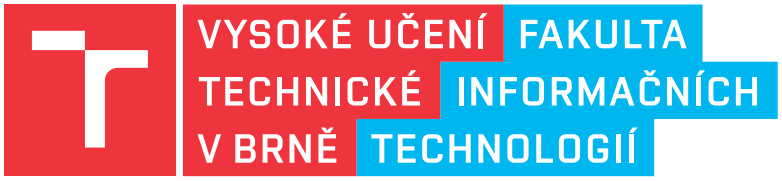

Kódování a komprese dat

Semestrální projekt

Komprese obrazových dat s využitím slovníkových metod komprese dat

13. května 2025 Martin Zmitko (xzmitk01)

1 Úvod

Tento projekt se zabývá tvorbou programu pro kompresi a dekompresi šedotónových obrazových dat s využitím slovníkových kompresních metod. Takové metody využívají slovníku (většinou tvořeného dynamicky při kompresi ze vstupních dat) a na základě hledání v něm nahrazují nalezené fráze jejími odkazy do slovníku. Řadí se mezi ně metody LZ77 (s variantami LZSS, LZOP a LZ4) a LZ78 (varianty LZFG, LZRW nebo LZW) a další.

V tomto projektu byla zvolena varianta LZSS. Jako slovník využívá vyhledávací buffer, který udržuje ve formě binárního vyhledávacího stromu. Dále udržuje předvídací buffer v kruhové frontě, pro který hledá shody ve vyhledávacím bufferu. Hledání produkuje vždy nejlepší (shodující se v nejvíce znacích) a nejlevější shodu. Pokud je shoda nalezena, je vyslána značka obsahující její offset od momentální pozice a délku. Pokud nebyla nalezena žádná shoda nebo je délka shody menší, než je zakódovaná značka, je vyslán nekomprimovaný kód. Pro odlišení značek od nekomprimovaných dat je nutný indikační bit, kvůli praktičnosti se často používá celý bajt těchto bitů vyslaný před osmi značkami či nekomprimovanými kódy.

2 Popis implementace

Projekt je napsaný v jazyce C++ a kromě standardních knihoven využívá také nádstavbu na standardní systémové `argparse`[1]. Vstupní bod programu se nachází v souboru `main.cpp`, kde je zpracování argumentů a čtení/zápis do souborů.

Zdrojový soubor `serialization.cpp` obsahuje funkce pro serializaci/deserializaci dat ze souborů, implementaci transformačního modelu, adaptivního módu a volá samotné funkce pro kompresi. Každému komprimovanému souboru je přidána hlavička, která obsahuje šířku obrazu dělenou 256 v jednom bajtu, jednobajtovou vlajku indikující použití transformačního modelu a počet bloků ve dvou bajtech. Poté nadchází samotné bloky, kdy každý má vlastní hlavičku obsahující jeden bajt s vlajkami indikujícími směr průchodu a jestli byl blok komprimován a čtyři bajty s velikostí bloku.

Jako model pro transformaci byla zvolena jednoduchá diference pixelů, kdy se od každého pixelu odečte hodnota jeho levého souseda, při dekompresi se tato hodnota přičte. Model je aplikován po skenování, ale před kompresí. V případě vertikálního skenování je daný blok transformován maticovou transformací (prohozením hodnot podél diagonály).

V případě statického režimu je uložen jeden blok s horizontálním skenováním, při adaptivním režimu se obrázek rozdělí na bloky o velikost 64x64 pixelů. Každý blok je zkomprimován pro horizontální i vertikální skenování, uložena je pouze varianta s menší velikostí. Pokud se komprese nepodaří, jsou uloženy nezkomprimované data a tato skutečnost je pro blok indikována odpovídající vlajkou.

Samotnou kompresi zajišťují funkce v souboru `lzss.cpp`. Tyto funkce čtou vstupní buffer, vyhledávají shody ve slovníku a vysílají zakódované značky. Pro značky je udržován krátký buffer, po jeho zaplnění osmi položkami je vyslán bajt s bitovými indikacemi, zda se jedná o značky či data následovaný osmi položkami.

Vyhledávací buffer implementuje třída `SearchBuffer` s implementací ve zdrojovém souboru `lzss_buffer.cpp`. Tato třída má odkaz na vstupní buffer, index momentální pozice v něm a udržuje si binární vyhledávací strom, kde každý uzel `Node` obsahuje index do bufferu. Třída má kromě konstruktoru a destruktoru dvě veřejné metody, a to `slide`, pro posunutí vyhledávacího bufferu o `n` pozic, přičemž jsou přidány do stromu nadcházející pozice a v případě

[1] `https://github.com/p-ranav/argparse`

plného bufferu odstraněny odchozí. Další metoda nalezne pozici a délku nejlepší shody. Neveřejnými metodami jsou metody pro manipulaci se stromem (vkládání, mazání, průchod při vyhledávání) a metody pro porovnání dvou řetězců ve vstupním bufferu podle odkazů v uzlech a zjištění délky shody. Vyhledávací buffer je velký 2 kB, předvídací buffer má 34 bajtů a pro vyslání značky je nutná shoda alespoň délky 3. Tyto konstanty byly zvoleny na základě experimentů.

3 Vyhodnocení efektivity

Soubor	Entropie	–	-m	-a	-ma
cb2.raw	6.91	.68	.66	.78	.80
cb.raw	1	.50	.50	.52	.52
df1h.raw	8	.50	.50	.64	.51
df1hvx.raw	4.51	.63	.70	.65	.75
df1v.raw	8	.53	.53	.64	.51
nk01.raw	6.47	8.00	8.00	7.97	7.98
shp1.raw	1.89	1.38	1.45	1.14	1.19
shp2.raw	1.87	1.50	1.62	1.30	1.41
shp.raw	0.87	.55	.56	.57	.58
Průměr	4.39	1.59	1.61	1.58	1.58

Tabulka 1: Efektivita komprese pro všechny soubory a kombinace parametrů

Soubor	–	-m	-a	-ma
cb2.raw	0.079 s	0.069 s	0.121 s	0.097 s
cb.raw	0.285 s	0.263 s	0.426 s	0.280 s
df1h.raw	0.144 s	7.694 s	1.470 s	3.976 s
df1hvx.raw	0.133 s	0.479 s	0.445 s	0.793 s
df1v.raw	1.735 s	6.905 s	1.413 s	2.798 s
nk01.raw	0.052 s	0.050 s	0.082 s	0.078 s
shp1.raw	0.050 s	0.054 s	0.101 s	0.099 s
shp2.raw	0.055 s	0.059 s	0.113 s	0.111 s
shp.raw	0.156 s	0.141 s	0.122 s	0.115 s
Průměr	0.30 s	1.75 s	0.48 s	0.93 s

Tabulka 2: Doba trvání komprese pro všechny soubory a kombinace parametrů