

POLITECHNIKA POZNAŃSKA

WYDZIAŁ ELEKTRYCZNY
INFORMATYKA



TEORIA INFORMACJI I KODOWANIE
DOKUMENTACJA PROJEKTU

Autorzy:

Michał MAJKA

Nr albumu: 112679

Piotr PARYSEK

Nr albumu: 106100

Prowadzący:

dr inż. Ewa IDZIKOWSKA

13 stycznia 2016

1	Wstęp	1
2	Algorytm	1
	2.1 Historia	1
	2.2 Zasada działania	1
3	Opis implementacji	2
	3.1 Kodowanie	2
	3.2 Dekodowanie	5
4	Użytkowanie programu	6
5	Testy	12
	5.1 Przedstawienie wyników:	13
6	Wnioski	16
	Literatura	16

1 Wstęp

Zadaniem projektowym była implementacja algorytmu kompresji bezstratnej **Run-Length Encoding (RLE)**.

Zadanie zrealizowano w środowisku programistycznym Qt Creator 5.5.1[4] korzystając z kompilatora GCC 4.9.1[5].

Do kontroli wersji oraz plików źródłowych wykorzystano oprogramowanie Git[6], projekt hostowano w repozytorium GitHub[7].

Dokumentację wykonano w L^AT_EX[1] w programie Texmaker 4.5 [2] oraz w edytorze online: ShareLaTeX[3].

2 Algorytm

2.1 Historia

Run-Length Encodings, również znane jako **Golomb Codings**, swoje „podwaliny” powstania wiąże z pracami, XVII-wiecznego francuskiego matematyka *Blaise’a Pascal’a*, związanymi z probabilistyką[9]. Koncepcja kodowania powtarzających się znaków była używana od początków istnienia teorii informacji (Shannon 1949, Laemmel 1951), jednakże metodę oraz sposób kodowania wynalazł i opracował *Solomon Wolf Golomb*[8][9].

2.2 Zasada działania

Algorytm jest relatywnie prosty → przedstawia powtarzające się wartości jako dany znak i licznik powtórzeń. Na przykład ciąg znaków:

pppppppuuuuutttt.....ppppooooozzzzznnnnnnnnnn....pppppplllll

Zostaje przedstawiony w postaci ciągu:

p7u5t5.6p4o6z5n4an5.4p6l5

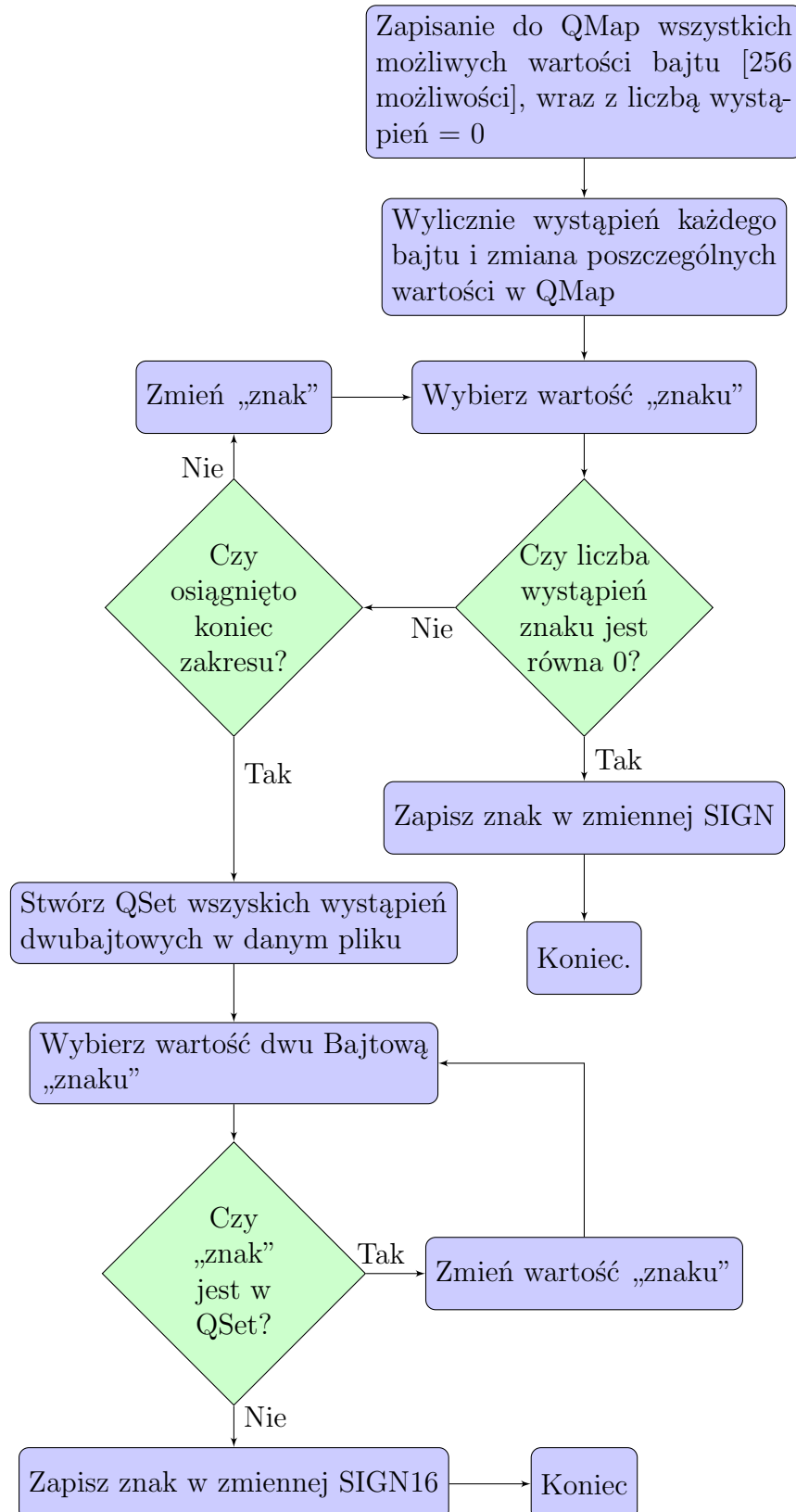
Ciąg znaków	Liczba
<i>pppppppuuuuutttt.....ppppooooozzzzznnnnnnnnnn....pppppplllll</i>	64
<i>p7u5t5.6p4o6z5n4an5.4p6l5</i>	26
	$26/64 \approx 0.41$

Tabela 1: Przykładowa kompresja znakowa

3 Opis implementacji

3.1 Kodowanie

Ustalenie znaku kodowania



Rysunek 1: Schemat blokowy wyszukiwania „znaku”.

Do wyszukania „znaków” wykorzystano kontener `QMap<quint, int>`, gdzie zmienna `quint8` (unsigned byte) wskazuje na poszczególne możliwe wartości bajta, a zmienna `int` wskazuje ilość wystąpień danego bajta w opracowywanym pliku.

```

1 QMap<quint8 , int> IIMap;
2 for (quint8 i = 0; i < 255; i++) {
3     IIMap.insert(i , 0);
4 }

```

Kod 1: Deklaracja wraz z inicjalizacją kontenera `QMap<quint8, int>`

W przypadku zdarzenia, że w pliku występują wszystkie możliwe kombinacje bajta, zamiast jedno bajtowego „znaku” zostaje wprowadzony znak dwu bajtowy:

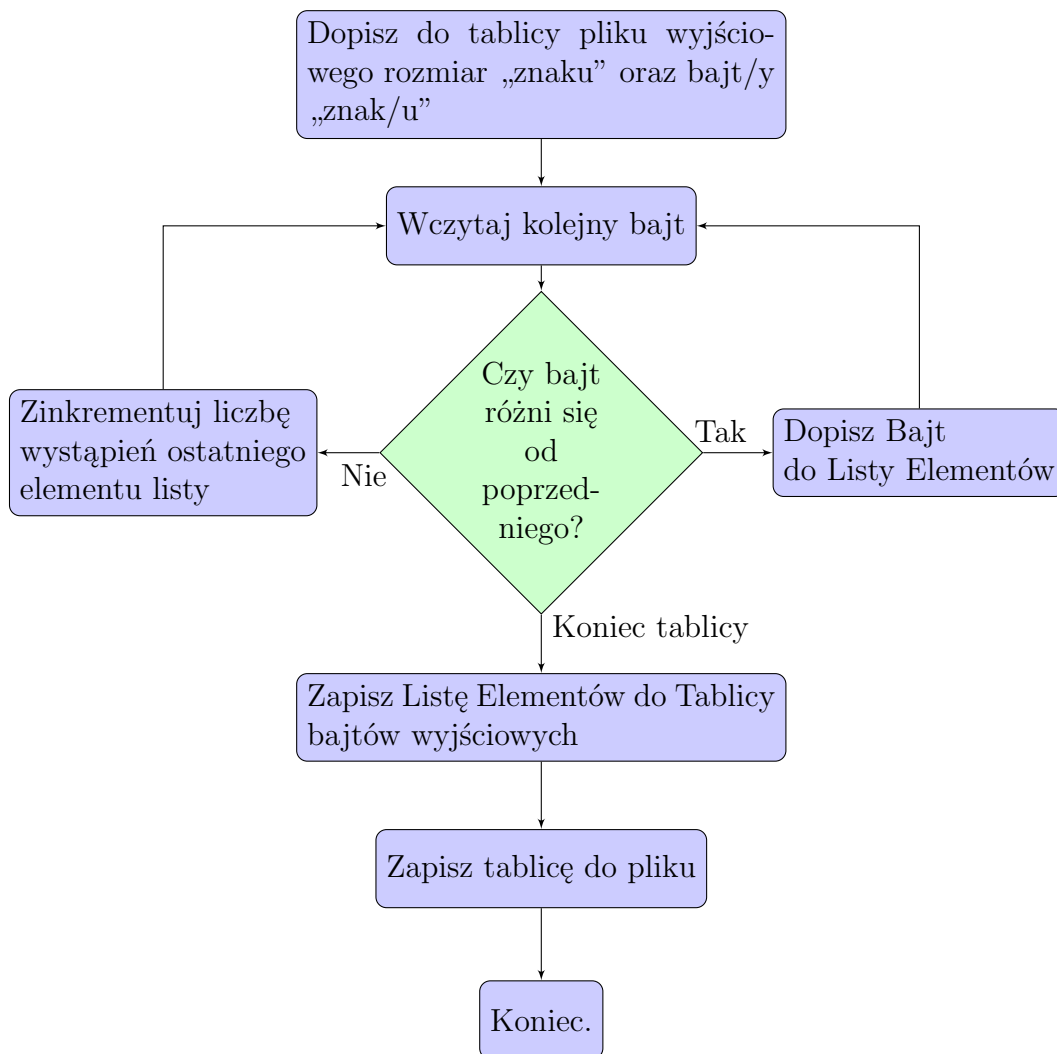
```

1 QPair<quint8 , quint8> SIGN16;

```

Kod 2: Deklaracja dwu bajtowej zmiennej znakowej

Kodowanie



Rysunek 2: Schemat blokowy kodowania pliku.

Do sprawnego wczytania, zliczenia i zakodowania poszczególnych bajtów stworzono kontener QList struktury Element. Struktura Element posiada dwa pola: quint8 item → oznaczające dany bajt oraz quint32 value → oznaczające ilość wystąpień (Założono, że dany bajt nie powtórzy się więcej jak 4294967295 razy).

```
1 struct Element {
2     quint8 item;
3     quint32 value;
4 };
5 QList<Element> Elements;
6 quint8 CurrentByte;
```

Kod 3: Główne struktury danych kodowania

Zapisanie danych do pliku odbywa się za pośrednictwem tablicy bajtów QByteArray. Analiza zapisanych znaków odbywa się poprzez przejście przez wcześniej wspomnianą listę: QList<Element> i odpowiednią interpretację wartości wystąpień danego znaku.

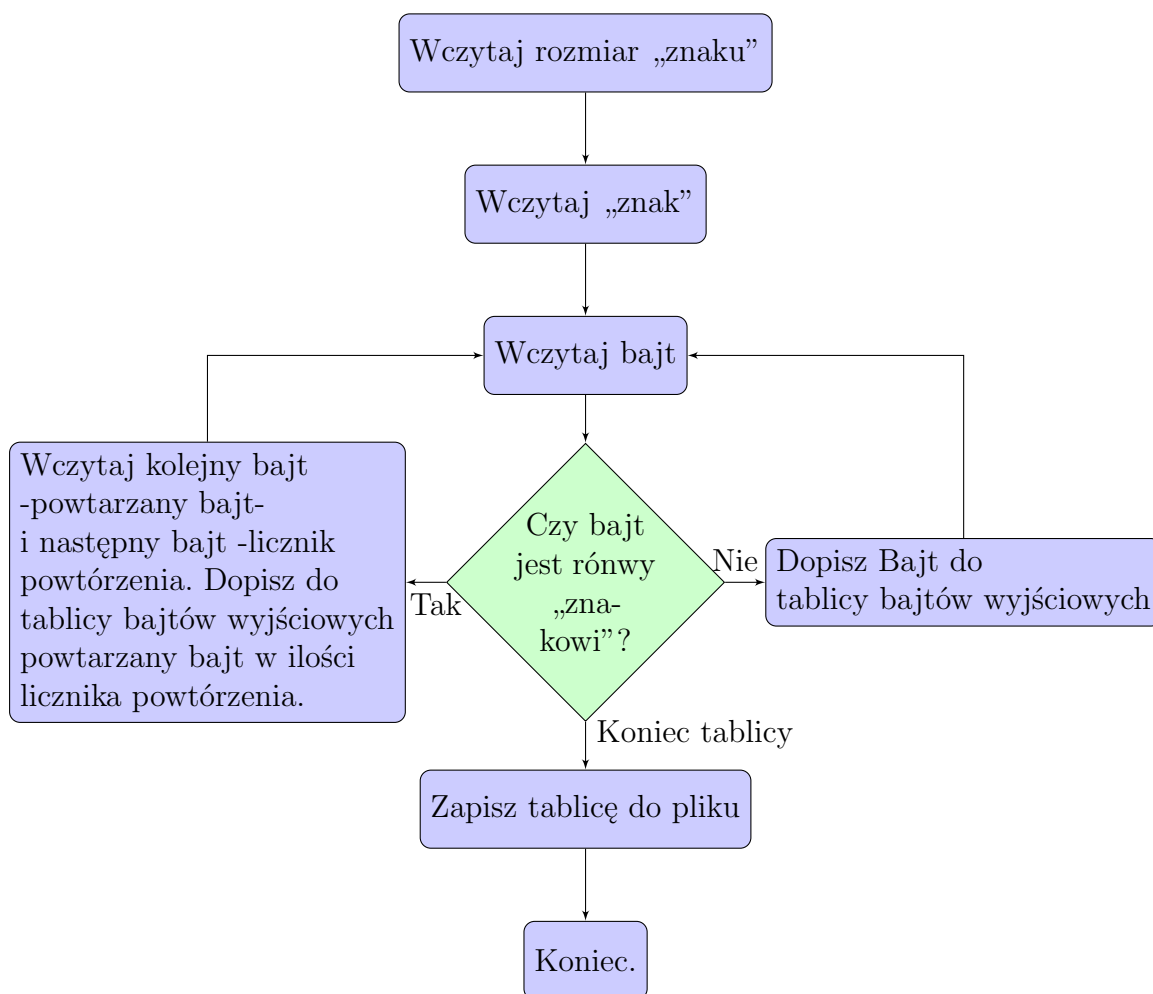
Jeżeli wartość powtórzenia danego znaku nie jest większa niż minimalna długość jego zastąpienia, który ma postać „znak” inicjujący, bajt powtarzany, ilość powtórzeń, to wpisywana jest „pierwotna postać”.

W przypadku, gdy wartość ilości powtórzeń bajtu jest większa niż maksymalna wartość jaką może osiągnąć bajt - 255 - to postać zastąpienia przybiera postać: czterech bajtów „znaku”, bajt powtarzany i cztery bajty licznika.

```
1 if (e.value < 256) {
2     OutByteArray.append(SIGN);
3     OutByteArray.append(e.item);
4     OutByteArray.append(e.value);
5 } else {
6     OutByteArray.append(SIGN);
7     OutByteArray.append(SIGN);
8     OutByteArray.append(SIGN);
9     OutByteArray.append(SIGN);
10    OutByteArray.append(e.item);
11    QByteArray TempArray;
12    TempArray = RLE::IntToHex(e.value);
13    OutByteArray.append(TempArray);
14 }
```

Kod 4: Zapisanie i zakodowania danych skompresowanych do pliku

3.2 Dekodowanie



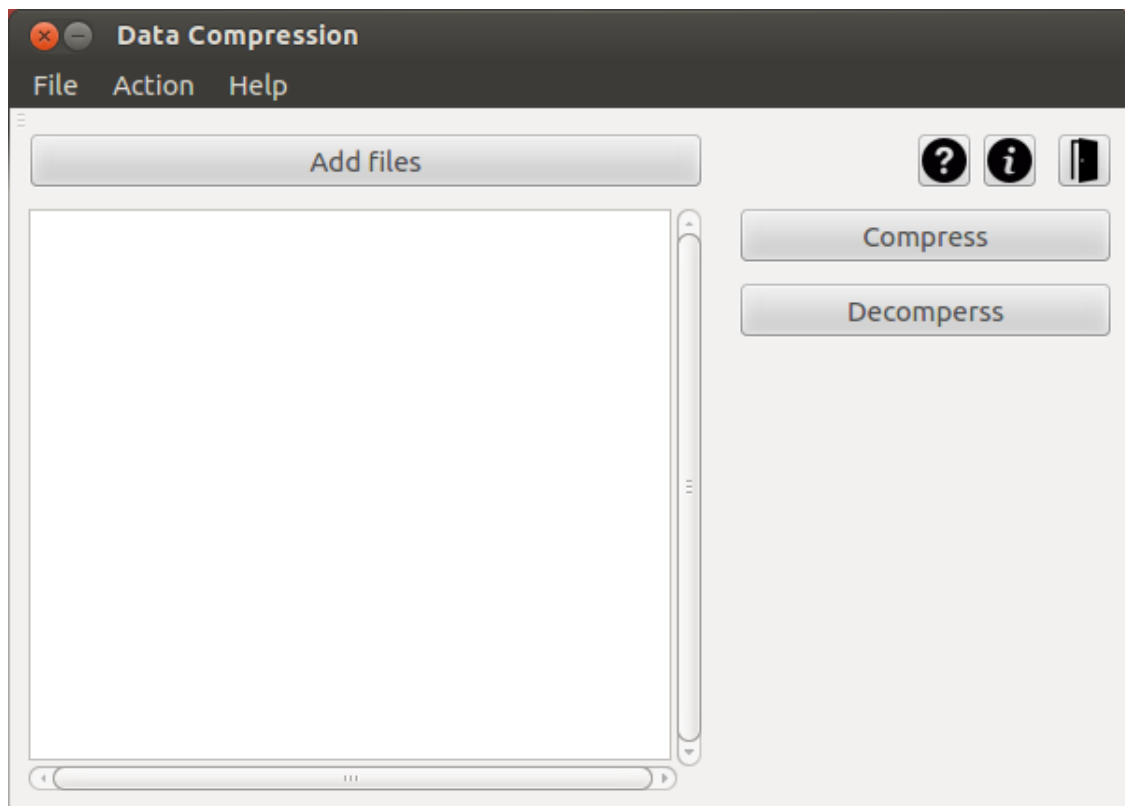
Rysunek 3: Schemat blokowy dekodowania pliku.

Dekodowanie odbywa się z pomocą analogicznych struktur / zasad / metod co zostały użyte podczas kodowania.

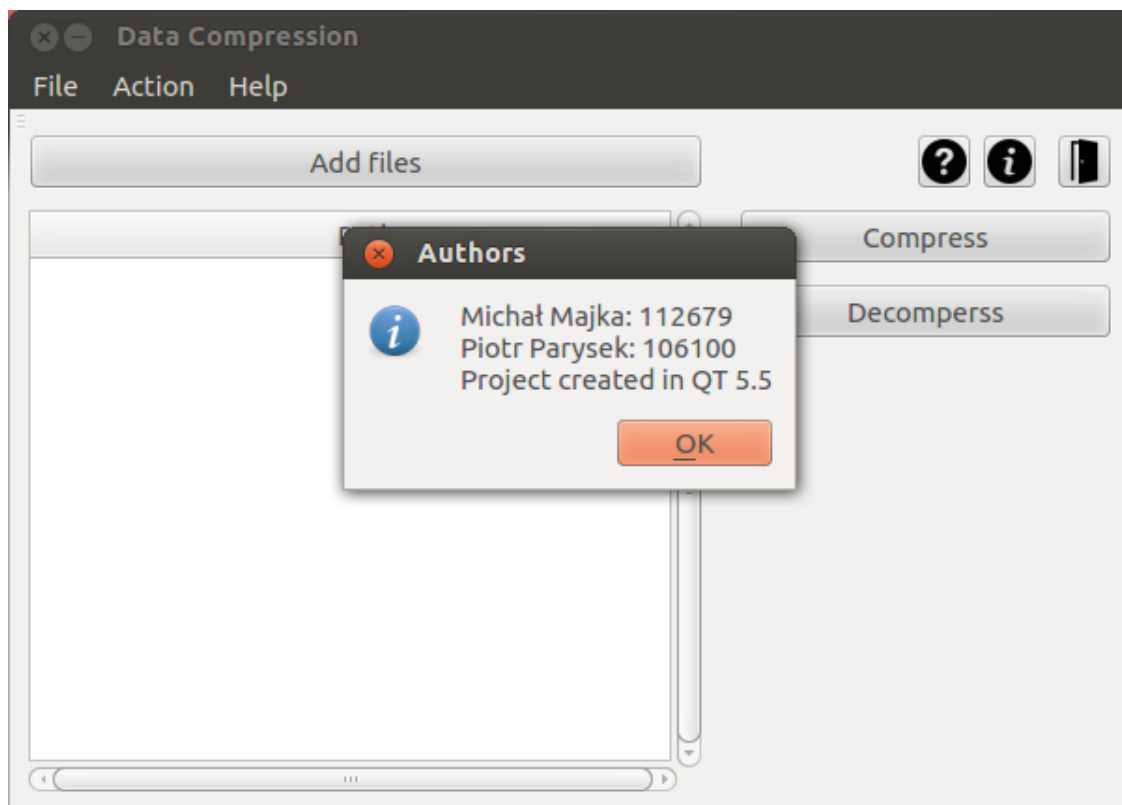
W celu ułatwienia kontroli nad plikami po kodowaniu mają one dodawany przyrostek . rlemama, a podczas dekodowania dodawany przed znacznikiem formatu pliku przyrostek _2

4 Użytkowanie programu

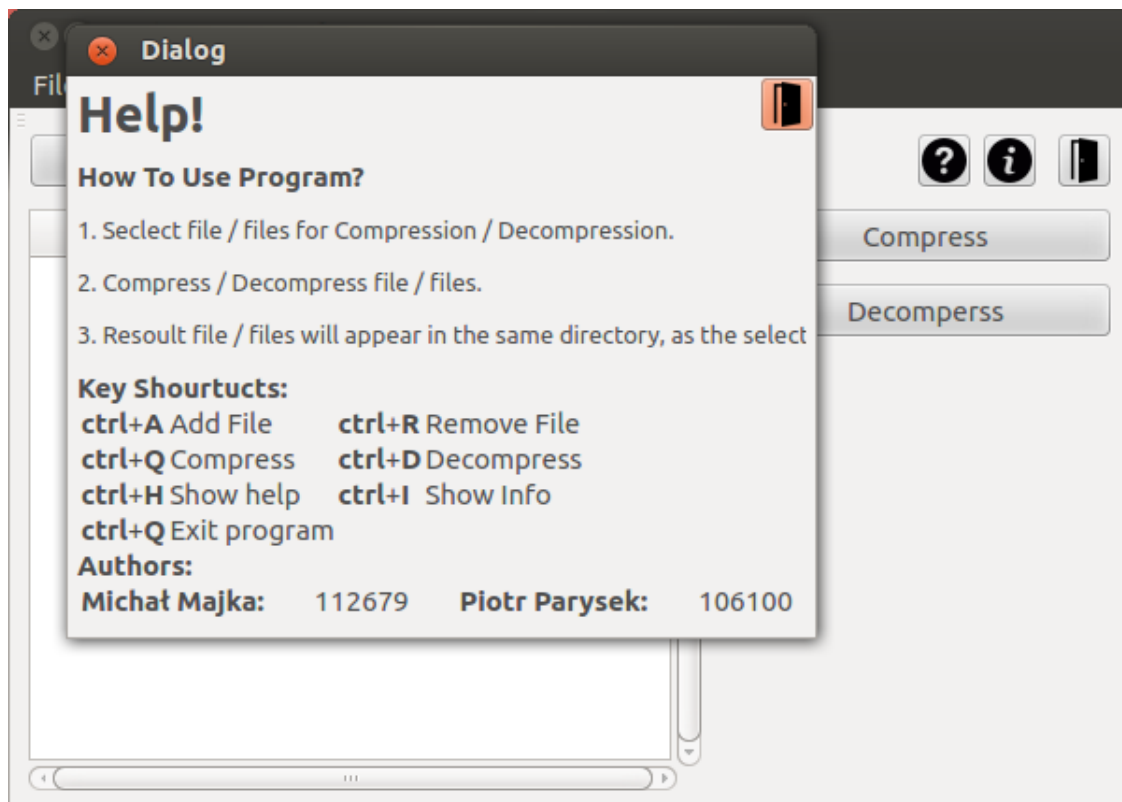
Ikony umieszczone w programie zostały pobrane z strony: <http://www.flaticon.com/>[10].



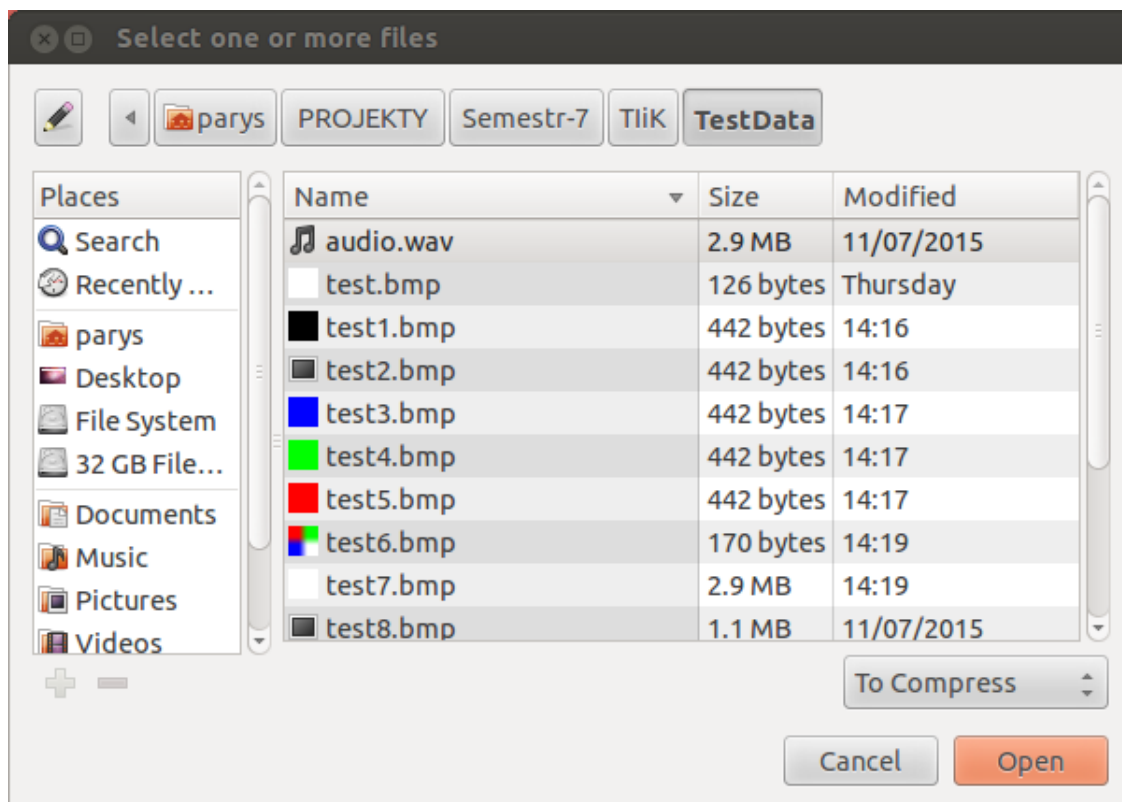
Zrzut ekranu 1: Wygląd po „starcie” programu.



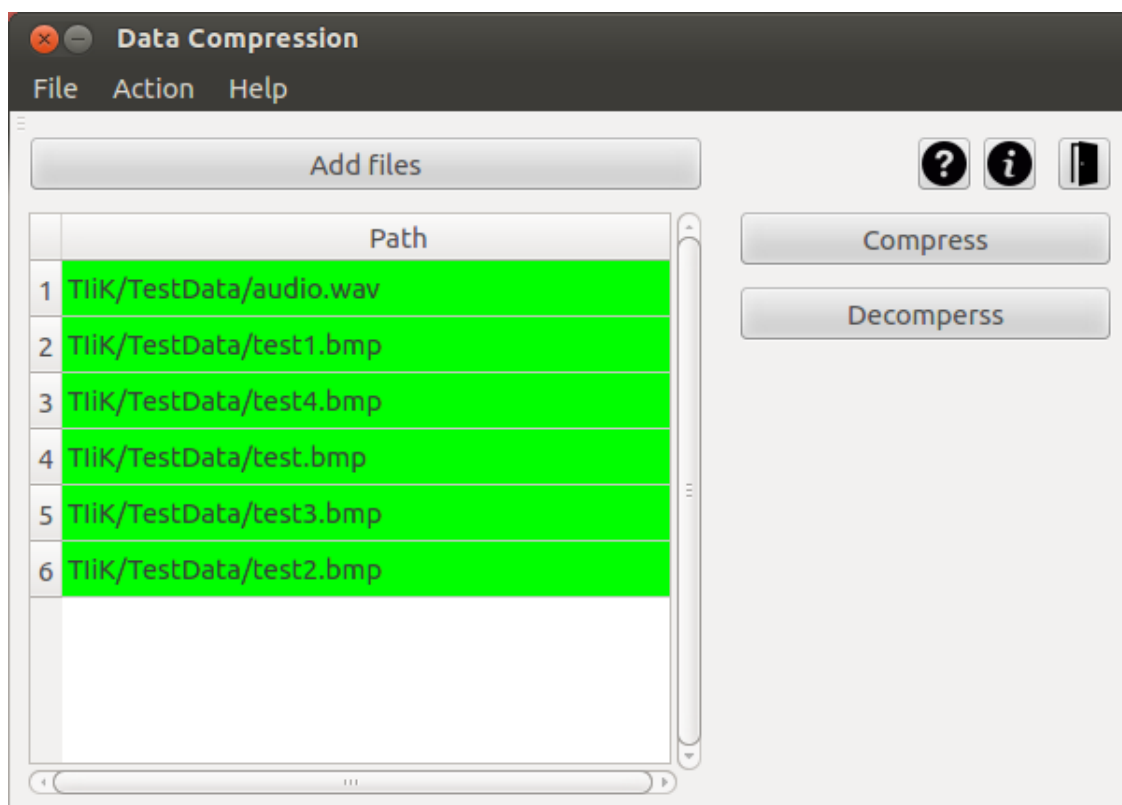
Zrzut ekranu 2: Włączenie informacji o autorach projektu.



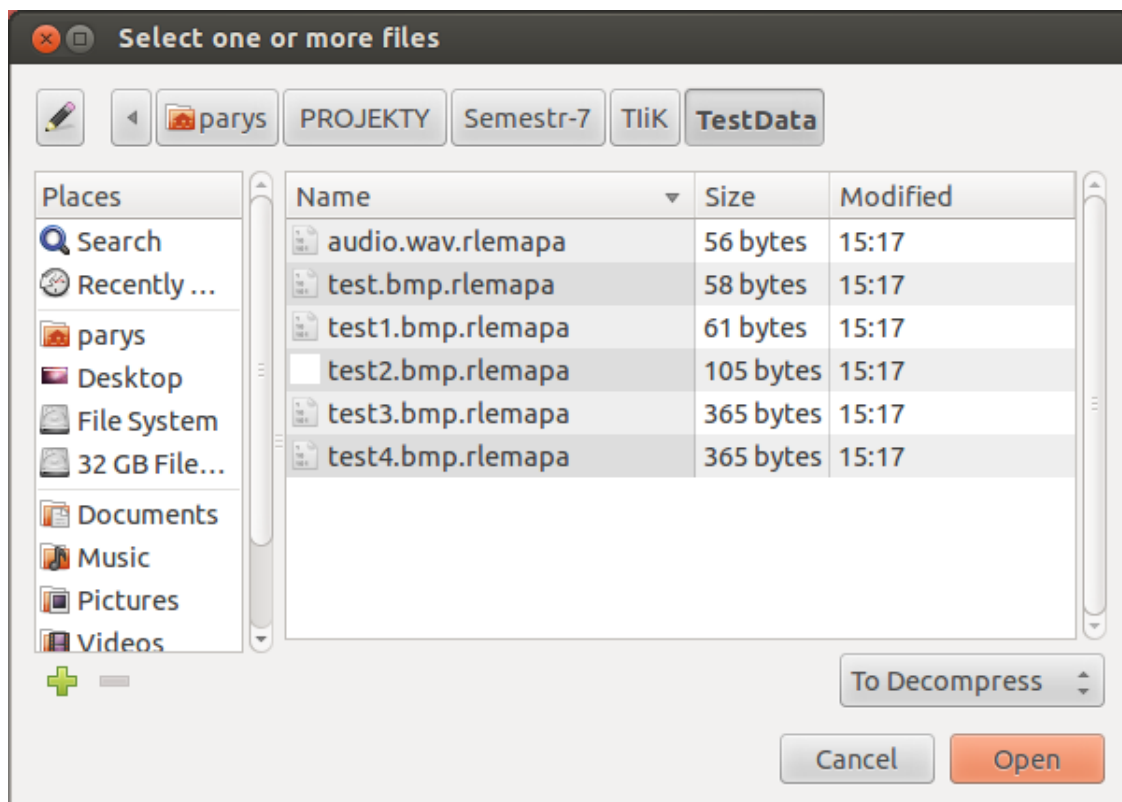
Zrzut ekranu 3: Włączenie okna pomocy.



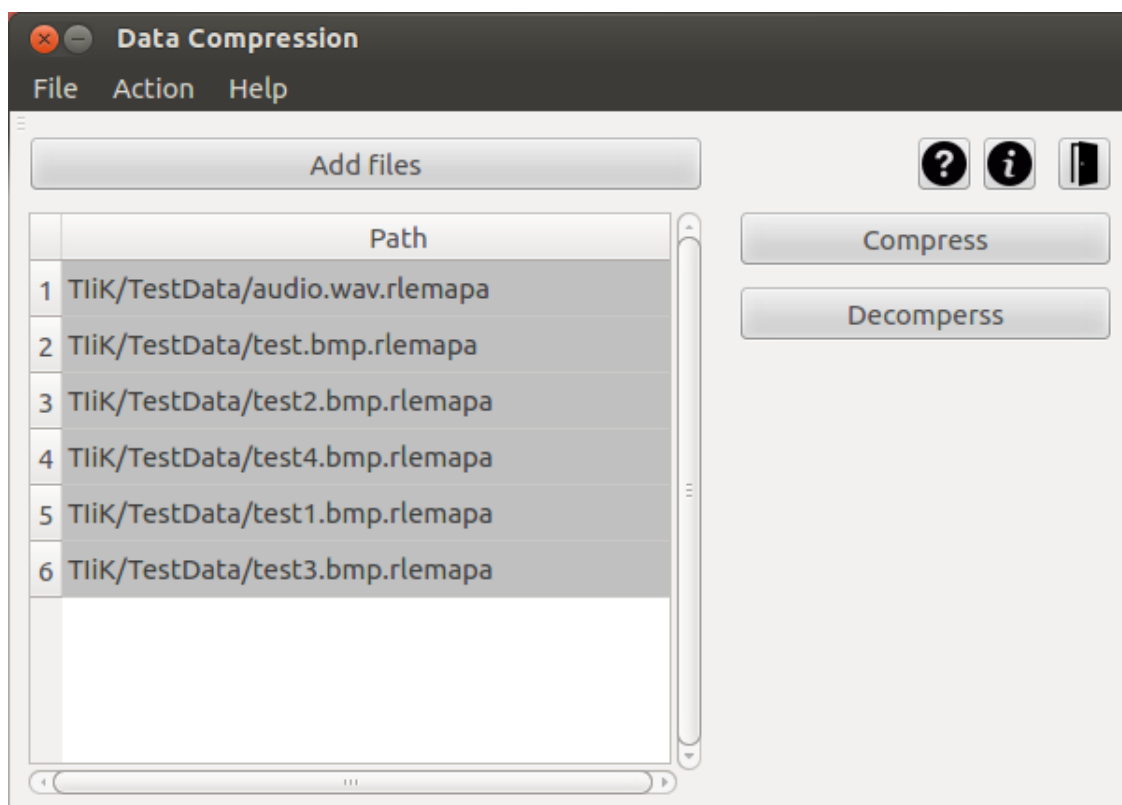
Zrzut ekranu 4: Przekształcenie okna dialogowego wyboru plików, które program może skompresować.



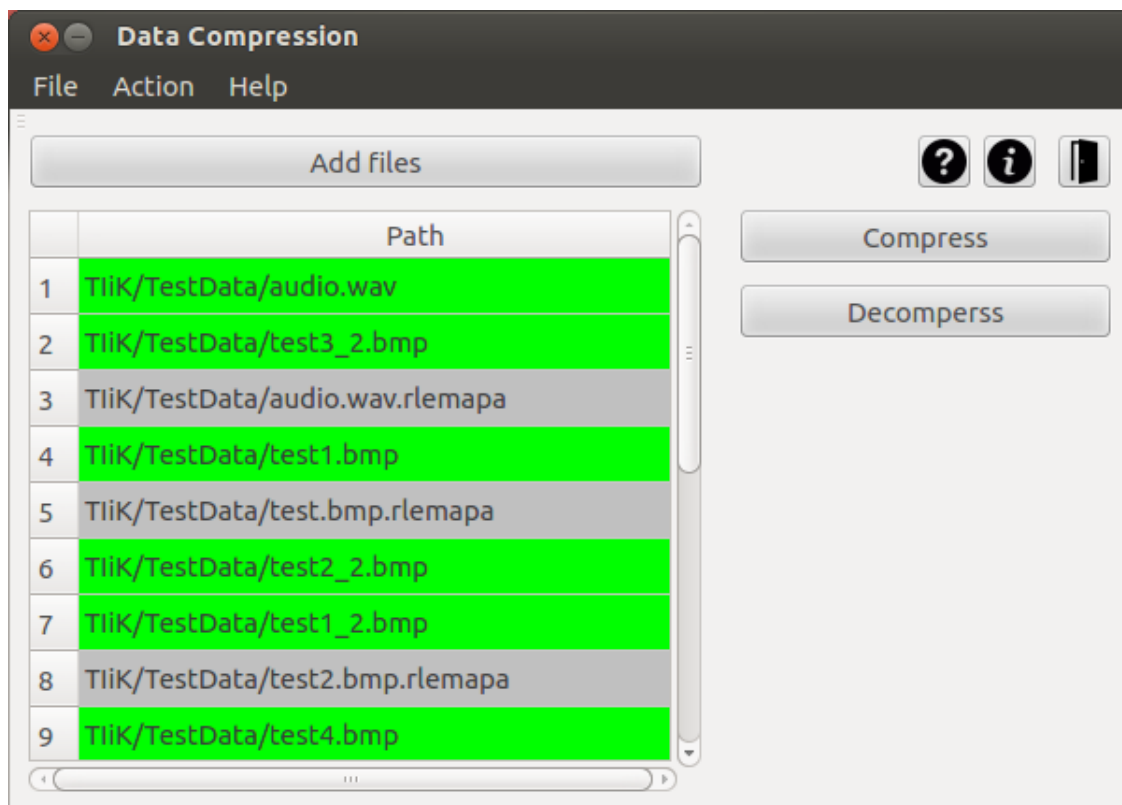
Zrzut ekranu 5: Przedstawienie plików gotowych do kompresji.



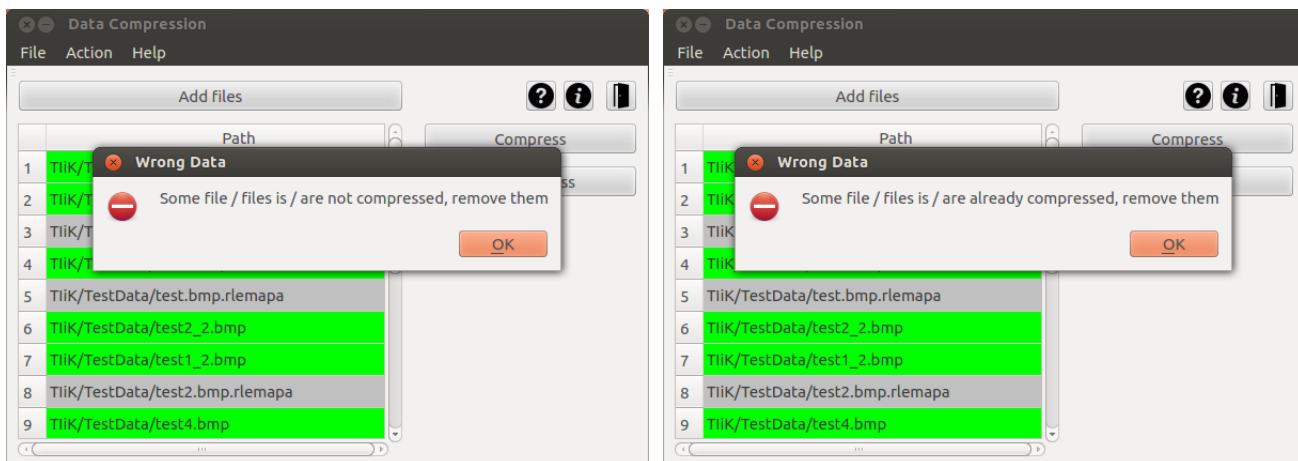
Zrzut ekranu 6: Przekształcenie okna dialogowego wyboru plików, które program może zdekompresować.



Zrzut ekranu 7: Przedstawienie plików gotowych do dekompresji.

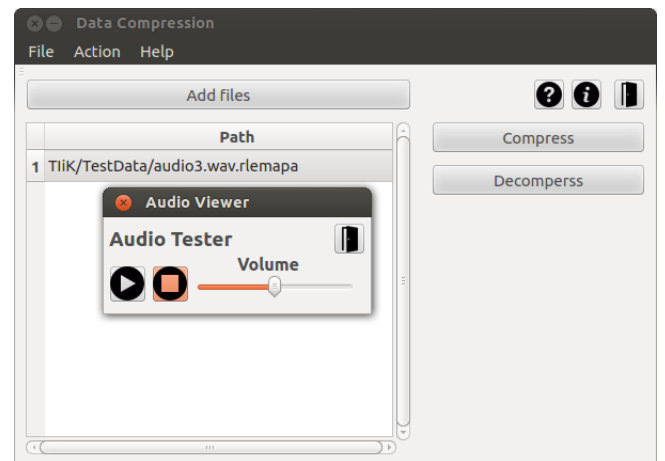
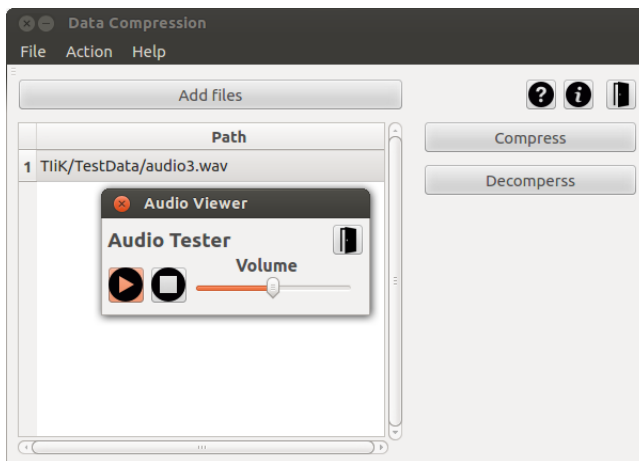


Zrzut ekranu 8: Przedstawienie plików z dekompresowanych i nie skompresowanych

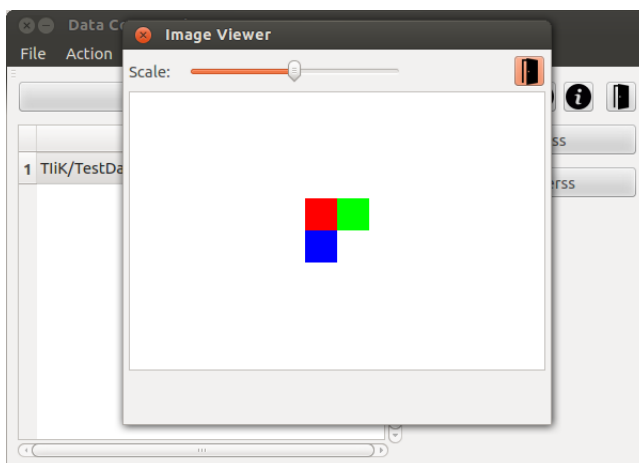


Zrzut ekranu 9: Przedstawienie komunikatów błędów, gdy nakażemy z dekompresować / skompresować pliki „przemieszane”.

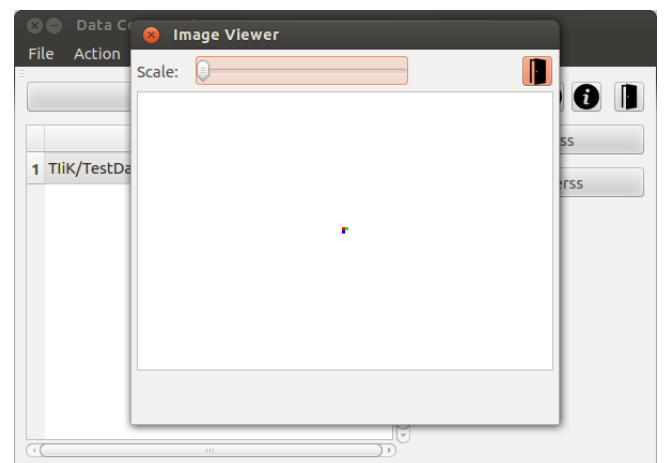
Dodatkowo w programie zaimplementowano mechanizmy „podglądu” przetwarzanych plików.



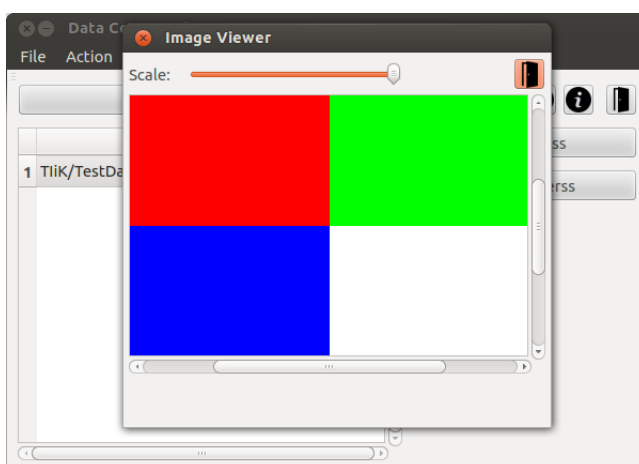
Zrzut ekranu 10: Okno umożliwiające przesłuchanie utworów muzycznych przed i po kompresji.



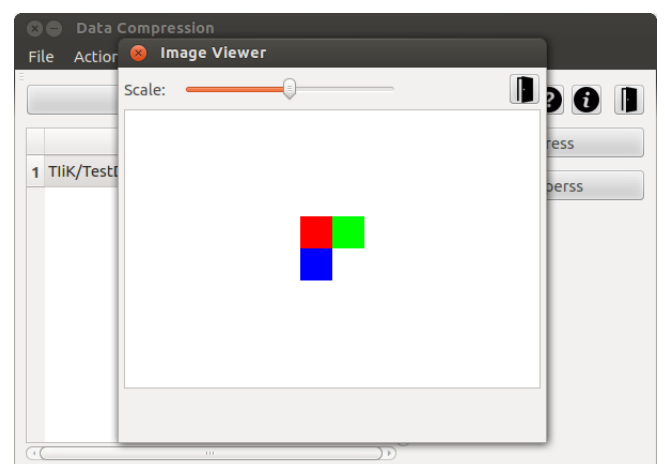
(a) Przed kompresją, bez skali.



(b) Przed kompresją, minimalna skala.



(c) Przed kompresją, maksymalna skala.



(d) Po kompresji, bez skali.

Zrzut ekranu 11: Okno umożliwiające przesłuchanie podgląd plików graficznych.

5 Testy

W celu przeprowadzenia testów wykonano kilka prostych obrazków formatu bmp oraz pobrano z Internetu inne, większe i bardziej skomplikowane obrazki. Dodatkowo do badań pobrano kilka plików dźwiękowych formatu wav z strony <http://download.wavetlan.com/SVV/Media/HTTP/http-wav.htm>[11].

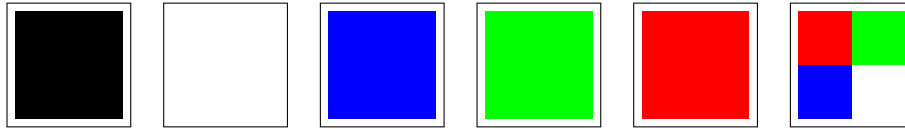


Tabela 2: Przykładowe pliki graficzne (powiększone):

1	NAZWA	ROZMIAR [B]	1	NAZWA	ROZMIAR [B]
2	audio.wav	3009870	2	audio.wav.rlemapa	56
3	audio2.wav	316002	3	audio2.wav.rlemapa	315928
4	audio3.wav	869028	4	audio3.wav.rlemapa	856106
5	audio4.wav	261262	5	audio4.wav.rlemapa	261239
6	highway.wav	8533723	6	highway.wav.rlemapa	8447839
7	test.bmp	126	7	test.bmp.rlemapa	58
8	test1.bmp	442	8	test1.bmp.rlemapa	61
9	test2.bmp	442	9	test2.bmp.rlemapa	105
10	test3.bmp	442	10	test3.bmp.rlemapa	365
11	test4.bmp	442	11	test4.bmp.rlemapa	365
12	test5.bmp	442	12	test5.bmp.rlemapa	364
13	test6.bmp	170	13	test6.bmp.rlemapa	95
14	test7.bmp	10922	14	test7.bmp.rlemapa	8216
15	test8.bmp	20138	15	test8.bmp.rlemapa	15076
16	test9.bmp	120122	16	test9.bmp.rlemapa	413885
17	test10.bmp	1163198	17	test10.bmp.rlemapa	1163182
18	test11.bmp	44264	18	test11.bmp.rlemapa	5689
19	test12.bmp	309464	19	test12.bmp.rlemapa	85103
20	test13.bmp	16000138	20	test13.bmp.rlemapa	8440758
21	test14.bmp	131554	21	test14.bmp.rlemapa	76746
22	test15.bmp	693122	22	test15.bmp.rlemapa	345212

Kod 5: Przedstawienie plików przed kompresją. Kod 6: Przedstawienie plików po kompresji.

1	NAZWA	ROZMIAR [B]	1	NAZWA	ROZMIAR [B]
2	audio.wav	3009870	2	audio_2.wav	3009873
3	audio2.wav	316002	3	audio2_2.wav	316001
4	audio3.wav	869028	4	audio3_2.wav	869034
5	audio4.wav	261262	5	audio4_2.wav	261262
6	highway.wav	8533723	6	highway_2.wav	8533722
7	test.bmp	126	7	test_2.bmp	126
8	test1.bmp	442	8	test1_2.bmp	445
9	test2.bmp	442	9	test2_2.bmp	442
10	test3.bmp	442	10	test3_2.bmp	442
11	test4.bmp	442	11	test4_2.bmp	442
12	test5.bmp	442	12	test5_2.bmp	442
13	test6.bmp	170	13	test6_2.bmp	170
14	test7.bmp	10922	14	test7_2.bmp	10922
15	test8.bmp	20138	15	test8_2.bmp	20141
16	test9.bmp	500138	16	test9_2.bmp	120122
17	test10.bmp	1163198	17	test10_2.bmp	1163198
18	test11.bmp	44264	18	test11_2.bmp	44270
19	test12.bmp	309464	19	test12_2.bmp	309464
20	test13.bmp	16000138	20	test13_2.bmp	16000138
21	test14.bmp	131554	21	test14_2.bmp	131599
22	test15.bmp	693122	22	test15_2.bmp	693121

Kod 7: Przedstawienie plików przed kompresją.

Kod 8: Przedstawienie plików po dekompresji.

5.1 Przedstawienie wyników:

PLIK:	audio.wav	audio2.wav	audio3.wav	audio4.wav	highway.wav
ROZMIAR:	3009870	316002	869028	261262	8533723
SKOMPRESOWANY:	56	315928	856106	261239	8447839
DEKOMPRESJA:	3009873	316001	869034	261262	8533722
KOMPRESJA %:	0,002%	99,977%	98,513%	99,991%	98,994%

PLIK:	test.bmp	test1.bmp	test2.bmp	test3.bmp	test4.bmp
ROZMIAR:	126	442	442	442	442
SKOMPRESOWANY:	58	61	105	365	365
DEKOMPRESJA:	126	445	442	442	442
KOMPRESJA %:	46,032%	13,801%	23,756%	82,579%	82,579%

PLIK:	test5.bmp	test6.bmp	test7.bmp	test8.bmp	test9.bmp
ROZMIAR:	442	170	10922	20138	120122
SKOMPRESOWANY:	364	95	8216	15076	96180
DEKOMPRESJA:	442	170	10922	20141	120122
KOMPRESJA %:	82,353%	55,882%	75,224%	74,863%	80,069%

PLIK:	test10.bmp	test11.bmp	test12.bmp	test13.bmp	test14.bmp	test15.bmp
ROZM.:	1163198	44264	309464	16000138	131554	693122
SKOM.:	1163182	5689	85103	8440758	76746	345212
DEKO.:	1163198	44270	309464	16000138	131599	693121
KOMP. %:	99,999%	12,852%	27,500%	52,754%	58,338%	49,805%

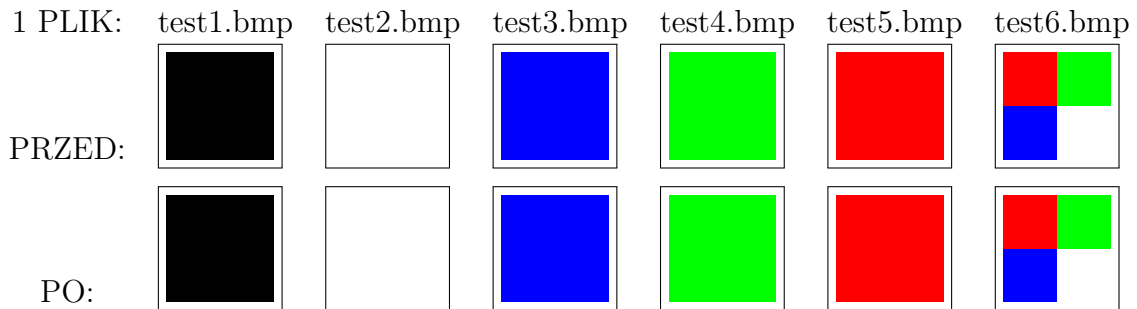


Tabela 3: Porównanie plików graficznych przed i po kompresji (odpowiednio powiększone):

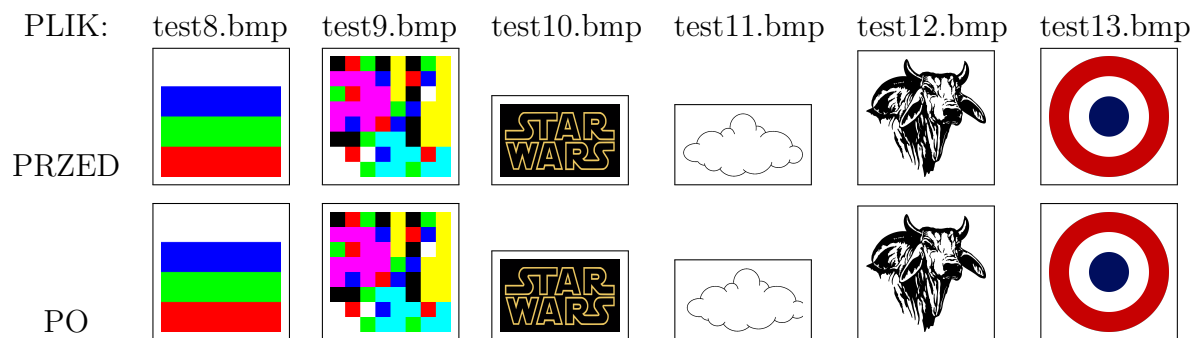


Tabela 4: Porównanie niektórych plików graficznych przed i po kompresji (odpowiednio przeskalowane):

Porównania bajtowe wykonano programem vbindiff[12].

test1.bmp									
0000 0000:	42	4D	BA	01	00	00	00	00	00
0000 0010:	00	00	0A	00	00	0A	00	00	00
0000 0020:	00	00	40	01	00	00	13	00	00
0000 0030:	00	00	00	00	00	42	47	52	73
0000 0040:	00	00	00	00	00	00	00	00	00
0000 0050:	00	00	00	00	00	00	00	00	00
0000 0060:	00	00	00	00	00	00	00	02	00
0000 0070:	00	00	00	00	00	00	00	00	00
0000 0080:	00	00	00	00	00	00	00	00	00
0000 0090:	00	00	00	00	00	00	00	00	00
0000 00A0:	00	00	00	00	00	00	00	00	00
0000 00B0:	00	00	00	00	00	00	00	00	00
0000 00C0:	00	00	00	00	00	00	00	00	00
0000 00D0:	00	00	00	00	00	00	00	00	00
0000 00E0:	00	00	00	00	00	00	00	00	00
0000 00F0:	00	00	00	00	00	00	00	00	00
test1 2.bmp									
0000 0000:	42	4D	BA	01	00	00	00	00	00
0000 0010:	00	00	0A	00	00	0A	00	00	00
0000 0020:	00	00	40	01	00	00	13	00	00
0000 0030:	00	00	00	00	00	00	42	47	52
0000 0040:	00	00	00	00	00	00	00	00	00
0000 0050:	00	00	00	00	00	00	00	00	00
0000 0060:	00	00	00	00	00	00	00	02	00
0000 0070:	00	00	00	00	00	00	00	00	00
0000 0080:	00	00	00	00	00	00	00	00	00
0000 0090:	00	00	00	00	00	00	00	00	00
0000 00A0:	00	00	00	00	00	00	00	00	00
0000 00B0:	00	00	00	00	00	00	00	00	00
0000 00C0:	00	00	00	00	00	00	00	00	00
0000 00D0:	00	00	00	00	00	00	00	00	00
0000 00E0:	00	00	00	00	00	00	00	00	00
0000 00F0:	00	00	00	00	00	00	00	00	00
Arrow keys move F find RET next difference ESC quit T move top									
C ASCII/EBCDIC E edit file G goto position Q quit B move bottom									

test1.bmp									
0000 0100:	00	00	00	00	00	00	00	00	00
0000 0110:	00	00	00	00	00	00	00	00	00
0000 0120:	00	00	00	00	00	00	00	00	00
0000 0130:	00	00	00	00	00	00	00	00	00
0000 0140:	00	00	00	00	00	00	00	00	00
0000 0150:	00	00	00	00	00	00	00	00	00
0000 0160:	00	00	00	00	00	00	00	00	00
0000 0170:	00	00	00	00	00	00	00	00	00
0000 0180:	00	00	00	00	00	00	00	00	00
0000 0190:	00	00	00	00	00	00	00	00	00
0000 01A0:	00	00	00	00	00	00	00	00	00
0000 01B0:	00	00	00	00	00	00	00	00	00
0000 01C0:	00	00	00	00	00	00	00	00	00
0000 01D0:	00	00	00	00	00	00	00	00	00
0000 01E0:	00	00	00	00	00	00	00	00	00
0000 01F0:	00	00	00	00	00	00	00	00	00
test1 2.bmp									
0000 0100:	00	00	00	00	00	00	00	00	00
0000 0110:	00	00	00	00	00	00	00	00	00
0000 0120:	00	00	00	00	00	00	00	00	00
0000 0130:	00	00	00	00	00	00	00	00	00
0000 0140:	00	00	00	00	00	00	00	00	00
0000 0150:	00	00	00	00	00	00	00	00	00
0000 0160:	00	00	00	00	00	00	00	00	00
0000 0170:	00	00	00	00	00	00	00	00	00
0000 0180:	00	00	00	00	00	00	00	00	00
0000 0190:	00	00	00	00	00	00	00	00	00
0000 01A0:	00	00	00	00	00	00	00	00	00
0000 01B0:	00	00	00	00	00	00	00	00	00
0000 01C0:	00	00	00	00	00	00	00	00	00
0000 01D0:	00	00	00	00	00	00	00	00	00
0000 01E0:	00	00	00	00	00	00	00	00	00
0000 01F0:	00	00	00	00	00	00	00	00	00
Arrow keys move F find RET next difference ESC quit T move top									
C ASCII/EBCDIC E edit file G goto position Q quit B move bottom									

Zrzut ekranu 12: Porównanie różnic dwóch tablic bajtowych przed kompresją i po dekompresji.

audio2.wav									
0000 0000:	52	49	46	46	5A	D2	04	00	57
0000 0010:	10	00	00	00	01	00	02	00	44
0000 0020:	04	00	10	00	64	61	74	61	70
0000 0030:	35	00	31	00	39	00	78	00	41
0000 0040:	52	00	A8	00	01	00	B5	00	00
0000 0050:	92	00	40	01	07	00	01	00	00
0000 0060:	87	00	3D	02	DA	00	57	02	FF
0000 0070:	2A	01	7E	03	33	01	C7	03	44
0000 0080:	93	01	E6	04	D5	01	71	05	03
0000 0090:	7B	02	09	07	BE	02	C3	07	ED
0000 00A0:	5B	03	F2	08	E2	03	28	0A	1A
0000 00B0:	D8	04	55	0C	7D	04	2E	0B	EB
0000 00C0:	AB	FC	F4	F2	32	FC	B6	F0	7C
0000 00D0:	2E	FC	42	EF	DC	FB	C7	ED	E9
0000 00E0:	BE	FB	87	EA	8B	FB	9D	E9	9C
0000 00F0:	DF	FB	7B	E6	E8	FB	E8	E5	11
audio2 2.wav									
0000 0000:	52	49	46	46	5A	D2	04	00	57
0000 0010:	10	00	00	00	01	00	02	00	44
0000 0020:	04	00	10	00	64	61	74	61	70
0000 0030:	35	00	31	00	39	00	78	00	41
0000 0040:	52	00	A8	00	01	00	B5	00	00
0000 0050:	92	00	40	01	07	00	01	00	00
0000 0060:	87	00	3D	02	DA	00	57	02	FF
0000 0070:	2A	01	7E	03	33	01	C7	03	44
0000 0080:	93	01	E6	04	D5	01	71	05	03
0000 0090:	7B	02	09	07	BE	02	C3	07	ED
0000 00A0:	5B	03	F2	08	E2	03	28	0A	1A
0000 00B0:	D8	04	55	0C	7D	04	2E	0B	EB
0000 00C0:	AB	FC	F4	F2	32	FC	B6	F0	7C
0000 00D0:	2E	FC	42	EF	DC	FB	C7	ED	E9
0000 00E0:	BE	FB	87	EA	8B	FB	9D	E9	9C
0000 00F0:	DF	FB	7B	E6	E8	FB	E8	E5	11
Arrow keys move F find RET next difference ESC quit T move top									
C ASCII/EBCDIC E edit file G goto position Q quit B move bottom									

audio2.wav									
0004 D200:	00	40	00	00	00	5C	00	00	00
0004 D210:	6B	2C	54	50	45	31	00	00	14
0004 D220:	65	77	61	76	65	73	61	6D	70
0004 D230:	54	44	52	43	00	00	00	05	00
0004 D240:	49	54	32	00	00	00	19	00	00
0004 D250:	20	54	47	31	30	30	20	53	71
0004 D260:	43	30	71	75	61	72	65	4C	64
0004 D270:	00	00	00	00	00	00	00	00	00
0004 D280:	00	00	00	00	00	00	00	00	00
0004 D290:	00	00	00	00	00	00	00	00	00
0004 D2A0:	00	00	00	00	00	00	00	00	00
0004 D2B0:	00	00	00	00	00	00	00	00	00
0004 D2C0:	00	00	00	00	00	00	00	00	00
0004 D2D0:	00	00	00	00	00	00	00	00	00
0004 D2E0:	00	00	00	00	00	00	00	00	00
0004 D2F0:	00	00	00	00	00	00	00	00	00
audio2 2.wav									
0004 D200:	00	40	00	00	00	5C	00	00	00
0004 D210:	6B	2C	54	50	45	31	00	00	14
0004 D220:	65	77	61	76	65	73	61	6D	70
0004 D230:	54	44	52	43	00	00	00	05	00
0004 D240:	49	54	32	00	00	00	19	00	00
0004 D250:	20	54	47	31	30	30	20	53	71
0004 D260:	43	30	71	75	61	72	65	4C	64
0004 D270:	00	00	00	00	00	00	00	00	00
0004 D280:	00	00	00	00	00	00	00	00	00
0004 D290:	00	00	00	00	00	00	00	00	00
0004 D2A0:	00	00	00	00	00	00	00	00	00
0004 D2B0:	00	00	00	00	00	00	00	00	00
0004 D2C0:	00	00	00	00	00	00	00	00	00
0004 D2D0:	00	00	00	00	00	00	00	00	00
0004 D2E0:	00	00	00	00	00	00	00	00	00
0004 D2F0:	00	00	00	00	00	00	00	00	00
Arrow keys move F find RET next difference ESC quit T move top									
C ASCII/EBCDIC E edit file G goto position Q quit B move bottom									

Zrzut ekranu 13: Porównanie różnic dwóch tablic bajtowych przed kompresją i po dekompresji.

htghway.wav																			
0000 0000:	52	49	46	46	46	00	90	01	57	41	56	45	66	60	74	20	RIFFF...	WAVEfmt	
0000 0010:	10	00	00	00	01	02	00	02	44	AC	00	00	10	B1	02	00	D.....	
0000 0020:	04	00	10	00	4C	49	53	54	1A	00	00	00	49	4E	46	4FLISTINFO	
0000 0030:	49	53	46	54	0E	00	00	00	4C	61	76	66	35	36	2E	33	ISFT....	Lavf56.3	
0000 0040:	30	2E	31	30	30	00	64	61	74	61	00	00	90	01	00	00	0.100.da	ta.....	
0000 0050:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 0060:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 0070:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 0080:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 0090:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 00A0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 00B0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 00C0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 00D0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 00E0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 00F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
highway 2.wav																			
0000 0000:	52	49	46	46	46	00	90	01	57	41	56	45	66	60	74	20	RIFFF...	WAVEfmt	
0000 0010:	10	00	00	00	01	02	00	02	44	AC	00	00	10	B1	02	00	D.....	
0000 0020:	04	00	10	00	4C	49	53	54	1A	00	00	00	49	4E	46	4FLISTINFO	
0000 0030:	49	53	46	54	0E	00	00	00	4C	61	76	66	35	36	2E	33	ISFT....	Lavf56.3	
0000 0040:	30	2E	31	30	30	00	64	61	74	61	00	00	90	01	00	00	0.100.da	ta.....	
0000 0050:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 0060:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 0070:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 0080:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 0090:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 00A0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 00B0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 00C0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 00D0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 00E0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000 00F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
Arrow keys move C ASCII/EBCDIC F find E edit file RET next difference G goto position ESC quit Q quit T move top B move bottom																			

htghway.wav									
0000 0000:	89 F3 F8 0F 73 F5 EF 15	83 FB 8A 17 AF FD 5F 13S....					
0000 0010:	5A FA 4C 11 B7 F8 90 14	26 FC 17 1B 03 03 7A 23	Z.L.....	&.....Z#					
0000 0020:	C8 0B 4C 29 FA 11 79 29	F8 12 8B 27 87 12 32 27	..L...y)	...'..2'					
0000 0030:	10 14 E7 26 46 16 B1 23	43 16 C9 20 6A 16 B8 23	...&F..#	C...J..#					
0000 0040:	EA 1A 3B 28 18 23 71 31	60 2B 2D 33 F5 30 38 34	...+.#q1	+3.084					
0000 0050:	C8 35 1C 36 06 3A 06 33	86 3B FB 2E 0A 3C 45 30	5.6...3	...<E0					
0000 0060:	9F 41 B6 2E EC 43 28 28	30 41 2D 23 69 3F C1 1F	A...C((0A...l?..					
0000 0070:	CC 3D A2 1C C7 3B 45 16	3D 37 6B 0D AE 30 72 0B	...=;;E.	=7k...0r.					
0000 0080:	8D 2F 31 0B 48 2E 8F 01	B6 23 01 F6 F2 17 D9 F3	/1.H....	..#.....					
0000 0090:	29 14 44 F9 86 15 58 FA	21 12 B0 F2 FF 06 F6 EE)D...X.	l.....					
0000 00A0:	A3 FF B5 F3 DA FF 64 F3	93 FB 6E E7 1A ED 7A D9d.	l.n...z.					
0000 00B0:	62 DC 0D D7 5F D5 40 E0	63 D9 4F E7 69 DC 41 EA	b.....	q. c.O.l.A.					
0000 00C0:	CA DB D6 F2 32 E0 5E FC	22 E6 F4 FD 2D E6 87 FC2..A.	".....					
0000 00D0:	9D E4 14 FA 40 E2 A5 F4	22 DD@... "	"..					
0000 00E0:									
0000 00F0:									
htghway.2.wav									
0000 0000:	89 F3 F8 0F 73 F5 EF 15	83 FB 8A 17 AF FD 5F 13S....					
0000 0010:	5A FA 4C 11 B7 F8 90 14	26 FC 17 1B 03 03 7A 23	Z.L.....	&.....Z#					
0000 0020:	C8 0B 4C 29 FA 11 79 29	F8 12 8B 27 87 12 32 27	..L...y)	...'..2'					
0000 0030:	10 14 E7 26 46 16 B1 23	43 16 C9 20 6A 16 B8 23	...&F..#	C...J..#					
0000 0040:	EA 1A 3B 28 18 23 71 31	60 2B 2D 33 F5 30 38 34	...+.#q1	+3.084					
0000 0050:	C8 35 1C 36 06 3A 06 33	86 3B FB 2E 0A 3C 45 30	5.6...3	...<E0					
0000 0060:	9F 41 B6 2E EC 43 28 28	30 41 2D 23 69 3F C1 1F	A...C((0A...l?..					
0000 0070:	CC 3D A2 1C C7 3B 45 16	3D 37 6B 0D AE 30 72 0B	...=;;E.	=7k...0r.					
0000 0080:	8D 2F 31 0B 48 2E 8F 01	B6 23 01 F6 F2 17 D9 F3	/1.H....	..#.....					
0000 0090:	29 14 44 F9 86 15 58 FA	21 12 B0 F2 FF 06 F6 EE)D...X.	l.....					
0000 00A0:	A3 FF B5 F3 DA FF 64 F3	93 FB 6E E7 1A ED 7A D9d.	l.n...z.					
0000 00B0:	62 DC 0D D7 5F D5 40 E0	63 D9 4F E7 69 DC 41 EA	b.....	q. c.O.l.A.					
0000 00C0:	CA DB D6 F2 32 E0 5E FC	22 E6 F4 FD 2D E6 87 FC2..A.	".....					
0000 00D0:	9D E4 14 FA 40 E2 A5 F4	22 DD@... "	"..					
0000 00E0:									
0000 00F0:									
Arrow keys move F find RET next difference ESC quit T move top									
C ASCII/EBCDIC E edit file G goto position Q quit B move bottom									

Zrzut ekranu 14: Porównanie różnic dwóch tablic bajtowych przed kompresją i po dekompresji.

6 Wnioski

Algorytm **Run-Length Encoding** doskonale się sprawdza do kompresji plików graficznych monochromatycznych (jedno kolorowych) - kompresuje pliki na poziomie 20 – 50%. Przy bardziej złożonych - bardziej kolorowych - plikach graficznych kompresja spada do poziomu 80 – 99%. Pliki dźwiękowe, z racji swojej zawłości również nie są podatne na kompresję tym algorytmem.

Algorytm jest prosty w implementacji i obsłudze oraz dla niektórych plików może dać „zachwycające” rezultaty - kompresja na poziomie tysięcznych części pliku wejściowego. Jednakże nie gwarantuje on „oszczędności” miejsca - kompresji.

W procesie kompresji / dekompresji błędy mogą powstać jedynie w skomplikowanych plikach o dużej złożoności pamięciowej / kolorystycznej / bajtowej, w innych nie powstały żadne znaczące wyjątki. „Przeinaczenia” bajtowe powstałe wskutek „wypełnienia” bajtami z dekompresowanych plików nie wpływają na obiekt końcowy.

Literatura

- [1] Kurs L^AT_EX w π^e minut http://www.fuw.edu.pl/~kostecki/kurs_latexa.pdf.
- [2] Program Texmaker 4.5 <http://www.xmlmath.net/texmaker/>.
- [3] ShareLaTeX online L^AT_EX editor <https://www.sharelatex.com/>.
- [4] Qt Creator – wieloplatformowe środowisko programistyczne <http://www.qt.io/>.
- [5] GCC, the GNU Compiler Collection <https://gcc.gnu.org/>.
- [6] Git rozproszony system kontroli wersji <http://git-scm.com/>.
- [7] GitHub Web-based Git repository <https://github.com/>.
- [8] Run-length encodings - S. W. Golomb (1966); IEEE Trans Info Theory 12(3):399 http://urchin.earth.li/~twic/Golombs_Original_Paper/.

- [9] Variable-length codes for data compression / David Salomon, London : Springer, 2007.
- [10] The largest database of free vector icons - flaticon <http://www.flaticon.com/>.
- [11] Sample WAV files <http://download.wavetlan.com/SVV/Media/HTTP/http-wav.htm>.
- [12] vbindiff - hexadecimal file display and comparison <http://manpages.ubuntu.com/manpages/hardy/man1/vbindiff.1.html>.