

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

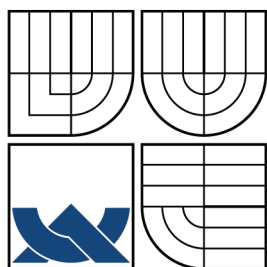
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

SPOJOVÁNÍ NEPŘEKRÝVAJÍCÍCH SE OBRAZŮ

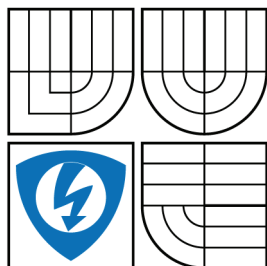
DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. LUKÁŠ BÁRNET



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY  
A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY



FACULTY OF ELECTRICAL ENGINEERING AND  
COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

## SPOJOVÁNÍ NEPŘEKRÝVAJÍCÍCH SE OBRAZŮ COUPLING OF IMAGES

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

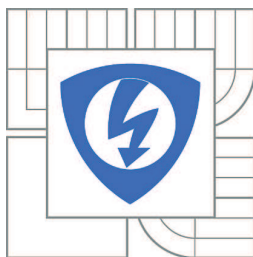
AUTOR PRÁCE  
AUTHOR

Bc. LUKÁŠ BÁRNET

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. MILOSLAV RICHTER, Ph.D.

BRNO 2012



**VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky  
a komunikačních technologií**

**Ústav automatizace a měřicí techniky**

# Diplomová práce

magisterský navazující studijní obor  
**Kybernetika, automatizace a měření**

**Student:** Bc. Lukáš Bárnét

**ID:** 106364

**Ročník:** 2

**Akademický rok:** 2011/2012

**NÁZEV TÉMATU:**

## Spojování nepřekrývajících se obrazů

### POKYNY PRO VYPRACOVÁNÍ:

Navrhněte systém pro automatizované řešení úloh typu "puzzle". Vytvořte předpis na pořizení dat, a na jeho základě vytvořte databáze pro testování algoritmů. Zhodnoťte možnosti řešení pomocí barevných a tvarových informací v závislosti na typu dílků. Popište získání dat pro jeden dílek a proces vyhledávání sousedících dílků. Stanovte možnost metod na rozdělení do skupin podle barevné "podobnosti", a zhodnoťte možnost jejich využití pro zrychlení prohledávacích algoritmů skládání. Vhodné metody implementujte do funkčního programu. Výsledky zpracování zhodnoťte pro databázové snímky.

### DOPORUČENÁ LITERATURA:

SONKA M., HLAVAC V., BOYLE R.: Image Processing, Analysis and Machine Vision. 3rd edition. Toronto : Thomson, 2008. 829 s. ISBN 978-0-495-08252-1.

Žára J., Beneš B., Sochor J., Felkel P.: Moderní počítačová grafika, Computer Press, 1998, ISBN 80-251-0454-0

KRAUS K.: Photogrammetrie 1 und 2, Ummeler / Bonn, 1996, ISBN: 3110177080

MAŘÍK V.: Umělá inteligence , Praha: Academia, 2001, ISBN: 80-200-0472-6

Faugeras O.: Three-Dimensional Computer Vision, The MIT Press 1993

**Termín zadání:** 6.2.2012

**Termín odevzdání:** 21.5.2012

**Vedoucí práce:** Ing. Miloslav Richter, Ph.D.

**doc. Ing. Václav Jirsík, CSc.**

*Předseda oborové rady*

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Diplomová práce se zabývá spojováním nepřekrývajících se obrazů. V první části jsou rozebrána teoretická východiska potřebná k úspěšnému splnění zadání. Část druhá pojednává o postupech, které vedou ke složení puzzle. Poslední část práce se zabývá ovládáním programu. Cílem předložené práce je navrhnout algoritmus pro řešení úloh typu puzzle.

## **KLÍČOVÁ SLOVA**

Puzzle, algoritmus, programování, spojování obrazů, C++

## **ABSTRACT**

The Diploma thesis is concerned with coupling of images. In the first part theoretical bases necessary for successful fulfilling of the assignment are described. The second part deals with the procedures that lead to composition of the jigsaw puzzle. The last part concentrates on controlling the program. The aim of the thesis is to propound an algorithm for solving a puzzle.

## **KEYWORDS**

Jigsaw puzzle, algorithm, programming, coupling of images, C++

BÁRNET, Lukáš *Spojování nepřekrývajících se obrazů*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2012. 97 s. Vedoucí práce byl Ing. Miloslav Richter, Ph.D.

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Spojování nepřekrývajících se obrazů“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Miloslavu Richterovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

(podpis autora)

# OBSAH

<b>1</b>	<b>Teoretický úvod</b>	<b>13</b>
1.1	Některé základní pojmy počítačového vidění . . . . .	13
1.1.1	Řetězec zpracování obrazu . . . . .	13
1.1.2	Základní barevné systémy . . . . .	16
1.2	Morfologické transformace . . . . .	20
1.2.1	Dilatace . . . . .	20
1.2.2	Eroze . . . . .	21
1.2.3	Otevření . . . . .	22
1.2.4	Uzavření . . . . .	22
1.3	Algoritmy použité v diplomové práci . . . . .	22
1.3.1	Cannyho hranový detektor . . . . .	22
1.3.2	Algoritmus K-means . . . . .	23
1.3.3	Odstranění zkreslení v obrazu . . . . .	24
1.4	Možné typy puzzle . . . . .	28
1.5	Výpočetní náročnost skládání puzzle . . . . .	29
1.6	Knihovny OpenCV . . . . .	30
1.7	Shrnutí . . . . .	31
<b>2</b>	<b>Řešení studentské práce</b>	<b>32</b>
2.1	Volba scény . . . . .	33
2.2	Databáze vstupních snímků . . . . .	36
2.3	Předzpracování obrazu a vyříznutí dílku . . . . .	37
2.3.1	Předzpracování obrazu . . . . .	37
2.3.2	Vyříznutí dílku . . . . .	40
2.3.3	Otočení dílku . . . . .	42
2.4	Popis dílku pomocí barev . . . . .	43
2.4.1	Popis barvy na hranách . . . . .	43
2.4.2	Popis barvy na dílku . . . . .	45
2.5	Popis dílku pomocí tvaru . . . . .	46
2.5.1	Popis tvaru pomocí normálových vektorů . . . . .	46
2.5.2	Popis zámků puzzle . . . . .	48
2.6	Složení puzzle . . . . .	49
2.6.1	Hledání nejlépe hodících se dílků . . . . .	49
2.6.2	Kontrola platnosti vytvořených párů . . . . .	59
2.6.3	Otočení dílků . . . . .	59
2.6.4	Složení puzzle . . . . .	60
2.6.5	Složení puzzle popsaného podle barvy . . . . .	62

2.7	Shrnutí . . . . .	64
<b>3</b>	<b>Výsledky studentské práce</b>	<b>65</b>
3.1	Základní požadavky . . . . .	65
3.2	Běh a ovládání programu . . . . .	65
<b>4</b>	<b>Závěr</b>	<b>69</b>
	<b>Literatura</b>	<b>70</b>
	<b>Seznam příloh</b>	<b>72</b>
<b>A</b>	<b>Vývojové diagramy</b>	<b>73</b>
<b>B</b>	<b>Databáze obrázků</b>	<b>77</b>
B.1	Funkční obrázky popisované pomocí barvy i tvaru . . . . .	77
B.2	Funkční obrázky popisované pouze podle barvy . . . . .	87
B.3	Nefunkční obrázky . . . . .	93



# SEZNAM OBRÁZKŮ

1.1	Řetězec zpracování obrazu. . . . .	13
1.2	Chromatický diagram CIE 1931 . . . . .	16
1.3	CRT gamut. . . . .	17
1.4	Barevný model RGB. . . . .	17
1.5	Barevný systém HSV v kuželovém zobrazení. . . . .	18
1.6	CbCr rovina při konstantním jasu $Y = 0,5$ . . . . .	19
1.7	Skládání barev pomocí složek CMY systému CMYK. . . . .	19
1.8	Původní obrázek a obrázek po dilataci. . . . .	20
1.9	Původní obrázek a obrázek po erozi. . . . .	21
1.10	Postup algoritmu K-means. . . . .	24
1.11	Poduškovité a soudkovité zkreslení v obrazu. . . . .	27
2.1	Scéna pro pořizování obrazů dílků ze Semestrálního projektu 1. . . . .	33
2.2	Scéna pro pořizování obrazů dílků. . . . .	34
2.3	Scéna pro pořizování obrazů dílků s kalibračními značkami. . . . .	35
2.4	Dílek popisovaný podle tvaru i podle barvy. . . . .	36
2.5	Dílek popisovaný pouze podle barvy. . . . .	36
2.6	Fotografie šachovnice použité pro určení zkreslení objektivu. . . . .	37
2.7	Vzory pro hledání kalibračních značek. . . . .	38
2.8	Vyznačené nalezené kalibrační značky. . . . .	38
2.9	Scéna focená z úhlu. . . . .	39
2.10	Scéna s odstraněnou perspektivou a vyříznutým kalibračním čtvercem. . . . .	39
2.11	Barevně označené clustery – označení pozadí. . . . .	40
2.12	Vyznačené dílky. . . . .	41
2.13	Původní obrázek, vyznačení vyříznutého dílku a vyříznutý dílek. . . . .	41
2.14	Dílky otočené do vodorovné polohy. . . . .	42
2.15	Průběh barevných složek na vrchním okraji dílku č. 1. . . . .	44
2.16	Dílky s odstraněnými zámky. . . . .	46
2.17	Dílky s vyznačenými rohy a těžištěm. . . . .	47
2.18	Dílky s vyznačenými vektory. . . . .	47
2.19	Průběh barevných složek popisující vrchní stranu dílku č. 1 pomocí barev. . . . .	51
2.20	Průběh barevných složek popisující spodní stranu dílku č. 9 pomocí barev. . . . .	52
2.21	Spojené dílky 1 a 9. . . . .	53
2.22	Průběh rozdílů jednotlivých barevných složek při spojení dílků 1 a 9. . . . .	53
2.23	Graf kumulačního průběhu absolutních hodnot rozdílů jednotlivých barevných složek při spojení dílků 1 a 9. . . . .	54

2.24	Graf kumulačního průběhu absolutní hodnoty součtu rozdílů absolutních hodnot jednotlivých barevných složek při spojení dílků 1 a 9. . . . .	54
2.25	Průběh barevných složek popisující vrchní stranu dílku č. 1 pomocí barev. . . . .	55
2.26	Průběh barevných složek popisující dolní stranu dílku č. 18 pomocí barev. . . . .	56
2.27	Spojené dílky 1 a 18. . . . .	57
2.28	Průběh rozdílů jednotlivých barevných složek při spojení dílků 1 a 18. . . . .	57
2.29	Graf kumulačního průběhu absolutních hodnot rozdílů jednotlivých barevných složek při spojení dílků 1 a 18. . . . .	58
2.30	Graf kumulačního průběhu absolutní hodnoty součtu rozdílů absolutních hodnot jednotlivých barevných složek při spojení dílků 1 a 18. . . . .	58
2.31	Ilustrační obrázek pro kontrolu umístění dílku. . . . .	60
2.32	Složené puzzle. . . . .	61
2.33	Původní fotografie, která byla rozřezána a následně složena. . . . .	62
2.34	Rozdělená fotografie určená ke složení. . . . .	63
2.35	Fotografie složená z dílků. . . . .	63
3.1	Obsah vstupního souboru obsahující cestu ke všem fotografiím dílků. . . . .	65
3.2	Ovládání programu z konzole. . . . .	66
3.3	Výpis informací do konzole. . . . .	66
3.4	Rozložené puzzle. . . . .	67
3.5	Složené puzzle. . . . .	67
3.6	Rozložené puzzle popsané podle barev. . . . .	68
3.7	Složené puzzle pomocí popisu barev. . . . .	68
A.1	Vývojový diagram funkce main. . . . .	73
A.2	Vývojový diagram předzpracování obrázku. . . . .	74
A.3	Vývojový diagram popisu dílku podle barvy na hranách. . . . .	75
A.4	Vývojový diagram popisu dílku podle normálových vektorů. . . . .	76
A.5	Vývojový diagram funkce přidávající shodu do seznamu nejlépe shodujících se dílků. . . . .	76
A.6	Vývojový diagram funkce kontrolující, zda je dílek na správném místě. . . . .	76
B.1	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 1). . . . .	77
B.2	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 2). . . . .	77
B.3	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 3). . . . .	78
B.4	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 4). . . . .	78
B.5	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 5). . . . .	79
B.6	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 6). . . . .	79

B.7	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 7).	80
B.8	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 8).	80
B.9	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 9).	81
B.10	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 10).	81
B.11	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 11).	82
B.12	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 12).	82
B.13	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 13).	83
B.14	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 14).	83
B.15	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 15).	84
B.16	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 16).	84
B.17	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 17).	85
B.18	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 18).	85
B.19	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 19).	86
B.20	Databáze obrázků popisovaných podle barvy i tvaru (obrázek č. 20).	86
B.21	Databáze obrázků popisovaných pouze podle barvy (obrázek č. 1).	87
B.22	Databáze obrázků popisovaných pouze podle barvy (obrázek č. 2).	87
B.23	Databáze obrázků popisovaných pouze podle barvy (obrázek č. 3).	88
B.24	Databáze obrázků popisovaných pouze podle barvy (obrázek č. 4).	88
B.25	Databáze obrázků popisovaných pouze podle barvy (obrázek č. 5).	89
B.26	Databáze obrázků popisovaných pouze podle barvy (obrázek č. 6).	89
B.27	Databáze obrázků popisovaných pouze podle barvy (obrázek č. 7).	90
B.28	Databáze obrázků popisovaných pouze podle barvy (obrázek č. 8).	90
B.29	Databáze obrázků popisovaných pouze podle barvy (obrázek č. 9).	91
B.30	Databáze obrázků popisovaných pouze podle barvy (obrázek č. 10).	91
B.31	Databáze obrázků popisovaných pouze podle barvy (obrázek č. 11).	92
B.32	Databáze obrázků popisovaných pouze podle barvy (obrázek č. 12).	92
B.33	Databáze obrázků – nefunkční obrázky obsahující podkladovou barvu (obrázek č. 1).	93
B.34	Databáze obrázků – nefunkční obrázky obsahující podkladovou barvu (obrázek č. 2).	93
B.35	Databáze obrázků – nefunkční obrázky obsahující podkladovou barvu (obrázek č. 3).	94
B.36	Databáze obrázků – nefunkční obrázky obsahující podkladovou barvu (obrázek č. 4).	94
B.37	Databáze obrázků – nefunkční obrázky obsahující podkladovou barvu (obrázek č. 5).	95
B.38	Databáze obrázků – nefunkční obrázky obsahující podkladovou barvu (obrázek č. 6).	95

B.39 Databáze obrázků – nefunkční obrázky obsahující podkladovou barvu (obrázek č. 7). . . . .	96
B.40 Databáze obrázků – nefunkční obrázky obsahující podkladovou barvu (obrázek č. 8). . . . .	96
B.41 Databáze obrázků – nefunkční obrázky obsahující podkladovou barvu (obrázek č. 9). . . . .	97

# ÚVOD

Každé malé dítě si mnohokrát složilo nějakou skládačku. Mezi nejznámější dětské hry patří puzzle, které jednak slouží k zábavě, jednak napomáhá zlepšovat logické myšlení. Vzhledem k pokroku moderní doby je dnes možné takové puzzle složit pomocí počítače, který tuto činnost zvládne za podstatně kratší čas než člověk.

Program pro sestavení puzzle je možné napsat hlavně díky oboru počítačového vidění, jenž napomáhá počítači vnímat obrazové vjemy podobným způsobem jako lidské oko, a knihovnám, které velice zjednodušují práci s obrazem.

Danou diplomovou práci jsem si vybral z toho důvodu, že mě velice zaujal obor počítačového vidění a úlohu softwarového skládání puzzle jsem shledal velmi zajímavou.

Cílem této práce je navrhnout algoritmus pro skládání puzzle. Ten by měl na jedné či několika fotografiích najít dílky patřící k puzzle, tyto dílky popsat a určit, jaké dílky patří k sobě, případně zobrazit složené puzzle.

Předložená práce navazuje na Semestrální projekty 1 a 2. První semestrální projekt se zabýval předzpracováním obrazu a nalezením dílků v obrazu.

V druhé práci byl přepracován postup pro nalezení dílku v obrazu, jelikož postup z prvního projektu se neukázal jako dostatečně univerzální. Byly zde také představeny algoritmy pro popis barev a tvaru dílků.

Diplomová práce, která na zmíněné semestrální projekty navazuje si klade za cíl navrhnout zbývající algoritmy pro poskládání puzzle a odladit algoritmy použité v předešlých pracích. Práce začíná teoretickým úvodem, ve kterém jsou vysvětleny důležité postupy použité v práci. Dále je uveden stručný “rádce” pro pořizování fotografií dílků. Následuje vysvětlení postupů pro popis a složení dílků.

Celá práce je programována v jazyku C/C++ za pomoci knihoven OpenCV<sup>1</sup> ve verzi 2.1.

---

<sup>1</sup>Knihovny OpenCV budou probírány v jedné z následujících kapitol.

# 1 TEORETICKÝ ÚVOD

V této části diplomové práce budou popsány některé základní pojmy počítačového vidění a také princip algoritmů používaných v této práci.

V první části budou zmíněny některé základní pojmy počítačového vidění – řetězec zpracování obrazů a barevné systémy.

Druhá část se bude zabývat morfologickými operacemi s obrazem – dilatací a erozí.

Na závěr budou zmíněny použité funkce a postupy v práci – algoritmus K-means, změna perspektivy pomocí matice homogenní transformace atd.

## 1.1 Některé základní pojmy počítačového vidění

Počítačové vidění je obor snažící se o to, aby počítače byly schopny vnímat obrazové vjemy způsobem, jakým je vnímá člověk. Úkolem tohoto oboru je tedy rozpoznávat důležité objekty v obraze a umět je pojmenovat.

### 1.1.1 Řetězec zpracování obrazu

V této kapitole je rozebrán základní řetězec pro zpracování obrazu. Také zde jsou stručně popsány jednotlivé bloky.



Obr. 1.1: Řetězec zpracování obrazu.

Tento řetězec se dá rozdělit do dvou skupin – **nižší úroveň**, do které patří předzpracování a segmentace obrazu, a **vyšší úroveň**, do které patří popis objektů a porozumění obsahu obrazu.

Nižší úroveň zpracování bývá, co se týče metod i výkonové náročnosti, většinou méně náročná.

Pro popis objektů a pro porozumění obsahu (vyšší úroveň) zpravidla potřebujeme sofistikovanější algoritmy, a díky tomu i vyšší výpočetní výkon.

#### Pořízení obrazu

Pořízení obrazu je prvním krokem řetězce pro zpracování obrazu. Obraz pořizujeme fotoaparátem, pokud jde o statický obraz, nebo kamerou, jedná-li se o obraz

dynamický. Princip obou záznamů je ale stejný. Světlo dopadá přes čočku na čip (většinou CCD), který transformuje energii fotonů na elektrickou energii.

Abychom mohli snímek (nebo videosekvenci) uložit a dále zpracovávat, musíme ho rozdělit na matici  $M \times N$ . Každý prvek v této matici se nazývá **pixel**. Tento proces se nazývá **vzorkování**.

Pokud máme snímek navzorkovaný, je nutné každé jasové úrovni (uvažujeme šedotónový obraz) přiřadit hodnotu, kterou bude reprezentována. Jelikož v počítači používáme dvojkovou soustavu, volí se počet úrovní jako mocnina dvou. Nejčastěji se používá 256 jasových úrovní ( $2^8 = 256$ , to znamená, že každý pixel potřebuje v paměti 8 b = 1 B). Tomuto postupu se říká **kvantování**.

Výše popsany postup byl pro **digitalizaci** obrazu šedotónových snímků. Pro snímky barevné potřebujeme nejčastěji 3 CCD snímače, každý pixel se tedy skládá ze 3 složek (většinou složky RGB – Red, Green, Blue). Každou složku musíme navzorkovat a nakvantovat. Barevný obrázek tudíž potřebuje (přibližně) třikrát více paměti.

## Předzpracování obrazu

Předzpracování obrazu je druhým článkem řetězce, jenž znázorňuje zpracování obrazu. Jedná se o metody, které dokáží potlačit (nebo zvýraznit) šum (a hrany) v obrazu. Pro předzpracování používáme většinou konvoluci s maskou. Velikost masky bývá typicky 3x3, 4x4 atd. Vyskytují se ale speciální případy, kdy volíme atypický rozměr masky.

Pro potlačení šumu v obraze používáme nejčastěji jeden z těchto filtrů: filtr průměrováním, Gaussův filtr, mediánový filtr a filtr rotující maskou. Pro zvýraznění hran (a šumu) používáme tyto operátory: Robertsův, Prewittové, Sobelův, Laplaceův atd. Rozdílem Laplaceova operátoru je, že na rozdíl od ostatních funguje Laplaceův operátor na principu diferencí druhého řádu [14].

Další metodou pro hledání hran je Cannyho hranový detektor [3]. Ten nepracuje pouze s diferencemi, ale nejprve se (většinou) použije Sobelův operátor, ze kterého dostáváme údaje o velikostech hran. Na tyto velikosti použijeme dva prahy – pokud leží velikost hrany nad vyšším prahem, je automaticky prohlášena za silnou hranu. Hran, jejichž velikost je menší než spodní práh jsou automaticky “zahozeny”. Hran, jejichž velikost leží mezi prahy jsou prohlášeny za hrany pouze v případě, že vedle nich již leží silná hrana.

Do sekce předzpracování obrazu patří také morfologické transformace – například dilatace, eroze, otevření, uzavření atd. [13].

## **Segmentace obrazu**

Třetí částí ve zpracovávacím řetězci je segmentace. Úkolem této části je zvýraznit důležité objekty v obraze a nedůležité objekty v obraze potlačit.

Nejjednodušší metodou pro segmentaci je segmentace prahováním, kdy u šedotónového obrázku určíme práh. Hodnoty jasu, které leží pod tímto prahem jsou černé, hodnoty nad prahem jsou bílé. Metod pro získání prahu je celá řada. Nejjednodušší je procentní prahování (práh má hodnotu například 70 % maxima) atd.

Při segmentaci nemusíme pracovat pouze s šedotónovým obrazem. Můžeme hledat například konkrétní barvy atd.

## **Popis objektů**

Čtvrtou částí ve zpracování obrazu je popis objektů.

Pro popis objektů můžeme použít jakoukoli významnou vlastnost objektu, např. barvu, jeho obvod, obsah, poměr stran atd. Tyto informace následně využíváme při porozumění obsahu obrazu.

## **Porozumění obsahu**

Posledním krokem zpracování obrazu je porozumění jeho obsahu. V nejjednodušších případech můžeme pouze třídit objekty na kulaté a hranaté, v jiných případech můžeme rozpoznávat znaky atd.



### 1.1.2 Základní barevné systémy

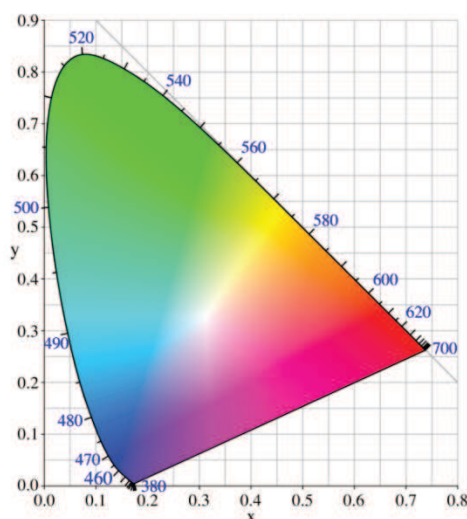
Každý člověk, který netrpí zrakovou vadou, rozeznává cca 10 milionů odstínů barev [7]. Abychom tyto odstíny byli schopni vyjádřit, je třeba použít barevný systém. Barevné systémy můžeme rozdělit na aditivní (součtové) a subtraktivní (rozdílové).

Aditivní systém sčítá jednotlivé barevné složky. Výsledkem součtu maximálních intenzit dílčích barev je bílá barva.

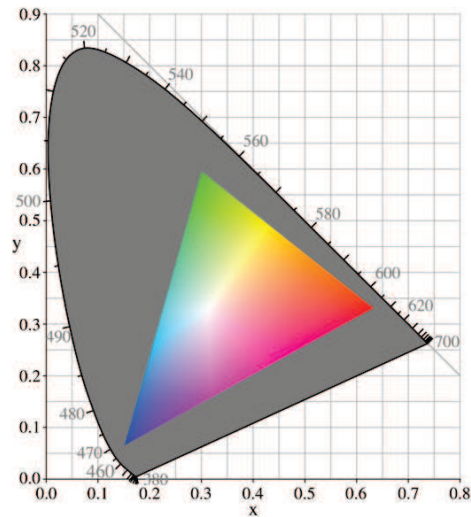
Subtraktivní systém odčítá jednotlivé složky a výsledkem rozdílu maximálních intenzit dílčích barev je černá barva.

#### Gamut

Gamut je oblast, kterou může barevný systém (či zobrazovací nebo nahrávací zařízení) dosáhnout v daném barevném prostoru. Nejčastěji používaným barevným prostorem je prostor CIE 1931.



Obr. 1.2: Chromatický diagram CIE 1931 [5].



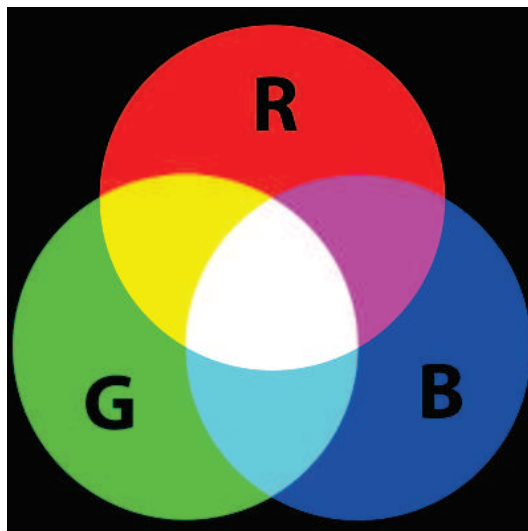
Obr. 1.3: CRT gamut [8].

## RGB

Barevný systém RGB je jedním z nejpoužívanějších barevných systémů. Systém je postaven na skutečnosti, že každou barvu je možné vyjádřit pomocí kombinace složek RGB – červená (Red), zelená (Green), modrá (Blue). K těmto složkám je možné ještě přidat složku  $\alpha$ , která vyjadřuje hodnotu sytosti.

Každá barva je většinou reprezentována 8 bity. Jeden pixel (obrazový bod) je tedy reprezentován 24 nebo 32 bity (3 nebo 4 byty).

Barevný model RGB se řadí mezi aditivní systémy.

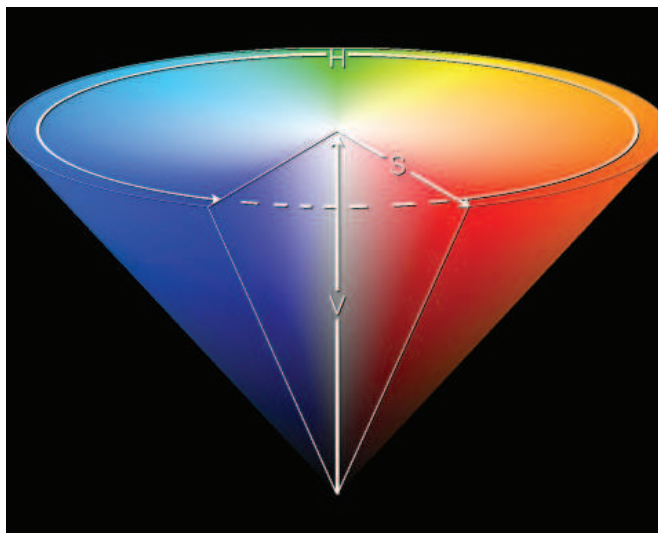


Obr. 1.4: Barevný model RGB [15].

## HSV

Systém HSV používá jako barvonosnou pouze složku H (Hue). Další 2 složky – S (Saturation) a V (Value) určují sytost a jas barvy.

Složka H udává úhel na kruhu, na kterém jsou poskládány barvy ( $0 - 360^\circ$ ), hodnota S udává sytost barvy ( $0 - 100\%$ ) a hodnota V udává množství jasu.

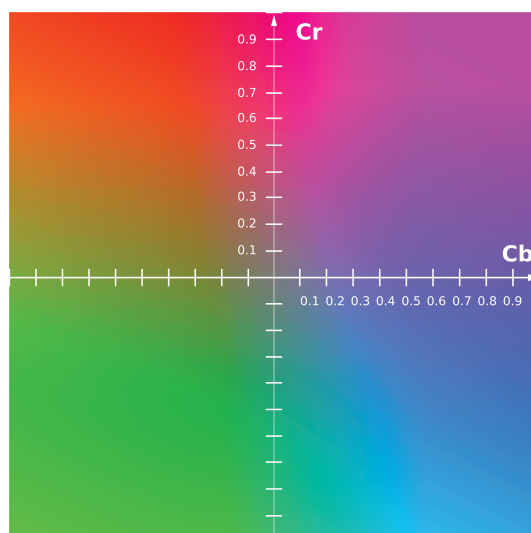


Obr. 1.5: Barevný systém HSV v kuželovém zobrazení [10].

## YCbCr

YCbCr je systém používaný především v kombinaci s videem. Jedná se o kódovaný zápis RGB systému, kde Y představuje jas a složky Cb a Cr představují modré a červené chrominanční komponenty.

Systém YCbCr byl zaveden hlavně proto, že v systému RGB je velké množství redundance. YCbCr využívá toho, že lidské oko je více citlivé na jas než na odstín barvy. Je proto možné přenášet složku Y ve vysokém rozlišení a složky Cb a Cr v redukovaném rozlišení, čímž se sníží datové toky např. v televizním vysílání.

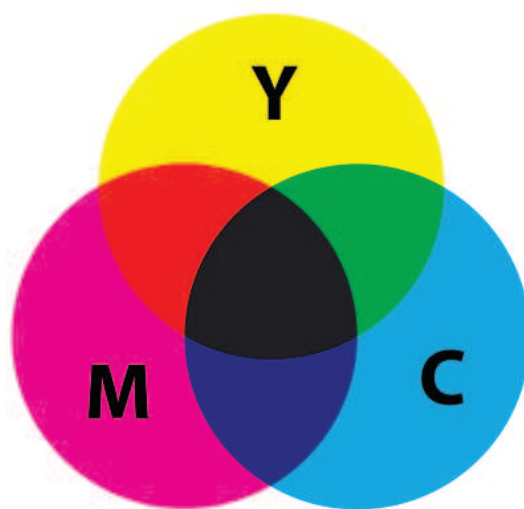


Obr. 1.6: CbCr rovina při konstantním jasů  $Y = 0,5$  [16].

## CMYK

Barevný systém CMYK patří jako jediný ze zde zmíněných modelů do skupiny subtraktivních systémů. Jeho největší využití je v tiskárnách.

Systém je jako většina ostatních pojmenován po prvních písmenech dílčích barevných složek – C (Cyane – azurová), M (Magenta – purpurová), Y (Yellow – žlutá) a K (Key, někdy používáno black).



Obr. 1.7: Skládání barev pomocí složek CMY systému CMYK [6].

## 1.2 Morfologické transformace

Morfologické transformace jsou postupy, při nichž se upravuje již vysegmentovaný obraz. Tyto transformace se provádějí až po segmentaci proto, že jejich použití pro šedotónový obraz by bylo velmi náročné.

Morfologické transformace se používají hlavně k zjednodušení tvaru objektů a zdůraznění jejich struktur.

### 1.2.1 Dilatace

Dilatace je operace, kdy k obrazu přičítáme jiný obraz. Musíme tedy chápat obraz jako matici čísel vyjadřujících barvu. K této matici přičítáme (Minkowského množinovým součtem) takzvanou **masku**, což je další matice, která má požadovaný rozměr (například jsou používány matice 2x1 nebo 3x3). Dilatace tedy zaplní díry (mezery v obraze), které mají šířku jednoho bodu a rozšíří celý obraz o jeden bod [13].

Výsledek dilatace je na obrázku 1.8. Pro ukázkou funkce dilatace a eroze byl použit Český lev.



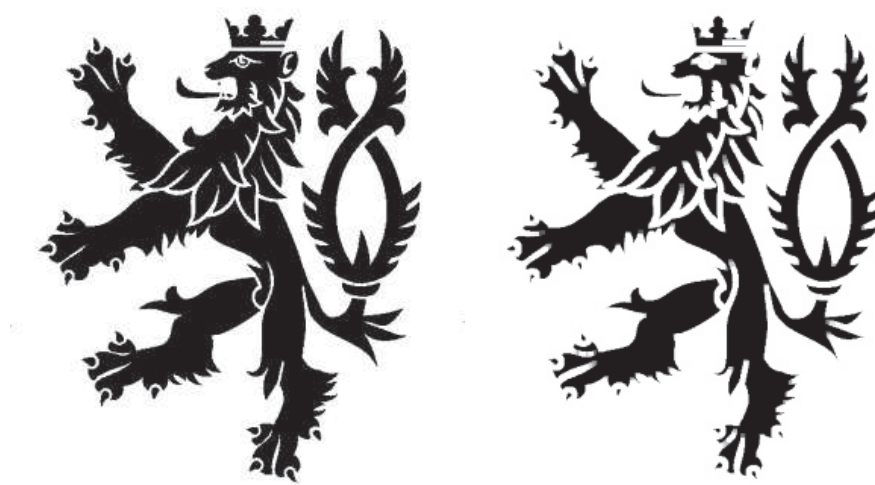
Obr. 1.8: Původní obrázek a obrázek po dilataci.

### 1.2.2 Eroze

Eroze je duální transformací k dilataci (není její inverzní transformací!). Jedná se o operaci, kdy od matice obrazu odečítáme Minkowského rozdíllem matici masky. Eroze odstraní objekty, které mají tloušťku jednoho bodu a ztenčí celý obraz o celý bod.

Pomocí eroze je možné získat obrysy obrazu, a to tak, že od původního obrazu odečteme jeho erozi [13].

Výsledek eroze je na obrázku 1.9.



Obr. 1.9: Původní obrázek a obrázek po erozi.

### 1.2.3 Otevření

Otevření je funkce, která je kombinací dilatace a eroze. Eroze, po které následuje dilatace, se nazývá **otevření**.

Otevření oddělí objekty, které jsou spojeny objektem o šířce 1 bodu. Tím dosáhneme zjednodušení struktury objektů [13].

### 1.2.4 Uzavření

Uzavření je funkce, která je kombinací dilatace a eroze. Dilatace, po které následuje eroze, se nazývá **uzavření**. Uzavření zaplní malé díry a také vyhladí obrys objektu [13].

## 1.3 Algoritmy použité v diplomové práci

### 1.3.1 Cannyho hranový detektor

Tento detektor je optimální hlavně pro hledání skokových hran vzhledem ke třem kritériím. První kritérium (detekční) zajišťuje, aby detektor neopomenul významné hrany, druhé (lokalizační) kritérium se stará o to, aby byl rozdíl mezi skutečnou a nalezenou pozicí hrany nejmenší a třetí kritérium zajišťuje, aby detektor nereagoval na jednu hranu vícenásobně.

Cannyho detektor pracuje s druhými derivacemi obrazové funkce a jeho výsledkem jsou velikosti i směry hran.

Funkce Cannyho hranového detektoru:

- eliminace šumu (např. Gaussovým filtrem);
- určení gradientu (první derivace obrazové funkce);
- nalezení lokálních maxim;
- eliminace nevýznamných hran [3].

Prvním krokem práce Cannyho hranového detektoru je odstranit šum. Tímto problémem se tato práce zabývá výše.

Druhým krokem je určení gradientu, tedy detekce hran. Nejčastěji se na hledání hran používá Sobelův operátor (ten bude probrán níže). Výstupem Sobelova operátoru je nejen velikost, ale i směr gradientu.

Jako další krok provádí Cannyho detektor hledání lokálních maxim, díky čemuž ztění hrany. Detektor tedy hledá bod, který má po směru i proti směru gradientu menší sousedy. Tento bod je potom označen jako hrana.

Posledním krokem je eliminace nevýznamných hran. Ta probíhá pomocí prahování s hysterezí. Určíme tedy, jaká hodnota gradientu je významná. Zvolíme minimální (T1) a maximální (T2) hodnotu (prahy), mezi kterými může gradient kolísat.

Pokud hodnota gradientu daného pixelu leží nad větším prahem  $T_2$ , je přímo označen jako hranový. Pokud posuzujeme bod, jehož hodnota leží mezi  $T_1$  a  $T_2$ , pak je označen jako hrana jedině pokud sousedí s bodem, který už byl jako hrana označen dříve [3].

### 1.3.2 Algoritmus K-means[1][11]

K-means (angl. K-means clustering) je metoda clusterové analýzy, která má rozdělit  $n$  bodů do  $k$  clusterů tak, že každý bod přiřadí nejbližšímu středu (clusteru).

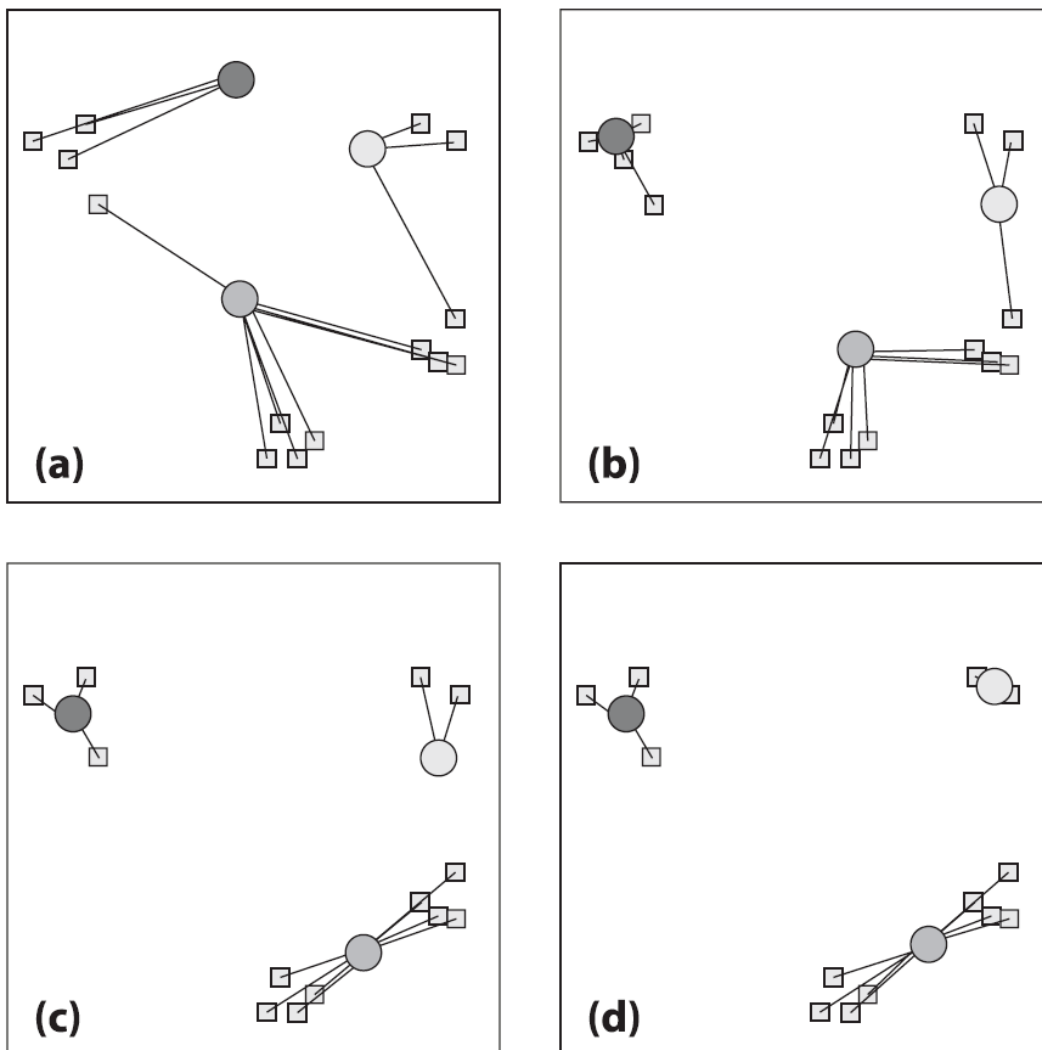
Pro řešení metody se používá standardní postup, který má následující kroky:

1. náhodně rozložit pozice  $k$  clusterů;
2. přiřadit každému clusteru body, které od něj mají nejmenší euklidovskou vzdálenost;
3. posunout střed do středu těchto přiřazených bodů;
4. opakovat kroky 2 – 3 dokud nebudou dvě po sobě jdoucí pozice clusterů shodné.

Tento postup nezaručuje, že bude výstup konvergovat ke globálnímu optimu. Výsledek algoritmu také může záviset na počátečním rozložení clusterů. Toto rozložení také může ovlivnit čas výpočtu algoritmu.

Na obrázku 1.10 je vidět postup algoritmu K-means.





Obr. 1.10: Postup algoritmu K-means [11].

### 1.3.3 Odstranění zkreslení v obrazu

Fotografování či filmování jakéhokoli objektu s sebou často přináší chyby ve formě zkreslení. Zkreslení mohou být způsobeny vadou čočky, konstrukcí snímacího čipu a objektivu a také špatnou perspektivou díky nepřesně upevněnému snímacímu zařízení.

## Změna perspektivy pomocí matice homogenní transformace [9]

Pro potřeby této diplomové práce je třeba, aby snímaný objekt ležel v rovině kolmo k objektivu. Vzhledem k tomu, že tato podmínka nemusí být vždy splněna, je nutné odstranit perspektivu. Vzorec pro odstranění perspektivy za pomoci matice homogenní transformace vypadá následovně:

$$(x_i, y_i) = \mathbf{T}(x, y), \quad (1.1)$$

kde  $x_i, y_i$  – body ve vstupním obrazu,  
 $x, y$  – body ve výstupním obrazu,  
 $\mathbf{T}$  – matice homogenní transformace.

Pokud do rovnice přidáme ještě váhovací funkci  $W$ , dostaneme:

$$\begin{pmatrix} Wx_i \\ Wy_i \\ W \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (1.2)$$

kde  $x_i, y_i$  – body ve vstupním obrazu,  
 $x, y$  – body ve výstupním obrazu,  
 $a, b, c, d, e, f, g, h$  – neznámé koeficienty pro změnu perspektivy.

Pokud soustavu rovnic podělíme váhovací funkcí, dostáváme 2 rovnice:

$$\begin{aligned} x_i &= \frac{ax + by + c}{gx + hy + 1} \\ y_i &= \frac{dx + ey + f}{gx + hy + 1}. \end{aligned} \quad (1.3)$$

Tyto rovnice se pro zjednodušení výpočtu dají přepsat do tvaru:

$$\begin{aligned} x_i &= ax + by + c - gxx_i - hyy_i \\ y_i &= dx + ey + f - gxy_i - hyy_i. \end{aligned} \quad (1.4)$$

Pro výpočet neznámých koeficientů vede řešení na metodu nejmenších čtverců.  
Pro výpočet koeficientů dostáváme soustavu 8 rovnic:

$$\begin{aligned}
\sum \frac{(ax + by + c - gxx_i - hyx_i - x_i)^2 + (dx + ey + f - gxy_i - hyy_i - y_i)^2}{\partial a} &= 0 \\
\sum \frac{(ax + by + c - gxx_i - hyx_i - x_i)^2 + (dx + ey + f - gxy_i - hyy_i - y_i)^2}{\partial b} &= 0 \\
\sum \frac{(ax + by + c - gxx_i - hyx_i - x_i)^2 + (dx + ey + f - gxy_i - hyy_i - y_i)^2}{\partial c} &= 0 \\
\sum \frac{(ax + by + c - gxx_i - hyx_i - x_i)^2 + (dx + ey + f - gxy_i - hyy_i - y_i)^2}{\partial d} &= 0(1.5) \\
\sum \frac{(ax + by + c - gxx_i - hyx_i - x_i)^2 + (dx + ey + f - gxy_i - hyy_i - y_i)^2}{\partial e} &= 0 \\
\sum \frac{(ax + by + c - gxx_i - hyx_i - x_i)^2 + (dx + ey + f - gxy_i - hyy_i - y_i)^2}{\partial f} &= 0 \\
\sum \frac{(ax + by + c - gxx_i - hyx_i - x_i)^2 + (dx + ey + f - gxy_i - hyy_i - y_i)^2}{\partial g} &= 0 \\
\sum \frac{(ax + by + c - gxx_i - hyx_i - x_i)^2 + (dx + ey + f - gxy_i - hyy_i - y_i)^2}{\partial h} &= 0.
\end{aligned}$$

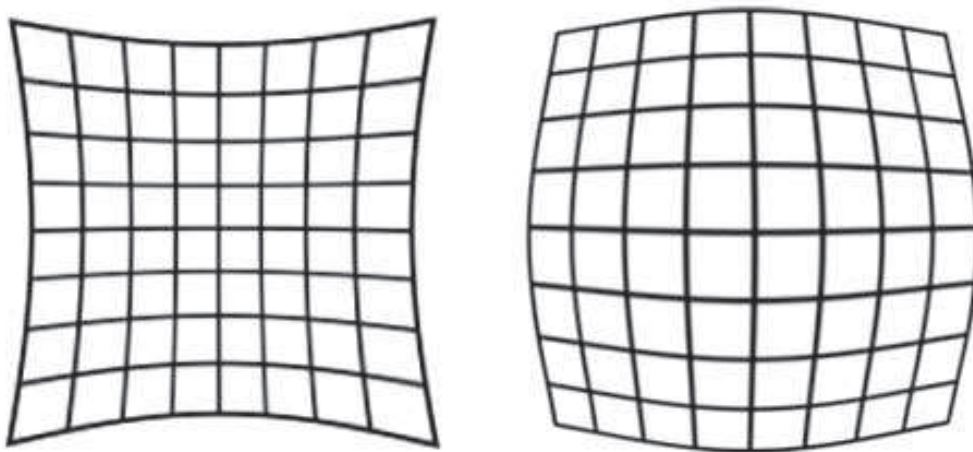
Po úpravě:

$$\begin{aligned}
\sum ax^2 + bxy + cx - gx^2x_i - hxyx_i - xx_i &= 0 \\
\sum axy + by^2 + cy - gxyx_i - hy^2x_i - y_xi &= 0 \\
\sum ax + by + c + gxx_i + hyx_i &= 0 \\
\sum dx^2 + exy + fx - gx^2x_i - hxyx_i - xx_i &= 0 \\
\sum dxy + ey^2 + fy - gxyx_i - hy^2x_i - y_xi &= 0 \quad (1.6) \\
\sum ax^2x_i + bxyx_i + cxx_i + dx^2y_i + exyy_i + fxy_i \\
- gx^2(x_i^2 + y_i^2) - hxy(x_i^2 + y_i^2) - x(x_i^2 + y_i^2) &= 0 \\
\sum axyx_i + by^2x_i + cyx_i + dxyy_i + ey^2y_i + fyy_i \\
- gxy(x_i^2 + y_i^2) - hy^2(x_i^2 + y_i^2) - y(x_i^2 + y_i^2) &= 0.
\end{aligned}$$

Dosazením všech (minimálně 4) korespondujících bodů je možné vypočítat neznámé  $a, b, c, d, e, f, g, h$  a dosadit do rovnice 1.4.

## Odstranění zkreslení snímacího zařízení

Vady objektivu jsou způsobeny zejména nedokonalostí a konstrukcí čočky v objektivu. Nejvíce se projevují 2 druhy zkreslení: **soudkové** a **poduškové**. Ukázky těchto zkreslení jsou na obrázku 1.11.



Obr. 1.11: Poduškovité a soudkovité zkreslení v obrazu [12].

Odstranění těchto zkreslení je možné pomocí rovnic:

$$\begin{aligned}x_i &= x(1 + c_1 r^2 + c_2 r^4 + c_3 r^6) \\y_i &= y(1 + c_1 r^2 + c_2 r^4 + c_3 r^6).\end{aligned}\tag{1.7}$$

Rovnice je možné vyřešit pomocí metody nejmenších čtverců. Tato metoda je popsána v předchozí kapitole.

## 1.4 Možné typy puzzle

Při průzkumu trhu byly zjištěny následující typy puzzle:

- **dětské puzzle**
  - nemá určitý tvar, barvy na sebe nenavazují
- **všechny dílky mají stejný tvar**
  - jedná se například o dětské kostky, ze kterých se dá složit šest různých obrázků
  - záleží pouze na barvě dílku u okrajů
- **klasické puzzle;**
  - záleží na tvaru i barvě dílku
- **všechny dílky mají stejnou barvu**
  - záleží pouze na tvaru dílků – typickým příkladem je puzzle Eternity II nebo “rozlité mléko”
- **3D puzzle**
  - většinou typické puzzle s tím rozdílem, že pokud složíme všechny dílky správně, dostaneme 3D objekt

Tato práce se bude zabývat skládáním klasických puzzle a puzzle, jehož dílky mají všechny stejný tvar (např. dětské kostky).

## 1.5 Výpočetní náročnost skládání puzzle

Pro skládání jednotlivých dílků, které musí být popsány lze použít několik postupů. První a výpočetně nejnáročnější metodou je kontrolovat všechny dílky mezi sebou. Protože má každý dílek čtyři strany, je počet kroků  $k$  ke složení puzzle, které má  $n$  dílků maximálně

$$k_{max} = \sum_{i=0}^{n-1} (4 + 2 \cdot i)(4 \cdot n - i). \quad (1.8)$$

Pro puzzle mající 9 dílků je maximální počet kroků 1680. Tato metoda je tedy v poměru počtu kroků a počtu dílků velmi náročná. Druhou možností je rozdělit si dílky do jednotlivých skupin a porovnávat dílky v rámci skupiny. Například rozdělení na tři skupiny – v první skupině jsou rohové dílky (mají dvě strany k otestování), ve druhé jsou hranové dílky (mají tři strany k otestování) a v poslední skupině jsou dílky ležící ve středu puzzle (testují se všechny čtyři strany). Dílky se porovnávají tak, že s rohovými dílky jsou porovnávány pouze dílky hranové. Nejprve se spojí všechny hranové dílky a následně se zkouší všechny vnitřní dílky. Pro puzzle, které má  $n$  dílků, z nichž čtyři dílky jsou v první skupině,  $n_2$  dílků je v druhé skupině a zbytek ve skupině třetí, je maximální počet kroků  $k$

$$k_{max} = 4 \cdot 2 \cdot 3 \cdot n_2 + \sum_{i=0}^{n_2-8} (n_2 - 8 - i)(3 \cdot 2) + \sum_{j=0}^{n-n_2-4} 4(n_2)(n - n_2 - 8 - j). \quad (1.9)$$

Pro puzzle o 9-ti dílcích, které rozdělíme do tří skupin (4 dílky rohové, 4 hranové a 1 dílek vnitřní) je počet kroků maximálně 100, což je výrazně méně než v první metodě.

## 1.6 Knihovny OpenCV

Pro programování metod probraných v následujících kapitolách musíme použít některé knihovny pro zpracování obrazu.

Knihovny používané v práci se nazývají **OpenCV**.

OpenCV (Open Source Computer Vision) jsou knihovny s otevřeným zdrojovým kódem, který spadá pod licenci **BSD** [2]. Licence BSD je jedna z nejmírnějších mezi otevřenými licencemi. Umožňuje šíření obsahu bez omezení pouze s tím, že musí být uveden autor a informace o licenci spolu s upozorněním na zřeknutí se odpovědnosti za dílo.

Po instalaci OpenCV knihoven do počítače je možno používat funkce, které jsou rozmístěny v různých souborech: **highgui**, **cv**, **cxcore** a v mnoha dalších. Uvedené 3 knihovny patří mezi nejdůležitější.

V souborech **highgui** se nacházejí funkce pro zobrazení a ukládání obrazu a práci s videem, **cv** obsahuje funkce počítačového vidění – filtry, hranové detektory, morfologické operace atd. **Cxcore** definuje základní statické a dynamické struktury.

Knihovny OpenCV jsou napsány v jazyce C a C++ a jsou funkční pod operačními systémy Linux, Windows a Mac OS X.

Knihovny jsou optimalizovány pro rychlé a efektivní zpracování obrazu v aplikacích reálného času. Mají kolem 500 funkcí, mezi které patří načtení a vytvoření obrázku, dilatace a eroze, zobrazení obrázku atp. Díky tomu výrazně urychlily řešení diplomové práce.[11].

## 1.7 Shrnutí

První část diplomové práce pojednávala o teoretických východiscích potřebných k sestavení puzzle.

Nejprve byl představen obor počítačového vidění a základní pojmy z tohoto oboru. Následně teoretická část pojednává o morfologických transformacích – dilataci, erozi, otevření a uzavření.

Další kapitola vysvětluje hlavní algoritmy použité v předložené práci – Cannyho hranový detektor, algoritmus K-means a metody pro odstranění zkreslení, ať už se jedná o zkreslení objektivu nebo zkreslení díky perspektivě.

Taktéž jsou uvedeny možné typy puzzle, které se na našem trhu běžně vyskytují.

V předposlední kapitole je zmíněna výpočetní náročnost puzzle pro dvě metody porovnávání – každý s každým nebo rozdělení na skupiny, přičemž skupiny se porovnávají zvlášť.

Knihovny OpenCV jsou představeny v poslední kapitole teoretického úvodu.



## 2 ŘEŠENÍ STUDENTSKÉ PRÁCE

Tato část diplomové práce se zabývá řešením dílčích kroků, kterými je nutné projít než je možné puzzle složit.

Před samotným zpracováním obrazu je nezbytné zvolit si scénu, ve které budou snímky pořizovány. Teprve poté probíhá zpracování vstupního obrazu. Postup zpracování obrazu je stručně uveden na následujících řádcích. Každé z hlavních částí zpracování se věnuje jedna z kapitol.

Nejprve probíhá předzpracování obrazu – odstranění zkreslení, geometrické transformace pro odstranění perspektivy a také vyříznutí dílku z pořízeného snímku.

Ve druhé fázi probíhá popis dílku. Popisy je možné rozdělit do dvou skupin, popis podle barvy a popis podle tvaru. Každému je věnována vlastní kapitola.

V posledním kroku je realizováno samotné složení puzzle. Zde bylo potřebné aplikovat algoritmy, které zkoumají propojení dílků, a díky tomu rozhodují o tom, jaké dílky k sobě opravdu patří.

## 2.1 Volba scény

Mezi první volby, které je nutné při návrhu scény udělat, patří volba barvy pozadí. Jednou z nejlepších a nejpoužívanějších barev pro účely snadno rozeznatelného pozadí se jeví jasně zelená. Tato barva se rovněž používá ve filmovém průmyslu pro takzvané *klíčování* [4]. Zelená barva je velmi kontrastní, a proto se v obraze dobře hledá.

Na obrázku 2.1 je zobrazena scéna používaná v Semestrálním projektu 1. Její nevýhodou je skutečnost, že ležící dílky vrhají stíny, které jsou následně vyhodnocovány jako části dílku. Další nevýhoda spočívá v nemožnosti kalibrace obrazu (složitě hledání kalibračních značek).



Obr. 2.1: Scéna pro pořizování obrazů dílků ze Semestrálního projektu 1.

Při volbě barvy pozadí je logické omezení – barva pozadí se nesmí vyskytovat na foceném dílku, jelikož by se při zpracování obrazu mohl spojit dílek s pozadím, a díky tomu by nebyl nalezen celý.

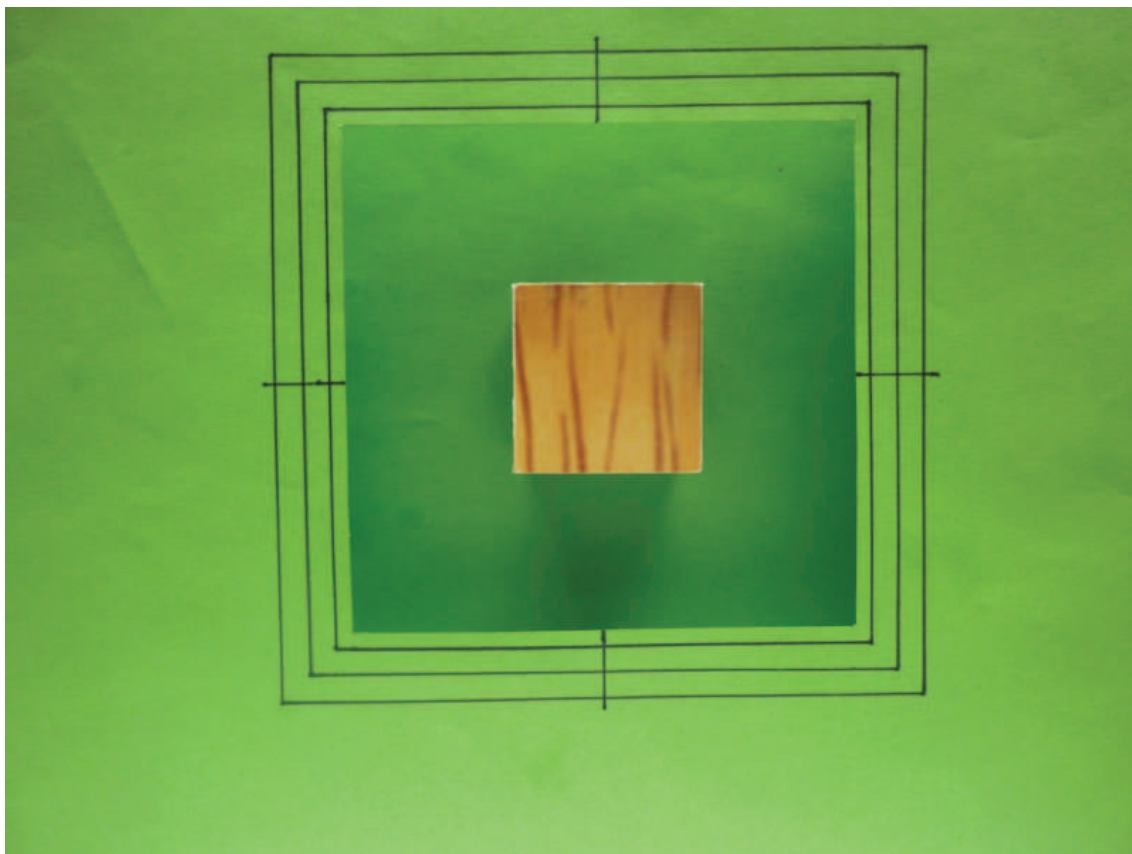
Po volbě pozadí je nutné zvolit počet dílků, které budou na jednom obraze pořizovány. Nejjednodušší vyhledávání dílku je ve chvíli, kdy v obraze leží pouze jeden dílek.

Následující volbou je pozice dílku. V dané práci leží dílek na vyvýšeném místě z důvodu minimalizace stínů vrhaných díky okolním zdrojům světla. Na obrázku 2.2 je vidět scéna pro pořízení obrazu. Černý rámeček má definovanou velikost a je určen pro kalibraci.



Obr. 2.2: Scéna pro pořizování obrazů dílků.

Pro základní potřebu by tato scéna vyhovovala, též by byly dílky velmi jednoduše nalezeny i zpracovány. Z důvodu následného spojování dílků je však nutné provádět přesnější kalibraci obrazu před zpracováním dílku. Z toho důvodu byly do scény přidány další kalibrační značky ve výšce samotného dílku, aby nedocházelo ke zkreslení mezi dílkem a značkami (obr. 2.3).

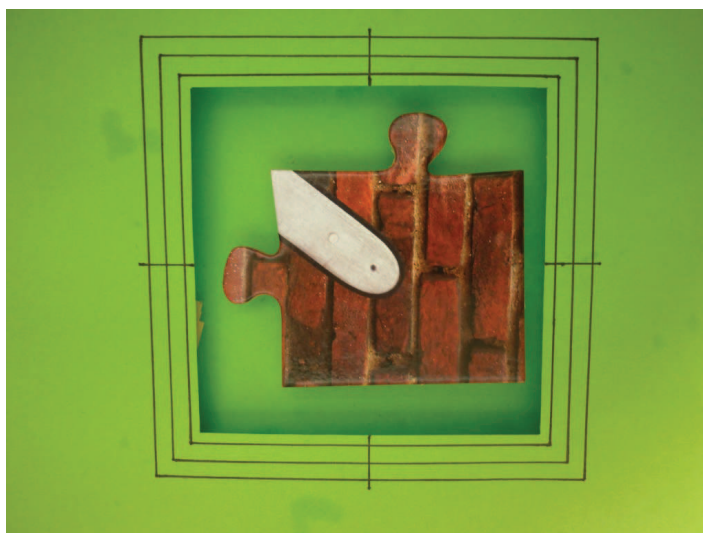


Obr. 2.3: Scéna pro pořizování obrazů dílků s kalibračními značkami.

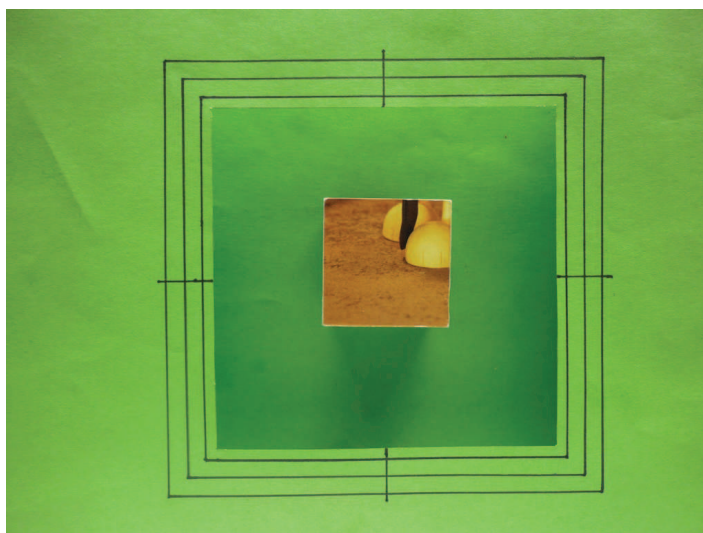
Z obrázku 2.3 je vidět, že by papír s kalibračními značkami vrhal stíny na spodní, podkladový papír. Tudíž musela být scéna osvětlena zespodu.

## 2.2 Databáze vstupních snímků

Z důvodu testování algoritmu byla vytvořena databáze obrázků obsahující dílky. Na obrázcích 2.4 a 2.5 jsou ukázky z těchto databází, v přílohách B.1 a B.2 jsou kompletní databáze funkčních vstupních obrázků. Příloha B.3 obsahuje obrázky nezpracovatelné z důvodu přítomnosti nezanedbatelného množství podkladové barvy uvnitř dílku.



Obr. 2.4: Dílek popisovaný podle tvaru i podle barvy.



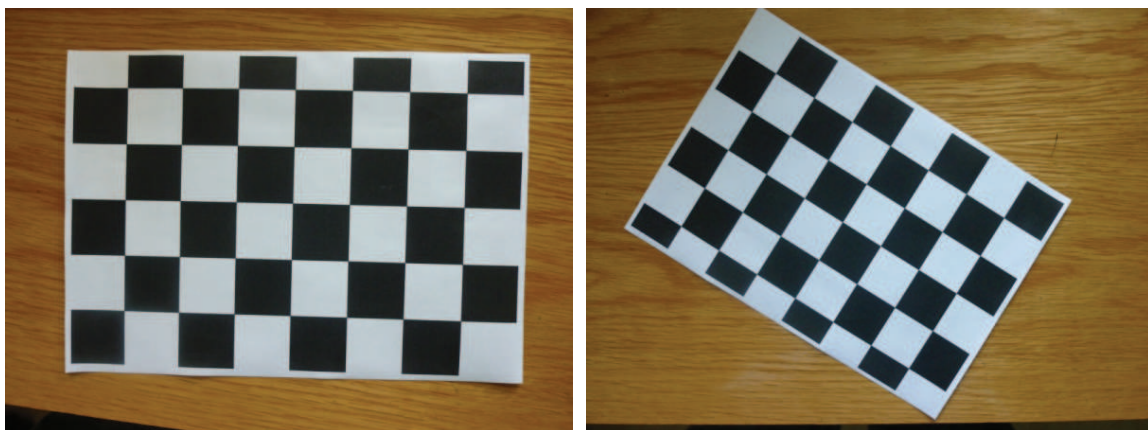
Obr. 2.5: Dílek popisovaný pouze podle barvy.

## 2.3 Předzpracování obrazu a vyříznutí dílku

Kapitola se zabývá první částí práce algoritmu pro skládání puzzle – předzpracováním obrazu a vyříznutím dílku z načteného snímku.

### 2.3.1 Předzpracování obrazu

Předzpracování obrazu pro potřeby předložené práce má za úkol odstranit vady objektivu a perspektivu. Nejprve probíhá odstranění vad objektivu, které by mohly zkreslovat pozici bodů použitých pro odstranění perspektivy. Vady objektivu se odstraňují pomocí funkce implementované v knihovnách OpenCV. Tato funkce potřebuje jako vstupní argument dvě či více fotografií, na kterých je šachovnice. Ta je funkcí na každé fotografii nalezena a z pozice bodů jsou vypočítány koeficienty potřebné k odstranění zkreslení (obrázek 2.6).



Obr. 2.6: Fotografie šachovnice použité pro určení zkreslení objektivu.

Rovnice použité pro odstranění zkreslení objektivu jsou následující [11]:

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2p_1 xy + p_2(r^2 + 2x^2) \\ p_1(r^2 + 2y^2) + 2p_2 xy \end{bmatrix}. \quad (2.1)$$

Z rovnice je patrné, že první část odstraňuje radiální zkreslení (poduškovité a soudkovité), protože je shodná s rovnicí probranou v kapitole 1.3.3. Zbytek rovnice představuje tzv. tangenciální zkreslení, které je způsobeno tím, že elementy v objektivu nejsou přesné.



## Odstranění perspektivy

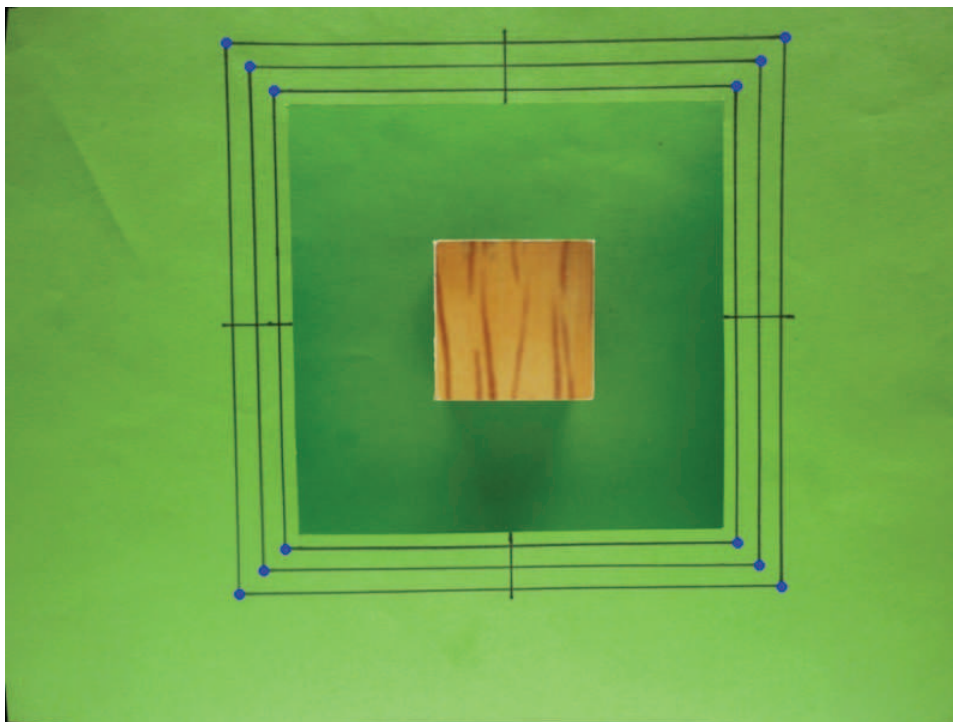
Po odstranění zkreslení je nutné odstranit perspektivu. Pro odstranění perspektivy je nezbytné nalézt kalibrační značky a jejich pozice (obrázek 2.8). Hledání kalibračních značek probíhá porovnáváním vzorů, které se mají v obrázku najít s celým obrázkem. Po daném procesu jsou v obrázku nalezeny maxima a jejich souřadnice se uloží. Vzory pro porovnávání jsou na obrázku 2.7.



Obr. 2.7: Vzory pro hledání kalibračních značek.

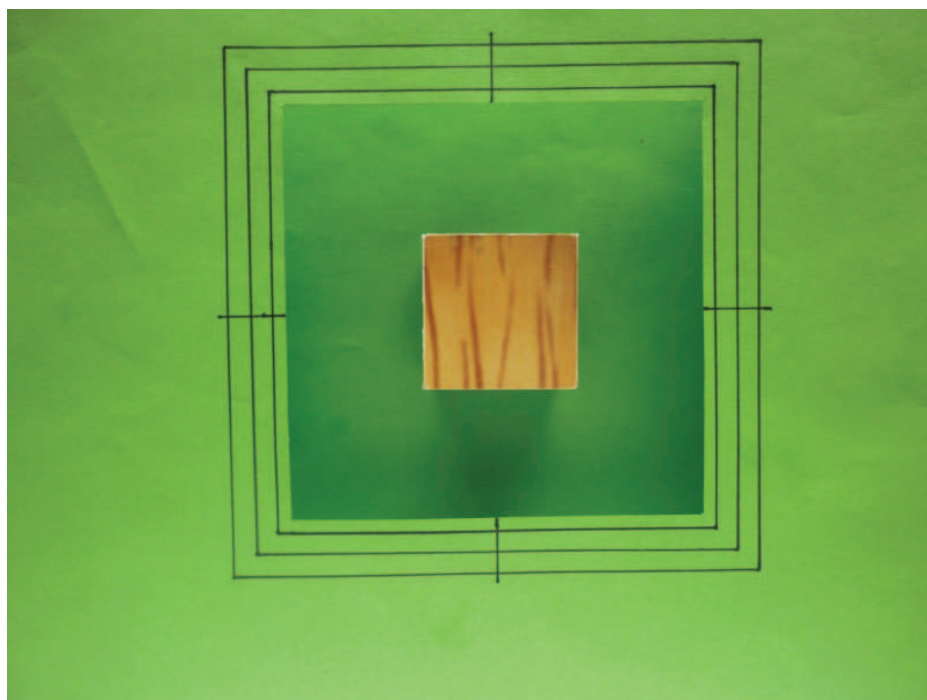
U vzorů je nutné, aby kalibrační značka ležela přesně uprostřed, protože následné nalezené maximum leží přesně na místě kalibrační značky.

Následně se vygeneruje seznam bodů, na které se mají nalezené značky transformovat. Ze soustavy dvou rovnic o osmi neznámých je podle kapitoly 1.3.3 metodou nejmenších čtverců vytvořeno osm rovnic o osmi neznámých. Tyto rovnice jsou naplněny nalezenými a zadanými body. Gaussovou eliminační metodou je soustava rovnic vyřešena. Koeficienty zjištěné při řešení soustavy se dosadí do původní soustavy dvou rovnic, do které dosadíme všechny body obrázku.

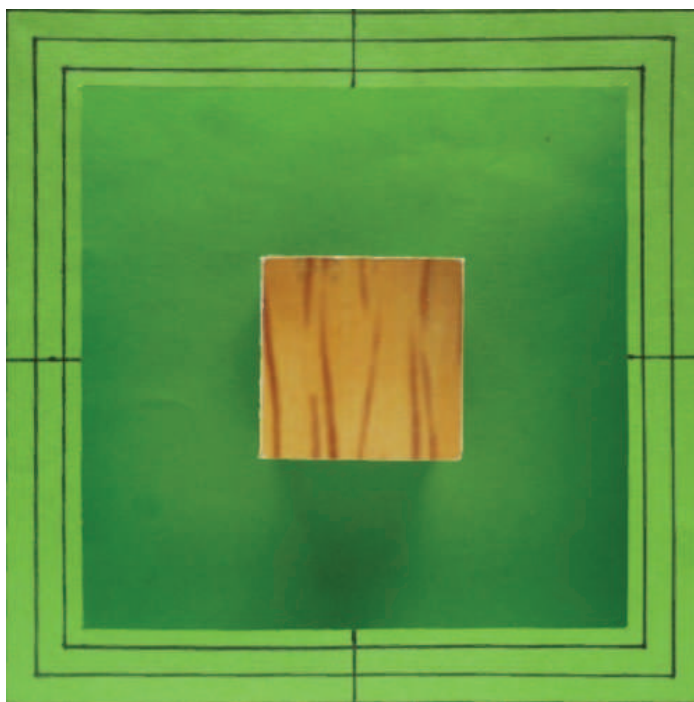


Obr. 2.8: Vyznačené nalezené kalibrační značky.

Na obrázku 2.9 je vidět obrázek focený z perspektivy, na obrázku 2.10 je scéna mající odstraněnou perspektivu a kde byl vyříznut pouze kalibrační čtverec.



Obr. 2.9: Scéna focená z úhlu.



Obr. 2.10: Scéna s odstraněnou perspektivou a vyříznutým kalibračním čtvercem.

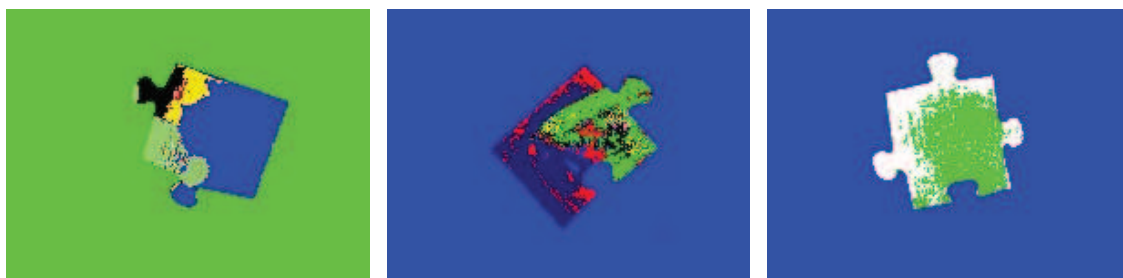


### 2.3.2 Vyříznutí dílku

Kapitola se zabývá metodou, pomocí které se ze vstupního obrazu vyřízne dílek.

K vyříznutí dílku je nutné najít v obrazu pixely, které patří k pozadí. Po neúspěchu z prvního semestrálního projektu, kde byla použita metoda hledání maxima histogramu, byl zvolen algoritmus K-means, který nejprve náhodně rozloží definovaný počet středů (clusterů), k nim se přiřadí nejbližší body, do jejichž středu se tyto clustery posunou. Tento postup se opakuje, dokud není splněna ukončovací podmínka – byl dokončen definovaný počet cyklů nebo jsou pozice středů v posledních dvou cyklech stejné.

Po odstranění zkreslení je obrázek zmenšen tak, aby jeho šířka byla 200 pixelů. Zmenšený obrázek je převeden do barevného systému HSV a následně je na něj v cyklech aplikován algoritmus K-means. Jedná se o dva vnořené cykly, v hlubším cyklu se mění počet clusterů, druhý cyklus je zde pro nalezení nejlepšího výsledku. K-means totiž inicializuje pozice clusterů náhodně, a tak je výsledek pro stejné nastavení pokaždé jiný. Na obrázku 2.11 jsou označeny body patřící k jednotlivým clusterům.



Obr. 2.11: Barevně označené clustery – označení pozadí.

Algoritmus probíhá tak, že ve výstupu K-means jsou spočítány body patřící k jednotlivým středům. Cluster, který má nejvíce bodů je považován za pozadí – ve výstupním obrázku je označen bílou barvou – zbytek bodů patří k dílku – označeny černou barvou. Tento obrázek se následně stává vstupem Cannyho hranového detektoru, v jehož výstupu se spočítají hrany (součet pixelů v obraze). Jako nejlepší je vyhodnoceno nastavení, ve kterém je nejmenší počet hran (minimální součet) a zároveň nalezený dílek zabírá v prostoru mezi kalibračními značkami 10 – 25 %. Označené dílky jsou na obrázku 2.12.

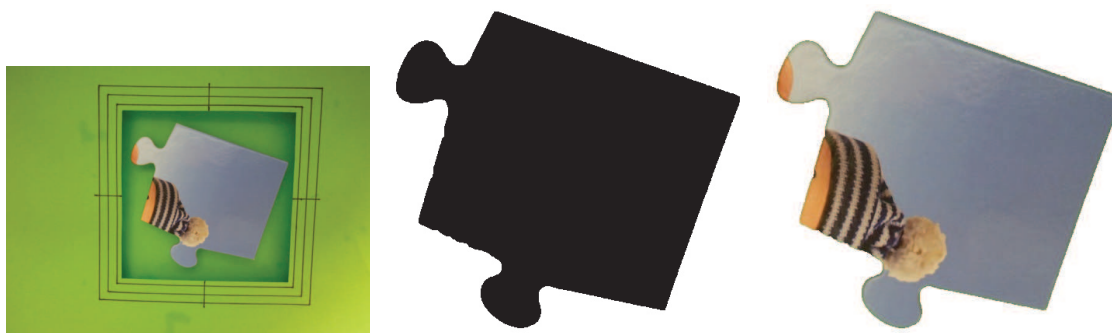


Obr. 2.12: Vyznačené dílky.

Obrázek získaný tímto postupem je nakonec zvětšen na původní velikost.

Ve zvětšeném obrázku je nalezen nejmenší vodorovně ležící obdélník, který se z něj vyřízne. Kolem hran nalezeného dílku je pomocí dilatace a eroze vytvořeno okolí, ve kterém se následně hledají hrany (ve zmenšeném obrázku by byly hrany nepřesné). Zpřesnění určení hran probíhá opět pomocí algoritmu K-means.

Původní a vyříznutý dílek jsou na obrázku 2.13.



Obr. 2.13: Původní obrázek, vyznačení vyříznutého dílku a vyříznutý dílek.

### 2.3.3 Otočení dílku

Po odstranění pozadí je dobré dílek otočit do vodorovné polohy kvůli lepšímu nalezení rohů a dalšímu popisu (například délka hran, označení zámků atd.). Samotné otočení probíhá tak, že jsou nalezeny kontury dílku, kterým se opíše nejmenší obdélník. V informacích o obdélníku je mimo jiné údaj o úhlu natočení. Dílek se natočí o zápornou hodnotu tohoto úhlu, a tím se získá dílek ve vodorovné poloze. Otočené dílky jsou vidět na následujícím obrázku.



Obr. 2.14: Dílky otočené do vodorovné polohy.

## 2.4 Popis dílku pomocí barev

Aby puzzle nebylo skládáno náhodně, ale pouze jak bylo výrobcem zamýšleno, je nutné si každý dílek popsat. Jedním z možných popisů je popis, kterým se zabývá tato kapitola.

### 2.4.1 Popis barvy na hranách

Pokud jsou porovnávány dílky patřící k sobě, platí, že na hraně, kterou k sobě pasují, mají stejnou barvu. Kvůli tomu je nutné barvu na hranách popsat.

S ohledem na rychlost porovnávání je důležité neporovnávat barvu pixel po pixelu, ale spíše po určitých úsecích. V nich se spočítá průměr barevných složek a zapíše se do popisu dílku.

Popis probíhá v následujících krocích:

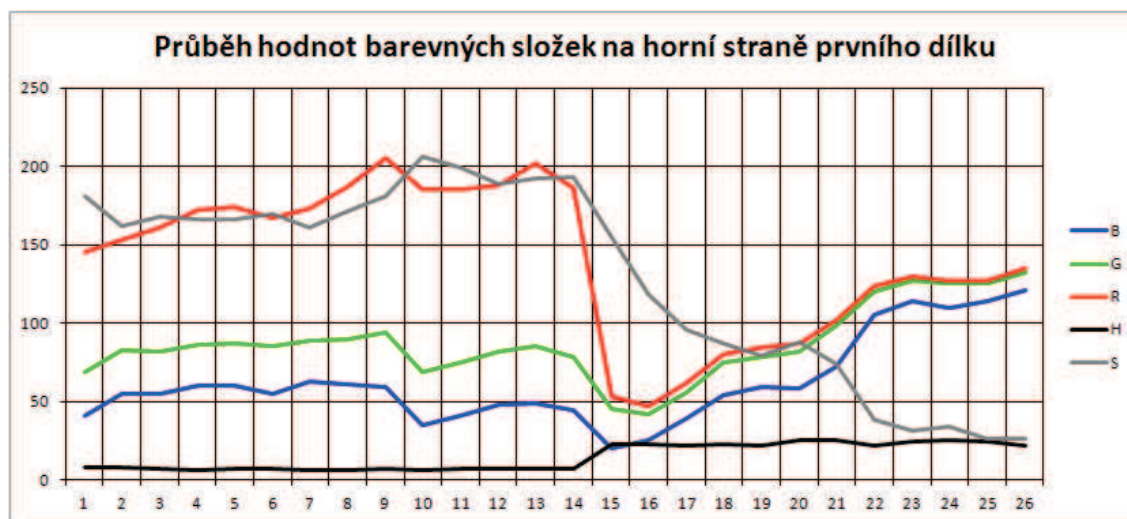
- nalezení rohů a výpočet délky každé hrany;
- rozdělení hran na stejně dlouhé díly;
- zmenšení dílku pomocí dilatace, aby nebyla popisována "porušená" hrana;
- součet po barevných složkách v jednotlivých pixelech;
- skok na krok s dilatací, dokud není popsáno v určeném počtu cyklů;
- vydělení každé barevné složky počtem bodů v každém dílu, ze kterého se skládá hrana – výpočet průměru v jednotlivých složkách.

Tímto způsobem se popíše všechny hrany. Popis probíhá v barevných systémech RGB a HSV.

Tabulka 2.1 obsahuje popis jedné strany jednoho dílku pomocí barvy na okraji. Obrázek 2.15 zobrazuje průběh jednotlivých barevných složek na hraně dílku.

Tab. 2.1: Popis jedné strany dílku č.1 pomocí barev na hranách.

část		1	2	3	4	5	6	7	8	9	10
Barevná složka	B	41	55	55	60	60	55	63	61	59	35
	G	69	83	82	86	87	85	89	90	94	69
	R	145	153	161	172	174	167	173	187	205	185
	H	8	8	7	6	7	7	6	6	7	6
	S	181	162	168	166	166	170	161	171	181	206
část		11	12	13	14	15	16	17	18	19	20
Barevná složka	B	41	48	49	44	20	25	39	54	59	58
	G	75	82	85	78	45	42	56	75	78	82
	R	185	188	202	186	53	47	62	80	84	87
	H	7	7	7	7	23	23	22	23	22	25
	S	199	189	192	193	155	118	96	87	79	88
část		21	22	23	24	25	26				
Barevná složka	B	72	105	114	110	114	121				
	G	98	120	127	125	125	132				
	R	102	124	130	127	127	135				
	H	25	22	24	25	24	22				
	S	74	38	31	34	26	26				



Obr. 2.15: Průběh barevných složek na vrchním okraji dílku č. 1.

### 2.4.2 Popis barvy na dílku

Pro urychlení porovnávání je možné použít další popis, který popisuje důležité barvy vyskytující se na dílku.

Důležité barvy vyskytující se na dílku je možné popsat až ve chvíli, kdy jsou načteny a zpracovány všechny dílky.

Algoritmus pro nalezení důležitých barev:

- každý vyříznutý barevný dílek převést do systému HSV;
- složky H a S pixelů, které náleží dílku nakopírovat postupně do dvouřádkové matice, která se stává vstupem pro algoritmus K-means;
- algoritmus K-means spouštět s čím dál menším počtem clusterů, pokud jsou nalezeny dvě blízké barvy;
- rozdělit výsledná data zpět k dílkům, ke kterým patří.

Po tomto procesu nese každý dílek informaci, které barvy v něm leží (přesněji nese informaci o čísle clusteru, ke kterému barva náleží). V tuto chvíli je nutné zjistit, kolik procent dílku jednotlivým clusterům náleží. Čísla clusterů, kterým náleží méně než jedno procento dílku se neuplatní a zbytek se seřadí podle toho, jakou část cluster v dílku pokrývá.

## 2.5 Popis dílku pomocí tvaru

Při skládání puzzle se člověk rozhoduje nejen podle barvy, ale také podle tvaru.

Popisovat podle tvaru je možné pomocí různých metod – aproximovat tvar dílku křivkou, použít směrové nebo normálové vektory, určit rozměry a tvar zámků a mnoho dalších.

V této práci byly použity normálové vektory spolu s popisem zámků.

### 2.5.1 Popis tvaru pomocí normálových vektorů

Pomocí normálových vektorů je možné popsat tvar dílku. Normálový vektor se získá ze směrového vektoru zaměněním souřadnic  $x$  a  $y$  a změnou znaménka u jedné z nich.

Před určením normálových vektorů je nutné určit pozici rohů dílku.

#### Nalezení pozice rohů

Pozice rohů dílku je potřebná nejen k popisu pomocí tvaru, ale také k popisu pomocí barev. Pro zjištění pozice rohů se používá binární obrázek znázorňující pixely, na kterých se vyskytuje dílek.

Pro zjištění pozice rohů je nutné nejprve odstranit zámký dílku, které vystupují mimo obdélník tvořící hlavní tělo dílku. Odstranění zámký je realizováno pomocí jednoduchého cyklu *while*, ve kterém je počítán počet bodů náležících dílku v průmětu do os  $x$  a  $y$ . V těchto osách potom algoritmus maže celé sloupce (řádky) dokud není počet bodů patřících dílku v průmětu do dané osy větší než deset procent výšky (šířky) dílku.



Obr. 2.16: Dílky s odstraněnými zámký.

V takto upraveném binárním obrázku se najde těžiště dílku, čtyři nejvzdálenější body náležící dílku jsou rohy. Na obrázku 2.17 jsou zobrazeny rohy a vyznačeno těžiště. Dílky jsou oproti běžnému zobrazení v předložené práci invertovány z důvodu hledání rohů, které v těchto invertovaných obrázcích probíhá.



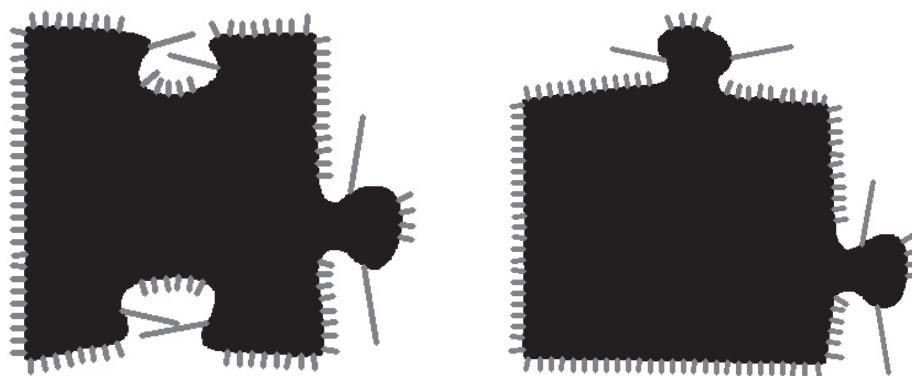
Obr. 2.17: Dílky s vyznačenými rohy a těžištěm.

Společně s pozicemi rohů jsou do popisu dílku zapsány i délky hran (vzdálenosti mezi jednotlivými rohy).

Po určení pozice rohů jsou popisovány hrany pomocí normálových vektorů. Postup pro získání tohoto popisu je následující:

- rozdělení každé hrany na stejně velké dílky;
- odečtení pozic konce a začátku dílku;
- převod směrového vektoru na normálový;
- posun na další dílek na hraně.

Na následujícím snímku jsou zobrazeny dílky s normálovými vektory.



Obr. 2.18: Dílky s vyznačenými vektory.



Tabulka 2.2 obsahuje popis pomocí normálových vektorů na horní straně prvního dílku.

Tab. 2.2: Velikosti normálových vektorů horní strany dílku č. 1.

část		1	2	3	4	5	6	7	8	9	10
Normála	x	0	1	0	0	0	1	1	-1	0	-63
	y	-7	-10	-10	-10	-10	-10	-10	-10	-10	-10
část		11	12	13	14	15	16	17	18	19	20
Normála	x	-2	-1	1	0	3	54	1	0	1	0
	y	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
část		21	22	23	24	25	26				
Normála	x	1	1	1	0	0	0				
	y	-10	-10	-10	-10	-10	-10				

## 2.5.2 Popis zámků puzzle

Druhý tvarový popis použitý v diplomové práci popisuje zámký (výstupky, díry) na puzzle.

Popis probíhá pouze na základě výšky (hloubky) zámků. Podle pozice rohů se nejprve nalezne průměrná pozice hrany. Od této hrany se hledají extrém (minimum a maximum). Velikost těchto hodnot rozhoduje o označení – díra, výstupek, hrana. K popisu se také přikládá velikost extrému.

## 2.6 Složení puzzle

Kapitola se zabývá metodami pro určení nejlépe se hodících dílků. Popsány jsou metody pro prvotní rozdělení nejlépe hodících se dílků, kontrolu s párovým dílkem, otočení dílku do správného směru a zobrazení složeného puzzle.

### 2.6.1 Hledání nejlépe hodících se dílků

Jedná se o algoritmus hledající určitý počet nejlépe shodujících se dílků.

Každá dvojice dílků je nejprve otestována, zdali obsahuje stejnou barvu a zda se nejedná o dva rohové dílky (oba dílky mají 2 strany bez zámků). Pokud dílky této podmínce vyhovují, jsou testovány jednotlivé strany dílků způsobem každý s každým. Hrany jsou nejprve otestovány, zda se jedná o dvojici díra – výstupek. Po splnění podmínky je počítán rozdíl mezi jednotlivými částmi. Vzhledem k tomu, že jsou dílky popisovány proti směru hodinových ručiček, je jedna strana kontrolována od začátku a druhá od konce. Rozdíl je počítán z každé složky zvlášť a absolutní hodnota rozdílů je přičtena k celkovému rozdílu. Stejným způsobem jsou sčítány normálové vektory (součet dvou normálových vektorů, které směřují ven z dílku, by měl být nula).

Při experimentování s nastavením algoritmu bylo zjištěno, že je nutné váhovat každý rozdíl jinou hodnotou. Vzhledem k tomu, že při počítání rozdílů barvy jsou používány složky RGB a složka H ze systému HSV, je žádoucí dát popisu podle složky H větší váhu, neboť se jedná o jedinou barvonosnou složku oproti třem složkám v systému RGB.

Stejným způsobem je potřebné váhovat rozdíl při počítání podle normálových vektorů.

Hodnota rozdílu (získaná součtem všech dílčích váhovaných rozdílů) se porovnává s ostatními rozdíly. Ty jsou uloženy v seznamu a seřazeny od nejmenšího po největší. Pokud je rozdíl menší než hodnota uložená v seznamu, vloží se aktuální hodnota spolu s odkazem na dílek a hranu do seznamu a následující odkazy a hodnoty se posunou dolů.

Po kontrole všech dílků vypadá popis každé hrany následovně: pořadí shody – číslo dílku, na který je ukazováno; číslo hrany, na kterou se ukazuje; velikost rozdílu (viz text níže).

12. dílek

Strana 0

0 – není par

Strana 1

0 – 7; 0; 3784 1 – 14; 2; 5839 2 – 6; 2; 7281 3 – není par

Strana 2

0 – 15; 3; 3761 1 – 17; 3; 5120 2 – 7; 3; 6056 3 – 3; 3; 7107  
4 – 17; 1; 7515 5 – 20; 3; 7936 6 – není par

Strana 3

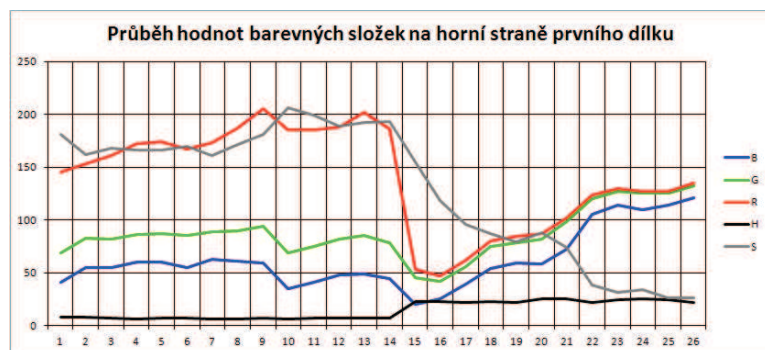
0 – 11; 0; 2697 1 – 2; 0; 9271 2 – není par

Tabulky 2.3 a 2.4 obsahují hodnoty pro porovnávání dvou stran dílků, jejichž složení vykresluje obrázek 2.21. Grafy zobrazující hodnoty barev na hranách a jejich rozdíl jsou na obrázcích 2.19, 2.20, 2.22, 2.23 a 2.24.

Po porovnání všech dílků je důležité tyto odkazy ověřit a chybné odkazy na dílky odstranit.

Tab. 2.3: Tabulka popisující horní hranu dílku 1 pomocí normálových vektorů a barev na hranách.

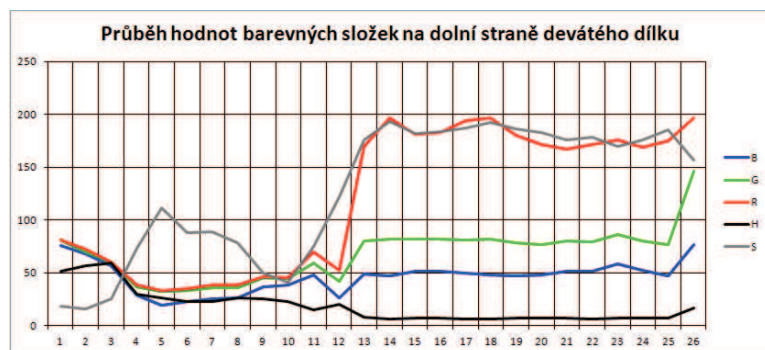
část		1	2	3	4	5	6	7	8	9	10
Barevná složka	<b>B</b>	41	55	55	60	60	55	63	61	59	35
	<b>G</b>	69	83	82	86	87	85	89	90	94	69
	<b>R</b>	145	153	161	172	174	167	173	187	205	185
	<b>H</b>	8	8	7	6	7	7	6	6	7	6
	<b>S</b>	181	162	168	166	166	170	161	171	181	206
Normála	<b>x</b>	0	1	0	0	0	1	1	-1	0	-63
	<b>y</b>	-7	-10	-10	-10	-10	-10	-10	-10	-10	-10
část		11	12	13	14	15	16	17	18	19	20
Barevná složka	<b>B</b>	41	48	49	44	20	25	39	54	59	58
	<b>G</b>	75	82	85	78	45	42	56	75	78	82
	<b>R</b>	185	188	202	186	53	47	62	80	84	87
	<b>H</b>	7	7	7	7	23	23	22	23	22	25
	<b>S</b>	199	189	192	193	155	118	96	87	79	88
Normála	<b>x</b>	-2	-1	1	0	3	54	1	0	1	0
	<b>y</b>	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
část		21	22	23	24	25	26				
Barevná složka	<b>B</b>	72	105	114	110	114	121				
	<b>G</b>	98	120	127	125	125	132				
	<b>R</b>	102	124	130	127	127	135				
	<b>H</b>	25	22	24	25	24	22				
	<b>S</b>	74	38	31	34	26	26				
Normála	<b>x</b>	1	1	1	0	0	0				
	<b>y</b>	-10	-10	-10	-10	-10	-10				



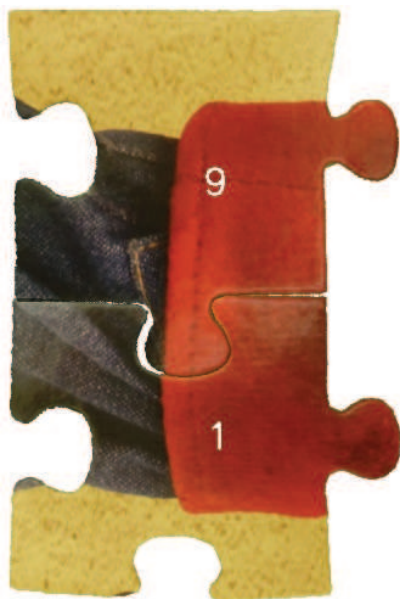
Obr. 2.19: Průběh barevných složek popisující vrchní stranu dílku č. 1 pomocí barev.

Tab. 2.4: Tabulka popisující dolní hranu dílku 9 pomocí normálových vektorů a barev na hranách.

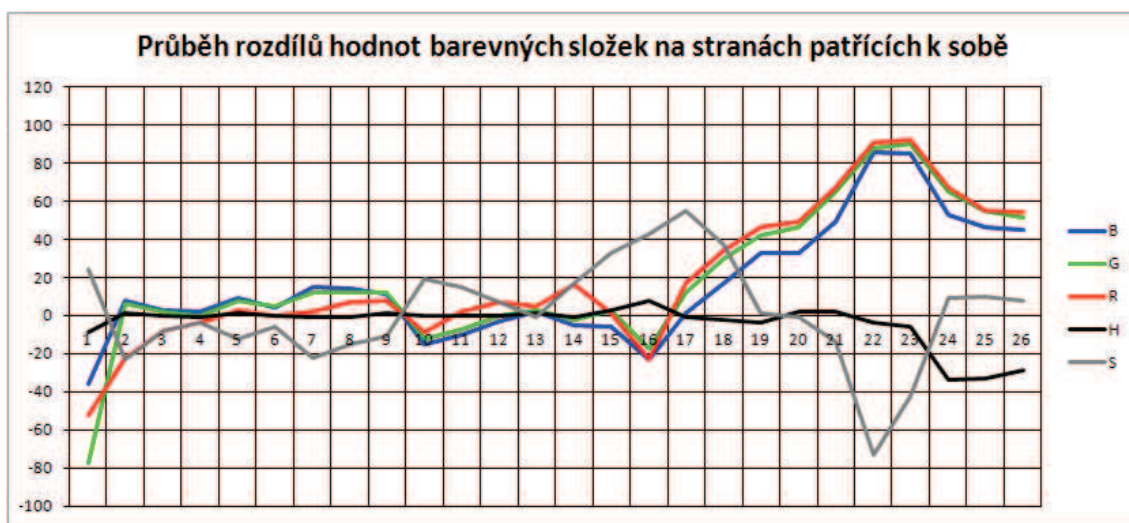
část		1	2	3	4	5	6	7	8	9	10
Barevná složka	B	76	68	57	29	19	23	25	26	37	38
	G	81	70	60	37	32	33	36	36	45	44
	R	81	72	60	38	33	35	38	38	46	45
	H	51	57	59	30	26	23	23	26	25	23
	S	18	16	25	73	111	88	89	78	50	41
Normála	x	-2	0	0	0	0	0	0	-1	0	0
	y	10	10	10	10	10	10	10	10	10	10
část		11	12	13	14	15	16	17	18	19	20
Barevná složka	B	48	26	49	47	51	51	50	48	47	48
	G	59	42	80	82	82	82	81	82	78	77
	R	70	52	170	197	181	183	194	197	180	171
	H	15	20	8	6	7	7	6	6	7	7
	S	75	122	176	193	182	184	187	192	186	183
Normála	x	-6	-51	-3	-1	1	2	6	58	1	0
	y	10	10	10	10	10	10	10	10	10	10
část		21	22	23	24	25	26				
Barevná složka	B	51	51	58	52	47	77				
	G	80	79	86	80	77	146				
	R	167	171	176	169	175	197				
	H	7	6	7	7	7	17				
	S	176	178	170	176	185	157				
Normála	x	0	0	0	0	0	2				
	y	10	10	10	10	10	5				



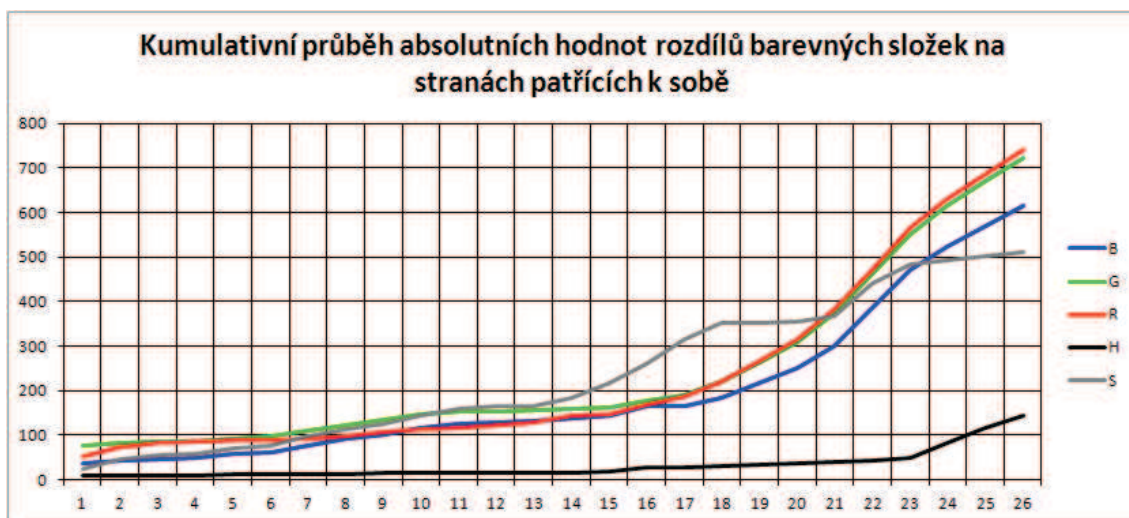
Obr. 2.20: Průběh barevných složek popisující spodní stranu dílku č. 9 pomocí barev.



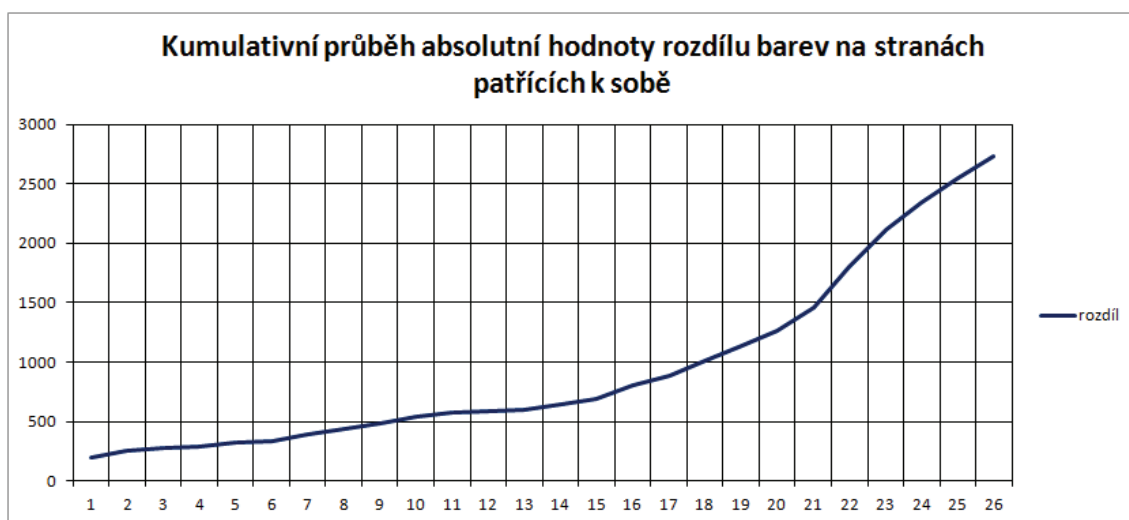
Obr. 2.21: Spojené dílky 1 a 9.



Obr. 2.22: Průběh rozdílů jednotlivých barevných složek při spojení dílků 1 a 9.



Obr. 2.23: Graf kumulačního průběhu absolutních hodnot rozdílů jednotlivých barevných složek při spojení dílků 1 a 9.

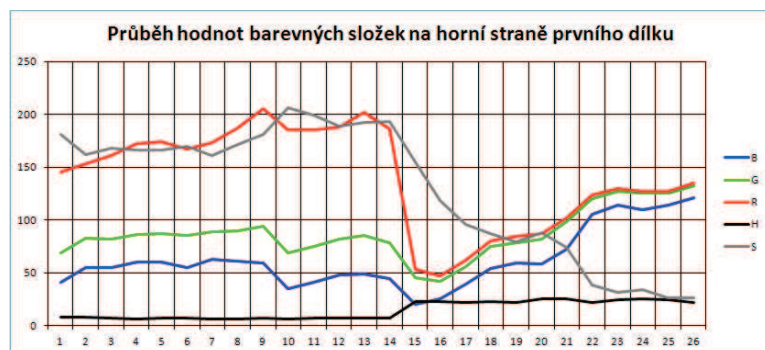


Obr. 2.24: Graf kumulačního průběhu absolutní hodnoty součtu rozdílů absolutních hodnot jednotlivých barevných složek při spojení dílků 1 a 9.

Dílky zobrazené na předchozích obrázcích představovaly nejlepší shodu (a dílky k sobě patřily). V tabulkách 2.5 a 2.6 obsahují hodnoty pro porovnávání dvou stran dílků, jejichž složení vykresluje obrázek 2.27 (obrázky k sobě nepatří). Grafy zobrazující hodnoty barev na hranách a jejich rozdíl jsou na obrázcích 2.25, 2.26, 2.28, 2.29 a 2.30.

Tab. 2.5: Tabulka popisující horní hranu dílku 1 pomocí normálových vektorů a barev na hranách.

část		1	2	3	4	5	6	7	8	9	10
Barevná složka	B	41	55	55	60	60	55	63	61	59	35
	G	69	83	82	86	87	85	89	90	94	69
	R	145	153	161	172	174	167	173	187	205	185
	H	8	8	7	6	7	7	6	6	7	6
	S	181	162	168	166	166	170	161	171	181	206
Normála	x	0	1	0	0	0	1	1	-1	0	-63
	y	-7	-10	-10	-10	-10	-10	-10	-10	-10	-10
část		11	12	13	14	15	16	17	18	19	20
Barevná složka	B	41	48	49	44	20	25	39	54	59	58
	G	75	82	85	78	45	42	56	75	78	82
	R	185	188	202	186	53	47	62	80	84	87
	H	7	7	7	7	23	23	22	23	22	25
	S	199	189	192	193	155	118	96	87	79	88
Normála	x	-2	-1	1	0	3	54	1	0	1	0
	y	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
část		21	22	23	24	25	26				
Barevná složka	B	72	105	114	110	114	121				
	G	98	120	127	125	125	132				
	R	102	124	130	127	127	135				
	H	25	22	24	25	24	22				
	S	74	38	31	34	26	26				
Normála	x	1	1	1	0	0	0				
	y	-10	-10	-10	-10	-10	-10				

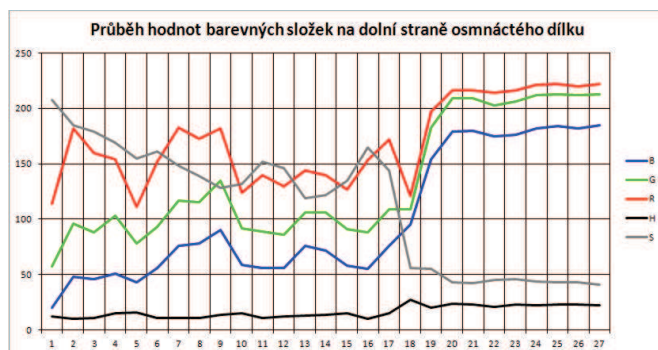


Obr. 2.25: Průběh barevných složek popisující vrchní stranu dílku č. 1 pomocí barev.



Tab. 2.6: Tabulka popisující dolní hranu dílku 18 pomocí normálových vektorů a barev na hranách.

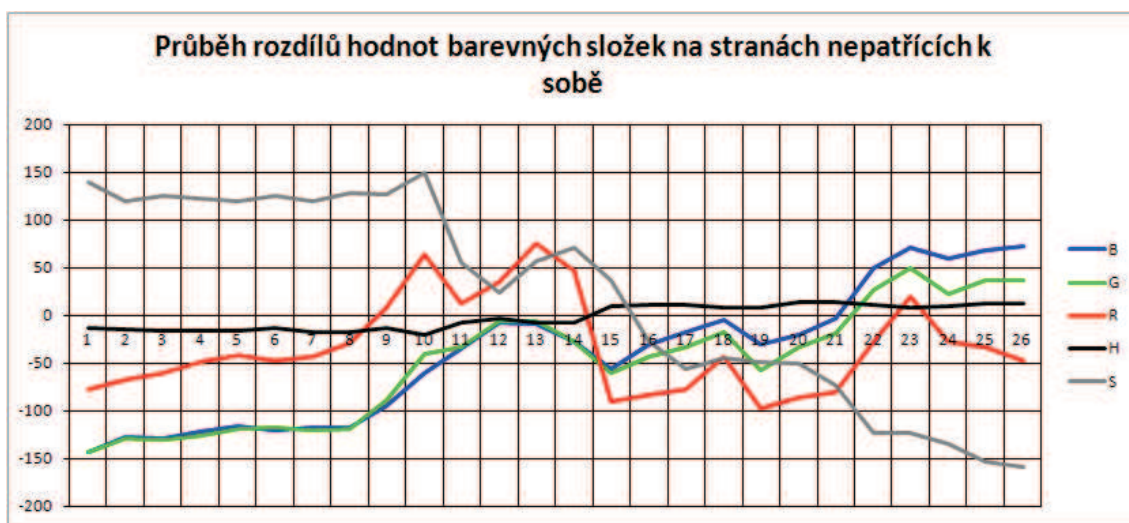
část		1	2	3	4	5	6	7	8	9	10
Barevná složka	B	20	48	46	51	43	56	76	78	90	59
	G	57	96	88	103	78	93	117	115	135	92
	R	114	182	160	154	111	152	183	173	182	124
	H	12	10	11	15	16	11	11	11	14	15
	S	208	185	179	169	155	161	148	139	128	132
Normála	x	0	0	0	-1	1	0	-1	1	0	3
	y	-7	-10	-10	-10	-10	-10	-10	-10	-10	-10
část		11	12	13	14	15	16	17	18	19	20
Barevná složka	B	56	56	76	72	58	55	76	95	154	179
	G	89	86	106	106	91	88	109	109	183	209
	R	140	130	144	140	127	153	172	121	197	216
	H	11	12	13	14	15	10	15	27	20	24
	S	152	146	119	122	135	165	144	56	55	43
Normála	x	71	0	-1	-4	-3	-9	-56	0	1	2
	y	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
část		21	22	23	24	25	26	27			
Barevná složka	B	180	175	176	182	184	182	185			
	G	209	203	206	212	213	212	213			
	R	216	214	216	221	222	220	222			
	H	23	21	23	22	23	23	22			
	S	42	45	46	44	43	43	41			
Normála	x	0	1	0	0	1	0	1			
	y	-10	-10	-10	-10	-10	-10	-10			



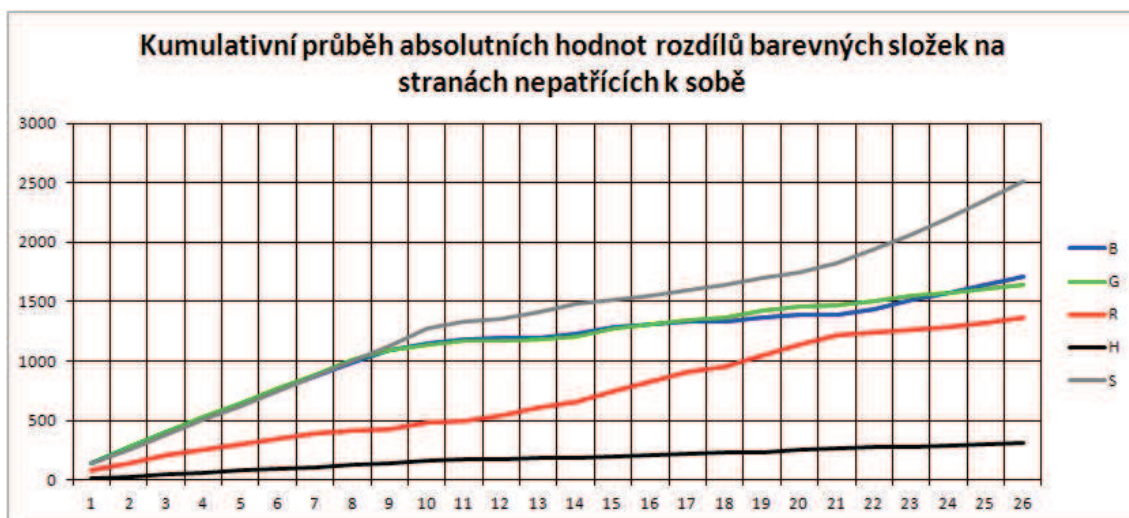
Obr. 2.26: Průběh barevných složek popisující dolní stranu dílku č. 18 pomocí barev.



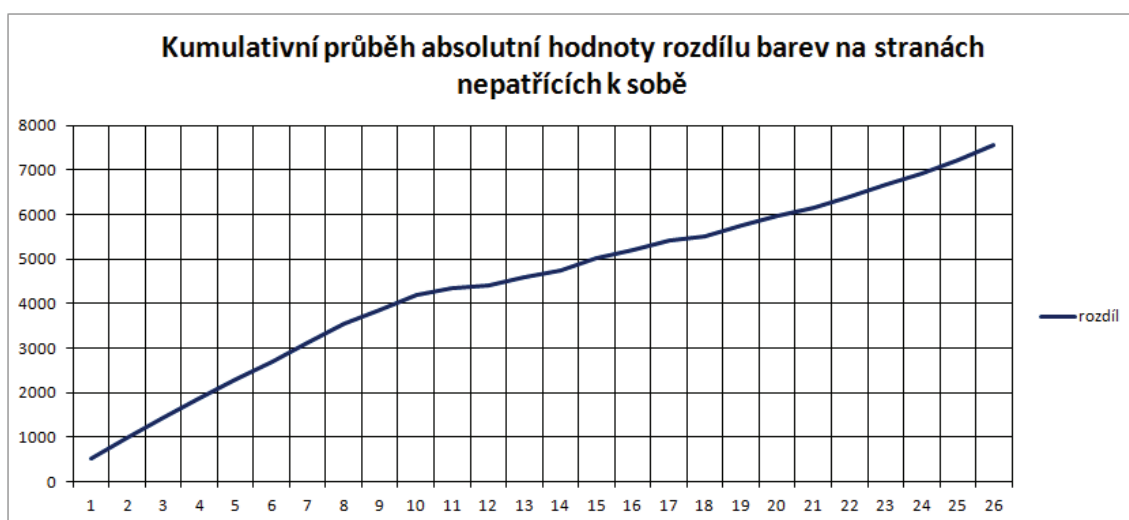
Obr. 2.27: Spojené dílky 1 a 18.



Obr. 2.28: Průběh rozdílů jednotlivých barevných složek při spojení dílků 1 a 18.



Obr. 2.29: Graf kumulačního průběhu absolutních hodnot rozdílů jednotlivých barevných složek při spojení dílků 1 a 18.



Obr. 2.30: Graf kumulačního průběhu absolutní hodnoty součtu rozdílů absolutních hodnot jednotlivých barevných složek při spojení dílků 1 a 18.

Z výsledků je zřetelné, že rozdíly mezi pasujícími a nepasujícími dílky jsou většinou znatelné. Pokud nastane situace, kdy bude nalezena nejlepší shoda s dílkem, který patří na jiné místo, další algoritmy by měly tuto shodu odstranit.

### 2.6.2 Kontrola platnosti vytvořených párů

Po vytvoření seznamu nejlépe shodujících se dílků je nutné seznam ověřit, aby dílky patřící k sobě odkazovaly na sebe navzájem.

Kontrolován je postupně každý dílek a je porovnáván s dílkem, na který ukazuje. Pokud neukazují na sebe navzájem, probíhá další kontrola – kontroluje se odkazovaný dílek s dílkem, na který tento dílek odkazuje. Vznikají dvě možnosti:

- odkaz platí, smaže se odkaz aktuálního dílku a kontrola probíhá znovu;
- odkaz neplatí, smaže se odkaz na odkazovaném dílku a kontrola probíhá znovu.

Po kontrole všech dílků by měly dvojice dílků odkazovat na jedné straně na sebe navzájem. Tyto odkazy stále ještě nejsou platné, protože dílky mohou být špatně natočeny.

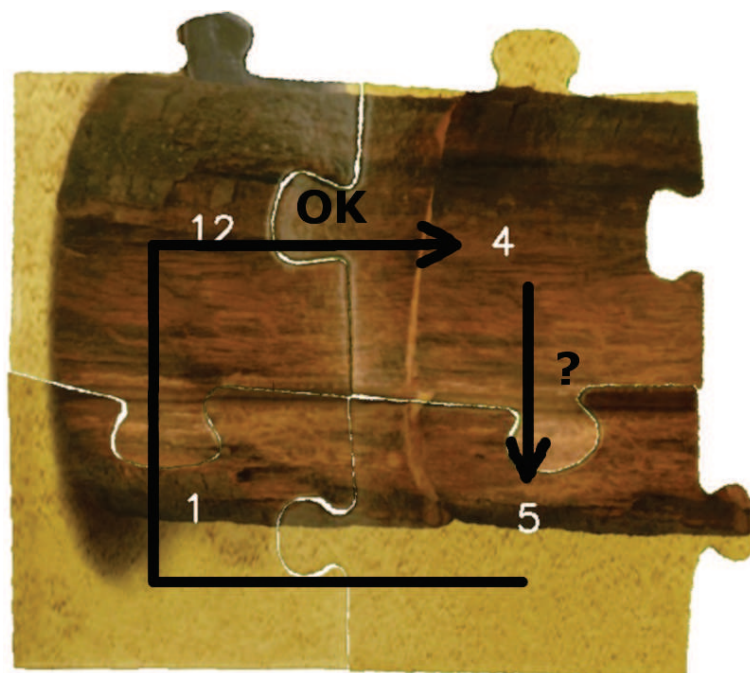
### 2.6.3 Otočení dílků

Na začátku této operace by měly všechny dílky odkazovat na správnou dvojici.

Nejprve je nalezen rohový dílek, na něm jsou uloženy dva směry, na kterých leží sousední dílek (nerohové hrany). Kontrola natočení probíhá od tohoto rohového dílku směrem č. 1, kde se po kontrole natočení dílku posouvá kontrola stále stejným směrem, dokud tímto směrem existuje dílek. Jakmile algoritmus dojde na kraj puzzle, vrátí se kontrola k prvnímu dílku a kontroluje se dílek ve druhém směru. Po jeho kontrole se opět pokračuje ve směru č. 1.

Posledním dílkem algoritmus končí a začíná kontrola, jestli dílek patří na místo, kam právě ukazuje. Každý dílek je kontrolován takovým způsobem, že pokud patří na dané místo, obsahuje na něj odkaz ještě jiný dílek kromě dílku, na který je ukazováno. Pro názornost je tento algoritmus vysvětlen na obrázku 2.31.

Pro názornost je kontrolován dílek č. 4, který obsahuje odkaz na dílky č. 5 (ve směru 1) a 12 (ve směru 2). Kontroluje se odkaz na dílek 5: obsahuje dílek ležící ve směru 2 (dílek č. 1) odkaz na další dílek v opačném směru než 1? Pokud ano, obsahuje tento dílek ve směru opačném ke směru 2 odkaz na dílek č. 4? V případě, že je odpověď kladná, je odkaz mezi dílky 4 a 5 platný. Pokud by byla odpověď záporná, odkazy se smažou.



Obr. 2.31: Ilustrační obrázek pro kontrolu umístění dílku.

#### 2.6.4 Složení puzzle

Po dokončení všech operací zmíněných v předchozích kapitolách dochází ke složení puzzle. Minulé algoritmy vytvořily odkazy mezi dílky a tyto odkazy je třeba zobrazit.

Nejprve probíhá hledání levého horního rohu (dítku, který má vlevo a nahoře hranu). Po jeho nalezení probíhá nejprve směrem doprava, a potom směrem dolů součet délek hran dílků. Tím je získán rozměr výsledného obrázku. Ve chvíli, kdy jsou nalezeny rozměry výsledného obrázku, jsou dílky procházeny znovu. Nyní jsou do popisu každého dítku ukládány souřadnice ve výsledném obrázku. Při vykreslování si algoritmus zjistí počáteční pozici a vykreslí pouze oblast, ve které se nachází obrázek.

Výsledek je zobrazen na obrázku 2.32.



Obr. 2.32: Složené puzzle.

Na obrázku je vidět, že dílky k sobě správně nepřiléhají. Tato skutečnost je dána tím, že nalezené rohy neleží naprosto přesně na skutečných rozích dílku. Při předzpracování se také mohou vyhodnotit některé pixely na okrajích jako patřící k pozadí. Z tohoto důvodu jsou dílky v některých místech mírně posunuty.



### 2.6.5 Složení puzzle popsaného podle barvy

V průběhu řešení diplomové práce byl testován algoritmus na ideálním případě – fotografii rozdělené na devět dílků. Vzhledem k tomu, že se jednalo o obdélníky, nebylo nutné popisovat tvar dílků. Díky tvaru dílků se velmi zjednodušilo výsledné skládání – nebylo třeba hledat rohy dílků.

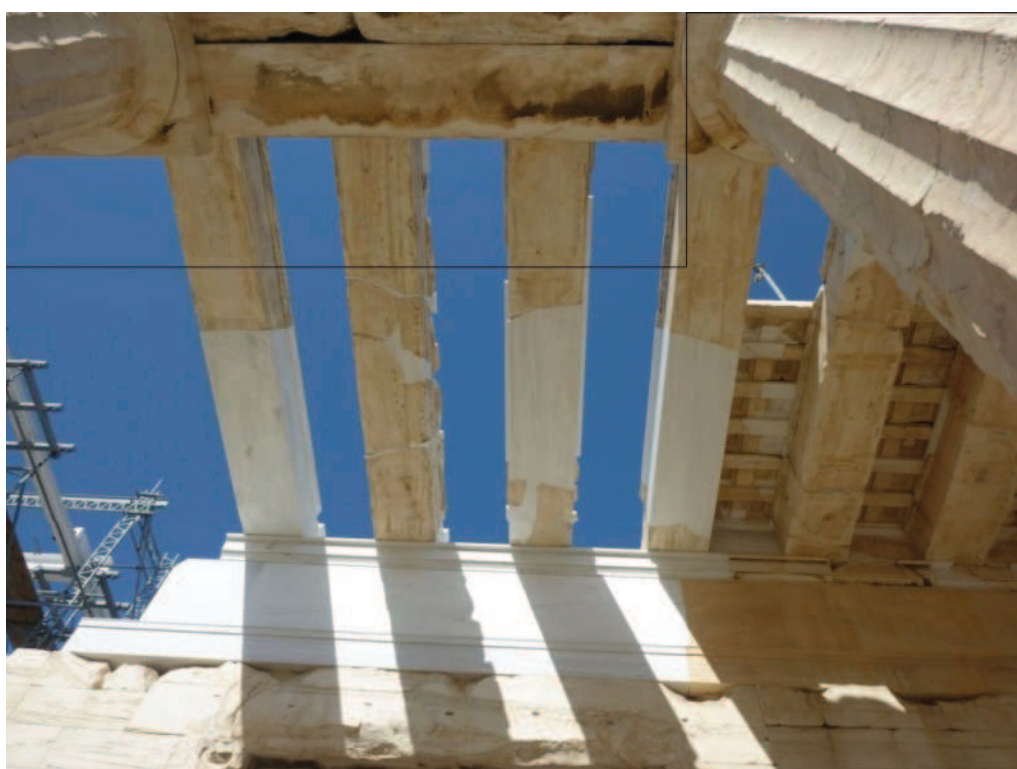
Původní rozřezaná a opět složená fotografie je na obrázcích 2.33, 2.34 a 2.35.



Obr. 2.33: Původní fotografie, která byla rozřezána a následně složena.



Obr. 2.34: Rozdělená fotografie určená ke složení.



Obr. 2.35: Fotografie složená z dílků.



## 2.7 Shrnutí

Druhá část diplomové práce pojednává o metodách pro vyříznutí dílku ze vstupního obrazu, jejich otočení, barevném a tvarovém popisu a nakonec o složení nalezených dílků k sobě.

Na začátku je uvedena scéna zvolená pro pořizování dílků. Zvolená scéna nevyžaduje stativ, protože vstupní obrázek obsahuje kalibrační body, s jejichž pomocí je možno zbavit se zkreslení způsobeného perspektivou. Následně je pomocí cyklického spouštění algoritmu K-means odstraněno pozadí. Nalezený dílek se otočí a vyřízne.

Po vyříznutí je dílek popsán podle barev – k tomuto účelu jsou použity popisy podle barvy na okraji dílků a nalezení nejdůležitějších barev uvnitř všech dílků.

Další použitou vlastností dílku je jeho tvar – zejména normálové vektory a vlastnosti zámků (díra, výstupek, hrana).

Předposlední částí je nalezení nejlépe se hodících dílků. To v prvním kroku nalezne několik nejlepších shod, mezi nimiž se rozhoduje o jejich platnosti.

Posledním krokem je složení dílku – zobrazení nalezených a potvrzených odkazů.

### 3 VÝSLEDKY STUDENTSKÉ PRÁCE

V poslední kapitole diplomové práce je uvedeno ovládání programu a také podmínky, které je nutné pro úspěšné spuštění programu splnit.

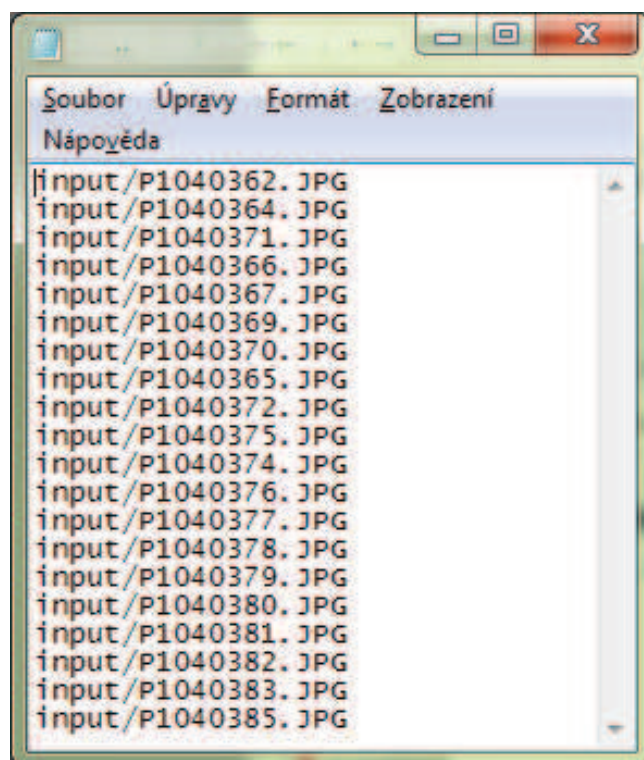
#### 3.1 Základní požadavky

Program byl vytvořen pomocí knihoven OpenCV v systému Windows 7 Professional a na jiném systému nebyl testován. Jako vývojové prostředí bylo použito Visual Studio. Pro spuštění programu je nutné mít nainstalovány knihovny OpenCV ve svém počítači.

#### 3.2 Běh a ovládání programu

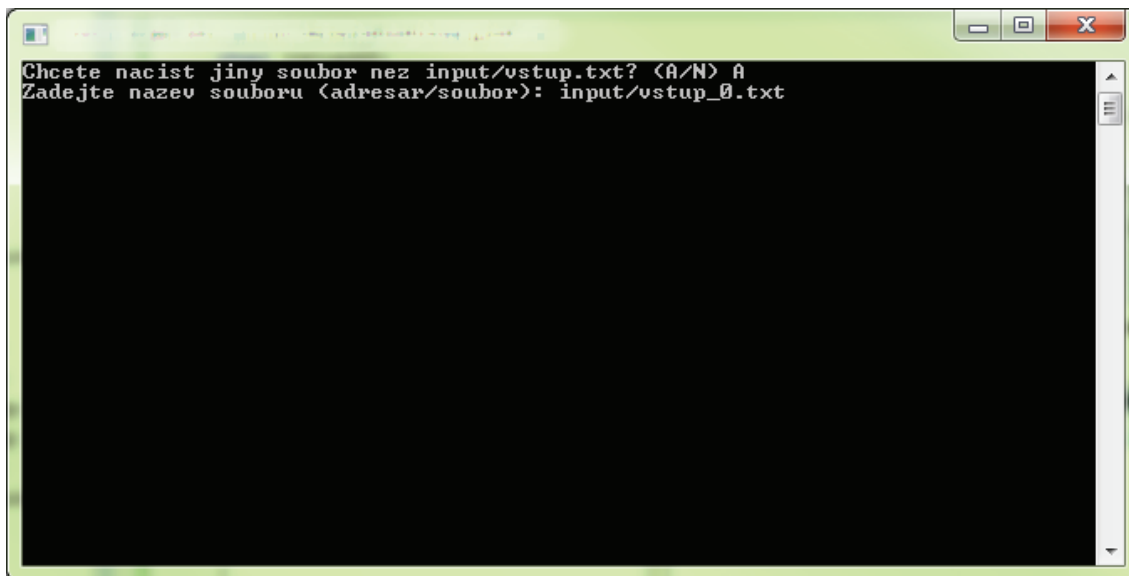
Vzhledem k tomu, že práce neobsahuje GUI, jsou vstupní data načítány z konzole, na kterou je také vypisován postup zpracování.

Program vyžaduje jediný vstupní soubor obsahující cestu k fotografiím všech dílků puzzle (obr. 3.1).



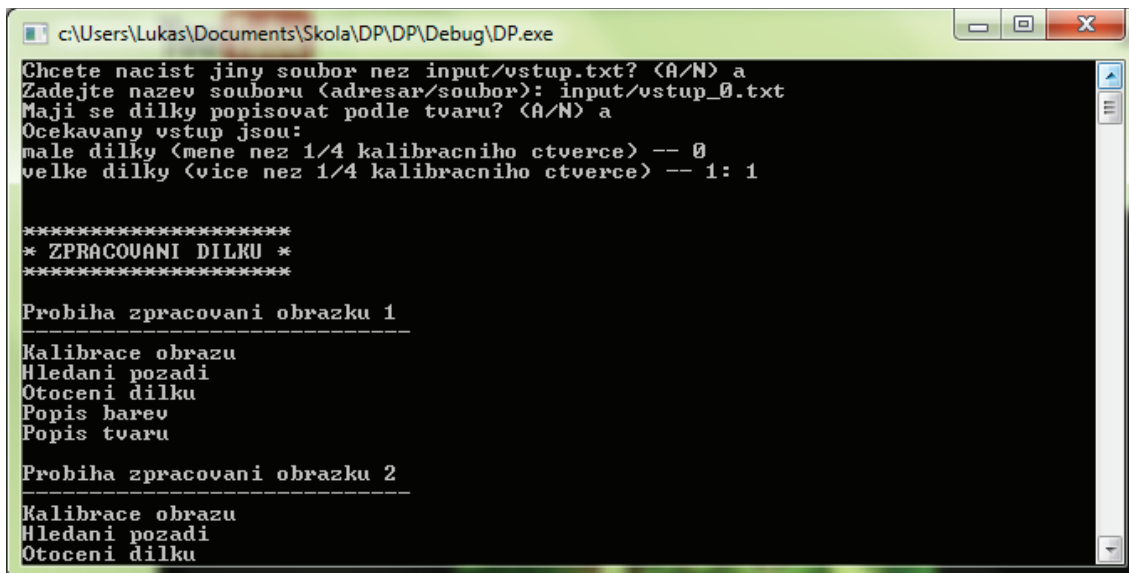
Obr. 3.1: Obsah vstupního souboru obsahující cestu ke všem fotografiím dílků.

Pro jednoduché testování se program nejprve dotáže, zda se má načítat jiný soubor než *input/vstup.txt*. Pokud je zvoleno ano (písmeny *a*, *A*, *y*, *Y*), je požadován název vstupního souboru (obrázek 3.2).



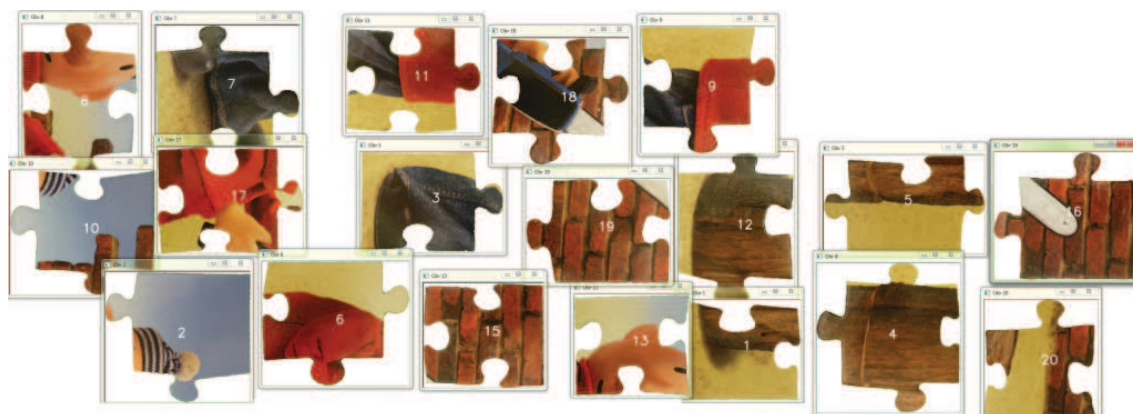
Obr. 3.2: Ovládání programu z konzole.

Program zároveň vypisuje průběh zpracování do konzole:

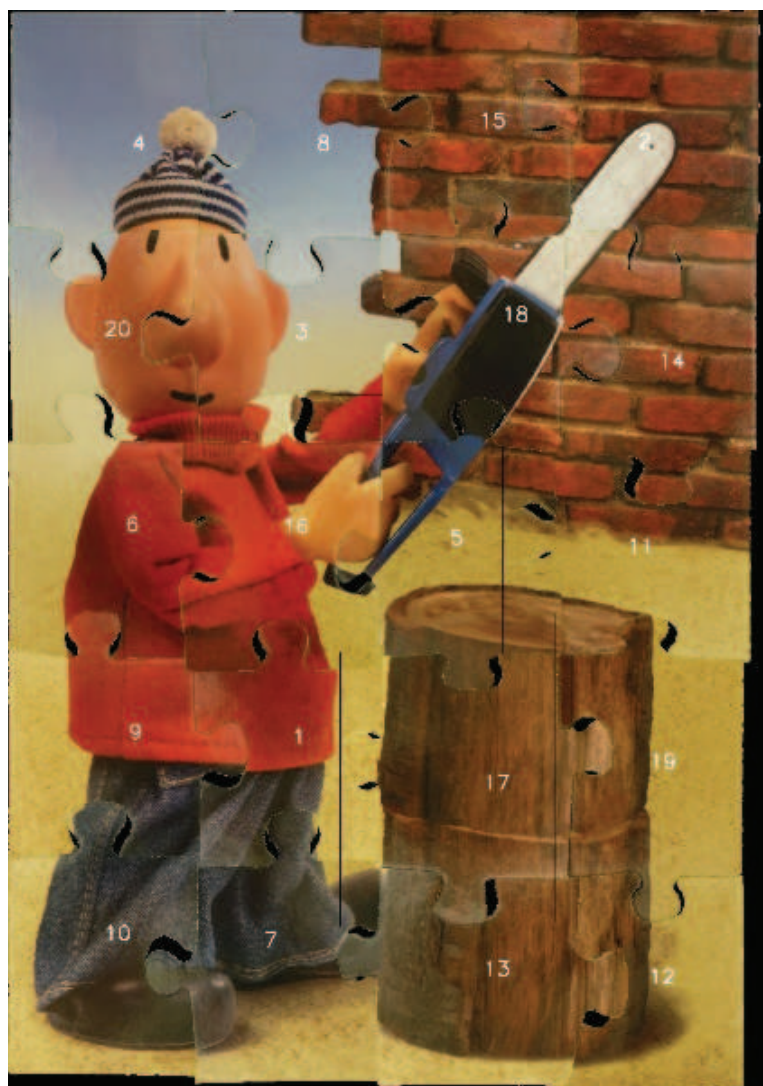


Obr. 3.3: Výpis informací do konzole.

Po zpracování všech obrázků probíhá skládání puzzle. Po dokončení ověřování platnosti dílků se objeví výstup – zvlášť nalezené dílky podle toho, jak byly načítány a složené puzzle (obrázky 3.4, 3.5).



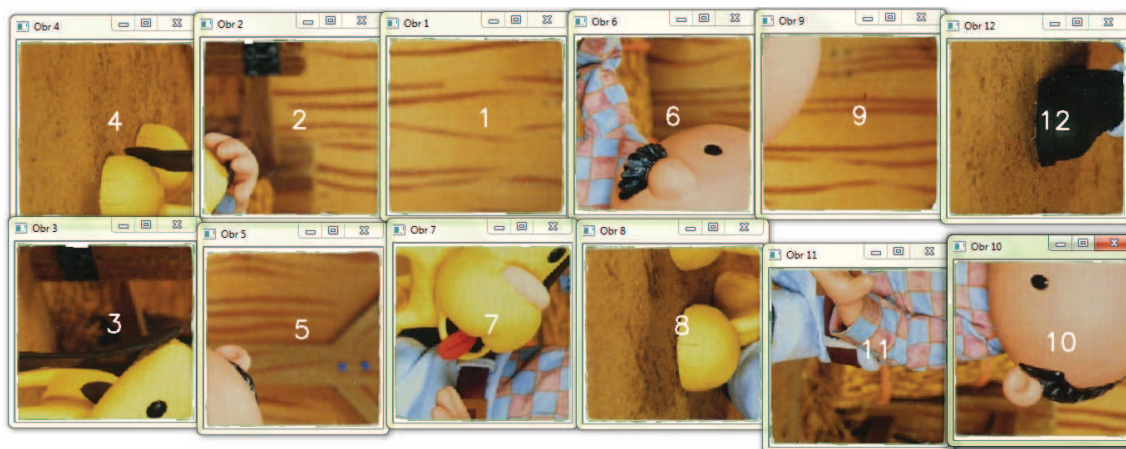
Obr. 3.4: Rozložené puzzle.



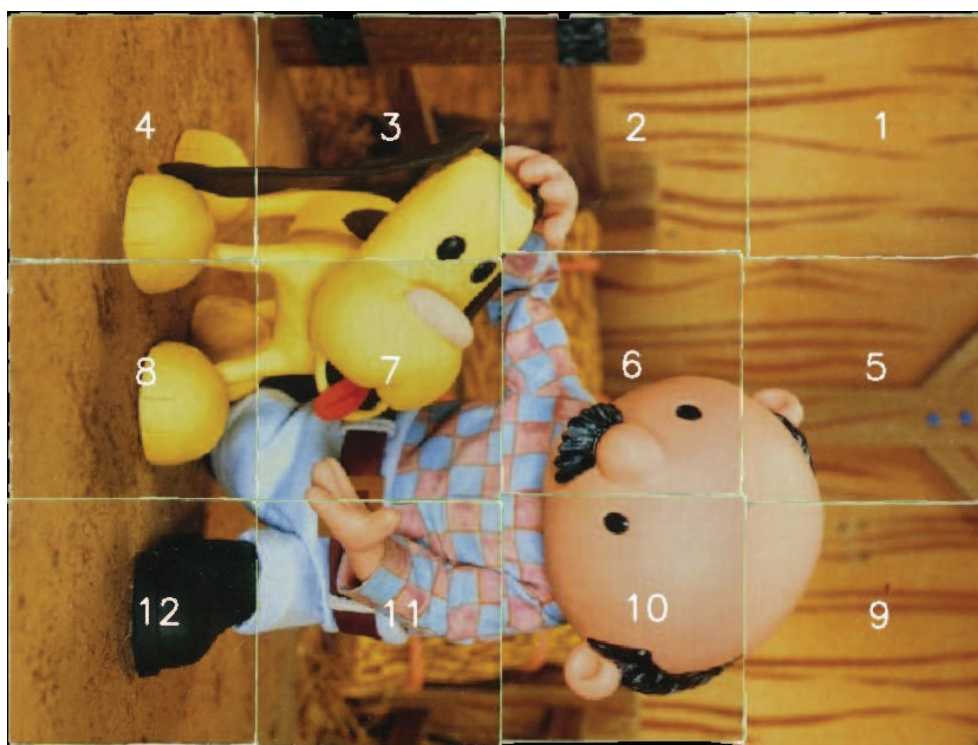
Obr. 3.5: Složené puzzle.



Program dokáže složit také puzzle bez použití tvarů. Pro otestování byly využity dětské kostky (výsledek je vidět na obrázcích 3.6 a 3.7).



Obr. 3.6: Rozložené puzzle popsané podle barev.



Obr. 3.7: Složené puzzle pomocí popisu barev.

## 4 ZÁVĚR

Předložená diplomová práce si kladla za cíl vytvořit algoritmus pro skládání puzzle. Hlavní cíl diplomové práce byl v zásadě splněn. Jediným problémem je zobrazení výsledného složeného obrázku. Dílky jsou v něm odchýleny od ideální pozice, a to hlavně z důvodu nepřesného hledání hran.

V prvním semestrálním projektu, na který předložená práce navazuje, byla zvolena scéna, na níž bylo možné fotit větší množství dílků. Jejimi nevýhodami však bylo vrhání stínů jednotlivými ležícími dílky a obtížné použití kalibračních bodů. Z tohoto důvodu byla scéna změněna – dílky se fotí samostatně, ve vyvýšené pozici. Vyvýšené místo pro focení bylo zvoleno s ohledem na minimalizaci stínů vrhaných dílkem. Snížení počtu focených dílků na jeden mělo odůvodnění v jednodušším hledání kalibračních značek a snadnějším hledání dílku v obraze.

Při předzpracování dílků dochází nejprve k odstranění zkreslení – nejprve vad objektivu a poté perspektivy, jejíž odstranění probíhá pomocí matice homogenní transformace.

Po odstranění zkreslení je v obraze hledáno zkreslení. V porovnání se semestrálním projektem se algoritmus opět rapidně změnil. Zatímco dříve bylo hledáno maximum histogramu, nyní je použit algoritmus K-means, který je univerzálnější. Jeho univerzálnost je bohužel vyvážena dobou zpracování, která je oproti metodě hledání histogramu výrazně delší.

Otočený dílek je popsán pomocí tvaru a barvy – použit je popis barvy na okrajích a jsou také nalezeny hlavní barvy vyskytující se na všech dílcích. Tvar je popsán normálovými vektory a typem (a velikostí) zámku.

Popsané dílky jsou nakonec vzájemně porovnávány. Nejlépe shodující se dílky jsou podrobeny zkoušce na platnost této shody, následně otočeny a složeny.

Program má jediný vstupní soubor, který obsahuje cestu ke všem fotografiím dílků. O postupu zpracování je uživatel informován přes konzolové rozhraní.

Zadání diplomové práce se podařilo splnit – program je schopen vstupní obrázky složit. Výstupem je jednak složené puzzle, jednak soubor vypisující, které dílky k sobě patří.

Program je schopen skládat jednak pomocí pouze barevného popisu, jednak pomocí popisu tvaru i hran.

## LITERATURA

- [1] KANUNGO, T., MOUNT, D., NETANYAHU, N., PIATKO, C., SILVERMAN, R., WU, A. An Efficient K-Means Clustering Algorithm: Analysis and Implementation. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* [online]. 2002, roč. 24, č. 7. [cit. 2012-04-28] Dostupné z URL: <<http://www.cs.umd.edu/~mount/Papers/pami02.pdf>>.
- [2] *BSD licences* [online]. poslední aktualizace 23. 4. 2012 [cit. 2012-04-28]. Dostupné z URL: <[http://en.wikipedia.org/wiki/BSD\\_licenses](http://en.wikipedia.org/wiki/BSD_licenses)>.
- [3] *Canny edge detector* [online]. poslední aktualizace 24. 4. 2012 [cit. 2012-04-28]. Dostupné z URL: <[http://en.wikipedia.org/wiki/Canny\\_edge\\_detector](http://en.wikipedia.org/wiki/Canny_edge_detector)>.
- [4] *Chroma key* [online]. poslední aktualizace 25. 4. 2012 [citováno 2012-04-28]. Dostupné z URL: <[http://en.wikipedia.org/wiki/Chroma\\_key](http://en.wikipedia.org/wiki/Chroma_key)>.
- [5] *CIE 1931 color space* [online]. poslední aktualizace 16. 4. 2012 [cit. 2012-04-28]. Dostupné z URL: <[http://en.wikipedia.org/wiki/CIE\\_1931\\_color\\_space](http://en.wikipedia.org/wiki/CIE_1931_color_space)>.
- [6] *CMYK color model* [online]. poslední aktualizace 19. 4. 2012 [cit. 2012-04-29]. Dostupné z URL: <[http://en.wikipedia.org/wiki/CMYK\\_color\\_model](http://en.wikipedia.org/wiki/CMYK_color_model)>.
- [7] *Color* [online]. poslední aktualizace 23. 4. 2012 [cit. 2012-04-28]. Dostupné z URL: <<http://en.wikipedia.org/wiki/Color>>.
- [8] *Gamut* [online]. poslední aktualizace 25. 3. 2012 [cit. 2012-04-28]. Dostupné z URL: <<http://en.wikipedia.org/wiki/Gamut>>.
- [9] WHITAKER, Ross T. *Geometric Transformations and Image Warping* [online]. (přednáška) Utah: SCI Institute, School of Computing University of Utah, 2009. [cit. 2012-04-29]. Dostupné z URL: <[http://www.coe.utah.edu/~cs6640/geometric\\_trans.pdf](http://www.coe.utah.edu/~cs6640/geometric_trans.pdf)>.
- [10] *HSV* [online]. poslední aktualizace 22. 3. 2012 [cit. 2012-04-29]. Dostupné z URL: <<http://cs.wikipedia.org/wiki/HSV>>.
- [11] BRADSKI, Gary a Adrian KAEHLER. *Learning OpenCV: Computer Vision with the OpenCV Library*. Sebastopol[USA]: O'Reilly Media, 2008. ISBN 978-0-596-51613-0.
- [12] *Optical Aberrations: Distortion* [online]. 2011. [cit. 2012-05-07]. Dostupné z URL: <[http://starizona.com/acb/basics/equip\\_optics101\\_distortion.aspx](http://starizona.com/acb/basics/equip_optics101_distortion.aspx)>.

- [13] HLAVÁČ, Václav a Milan ŠONKA. *Počítačové vidění*. Praha: Grada, 1992. ISBN 80-85424-67-3.
- [14] HORÁK, K., KALOVÁ, I., PETYOVSKÝ, P., RICHTER, M. *Počítačové vidění*. Brno: FEKT VUT Brno, 2007.
- [15] *RGB color model* [online]. poslední aktualizace 25. 4. 2012 [cit. 2012-04-28]. Dostupné z URL: <[http://en.wikipedia.org/wiki/RGB\\_color\\_model](http://en.wikipedia.org/wiki/RGB_color_model)>.
- [16] *YCbCr* [online]. poslední aktualizace 16. 4. 2012 [cit. 2012-04-29]. Dostupné z URL: <<http://en.wikipedia.org/wiki/YCbCr>>.



# SEZNAM PŘÍLOH

<b>A</b>	<b>Vývojové diagramy</b>	<b>73</b>
<b>B</b>	<b>Databáze obrázků</b>	<b>77</b>
B.1	Funkční obrázky popisované pomocí barvy i tvaru . . . . .	77
B.2	Funkční obrázky popisované pouze podle barvy . . . . .	87
B.3	Nefunkční obrázky . . . . .	93