SM LAB_3 Sprawozdanie
Lewicki Maciej

1. Do zaimplementowania:
    1. Metoda μ-law (0.2 pkt)
    2. Metodę DPCM bez predykcji (0.2 pkt)

μ-law –

```
def encode_mu_law(_data, u):
    assert _data.min() >= -1 and _data.max() <= 1 and np.issubdtype(_data.dtype, np.floating)
    out_val = np.sign(_data) * (np.log(1+u*np.abs(_data)) / np.log(1+u))
    return out_val


def decode_mu_law(_data, u):
    assert _data.min() >= -1 and _data.max() <= 1 and np.issubdtype(_data.dtype, np.floating)
    out_val = np.sign(_data) * (1/u) * (np.power(1+u, np.abs(_data))-1)
    return out_val
```

DPCM bez predykcji –

```
def encode_DPCM(_data, bits):
    out_val = _data.copy()
    E = out_val[0]
    for x in range(1, _data.shape[0]):
        diff = _data[x] - E
        diff = quantize2(diff, bits, _data.min(), _data.max())
        out_val[x] = diff
        E += diff
    return out_val


def decode_DPCM(_data):
    out_val = _data.copy()
    for x in range(1, _data.shape[0]):
        out_val[x] = out_val[x - 1] + _data[x]
    return out_val
```

2. Przebadać wpływ metod kompresji na jakość plików dźwiękowych Na podstawie kilku różnych plików *sing_* (0.6 pkt)
    1. Przeanalizować działanie obu metod przemyśleć i krótko opisać własnymi słowami ich działanie. (Co ulega kompresji etc.)
    2. Zbadać jakość dźwięku po działaniu domyślnym obu metod (8 bitów).
    3. Sprawdzić czy i do jak niskiej ilości bitów informacji jesteśmy w stanie skompresować nasz sygnał, aby być w stanie jeszcze rozpoznać jego zawartość. Zacząć od 8 bitów i obniżać ich ilość aż do momentu, którym przestajemy rozpoznawać zawartość.

2.1

DPCM – kompresja bez predykcji polega na zapisaniu różnic pomiędzy kolejnymi wartościami, które następnie są kwantyzowane do żądanej bitowości. Dekompresja polega na odtworzeniu wartości na podstawie sumy skompresowanych różnic. Kompresji ulegają tylko różnice pomiędzy kolejnymi wartościami. Kompresja nadaje się do sygnałów w których kolejne próbki nie różnią się od siebie znacząco i różnice pomiędzy nimi można zmieścić w mniejszej ilości bitów.

µ-law – kompresja polega na zmniejszeniu „rozpiętości tonalnej” czyli sprowadza małe i duże wartości do jednego zakresu (-1, 1). Metoda używa do kompresji odwracalnej funkcji, więc dekompresja polega na przepuszczeniu sygnału przez funkcję odwrotną.

2.2/2.3

	sing_low1.wav	
bit	DPCM	µ-law
8	nie słychać różnicy	nie słychać różnicy
6	nie słychać różnicy	wyraźny szum w tle
4	mocny szum	bardzo silny szym i zniekształcony dźwięk
2	bardzo zniekształcony dźwięk + szum	bardzo zniekształcony dźwięk + szum

	sing_medium1.wav	
bit	DPCM	µ-law
8	delikatny szum	nie słychać różnicy
6	wyraźny szum + drobne zniekształcenia	wyraźny szum
4	bardzo zniekształcony dźwięk + szum	głośny szum + trochę zniekształcony dźwięk
2	bardzo zniekształcony dźwięk + trzaski	bardzo zniekształcony dźwięk

	sing_high1.wav	
bit	DPCM	µ-law
8	nie słychać różnicy	nie słychać różnicy
6	wyraźny szum	nie słychać różnicy
4	mocne zniekształcenia	mocne zniekształcenia + szum
2	trzaski+zmiany głośności+mocne zniekształcenia	mocne zniekształcenia + głośny szum

2.2

Dźwięki skompresowane do 8 bitów są bardzo podobne do oryginałów. Lepiej spisuje się kompresja μ-law w której nie było słyszalnych różnic od oryginały. Kompresja DPCM przy niektórych próbkach powodowała delikatne szumy.

2.3

Generalnie najlepiej poradził sobie algorytm μ-law który dla 8 bitów osiągał jakość podobną do oryginału, dla 6 bitów dźwięk był dalej bardzo rozpoznawalny, ale często występowały szumy. Poniżej 6 bitów dźwięk staje się bardzo zniekształcony a szumy głośniejsze.

DPCM dla niektórych próbek przy kompresji do 8 bitów wykazywał szumy, z kolei poniżej pojawiały się już zniekształcenia dźwięku i trzaski. Dla niższych bitowości zniekształcenia są silniejsze niż w μ-law chociaż pojawia się odrobinę mniej szumu.