

Načítání a ukládání obrazu

Projekt k předmětu ZPO

Vojtěch Šimetka xsimet00

Adam Široký xsirok07

Klára Vaňková xskota05

9. května 2014

1 Zadání

Zpracujte funkci pro načítání a ukládání obrazu ve vhodném rastrovém formátu (ne ve formátech .JPG, .BMP, nekomprimovaný .TIF, které jsou k dispozici v knihovně DigILib, zájem naopak je o formáty .GIF, .PCX, .PNG, .TIF s kompresí apod.). Funkci pokud možno zpracujte tak, aby nebyla závislá na jiných knihovnách. Měl by být podporován formát grayscale (256 úrovní šedi - 8 bitů/pixel) a RGB (24 bitů/pixel).

Ke zpracování jsme si zvolili formát .GIF s kompresí.

2 O programu

Program je implementován v programovacím jazyce C++ a tedy pro překlad stačí zadat příkaz *make*. Pro úspěšný překlad je vyžadován překladač g++ a knihovna OpenCV. Program je ovládán čistě konzolovými příkazy a až na zobrazení výstupu nemá žádné grafické uživatelské rozhraní.

Spuštění

`./ImageConverter [vstupní soubor] [možnosti] [výstupní typy]`

[vstupní soubor]

-f Vstupní soubor podporující formáty: bmp, dib, jpeg, jpg, jpe, jp2, png, pbm, pgm, ppm, sr, ras, tiff, tif, gif

[možnosti]

-s x y Změna velikosti v %

-r width height Změna velikosti jako absolutní hodnoty

-d Zobrazení výstupního obrázku

-g Převod do odstínů šedi

-o folder Výstupní složka

[výstupní typy]

bmp, dib Windows bitmaps

jpeg, jpg, jpe JPEG format

jp2 JPEG 2000 format

png Portable Network Graphics

pbm, pgm, ppm Portable image format

sr, ras Sun rasters

tiff, tif TIFF format

gif GIF format

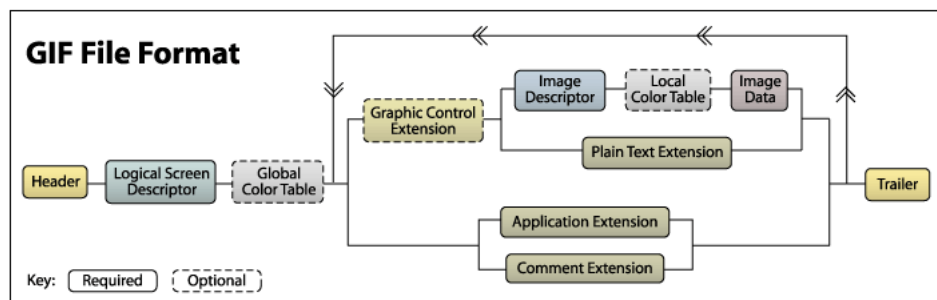
Varování: Na některých Linuxových a BSD systémech může být pro načítání a ukládání některých obrazových formátů nutno doinstalovat externí knihovny: libjpeg, libpng, libtiff a libjasper.

Omezení programu

Program umí načítat soubory ve formátu GIF pouze při 8bitové barevné paletě na každý kanál. Výstupní GIF soubor podporuje pouze 256 barev a v některých případech dochází k posunu pixelů.

3 Formát GIF

Formát GIF je složen z několika rozdílných bloků dat, které jsou ukázány na obrázku 1.



Obrázek 1: Formát GIF

Hlavička

Prvním blokem GIFu je hlavička, která zabírá 6 bytů souboru. První tři byty jsou nazývány *signature*. Vždy by měly mít hodnotu "GIF", tzn. "47 49 46". Další tři byty určují **verzi normy** kódování obrázku, známe verze "89a" a "87a".

Globální deskriptor obrazovky

Druhý blok zabírá 8 bytů. Na prvních dvou bytech je uvedena **šířka** obrazovky v pixelech, na dalších dvou bytech je **výška** obrazovky. V 5. bytu je uloženo více informací: první bit udává, zda je použita **globální barвовá paleta**, 2.-4. bit určuje **počet bitů na pixel**, 5. bit říká, zda je barвовá paleta **setříděna** a 6. a 7. bit obsahují **délku** barвовé palety. 6. byte obsahuje

index barvy, která je použita jako pozadí. V posledním bytu je uveden **poměr mezi výškou a šířkou** pixelu.

Globální barvová paleta

Paleta obsahuje **seznam všech barev**, které mohou být v obrázku. Každá barva je uložena jako hodnota RGB na třech bytech. Tento blok není ve formátu .GIF povinný.

Data - obrázek

První byte tohoto bloku je **minimální velikost kódu LZW**, kterým jsou data kódována. Následuje byte udávající **kolik bytů dat následuje** - hodnota od 0 do 255. Po přečtení všech těchto bytů dat následuje další byte informující o délce následujících dat. Takto se pokračuje až po byte udávající nulovou délku dat.

4 Dekompresi formátu GIF89a

Každý blok dat, je zpracován následujícím algoritmem:

1. Načtení počáteční hodnoty LZW.
2. Načtení velikosti bloku.
3. Inicializace slovníku.
4. Načtení kódového slova (dáno aktuální hodnotou LZW).
5. Je načtená hodnota rovna CC (*Clear code*)?
 - ANO - pokračuj bodem 6.
 - NE - skonči s chybou.
6. Načtení kódového slova.
7. Je načtená hodnota platným indexem do slovníku?
 - ANO - zpracuj tuto hodnotu a ulož ji jako *poslední načtené kódové slovo*.
 - NE - skonči s chybou.
8. Cyklus dokud nejsou zpracovány všechny pixely daného bloku dat.
 - (a) Načtení kódového slova.
 - i. Je načtená hodnota rovna CC (*Clear Code*)?
 - ANO - reinitializuj celý proces a přejdi na bod 8.

- NE - pokračuj na bod ii.
- ii. Je načtená hodnota rovna EOI (*End Of Information*)?
 - ANO - jsou zpracovány všechny pixely?
 - ANO - ukonči cyklus.
 - NE - skonči s chybou.
 - NE - pokračuj na bod iii.
- iii. Je načtená hodnota platným indexem do slovníku?
 - ANO - zpracuj posloupnost barev, která je dána lineárním seznamem barev ve slovníku na indexu, jenž je určen načteným kódovým slovem. Ulož první položku seznamu do proměnné K .
 - NE - zpracuj seznam barev ze slovníku na indexu, který je roven hodnotě *poslední kódové slovo*. Proměnná K je rovna prvnímu prvku tohoto seznamu. Zpracuj proměnnou K .
- iv. Vytvoř nový záznam do slovníku na první volnou položku ze seznamu na indexu *poslední kódové slovo* a proměnné K .
- (b) Hodnotu *poslední kódové slovo* nahraď hodnotou načteného kódového slova.

5 Algoritmus LZW

Formát GIF je kódován pomocí algoritmu Lempel–Ziv–Welch. Ten vyhledává opakující se posloupnosti pixelů a ty kóduje jako klíčové hodnoty. Výhodou LZW algoritmu je, že není potřeba ukládat slovník. Lze jej totiž odvodit ze vstupních dat.

1. Na výstup zapiš hodnotu CC
2. Zakóduj barvu současného pixelu ze vstupu a přidej ji do načtené posloupnosti
3. Obsahuje-li načtená posloupnost pouze jednu hodnotu pokračuj bodem 2
4. Vyhledej načtenou posloupnost ve slovníku
 - (a) Byla-li nalezena pokračuj bodem 2
 - (b) Přidej právě načtenou posloupnost do slovníku
 - i. Je-li velikost slovníku větší než 2^{12} , vymaž všechny záznamy ze slovníku a na výstup zapiš hodnotu CC
5. Do výstupního souboru ulož kód odpovídající načtené posloupnosti bez posledního prvku

6. Vymaž načtenou posloupnost, kromě posledního prvku
 - (a) Není-li vyčerpán vstup, pokračuj bodem 2
 - (b) Jinak vypiš kód poslední načtené posloupnosti, EOI a řádně ukonči algoritmus

Slovník

Slovník je implementován jako pole lineárních seznamů barev, kde každá barva je mimo jiné reprezentována strukturou třech osmi-bitových proměnných, jenž odpovídají barevnému modelu RGB. Zpracováním seznamu se rozumí uložení všech barev daného seznamu do výstupního pole, které je popsáno v následující části. Zpracování jedné položky je stejné jako zpracování seznamu délky jedna.