



Semestrální práce z KIV/PC  
PŘEBARVOVÁNÍ SOUVISLÝCH OBLASTÍ  
VE SNÍMKU

Lukáš Runt (A20B0226P)

29. prosince 2021

# Obsah

<b>Obsah</b>	<b>1</b>
<b>1 Zadání</b>	<b>2</b>
<b>2 Analýza úlohy</b>	<b>3</b>
2.1 Formát PGM souboru . . . . .	3
2.2 Algoritmus Přebarvování souvislých oblastí . . . . .	3
2.2.1 První průchod . . . . .	3
2.2.2 Druhý průchod . . . . .	4
2.3 Zaznamenání kolizí . . . . .	4
2.3.1 Množina . . . . .	5
2.3.2 Spojový seznam . . . . .	5
2.3.3 Pole . . . . .	5
<b>3 Popis implementace</b>	<b>5</b>
<b>4 Uživatelská příručka</b>	<b>7</b>
4.1 Sestavení . . . . .	7
4.2 Spuštění . . . . .	7
<b>5 Závěr</b>	<b>8</b>

# 1 Zadání

Naprogramujte v ANSI C přenositelnou **konzolovou aplikaci**, která provede v binárním digitálním obrázku (tj. obsahuje jen černé a bílé body) **obarvení souvislých oblastí** pomocí níže uvedeného algoritmu *Connected-Component Labeling* z oboru počítačového vidění. Vaším úkolem je tedy implementace tohoto algoritmu a funkcí rozhraní (tj. načítání a ukládání obrázku, apod.). Program se bude spouštět příkazem

```
ccl.exe <input-file[.pgm]> <output-file>
```

Symbol `<input-file>` zastupuje jméno vstupního souboru s binárním obrázkem ve formátu *Portable Gray Map*, přípona souboru nemusí být uvedena; pokud uvedena není, předpokládejte, že má soubor příponu `.pgm` Symbol `<output-file>` zastupuje jméno výstupního souboru s obarveným obrázkem, který vytvoří vaše aplikace. Program tedy může být během testování spuštěn například takto:

```
... \>ccl.exe e:\images\img-inp-01.pgm e:\images\img-res-01.pgm
```

Úkolem programu je tedy vytvořit výsledný souboru s obarveným obrázkem uvedeným umístění a s uvedeným jménem. Vstupní i výstupní obrázek bude uložen ve formátu PGM.

Testujte, zda je vstupní obraz skutečně černobílý. Musí obsahovat pouze pixely s hodnotou `0x00` a `0xFF`. Pokud tomu tak není, vypište krátké chybové hlášení (anglicky) a oznamte chybu operačnímu prostředí pomocí nenulového návratového kódu.

Hotovou práci odevzdejte v jediném archivu typu ZIP prostřednictvím automatického odevzdávacího a validačního systému. Postupujte podle instrukcí uvedených na webu předmětu. Archiv nechť obsahuje všechny zdrojové soubory potřebné k přeložení programu, **makefile** pro Windows i Linux (pro překlad v Linuxu připravte soubor pojmenovaný **makefile** a pro Windows **makefile.win**) a dokumentaci ve formátu PDF vytvořenou v typografickém systému  $\text{\TeX}$ , resp.  $\text{\LaTeX}$ . Bude-li některá z částí chybět, kontrolní skript Vaši práci odmítne.

Celé zadání je dostupné na adrese: <http://www.kiv.zcu.cz/studies/predmety/pc/data/works/sw2021-02.pdf>

## 2 Analýza úlohy

Zadaný problém lze rozdělit na několik dílčích podproblémů, a to na kontrolu vstupních argumentů, načítání a vytváření pgm souboru a následně algoritmus, který přebarví souvislé oblasti.

### 2.1 Formát PGM souboru

Na vstupu a výstupu programu je soubor ve formátu PGM. Formát tohoto souboru je následující:

<b>P5</b>	Každý PGM soubor obsahuje z "Magické
<b>1024 768</b>	číslo" pro identifikaci typu souboru (v našem
<b>255</b>	případě 'P5', tedy binární šedotónová data),
.....	bílý znak (může být mezera, TAB, CR, LF
.....	...), šířku (počet sloupců, řetězcem znaků),
	bílý znak, výšku (počet řádků, řetězcem
	znaků), bílý znak, index nejvyšší hodnoty
	šedé (řetězcem znaků), bílý znak, byty (v
	jazyce C datový typ <code>unsigned char</code> ) před-

Obrázek 1: PGM soubor

stavující hodnoty jednotlivých pixelů.

### 2.2 Algoritmus Přebarvování souvislých oblastí

Přebarvování souvislých oblastí (Connected-Component Labeling, CCL) je dvouprůchodový algoritmus z oblasti počítačového vidění. Jeho vstupem je binární obrázek (takový který obsahuje jen černé a bílé pixely). Bílé pixely představují objekty, které ten se tento algoritmus pokouší izolovat a označit různými barvami, resp. hodnotami intenzity šedé barvy (tedy tzv. labels, česky štítky, značkami). Černé pixely pak představují pozadí.

#### 2.2.1 První průchod

Procházíme obrázek po řádcích. Každému nenulovému pixelu ne souřadnicích  $[i, j]$  přiřadíme hodnotu podle hodnoty jeho sousedních pixelů, pokud nenulové sousední pixely existují (poloha sousedů je daná maskou na obrázku č. 1). Všechny sousední pixely dané maskou již byly obarveny v předchozích krocích (to je zajištěno tvarem masky).

- Jsou-li všechny sousední pixely součástí pozadí (mají hodnotu 0x00), přiřadíme pixelu  $[i, j]$  dosud nepřidělenou hodnotu, nebo-li novou barvu.

- Má-li právě jeden ze sousedních pixelů nenulovou hodnotu, přiřaíme obarvovanému pixelu hodnotu nenulového sousedního pixelu.
- Je-li více sousedních pixelů nenulových, přiřadíme obarvovanému pixelu hodnotu kteréhokoli nenulového pixelu ze zkoumaného okolí. Pokud byly hodnoty pixelů v masce různé (došlo k tzv. *kolizi barev*), zaznamenáme ekvivalenci dvojic do zvláštní datové struktury - tabulky ekvivalence barev.



Obrázek 2: Maska pro přebarvování

### 2.2.2 Druhý průchod

Po prvním průchodu celého obrázku jsou všechny pixely náležející oblastem (objektům) obarveny, některé oblasti jsou však obarveny více barvami (díky kolizím). Proto musíme znova projít celý obrázek a podle informací o ekvivalenci barev (z tabulky ekvivalence barev získané v průběhu 1. průchodu) přebarvíme pixely těchto oblastí. Z množiny kolizních barev je možné nějak vybrat jednu (první, poslední, náhodnou) nebo opět postupovat při přiřazování barev "od začátku" a jako novou barvu použít index množiny (nesmí být samozřejmě nulový, protože barva 0x00 představuje pozadí).

Po tomto kroku odpovídá každé oblasti označení (obarvení) jedinou, v jiné oblasti se nevyskytující hodnotou (barvou).

## 2.3 Zaznamenání kolizí

Při prvním průchodu je potřeba si zaznamenávat kolize, pokud je přítomno v masce více rozdílných barev. Tyto kolize lze zaznamenávat pomocí několika níže uvedených metod.

### 2.3.1 Množina

Množina je abstraktní datový typ, který je schopen uložit určité hodnoty bez jakéhokoliv pořadí a bez opakujících se hodnot. Narozdíl od pole jsou množiny neuspořádané a neindexované. V informatice je teorie množin užitečná pokud potřebujeme shromáždit data a nezáleží nám na jejich násobnosti nebo pořadí, což je právě náš případ. V případě kolize by se prvky přidávali do množin a při druhém průchodu by se pak bral nejmenší prvek.

### 2.3.2 Spojový seznam

Spojový seznam je dynamická datová struktura, určená k ukládání dat předem neznámé délky. Základní stavební jednotkou spojového seznamu je uzel, který vždy obsahuje ukládanou hodnotu a ukazatel na následující prvek. V našem případě by toto šlo implementovat tak, že by jeden ukazatel měl více pointerů a nebo by vždy při kolizi vložil prvky do svého seznamu pokud by ještě prvky v seznamu nebyly.

### 2.3.3 Pole

Tento způsob je velice podobný způsobu se spojovými seznamy. Jedná se o pole, kde každý index, kromě indexu 0, znamená číslo barvy. Obsah indexu pak znamená v případě, že není rovno -1, ukazatel na další barvu (index), se kterou je v kolizi. Pokud je hodnota na indexu rovna hodnotě -1, pak neexistuje žádná další kolize s "menší" barvou.

## 3 Popis implementace

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed purus tellus. Ut faucibus, nibh eu convallis malesuada, neque magna convallis augue, nec vehicula risus felis ac urna. Nunc porttitor est eu erat iaculis, in pharetra elit consequat. Nulla nibh massa, facilisis in efficitur ut, suscipit et massa. Nunc id tristique velit, eu rhoncus leo. Praesent lobortis est eu lacus vulputate suscipit. Proin nec dictum felis, et auctor mi. Praesent justo diam, finibus ac neque ut, vestibulum maximus ligula. Nullam feugiat imperdiet gravida.

Integer magna lectus, faucibus sit amet sem non, dapibus cursus ligula. Nunc nibh dolor, fringilla ut tortor sit amet, iaculis malesuada dolor. Cras vitae ante purus. Integer consectetur, diam nec sagittis egestas, dolor nisi vestibulum risus, vel tincidunt neque tellus quis odio. Donec feugiat non magna ut lacinia. Suspendisse eu risus mollis, ultricies nulla ac, egestas orci.

Donec convallis quis massa ut ullamcorper. In hac habitasse platea dictumst. In eleifend metus et sem consectetur, vitae ornare velit accumsan. Phasellus faucibus justo quis commodo luctus. Donec mollis purus in ex rhoncus, id ullamcorper eros euismod. Pellentesque ac ante consequat, interdum orci in, sollicitudin massa. Morbi non tincidunt quam. Ut euismod consectetur ante eget imperdiet. Nam et suscipit justo.

Sed consequat nibh vitae nisl hendrerit, pulvinar hendrerit tellus suscipit. Nullam sed nibh ut felis interdum viverra ac vel erat. Suspendisse diam odio, ultrices et enim mollis, porttitor condimentum justo. Ut sapien enim, viverra at maximus eu, hendrerit non tellus. Interdum et malesuada fames ac ante ipsum primis in faucibus. Sed metus mi, convallis sit amet metus quis, rutrum malesuada eros. Etiam mattis mauris arcu, ac consectetur metus tincidunt non. Aenean finibus augue id ornare semper. Aliquam erat volutpat. Donec et ullamcorper sapien, in venenatis sem. Mauris maximus massa at urna auctor hendrerit. Proin felis elit, vehicula at nisi in, imperdiet sodales ipsum. Praesent eget odio ut nunc gravida feugiat vitae non lectus.

Donec diam libero, rutrum eget mauris in, commodo dapibus sem. Ut suscipit cursus dapibus. Curabitur et pharetra risus. Donec luctus ipsum at metus ullamcorper eleifend. Praesent molestie commodo augue quis tincidunt. Donec eget ligula gravida, tristique velit eget, congue ligula. Nullam ut maximus augue. Nulla eleifend mauris eu vulputate sollicitudin. Phasellus quis tristique ex. Sed viverra odio sit amet dictum facilisis. In fermentum consectetur quam non dapibus.

Sed consequat nibh vitae nisl hendrerit, pulvinar hendrerit tellus suscipit. Nullam sed nibh ut felis interdum viverra ac vel erat. Suspendisse diam odio, ultrices et enim mollis, porttitor condimentum justo. Ut sapien enim, viverra at maximus eu, hendrerit non tellus. Interdum et malesuada fames ac ante ipsum primis in faucibus. Sed metus mi, convallis sit amet metus quis, rutrum malesuada eros. Etiam mattis mauris arcu, ac consectetur metus tincidunt non. Aenean finibus augue id ornare semper. Aliquam erat volutpat. Donec et ullamcorper sapien, in venenatis sem. Mauris maximus massa at urna auctor hendrerit. Proin felis elit, vehicula at nisi in, imperdiet sodales ipsum. Praesent eget odio ut nunc gravida feugiat vitae non lectus.

Donec diam libero, rutrum eget mauris in, commodo dapibus sem. Ut suscipit cursus dapibus. Curabitur et pharetra risus. Donec luctus ipsum at metus ullamcorper eleifend. Praesent molestie commodo augue quis tincidunt. Donec eget ligula gravida, tristique velit eget, congue ligula. Nullam ut maximus augue. Nulla eleifend mauris eu vulputate sollicitudin. Phasellus quis tristique ex. Sed viverra odio sit amet dictum facilisis. In fermentum consectetur quam non dapibus.

## 4 Uživatelská příručka

### 4.1 Sestavení

Pro snadné sestavení je připraven `makefile`, který funguje na operačním systému Linux a `makefile.win`, který funguje na operačním systému Windows. Překlad lze provést příkazem `make -f makefile.win`. Příkazy pro přeložení se mohou lišit podle použitého překladače a nastavení.

### 4.2 Spuštění

Program lze spustit pomocí příkazu:

```
ccl.exe <input-file[.pgm]> <output-file>
```

Kde `input-file` je cesta k pgm souboru, který obsahuje jen hodnoty `0x00` a `0xFF`. Při spuštění se může soubor zadat s příponou `.pgm` i bez. `output-file` pak obsahuje cestu a jméno výstupního souboru, který se může zadat s příponou `.pgm` i bez.

```
Microsoft Windows [Version 10.0.19043.1415]  
(c) Microsoft Corporation. Všechna práva vyhrazena.  
  
D:\PC\Semestrální práce>ccl.exe D:\PC\sw2021-02-data\w2test.pgm obarvenyObraz.pgm
```

Obrázek 3: Příklad spuštění

Formát vstupního i výstupního souboru je typu `pgm` (formát je detailně popsán viz.2.1). Přebarvení obrázku by mělo vypadat následovně:



Obrázek 4: Příklad vstupu



Obrázek 5: Příklad výstupu



## 5 Závěr

Celkovou práci hodnotím pozitivně, neboť jsem si vyzkoušel práci s jazykem C. Byl to pro mne nepopsatelný zážitek, který mě studijně obohatil a posunul o krok blíže k praktickým aplikacím teoreticky získaných vědomostí. Nyní si bez Cčka nedokážu představit svůj život. Díky jazyku C mám prstoklad na klávesnici, jak kdybych hrál na klavír. Celou noc jsem programoval, byl jsem jako netopýr. Když se mne někdo zeptá na znalost jazyků řeknu, že umím v Cčku, javě, sql a basicu. Odedneška už na nebi svítí poitery místo hvězd. Už se nemůžu dočkat, jak si zase něco v tomto úžasném jazyce zase něco naprogramuji.