**Informacije in kodi**

Poročilo projektne naloge

Študent: Tilen Tinta

Mentorja: izr. prof. dr. Simon Dobrišek, as. dr. Klemen Grm

Predmet: Informacije in kodi

Datum: 12.5.2024

Vsebina

[1 Uvod 1](#_Toc166429112)

[1 Kodirnik 1](#_Toc166429113)

[1.1 QOI\_OP\_RGB 1](#_Toc166429114)

[1.2 QOI\_OP\_RGBA 2](#_Toc166429115)

[1.3 QOI\_OP\_INDEX 2](#_Toc166429116)

[1.4 QOI\_OP\_DIFF 2](#_Toc166429117)

[1.5 QOI\_OP\_LUMA 2](#_Toc166429118)

[1.6 QOI\_OP\_RUN 2](#_Toc166429119)

[2 Dekodirnik 2](#_Toc166429120)

[3 Vrednotenje 2](#_Toc166429121)

[4 Zaključek 2](#_Toc166429122)

[5 Reference 2](#_Toc166429123)

**Ključne besede: QOI, kodiranje, dekodiranje, vrednotenje**

# Uvod

Pri izbirnem projektu pri predmetu Informacije in kodi smo se tokrat ukvarjali z implementacijo kodirnika in dekodirnika slik QOI. Za preverjanje kakovosti kompresije smo izvedi tudi vrednotenje retultatov.

To je razmeroma nov algoritem razvit leta 2022 kot dokaz, da tudi brez izjemno zahtevne matematike lahko dosežemo dobre rezultate kompresije slik. Iz tega izhaja tudi samo ime (the Quite Ok Image format). Razvil ga je Dominic Szablewski.

# Kodirnik

Prva naloga je bilo napisati kodirnik slik za omenjen format. Kot že omenjeno je celoten algoritem precej kratek s strani pravil. Ta ima svojo logiko zapisano na samo eni strani lista A4 formata in ga sestavlja 6 možnih kodirnih funkcij, ki so opisane v nadaljevanju.

Format zapisa bi lahko razdelili v tri dele. To so glava dokumenta, kodiran zapis same slike in konec dokumenta.

Glavo sestavljajo podatki o sliki ter zapis, ki takoj kaže da gre za QOI format.

**qoi\_header {**

**char magic[4];**

**uint32\_t width;**

**uint32\_t height;**

**uint8\_t channels;**

**uint8\_t colorspace;**

**};**

V »originalu« je algoritem napisan v jeziku C, zato je tudi zgornji zapis primerne oblike.

Spremenlivka *magic[4]* vsebuje zapis »qoif«, ki predstavlja sam zapis in je dolžine 4 bajte. Enako dolgi sta tudi naslednji dve spremenljivki, ki predstavljata širino in višino same slike. Zadnji dve spremenljivki sta dolgi le po en bajt. Prva predstavlja število barvnih kanalov, ki so zastopani v sliki. To so trije v primeru RGB zapisa oz. 4 če gre za RGBA zapis. *Colorspace* predstavlja zapis alfa kanala. Zadnji dve spremenlivki, torej *colorspace* in *cannels* sta zgolj informativni in ne spremenita delovanje algoritma.

Kodiranje slike se izvaja na sliki od leve proti desni od zgoraj navzdol.

## QOI\_OP\_RGB

## QOI\_OP\_RGBA

## QOI\_OP\_INDEX

## QOI\_OP\_DIFF

## QOI\_OP\_LUMA

## QOI\_OP\_RUN

# Dekodirnik

# Vrednotenje

# Zaključek

V nalogi smo spoznali tako kompresijo kot dekompresijo datotek z algoritmom LWZ. Hkrati je naloga od nas zahtevala ustrezno računanje uspešnosti kompresije katera se glede na velikost datoteke spreminja. To prikazuje tudi graf s slike 2-1, ki prikazuje padanje nivoja kompresije saj se v besedilu začne več vsebine ponavljati in s tem se to lahko optimalnejše zapiše. Hkrati sem algoritem preizkusil tudi na različnih datotekah tako »surovih« kot shranjenih v bolj optimalnih oblikah, kar mi je podalo zanimive rezultate.

# Reference

* Prosojnice vaj: Informacije in kodi, as. Klemen Grm
* Spletni forumi za uporabo python funkcij
* ChatGPT za razlago nekaterih pojmov, algoritmov ter funkcij programskega jezika in snovi
* Implementacija QOI kodirnika: <https://www.youtube.com/watch?v=GgsRQuGSrc0&ab_channel=LowByteProductions>
* Implementacija QOI dekodirnika: <https://www.youtube.com/watch?v=5bWopQj-oQs&ab_channel=LowByteProductions>
* Repozitorij QOI algoritma: <https://github.com/phoboslab/qoi>