pixely, variací získáváme  $N(N - 1)$  variant trojic pro které musíme provést výpočet. Z principu by při tomto množství operací zmiňovaný postup produkoval nadměrně kvalitní výsledky, ale za cenu extrémní časové zátěže. Z tohoto důvodu byly výpočty prováděny pouze nad každým  $k$ -tým bodem.



Obrázek 4.6: Rychlost metod Účinná detekce, Modifikovaná Houghova transformace a RANSAC.

Již z principu Účinné detekce se dá očekávat silná redundance. Poměrně troufale totiž považuje každé dva body za konce úsečky hlavní osy. Pakliže je v obraze pouze jedna elipsa, provádíme neefektivní výpočty pro všech  $N - 2$  pixelů popředí. Některým z nich pochopitelně přejde test délky hlavní osy, stejně tak kontrola vzdálenosti mezi zvažovaným středem elipsy a výběrem třetího bodu.

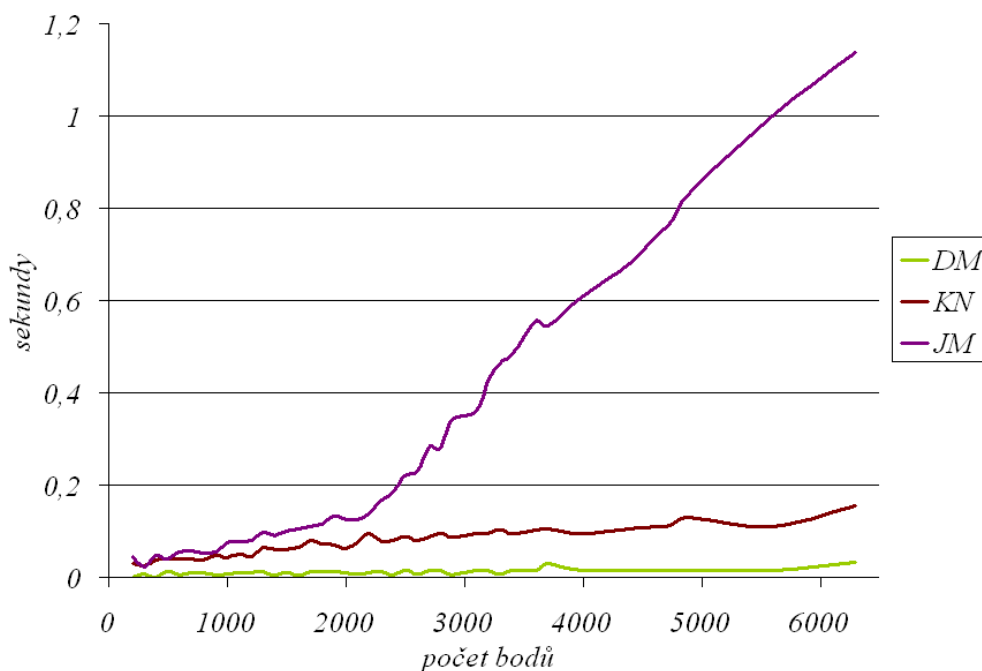
### Modifikovaná Houghova transformace

Z grafu na obrázku 4.6 je patrné, že druhou nejpomalejší metodou testu se stala Modifikovaná Houghova transformace. Je to převážně kvůli opakovanému hledání lokálního maxima, tedy středu v histogramu akumulátoru. Ten je při aplikaci na celý obrázek doslova přeplněný od příspěvků, které nepatří elipse a kterých je obvykle více nežli těch od elipsy. Další výpočty se proto musí provádět i pro falešné středy.

Časovou náročnost ovlivňuje zmíněné hledání středu. Při jeho vyhodnocení záleží na rozložení bodů v obraze. I velké množství rušivých pixelů může akumulovat malou oblast histogramu. V takovém případě se provede pouze jeden neúspěšný test. Naopak časovou zátěží se může stát méně, ale nevhodně rozmístěných hran. Ty způsobí četné vrcholky, které musíme vyhnout na přítomnost elipsy (pochopitelně neúspěšně, tedy neefektivně).

### Detekce MNČ

Z opačného pólu shlíží Detekce metodou nejmenších čtverců hledající elipsy na jednotlivých bodových segmentech. Toto absolutní zjednodušení přináší téměř konstantní křivku grafu. Hodnoty pro jednotlivé obrázky nepřekračují setinu sekundy.



Obrázek 4.7: Rychlost algoritmů Detekce MNČ, Kombinace NMHT a Jednoduché metody.

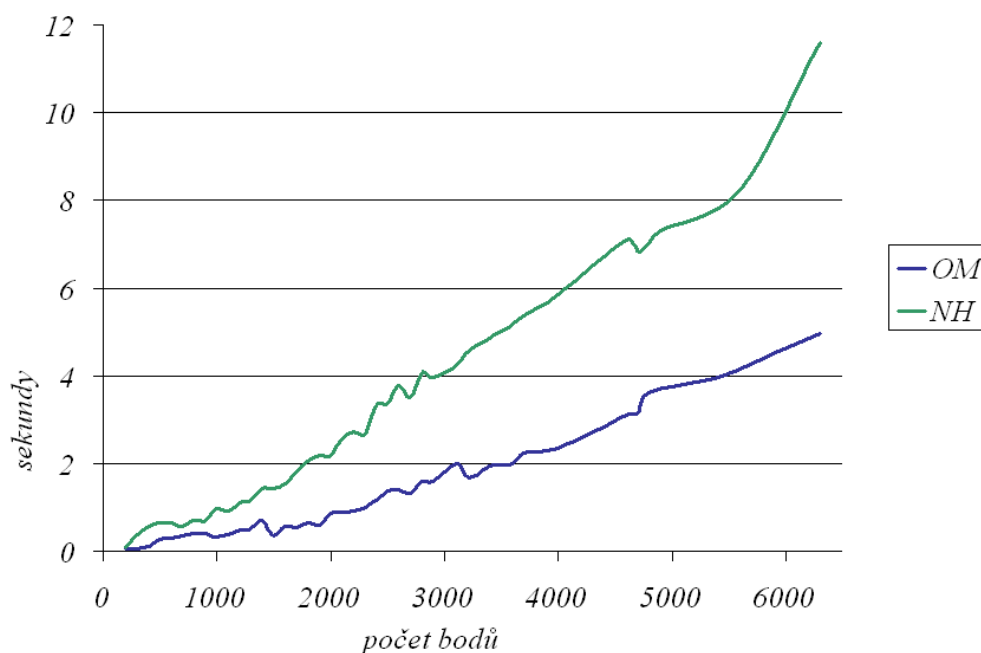
Tato metoda je tedy schopna pracovat i na snímcích přicházejících například z kamery v reálném čase. Při zpracování mohou být zohledněny i další mechanismy pro zkvalitnění

detekce - zbývá na ně čas. Jde například o filtrování obrazu.

Algoritmus lze prohlásit za absolutně nejrychlejší testovanou metodu u které je dále možné obětovat část výpočetního času ke zlepšení kvality (popřípadě dosáhnout optimalizace pro konkrétní úlohu).

### Varianty MHT a NHT

Modifikovaná Houghova transformace prováděná na jednotlivých významných bodových segmentech podle srovnání grafů na obrázcích 4.6 a 4.8 několikanásobně urychlila vyhledávání elips. Je to hlavně z důvodu neopakování průchodů pro možné další elipsy, jelikož u každého bodového segmentu předpokládáme maximálně právě jednu. Díky tomuto omezení vybíráme z akumulátoru středu i z akumulátoru hodnot  $R$  (viz vzorec 2.2) globální maxima.



Obrázek 4.8: Rychlost Varianty MHT a Náhodné Houghovy transformace.

Obdobně došlo k řádovému zvýšení rychlosti i u Kombinace NMHT. Z grafu na obrázku 4.7 je jasně patrné, že jednotlivé detekce nepřekračují délku dvou desetín sekundy. Optimalizace byla dosažena díky obdobnému důvodu jako u Obloukové MHT. Jeden bodový segment reprezentuje maximálně jednu elipsu. Měli bychom být schopni její parametry získat jedním průchodem. Jelikož výpočet parametrů podle rovnice 2.20 nevrátí vždy platné koeficienty, je nutné jej případně zopakovat. Následné iterace se již zjištění pozice středu netýkají.

Tento algoritmus lze prohlásit za druhý nejrychlejší při porovnávání s ostatními.

### RANSAC a Jednoduchá metoda

Algoritmus RANSAC na zvyšující se počet bodů reaguje lineárně. Počet iterací se totiž nastavuje právě podle množství hranových bodů. S jejich narůstáním se snižuje pravděpodobnost

vybrání pěti pixelů náležících elipse. Aby tato šance zůstávala konstantní, musí se zvýšit počet iterací.

Jednoduchá metoda je stejně jako Účinná detekce vyjádřena křivkou s nelineárním průběhem. Vyšší počet bodů znamená více detekovaných pravděpodobných os šikmé symetrie, které se spolu musí porovnat. Zvyšuje se i šance na nalezení průsečíku neplatných os a následných výpočtů, které elipsu neodhalí, ale v grafu na obrázku 4.7 se projeví.

Jak se ukázalo, rychlost detekce je vskutku parametrem algoritmu, který může značně ovlivnit jeho hodnocení. Nelze jej brát v úvahu jako jedinný kvalifikátor. Dokážeme představit uplatnění pomalé, leč robustní metody. Nefunkční byť seberychlejší algoritmus své využití nenažde.

Žebříček algoritmů podle rychlosti detekce je následující.

1. Detekce MNČ
2. Kombinace NMHT
3. Jednoduchá metoda
4. Oblouková MHT
5. Náhodná Houghova transformace
6. RANSAC
7. Modifikovaná Houghova transformace
8. Účinná detekce

## 4.6 Shrnutí

V závěru kapitoly Testování budou diskutovány celkové výsledky složené z hodnocení úspěšnosti, přesnosti a rychlosti detekce.

### Účinná detekce

Nejhorší časový výkon předvedl algoritmus Účinná detekce. Alarmující je hlavně nárůst délky výpočtu při více bodech hran obrazu. Neexceloval ani při měření kvality a proto obsadil pomyslnou poslední příčku.

Metoda se jeví jako daleko vhodnější pro detekci čistých kruhů. Zde by byla větší pravděpodobnost nalézt dva body, které spojuje úsečka jdoucí středem kružnice. V případě elipsy nehledáme libovolné dva body, zmíněnou úsečkou je hlavní osa.

Jak bylo uvedeno, s přidělením dalšího času by se kvalita zlepšila, ale to platí i u jiných metod.

### RANSAC

Metoda RANSAC by se zvýšením počtu iterací dosáhla pochopitelně lepšího výsledku. Projevilo by se to ale i v časové oblasti. Pokud srovnáváme jednotlivé metody se zhruba stejnou či přibližnou délkou trvání, můžeme říci, že metoda RANSAC vykazuje podobné chování.

Na měřeném výsledku se mohla podepsat mírná modifikace vyhledávání více elips. Algoritmus je původně použit pro hledání pouze jednoho primitiva. Hledání se neustále snaží najít lepší parametry s vyšším hodnocením. V našem případě bereme v úvahu všechna nalezená řešení přesahující daný práh, což z původní myšlenky vychází.

### **Náhodná Houghova transformace**

Pro vysoké prahy dává metoda NHT kvalitní výsledky téměř bez chybně identifikovaných elips. Pak dochází k ostrému zlomu, který bychom rádi viděli i u ostatních metod, ale podstatně blíž levému hornímu rohu grafu.

Výpočet parametrů elipsy pomocí rovnice 2.20 nenalezne vždy optimální řešení. Na vině je diskretizace prostoru na rozlišení 1 pixel a zaokrouhlovací chyby. Výsledkem je elipsa, jejíž velikost či pozice nejsou úplně shodné s detekovaným objektem

### **Jednoduchá metoda**

Jednoduchá metoda nachází řešení pouze při absenci hran uvnitř elipsy. V našem testu se často hrany uvnitř elips nachází, proto se tato nevýhoda výrazně projevila.

Váha této metody leží v originalitě jejího řešení. Počet hranových pixelů značně ovlivňuje její rychlost. Ta je i přes tento nedostatek poměrně vysoká. Pokud by se testovaly prázdné elipsy, mohla by obsadit vyšší příčky. Hodnocení je i přes tento nedostatek vysoké.

### **Oblouková MHT**

Při srovnání původní Modifikované Houghovy transformace a její obloukové varianty vidíme, že došlo velkému zrychlení. Projevilo se i značné zlepšení při snížení počtu špatných identifikací, dokonce na hodnotu nula. Cenou za tuto úsporu je odmítání i správných elips.

Metoda je vhodná k další optimalizaci a následnému použití při offline detekcích. Pokud požadujeme záruku téměř stoprocentního odmítnutí nesprávných elips, sáhneme po této modifikaci.

### **Modifikovaná Houghova transformace**

Algoritmus předvedl výborný výsledek v kvalitativním testu, avšak jeho časová náročnost patří mezi nejvyšší. Zrychlení by se dalo dosáhnout omezením výpočtů pro některé falešné vrcholky akumulátoru způsobené koncentrací hranových bodů mimo elipsu.

Může najít uplatnění při offline zpracování stejně jako jeho modifikace.

### **Kombinace NMHT**

Kombinace Náhodné a Modifikované Houghovy transformace přináší velmi zajímavý výsledek v testu rychlosti. Úspěšnost při detekci elips je totiž téměř totožná s ostatními původními metodami, ale rychlost je o dost vyšší. Jde o druhou nejrychlejší metodu v testu. Cíl zrychlení byl tedy splněn i pro tuto modifikaci.

Za zmínku stojí i dvě další (v této práci neuvedené) varianty s NHT. Jednalo se o čistou Náhodnou Houghovu transformaci na jednotlivých bodových segmentech. Druhá modifikace byla složená z hledání středu pomocí tečen a výpočtu zbývajících parametrů jako v MHT. Výsledky obou postupů dosáhly průměrných hodnot.

## Detekce MNČ

V čase i kvalitě je absolutně nejlepší Detekce metodou nejmenších čtverců na bodových segmentech.

Tato metoda obstála i při implementaci na zpracování videosekvencí v reálném čase. Své využití najde tam, kde si můžeme dovolit za cenu nalezení všech elips i drobné chybně identifikované shody.

Metoda je připravena na další úpravy pro konkrétní nasazení. Při detekci na obloukových segmentech a ověření pomocí inliers pracuje velmi dobře. O jeho úspěšnosti, přesnosti a rychlosti není pochyb, jelikož v každém z těchto testů obstál nejlépe.

Testování hovoří jednoznačně pro Detekci metodou nejmenších čtverců. Považuji za nezbytné zdůraznit, že test byl proveden na konkrétních datech s konkrétní implementací algoritmů. Nastavení počtů iterací algoritmů zásadním způsobem ovlivňuje dosažené výsledky. Tuto skutečnost jsem se pokusil zachytit ve slovním komentáři k jednotlivým výstupům testu.

Na konec souhrnu testování je více než nutné zmínit, že nemenší roli než konkrétní algoritmus hraje roli i hranová funkce, kvalita vstupního obrazu a znalost prostředí detekce.

# Kapitola 5

## Závěr

V této práci je teoreticky popsáno dvanáct metod detekce elipsy. Snažil jsem se jednotlivé algoritmy zachytit co nejlépe, se zaměřením na jejich unikátní rysy. Dbal jsem především na srozumitelnost a přehlednost. Pro podrobnější informace o jednotlivých metodách je možné sáhnout po zdrojích uvedených v Literatuře.

Součástí teorie jsou i tři modifikace či inovace, které jsem považoval za vhodné a zajímavé. Vynikají svojí rychlostí a robustností při získávání parametrů elips detekovaných v obraze.

Výsledky praktické části práce jsou uvedeny v kapitole Testování. Implementoval jsem pětici původních metod a všechny tři modifikace. U těchto algoritmů jsem se zaměřil na tři kvalifikátory. Prvním byla rychlost při zpracování obrazů v závislosti na počtu hranových bodů obrazu. Druhým atributem byla úspěšnost hledání elipsy - poměr správně a špatně detekovaných elips. Poslením parametrem byla přesnost detekce elips. Problematiku jsem testoval na konkrétním případě hledání dopravních značek. Zachytil jsem reálné obrázky a zpracoval anotace poloh elips. Touto soustavou byl každý algoritmus automaticky vyhodnocen.

Všechny tři testy demonstrovaly kvalitu jednotlivých metod a prokázaly správný směr vytvořených variant původních postupů. Totiž že využití informace vyšších topologických celků jakými jsou spojené bodové segmenty vede ke zvýšení výkonu. Byla představena i metoda na hledání elips v reálném čase.

Práce na tomto závěrečném projektu vycházela z poznatků získaných při studiu a hledáním zdrojů zaměřených na konkrétní problematiku detekce elipsy. Exkurse do odvětví získávání sémantických informací z obrázků pro mne byla nesmírně poučná a zajímavá. Další možné uplatnění poznatků z této práce vidím v úpravě pro konkrétní využití nebo nasazení v soustavě detektorů.

# Literatura

- [1] M. S. Aguado, A. S. Nixon. A new hough transform mapping for ellipse detection. <http://www.ecs.soton.ac.uk/publications/rj/1995-1996/pdf/isis/asag93r.pdf>, 1996.
- [2] Y. Cheng, Z. Liu. *Efficient technique for ellipse detection using restricted randomized Hough transform*. In: In: In: In: Institute of Image Processing and Pattern Recognition, 2004. ISBN 0-7695-2108-8.
- [3] X. Jianfei, M. Rong. *A Compound and Robust Algorithm for Ellipse Detection*. Zhejiang University of Technology, 2006. ISBN 0-7695-2754-X.
- [4] N. Jie, Y. Kharna. *Fast robust GA-based ellipse detection*. Proceedings of the 17th International Conference, 2004. ISBN 0-7695-2128-2.
- [5] R. Kawaguchi, T. Nagata. *Ellipse Detection Using a Genetic Algorithm*. Oita University, 1998. ISBN 0-8186-8512-3.
- [6] R. A. McLaughlin. Randomized hough transform: Improved ellipse detection with comparison. <http://citeseer.ist.psu.edu/mclaughlin98randomized.html>, 1998.
- [7] R.M. Qiang, J. Haralick. *A statistically efficient method for ellipse detection*. Nevada University, 1999. ISBN 0-7803-5467-2.
- [8] Wikipedia the free encyclopedia. Ransac. <http://en.wikipedia.org/wiki/RANSAC>.
- [9] F. Tsuji, S. Matsumoto. Detection of ellipses by a modified hough transformation. In *IEEE Transactions on Computers*, pages 777–781. IEEE Computer Society, 1996. ISSN 0018-9340.
- [10] A. Weiss, I. Rosenfeld. A simple method of ellipse detection. In *UMD*, 1996.
- [11] J. Yonghong, X. Qiang. *A new efficient ellipse detection method*. Nevada University, 2002. ISBN 0-7695-1695-X.