

Kódování a komprese dat

Komprese obrazových dat s využitím statického a adaptivního Huffmanova kódování

Bc. David Kolečkář, `xkolec07`
`xkolec07@stud.fit.vutbr.cz`

30. dubna 2019

1 Úvod

V této dokumentaci je popsána konzolová aplikace pro kompresi šedo tónových obrazových dat, kde se uplatní principy statického a adaptivního Huffmanova kódování. Projekt byl vytvořen v rámci projektu Kódování a komprese dat. Dokumentace obsahuje rozbor problému, popis implementace, návod k ovládání programu a vyhodnocení činnosti aplikace.

2 Rozbor a analýza problému

2.1 Statické:

Huffmanovo kódování je založeno na sestavení binárního stromu, ve kterém nejkratší cesta vede ke znakům s nejvyšší frekvencí (četností). Algoritmus si po přečtení vstupu spočítá četnosti (pravděpodobnosti) jednotlivých znaků ve vstupní abecedě. Na základě těchto četností poté vygeneruje binární strom, jehož hrany nesou hodnotu 0 nebo 1, a v jehož listech jsou znaky vstupní abecedy. Navíc platí, že čím vyšší má znak pravděpodobnost, tím blíže ke kořenu se ve stromu vyskytuje. Místo těchto znaků je do listů následně přiřazen bitový kód, který odpovídá zřetězení hodnot hran, kterými projdeme při cestě z kořene do daného listu. Tento strom, nebo alespoň informace o četnostech výskytů jednotlivých znaků ve vstupních datech, je také potřeba přiložit k výsledné posloupnosti, aby byla možná dekomprimace.

2.2 Adaptivní:

Nebylo implementováno

3 Popis implementace

Aplikace byla napsána v jazyce C++. Vstupem programu je v případě kódování obrázků ve formátu RAW, kde jednotlivé pixely nabývají 8 bitových hodnot odstínu šedi (0-255). Poté je spočítán histogram obrázku, který je nutný pro vytvoření huffmanova stromu. V případě, že je vyžadováno použití modelu je obrázek nejdříve předzpracován. Aplikace jako model využívá diferenci sousedních pixelů. Poté je vytvořen samotný huffmanův strom, struktura uzlu stromu je: hodnota odstínu šedi, četnost výskytu, levý a pravý ukazatel. Vytvoří se řetězec, který obsahuje posloupnost huffmanových kódů odpovídající datům ve vstupním obraze. Následně se tenhle řetězec uloží včetně informací nutných pro vytvoření huffmanova stromu. Struktura zakódovaného souboru je následující. Nejprve je zapsána dvojice hodnota odstínu šedi (*int16_t*) a jeho četnost výskytu (*uint32_t*). Poté je zapsána délka přípony

(*uint16_t*) řetězce obsahující posloupnost huffmanových kódů. Délka této přípony je vypočítána jako zbytek po dělení 16 délky řetězce (ten je tak doplněn na konci o určitý počet nul). A to z důvodu, že řetězec (posloupnost bitů) je rozdělena po násobcích 16 a uložena do výstupního souboru jako číslo (*uint16_t*). Na konci souboru je zapsána šířka obrázku, potřebná pro reverzní operaci modelu.

V případě dekódování je postup následující. Program načte vstupní zakódovaný soubor a uloží jeho informace jako hodnota odstínu šedi, četnost výskytu do vektorů a převede čísla zpět do řetězce reprezentující posloupnost bitů kódovaného obrazu. V případě že byl do řetězce doplněn určitý počet nul, je tahle přípona odstraněna. Následně se rekonstruuje huffmanův strom, aby bylo možné dekódovat řetězec zpět do jednotlivých hodnot odstínu šedi. V případě že byl použit model, provede se reverzní operace modelu k získání původních dat. Na závěr jsou data uložena ve formátu, kde jednotlivé pixely nabývají 8 bitových hodnot odstínu šedi (0-255).

4 Ovládání a překlad programu

Pro překlad programu slouží soubor *Makefile*, který lze spustit příkaz *make*. Tento příkaz vytvoří spustitelný soubor *huff_codec*. V případě že není zadán parametr *-i,-o,-w* program vyhodí chybu včetně stavu, kdy za parametr *-h* uživatel napíše něco jiného než *static* nebo *adaptive*. Parametry aplikace:

- **-c** aplikace bude vstupní soubor komprimovat
- **-d** aplikace bude vstupní soubor dekomprimovat
- **-h static** zvolí režim statického Huffmanova kódování
- **-h adaptive** zvolí režim adaptivního Huffmanova kódování
- **-m** aktivuje model, předzpracování vstupních dat
- **-i** ifile název vstupního souboru. Dle režimu činnosti aplikace, viz přepínače *-c/-d* výše, se jedná buď o data určená ke kompresi nebo dekompresi
- **-o** ofile název výstupního souboru. Dle režimu činnosti aplikace, viz přepínače *-c/-d* výše, se jedná buď o zkomprimovaná či dekomprimovaná data
- **-w** width_value udává šířku obrazu
- **-h** vypíše nápovědu na standardní výstup a ukončí se

Příklady spuštění aplikace:

Komprimace *hd01.raw* do souboru *hd01.bin* šířka obrázku 512 pixelů s využitím modelu
`./huff_codec -h static -m -c -i hd01.raw -o hd01.bin -w 512`

Dekomprimace *hd01.bin* do souboru *myhd01.raw* šířka obrázku 512 pixelů s využitím modelu
`./huff_codec -h static -m -d -i hd01.bin -o myhd01.raw`

Vypsání nápovědy

`./huff_codec -h`

5 Vyhodnocení činnosti algoritmu

V následující tabulce [1](#), lze vidět vyhodnocení algoritmu. Tabulka je rozdělena na dvě poloviny statické huffmanovo kódování s modelem a bez modelu. V jednotlivých sloupcích lze vidět velikost kódovaného souboru v kilobajtech, průměrný počet bitů na bajt obrazu a čas komprese. Pro měření času komprese

byla využita knihovna *chrono*. Čas komprese se počítá od zpracování vstupního souboru až po zápis do kódovaného souboru. Testování probíhalo na obrázcích přiložených k zadání projektu na serveru Merlin.

	Statické Huffmanovo kódování bez modelu			Statické Huffmanovo kódování s modelem		
Název Souboru	Velikost souboru	Průměrný počet bitů na bajt obrazu	Čas komprese	Velikost souboru	Průměrný počet bitů na bajt obrazu	Čas komprese
hd01.raw	126 KB	3,87	1349 ms	111 KB	3,40	1256 ms
hd02.raw	120 KB	3,69	1179 ms	109 KB	3,33	1210 ms
hd07.raw	181 KB	5,60	2426 ms	125 KB	3,85	1453 ms
hd08.raw	136 KB	4,23	1369 ms	114 KB	3,52	1190 ms
hd09.raw	215 KB	6,65	3626 ms	152 KB	4,69	2110 ms
hd12.raw	200 KB	6,19	3107 ms	142 KB	4,39	1750 ms
nk01.raw	210 KB	6,50	3413 ms	197 KB	6,07	3257 ms
Průměr	169 KB	5,24	2352 ms	135 KB	4,17	1746 ms

Tabulka 1: Vyhodnocení činnosti algoritmu

6 Závěr

Algoritmus byl úspěšně otestován na školním serveru Merlin a na osobním počítači s operačním systémem Ubuntu 18. Program byl testován na datech přiložených k zadání projektu. Z časových důvodů, nebyla implementována varianta s adaptivním huffmanovým kódováním.

7 Literatura

- http://www.stringology.org/DataCompression/sh/index_cs.html
- <https://www.geeksforgeeks.org/huffman-decoding/>
- Přednášky předmětu KKO