



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

## ÚSTAV MATEMATIKY

INSTITUTE OF MATHEMATICS

# ZPRACOVÁNÍ OBRAZOVÉ INFORMACE PRO OPTIMÁLNÍ ZAOSTŘENÍ SVĚTELNÉHO MIKROSKOPU

IMAGE PROCESSING FOR OPTIMAL FOCUSING OF LIGHT MICROSCOPE

## BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

PETRA KVAŠŇOVSKÁ

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. PAVEL ŠTARHA, Ph.D.

BRNO 2023

## Zadání bakalářské práce

Ústav: Ústav matematiky  
Studentka: **Petra Kvašňovská**  
Studijní program: Matematické inženýrství  
Studijní obor: bez specializace  
Vedoucí práce: **doc. Ing. Pavel Štarha, Ph.D.**  
Akademický rok: 2022/23

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

### **Zpracování obrazové informace pro optimální zaostření světelného mikroskopu**

#### **Stručná charakteristika problematiky úkolu:**

Manuální ostření obrazu pozorovaného objektu ve světelné mikroskopii je vždy závislé na zkušenostech pozorovatele. K optimálnímu zaostření lze použít různé numerické a stochastické metody. Práce se bude zabývat nalezením parametrů, které by byly vhodným indikátorem optimálního zaostření sledovaného objektu.

#### **Cíle bakalářské práce:**

- Nastudovat a popsat základní numerické metody zpracování obrazové informace.
- Vybrat vhodné parametry pro určení míry ostrosti obrazu.
- Navržené metody otestovat na reálných datech pořízených světelným mikroskopem.

#### **Seznam doporučené literatury:**

PRATT, W. K. Digital Image Processing (Third Edition) PIKS Inside [online]. 3rd ed. New York: Wiley-Interscience, 2001 [cit. 2014-08-07]. ISBN 04-712-2132-5. Dostupné z: <http://www.csupomona.edu/~kding/materials/Digital Image Processing - Third Edition - William K. Pratt.pdf>.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2022/23

V Brně, dne

L. S.

---

doc. Mgr. Petr Vašík, Ph.D.  
ředitel ústavu

---

doc. Ing. Jiří Hlinka, Ph.D.  
děkan fakulty

## **Abstrakt**

Cieľom tejto bakalárskej práce je zvolenie vhodných parametrov pre určenie miery ostrosti obrazu. Práca obsahuje matematický aparát potrebný k riešeniu tejto problematiky. Tak tiež je tu popísaná digitalizácia obrazu potrebná k práci s obrazom na počítači. Výsledkom práce je metóda, pomocou ktorej boli nájdené pozície optimálne zaostrených testovacích snímok objektov pozorovaných svetelným mikroskopom.

## **Summary**

The aim of this bachelor thesis is a selection of appropriate parameters to gauge image sharpness. The thesis contains the mathematical theory needed to solve this issue. Picture digitalization necessary for working with an image on a computer is also described. The outcome of the thesis is a method, using which the positions of optimally sharpened test images of objects observed by a light microscope were found.

## **Klíčová slova**

detekcia hrán, optimálne zaostrenie, spracovanie obrazovej informácie

## **Keywords**

edge detection, optimal focusing, image processing

KVAŠŇOVSKÁ, P. *Zpracování obrazové informace pro optimální zaostření světelného mikroskopu*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2023. 30 s. Vedoucí doc. Ing. Pavel Štarha, Ph.D.

Prehlasujem, že som túto bakalársku prácu vypracovala samostatne pod vedením doc. Ing. Pavla Štarhy, Ph.D. a uviedla som všetky literárne pramene, z ktorých som čerpala.

Petra Kvašňovská



Moja vďaka patrí môjmu vedúcemu doc. Pavlu Štarhovi za cenné pripomienky a rady týkajúce sa tejto bakalárskej práce. Taktiež ďakujem rodine a priateľom za ich nekonečnú podporu.

Petra Kvašňovská

# Obsah

<b>Úvod</b>	<b>2</b>
<b>1 Matematický aparát</b>	<b>3</b>
1.1 Funkcie viacerých premenných . . . . .	3
1.2 Numerické metódy . . . . .	4
1.2.1 Riešenie sústav lineárnych rovníc . . . . .	4
1.2.2 Riešenie nelineárnych rovníc . . . . .	7
1.2.3 Aproximácia funkcie . . . . .	8
1.2.4 Numerické derivovanie . . . . .	10
1.3 Voroného diagram a Delaunayho triangulácia . . . . .	11
1.3.1 Voroného diagram . . . . .	11
1.3.2 Delaunayho triangulácia . . . . .	12
1.3.3 Základné algoritmy . . . . .	14
<b>2 Spracovanie obrazu</b>	<b>15</b>
2.1 Digitalizácia obrazu . . . . .	15
2.2 Fúzia obrazu . . . . .	17
<b>3 Implementácia</b>	<b>18</b>
3.1 Metóda určenia roviny ostrosti . . . . .	18
3.1.1 Detekcia hrán . . . . .	19
3.1.2 Pozícia optimálne zaostrenej snímky . . . . .	21
3.2 Metóda určenia roviny ostrosti pre viac objektov . . . . .	23
3.2.1 Fúzia snímok . . . . .	24
3.2.2 Voroného rozklad fúzie snímok . . . . .	24
3.2.3 Výsledok algoritmu . . . . .	25
<b>Záver</b>	<b>27</b>
<b>Literatúra</b>	<b>28</b>
<b>Zoznam použitých skratiek a symbolov</b>	<b>29</b>
<b>Prílohy</b>	<b>30</b>



# Úvod

Pozorovanie objektov zachytených na mikroskopických snímkach je vždy subjektívne a závisí na skúsenostiach pozorovateľa. Hľadanie optimálne zaostrenej snímky je v takomto prípade náročná úloha. Cieľom tejto bakalárskej práce je objektívne určenie miery ostrosti obrazu za použitia numerických metód.

V práci sú použité dve série testovacích snímok. Prvá séria je tvorená snímkami rovinného objektu, druhá séria obsahuje snímky pelových zrn. Práca sa zaoberá problematikou pri peľovej analýze, kedy sa jednotlivé zrná nachádzajú v rôznych výškach a nie sú teda v jednej hladine ostrosti. Pozorovateľ musí analyzovaný preparát preosť, aby bolo možné pelové zrná relevantne klasifikovať. Ďalší problém nastáva pri uchovaní snímok pozorovaného preparátu, pretože by sme museli archivovať viac snímok zaostrených do rôznych hladín. Metóda umožňuje uchovať len jednu snímku obsahujúcu jednotlivé pelové zrná so správnou hladinou ostrosti.

Prvú kapitolu tvorí matematický aparát potrebný k spracovaniu snímok. Obsahuje popis základných vlastností funkcií dvoch premenných, numerické metódy použité v implementácii a teóriu týkajúcu sa Voroného diagramu a Delaunayho triangulácie. Druhá kapitola je venovaná digitalizácii obrazu a práci s ním. Poslednú kapitolu tvorí implementácia teoretických poznatkov a popis použitých metód.

# 1. Matematický aparát

Bakalárska práca sa zaoberá spracovaním obrazovej informácie. Obraz je vnímaný ako funkcia dvoch premenných a preto je nasledujúca kapitola venovaná funkciám viacerých premenných. Taktiež bude dôležité zavedenie parciálnych derivácií, gradientu funkcie a numerických výpočtov rozkladu obrazu na disjunktné podmnožiny pomocou Voroného diagramu.

## 1.1. Funkcie viacerých premenných

V tejto kapitole sa nachádzajú informácie o funkcii viacerých premenných, jej parciálnych deriváciách a gradiente popísaných v [4].

**Definícia 1.1.** Zobrazenie  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  nazveme *funkciou  $n$  premenných*. Množina  $D(f) := \{[x_1, x_2, \dots, x_n] \in \mathbb{R}^n : \exists y \in \mathbb{R} \text{ tak, že } [x_1, x_2, \dots, x_n, y] \in f\}$  sa nazýva *definičný obor* funkcie  $f$ , množina  $H(f) := \{y \in \mathbb{R} : \exists [x_1, x_2, \dots, x_n] \in \mathbb{R}^n \text{ tak, že } [x_1, x_2, \dots, x_n, y] \in f\}$  sa nazýva *obor hodnôt*. Zapisujeme:  $y = f(x_1, x_2, \dots, x_n)$ , stručne  $y = f(X)$ . Pre  $n = 2 : z = f(x, y)$ , pre  $n = 3 : u = f(x, y, z)$ . Množina  $G(f) := \{[X, f(X)] \in \mathbb{R}^{n+1} : X \in D(f)\}$  sa nazýva *grafom funkcie  $n$  premenných*.

**Veta 1.2.** Nech  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  je funkcia a  $[x_0, y_0]$  je hromadný bod  $D(f)$ . Povieme, že  $f$  má v bode  $[x_0, y_0]$  limitu  $a \in \mathbb{R}^*$ , píšeme  $\lim_{[x,y] \rightarrow [x_0,y_0]} f(x, y) = a$ , ak ku každému okoliu  $O(a)$  existuje okolie  $O(X_0)$  také, že pre každé  $X \in O^*(X_0) \cap D(f)$  platí  $f(X) \in O(a)$ .

**Veta 1.3.** Funkcia  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  má v bode  $[x_0, y_0]$  limitu  $a \in \mathbb{R}$ , ak existuje  $r^* > 0$  a funkcia  $g : (0, \infty) \rightarrow \langle 0, \infty \rangle$  taká, že  $\lim_{r \rightarrow r^+} g(r) = 0$  a  $|f(x_0 + r \cos \varphi, y_0 + r \sin \varphi) - a| \leq g(r)$  pre každé  $r \in (0, r^*)$  a každé  $\varphi \in \langle 0, 2\pi \rangle$ .

*Dôkaz.* Buď  $\varepsilon > 0$  ľubovoľné. Pretože  $\lim_{r \rightarrow r^+} g(r) = 0$ , existuje  $\delta > 0$  také,

že pre  $\forall r \in (0, \delta)$  platí  $0 \leq g(r) < \varepsilon$ . Ak uvažujeme na  $\mathbb{R}^2$  metriku  $\rho_2$ , potom tiež  $\forall [x, y] \in O_{\delta^*}([x_0, y_0])$  platí  $|f(x, y) - a| < \varepsilon$ , tj.

$$\lim_{[x,y] \rightarrow [x_0,y_0]} f(x, y) = a.$$

□

**Definícia 1.4.** Nech  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  je funkcia a  $[x_0, y_0]$  je vnútorný bod  $D(f)$ . Položme  $\varphi(x) = f(x, y_0)$ . Ak má funkcia  $\varphi$  deriváciu v bode  $x_0$ , nazývame túto deriváciu *parciálnou deriváciou funkcie  $f$  podľa premennej  $x$  v bode  $[x_0, y_0]$*  (značíme  $f'_x(x_0, y_0)$ , alebo  $f_x(x_0, y_0)$ , alebo  $\frac{\partial f}{\partial x}(x_0, y_0)$ ). To znamená, že

$$f_x(x_0, y_0) = \lim_{h \rightarrow 0} \frac{\varphi(x_0 + h) - \varphi(x_0)}{h} = \lim_{h \rightarrow 0} \frac{f(x_0 + h, y_0) - f(x_0, y_0)}{h}.$$

Podobne, ak má funkcia  $\phi(y) = f(x_0, y)$  deriváciu v bode  $y_0$ , nazývame túto deriváciu *parciálnou deriváciou funkcie  $f$  podľa premennej  $y$  v bode  $[x_0, y_0]$*  (značíme  $f'_y(x_0, y_0)$ , alebo  $f_y(x_0, y_0)$ , alebo  $\frac{\partial f}{\partial y}(x_0, y_0)$ ).



**Gaussova eliminačná metóda (GEM)**

GEM je základnou priamou metódou riešenia SLR. GEM sa skladá z dvoch častí, z priameho chodu a spätného chodu. V priamom chode sa sústava prevedie na ekvivalentnú sústavu v tvare

$$\mathbf{U}\mathbf{x} = \mathbf{c}, \quad (1.5)$$

kde  $\mathbf{U}$  je matica, ktorá má pod hlavnou diagonálou všetky prvky nulové, tj.  $\mathbf{U} = \{u_{ij}\}_{i,j=1}^n$  a  $u_{ij} = 0$  pre  $i > j$ . Táto matica sa nazýva *horná trojuholníková matica*.

$$\mathbf{U} = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1,n-1} & u_{1n} \\ 0 & u_{22} & u_{23} & \cdots & u_{2,n-1} & u_{2n} \\ 0 & 0 & u_{33} & \cdots & u_{3,n-1} & u_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & u_{n-1,n-1} & u_{n-1,n} \\ 0 & 0 & 0 & \cdots & 0 & u_{nn} \end{pmatrix}.$$

V spätnom chode sa rieši sústava 1.5. Vďaka tomu, že matice  $\mathbf{A}$  a  $\mathbf{U}$  sú regulárne, môžeme vypočítať z poslednej rovnice  $x_n$ , z predposlednej  $x_{n-1}$  a nakoniec z prvej rovnice  $x_1$ .

**Priamy chod GEM**

Pre výpočet priameho chodu GEM označíme  $\mathbf{A}^{(0)} = \mathbf{A}$ ,  $\mathbf{b}^{(0)} = \mathbf{b}$  a prvky matice  $\mathbf{A}^{(0)}$  označíme  $a_{ij}^{(0)} \equiv a_{ij}$  a prvky vektoru  $\mathbf{b}^{(0)}$  položíme  $b_i^{(0)} \equiv b_i$ . Priamy chod GEM bez výberu hlavného prvku označíme GEMz. Výpočet priameho chodu GEMz má  $n-1$  krokov.

**Algoritmus GEMz**

```

for k := 1 to n - 1 do
   $\mathbf{A}^{(k)} := \mathbf{A}^{(k-1)}$ 
   $\mathbf{b}^{(k)} := \mathbf{b}^{(k-1)}$ 
  for i := k + 1 to n do
     $l_{ik} := a_{ik}^{(k)} / a_{kk}^{(k)}$ 
    for j := k + 1 to n do
       $a_{ij}^{(k)} := a_{ij}^{(k)} - l_{ik}a_{kj}^{(k)}$ 
    end
     $b_i^{(k)} := b_i^{(k)} - l_{ik}b_k^{(k)}$ 
  end
end

```

Multiplikátory  $l_{ik}$  zaisťujú, že na pozícii  $(i, k)$  matice  $\mathbf{A}^{(k)}$  vznikne nula:

$$a_{ik}^{(k)} - l_{ik}a_{kk}^{(k)} = 0 \quad \implies \quad l_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}. \quad (1.6)$$

Číslo  $a_{kk}^{(k)}$  sa nazýva *hlavný prvok* alebo *pivot*. Pri výpočte multiplikátorov  $l_{ik}$  môže nastať situácia, kedy  $a_{kk}^{(k)} = 0$ , čo znamená zlyhanie algoritmu GEMz. Tomuto problému je možné sa vyhnúť prehodením  $k$ -tej rovnice s niektorou z ďalších rovníc, ktorá nemá pri premennej  $x_k$  nulový koeficient. Táto úprava sa nazýva výber hlavného prvku. Značenie GEM teda predstavuje algoritmus s výberom hlavného prvku.

## 1.2. NUMERICKÉ METÓDY

### Spätný chod GEMz

```

for  $i := 0$  to  $n - 1$  do
   $x_{n-i} := b_{n-i}$ 
  for  $j := 0$  to  $i - 1$  do
     $x_{n-i} := x_{n-i} - a_{(n-i,n-j)}x_{n-j}$ 
  end
   $x_{n-i} := x_{n-i}/a_{(n-i,n-i)}$ 
end

```

### LU rozklad

Po ukončení priameho chodu je horná trojuholníková matica  $\mathbf{U}$  určená diagonálnymi a naddiagonálnymi prvkami matice  $\mathbf{A}^{(n-1)}$ :

$$u_{ij} := \begin{cases} 0 & \text{pre } j = 1, 2, \dots, i - 1 \\ a_{ij}^{(n-1)} & \text{pre } j = i, i + 1, \dots, n, \end{cases} \quad i = 1, 2, \dots, n. \quad (1.7)$$

Vektor  $\mathbf{c}$  je transformovanou pravou stranou  $\mathbf{b}^{(n-1)}$ :

$$c_i := b_i^{(n-1)}, \quad i = 1, 2, \dots, n. \quad (1.8)$$

Multiplikátory  $l_{ij}$  umiestnime do dolnej trojuholníkovej matice

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ l_{21} & 1 & 0 & \cdots & 0 & 0 \\ l_{31} & l_{32} & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{n-1,1} & l_{n-1,2} & l_{n-1,3} & \cdots & 1 & 0 \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{n,n-1} & 1 \end{pmatrix}. \quad (1.9)$$

Potom

$$\mathbf{A} = \mathbf{LU}. \quad (1.10)$$

Rozklad matice  $\mathbf{A}$  na súčin dolnej trojuholníkovej matice  $\mathbf{L}$  a hornej trojuholníkovej matice  $\mathbf{U}$  sa nazýva **LU rozklad**.

Označením

$$\mathbf{Ly} = \mathbf{b}, \quad \mathbf{Ux} = \mathbf{y} \quad (1.11)$$

je možné sústavu  $\mathbf{LUx} = \mathbf{b}$  efektívne vyriešiť. Určíme najskôr  $\mathbf{y}$  ako riešenie sústavy  $\mathbf{Ly} = \mathbf{b}$  a potom  $\mathbf{x}$  ako riešenie sústavy  $\mathbf{Ux} = \mathbf{y}$ .

Sústavu  $\mathbf{Ly} = \mathbf{b}$  vyriešime tak, že z prvej rovnice vypočítame  $y_1$ , z druhej rovnice  $y_2$ , až nakoniec z poslednej rovnice vypočítame  $y_n$ . Sústavu  $\mathbf{Ux} = \mathbf{y}$  riešime tak, že z poslednej rovnice vypočítame  $x_n$ , z predposlednej  $x_{n-1}$  až nakoniec z prvej rovnice vypočítame  $x_1$ . **LU rozklad** je označovaný ako priamy chod a výpočet podľa 1.11 sa nazýva spätný chod. Výhodou **LU rozkladu** je možnosť riešenia rovnakej sústavy rovníc s rôznymi pravými stranami  $\mathbf{b}$ .

### LU rozklad s čiastočným výberom hlavného prvku

Čiastočný výber hlavného prvku je úprava GEM, ktorá zaisťuje, že absolútna hodnota multiplikátorov je menšia alebo rovná jednej.

**Spätný chod pomocou LU rozkladu**

```

for  $i := 1$  to  $n$  do
  |
  |  $y_i := (b_i - \sum_{j=1}^{i-1} l_{ij}y_j)/l_{ii}$ 
  |
end
for  $i := n$  downto  $1$  do
  |
  |  $x_i := (y_i - \sum_{j=i+1}^n u_{ij}x_j)/u_{ii}$ 
  |
end

```

V  $k$ -tom kroku eliminácie je vybraný prvok matice  $\mathbf{A}$  s najväčšou absolútnou hodnotou v zatiaľ neeliminovanej časti  $k$ -teho stĺpca matice  $\mathbf{A}^{(k-1)}$ . Riadkový index, pre ktorý táto podmienka platí, nazveme  $r$ . Potom sa navzájom vymenia  $k$ -ta a  $r$ -ta rovnica, a  $k$ -ty a  $r$ -tý prvok vektoru  $\mathbf{b}$ . Výmena riadkov je aplikovaná aj na *permutačnú maticu*  $\mathbf{P}$ , ktorá je na začiatku rovná jednotkovej matici. Dolnú trojuholníkovú maticu  $\mathbf{L}$  získame tak, že do hlavnej diagonály dáme jednotky a poddiagonálne prvky prevezmeme z výslednej matice  $\mathbf{A}^{(n-1)}$ . Hornú trojuholníkovú maticu  $\mathbf{U}$  dostaneme z diagonálnych a naddiagonálnych prvkov výslednej matice  $\mathbf{A}^{(n-1)}$ . Výsledkom LU rozkladu s čiastočným výberom hlavného prvku nie sú len matice  $\mathbf{L}$  a  $\mathbf{U}$ , ale aj matica  $\mathbf{P}$ , pričom platí

$$\mathbf{LU} = \mathbf{PA}.$$

### 1.2.2. Riešenie nelineárnych rovníc

Pri riešení nelineárnej rovnice  $f(x) = 0$  vo všeobecnosti nevieme korene tejto rovnice vyjadriť vzorcom. Na jej riešenie preto používame iteračné metódy, pri ktorých tvoríme postupnosť  $x_0, x_1, x_2, \dots$ , ktorá konverguje ku koreňu  $x^*$ .

#### Metóda bisekcie

**Veta 1.8.** (prvá Bolzanova veta). Nech je funkcia  $f(x)$  spojitá na  $\langle a, b \rangle$  a nech platí  $f(a)f(b) < 0$ . Potom existuje  $c \in (a, b)$  také, že  $f(c) = 0$ .

Metóda bisekcie, viď obrázok 1.1, je určená k počiatočnej aproximácii koreňa rovnice  $f(x) = 0$ . Predpokladáme, že spojitá funkcia  $f(x)$  má v koncových bodoch intervalu  $(a_0, b_0)$  opačné znamienka, tj. platí  $f(a_0)f(b_0) < 0$ . Z prvej Bolzanovej vety vyplýva, že v danom intervale leží koreň funkcie  $f(x)$ .

Tvoríme postupnosť intervalov  $(a_1, b_1) \supset (a_2, b_2) \supset (a_3, b_3) \supset \dots$  tak, aby obsahovali koreň. Bod  $x_{k+1} = \frac{1}{2}(a_k, b_k)$  je stred intervalu  $(a_k, b_k)$ ,  $k = 0, 1, \dots$ . Ak platí, že  $f(x_{k+1}) = 0$ , potom  $x^*$  je nájdený koreň. Ak  $f(x_{k+1}) \neq 0$  položíme

$$(a_{k+1}, b_{k+1}) = \begin{cases} (a_k, x_{k+1}), & \text{ak } f(a_k)f(x_{k+1}) < 0, \\ (x_{k+1}, b_k), & \text{ak } f(a_k)f(x_{k+1}) > 0. \end{cases} \quad (1.12)$$

Pre nový interval  $(a_{k+1}, b_{k+1})$  platí  $f(a_{k+1})f(b_{k+1}) < 0$ , takže každý interval  $(a_k, b_k)$  obsahuje koreň. Po  $k$  krokoch je koreň v intervale  $I_k := (a_k, b_k)$  dĺžky

$$|I_k| = b_k - a_k = 2^{-1}(b_{k-1} - a_{k-1}) = \dots = 2^{-k}(b_0 - a_0).$$

## 1.2. NUMERICKÉ METÓDY

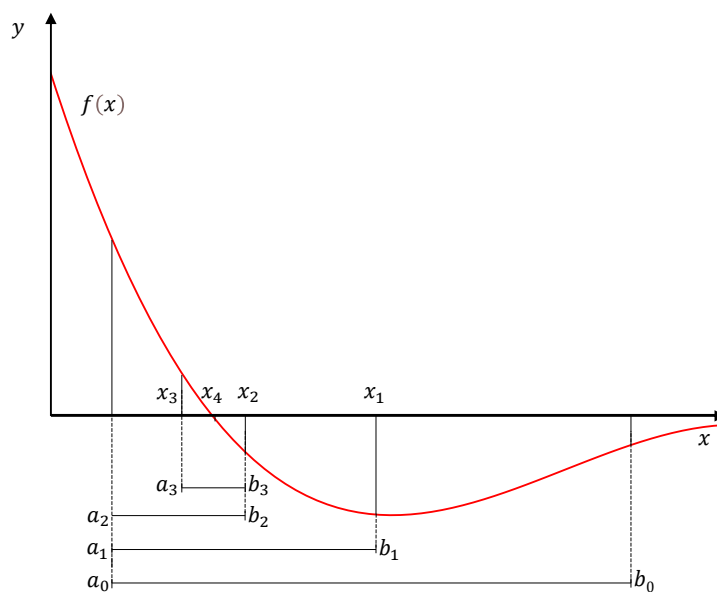
Stred  $x_{k+1}$  intervalu  $(a_k, b_k)$  aproximuje koreň  $x^*$  s chybou

$$|x_{k+1} - x^*| \leq \frac{1}{2}(b_k - a_k) = 2^{-k-1}(b_0 - a_0).$$

Pre  $k \rightarrow \infty$  platí  $|I_k| \rightarrow 0$  a  $x_k \rightarrow x^*$ .

Pretože metóda bisekcie konverguje pomaly, môžeme použiť spresňujúce metódy, napríklad Newtonovu metódu.

**Poznámka 1.9.** V bakalárskej práci bola použitá metóda bisekcie, pretože nebolo potrebné dosahovať veľké presnosti. Použitie metódy bisekcie je jednoduchším riešením, nakoľko pri Newtonovej metóde je potrebné overenie splnenia Fourierových podmienok pre konvergenciu metódy.



Obr. 1.1: Metóda bisekcie

### 1.2.3. Aproximácia funkcie

Aproximáciou funkcie  $y(x)$  rozumieme jej nahradenie funkciou  $\varphi(x)$  tak, že funkcie  $y(x)$  a  $\varphi(x)$  sú si v istom zmysle blízke. Píšeme  $\varphi(x) \approx y(x)$ .

#### Metóda najmenších štvorcov

Metóda najmenších štvorcov je taká aproximácia, pri ktorej  $\varphi(x)$  „prekladáme“ medzi zadanými bodmi  $[x_i, y_i]$  tak, aby „vzdialenosť“ funkcií  $y(x)$  a  $\varphi(x)$  bola v istom zmysle minimálna. Funkcia  $\varphi(x)$  pritom nemusí prechádzať zadanými bodmi. Metódu najmenších štvorcov riešime nasledovne.

Nech  $x$  je nezávislá premenná, napríklad čas, a  $y(x)$  je neznáma funkcia premennej  $x$ , ktorú chceme aproximovať. Pri  $m$  pozorovaniach približne zmeriame hodnoty  $y$  pre navzájom rôzne hodnoty  $x$

$$y_i \approx y(x_i), \quad i = 1, 2, \dots, m,$$

pričom  $\approx$  vyjadruje približnú rovnosť. Cieľom je modelovať  $y(x)$  lineárnou kombináciou  $n$  bázových funkcií pre  $n \leq m$ :

$$y(x) \approx \beta_1 \varphi_1(x) + \beta_2 \varphi_2(x) + \dots + \beta_n \varphi_n(x) =: R_n(x).$$

Funkcia  $R_n(x)$  sa v štatistike nazýva *lineárna regresná funkcia*. Bázové funkcie navrhujeme podľa očakávaného priebehu neznámej funkcie  $y(x)$ . Určujeme parametre  $\beta_1, \beta_2, \dots, \beta_n$  a to tak, aby

$$y_i \approx R_n(x_i), \quad i = 1, 2, \dots, m, \quad \text{maticovo } \mathbf{y} \approx \mathbf{A}\boldsymbol{\beta},$$

kde  $\mathbf{y} = (y_1, y_2, \dots, y_m)^T$  sú namerané dáta,  $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_m)^T$  je vektor neznámych parametrov a  $\mathbf{A}$  sa nazýva *návrhová matica*,

$$\mathbf{A} = \begin{pmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \cdots & \varphi_n(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & \cdots & \varphi_n(x_2) \\ \vdots & \vdots & \cdots & \vdots \\ \varphi_1(x_m) & \varphi_2(x_m) & \cdots & \varphi_n(x_m) \end{pmatrix} \equiv (\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \dots, \boldsymbol{\varphi}_n).$$

Vektor  $\boldsymbol{\varphi}_i = (\varphi_i(x_1), \varphi_i(x_2), \dots, \varphi_i(x_m))$ ,  $i = 1, 2, \dots, n$ , je  $i$ -ty stĺpec matice  $\mathbf{A}$ . *Rezíduá* sú rozdiely medzi pozorovanými  $y_i$  a modelovanými hodnotami  $R_n(x_i)$ :

$$r_i = y_i - R_n(x_i) = y_i - \sum_{j=1}^n \varphi_j(x_i) \beta_j \equiv y_i - \sum_{j=1}^n a_{ij} \beta_j, \quad i = 1, 2, \dots, m,$$

kde  $a_{ij} = \varphi_j(x_i)$ . V maticovom zápise

$$\mathbf{r} = \mathbf{y} - \mathbf{A}\boldsymbol{\beta}.$$

Parametre  $\beta_i$  určujeme tak, aby rezíduá boli čo najmenšie. Metódu najmenších štvorcov dostaneme, keď minimalizujeme súčet štvorcov rezíduí:

$$\|\mathbf{r}\|_2^2 = \sum_{i=1}^m r_i^2 \rightarrow \min. \quad (1.13)$$

Označíme

$$F(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{A}\boldsymbol{\beta}\|_2^2 \equiv \|\mathbf{r}\|_2^2.$$

Riešenie minimalizačnej úlohy 1.13 musí spĺňať podmienku pre extrém:

$$\frac{\partial F(\boldsymbol{\beta})}{\partial \beta_k} = \frac{\partial}{\partial \beta_k} \sum_{i=1}^m (y_i - \sum_{j=1}^n a_{ij} \beta_j)^2 = 0, \quad k = 1, 2, \dots, n.$$

Po derivácii dostaneme

$$\frac{\partial F(\boldsymbol{\beta})}{\partial \beta_k} = 2 \sum_{i=1}^m (y_i - \sum_{j=1}^n a_{ij} \beta_j) (-a_{ik}) = 0, \quad k = 1, 2, \dots, n$$

a následne

$$\sum_{j=1}^n \left( \sum_{i=1}^m a_{ik} a_{ij} \right) \beta_j = \sum_{i=1}^m a_{ik} y_i, \quad k = 1, 2, \dots, n,$$



## 1.2. NUMERICKÉ METÓDY

čo možno zapísať ako

$$\mathbf{A}^T \mathbf{A} \boldsymbol{\beta} = \mathbf{A}^T \mathbf{y}.$$

Sústava 1.2.3 sa nazýva *normálna sústava rovníc*. Keď sú stĺpce matice  $\mathbf{A}$  lineárne nezávislé, je matica  $\mathbf{G} := \mathbf{A}^T \mathbf{A}$  pozitívne definitná, takže riešenie  $\boldsymbol{\beta}^*$  normálnej sústavy rovníc minimalizuje  $F(\boldsymbol{\beta}) \equiv \|\mathbf{r}\|_2^2$  a je riešením úlohy 1.13:

$$\|\mathbf{y} - \mathbf{A}\boldsymbol{\beta}^*\|_2^2 = \min_{\boldsymbol{\beta} \in \mathbb{R}^n} \|\mathbf{y} - \mathbf{A}\boldsymbol{\beta}\|_2^2.$$

Vyjadrením normálnej sústavy rovníc pomocou vektorov  $\boldsymbol{\varphi}_i$ , dostaneme

$$\begin{pmatrix} (\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_1) & (\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2) & \cdots & (\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_n) \\ (\boldsymbol{\varphi}_2, \boldsymbol{\varphi}_1) & (\boldsymbol{\varphi}_2, \boldsymbol{\varphi}_2) & \cdots & (\boldsymbol{\varphi}_2, \boldsymbol{\varphi}_n) \\ \vdots & \vdots & \ddots & \vdots \\ (\boldsymbol{\varphi}_n, \boldsymbol{\varphi}_1) & (\boldsymbol{\varphi}_n, \boldsymbol{\varphi}_2) & \cdots & (\boldsymbol{\varphi}_n, \boldsymbol{\varphi}_n) \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix} = \begin{pmatrix} (\boldsymbol{\varphi}_1, \mathbf{y}) \\ (\boldsymbol{\varphi}_2, \mathbf{y}) \\ \vdots \\ (\boldsymbol{\varphi}_n, \mathbf{y}) \end{pmatrix}, \quad (1.14)$$

kde

$$(\boldsymbol{\varphi}_k, \boldsymbol{\varphi}_j) = \sum_{i=1}^m \varphi_k(x_i) \varphi_j(x_i) \quad \text{a} \quad (\boldsymbol{\varphi}_k, \mathbf{y}) = \sum_{i=1}^m \varphi_k(x_i) y_i$$

sú skalárne súčiny vektorov  $\boldsymbol{\varphi}_k, \boldsymbol{\varphi}_j$  a  $\boldsymbol{\varphi}_k, \mathbf{y}$ . Matica  $\mathbf{G}$  sústavy 1.14 sa nazýva *Gramova matica* sústavy vektorov  $\{\boldsymbol{\varphi}_j\}_{j=1}^n$ .

### 1.2.4. Numerické derivovanie

Približný výpočet derivácie  $f'(x)$  má zmysel vtedy, keď

- pre dané  $x$  vieme získať odpovedajúcu hodnotu  $y = f(x)$ , ale vyjadrenie funkcie  $f(x)$  nepoznáme a preto vzorec pre  $f'(x)$  nevieme napísať
- funkcia  $f(x)$  je príliš zložitá a jej výpočet je prácny
- hodnoty funkcie  $f(x)$  poznáme len v niekoľkých tabuľkových bodoch

V týchto prípadoch nahradíme funkciu  $f(x)$  vhodnou aproximáciou  $\varphi(x)$  a hodnotu derivácie  $\varphi'(x)$  považujeme za približnú hodnotu derivácie  $f'(x)$ . Podobným postupom nahradíme  $f^{(k)}(x)$  pomocou  $\varphi^{(k)}(x)$  pri výpočte vyšších derivácií.

Nasledujúce formule sú založené na derivovaní Lagrangeovho interpolačného polynómu  $P_n(x)$ , tj.  $f'(x)$  aproximujeme pomocou  $P'_n(x)$ .

#### Chyba aproximácie v uzlovom bode

Nech  $f \in C^{n+1}(a, b)$ , kde  $a$  je najmenší a  $b$  je najväčší z uzlov interpolácie. Potom pre chybu  $f'(x_s) - P'_n(x_s)$  v niektorom uzle  $x_s$  platí

$$f'(x_s) - P'_n(x_s) = \frac{f^{(n+1)}(\xi_s)}{(n+1)!} \omega'_{n+1}(x_s), \quad (1.15)$$

kde  $\omega_{n+1}(x) = (x - x_0)(x - x_1) \cdots (x - x_n)$  a  $\xi_s$  je nejaký bod z intervalu  $(a, b)$ .

**Prehľad užitočných vzorcov.**

Uvažujme prípad, kedy uzly  $x_i$  sú *ekvidistantné* s krokom  $h$ , tj,  $x_i = x_0 + ih, i = 1, 2, \dots, n$ . Uzol  $x_s$  v ktorom počítame približnú hodnotu derivácie, označíme ako  $x$ . Ostatné uzly vyjadrujeme pomocou  $x$  ako  $x + h$  a  $x - h$ .

Pomocou vzťahu 1.15 dostaneme:

prvá dopredná diferenciacia:

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{1}{2}hf''(\xi), \quad \xi \in (x, x+h), \quad (1.16)$$

prvá spätná diferenciacia:

$$f'(x) = \frac{f(x) - f(x-h)}{h} + \frac{1}{2}hf''(\xi), \quad \xi \in (x-h, x), \quad (1.17)$$

prvá centrálna diferenciacia:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{1}{6}h^2f'''(\xi), \quad \xi \in (x-h, x+h). \quad (1.18)$$

**Numerický výpočet parciálnej derivácie.**

Pri výpočte parciálnej derivácie podľa premennej  $x_i$  si ostatné premenné  $x_j \neq x_i$  nevyšímame a niektorý z vyššie uvedených vzťahov aplikujeme len na  $x_i$ . Pre doprednú diferenciaciu platí

$$\begin{aligned} \frac{\partial f(x_1, x_2)}{\partial x_1} &\approx \frac{f(x_1+h, x_2) - f(x_1, x_2)}{h}, \\ \frac{\partial f(x_1, x_2)}{\partial x_2} &\approx \frac{f(x_1, x_2+h) - f(x_1, x_2)}{h} \end{aligned}$$

a pomocou centrálnej diferencie dostaneme

$$\begin{aligned} \frac{\partial f(x_1, x_2)}{\partial x_1} &\approx \frac{f(x_1+h, x_2) - f(x_1-h, x_2)}{2h}, \\ \frac{\partial f(x_1, x_2)}{\partial x_2} &\approx \frac{f(x_1, x_2+h) - f(x_1, x_2-h)}{2h}. \end{aligned}$$

**1.3. Voroného diagram a Delaunayho triangulácia**

V tejto kapitole sa nachádzajú základné vlastnosti Voroného diagramu, Delaunayho triangulácie a možnosti ich zostrojenia. Zostrojenie Voroného diagramu je možné v rôznych metrikách, ďalej však budeme uvažovať euklidovskú metriku. Definície je možné nájsť v publikácii [1].

**1.3.1. Voroného diagram**

Uvažujme  $S$  ako množinu  $n \geq 3$  bodov  $p, q, r, \dots$  v euklidovskej rovine  $\mathbb{R}^2$ . Pre body  $p = [p_1, p_2]$  a  $x = [x_1, x_2]$  platí vzdialenosť

$$\rho(p, x) = \sqrt{(p_1 - x_1)^2 + (p_2 - x_2)^2}.$$

### 1.3. VORONÉHO DIAGRAM A DELAUNAYHO TRIANGULÁCIA

Množinu bodov v  $\mathbb{R}^2$ , pre ktoré platí, že ich vzdialenosť s bodom  $p$  je rovnaká ako vzdialenosť s bodom  $q$ , nazveme *bisektor*  $B(p, q)$ . Bisektor je os úsečky  $pq$ . Oddeluje polrovinu

$$D(p, q) = \{x \mid \rho(p, x) \leq \rho(q, x)\}$$

bližšie k bodu  $p$  od polroviny  $D(q, p)$  bližšie k bodu  $q$ .

Oblasť prislúchajúca bodu  $p \in S$  sa nazýva *Voroného oblasť* alebo *Voroného bunka*, značíme  $\nu(p, S)$ , a bod  $p$  nazveme *jadrom bunky*  $\nu(p, S)$ . Voroného bunka  $\nu(p, S)$  je prienik  $n - 1$  polrovín  $D(p, q)$ , kde  $q \neq p$  sú body množiny  $S$ . Zapisujeme

$$\nu(p, S) = \bigcap_{q \in S, q \neq p} D(p, q).$$

Voroného oblasť  $\nu(p, S)$  je tvorená bodmi  $x \in \mathbb{R}^2$ , ktorých vzdialenosť k bodu  $p$  je menšia, než vzdialenosť k ostatným bodom z  $S$ .

Spoločná hranica dvoch Voroného oblastí sa nazýva *Voroného hrana*, ak obsahuje viac ako jeden bod. Zjednotenie všetkých Voroného hrán sa nazýva *Voroného diagram* množiny  $S$ , značíme  $V(S)$ .

#### 1.3.2. Delaunayho triangulácia

Triangulácia množiny  $S$  je rovinný graf s vrcholmi  $S$  a hranami, pre ktoré platí, že ich počet je maximálny, a nie je možné pridať ďalšiu hranu tak, aby nekřížila ostatné hrany. *Delaunayho trianguláciu* získame spojením bodov  $p, q \in S$  úsečkami tak, že existuje kružnica prechádzajúca bodmi  $p$  a  $q$  neobsahujúca žiaden ďalší bod. Delaunayho trianguláciu značíme  $DT(S)$ . Hrany  $DT(S)$  sa nazývajú *Delaunayho hrany*.

#### Vlastnosti Delaunayho triangulácie

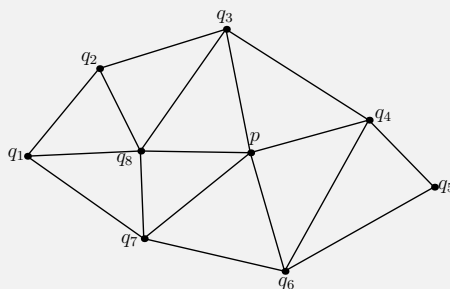
Delaunayho triangulácia je duálna úloha k Voroného diagramu, čo znamená, že z jedného je možné zostrojiť druhé.

Dva body patriace množine  $S$  sú spojené Delaunayho hranami, ak ich Voroného oblasti zdieľajú Voroného hranu.

$DT(S)$  maximalizuje minimálny uhol v každom trojuholníku.

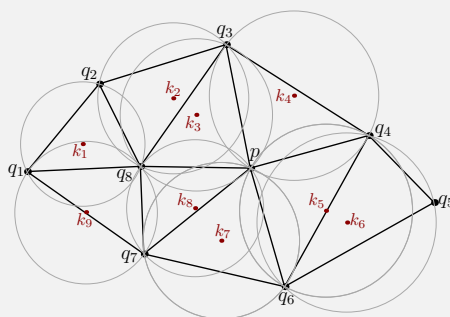
Vo vnútri kružnice opísanej ľubovoľnému trojuholníku  $DT(S)$  neleží žiaden ďalší bod  $S$ .

**Príklad 1.10.** Z Delaunayho triangulácie množiny bodov na obrázku 1.2 zostrojíme Voroného diagram.



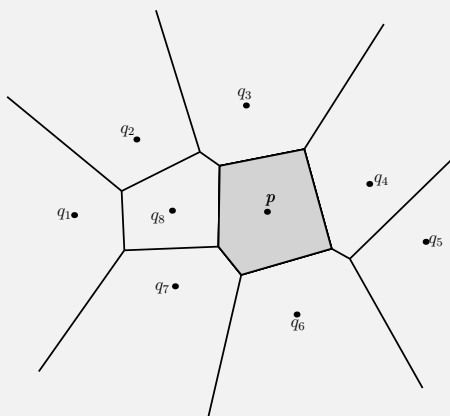
Obr. 1.2: Delaunayho triangulácia

Ako ukazuje obrázok 1.3, vytvoríme kružnice opísané trojuholníkom Delaunayho triangulácie so stredmi  $k_1, k_2, \dots, k_9$ .



Obr. 1.3: Delaunayho triangulácia a stredy opísaných kružníc

Spojením stredov opísaných kružníc získame Voroného diagram, viď obrázok 1.4. Pre každý bod  $x$  z oblasti prislúchajúcej bodu  $p$  platí  $\rho(x, p) \leq \rho(x, q_i)$ , kde  $i = 1, 2, \dots, 8$ .



Obr. 1.4: Voroného diagram

### 1.3. VORONÉHO DIAGRAM A DELAUNAYHO TRIANGULÁCIA

#### 1.3.3. Základné algoritmy

Uvedieme základné algoritmy zostrojenia Voroného diagramu a jeho duálnej úlohy, Delaunayho triangulácie. Pre zjednodušenie budeme uvažovať  $n$  bodov množiny  $S$  tak, že žiadne 4 body neležia na jednej kružnici, a žiadne 3 nie sú kolineárne.

##### Inkrementálny algoritmus

Inkrementálny algoritmus je založený na postupnom pridávaní bodov. Na začiatku je Voroného diagram vytvorený z dvoch bodov, v takom prípade je  $V(S)$  bisektorom daných bodov. Ako vidieť na obrázku 1.5a, pridávaním ďalších bodov sa tvoria nové hrany a predošlé, neplatné hrany, sú vymazané.

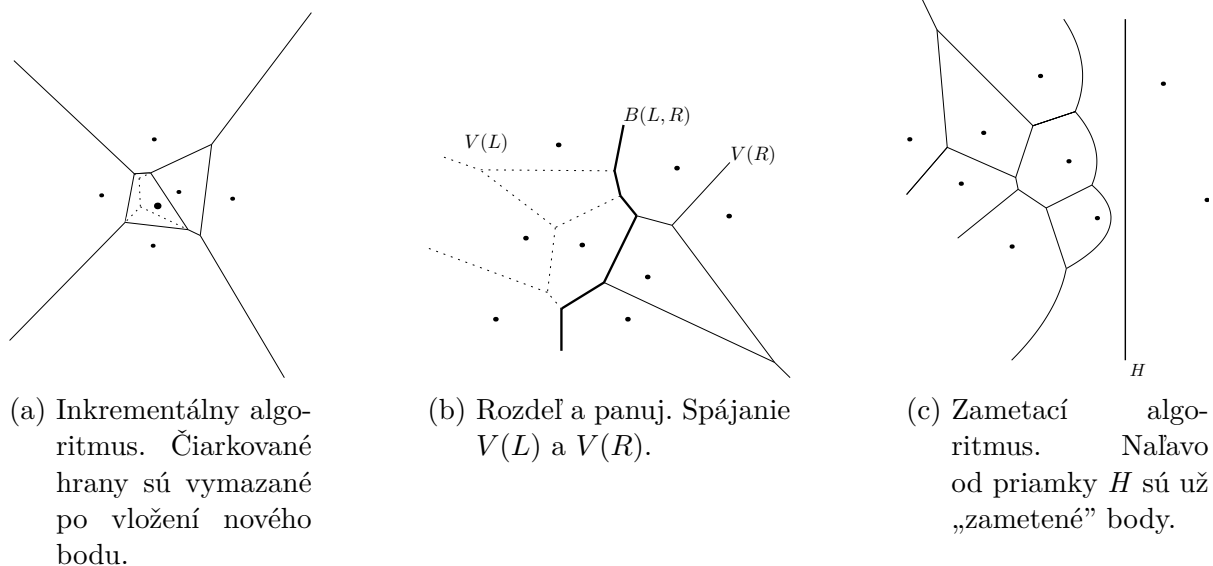
##### Rozdeľ a panuj

Algoritmus Rozdeľ a panuj rozdeľuje množinu bodov  $S$  hranou na približne rovnaké podmnožiny  $L$  a  $R$ , vid' obrázok 1.5b. Voroného diagram  $V(L)$  a  $V(R)$  je určený rekurzívne, to znamená, že rovnaký postup je aplikovaný na množiny  $L$  a  $R$ . Ak sa v množine nachádzajú už len dva alebo tri body, je možné zostrojiť diagram.

Dôležitou časťou algoritmu je nájdanie hrán a spájanie  $V(L)$  a  $V(R)$ , aby sme získali  $V(S)$ .

##### Zametací

Zametací (Fortuneho) algoritmus používa tzv. *zametaciu priamku*, ktorá sa posúva horizontálne alebo vertikálne a postupne prechádza všetkými bodmi množiny. Zametacia priamka „zametie“ body, cez ktoré prejde a rozdeľuje tak množinu na spracovanú a nespracovanú oblasť, čo možno vidieť na obrázku 1.5c. Spracovaná oblasť v ľubovoľnom čase obsahuje hotový Voroného diagram (Delaunayho trianguláciu).



Obr. 1.5: Ukážky výpočtu Voroného diagramu

## 2. Spracovanie obrazu

Táto kapitola sa venuje základným pojmom používaných pri spracovaní obrazovej informácie. Odborný text v tejto kapitole je čerpaný z [3], [5], [6] a [7].

### 2.1. Digitalizácia obrazu

*Signál* je vnímaný ako funkcia závislá na nejakej premennej s fyzikálnym významom. Takáto funkcia môže byť jednorozmerná, závislá napríklad na čase, dvojrozmerná funkcia predstavujúca napríklad obraz závislý na dvoch súradniciach v rovine, alebo viacrozmerná. Funkcia môže byť skalárna, popisujúca *monochromatický* (šedotónový) obraz, alebo vektorová, popisujúca napríklad farebný obraz.

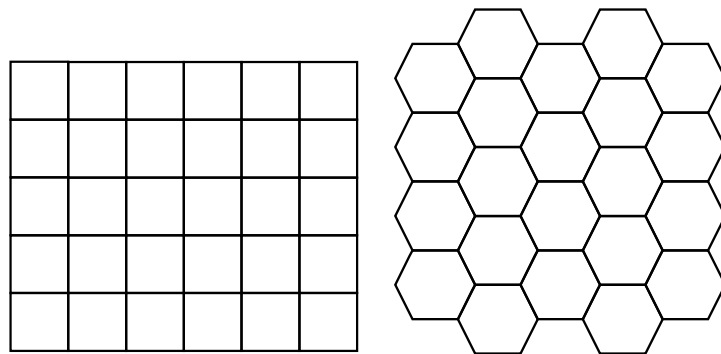
Ďalej môžeme tieto funkcie deliť na *spojité*, *diskrétné* a *digitálne*. Spojitá funkcia má spojitý definičný obor aj obor hodnôt, diskretná funkcia má diskretný definičný obor. Digitálnou funkciou nazveme funkciu s diskretným definičným oborom aj oborom hodnôt. Funkčná hodnota obrazovej funkcie značí jas, intenzitu svetla, teplotu alebo iné fyzikálne veličiny.

Aby bolo možné s obrazom v počítači ďalej pracovať, je potrebné ho *digitalizovať*. Obraz zachytený senzorom je spojitá funkcia  $f_I(x, y)$ , kde  $x, y \in \mathbb{R}^2$ . Digitalizácia takejto funkcie pozostáva z dvoch krokov, zo *vzorkovania* a *kvantizácie*.

#### Vzorkovanie

Vzorkovanie je proces prevedenia funkcie  $f_I(x, y)$  do matice s  $m$  riadkami a  $n$  stĺpcami. Platí, že čím jemnejšie je vzorkovanie, t.j. čím väčšie  $m, n$ , tým lepšiu aproximáciu spojitej obrazovej funkcie získame.

Funkciu  $f_I(x, y)$  digitalizujeme pomocou *vzorkovacej mriežky*. Najčastejšou štruktúrou vzorkovacej mriežky je štvorcová a hexagonálna, obe možno vidieť na obrázku 2.1a, resp. 2.1b. Jeden prvok obrazovej matice sa nazýva *pixel* alebo *obrazový element*.



(a) Štvorcová mriežka

(b) Hexagonálna mriežka

Obr. 2.1: Vzorkovacie mriežky

Ideálne vzorkovanie získame vynásobením spojitej obrazovej funkcie  $f_I(x, y)$  vzorkovacou funkciou

$$S(x, y) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \delta(x - j\Delta x, y - k\Delta y) \quad (2.1)$$

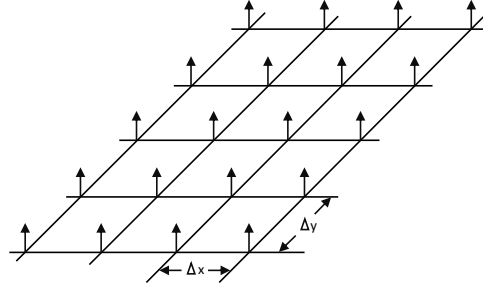
## 2.1. DIGITALIZÁCIA OBRAZU

pozostávajúcou z nekonečného poľa *Diracových delta funkcií*, znázornených na obrázku 2.2.

**Definícia 2.1.** Diracova delta funkcia je operátor (z matematického hľadiska nejde o funkciu), pre ktorý platí:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(x, y) dx dy = 1$$

a  $\delta(x, y) = 0$  pre  $(x, y) \neq 0$ .



Obr. 2.2: Pole Diracových delta funkcií [5]

Vzorkovaný obraz je v tvare

$$F_p(x, y) = F_I(x, y)S(x, y) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} F_I(j\Delta x, k\Delta y)\delta(x - j\Delta x, y - k\Delta y). \quad (2.2)$$

Pri reálnom vzorkovaní majú impulzy konečnú veľkosť a rozmery vzorkovacieho obrazu sú konečné. Pre vzorkovacie impulzy platí

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P(x, y) dx dy = 1 \quad (2.3)$$

a vzorkovacia funkcia je v tvare

$$S(x, y) = \sum_{j=-J}^J \sum_{k=-K}^K P(x - j\Delta x, y - k\Delta y). \quad (2.4)$$

Dosadením do rovnice

$$F_P(x, y) = F_I(x, y)S(x, y) \quad (2.5)$$

dostaneme

$$F_P(x, y) = \sum_{j=-J}^J \sum_{k=-K}^K F_I(j\Delta x, k\Delta y)P(x - j\Delta x, y - k\Delta y). \quad (2.6)$$

### Kvantizácia

Kvantizácia je prevedenie oboru hodnôt spojitej obrazovej funkcie do diskretných hodnôt. Počet úrovní kvantizácie je daný *bitovou hĺbkou*  $b$ . Väčšina zariadení pracujúcich s obrazom používa kvantizáciu v  $k$  rovnakých úrovniach, kde  $k = 2^b$ . Najčastejšie sa používa hodnota  $b = 8$ , čím získame 256 úrovní kvantizácie, ale časté je aj použitie 12 alebo 16-bitovej kvantizácie.

## 2.2. Fúzia obrazu

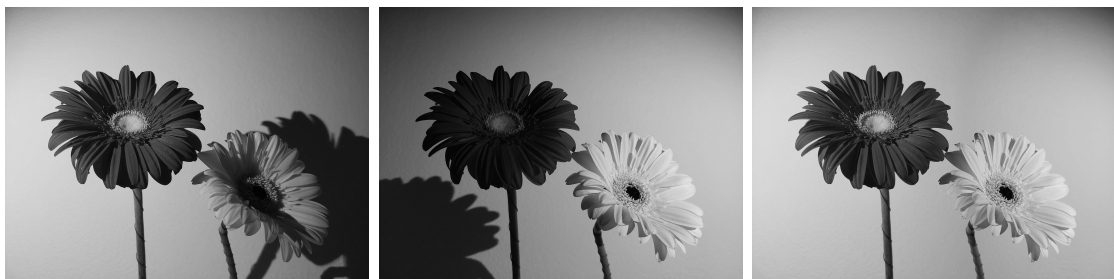
*Fúzia obrazu* je využitie informácií z viacerých snímok za účelom vytvorenia jedného obrazu. Zvyčajne zjednocujeme rôzne zaostrené snímky zaobstarané pri rovnakej scéne. Vstupné snímky a techniky fúzie volíme tak, aby bol výsledný obraz čo najkvalitnejší.

Techniky fúzie sa delia na fúziu obrazu na *pixelovej úrovni* (pixel level image fusion), na *úrovni charakteristík* (feature level) a na *úrovni rozhodovania* (decision level).

Pri pixelovej úrovni použijeme priamo informácie z pôvodných snímok. Metódy použitia pixelovej úrovne sú napríklad priemer pixelov vstupných snímok alebo voľba minimálneho, alebo maximálneho pixelu, ako ukazuje obrázok 2.3.

Fúzia obrazu na úrovni charakteristík vyberie zo snímok časti obsahujúce hrany, línie, alebo rôzne textúry.

Pri fúzii na úrovni rozhodovania sú informácie zo vstupných snímok spracované jednotlivo použitím viacerých algoritmov. Príkladom fúzie obrazu na úrovni rozhodovania je *multi-focus*, t.j. zjednotenie rôzne zaostrených snímok zaobstaraných pri rovnakej scéne za účelom jedného, čo najlepšie zaostreného obrazu.



(a) Prvá vstupná snímka

(b) Druhá vstupná snímka

(c) Fúzia snímok

Obr. 2.3: Fúzia snímok rovnakej scény za rôznych svetelných podmienok s použitím voľby maximálnej hodnoty pixelu



# 3. Implementácia

Cieľom implementácie bolo navrhnuť metódu na nájdenie optimálne zaostrenej snímky zo série snímok. Na spracovanie dát bol zvolený programovací jazyk *C++* a pre načítanie snímok a základné operácie s nimi bola použitá knižnica *OpenCV*.

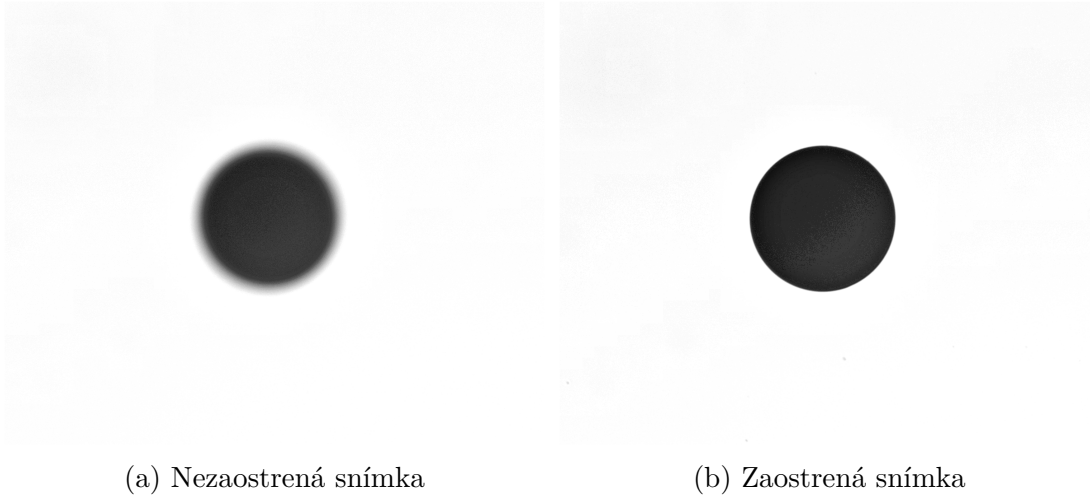
Táto kapitola popisuje algoritmy pre nájdenie optimálne zaostrenej snímky. Boli použité dve série testovacích snímok  $\mathcal{S}_i^N$ , kde  $N$  je označenie série,  $i = 1, \dots, l$  je poradie snímky v sérii a  $l$  je počet snímok v danej sérii. Prvý algoritmus je aplikovateľný na snímky objektu, ktorý možno považovať za dvojrozmerný. Použité boli snímky dvojrozmerného objektu pozorovaného svetelným mikroskopom, pričom pozorovaný objekt leží v jednej rovine ostrosti.

Druhý algoritmus je modifikáciou prvého a je vhodný pre sériu snímok trojrozmerných objektov ležiacich v rôznych rovinách ostrosti.

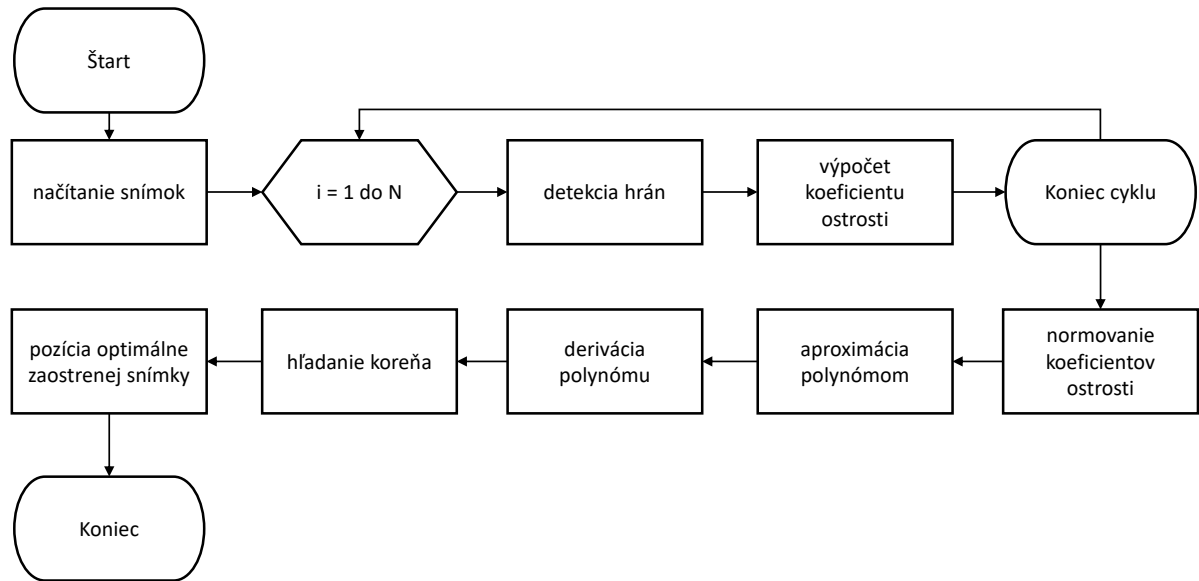
Použité snímky sú 16-bitové vo formáte *tif* a jedná sa o snímky zo svetelného mikroskopu.

## 3.1. Metóda určenia roviny ostrosti

Na nájdenie optimálne zaostrenej snímky dvojrozmerného objektu bolo použité porovnanie snímok na základe koeficientu ostrosti  $\kappa$ . Koeficient ostrosti  $\kappa$  bol počítaný pre všetky snímky v sérii  $\mathcal{S}^1$ , dáta aproximované polynómom  $q(t)$  a následne bol nájdený stacionárny bod polynómu  $q(t)$ . Stacionárny bod predstavuje pozíciu optimálne zaostrenej snímky v sérii. Vývojový diagram metódy použitej pre testovacie snímky série  $\mathcal{S}^1$  možno vidieť na obrázku 3.2. Na obrázku 3.1 je ukážka testovacích snímok série  $\mathcal{S}^1$ .



Obr. 3.1: Ukážka testovacích snímok série  $\mathcal{S}^1$



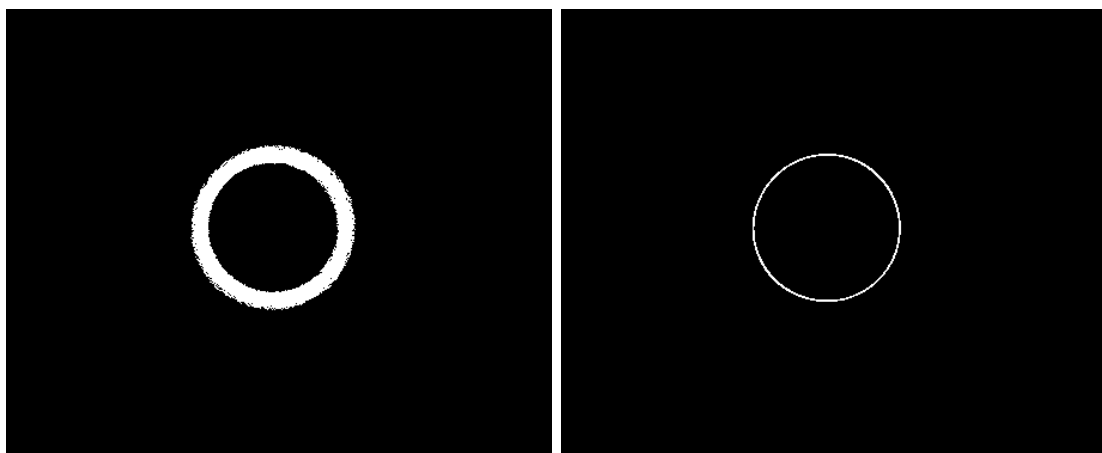
Obr. 3.2: Vývojový diagram metódy

### 3.1.1. Detekcia hrán

Prvým krokom algoritmu po načítaní snímky je detekcia hrán. Doprednou diferenciou boli vypočítané hodnoty gradientov pixelov snímok  $\mathcal{S}_i^1$ . Normy týchto gradientov boli zapísané do matice  $\mathcal{G}_i$ . Dopredná diferencia bola zvolená z dôvodu, že narozdiel od centrálnej diferencie nedochádza k vyhladeniu hodnôt pixelov. Pre každú snímku bola zistená maximálna hodnota normy gradientu  $\mathcal{G}_i(j, k)$  a z nej bola stanovená prahová hodnota, ktorú budeme značiť  $\mathcal{P}_i$ :

$$\mathcal{P}_i = \max\{\mathcal{G}_i(j, k)\} \cdot c \quad j = 0, \dots, m-1; \quad k = 0, \dots, n-1,$$

kde  $m, n$  sú rozmery obrazovej matice a  $c$  je vhodne zvolená konštanta, konkrétne  $c = 0,4$ . Hrany snímky teda tvoria pixely, ktorých hodnota je väčšia ako 40% maximálnej hodnoty normy gradientu danej snímky.



(a) Hrana nezaostrenej snímky

(b) Hrana zaostrenej snímky

Obr. 3.3: Hrany snímok z 3.1

### 3.1. METÓDA URČENIA ROVINY OSTROSTI

Na obrázku 3.3 sú bielou farbou zvýraznené hrany snímok 3.1. Zaostrená snímka nadobúda vyššiu maximálnu hodnotu normy gradientu než nezaostrená snímka, zvyšuje sa prahová hodnota  $\mathcal{P}_i$  a menej pixelov tvorí hranu. Preto je viditeľné, že nezaostrená snímka má hrany širšie a zaostrená snímka užšie.

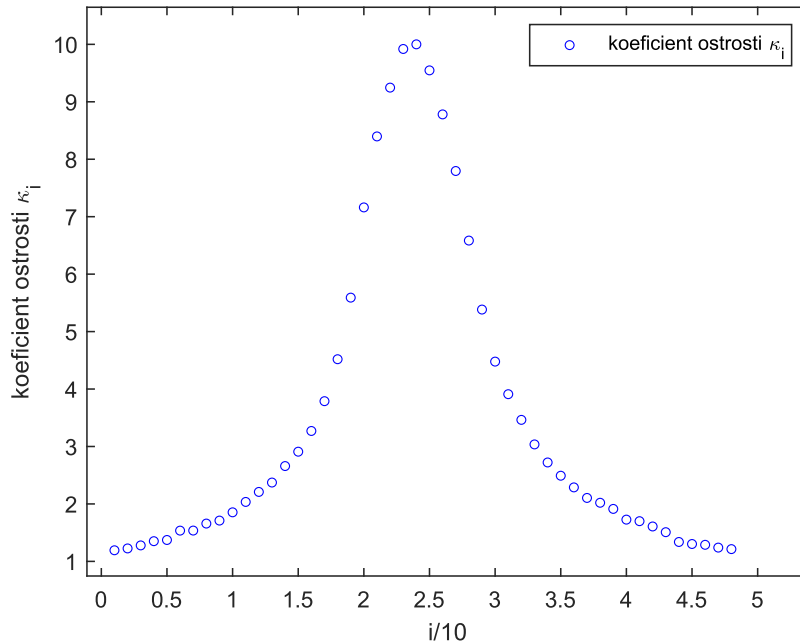
#### Výpočet koeficientu ostrosti

Po detekcii hrán je možné vypočítať koeficient ostrosti  $\kappa_i$ . Koeficient ostrosti  $\kappa_i$  je počítaný ako stredná hodnota odhadnutá aritmetickým priemerom veľkostí gradientov, pre ktoré platí  $\mathcal{G}_i(j, k) > \mathcal{P}_i$ :

$$\kappa_i = \frac{\sum_{j=0}^{m-1} \sum_{k=0}^{n-1} \mathcal{G}_i(j, k)}{p} \quad \forall \mathcal{G}_i(j, k) > \mathcal{P}_i,$$

kde  $p$  je počet pixelov spĺňajúcich podmienku  $\mathcal{G}_i(j, k) > \mathcal{P}_i$ .

Po vypočítaní koeficientov ostrosti  $\kappa_i$  pre všetky snímky v sérii  $\mathcal{S}^1$  boli hodnoty vykreslené do grafu na obrázku 3.4. Platí, že čím je väčšia hodnota koeficientu ostrosti  $\kappa_i$ , tým je daná snímka lepšie zaostrená.



Obr. 3.4: Koeficienty ostrosti  $\kappa_i$  snímok série  $\mathcal{S}^1$

Vypočítané koeficienty ostrosti  $\kappa_i$  boli kvôli lepšej aproximácii polynómom normované na hodnoty 1 až 10.

Po vykreslení koeficientov ostrosti  $\kappa_i$  do grafu získame prehľad o tom, na ktorej pozícii sa nachádza optimálne zaostrená snímka.

Nájdenie konkrétnej pozície optimálne zaostrenej snímky je však o niečo zložitejšie, pretože dve snímky môžu dosahovať rovnaké hodnoty koeficientov ostrosti  $\kappa_i$ , alebo môže nastať situácia, kedy sa optimálne zaostrená snímka nenachádza v sérii  $\mathcal{S}^1$  vstupných snímok. Maximum koeficientov ostrosti  $\kappa_i$  teda z dôvodu šumu v obraze nemusí byť jednoznačné. V týchto prípadoch je ťažké určiť pozíciu optimálne zaostrenej snímky odčítaním

z grafu. Riešením tohto problému je aproximácia dát polynómom.

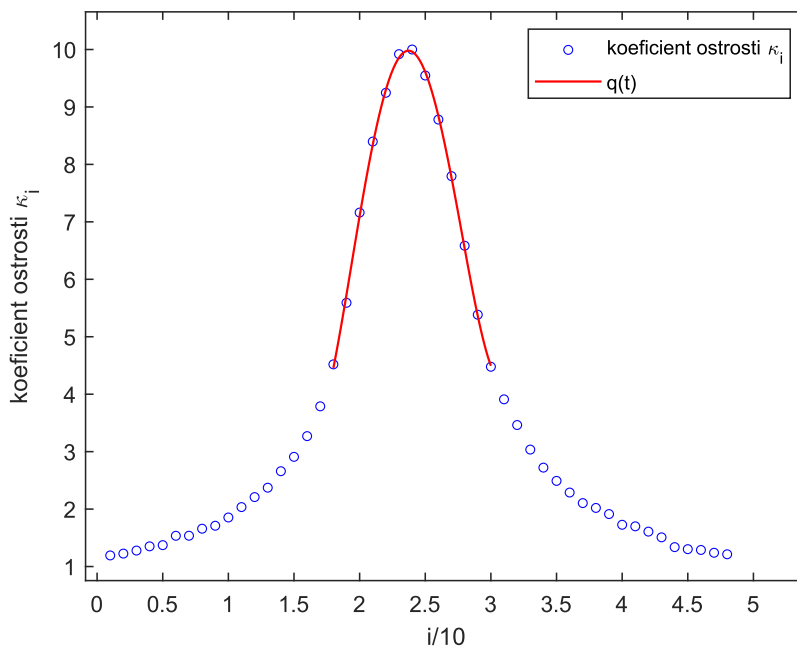
V bakalárskej práci bola aproximácia určená metódou najmenších štvorcov popísanej v 1.2.3. Dáta boli aproximované polynómom  $q(t)$  štvrtého stupňa:

$$q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4.$$

Keďže pozícia optimálne zaostrenej snímky je očakávaná medzi snímkami s vyššími hodnotami koeficientu ostrosti  $\kappa_i$ , je možné snímky s nižším koeficientom ostrosti z aproximácie vylúčiť a dosiahnuť tak lepšiu aproximáciu. Bola stanovená hranica  $\tau$ :

$$\tau = d \cdot (\max\{\kappa_i\} - \min\{\kappa_i\}) + \min\{\kappa_i\},$$

kde  $d$  je vhodne zvolená konštanta, ktorej hodnota bola experimentálne určená ako  $d = \frac{1}{3}$ . Do aproximácie vstupujú snímky, pre ktoré platí  $\kappa_i > \tau$ . Na obrázku 3.5 je vidieť, že do aproximácie vstupovalo 13 snímok z celkového počtu 48 snímok série  $\mathcal{S}^1$ .



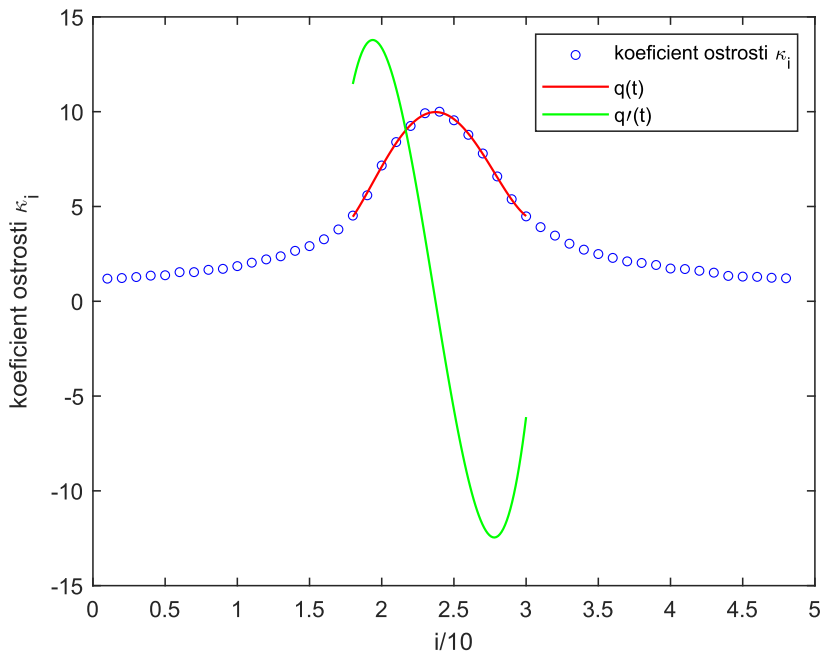
Obr. 3.5: Aproximácia dát polynómom  $q(t)$

### 3.1.2. Pozícia optimálne zaostrenej snímky

Po zistení predpisu aproximačného polynómu  $q(t)$  je možné nájsť pozíciu optimálne zaostrenej snímky.

Pozícia optimálne zaostrenej snímky je zhodná s pozíciou extrému polynómu  $q(t)$ . Úlohu hľadania extrému polynómu je možné premeniť na úlohu hľadania stacionárneho bodu polynómu tak, že aproximačný polynóm zderivujeme a nájdeme koreň derivácie  $q'(t)$ , znázornenej na obrázku 3.6. Nájdením koreňa polynómu  $q'(t) = 0$  vyšetříme extrém funkcie a zároveň zistíme pozíciu optimálne zaostrenej snímky.

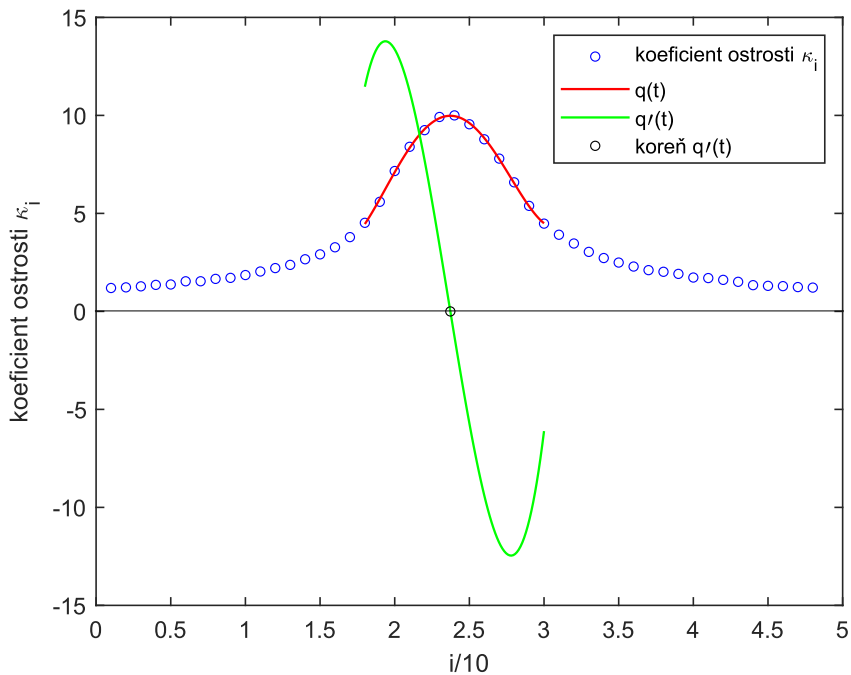
### 3.1. METÓDA URČENIA ROVINY OSTROSTI



Obr. 3.6: Derivácia aproximačného polynómu  $q(t)$

#### Hľadanie extrému

Na nájdenie koreňa polynómu  $q'(t)$  bola použitá metóda bisekcie popísaná v 1.2.2. Interval pre hľadanie koreňa je stanovený snímkami, ktoré vstupovali do aproximácie a závisí na konštante  $d$ . Nájdený koreň polynómu  $q'(t)$  predstavuje pozíciu optimálne zaostrenej snímky, viď obrázok 3.7.



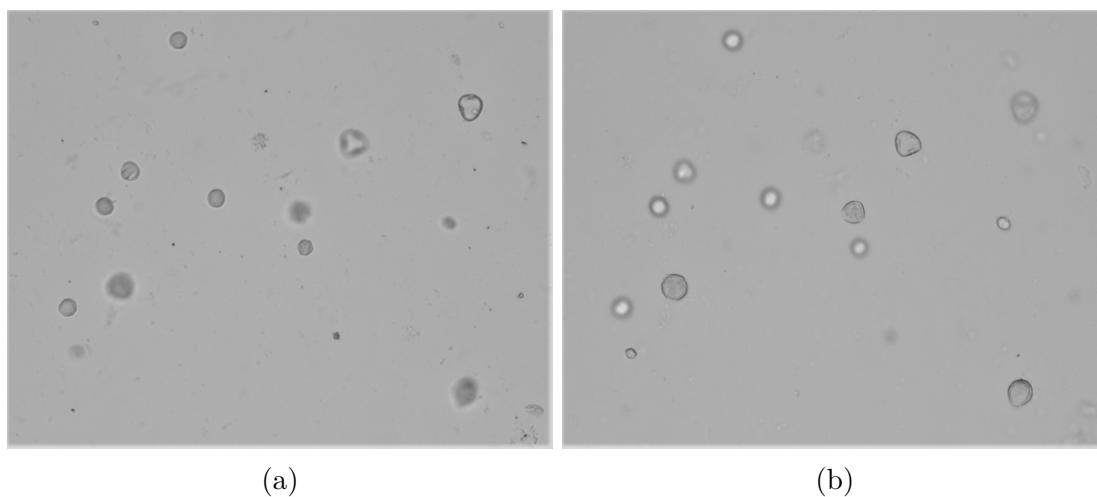
Obr. 3.7: Nájdená pozícia optimálne zaostrenej snímky série  $\mathcal{S}^1$

### 3.2. Metóda určenia roviny ostrosti pre viac objektov

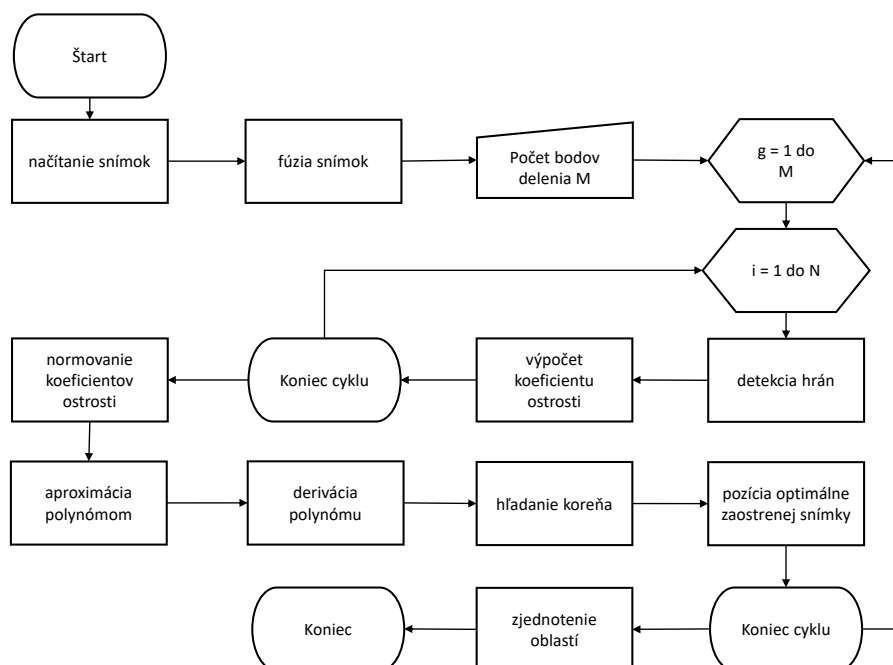
Druhá séria testovacích snímok  $\mathcal{S}^2$  obsahuje snímky pelových zrn pozorovaných svetelným mikroskopom.

Pelové zrná sú trojrozmerné objekty a každé leží v inej výške, teda v inej rovine ostrosti. To znamená, že rôzne pelové zrná sú optimálne zaostrené na rôznych snímkach, čo možno vidieť na obrázkoch 3.8a a 3.8b.

Koeficient ostrosti  $\kappa$  v tomto prípade nie je možné počítat z jednej snímky ako z celku, ale je potrebné snímku rozdeliť na jednotlivé zrná (oblasti) a následne počítat koeficient ostrosti pre každú oblasť zvlášť. Vývojový diagram metódy pre pelové zrná je znázornený na obrázku 3.9.



Obr. 3.8: Ukážka testovacích snímok  $\mathcal{S}^2$



Obr. 3.9: Vývojový diagram metódy

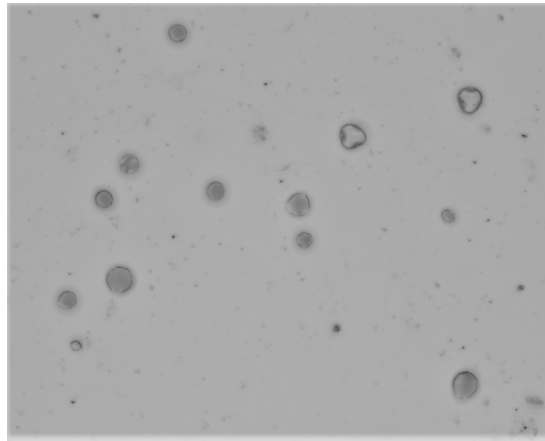
## 3.2. METÓDA URČENIA ROVINY OSTROSTI PRE VIAC OBJEKTOV

### 3.2.1. Fúzia snímok

Aby bolo možné snímky zo série rozdeliť na zvolené oblasti, je nutné zvoliť snímku, na ktorej bude rozklad vykonaný, uložený a aplikovaný pre ostatné snímky  $\mathcal{S}_i^2$ . Pre takýto rozklad je vhodná snímka, na ktorej sú čo najlepšie viditeľné objekty zo všetkých snímok zo série. Snímku, na ktorej sú zvýraznené všetky objekty vyskytujúce sa v sérii, získame fúziou  $\mathcal{F}$  snímok  $\mathcal{S}_i^2$  na pixelovej úrovni výberom minimálnej hodnoty pixelu.

$$\mathcal{F}(j, k) = \min\{\mathcal{S}_i^2(j, k)\}, \quad i = 1, 2, \dots, l; \quad j = 0, 1, \dots, m - 1; \quad k = 0, 1, \dots, n - 1.$$

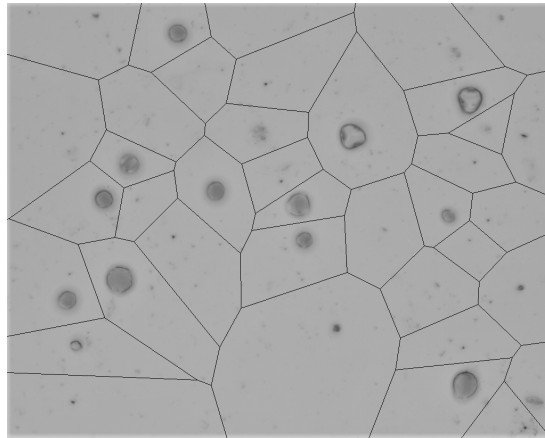
Ako ukazuje obrázok 3.10, výsledkom fúzie  $\mathcal{F}$  je tmavá snímka, na ktorej sú zvýraznené nielen všetky objekty, ktoré chceme pozorovať, ale aj zdanlivé nečistoty vyskytujúce sa na snímkach v sérii  $\mathcal{S}^2$ .



Obr. 3.10: Fúzia  $\mathcal{F}$  snímok zo série  $\mathcal{S}^2$

### 3.2.2. Voroného rozklad fúzie snímok

Fúziou  $\mathcal{F}$  bola získaná snímka, ktorá bola použitá ako vstupná snímka pre zvolenie rozkladu. Rozklad vstupnej snímky vznikol manuálnym volením množiny bodov (jadier)  $\mathcal{Z} := \{z_1, \dots, z_M \in \mathcal{F}\}$ . Pre množinu bodov  $\mathcal{Z}$  bola pomocou knižnice *OpenCV* vytvorená Delaunayho triangulácia a z nej bol zostrojený Voroného diagram  $V(\mathcal{Z})$ , viď obrázok 3.11. Pre vhodne zvolenú množinu bodov  $\mathcal{Z}$  je možné fúziu  $\mathcal{F}$  rozdeliť tak, aby sa každé peľové zrno, ktoré chceme pozorovať, nachádzalo v samostatnej Voroného bunke  $\nu(z, \mathcal{Z})$ .

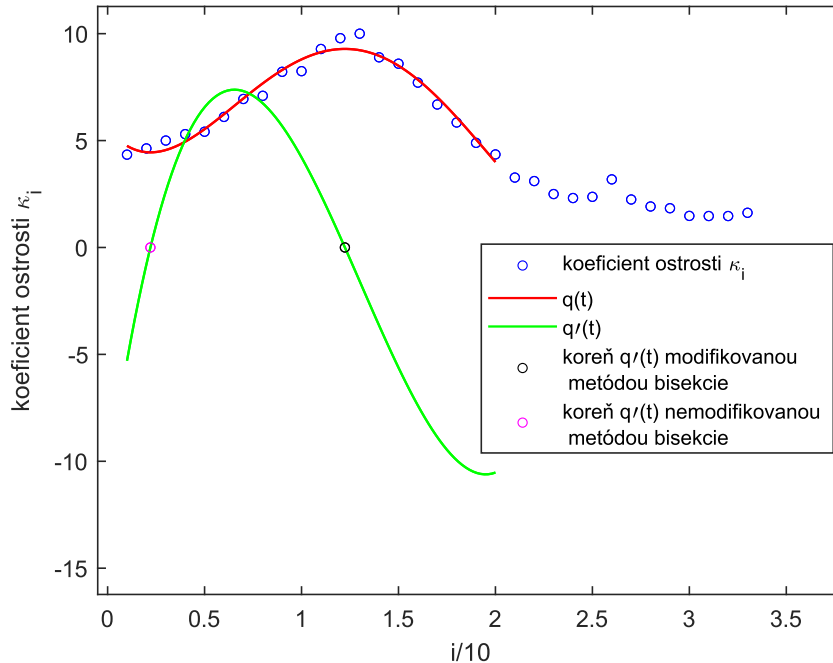


Obr. 3.11: Voroného diagram  $V(\mathcal{Z})$

### Výpočet pre Voroného bunku

Metódu bisekcie bolo potrebné pre sériu  $\mathcal{S}^2$  modifikovať. Keďže aproximačný polynóm  $q(t)$  je stupňa 4, môže nadobúdať najviac tri extrémny. Ako vidieť na obrázku 3.12, v prípade niektorých Voroného buniek obsahujúcich nečistoty teda nastávala situácia, kedy interval snímok vstupujúcich do aproximácie obsahoval dva alebo tri korene, pričom len jeden z nich predstavoval pozíciu optimálne zaostrenej snímky.

Úprava metódy bisekcie spočíva v overení všetkých podintervalov obsahujúcich koreň a zameraní sa na ten podinterval, ktorého koreň nadobúda najväčších hodnôt koeficientu ostrosti  $\kappa_i$ .



Obr. 3.12: Výpočet pozície optimálne zaostrenej snímky pre interval s dvomi koreňmi

Algoritmus výpočtu koeficientu ostrosti  $\kappa_i$  je rovnaký ako v prípade dvojrozmerného objektu, ale koeficienty ostrosti  $\kappa_i$  nepočítame zo snímky ako celku, ale z jednotlivých buniek Voroného diagramu  $V(\mathcal{Z})$ . Na obrázku 3.13b je znázornený výpočet pre Voroného bunku na obrázku 3.13a.

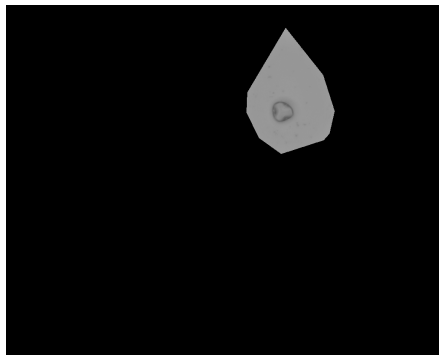
Pre zvolenú Voroného bunku  $\nu(z, \mathcal{Z})$  sú detekované hrany, vypočítaný koeficient ostrosti  $\kappa_i$ , vykreslené a aproximované dáta, následne je aproximačný polynóm  $q(t)$  derivovaný a je nájdený jeho koreň, a tým pozícia snímky  $\mathcal{S}_i^2$ , na ktorej je daná Voroného bunka  $\nu(z, \mathcal{Z})$  optimálne zaostrená.

### 3.2.3. Výsledok algoritmu

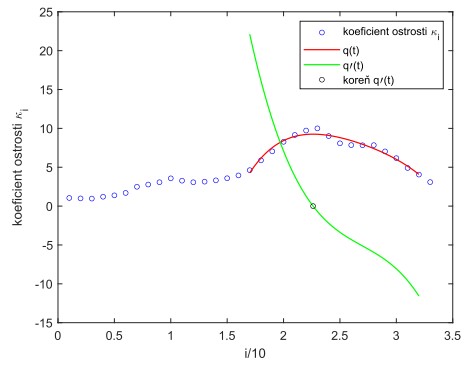
Výsledkom algoritmu pre sériu testovacích snímok  $\mathcal{S}^2$  nie je jedna pozícia optimálne zaostrenej snímky, ale toľko pozícií, koľko buniek obsahuje Voroného diagram  $V(\mathcal{Z})$ . Každý Voroného oblasti  $\nu(z, \mathcal{Z})$  je priradená pozícia snímky  $\mathcal{S}_i^2$ , na ktorej je daná oblasť optimálne zaostrená. Zjednotením Voroného oblastí pre správne roviny ostrosti bola vytvorená jedna výsledná snímka  $\mathcal{S}$ , ktorú možno vidieť na obrázku 3.14.



### 3.2. METÓDA URČENIA ROVINY OSTROSTI PRE VIAC OBJEKTOV

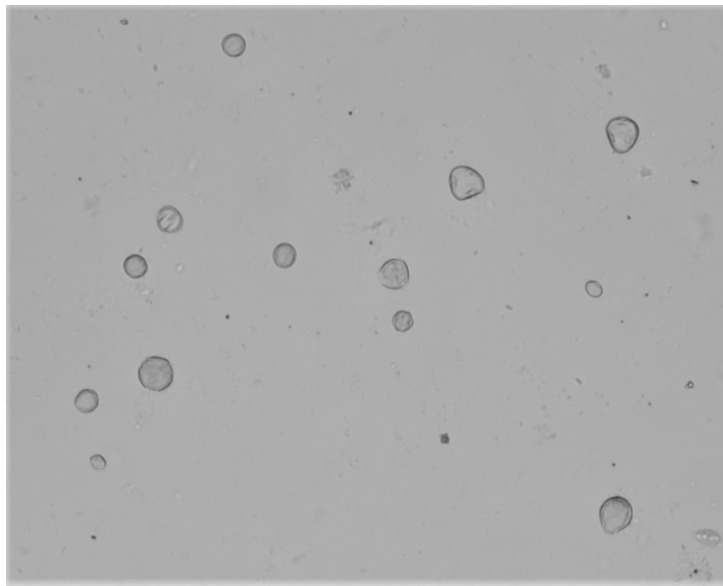


(a) Voroného bunka  $\nu(z, \mathcal{Z})$



(b) Výpočet pre Voroného bunku  $\nu(z, \mathcal{Z})$

Obr. 3.13: Výpočet pre jednu Voroného bunku  $\nu(z, \mathcal{Z})$



Obr. 3.14: Výsledná snímka  $\mathcal{S}$

# Záver

Táto bakalárska práca sa zaoberala hľadaním optimálne zaostrenej snímky. Téma práce vychádza z problematiky pozorovania snímok voľným okom, kedy je hľadanie roviny ostrosti subjektívne. Cieľom práce bolo nájsť objektívnu metódu pre určenie miery ostrosti obrazu.

V prvej kapitole bol zavedený matematický aparát potrebný pre spracovanie obrazu a numerické metódy použité pre nájdenie roviny ostrosti objektov na testovacích snímkach. Druhá kapitola bola venovaná digitalizácii obrazu. Implementačná časť popisuje metódy použité pre hľadanie roviny ostrosti rovinného a trojrozmerného objektu. Metódy boli otestované programom vytvoreným v programovacom jazyku C++ za použitia knižnice OpenCV, ktorá umožňuje prácu s obrazom.

Metódy sú založené na detekcii hrán objektov, výpočte koeficientu ostrosti a nájdení pozície optimálne zaostrenej snímky. Boli použité dve série testovacích snímok objektov pozorovaných svetelným mikroskopom. Prvá séria obsahovala snímky rovinného objektu, druhú sériu tvorili snímky peľových zrn. Peľové zrná na snímkach neležia v jednej rovine ostrosti, preto bolo potrebné snímky rozdeliť na oblasti. Následne bola pre každú oblasť nájdená rovina ostrosti a zjednotením oblastí vznikla výsledná snímka. Pri delení snímok peľových zrn na oblasti boli zvolené jadrá, na základe ktorých bol zostrojený Voroného diagram. Jadrá Voroného diagramu boli volené manuálne, pretože automatické hľadanie reprezentačného bodu jadra sa ukázalo ako zložitá úloha, pri ktorej dochádzalo k rozpadaniu peľových zrn na viacero častí. Automatické nájdenie týchto deliacich bodov (jadier) Voroného diagramu môže byť predmetom nadväzujúcej diplomovej práce.

# Literatúra

- [1] AURENHAMMER, Franz, Rolf KLEIN a Der-Tsai LEE. *Voronoi diagrams and Delaunay triangulations*. 1. Syngapore: World Scientific Publishing, c2013. ISBN 978-9814447638.
- [2] ČERMÁK, Libor a Rudolf HLAVIČKA. *Numerické metody*. Brno: Akademické nakladatelství CERM, 2005. ISBN 80-214-3071-0.
- [3] KAUR, Harpreet, Deepika KOUNDAL a Virender KADYAN. *Image Fusion Techniques: A Survey* [online]. In: . 24 January 2021 [cit. 2023-05-14]. Dostupné z: <https://doi.org/10.1007/s11831-021-09540-7>
- [4] NECHVÁTAL, Luděk. Ústav matematiky. *MATEMATIKA online: Matematické inženýrství: Matematická analýza II* [online]. Brno: Ústav matematiky FSI VUT, 2005 [cit. 2023-05-14]. Dostupné z: <https://math.fme.vutbr.cz>
- [5] PRATT, William K. *Digital image processing: PIKS Scientific inside*. 4th ed. Hoboken.: Wiley-Interscience, c2007. ISBN 978-0-471-76777-0.
- [6] Study of Image Fusion- Techniques, Method and Applications. *International Journal of Computer Science and Mobile Computing* [online]. November- 2014, 469-476 [cit. 2023-05-14]. ISSN 2320—088X. Dostupné z: <https://www.ijcsmc.com/docs/papers/November2014/V3I11201493.pdf>
- [7] ŠONKA, Milan, Václav HLAVÁČ a Roger BOYLE. *Image Processing, analysis, and machine vision*. 4th ed. Stamford: Cengage Learning, 2015. ISBN 978-1-133-59360-7.

# Zoznam použitých skratiek a symbolov

$\mathbb{R}$	množina reálnych čísel
$[a_1, \dots, a_n]$	súradnice bodu $A$
$\mathbf{u}, \mathbf{v}$	vektory
$f'$	derivácia funkcie
$f_x$	parciálna derivácia funkcie podľa premennej $x$
$f_y$	parciálna derivácia funkcie podľa premennej $y$
$\mathcal{G}_i(j, k)$	veľkosť gradientu pixelu
$m, n$	rozmery obrazovej matice
$\mathcal{S}_i^N$	$N$ -tá séria s $l$ snímkami
$\kappa_i$	koeficient ostrosti $i$ -tej snímky
$\mathcal{P}$	prahová hodnota pre detekciu hrán
$\tau$	hranica pre aproximáciu koeficientov ostrosti
$\mathcal{F}$	fúzia snímok
$\mathcal{Z}$	množina bodov Voroného diagramu
$\nu(z, \mathcal{Z})$	Voroného bunka bodu $z \in \mathcal{Z}$
$V(\mathcal{Z})$	Voroného diagram množiny $\mathcal{Z}$

# Prílohy

```
/prilohy
├── serial1 ..... testovacie snímky rovinného objektu
│   ├── s1.tif
│   ├── s2.tif
│   ├── :
│   └── s48.tif
├── serial2 ..... testovacie snímky peľových zrn
│   ├── s1.tif
│   ├── s2.tif
│   ├── :
│   └── s33.tif
├── Bakalarska_praca ..... adresár s kódom
│   ├── cvplot1.2.2 ..... knižnica na vykreslenie grafov
│   ├── Bakalarska_praca.sln ..... aplikácia
│   ├── Bakalarska_praca.cpp
│   ├── Bakalarska_praca.vcxproj
│   └── jadra.txt ..... textový súbor obsahujúci jadrá Voroného diagramu
├── readme.txt ..... návod na spustenie programu
├── voroneho_diagram.jpg ..... Voroného diagram snímky peľových zrn
└── zjednotenie.tif ..... výsledná snímka S
```