

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



## Kódování a komprese dat - projekt

# 1 Dokumentace

Projekt je rozdělen do několika podstatných částí. `scanner.hpp`, `model.hpp`, `compressor.hpp`, `decompressor.hpp`. Jednotlivé části popíšu v následujících kapitolách.

## 1.1 Skener

Scanner používá adaptivní skenování a má naimplementovány 2 směry - horizontální a vertikální. Mezi těmito směry se dynamicky přepíná podle toho, který kompresně vychází lépe. Jako velikost bloku jsem zvolil 16x16.

## 1.2 Modelovací algoritmus

Pro model dat jsem použil v zadání zmíněný algoritmus - difference sousedních pixelů. Tedy hodnota dat je rovna odečtení hodnoty předchozího pixelu. Samozřejmě je implementována i verze pro dekompresi, která tento model reverzuje.

$$y_n = \begin{cases} x_n, & \text{if } n = 1 \\ (x_n - x_{n-1}) \bmod 256, & \text{if } n > 1 \end{cases}$$

## 1.3 Kompresní algoritmus

Pro tento projekt jsem použil kompresní algoritmus LZ77. Ačkoliv se jedná o nejjednodušší z prezentovaných metod, je překvapivě uspokojivá na testovacím datasetu. Velikost vyhledávací paměti jsem zvolil na 256 B a předvídací paměť jsem nechal pouze na 16 B. Ačkoliv se hodnoty můžou zdát malé, k projektu mi přišly dostačující.

## 1.4 Zajímavé struktury

Je implementována struktura pro reprezentaci tokenů LZ77. Nachází se v `compressor.hpp`, ve stejném souboru se také nachází struktura `CompressedBlock` použitá při kompresi dat. `CompressedBlock` reprezentuje každý blok dat, které se komprimují nezávisle na sobě. Tedy při zapnutém adaptivním skenování se jedná o mnoho bloků o velikosti 16x16, zatímco s vypnutým přepínačem se jedná pouze o jeden blok o velikosti celého obrázku.

## 1.5 Struktura komprimovaného souboru

Komprimovaný soubor začíná hlavičkou o velikosti 2B. Poté je následován 2B určenými pro vlajky, ačkoliv se používají pouze nejnížší 2 bity pro označení využití modelu a adaptivního skenování. Poté se nachází 2B počet použitých bloků. Za počtem bloků se nacházejí 2 stejně dlouhá, bitová pole. První se používá pro určení skenování jednotlivých bloků (v případě, že nebylo adaptivní skenování použito, se pole neukládá). Druhé bitové pole značí, zda bloky byly ponechány v původní velikosti nebo byly zkomprimovány. Poté jsou uloženy samotné bloky naplněné tokeny. U každého bloku je poznačena zkomprimovaná velikost samotného bloku, tedy počet uložených bytů - ať už tokenů nebo původních dat.

## 2 Výsledky

Bohužel se mi nepovedlo oddebugovat nějaké chyby, tedy pár testů mi neprošlo. Mezi testy ani není žádná podobnost, tedy nepovedlo se mi najít chybu.

obrazová data	entropie	–		–m		–a		–m –a	
		bpp	ct [s]	bpp	ct [s]	bpp	ct [s]	bpp	ct [s]
cb.raw	1.000	1.8828	0.0414	1.8830	0.0530	2.0705	0.0351	2.1330	0.0390
cb2.raw	6.906	5.7503	0.0474	2.0006	0.0391	3.9455	0.0150	2.3205	0.0285
df1h.raw	8.000	1.9121	0.0429	1.8828	0.0769	2.1955	0.0121	2.3127	0.0307
df1hvx.raw	4.514	3.5801	0.0452	2.0642	0.0535	3.0187	0.0162	2.7481	0.0265
df1v.raw	8.000	-	-	-	-	2.1955	0.0192	2.3127	0.0306
nk01.raw	6.4729	-	-	-	-	8.0668	0.0667	8.0685	0.0710
shp.raw	0.8675	-	-	-	-	3.3205	0.0303	3.5705	0.0326
shp1.raw	1.8961	3.3698	0.0537	-	-	3.2424	0.0275	3.4566	0.0318
shp2.raw	1.8700	3.8690	0.0668	4.0197	0.0768	3.1597	0.0297	3.3668	0.0309
<b>průměr</b>		3.940		2.3701		3.4683		3.3655	

Tabulka 1: Srovnání efektivity komprese a doby běhu pro různé metody a obrazová data

Z dat vychází, že zapnutí adaptivního skenování souboru pomáhá spíše výjimečně, zatímco používání modelu pomáhá skoro vždy. Bohužel je u testování samotného modelu hodně chybějících dat na to, abych mohl říct, že je lepší program spouštět s ním. Vylepšení adaptivního modelu by bylo používání větších úseků dat, než pouze 16x16B, kdy by se pravděpodobně více projevila výhoda slovníkových metod.

## Použití

```
./lz_codec [-c|-d] [-m] [-a] -i ifile -o ofile [-w width_value]
```

where:

**-c** Compress the input file (this is the default mode).

**-d** Decompress the input file.

**-m** Enable modeling (e.g., differential encoding) for input data processing. This can be used in both compression and decompression.

**-a** Enable adaptive image scanning. Otherwise, horizontal sequential scanning is used. Can be used for both compression and decompression.

**-i ifile** Specify the input file name (used for either compression or decompression, depending on the selected mode).

**-o ofile** Specify the output file name (compressed or decompressed depending on mode).

**-w width\_value** Specify the image width, where  $\text{width\_value} \geq 1$ . This is required only during compression.