

UNIVERZITA PALACKÉHO V OLOMOUCI  
PŘÍRODOVĚDECKÁ FAKULTA

**DIPLOMOVÁ PRÁCE**

Metody zpracování obrazu



**Katedra matematické analýzy a aplikací matematiky**

Vedoucí bakalářské práce: **RNDr. Tomáš Fürst, Ph.D.**

Vypracoval: **Jan Walach**

Studijní program: B1103 Aplikovaná matematika

Studijní obor Aplikace matematiky v ekonomii

Forma studia: prezenční

Rok odevzdání: 2015

## BIBLIOGRAFICKÁ IDENTIFIKACE

**Autor:** Jan Walach

**Název práce:** Metody zpracování obrazu

**Typ práce:** Diplomová práce

**Pracoviště:** Katedra matematické analýzy a aplikací matematiky

**Vedoucí práce:** RNDr. Tomáš Fürst, Ph.D.

**Rok obhajoby práce:** 2015

**Abstrakt:** Cílem diplomové práce bude prozkoumat několik metod zpracování obrazu, zejména detekce hran a prahování, a tyto metody využít na návrh a implementaci programového kódu pro automatické vyhodnocení dat z fenotypizační linky. Téma je součástí rozsáhlejšího výzkumu, který probíhá ve spolupráci s Centrem regionu Haná UP pro biotechnologický a zemědělský výzkum.

**Klíčová slova:** RGB, HSV, Fourierova transformace, Cannyho detekce obrazu, Bodová fuzzy metoda, Gradientní fuzzy metoda

**Počet stran:** 65

**Počet příloh:** 0

**Jazyk:** český

## BIBLIOGRAPHICAL IDENTIFICATION

**Author:** Jan Walach

**Title:** Image processing methods

**Type of thesis:** Masters's

**Department:** Department of Mathematical Analysis and Application of Mathematics

**Supervisor:** RNDr. Tomáš Fürst, Ph.D.

**The year of presentation:** 2015

**Abstract:** The main goal of the Thesis is to explore several image processing methods focused on edge detection and thresholding, then use these methods to implement a program code for automatic evaluation of data sets from a phenotyping line. The Thesis topic is part of a larger research conducted in cooperation with the Centre of the Region Hana for Biotechnological and Agricultural Research.

**Key words:** Image processing, RGB, HSV, Fourier transform, Canny edge detection, Gradient fuzzy method, Fuzzy logic method

**Number of pages:** 65

**Number of appendices:** 0

**Language:** Czech

### **Prohlášení**

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně pod vedením pana Tomáše Fürsta a všechny použité zdroje jsem uvedl v seznamu literatury.

V Olomouci dne .....

.....

podpis

# Obsah

Úvod	7
<b>1 Teoretická část</b>	<b>8</b>
1.1 Obraz	8
1.1.1 Digitální obraz	8
1.2 Barva	9
1.2.1 Světelné spektrum	10
1.2.2 Lidské oko	11
1.2.3 Barevné vidění	12
1.2.4 Tvorba barev	13
1.2.5 Barevný model	13
1.3 Filtrování	18
1.3.1 Základy filtrování v prostorové doméně	18
1.3.2 Prostorové filtrování	21
1.3.3 Práce s histogramem	23
1.3.4 Filtrování v frekvenční doméně	27
1.3.5 Fourierova transformace	27
1.3.6 Základy filtrování ve frekvenční doméně	29
1.4 Cannyho detekce hran	33
1.5 Detekce hran pomocí fuzzy logiky	39
1.5.1 Bodová fuzzy metoda	39
1.5.2 Gradientní fuzzy metoda	43
<b>2 Praktická část</b>	<b>46</b>
2.1 Arabidopsis thaliana	47
2.2 Hrášek	50
2.2.1 Problémy	52
2.2.2 Hodnocení použitých metod	52
Závěr	58
Literatura	64

## **Poděkování**

Děkuji vedoucímu své diplomové práce RNDr. Tomáši Fürstovi, Ph.D. za čas, který mi věnoval, za jeho cenné připomínky, pomoc a přátelskou podporu. Rád bych také poděkoval své rodině a přátelům za neocenitelnou motivaci, kterou mi po celou dobu studia projevovali.

# Úvod

V roce 2013 bylo v Olomouci otevřeno Centrum regionu Haná pro biotechnologický a zemědělský výzkum. Na první pohled se může zdát, že možnosti využití matematiky v takovémto centru jsou velmi omezené. Když se však zamyslíme hlouběji, zjistíme, že matematiku lze aplikovat v nejrůznějších oborech.

Jedním z aktuálních výzkumných projektů Centra regionu Haná je analýza vlivu složení hnojiv na růst konkrétních typů rostlin. Celý proces výzkumu probíhá tak, že jsou zasazeny tisíce vzorků rostlin do půd obohacených různými typy hnojiv, a následně jsou v daných intervalech fotografovány. Protože žijeme v technologicky vyspělé době, proč nevyužít k měření velikosti rostlin jednoho z nejlepších vynálezů lidstva – počítače. A právě v tomto bodě přichází na řadu aplikovaná matematika.

Stěžejním cílem diplomové práce je vytvoření matematického programu schopného určení velikosti rostlin na konkrétních sériích snímků poskytnutých Centrem regionu Haná. Aby byl stěžejní cíl naplněn, je potřeba vyhledat a porozumět takovým metodám zpracování digitálního obrazu, které jsou teoreticky schopné stěžejního cíle dosáhnout. Během procesu psaní diplomové práce došlo k užšímu výběru těchto metod, a proto je dalším dílčím cílem práce tyto metody zhodnotit, popřípadě vybrat tu nejlepší.

Tato práce je rozčleněna na dvě hlavní části. V teoretické části se autor diplomové práce zabývá teorií digitálního obrazu, teorií barev, obrazového filtrování a detekcí hran. V praktické části je popsán vytvořený program a jsou prezentovány výsledky, které byly pomocí tohoto programu dosaženy.

# Kapitola 1

## Teoretická část

### 1.1. Obraz

První kapitola této práce se zabývá problematikou digitálního obrazu. Konkrétně se jedná o diskreditaci nutnou k zobrazení obrazu a také je řešena obrazová barevnost. Čerpáno bylo zejména z [1]

#### 1.1.1. Digitální obraz

Digitální reprezentace obrazu spočívá v převedení spojitého obrazu do diskrétního a to hned ve dvou ohledech.

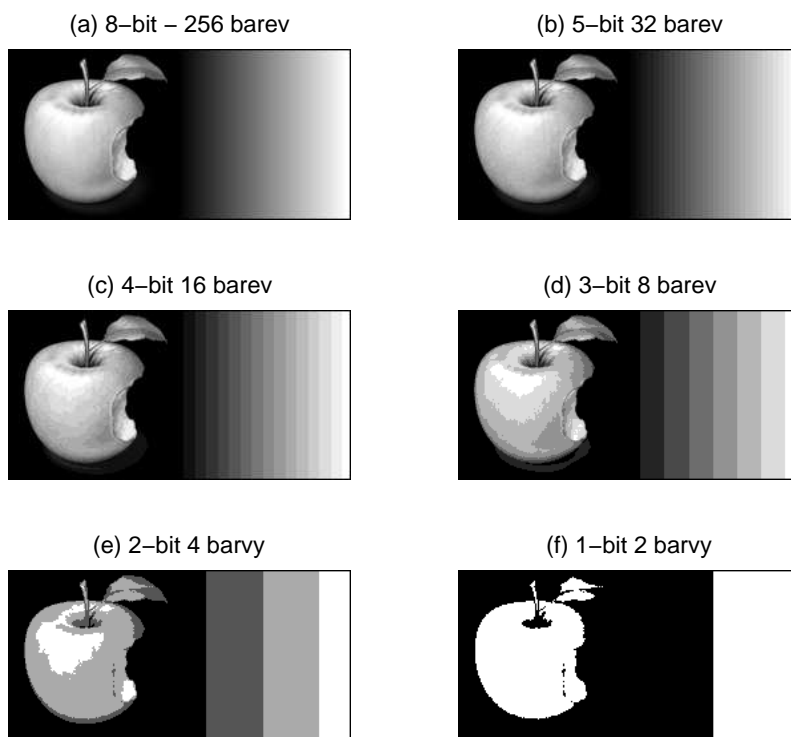
Za prvé, se jedná to diskreditaci v x-ové a y-ové ose. Spojitý obraz reprezentovaný spojitou funkcí  $f(s, t)$ , kde  $s, t$  jsou proměnné, rozdělíme do dvou-rozměrného pole  $f(x, y)$  o  $M$  řádcích a  $N$  sloupcích, kde  $x = 0, 1, 2, \dots, M - 1, y = 0, 1, 2, \dots, N - 1$ .

Potom funkci  $f(x, y)$  můžeme chápat jako matici:

$$f(x, y) = \begin{pmatrix} f(0, 0) & f(0, 1) & f(0, 2) & \cdots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & f(1, 2) & \cdots & f(1, N - 1) \\ \vdots & \vdots & \ddots & \vdots & \\ f(M, 0) & f(M, 1) & f(M, 2) & \cdots & f(M - 1, N - 1) \end{pmatrix} \quad (1.1)$$

Každá tato hodnota v matici tvoří tzv. *pixel*. Pixel je tedy vzorek z původního obrazu. Důležitou roli hraje počet takových vzorků pořizovaných z původního ob-





Obr. 1: Různá barevná hloubka [Autor práce]

razu. Poté se hovoří o tzv. rozlišení obrazu. Čím více pixelů digitální obraz obsahuje, tím je podobnější původnímu obrazu a také tím více působí celistvě.

Druhá diskreditace spočívá v tom, že jsme schopni zobrazit pouze konečné množství barev. Obvykle jsou obrazy ukládány v 8-bitovém formátu, což znamená, že existuje  $2^8 = 256$  odstínů jedné barvy. Obvyklé digitální obrazy fungují na třísloužkovém principu (viz kapitola 1.2.5). Což dohromady vytváří  $256^3 = 16777216$  možných barev.

Obr. 1 znázorňuje, jak vypadá černobílý obrázek při různých barevných hloubkách.

## 1.2. Barva

Barva je dalším velmi významným aspektem při digitální reprezentaci obrazu. Na barvu se můžeme dívat ze dvou hledisek.

Za prvé se na barvu můžeme dívat z hlediska světla. Pod různým světlem se barva objektu může zdát různá a navíc platí, že bez světla barva vlastně neexistuje. Druhým pohledem na věc je to, že barvu vnímáme subjektivně. Díky tomu, že je informace o barvě zpracována v mozku, ji každý vnímá trochu jinak. Následující odstavce popisují barvu jako fyzikální jev, představí lidské vnímání barvy, a také modely, které se používají v digitální reprezentaci. Tato kapitola čerpá zejména ze zdrojů [3].

### 1.2.1. Světelné spektrum

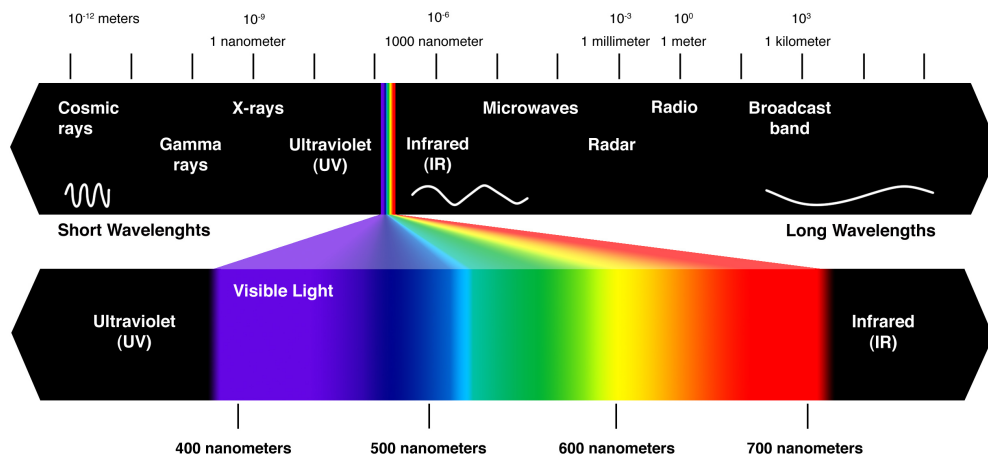
Světlo je v podstatě elektromagnetické záření. To může mít různou vlnovou délku, která se obvykle označuje jako  $\lambda$ . Vlnovou délku lze charakterizovat jako vzdálenost mezi vrcholy jednotlivých vln elektromagnetického záření. Neboli:

$$\lambda = c/f \tag{1.2}$$

kde  $c$  je rychlost světla ve vakuu, která je rovna 299 792 458 m/s [3] a  $f$  je frekvence vlnění. Záření může dosahovat jak velmi krátkých vlnových délek, takovéto záření se nazývá *ultrazářením*, nebo velmi dlouhých vlnových délek, neboli *infrazářením*. Jako viditelné světlo potom označíme takové vlnění, jehož délka je někde „uprostřed“, a významné je tím, že je pro člověka viditelné. O viditelném světle hovoříme při vlnových délkách přibližně od 350 do 780 nanometrů (nm).[3]

Nejnižší frekvence viditelného světla člověk vnímá jako červenou barvu a s rostoucí frekvencí (se zmenšující se vlnovou délkou) je postupně světlo vnímáno jako oranžová, žlutá, zelená, indigová dále modrá až po největší frekvenci, kterou nazýváme fialovou (viz Obr. 2).

Je dobré si také uvědomit, že každý objekt pohlcuje určité vlnové délky a jiné odráží, čímž je tvořena barva daného objektu.

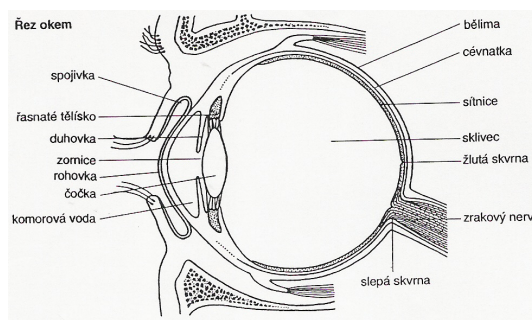


Obr. 2: Světelné spektrum [5]

### 1.2.2. Lidské oko

Proto, aby byl navrhnout digitální systém zobrazení barevných obrazů, je dobré pochopit, jak lidské oko funguje. O fyzikálních vlastnostech světla je toho známo hodně. Již méně se ví o tom, jak přesně oko pracuje. V další sekci práce bylo čerpáno z [3]

#### Fyziologie lidského oka



Obr. 3: Řez lidským okem [8]

Na Obr. 3 můžeme vidět řez okem. Přední část oka se nazývá rohovka. Je to průhledná vrstva, která chrání další části našeho zrakového orgánu. Jedná se v podstatě o „okno“, které umožňuje světlu vstoupit do oka. Dále světlo postoupí do kruhového vstupu, nazývaného zornice, který je obklopen duhovkou. Zornice

má černou barvu, protože se jedná o průchod, skrz který všechno světlo postupuje dále do oka. Výkyvy v intenzitě světla ovlivňují chování duhovky, která se může zmenšovat nebo naopak zvětšovat. Světlo je dále upraveno čočkou, která také obraz obrátí vzhůru nohama. Světlo poté putuje sklivcem, který zabírá 80% objemu oka a je obklopen sítnicí, která obsahuje světločivé buňky (neboli receptory) - čípky a tyčinky. Čípky obsahují tři typy pigmentu, díky kterým čípky reagují zejména na určité vlnové délky, které přibližně odpovídají třem barvám. Konkrétně se jedná o:

1. Chlorolabe - vnímá zejména odstíny zelené barvy
2. Erythrolabe - vnímá zejména odstíny červené barvy
3. Cyanolabe - vnímá zejména odstíny modré barvy

Vidění pomocí čípků se nazývá *fotopické vidění*. Problémem je, že čípky přestávají být efektivní při malé světelné intenzitě. V tom případě se aktivují druhé světločivé buňky, tyčinky. Tyčinky dokážou reagovat už na dopad pouze jediného fotonu, zato ovšem rozlišují pouze intenzitu. Takovéto vidění se nazývá *mezopické*.

### **1.2.3. Barevné vidění**

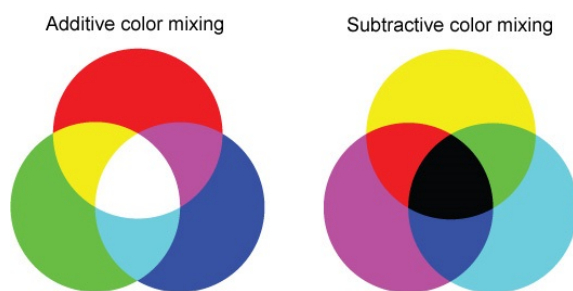
Jak již víme, světločivé buňky nejvíce reagují na tři barvy: zelenou, červenou a modrou. Ještě dříve než bylo možné ověřit, že jsou okem rozeznávány pouze tyto tři barvy, přišel v roce 1802 Thomas Young s teorií, že tomu tak je a tuto teorii nazval trichromazie. Pojmem tristimulace rozumíme experiment, kdy se subjekt snaží vytvořit součet tří barevných složek tak, aby se barva shodovala se zadanou barvou. Tristimulace je vlastně částečný důkaz trichromazie. Díky tristimulaci jsme zjistili, že při správně zvolených základních barvách lze jejich kombinací vytvořit téměř jakoukoli jinou barvu. Navíc se ukazuje, že dvě složky by nebyly dostačující a čtyři již nejsou nezbytné.

### 1.2.4. Tvorba barev

Z předchozí kapitoly víme, že kombinace vhodně zvolených tří barev nám umožňují vytvořit jakoukoli jinou barvu. Následující odstavce popisují, jak to funguje.

První možnost využívá tzv. aditivních barev. Pokud zvolíme překrývající se spektra, které rozdělují viditelné spektrum přibližně na třetiny, pak každá vyvolá reakci u jednoho z receptorů. Právě pokud rozdělíme viditelné spektrum na třetiny, tak získáme světlo v barvách, kterým dnes říkáme zelená, červená a modrá. Začneme od černé, která neobsahuje žádné světlo, a postupně budeme přidávat do výsledného světla jednotlivé vlnové délky, dokud nezískáme bílé světlo.

Druhá možnost, subtraktivní barevná teorie, je v podstatě opačná. Již nevyhází z červené, zelené a modré, ale místo toho z azurové, purpurové a žluté. Rozdílem také je, že barvy již nepřičítáme k černé, ale odčítáme z bílého zdroje světla. Tedy azurová je vlastně odečtení světla s velkou vlnovou délkou (odpovídající červené) od bílého světla, purpurovou dostaneme jako odečtení světla se střední vlnovou délkou (zelená) a žlutá je vlastně odečtení světla s krátkou vlnovou délkou (modrá).



Obr. 4: Aditivní a subtraktivní model [6]

### 1.2.5. Barevný model

Účelem barevného modelu, někdy nazývaný také barevný prostor, je vytvoření specifických barevných standardů. Jedná se v podstatě o vytvoření koordinačního prostoru, kdy v určitém podprostoru tohoto prostoru každý bod reprezentuje

konkrétní barvu. Dnes jsou barevné modely založené na hardwarovém zobrazení, popřípadě na aplikacích, které s barvami pracují. Mezi nejčastěji používané modely patří model RGB (red, green, blue), CMY (cyan, magenta, yellow), což je model používaný pro tisk, a HSV (hue, saturation, value), který je podobný tomu, jak rozlišuje barvy člověk. Model HSV je výhodný v tom, že jeho rozkladem získáme také jednu složku, která odpovídá barvě. To je výhodné, protože existují metody zpracování obrazu, které jsou vhodné pouze pro černobílé (jednosložkové) obrazy.

Praktická část této práce využívá modelu RGB a HSV. Proto je těmto modelům věnováno více prostoru v následujících sekcích.

## **Model RGB**

RGB model [3] je v podstatě aditivní model, jak jsme si ho popsali v kapitole (1.2.4). V RGB modelu se tedy každá barva jeví jako kombinace červené, zelené a modré. Model je založen na kartézském systému souřadnic. Všechny barvy pak tvoří krychli, kdy vrcholy na souřadnicích odpovídají základním barvám (RGB) a ostatní vrcholy odpovídají CMY (azurové, fialové a žluté). Počátek souřadnic odpovídá černé. Bílá je nejvzdálenější vrchol od počátku. Spojením černé a bílé barvy bychom dostali úsečku, která odpovídá stupňům šedi.

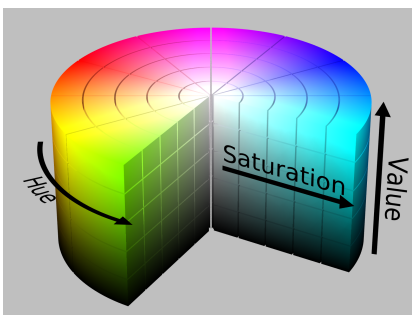
Obrazy v tomto modelu jsou reprezentovány třemi komponenty. Každá komponenta je vlastně číslo od 0 do 1, což odpovídá množství dané barvy. V tomto digitálním zobrazení, například na RGB monitoru, jsou hodnoty těchto barev pro každý pixel zkombinovány v barevný obraz. Důležité je i to, jaký software používáme pro zobrazení RGB modelu. Může se stát, že na různých zařízeních může být stejný RGB obraz zobrazen trochu rozdílně.



Obr. 5: Rozklad modelu RGB [Autor práce]

## Model HSV

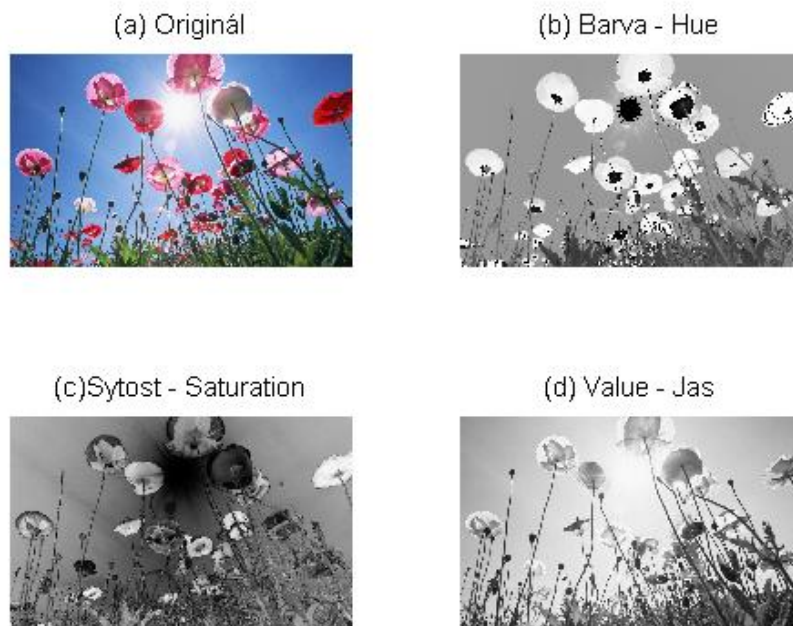
Jak jsme mohli vidět v předchozích odstavcích, RGB model je vhodný pro hardwarové zobrazení obrazu, a navíc je podobný tomu, jak lidské oko vnímá barvy.



Obr. 6: HSV model [7]

Problém je v tom, že o barvě nehovoříme jako o procentuální skladbě ze tří základních barev. Proto se zavádí HSV model [3], který více odpovídá tomu,

jak lidé barvy popisují. Tento model je také tří-složkový, ale v tomto jsou jednotlivými složkami barva, sytost a jas. HSV se zobrazuje pomocí válcové soustavy souřadnic (viz Obr. 6). Úhel mezi centrální horizontální osou procházející červenou, kterou označíme jako  $0^\circ$ , koresponduje s barvou. Vzdálenost od osy poté určuje sytost. Poslední složkou je výška, která je vnímána jako jas. Právě zobrazení jasu je v digitální podobě velmi problematické z toho důvodu, že každý monitor, popřípadě jiné zobrazovací zařízení má jinak nastavený základní jas. Z tohoto důvodu bylo vyvinuto několik modelů, které se snaží tento problém řešit. Jedná se například o HSI nebo HSB modely. Více o těchto modelech například v [2]. Úhel vždy odpovídá stejné barvě. Konkrétně úhel 0 stupňů odpovídá červené, 120 stupňů zelené a 240 modré. Vertikální osa je tvořena stupni šedi, kdy 0 odpovídá černé a 1 bílé.



Obr. 7: Rozklad modelu HSV [Autor práce]

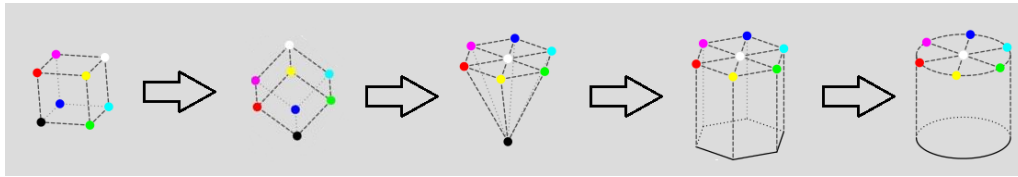


## Převod RGB do HSV

Již bylo zmíněno, že HSV je vlastně nelineární transformací RGB modelu.

Vizuálně tuto transformaci znázorňuje Obr.8.

Jedná se o projekci RGB trojrozměrné krychle do dvourozměrného šestiúhelníku (hexagonu), a postupně až do třírozměrného válce. RGB krychli otočíme tak, aby počátek odpovídal černé barvě. Krychli dále promítneme do hexagonu tak, aby červená odpovídala  $0^\circ$ .



Obr. 8: Převod RGB do HSV [7]

V různé literatuře se vyskytuje několik možných algoritmů této transformace. Níže je použit tradiční algoritmus, popsán například v [9]

$$V_{ij} = \max(R_{ij}, G_{ij}, B_{ij}) \quad (1.3)$$

$$m_{ij} = \min(R_{ij}, G_{ij}, B_{ij}) \quad (1.4)$$

$$c_{ij} = V_{ij} - m_{ij}, \quad (1.5)$$

kde  $i, j$  označuje pozici pixelu a  $V$  odpovídá třetí složce HSV modelu neboli jas. Jas je tedy dán jako největší složka RGB modelu.

Sytost chápeme jako normalizovanou vzdálenost od osy spojující černou a bílou, která představuje stupně šedi.

$$S_{ij} = \begin{cases} 0 & \text{pokud } c_{ij} = 0. \\ \frac{c_{ij}}{V_{ij}} & \text{jinak.} \end{cases} \quad (1.6)$$

Je dobré si uvědomit, že pokud jsou všechny složky RGB modelu stejné, sytost bude nulová, protože  $\frac{x-x}{x} = 0$ . Jedná se totiž o stupně šedi. Naopak pokud budou dvě složky rovny nule a poslední složka bude rovna jedné, sytost bude rovna jedné

$\frac{1-0}{1} = 1$ . Jedná se o případ, kdy má pixel jednu ze základních barev RGB modelu, přičemž tyto barvy jsou nejvzdálenější od černobílé osy.

Nakonec barva odpovídá:

$$H_{ij} = \begin{cases} 0 & c=0 \\ 1 + \frac{G_{ij}-B_{ij}}{6c} & \text{pokud } V = R \\ \frac{2}{6} + \frac{B_{ij}-R_{ij}}{6c} & \text{pokud } V_{ij} = G_{ij} \\ \frac{4}{6} + \frac{R_{ij}-G_{ij}}{6c} & \text{pokud } V_{ij} = B_{ij} \end{cases} \quad (1.7)$$

Barva nabývá hodnot z intervalu  $\langle 0, 1 \rangle$ , což po převedení na stupně (násobkem  $2\pi$ , popřípadě 360 stupni), odpovídá úhlu v hexagonu.

## 1.3. Filtrování

V této části práce je rozebrána teorie filtrů. Postupně se práce zabývá filtrováním v prostorové doméně a následně je využita Fourierova transformace pro filtrování v doméně frekvenční. Bylo využito zejména literatury [1], [2] a [3]

### 1.3.1. Základy filtrování v prostorové doméně

Všechny metody popsané v této části práce jsou prováděny v prostorové doméně. To znamená, že jsou prováděny přímo na hodnotách jednotlivých pixelů. V určitých případech je tento přístup výhodnější, a to zejména z důvodu snazší implementace. Nevýhodou je výpočetní náročnost oproti frekvenčnímu filtrování, kterým se práce zabývá v kapitole (1.3.5).

Operace v prostorové doméně označíme jako

$$g(x, y) = T[f(x, y)], \quad (1.8)$$

kde  $f(x, y)$  je vstupní obraz,  $g(x, y)$  je výstupní obraz a  $T$  je operace na  $f$  definovaná na sousedních pixelech  $(x, y)$ .

Obvykle za sousední body považujeme všechny body, které se dotýkají daného bodu. Tedy i body dotýkající se pouze rohy. Celkem má každý bod 8 takovýchto sousedů. Proto se takovýto systém nazývá *systém 8 sousedů*, popřípadě sousedé

$3 \times 3$ . Čtenář si jistě dokáže přestavit podobný systém, například  $4 \times 4$ , neboli obecně  $n \times n$ .

Operaci  $T$  tedy provádíme postupně na sousedech každého pixelu. Navíc pro každou polohu  $(x, y)$  platí, že výsledný obraz  $g$  je na těchto souřadnicích roven výsledku aplikace operace  $T$  na vstupním obraze v bodech  $(x, y)$ . Problém nastane v krajních pixelech, které nemají 8 sousedů. Tyto body poté buď ignorujeme, nebo metodu vhodně upravíme tak, aby mohla být použita i pro tyto pixely. Procedura popsaná výše se nazývá *prostorová filtrace*. Sousedé spolu s danou operací se poté nazývají *prostorový filtr*. Operace a počet sousedů tedy určují podstatu prostorového filtru.

Účelem prostorové filtrace v této práci je zejména změna obrazu tak, aby byl obraz vhodný pro aplikaci dalších specifických metod. Je dobré zdůraznit, že pro různé aplikace se hodí různé prostorové filtry.

### **Příklady filtračních operací**

Na filtrační operace v této části můžeme nahlížet jako na transformace. Dále v kapitole označíme  $s$  jako hodnotu pixelu před a  $r$  jako hodnotu pixelu po filtraci. Pro hodnoty pixelů platí:

$$s = T(r), \tag{1.9}$$

kde  $T$  je již zmíněná transformace.

V dalším textu budou popsány tři typy funkcí, které jsou často používány pro zlepšení obrazu před dalším zpracováním.

### **Negativní transformace**

Negativní transformace obrazu dostaneme tím, že odstíny černé barvy změníme na odstíny bílé a naopak. Matematicky je tato transformace vyjádřena jako:

$$s = L - r, \tag{1.10}$$

kde  $L$  je maximální možná hodnota pixelu. Tato transformace se používá, pokud potřebujeme zvýraznit bílé nebo šedé části obrazu v černých regionech, zejména pokud černé oblasti dominují celkovému obrazu.

### Log transformace

Klasická forma log transformace má tvar:

$$s = c \cdot \log(1 + r), \quad (1.11)$$

kde  $c$  je konstanta. Vzhledem k hodnotám pixelů předpokládáme, že  $r \geq 0$ . Podle klasického tvaru logaritmicke funkce je zřejmé, že tato transformace zvyšuje rozdíly u malých hodnot pixelů, které obvykle odpovídají černým odstínům. Tato funkce také zmenšuje rozdíly velkých hodnot pixelů. Log transformace se obvykle využívá při zpracování hodně tmavých nebo naopak hodně světlých obrázků<sup>1</sup>.

Dalším klasickým použitím je aplikace pro Fourierovo spektrum, které může obsahovat hodnoty od 0 do velmi velkých hodnot v řádu  $10^6$ <sup>2</sup>. Pokud tyto hodnoty lineárně zobrazíme do 8 bitového systému, tak bude viditelných pouze několik málo nejjasnějších pixelů, a zbylé, stejně důležité, budou černé. Tento problém se dá zmírnit právě logaritmicke transformací.

### Power-Law transformace

Power-Law, někdy také nazývána Gamma transformace má tvar:

$$s = cr^\gamma, \quad (1.12)$$

kde  $c, \gamma$  jsou pozitivní konstanty. Podobně jako u log transformace zvětšujeme rozestupy mezi tmavými, respektive světlými pixely. Rozdílem je, že v tomto případě lze koeficienty jednodušeji ovlivňovat, a tím docílit konkrétního výsledku.

Tato transformace je hojně využívána při tisku a skenování a také při zobrazení například na monitorech. Problémem je, že každý monitor má trochu jiné

---

<sup>1</sup>Při použití u světlých obrazů se prvně použije negativní transformace a až poté log transformace.

<sup>2</sup>Jedná se pouze o ilustraci hodnoty, maximální hodnota Fourierovy transformace závisí na konkrétním vstupním obraze

nastavení. Bez použití Gamma transformace by obrázky na hodně zařízeních mohly působit tmavě. Proto se v praxi používá tato transformace a vytvoří se obrázek tak, aby na „průměrném“ zobrazovacím zařízení působil dobře.

### Transformace na zvětšení kontrastu

Obrazy s malým kontrastem mohou působit málo světle. Obvykle je nízký kontrast dán malými barevnými rozdíly objektů, popřípadě pak chybným nastavením čočky fotoaparátu.

Zvětšování kontrastu je proces zvětšování rozsahu hodnot obrazu, tak aby obraz obsahoval jak největší, tak i nejmenší hodnoty. Nejjednodušší metoda zvýšení kontrastu je v anglické literatuře nazývána *linear mapping*. Metoda spočívá v tom, že největší a nejmenší hodnotu pixelu v obraze označíme jako  $G_{max}$ , respektive  $G_{min}$ . Dále nastavíme  $G_{max} = a$  a  $G_{min} = b$ , kde  $a$  je nejvyšší hodnota, a  $b$  nejmenší hodnota, kterou chceme aby obraz obsahoval. Ostatní hodnoty potom spočteme jako lineární interpolaci mezi  $a$  a  $b$ . Neboli:

$$s = \left( \frac{r - G_{min}}{G_{max} - G_{min}} \right) \cdot b \quad (1.13)$$

### 1.3.2. Prostorové filtrování

Jak již bylo zmíněno, základem prostorových filtrů jsou za prvé susedé, a za druhé operace na nich prováděná. Filtrace vytvoří novou hodnotu pixelu, se stejnou pozicí jako původní pixel. Pokud je operace prováděná na susedech jednotlivých pixelů lineární, pak hovoříme o *lineárním prostorovém filtru*. V opačném případě se jedná o *nelineární prostorový filtr*. Nejprve se budeme věnovat filtrům lineárním.

Pro každý bod  $(x, y)$  platí, že filtrovaný obraz na stejné pozici je součtem produktů koeficientů filtru a hodnot susedních pixelů, neboli:

$$g(x, y) = h(-1, 1)f(x - 1, y - 1) + h(-1, 0)f(x - 1, y) + \dots + h(0, 0)f(x, y) + \dots + h(1, 1)f(x + 1, y + 1) \quad (1.14)$$

Obecně lze lineární prostorový filtr zapsat následovně:

$$(f * h)(x, y) = g(x, y) = \sum_{s=-a}^a \sum_{t=-a}^a f(x + s, y + t)h(s, t), \quad (1.15)$$

kde  $*$  značí konvoluci,  $x, y$  jsou proměnné tak, aby každý pixel v  $h$  prošel každým pixelem v  $f$ .

Při tvorbě filtrů musíme také specifikovat koeficienty filtru. To záleží na tom, co podaném filtru požadujeme. Například můžeme chtít nahradit hodnoty pixelů průměrem hodnot pixelů z okna  $3 \times 3$ . Jedná se poté o průměr devíti hodnot:

$$h(x, y) = \frac{1}{9} \sum_{i=1}^9 z_i, \quad (1.16)$$

kde  $z_i$  jsou hodnoty původních pixelů. Dále jsou všechny koeficienty filtru rovny  $1/9$ . Tato operace povede k vyhlazení obrazu.

Podobně můžeme uvažovat spojitou funkci dvou proměnných. Příkladem je hojně využívaná Gaussovská funkce tvaru:

$$h(x, y) = \frac{1}{2\pi\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1.17)$$

kde  $\sigma$  značí standardní odchylku. Při použití takového filtru intuitivně předpokládáme, že šum v obraze Gaussovsky rozdělen.

Tvar koeficientů Gaussovského filtru pro okno  $3 \times 3$  vypadá následovně:

$$\frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Lineární filtry popsané výše jsou v určitých situacích velmi výhodné. Obvykle poskytují dobré výsledky týkající se zbavení šumu, v případě, že je Gaussovsky rozdělen. Pro určité další zpracování obrazu může být problém, že lineární filtry obvykle obraz rozmažou. Pokud se spíše jedná o ostrý šum, někdy také nazývaný

*salt-and-pepper* šum, je výhodné použít nelineární filtr. Takovýto filtr šum odstraní a navíc obraz tolik nerozmaže.

Nejpoužívanějším nelineárním filtrem je tzv. *mediánový filtr*, který znovu využívá překrývající se okna, kdy centrální pixel je vždy, jak jeho název napovídá, nahrazen mediánem jeho sousedů v okně.

Pro okno o velikosti  $n \times n$  vypadá mediánový filtr následovně:

$$g(x, y) = \max\{f(a, b)\} \quad (1.18)$$

Mediánový filtr samozřejmě není jediným zástupcem nelineárních filtrů. Pokud bychom místo mediánu dosadili do aktuálního pixelu maximální hodnotu jeho sousedů, dostaneme tzv. *maximální filtr*.

### 1.3.3. Práce s histogramem

Histogram digitálního obrazu s hodnotami v rozsahu  $[0, L - 1]$  je diskrétní funkce

$$h(r_k) = n_k, \quad (1.19)$$

kde  $r_k$  je  $k$ -tá hodnota a  $n_k$  je počet pixelů obrazu s hodnotou  $r_k$ . Obvykle se také používá normalizovaný histogram:

$$p(r_k) = \frac{n_k}{M \cdot N}, \quad (1.20)$$

kde  $M \cdot N$  je celkový počet pixelů v obraze. Součet  $p(r_k)$  je poté roven 1.

Histogram je základem pro několika metod zpracování obrazu v prostorové doméně. Úprava histogramu při správné implementaci vede k zlepšení obrazu, zejména pro další zpracování. Například při kompresi nebo segmentaci obrazu. Výhoda metod založených na úpravě histogramu spočívá zejména v výpočetní nenáročnosti.

#### Vyrovnaní histogramu

Uvažujme, že hodnoty pixelů ( $r$ ) jsou spojité. Stále přitom platí, že  $r$  patří do intervalu  $\langle 0, L - 1 \rangle$ , kdy  $r = 0$  reprezentuje černou a  $r = L - 1$  bílou. Dále

uvažujme transformaci  $T$  tvaru:

$$s = T(r), \quad (1.21)$$

která produkuje výstupní hodnotu  $s$  pro každý pixel vstupního obrazu.

Dále předpokládejme:

1.  $T(r)$  je monotónně rostoucí funkce pro  $0 \leq r \leq L - 1$
2.  $T(r)$  je omezená:  $0 \leq T(r) \leq L - 1$  pro  $0 \leq r \leq L - 1$

Pro inverzní funkci poté platí:

$$r = T^{-1}(s) \quad (1.22)$$

přičemž  $0 \leq s \leq L - 1$ . První podmínka zaručuje, že výstupní hodnota nebude menší než vstupní hodnota. Druhá podmínka zaručuje to, že rozsah původních hodnot bude stejný jako rozsah výstupních hodnot. V praxi jsou ovšem hodnoty pixelů diskrétní, a proto je nutné po použití určité funkce zaokrouhlit na nejbližší celé číslo.

Na hodnotu pixelů se můžeme dívat jako na náhodnou proměnnou z intervalu  $[0, L - 1]$ . Popis této náhodné proměnné může dát *hustota pravděpodobnosti* (PDF). Nechtě  $p_r(r)$  a  $p_s(s)$  jsou PDF z  $r$ , respektive  $s$ . Dolní index u  $p_r$  a  $p_r$  značí pouze to, že se obecně jedná o rozdílné funkce. Pokud jsou  $p_r(r)$  a  $T_r$  známy a dále pro  $T_r$  platí, že je spojitá a diferencovatelná na požadovaném intervalu, pak PDF transformované proměnné  $s$  lze získat vzorcem:

$$p_s(s) = p_r(r) \left| \frac{dr}{ds} \right| \quad (1.23)$$

Dále vidíme, že PDF výsledných hodnot proměnné  $s$  je určena PDF vstupních hodnot a funkcí použitou při transformaci. Transformační funkci můžeme přepsat do tvaru:

$$s = T(r) = (L - 1) \int_0^r p_r(\omega) d\omega \quad (1.24)$$



Pravá část rovnice se nazývá kumulativní distribuční funkcí (CDF) náhodné proměnné  $r$ .

Z toho, že PDF je vždy kladná, vychází, že transformační funkce splňuje první podmínku - protože plocha pod funkcí nemůže poklesnout, když zvětšujeme  $r$ . Pokud je horní limit v této rovnici roven  $r = L - 1$ , integrál se vyhodnotí jako 1, neboli maximální hodnota je  $L - 1$ , čímž je uspokojena i druhá podmínka.

Pro nalezení  $p_{sd}(s)$  použijeme tedy rovnici 1.26

$$\frac{ds}{dr} = \frac{dT(r)}{dr} = (L - 1) \frac{d}{dr} \left[ \int_0^r p_r(\omega) d\omega \right] = (L - 1) p_r(r) \quad (1.25)$$

Substitucí  $dr/ds$  v rovnici dostaneme

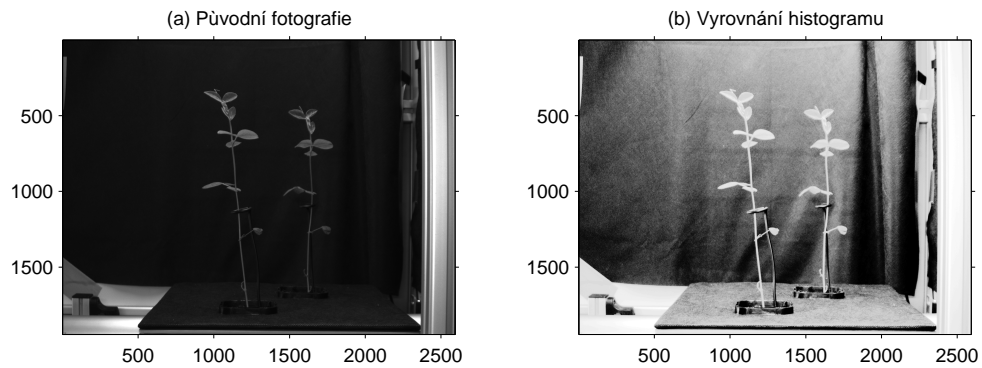
$$p_s(s) = p_r(r) \left| \frac{dr}{ds} \right| = p_r(r) \left| \frac{1}{(L - 1) p_r(r)} \right| = \frac{1}{L - 1} \quad (1.26)$$

Což je rovnice pro rovnoměrnou pravděpodobnostní funkci. Bylo tedy ukázáno, že při použití transformace v rovnici (1.27) dostaneme náhodnou proměnnou  $s$  charakterizovanou rovnoměrnou PDF. Je důležité říci, že výsledný  $p_s(s)$  je vždy rovnoměrný a to nezávisle na  $p_r(r)$ .

Diskrétní forma této transformace vypadá následovně.

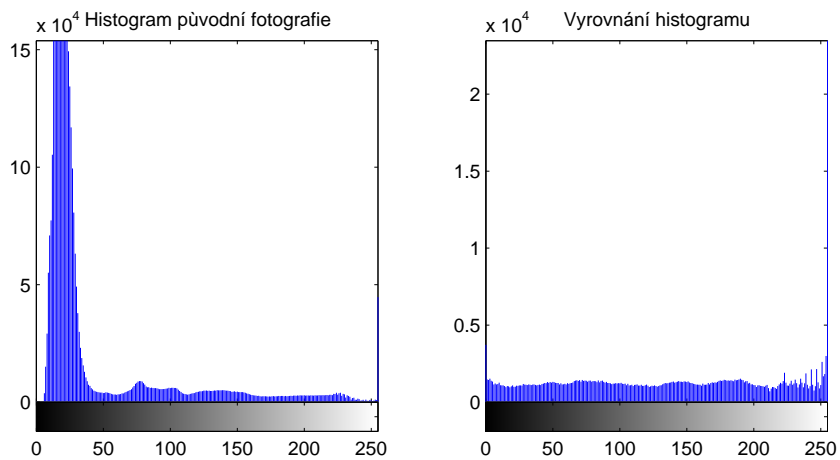
$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j) = \frac{(L - 1)}{MN} \sum_{j=0}^k n_j, \quad (1.27)$$

kde  $k = 0, 1, 2, \dots, L - 1$ .



Obr. 9: Vyrovnání histogramu [Autor práce]

Takto transformovaný obraz je získán zpracováním každého pixelu původního obrazu použitím rovnice (1.27). Transformace  $T(r_k)$  v této rovnici se poté nazývá *histogram equalization* - vyrovnání histogramu.



Obr. 10: Vyrovnání histogramu [Autor práce]

### 1.3.4. Filtrování v frekvenční doméně

Lineární filtrování lze provádět i ve frekvenční doméně. Navíc se ukazuje se, že tento způsob je rychlejší. Pro převod dvourozměrných signálů, kterým je i obraz, do frekvenční domény se využívá Fourierova transformace. V této části je popsána nejprve klasická, jednorozměrná Fourierova transformace. Dále je zde rozebrána dvourozměrná Fourierova transformace a nakonec teorie filtrů ve frekvenční doméně s konkrétními příklady.

Část textu je přezvaná z autorovy bakalářské práce [16], dále bylo čerpáno zejména ze zdrojů [1] a [3].

### 1.3.5. Fourierova transformace

Fourierova transformace dostala svůj název po francouzském matematikovi Jean-Baptiste Joseph Fourierovi, žijícím na přelomu 18. století. Základní myšlenkou předcházející Fourierově transformaci bylo to, že lze libovolnou periodickou funkci reprezentovat trigonometrickou řadou. To znamená, že jí lze rozložit na součty sinusoid s různými periodami. K takovému rozkladu se používají Fourierovy řady. Podmínku periodicity pak zjemňuje Fourierův integrál, který předpokládá, že každá funkce je periodická, s tím, že některé funkce jsou periodické s nekonečnou periodou. Tato práce se nezabývá konkrétním rozбором Fourierovy řady, ani Fourierovým integrálem. Tento rozbor je popsán v [?].

Fourierova transformace je v dnešní době velice využívána. Používá se například při řešení diferenciálních rovnic, při rychlém řešení násobení velkých čísel, v oblasti zpracování signálu a podobně. Je také základem zpracování a analyzování obrazu. Fourierova transformace převádí signály z časové oblasti do oblasti frekvenční. V různé literatuře, se definice Fourierovy transformace mírně liší, v této práci je využita následující definice:

Nechť  $f(t)$  je reálná nebo komplexní funkce s reálnou proměnnou  $t$ . Pak její Fourierova transformace  $f(\nu)$  s reálnou proměnnou  $\nu$  je definovaná jako

$$F(\nu) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i\nu t} dt \quad (1.28)$$

kde  $t$  reprezentuje čas a  $\nu$  reprezentuje frekvenci.  $|F(\nu)|d\nu$  nazývá se spektrem a měří energii nesenou mezi frekvencemi  $\nu$  a  $\nu + d\nu$ . Inverzní Fourierova transformace je potom dána vztahem:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\nu)e^{2\pi i\nu t} d\nu \quad (1.29)$$

Inverzní Fourierova transformace je „opakem“ Fourierovy transformace. Převádí totiž signál z frekvenční domény do domény časové.

### Diskrétní Fourierova transformace

Fourierova transformace je vhodná pro použití při zpracování spojitých signálů, ale i tak někdy neexistuje analytické řešení a Fourierova transformace se musí spočítat numericky. Problém také nastane, když chceme zpracovat digitální, tedy diskrétní, signály. U těchto signálů nemáme spojitou funkci, a tedy Fourierovu transformaci nemůžeme použít. V dnešní době je však většina signálů zpracovávána pomocí počítačů a tedy digitálně. Proto byla zavedena diskrétní Fourierova transformace, zkráceně DFT, která pracuje právě s diskretizovanými funkcemi.

Pokud zobecníme rovnici (1) pro diskrétní případ, kdy platí, že vstupní signál  $f(t)$  je nyní roven diskrétní posloupnosti  $f(t_k) = \{f_0, f_1, \dots, f_{N-1}\}$  a označením  $f_k \equiv f(t_k)$ , kde  $t_k \equiv \Delta k$  pro  $k = 0, 1, \dots, N - 1$ , pak Diskrétní Fourierova transformace  $f_n$  je definována jako

$$F_k = \sum_{n=0}^{N-1} f_n e^{-2\pi i \frac{nk}{N}} \quad (1.30)$$

Inverzní diskrétní Fourierovu transformaci (IDFT) pak dostaneme úpravou rovnice (4), kterou vynásobíme  $e^{2\pi i \frac{kl}{N}}$  a vytvoříme součty podle  $k$ .

$$\sum_{k=0}^{N-1} F_k e^{2\pi i \frac{kl}{N}} = \sum_{n=0}^{N-1} f_n \left[ \sum_{k=0}^{N-1} e^{2\pi i \frac{k(l-n)}{N}} \right] \quad (1.31)$$

Pokud  $l \neq n$  pak suma v hranatých závorkách je rovna  $N$  a pokud  $l = n$  pak je tato suma rovna nule. Dostaneme tedy:

$$f_l = \frac{1}{N} \sum_{k=0}^{N-1} F_k e^{2\pi i \frac{kl}{N}}, \quad (1.32)$$

kde  $l = 0, 1, 2, \dots, N-1$ . Podobně jako u spojité verze Fourierovy transformace  $f_k$  označuje signál, v tomto případě diskrétní,  $F_n$  označuje diskrétní Fourierovu transformaci, jejíž množina hodnot se nazývá spektrum.

### Dvourozměrná diskrétní Fourierova transformace

Fourierovu transformaci je možno rozšířit do dvourozměrné formy, která je využívána při aplikacích frekvenčních filtrů na obrazy. Dvourozměrná Fourierova transformace má tvar:

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-2\pi i \left( \frac{ux}{M} + \frac{vy}{N} \right)}, \quad (1.33)$$

kde  $f(x, y)$  je digitální obraz velikosti  $M \times N$ . Dále  $u = 0, 1, 2, \dots, M-1$ ,  $v = 0, 1, 2, \dots, N-1$ .

Inverzní transformace má tvar:

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{-2\pi i \left( \frac{ux}{M} + \frac{vy}{N} \right)}, \quad (1.34)$$

přičemž hodnoty  $(u, v)$  jsou nazývány frekvence.

### 1.3.6. Základy filtrování ve frekvenční doméně

Filtrování obrazu ve frekvenční doméně spočívá převedení do frekvenční domény pomocí Fourierovy transformace, dále její úpravou následněm spočtení inverzní

transformace tak, abychom abychom dostali výsledek.

Mějme digitální obraz  $f(x, y)$  velikosti  $M \times N$ . Poté lze filtrační rovnice zapsat jako:

$$g(x, y) = F^{-1}[H(u, v)F(u, v)], \quad (1.35)$$

kde  $F^{-1}$  je IDFT a  $F(u, v)$  je DFT vstupního obrazu  $f(x, y)$ . Dále  $H(u, v)$  je tzv. *filtrační transferová funkce*, někdy nazývaná *filtr*, nebo *filtrační funkce*. Nakonec  $g(x, y)$  je výsledný, filtrovaný obraz. Funkce  $F, H, g$  mají stejnou velikost  $M \times N$ , tedy stejnou jako vstupní obraz. Rovnice (1.35) je dokázaná například v [1].

Je dobré si uvědomit, vztah mezi prostorovou a frekvenční doménou. Víme, že frekvence je úměrná prostorovým změnám. Z toho vyplývá, že malé frekvence odpovídají pomalu se měnícím objektům na obraze, například jednobarevným stěnám nebo bezmračnému nebi. Vyšší frekvence poté odpovídají rychlejším a rychlejším změnám odpovídajícím hranám nebo šumu.

Proto můžeme očekávat, že filtr oslabující pouze vysoké frekvence a neupravující nízké frekvence obraz rozmaže. Takovýto filtr se nazývá *lowpass filtr*. Opačný filtr tzv. *highpass filtr* způsobí zostření hran, ale sníží kontrast obrazu.

Dvourozměrná diskretní Fourierova transformace je obecně komplexní. Proto může být vyjádřena v polárních souřadnicích:

$$F(u, v) = |F(u, v)|e^{i\phi(u, v)}, \quad (1.36)$$

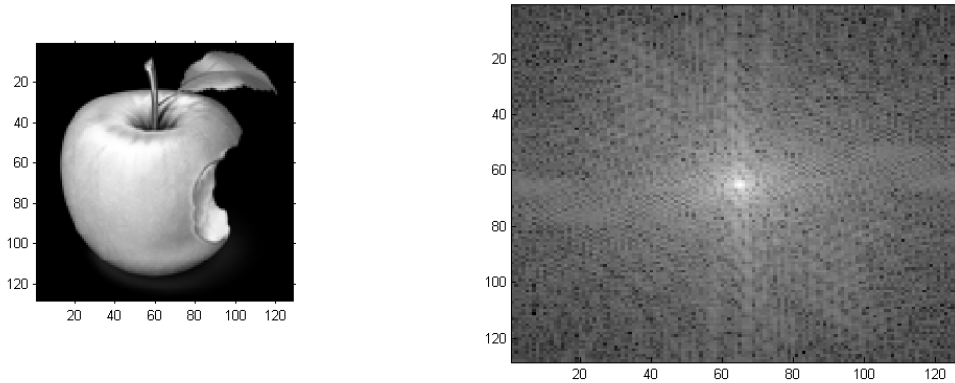
kde

$$|F(u, v)| = [R^2(u, v) + I^2(u, v)]^{\frac{1}{2}}, \quad (1.37)$$

se nazývá Fourierovo spektrum. Dále  $R(u, v)$  představuje reálnou a  $I(u, v)$  imaginární část Fourierovy transformace. Dále

$$\phi(u, v) = \arctan \left[ \frac{I(u, v)}{R(u, v)} \right] \quad (1.38)$$

se nazývá *fázový úhel*. Nakonec



Obr. 11: Fourierovo spektrum (vpravo) konkrétního snímku (vlevo) [Autor práce]

$$P(u, v) = |F(u, v)|^2 = R^2(u, v) + I^2(u, v) \quad (1.39)$$

se nazývá *výkonové spektrum*. Pro připomenutí,  $u$  a  $v$  jsou diskrétní proměnné nabývající hodnot  $u = 0, 1, 2, \dots, M - 1$  a  $v = 0, 1, 2, \dots, N - 1$ . Proto  $F(u, v)$ ,  $\phi(u, v)$  a  $P(u, v)$  mají velikost  $M \times N$ .

Navíc platí, že Fourierova transformace reálné funkce je symetrická<sup>1</sup>, platí tedy:

$$F(u, v) = F(-u, -v) \quad (1.40)$$

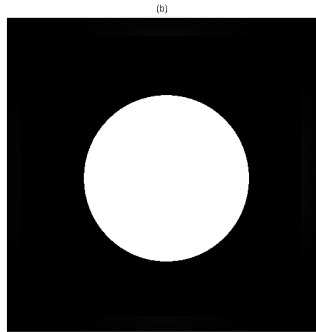
### Ideální lowpass filtr

Ideální lowpass filtr odstraní všechny frekvence kromě určitého kruhového rádiusu o velikosti  $D_0$  kolem centra. Je specifikován funkcí:

$$H(u, v) = \begin{cases} 1 & \text{pokud } D(u, v) \leq D_0 \\ 0 & \text{pokud } D(u, v) > D_0 \end{cases} \quad (1.41)$$

kde  $D(u, v)$  je vzdálenost mezi body  $(u, v)$  ve frekvenční doméně a centrem.

<sup>1</sup>Důkaz tohoto tvrzení je popsán například v [1]



Obr. 12: Ideálního lowpass filtr [Autor práce]

### Gaussovský lowpass filtr

Gaussovský lowpass filtr je v aplikacích hojně využíván. Jedná se o filtr tvaru:

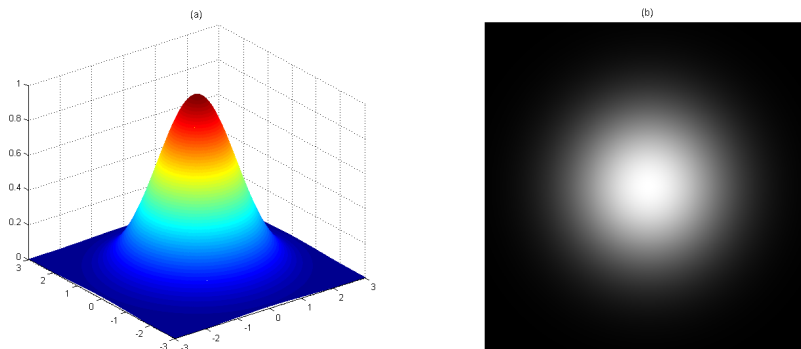
$$H(u, v) = \begin{cases} G(u, v) & \text{pokud } D(u, v) \leq D_0 \\ 0 & \text{pokud } D(u, v) > D_0 \end{cases} \quad (1.42)$$

kde

$$G(u, v) = e^{-\frac{1}{2}[(s_u u)^2 + (s_v v)^2]} \quad (1.43)$$

kde  $D(u, v)$  znovu označuje vzdálenost bodu  $(u, v)$  od centra a  $s_u, s_v$  jsou zvolené směrodatné odchylky.

Gaussovský lowpass filtr je často používaným rozšířením klasického Gaussovského filtru. Ten není použit pouze na body v určité vzdálenosti od centra, ale je použit na všechny frekvence.



Obr. 13: Gaussovský lowpass filtr [Autor práce]



## 1.4. Cannyho detekce hran

Cannyho detekce hran je jednou z nejpoužívanějších metod zpracování obrazu, v tom případě, že nás zajímají tvary nebo velikosti předmětů v obraze. Cannyho algoritmus byl navržen roku 1986 Johnem F. Cannyem [11]. Je nutno říci, že existuje více algoritmů na rozpoznání hran, ale tento je i v dnešní době díky rychlosti a kvalitě velmi oblíbený.

Problémem je to, že obraz obvykle obsahuje velké množství informací. Přitom je pro určitý typ zpracování obrazu důležitá pouze malá část těchto informací. Cílem algoritmu je tedy nalezení těchto důležitých informací (v našem případě nalezení hran objektů) a jejich zdůraznění.

Dále je nutno podotknout, že originální Cannyho algoritmus pracuje s černobílými obrazy. Byly ovšem vyvinuty zobecnění, díky kterým je možno pracovat i s obrazy barevnými. Více informací o těchto zobecněních lze nalézt například v [10].

Při implementaci algoritmu se vycházelo ze tří cílů:

1. Detekce: chceme, aby pravděpodobnost toho, že najdeme reálné hrany byla co největší a zároveň chceme minimalizovat pravděpodobnost odhalení falešných hran. Toto odpovídá maximalizaci tzv. *Signal-to-noise ratio* (SRN), neboli poměru informací a šumu v obraze.
2. Lokace: nalezené hrany by měli být co nejblíže skutečným hranám
3. Počet hran: jedna hrana by měla být nalezena pouze jednou

Samotný algoritmus [12] se skládá z pěti kroků:

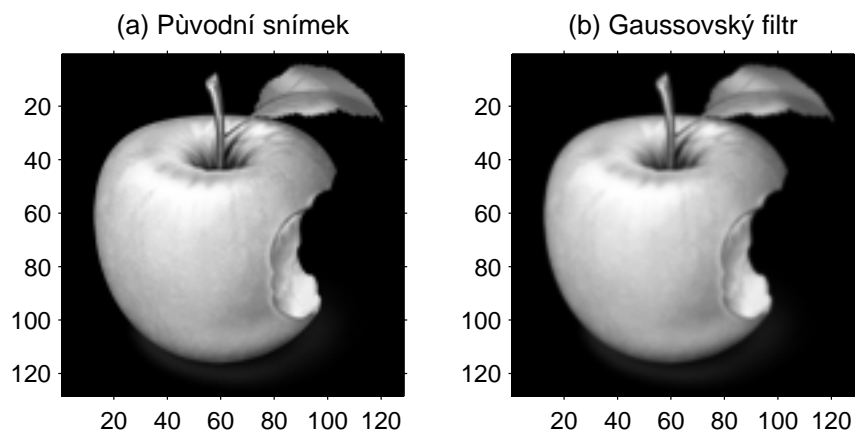
1. Vyhlazení: jedná se o filtraci obrazu za účelem odstranění šumu
2. Hledání gradientů: základním konceptem je, že objekt má jiné hodnoty pixelů než pozadí. Proto je hrana charakteristická velkým gradientem.
3. Lokální maxima gradientů: za potencionální hrany považujeme pouze lokální maxima gradientů

4. Dvojití prahování: potenciální hrany jsou dále určeny pomocí prahů
5. Sledování hran: způsob odstranění pravých hran od falešných

Kroky jsou postupně popsány v následujících odstavcích.

#### 1. Vyhlazení

Základním prvkem každého zpracování obrazu, včetně Cannyho algoritmu, je zbavení se šumu v obraze. Díky tomu snížíme pravděpodobnost, že šum bude považován za hranu objektů v obrázku. Obvykle se k odstranění šumu používá Gaussovský filtr (viz kapitola 1.3.6). Použití tohoto filtru obvykle vede k „rozmazání“ obrazu, ale k dobrým výsledkům při rozpoznání hran.



Obr. 14: Krok 1: Vyhlazení [Autor práce]

#### 2. Hledání gradientů

V tomto kroku sledujeme intenzitu šedé barvy jednotlivých pixelů a náhlé rozdíly v těchto hodnotách budeme považovat za hrany. Hledáme vlastně

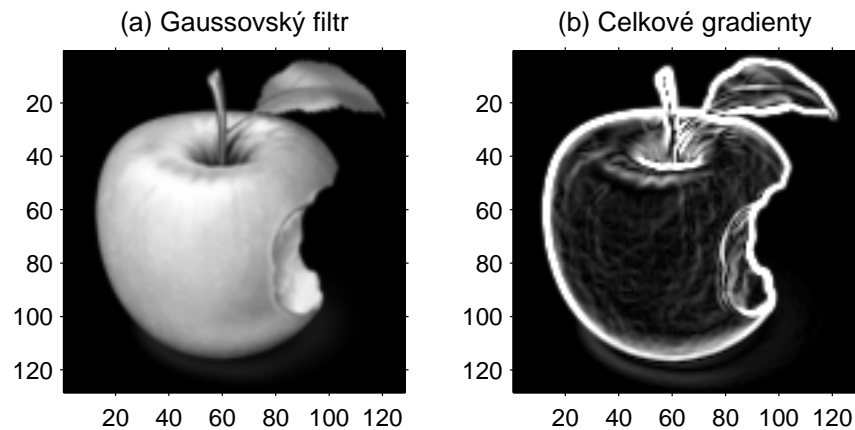
gradienty obrazu. Na ně můžeme pohlížet za dvou směrů. Konkrétně z x-ového směru a y-ového směru. Celkové velikosti gradientu potom určíme jako Eukleidovskou normu, popřípadě normu  $L^1$  těchto dvou gradientů.

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (1.44)$$

$$|G| = |G_x| + |G_y| \quad (1.45)$$

Z obrázku (číslo) lze vidět, že gradienty často naleznou hrany. Hrany jsou ovšem často široké, a proto neurčují přesně, kde se hrany nacházejí. Tímto problémem se zabývá další krok, ale již nyní je potřeba uložit hodnoty směru hran. Směr pixelu je potom určen následující rovnicí.

$$\theta = \arctan\left(\frac{|G_y|}{|G_x|}\right) \quad (1.46)$$



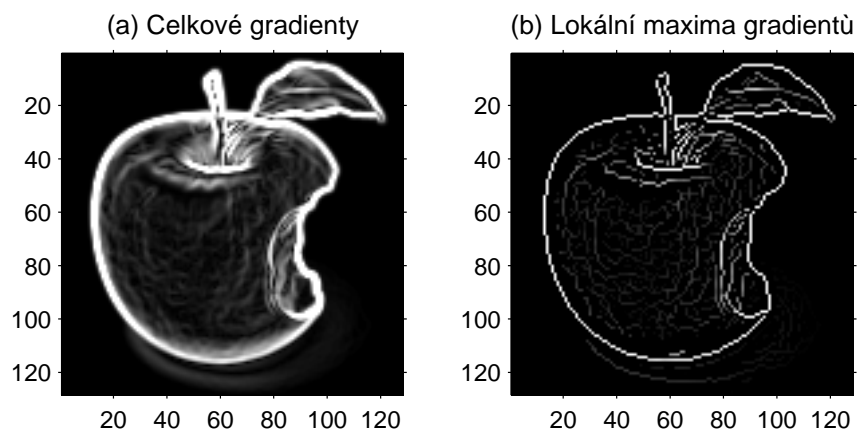
Obr. 15: Krok 2: Hledání gradientů [Autor práce]

### 3. Lokální maxima gradientů

Dále se s ohledem na hodnoty celkových gradientů snažíme přeměnit rozmazané hrany na hrany ostré. Dosáhneme toho zachováním všech lokálních maxim celkových gradientů, podle kroků popsaných níže, a vynulováním všech ostatních gradientů.

Provedeme tedy tyto kroky:

- (a) Zaokrouhlíme směr gradientů na nejbližších  $45^\circ$
- (b) Porovnáme hodnoty gradientu v pixelu s hodnotou pixelů v pozitivním a negativním směru. Například, pokud je  $\Theta = 90^\circ$ , pak srovnáváme s pixelem nahoře a dole.
- (c) Pokud je hodnota gradientu daného pixelu největší z porovnávaných pixelů, pak hodnotu tohoto gradientu zachovej, jinak ho vynuluj.

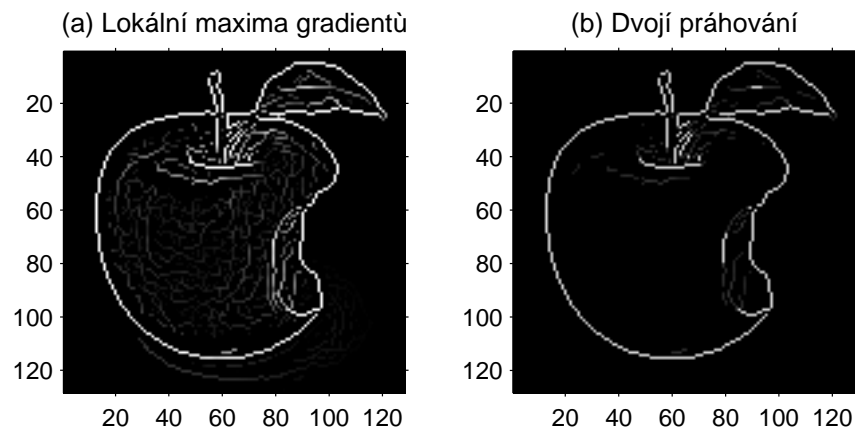


Obr. 16: Lokální maxima gradientů [Autor práce]

#### 4. Dvojí prahování

Hodnoty gradientů stále mají svou hodnotu. Některé jsou nejspíše pravé hrany v obraze, ovšem jiné mohou být způsobeny šumem nebo například nerovným nebo drsným povrchem.

Nabízí se využít jednoduchý práh, tedy potlačení hran, které jsou menší než daná hodnota. V tomto případě ovšem můžeme přerušit spojitost hran objektu. Proto využijeme dvojího prahování. Dvojí prahování funguje tak, že stanovíme dvě hodnoty  $T$ ,  $t$ . Pokud celková hodnota gradientu pixelu bude větší než  $T$ , pak pixel označíme jako silnou hranu. Pokud bude hodnota menší než  $t$  pak gradient potlačíme, protože hrana pro nás není důležitá. Pokud je hodnota mezi hodnotami  $T$ ,  $t$ , pak pixel označíme jako slabou hranu.



Obr. 17: Dvojí prahování [Autor práce]

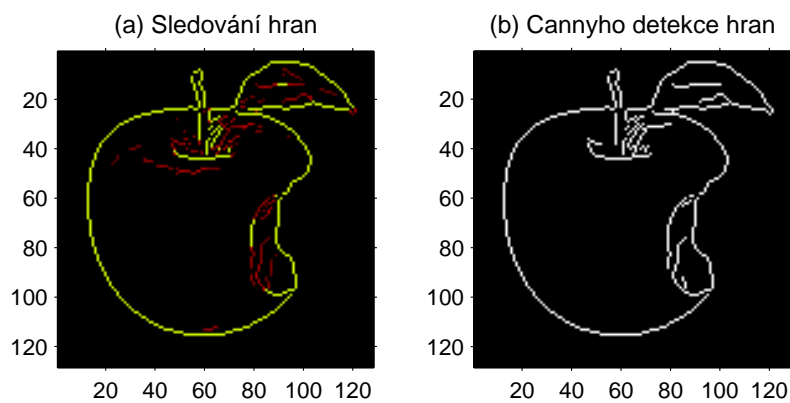
#### 5. Sledování hran

V posledním kroku rozhodujeme, které hrany jsou pravé, a které falešné.

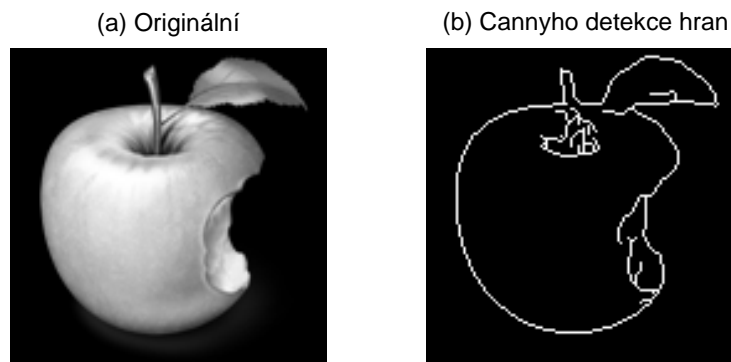
Silné hrany rovnou označíme jako jisté a zahrneme je do výsledného obrázku jako hrany objektů. Slabé hrany považujeme za pravé pouze pokud jsou napojené na silné hrany.

Tento krok vychází z toho, že falešné hrany obvykle stojí samostatně a nebudou příliš silné. Naopak některá části hran mohou být, například špatným osvětlením, málo viditelné, ale díky svojí návaznosti na viditelné části hran je také zahrneme.

Tato část algoritmu je založena na podobném způsobu, jako objekty vnímají lidé. Lidé sledují hrany a pokud jsou přerušeny automaticky se snažíme domyslet si, kam by hrana pokračovala.



Obr. 18: Sledování hran [Autor práce]



Obr. 19: Cannyho detekce hran [Autor práce]

## 1.5. Detekce hran pomocí fuzzy logiky

Dalším způsobem, detekce hran v obraze je *fuzzy logika*. V této práci jsou popsány dva způsoby využívající fuzzy logiku.

### 1.5.1. Bodová fuzzy metoda

Na rozdíl od Cannyho detekce hran, kde základem byla práce s gradienty, je tato metoda založena na myšlence báze pravidel, která zpracovává změny v hodnotách jednotlivých pixelů. Ukazuje se totiž, že Cannyho metoda je velmi citlivá na šum. Pokud se tedy šum neodstraní, Cannyho metoda selhává. Naopak metoda používající fuzzy logiku je na šum méně citlivá, a proto se jeví v určitých situacích výhodněji. Tato metoda je popsána v [14].

Jedná se o Mamdaniho metodu [13]. Podstata této metody spočívá ve třech krocích:

1. Fuzzifikace

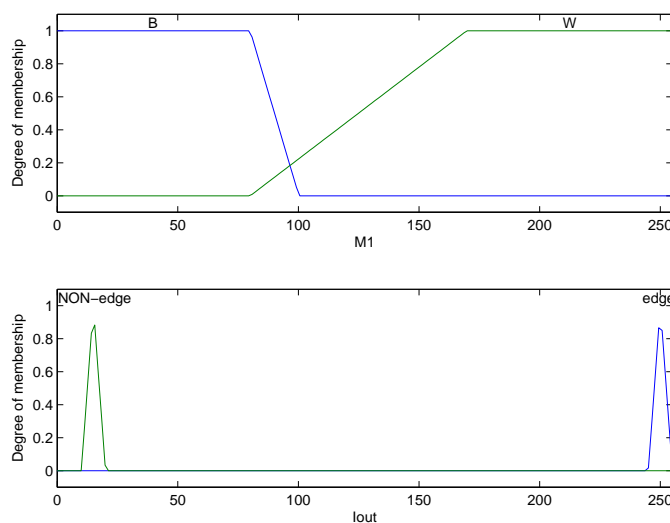
## 2. Báze pravidel

## 3. Defuzzifikace

Je nutno dodat, že metoda se používá pouze pro černobílé obrazy, proto musí být vstupní obraz převeden do stupňů šedi. V algoritmu budeme předpokládat, že hodnoty pixelů se pohybují v rozmezí od 0 do 255.

### 1. Fuzzifikace

V prvním kroku vytvoříme fuzzy množiny „Černá“ a „Bílá“, které přiřazují jednotlivým pixelům příslušnost k těmto fuzzy množinám, viz Obr. 20. Výsledná hodnota pixelu je také tvořena fuzzy množinou, kdy je hodnota označena jako „hrana“ nebo „nehrana“. V případě hrany je výsledná hodnota z intervalu od 245 do 255, v případě nehrany pak od 10 do 20.



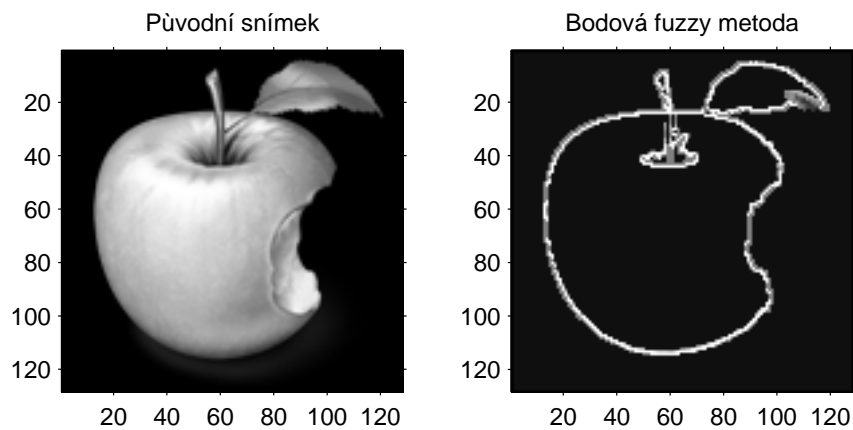
Obr. 20: Vstupní (nahore) a výstupní (dole) fuzzy množina [Autor práce]

## 2. Báze pravidel

V tomto kroku je každý pixel vyhodnocován na základě jeho sousedů, neboli v okně  $n \times n$ , kolem zkoumaného pixelu. Obvykle se za  $n$  volí 3, protože při větším  $n$  by báze pravidel neúměrně rostla. Jednotlivé pixely v tomto okně poté nazveme  $M1, M2, \dots, Mn$ , kde  $n$  je počet pixelů v okně. Toto okno



se dále posouvá přes všechny pixely v obraze. V závislosti na hodnotách sousedů, které byly v prvním kroku vyhodnoceny jako „černé“ a „bílé“, je poté centrální pixel vyhodnocen jako „hrana“ nebo „nehrana“. Celková tabulka báze pravidel je k nahlédnutí v [tabulce 1 přidat].



Obr. 21: Cannyho detekce hran [Autor práce]

M1	M2	M3
M4	M5	M6
M7	M8	M9

Obr. 22: Okno pro aktuální pixel  $M5$

W	W	W	B	B	B	B	B	B	W	W	W
W	W	W	B	B	B	B	W	B	W	B	W
W	W	W	B	B	B	B	B	B	W	W	W

Obr. 23: Případy, kdy je pixel vyhodnocen jako NEHRANA

### 3. Defuzzifikace

Výsledná hodnota je poté určena na základě centroidní defuzzifikace.

M1	M2	M3	M4	M5	M6	M7	M8	M9	Výsledek
B	B	B	W	W	W	W	W	W	E
B	W	W	B	W	W	B	W	W	E
W	W	W	W	W	W	B	B	B	E
W	W	B	W	W	B	W	W	B	E
B	B	W	B	W	W	W	W	W	E
W	B	B	W	W	B	W	W	W	E
W	W	W	B	W	W	B	B	W	E
W	W	W	W	W	B	W	B	B	E
B	B	B	B	W	W	W	W	W	E
B	B	B	W	W	B	W	W	W	E
B	B	W	B	W	W	B	W	W	E
B	W	W	B	W	W	B	B	W	E
W	W	W	B	W	W	B	B	B	E
W	W	W	W	W	B	B	B	B	E
W	B	B	W	W	B	W	W	B	E
W	W	B	W	W	B	W	B	B	E
B	B	B	W	W	B	W	W	B	E
W	W	B	W	W	B	B	B	B	E
B	W	W	B	W	W	B	B	B	E
B	B	B	B	W	W	B	W	W	E
B	B	B	B	W	B	W	W	W	E
W	B	B	W	W	B	W	B	B	E
W	W	W	B	W	B	B	B	B	E
B	B	W	B	W	W	B	B	W	E
B	B	B	W	W	B	W	B	B	E
W	W	B	B	W	B	B	B	B	E
B	B	W	B	W	W	B	B	B	E
B	B	B	B	W	B	B	W	W	E
B	B	W	W	W	W	W	W	W	E
W	B	B	W	W	W	W	W	W	E
W	W	B	W	W	B	W	W	W	E
W	W	W	W	W	B	W	W	B	E
W	W	W	W	W	W	W	B	B	E
W	W	W	W	W	W	B	B	W	E
W	W	W	B	W	W	B	W	W	E
B	W	W	B	W	W	W	W	W	E

M1	M2	M3	M4	M5	M6	M7	M8	M9	Výsledek
B	B	B	B	W	B	B	B	B	NE
B	B	B	B	B	B	B	B	B	NE
W	W	W	W	B	W	W	W	W	NE
W	W	W	W	W	W	W	W	W	NE

Tabulka 1: Báze pravidel

### 1.5.2. Gradientní fuzzy metoda

Gradientní fuzzy metoda je dalším způsobem detekce hran. Tato metoda, podobně jako Cannyho detekce hran zpracovává gradienty obrazu, ovšem za použití fuzzy logiky.

Metoda se skládá z těchto kroků.

#### 1. Normování

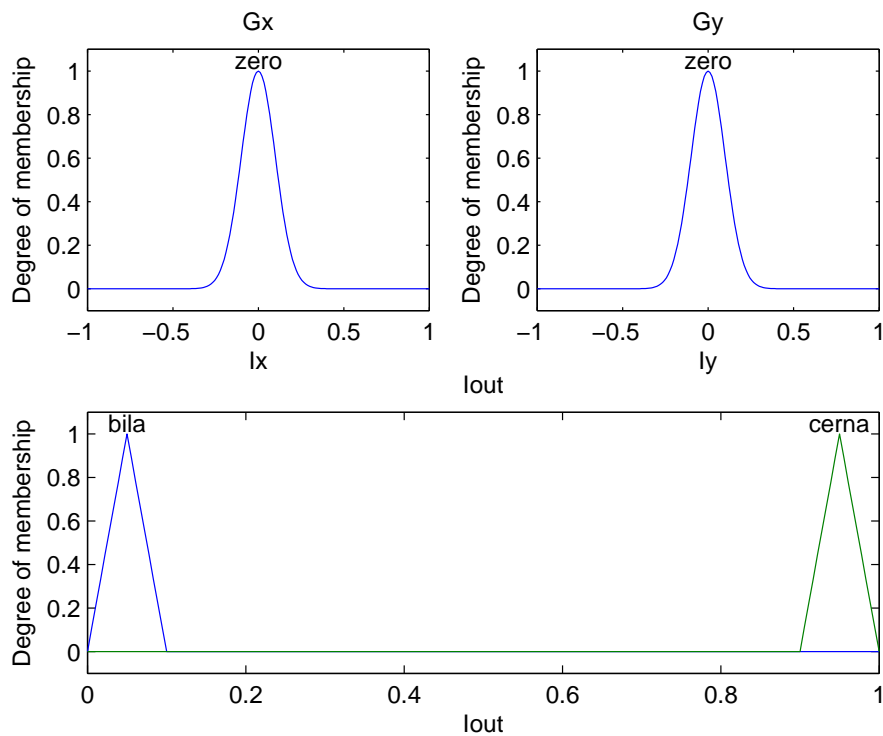
V prvním kroku normujeme hodnoty pixelů, které jsou obvykle 8-bitové a tedy v rozmetí  $\langle 0; 2^8 - 1 \rangle$  na rozmězí  $\langle 0; 1 \rangle$ .

#### 2. Gradienty

Dále vypočteme gradienty  $G_x$ ,  $G_y$  ve směrech  $x$  a  $y$  v obraze způsobem, který již byl popsán v kapitole (1.3). Díky normování v kroku 1 se hodnoty gradientů pohybují v rozmezí  $\langle -1, 1 \rangle$ .

#### 3. Fuzzifikace

Oproti prostorové fuzzy metodě tentokrát fuzzifikujeme gradienty získané v prvním kroku, pomocí definovaných fuzzy množin (viz Obr. 24). Tato metoda tedy obsahuje dva vstupy.



Obr. 24: Vstupní (nahore) a výstupní (dole) fuzzy množina [Autor práce]

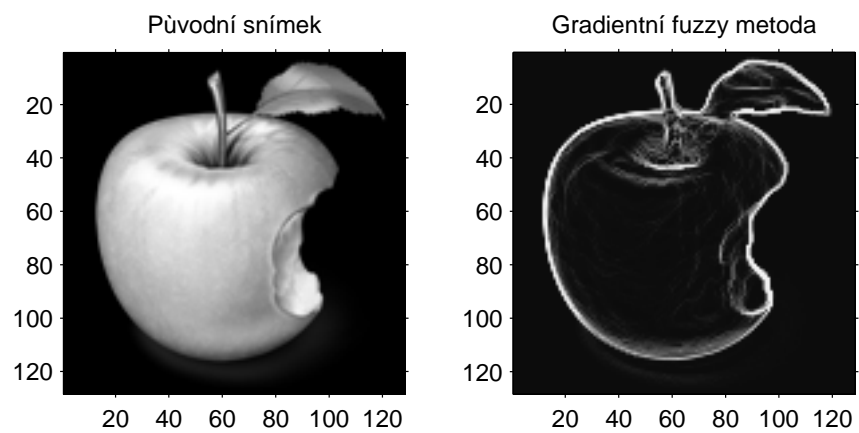
#### 4. Báze pravidel

Báze pravidel se skládá pouze ze dvou pravidel. Konkrétně:

- Pokud  $G_x$  je rovno 0 a  $G_y$  je rovno 0, pak se nejedná o hranu
- Pokud  $G_x$  není rovno 0 nebo  $G_y$  není rovno 0, pak se jedná o hranu

#### 5. Defuzzifikace

V posledním kroku znovu použijeme centroidní metodu defuzzifikace [14].



Obr. 25: Gradientní fuzzy metoda

# Kapitola 2

## Praktická část

Praktická část této diplomové práce probíhala ve spolupráci s Centrem regionu Haná pro biotechnologický a zemědělský výzkum. Jedním z výzkumných projektů tohoto centra je analýza vlivu složení hnojiv na růst konkrétních typů rostlin. Celý proces výzkumu probíhá tak, že jsou zasazeny tisíce vzorků rostlin do půd obohacených různými typy hnojiv, a následně jsou v daných intervalech fotografovány. Z těchto fotografií je zjišťováno, jak rychle rostliny rostou. Vzhledem k počtu zasazených rostlin, který se tedy pohybuje v tisících, není možné ručně měřit velikosti jednotlivých rostlin.

Hlavní náplní této diplomové práce, je vytvořit vhodný program v prostředí MATLAB, zaměřený na analýzu obrazů. Konkrétně musí být schopen:

1. Najít na daném obraze rostliny,
2. Ohraničit nalezenou rostlinu,
3. Zjistit počet pixelů, které danou rostlinu tvoří,
4. Zapsat zjištěné informace do souboru xlsx.

V dalším textu jsou popsány dva konkrétní druhy rostlin, kterými se Centrum regionu Haná zabývá. Kvůli různorodosti těchto úkolů byl pro každý z nich vytvořen speciální program.

## 2.1. Arabidopsis thaliana

V této sekci je popsán program, pro kvantifikaci zelené plochy rostliny *Arabidopsis thaliana*.

Jedná se o experiment *in vitro*,<sup>1</sup> při kterém bylo zasazeno přibližně 72 000 těchto rostlin. Na tomto vzorku byla zkoumána zejména závislost chemického sloužení půdy a růstu rostlin.

Vstupem je sada obrazů rostlin focených RGB fotoaparátem. Příklad takového snímku můžeme vidět na Obr.26. M-fily lze nalézt na přiloženém CD, pod názvem „Arabidopsis thaliana“.



Obr. 26: Arabidopsis thaliana

Tento konkrétní úkol je specifický tím, že každá fotografie obsahuje vzorek, na kterém je 24 kulatých misek, ve kterých rostou tyto rostliny. Do vstupního obrazu zasahují i další vzorky, které nejsou předmětem zkoumání aktuální fotografie. Proto je nutné v prvním kroku programu najít zkoumaný vzorek s 24 miskami.

Je zapotřebí nejprve najít čtyři strany, respektive čtyři rohy zkoumaného vzorku. Pro jednodušší nalezení bočních hran byly na tyto strany nakresleny modré pruhy. Z kapitoly o RGB modelu víme, že modrá barva má první dvě složky

---

<sup>1</sup>Prováděný v laboratorních podmínkách

malé (odpovídající R a G) a poslední velkou (odpovídající B). Proto hledáme pixely, které toto splňují. V tomto případě byl použit součin kanálu R a G, který, i pokud jeden z těchto kanálů bude větší, bude díky součinu zmenšen. Druhou podmínkou je poté již zmíněná větší hodnota modrého kanálu.

```
% Soucin R a G kanalu
soucin = double(obr(:, :, 1)) .* double(obr(:, :, 2));
soucin = soucin - min(soucin(:));
% Normovani do [0 1]
soucin = soucin ./ max(soucin(:)); % Normovani do [0 1]
% Podminka pro modre pruhy
pruhy = soucin < .15 & obr(:, :, 3) > 100;
% Podminka pro cislo nahore
pruhy_1 = soucin < .25;
```

Poté modré pruhy nalezneme jako maximální hodnoty, splňující již zmíněnou podmínku, v kódu označenou jako *pruhy*. Podobným způsobem nalezneme i pravou stranu.

Dalším krokem je nalezení horní hrany. Zde využijeme toho, že horní hrana obsahuje identifikační číslo, které je černé. Černá barva je typická nízkými hodnotami všech složek. Proto hledám takový řádek v obrázku, ve kterém je dostatek „černých“ pixelů.

Tento způsob ovšem poměrně často selhával, proto je v práci implementován další způsob nalezení horní strany vzorku. Konkrétně se jedná o to, že existují barevné rozdíly mezi pozadím obrazu a vzorkem. Tyto rozdíly jsou v programu měřeny derivacemi. Dále využiji toho, že barva uvnitř vzorku se blíží bílé, a proto jsou hodnoty pixelů vysoké.<sup>4</sup> Podobně je nalezena také spodní hrana.

V této chvíli již je nalezen vzorek, který potřebujeme zpracovávat. Dále hledáme zelenou plochu rostlin. Ke klasifikaci zelené plochy použijeme převod do barevného prostoru HSV a na základě vlastního pozorování definujeme zelenou pomocí konkrétních podmínek na jednotlivé složky pixelů.

```
H = hsv(:, :, 1); % Hue
S = hsv(:, :, 2); % Saturation
```

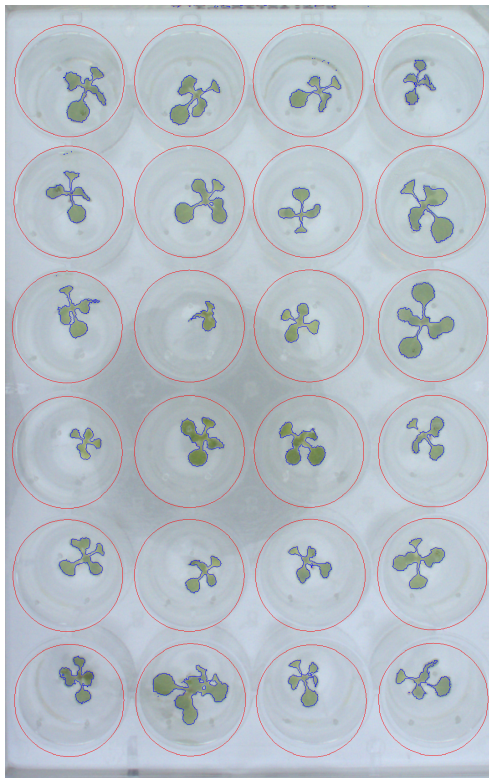
---

<sup>4</sup>Pro jednoduchost bylo dále pracováno pouze s modrou složkou obrazu.



```
V = hsv (:, :, 3); % Value  
zelena = (H<.4) & (S>.1) & (V<.75);
```

Dále je nutno spočítat počet pixelů, označených jako zelená v jednotlivých miskách. K tomuto účelu byl uměle vytvořen obraz s 24 miskami, které velikostně odpovídají miskám na fotografii.



Obr. 27: Arabidopsis thaliana

### Poznámky

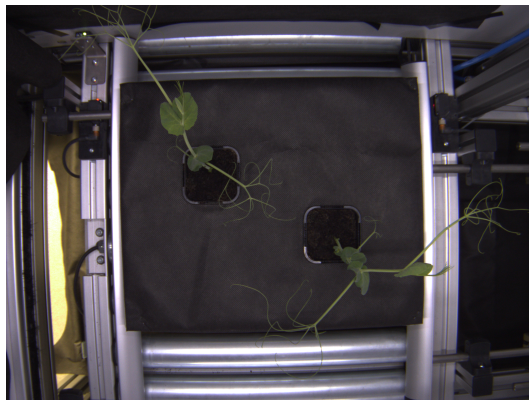
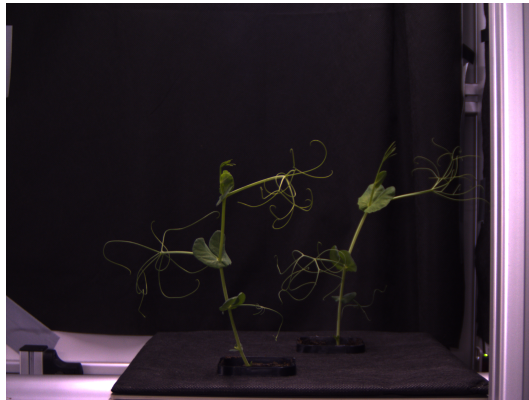
- V tomto konkrétním úkolu byla využita pouze klasifikace zelené pomocí využití podmínek v HSV modelu. Ukázalo se, že při kombinaci dalších metod popsaných v práci není tomto případě dosaženo lepších výsledků. Pravděpodobně proto, že pozadí má vždy stejnou barvu, a proto se odlesky zelené barvy neobjevují jinde.
- V této práci je popsána pouze „kostra kódu“, který byl použit.

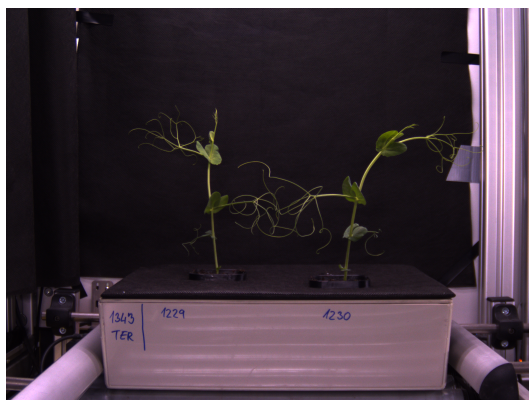
- V programu jsou dále ošetřeny chyby, které mohou nastat. Pokud by čtenáře více zajímaly, je celý program i s komentářem uložen na přiloženém CD.

## 2.2. Hrášek

Druhým zkoumaným příkladem v této práci je růst hrášku. Znovu se jedná o analýzu fotografií, kdy úkolem je spočítat počet pixelů, které ve snímku tvoří danou rostlinu - hrášek. Na každé fotografii jsou navíc dvě rostliny.

K dispozici jsou tři typy snímků: Z pravé strany, shora a zepředu. Na Obr. 28 jsou typické vstupní obrazy.





Obr. 28: Vstupní fotografie

Tento program se skládá z hlavního programu a několika pomocných funkcí. „Srdcem“ programu je část, která vyhodnocuje zelenou barvu, neboli hledá rostlinu na obrázku. To budeme dále nazývat maskou rostliny. Úkolem je tedy získat masku rostliny a potlačit všechno ostatní.

Nejprve je použit mediánový filtr tak, jak je popsán v kapitole 1.3.2. Dále je RGB obraz převeden do HSV modelu, přičemž víme, že se zelená vyznačuje malou hodnotou v první složce modelu HSV, který odpovídá barvě (Hue). Po nalezení zelené barvy se ovšem ukázalo, že ve snímku je mnoho odlesků a záhybů, které jsou vyhodnoceny jako zelené. Proto je potřeba pro přesné nalezení rostlin tuto metodu zkombinovat s některou z metod detekce hran.

Hledáme tedy hrany pomocí jedné z uživatelem vybraných metod. Konkrétně je možné si vybrat z následujících metod:

- Cannyho detekce hran,
- Gradientní fuzzy metoda,
- Bodová fuzzy metoda.

Dále je potřeba zjištěné informace zkombinovat. Program využívá následující pravidlo.

- *Zelená* je vždy označena jako maska,
- *Hrany* jsou jako maska označeny pouze v případě, že jsou v dostatečně malé vzdálenosti od zelené.

### 2.2.1. Problémy

V průběhu analyzování zelené plochy rostlin se ukázalo několik specifikací a problémů, které bylo potřeba vyřešit. V této sekci jsou tyto problémy popsány. Na přiloženém CD může čtenář najít okomentované funkce, které popsané problémy řeší.

Konkrétně se jedná o:

- Rozdělení dvou rostlin

Na každém snímku jsou dvě rostliny. Proto je nutné je najít a oddělit.

- Odlesky

Na okrajích květináčů vznikaly zelené odlesky, které byly vyhodnoceny jako součást rostliny.

- Tyčka

V některých případech byla pro stabilitu rostliny přidána podpurná tyčka, která rostlinu rozdělovala na dvě části.

- Splynutí rostlin

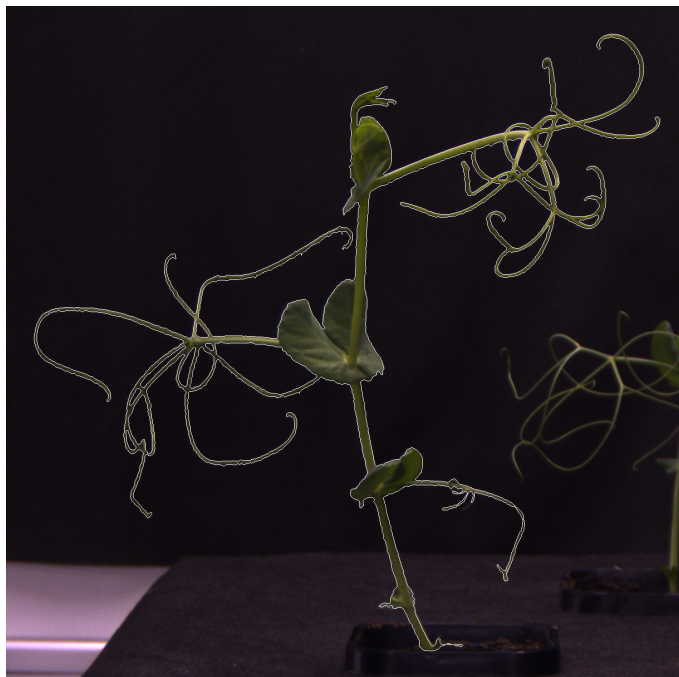
Vzhledem k nedostatečným rozestupům mezi dvěma rostlinami na snímku, v několika případech došlo k splynutí rostlin, a nebylo možno rozeznat, přední a zadní rostlinu. Byl tedy vytvořen program, kdy uživatel ručně rozdělí tyto rostliny.

### 2.2.2. Hodnocení použitých metod

Výsledek záleží na metodě, kterou se uživatel rozhodně použít. Bohužel na použitých obrázcích nelze aplikovat objektivní kritérium, které by určilo, která z vybraných metod je nejlepší. Proto je v následující části použito subjektivního hodnocení jednotlivých metod autorem práce.

## Gradientní fuzzy metoda

Gradientní fuzzy metoda dokáže spolehlivě najít hrany rostliny. Ovšem v některých případech tato metoda nenašla tenké výběžky rostlin, což s ohledem na jejich velikost není pro výsledky stěžejní. Problémem je, že díky své fuzzy povaze jsou hrany „neostré“.<sup>2</sup> Proto je nutno při aplikaci této metody na podobný druh problémů výstup gradientní fuzzy metody dále upravit. Navíc metoda využívá pouze lokální hodnocení hrany, tzn. hodnotí hrany pouze podle určitého okolí pixelu, tím pádem neposuzuje obraz jako celek. Proto se může stát, že některé, například špatně nasvícené, hrany označí jako hrany s nízkou příslušností. Vzhledem k tomu, že tento výzkum probíhá stále za stejných podmínek, se gradientní fuzzy metoda zdá být vhodným řešením problému. Při hodnocení několika tisíc snímků se může problematickým jevit to, že tato metoda je oproti Cannyho detekce hran velice časově náročná. Výpočetní čas se pohybuje v rozmezí pod 1 minutu na snímek.



Obr. 29: Gradientní fuzzy metoda

---

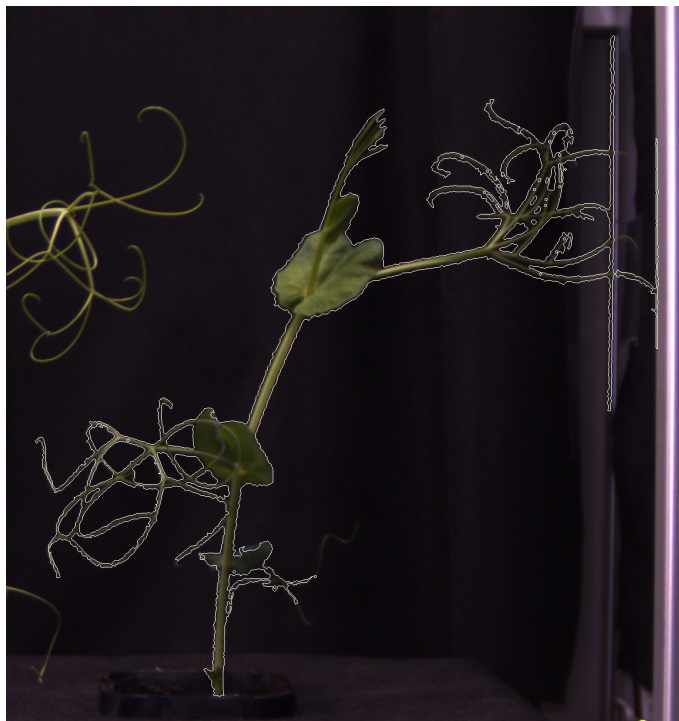
<sup>2</sup>Neostrou hranu chápeme jako hranu o šířce větší než jeden pixel.



Obr. 30: Výřez: Gradientní fuzzy metoda

### **Bodová fuzzy metoda**

Výsledky této metody jsou velmi podobné gradientní fuzzy metodě. To znamená, že výsledkem jsou „neostré“ hrany. Doba zpracování je navíc ze všech metod jednoznačně nejhorší, jedná se řádově o hodiny potřebné na analýzu jednoho snímku. Proto autor práce nepovažuje tuto metodu za vhodnou pro aplikaci ta takovýto problém.



Obr. 31: Bodová fuzzy metoda

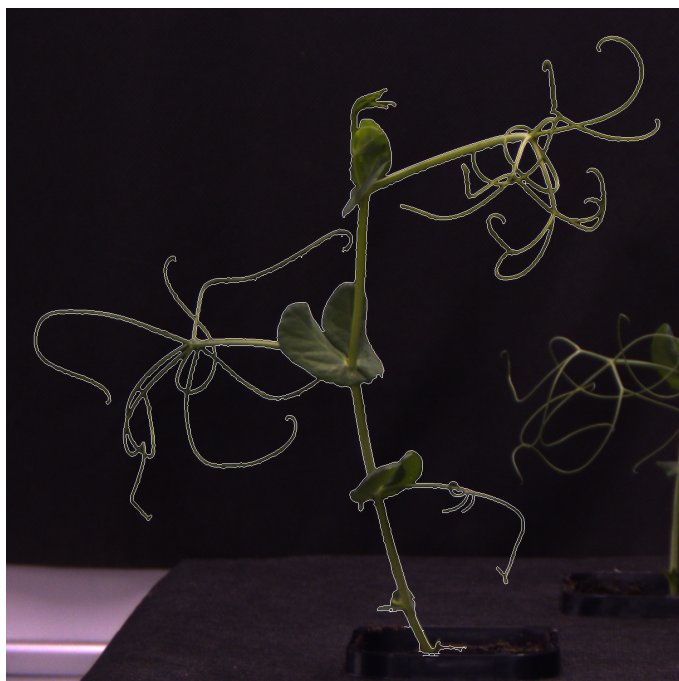


Obr. 32: Výřez: Bodová fuzzy metoda

## Cannyho detekce hran

Tato metoda používá promyšlený algoritmus, který v mnohých případech odhalí hrany, které jiné metody nezachytí. Navíc jsou výsledkem „ostré hrany“, a proto je možno Cannyho metodu použít bez dalších úprav. Problémem jsou ovšem hodnoty prahů. Pro každou aplikaci je potřeba vizuálně posoudit jaké hodnoty prahů zvolit. Znovu je nutno dodat, že tento konkrétní případ se odehrává za stálých podmínek, a proto stačí prahy posoudit pouze jednou, a poté je lze používat na hodnocení tisíců snímků. Velkou výhodou této metody je časová úspora. Doba zpracování se pohybuje řádově v desítkách sekund.

Při vizuálním hodnocení výsledků došel autor práce k závěru, že všechny tři metody jsou si výsledkově velmi podobné. Rozdíly se projevují až při přiblížení snímků, kdy metody používající fuzzy logiku nepůsobí tak hladce jako Cannyho metoda. Rozdíly mezi těmito metodami je ovšem v mnohých případech velmi těžké najít. Vzhledem k tomu je důležitá časová náročnost jednotlivých metod. Z tohoto důvodu autor práce pro podobné aplikace doporučuje používat Cannyho detekci hran.



Obr. 33: Cannyho detekce hran





Obr. 34: Výřez: Cannyho detekce hran

# Závěr

Diplomová práce probíhala ve spolupráci s Centrem regionu Haná pro biotechnologický a zemědělský výzkum. Jedním z výzkumných projektů tohoto centra je analýza vlivu složení hnojiv na růst konkrétních typů rostlin. Celý proces výzkumu probíhá tak, že jsou zasazeny tisíce vzorků rostlin do půd obohacených různými typy hnojiv, a následně jsou v daných intervalech fotografovány.

Stěžejním cílem diplomové práce bylo vytvoření matematického programu schopného určení velikosti rostlin na konkrétních snímcích poskytnutých Centrem regionu Haná.

Aby bylo stěžejního cíle dosaženo, bylo nejdříve nutno nastudovat a pochopit teorii týkající se zpracování obrazu. Z toho důvodu se první část diplomové práce zabývá teorií a konkrétními metodami potřebnými k pochopení a následnému řešení stěžejního cíle. Autor zde vychází především z odborné literatury.

V práci je rozebrána teorie barev, zejména modely RGB a HSV. Model RGB je třísloužkový model, který funguje na základě aditivního přičítání barev. Máme tedy k dispozici tři základní barvy a kombinací těchto barev se vytváří libovolná jiná barva. Model HSV je také třísloužkový. Navíc první složka vyjadřuje přímo barvu. Proto je tento model výhodný při barevném rozpoznávání objektů v obraze.

Dále se práce zabývá teorií filtrování. Pro správné fungování některých metod detekce hran je potřeba zbavit obraz šumu. Proto je filtrování důležitou součástí zpracování obrazu. Filtrování lze provádět v prostorové nebo frekvenční doméně. V případě prostorového filtrování jsou filtrační operace prováděny přímo na hodnotách jednotlivých pixelů. Prostorové filtrování je jednoduché na implementaci, ale výpočetně je velmi náročné. Druhou možností je tedy filtrování ve frekvenční

doméně. Takovéto filtrování probíhá ve třech krocích. Nejprve je obraz pomocí Fourierovy transformace převeden do frekvenční domény. V druhém kroku je Fourierovo spektrum upraveno pomocí konkrétních filtrů a poté je zpět převedeno do prostorové domény pomocí inverzní Fourierovy transformace. Výhodou frekvenčního filtrování je zejména výpočetní rychlost.

V práci jsou představeny tři metody detekce hran. První dvě metody pracují na základě fuzzy logiky. Bodová fuzzy metoda nejprve určí s jakou příslušností je daný pixel bílý a černý a poté na základě báze pravidel vyhodnocuje to, s jakou příslušností je každý pixel hranou nebo nehranou. Gradientní fuzzy metoda v prvním kroku spočte gradienty v obraze, a znovu na základě určité báze pravidel určí hrany objektů v obraze. V neposlední řadě práce popisuje Cannyho detekci hran. Tato metoda také pracuje s gradienty. Cannyho metoda používá dvojité prahování, díky čemuž jsou potlačeny nejslabší hrany, jsou získány silné hrany, a také jsou určeny slabé hrany, které jsou do výsledku použity pouze pokud navazují na hrany silné.

Praktická část této diplomové práce je tvořena několika sériemi dvou typů rostlin: *arabidopsis thaliana* a hráškem. I přesto, že jsou tyto série podobného rázu, musely být vytvořeny dva různé programy v prostředí MATLAB, které jsou schopny rozeznat rostliny na snímcích, spočítat jejich velikost a zapsat tyto informace do souboru. Tím byl stěžejní cíl diplomové práce naplněn.

První sérii snímků (rostlina *arabidopsis thaliana*), byla specifická tím, že každý snímek obsahoval vzorek, na kterém je umístěno 24 kulatých misek s touto rostlinou. Navíc do snímku zasahovaly i další vzorky. Proto v první řadě program hledá čtyři strany aktuálního vzorku. V dalším kroku se pro nalezení masky rostlin ukázalo dostačující využití barevného modelu HSV a rozpoznání zelené barvy, a to nejspíše díky svému typu a „sterilním“ podmínkám.

Druhá série (hrášek), je tvořena třemi typy fotografií: zepředu, zprava a shora. Navíc jsou na každém snímku dvě rostliny. Tento program využívá pro nalezení masky rostlin kombinaci rozpoznání barev pomocí podmínek, spolu s kombinací s jednou z výše zmíněných metod detekce hran. Program dává uživateli možnost

zvolit z těchto tří metod. Kvůli své časové náročnosti v řádu hodin na jednu fotografii není bodová fuzzy metoda použitelná na větší počet snímků, a proto ji autor práce nedoporučuje pro tuto sérii používat. Gradientní fuzzy metodu autor práce hodnotí jako vhodnou pro danou sérii fotografií, zejména z důvodů poměrně vysoké přesnosti nalezení rostlin v obraze. Tato metoda však v některých případech nedokáže rozeznat malé výběžky rostliny. Další nevýhodou je větší výpočetní doba, která se pohybuje kolem 40 sekund. Cannyho detekce hran poskytuje velmi kvalitních výsledků rozpoznání rostlin. Navíc i díky výpočetnímu času kolem 15 sekund se jeví jako nejvhodnější metoda.

Doufám, že zjištěné závěry pomohou jak Centru regionu Haná v dalších výzkumech, tak i čtenářům této práce, kteří se zabývají podobnou problematikou.

# CD příloha

Na přiloženém CD najdete tři složky.

- Arabidopsis thaliana

Obsahuje program v prostředí MATLAB, pro analýzu zelené plochy pro první sérii fotografií. Součástí složky je soubor README.txt popisující fungování programu.

- Hrášek

Obsahuje program v prostředí MATLAB, na analýzu zelené plochy pro druhou sérii fotografií. Součástí složky je soubor README.txt popisující fungování programu.

- Diplomová práce

Obsahuje tuto diplomovou práci ve formátu PDF.

# Seznam obrázků

- Obr. 1: Různá barevná hloubka
- Obr. 2: Světelné spektrum
- Obr. 3: Řez lidským okem
- Obr. 4: Aditivní a substraktivní model
- Obr. 5: Rozklad modelu RGB
- Obr. 6: HSV model
- Obr. 7: Rozklad modelu HSV
- Obr. 8: Převod RGB do HSV
- Obr. 9: Vyrovnání histogramu
- Obr. 10: Vyrovnání histogramu
- Obr. 11: Fourierovo spektrum (vpravo) konkrétního snímku (vlevo)
- Obr. 12: Ideálního lowpass filtr
- Obr. 13: Gaussovský lowpass filtr
- Obr. 14: Krok 1: Vyhlazení
- Obr. 15: Krok 2: Hledání gradientů
- Obr. 16: Lokální maxima gradientů
- Obr. 17: Dvojitá prahování
- Obr. 18: Sledování hran
- Obr. 19: Cannyho detekce hran
- Obr. 20: Vstupní (nahore) a výstupní (dole) fuzzy množina
- Obr. 21: Okno pro aktuální pixel  $M5$
- Obr. 22: Cannyho detekce hran
- Obr. 23: Případy, kdy je pixel vyhodnocen jako NEHRANA

- Obr. 24: Vstupní (nahore) a výstupní (dole) fuzzy množina
- Obr. 25: Gradientní fuzzy metoda
- Obr. 26: Arabidopsis thaliana
- Obr. 27: Arabidopsis thaliana
- Obr. 28: Vstupní fotografie
- Obr. 29: Gradientní fuzzy metoda
- Obr. 30: Výřez: Gradientní fuzzy metoda
- Obr. 31: Bodová fuzzy metoda
- Obr. 32: Výřez: Bodová fuzzy metoda
- Obr. 33: Cannyho detekce hran
- Obr. 34: Výřez: Cannyho detekce hran

# Literatura

- [1] GONZALEZ, Rafael C a Richard E WOODS. Digital image processing. 3rd ed. Upper Saddle River: Pearson, c2008, xxii, 954 s. ISBN 0-13-168728-x
- [2] PRATT, William K. Digital image processing: PIKS inside. 3rd ed. New York: Wiley, c2001, xix, 735 p. ISBN 04-713-7407-5
- [3] FRASER, Bruce, Chris MURPHY a Fred BUNTING. Real world color management: industrial-strength production techniques. 2nd ed. Berkeley, CA: Peachpit Press, c2005, 582 p. ISBN 0201773406.
- [4] [online]. [cit. 2015-01-03]. Dostupné z: [http://www.visusoptik.cz/?page\\_id=49](http://www.visusoptik.cz/?page_id=49)
- [5] [online]. [cit. 2015-01-07]. Dostupné z: <https://www.pinterest.com/mann0485/electromagnetic-spectrum/>
- [6] [online]. [cit. 2015-01-07]. Dostupné z: <http://www.cvrl.org/gallery/Additive-subtractive-mixing.htm>
- [7] Wikimedia Commons, 2010
- [8] [online]. [cit. 2015-04-07]. Dostupné z: <http://lidsketelo.webnode.cz/regulacni-soustavy/>
- [9] JOBLOVE, George H. a Donald GREENBERG. Color spaces for computer graphics. ACM SIGGRAPH Computer Graphics [online]. 1978, vol. 12, issue 3, s. 20-25, DOI: 10.1145/965139.807362.
- [10] ZENG, Jun a Dehua LI. Color image edge detection method using VTV denoising and color difference. Optik - International Journal for Light and Electron Optics [online]. 2012, roč. 123, č. 22, s. 2072-2075 [cit. 2015-04-08]. DOI: 10.1016/j.ijleo.2011.10.009.
- [11] Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679-698, 1986.



- [12] Moeslund, T., Canny Edge Detection. March 3, 2009, Retrieved December 3, 2014
- [13] Mamdani, E. H., Application of fuzzy algorithms for the control of a simple dynamic plant. In Proc IEEE (1974), 121-159.
- [14] KAVITHA, C. a S. Denis ASHOK. Edge Detection of Images Using Fuzzy Logic Technique. International Journal of Applied Engineering Research. 2013, roč. 8, č. 19, s. 2475-2478.
- [15] WALACH, Jan. Aplikace Fourierovy analýzy na rozpoznávání kvality samohlásek podle jejich formantů. Olomouc, 2013. Bakalářská práce. Univerzita Palackého v Olomouci. Vedoucí práce Tomáš Fürst.
- [16] Benson, D.: Music: A Mathematical Offering, Department of Mathematical Sciences, University of Aberdeen, Scotland, UK, 2006