Brezizgubno stiskanje digitalnega avdia s prileganjem daljic in kvadratnih Bézierovih krivulj

Luka Železnik, Damjan Strnad, Borut Žalik, David Podgorelec

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko
E-pošta: luka.zeleznik@student.um.si

Lossless Digital Audio Compression by Fitting Line Segments and Quadratic Bézier Curves

In this paper we survey existing lossless audio formats, their way of predicting audio samples and encoding differences between the samples and predicted values. The considered formats are Shorten, FLAC, WavPack and MPEG-4 ALS. Then we propose a different way of predicting the audio samples, based on quadratic Bézier curves and line segments. Finally, we compare the compression ratios and the encoding and decoding speeds of the existing and our prototype system.

1 Uvod

Stiskanje avdia je bilo in je še vedno aktualen izziv. Čeprav je avdio eden izmed najbolj kompaktnih tipov multimedije, je bilo njegovo stiskanje zaradi kopiranja posnetkov na optične diske in pošiljanja prek medmrežja zelo aktualno že na prelomu tisočletja. Prav v tem času so bili zasnovani prvi formati za stiskanje avdia. Kasneje se je zaradi boljših in cenovno bolj dostopnih zvočnikov in slušalk pojavila potreba po višji kvaliteti avdioposnetkov. Arhiviranje digitalnega avdia postaja vedno bolj pomembno, avdiofili pa prisegajo na izvorne, nedegradirane skladbe. Vendar so slednje prostorsko zelo potratne, saj splošni brezizgubni algoritmi stiskanja ne zagotavljajo želene kakovosti. Rešitev so namenski algoritmi za brezizgubno stiskanje avdia. V tem prispevku bomo povzeli opise najbolj razširjenih tovrstnih metod in predlagali lasten postopek, temelječ na prileganju avdiosignala daljicam in kvadratnim Bézierovim krivuljam. Prototipno implementacijo bomo primerjali z obstoječimi metodami glede na stopnjo stiskanja in hitrost kodiranja ter dekodiranja.

2 Pregled obstoječih metod brezizgubnega stiskanja avdia

Kljub manj raziskanemu področju brezizgubnega stiskanja avdia napram izgubnemu obstaja kar nekaj brezizgubnih algoritmov. Za določitev smiselnosti naše metode jih je potrebno razumeti in njihove lekcije upoštevati v naši implementaciji. Predstavili bomo najpogosteje uporabljane prostodostopne metode za brezizgubno stiskanje avdia, to so Shorten, FLAC, WavPack in MPEG4-ALS.

2.1 Shorten

Shorten [1] je najstarejši izmed predstavljenih formatov in je bil zamišljen za zapis zvoka v kakovosti CD-ROM (stereo, 44,1 kHz, 16-bitni).

Prvi korak postopka je razdelitev avdia na bloke dolžine 256 vzorcev. V postopku poskušamo interpolirati točko (amplitudo naslednjega vzorca) iz amplitud prejšnjih vzorcev. Napoved naslednjega vzorca $\hat{s}(t)$ je linearna kombinacija p prejšnjih amplitud $s(t)$, ki jim dodelimo uteži a_i (1).

$$\hat{s}(t) = \sum_{i=1}^{p} a_i \, s(t-i) \qquad (1)$$

V običajni implementaciji srečamo štiri napovedi $\hat{s}_p(t)$, $0 \leq p \leq 3$ [2]:

1. $\hat{s}_0(t) = 0$,
2. $\hat{s}_1(t) = s(t-1)$,
3. $\hat{s}_2(t) = 2s(t-1) - s(t-2)$,
4. $\hat{s}_3(t) = 3s(t-1) - 3s(t-2) + s(t-3)$.

Algoritem izračuna odstopanja izbranih napovedi od signala in izbere najboljšo. Tem odstopanjem bomo v nadaljevanju rekli preostanki, saj gre za informacije, ki jih je treba shraniti po uporabi napovedi. S parametri lahko upravljamo hitrost algoritma, tako da zmanjšamo iskalno območje – manj stopenj polinoma. Preostanke kodiramo z Golomb-Riceovimi kodami [2].

2.2 FLAC

FLAC [3] je novejši odprtokodni format za brezizgubno kodiranje avdia. Je naslednik formata Shorten, saj v veliki meri uporablja podobne principe, le da ima več načinov stiskanja blokov in več metapodatkov, ki naredijo format bolj fleksibilen.

Avdiokanale najprej pretvori v srednji ($srednji = (levi + desni)/2$) in stranski ($stranski = levi - desni$) kanal. S tem poskuša izkoristiti običajno korelacijo med kanali.

Za obdelavo blokov obstajajo štiri funkcije – prepisovalna, konstantna, linearna in linearna napoved FIR (končni impulzni odziv, angl. finite impulse response) [2], ki imajo ločene kodirnike preostankov. Pri prepisovalni je napoved vedno 0, kar pomeni, da kodiramo sam vzorec. Konstantna napoved je za kodiranje blokov tišine. Linearna napoved vključuje vse štiri napovedi algoritma Shorten, ob le-teh pa še peto napoved $\hat{s}_4(t) = 4s(t-1) - 6s(t-2) + 4s(t-3) -$

$s(t-4)$. Linearna napoved FIR je naprednejša in uporabi rekurzijo Levinsona in Durbina [4] za določitev optimalnih uteži v (1), kjer je p vhodni parameter algoritma. FLAC uporablja za kodiranje preostankov Golomb-Riceove kode [2].

2.3 WavPack

WavPack [5] omogoča obdelavo avdiosignalov, katerih vzorci so veliki do 32 bitov, vključno s števili s plavajočo vejico. Format je odprt in prenosljiv ter omogoča dober kompromis med stopnjo in hitrostjo stiskanja [2].

Prvi korak algoritma je izračunati srednji in stranski kanal, podobno kot pri formatu FLAC. Nato poskusi algoritem predvideti naslednji vzorec z napovednim filtrom. Posebnost je, da stopnjo stiskanja in hitrost algoritma kontroliramo s številom prehodov filtra. Obstaja 13 variacij filtrov za napoved $\hat{s}(t)$, ki se delijo na tri podskupine:

- 8 napovedi na podlagi prejšnjih vzorcev,
- 2 linearni kombinaciji zadnjih dveh vzorcev,
- 3 napovedi na podlagi drugega avdiokanala.

Po izbiri najboljšega filtra izračunamo preostanek $e(t) = s(t) - w(t)\hat{s}(t)$, kjer je $w(t)$ dinamična utež filtra za vzorec $s(t)$. Utež se nato posodobi po enačbi (2), kjer je parameter d majhno pozitivno celo število.

$$w(t+1) = w(t) + d \cdot sgn\big(\hat{s}(t)\big) \cdot sgn\big(e(t)\big) \quad (2)$$

Preostanek napovedi zakodiramo s prilagojenimi Golombovimi kodami, kjer morebitne negativne preostanke pretvorimo v nenegativne vrednosti, t.j. $e(t) < 0 \Longrightarrow e(t) = -(e(t)+1)$, predznak pa se priloži na konec zakodirane vrednosti [2].

2.4 MPEG-4 Audio Lossless Coding (ALS)

MPEG-4 ALS je eden izmed avdiokodekov iz družine MPEG-4 [6]. Avdio razdeli na bloke, omogoča tudi stiskanje formata s plavajočo vejico [2].

Napoved za posamezen vzorec je kombinacija kratkoročne (FIR linearna napoved) in dolgoročne linearne napovedi skupaj z večkanalnim kodiranjem.

Pri vzorčni frekvenci 48 kHz vsebuje blok 2048 vzorcev. Po potrebi lahko kodirnik ta blok razpolovi na 1024, 512 itd. vzorcev, da omogoči karseda dobre lastnosti za stiskanje. Te manjše bloke nato poljubno sestavi v blok izvorne velikosti.

Koncept dolgoročne napovedi sloni na višjih harmonikih, ki so večkratniki osnovne frekvence in jih dobro poznamo pri vseh glasbilih, pojavljajo pa se tudi v drugih avdiosignalih in prispevajo h korelaciji podnizov vzorcev znotraj le-teh [2]. Toda te korelacije so ločene z več 100 vzorci, zato uporabimo za dopolnitev kratkoročne napovedi $\hat{s}(t)$ v končno napoved $\varepsilon(t)$ enačbo (3).

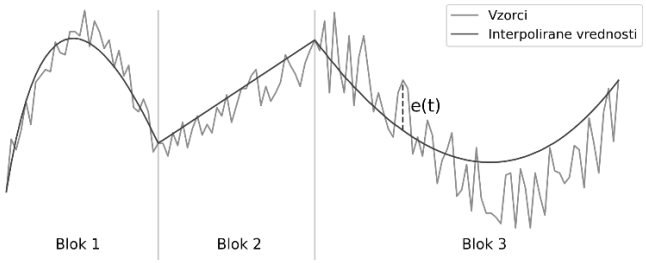
$$\varepsilon(t) = \hat{s}(t) - \sum_{j=-2}^{2} \gamma_{r+j} \cdot \hat{s}(t - r + j) \quad (3)$$

V enačbi je r zamik, ki je določen glede na prisotno frekvenco in hitrost vzorčenja, γ pa kvantizirana utež.

Razlike se kodirajo z Golomb-Riceovimi kodami ali z BGMC (Block Gilbert-Moore Codes) [6].

3 Opis naše metode

Naša metoda ne napoveduje posameznih vzorcev, ampak daljše sekvence (bloke) vzorcev, ki jih prilegamo krivuljam iz vnaprej določenega nabora. Trenutno operiramo s 16 napovedmi, ki jih določajo daljice dolžin 1, 2, 4, 8, 16, 32, 64 in 128 vzorcev ter Bézierove krivulje 2. reda [7], ki se raztezajo čez 32, 64, 128, 256, 512, 1024, 2048 ali 4096 vzorcev. Krajišči daljice ali robni kontrolni točki krivulje se ujemata z ustreznima vzorcema krivulje, za vmesne vzorce pa kodiramo lokalne preostanke napovedi, t.j. odstopanja od krivulje ali daljice. Opcijsko lahko namesto odstopanj uporabimo tudi razlike med zaporednimi odstopanji. V primeru rabe krivulje moramo pred tem določiti srednjo kontrolno točko, pri čemer uporabimo postopek štiritočkovne parabolične interpolacije [8]. Primer blokov vidimo na Sliki 1. Razlike nato kodiramo z metodo BASC (Binary adaptive sequence coding) [9].



Slika 1. Prikaz blokov in lokalnih preostankov (e(t))

3.1 Štiritočkovna parabolična interpolacija

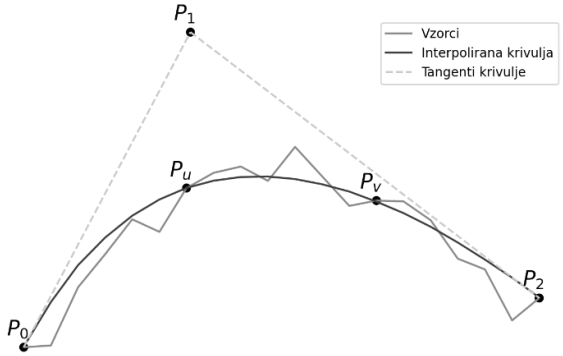
Za določitev zaporedja točk Bézierove krivulje 2. reda, t.j. paraboličnega loka, ki jih bomo primerjali z ustreznimi vzorci avdiosignala, potrebujemo tri kontrolne točke P_0, P_1 in P_2. Točke vzdolž krivulje potem določa enačba (4).

$$B(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2, 0 \le t \le 1 (4)$$

Očitno velja $B(0) = P_0$ in $B(1) = P_2$, zato je s ciljem čim boljšega prileganja smiselno izbrati začetno in končno kontrolno točko kar med vzorci avdiosignala. Vmesna kontrolna točka P_1 pa ne leži na krivulji in jo je treba določiti čim bolj optimalno glede na navedeni cilj. Optimizacijo lahko izvedemo na različne načine, v trenutnem prototipu pa smo izbrali štiritočkovno interpolacijo [8]. V intervalu med P_0 in P_2 izberemo še dva vzorca avdiosignala, t.j. točki $P_u = B(t_u)$ in $P_v = B(t_v)$, kjer zahtevamo popolno prileganje h krivulji. Štiri

točke paraboličnega loka $P_0(x_0, y_0)$, $P_u(x_u, y_u)$, $P_v(x_v, y_v)$ in $P_2(x_2, y_2)$ omogočajo enolično določitev vmesne kontrolne točke $P_1(x_1, y_1)$ ob izpolnitvi pogoja, da je štirikotnik $P_0P_uP_vP_2$ izbočen. V kolikor pogoj ni izpolnjen, je še vedno na voljo 15 preostalih napovedi, med katerimi so tisti, ki temeljijo na daljicah, brezpogojno uporabni.

Točki P_u *in* P_v lahko izberemo na različne načine. V naši pototipni implementaciji smo ju glede na časovni potek signala enakomerno razporedili med robni točki P_0 *in* P_2, kar pomeni: $x_u = x_0 + \left\lfloor \frac{1}{3}(x_2 - x_0) \right\rfloor$ in $x_v = x_0 + \left\lfloor \frac{2}{3}(x_2 - x_0) \right\rfloor$. Slika 2 predstavlja lege kontrolnih in interpoliranih točk krivulje.



Slika 2. Interpolirana krivulja glede na vzorce in točke

Neznane vrednosti t_u, t_v, x_1 in y_1 bi lahko direktno določili s sistemom štirih enačb (4) ločeno za x_u, y_u, x_v in y_v, a smo rajši izbrali postopek s substitucijo spremenljivk, opisan v [8], ki se je izkazal za računsko enostavnejšega. Prvi korak je najti takšna h in k, da velja (5):

$$P_v(x_v, y_v) = P_0 + h(P_u - P_0) + k(P_2 - P_0) = (1 - h - k)P_0 + hP_u + kP_2. \quad (5)$$

Z rešitvijo sistema dveh enačb (za x_v in y_v) z dvema neznankama dobimo (6) in (7).

$$k = \frac{x_v - x_0 - h(x_u - x_0)}{x_2 - x_0} \quad (6)$$

$$h = \frac{(x_2 - x_0)(y_v - y_0) - (x_v - x_0)(y_2 - y_0)}{(x_2 - x_0)(y_u - y_0) - (x_u - x_0)(y_2 - y_0)} \quad (7)$$

Ko v (5) upoštevamo $P_v = B(t_v)$ in $P_u = B(t_u)$, vstavimo ustrezna izraza iz (4) ter izenačimo koeficiente pri P_0, P_1 in P_2 na obeh straneh enačbe, dobimo sistem treh enačb (8).

$$\begin{aligned} (1 - t_v)^2 &= 1 - h - k + h(1 - t_u)^2 \\ 2t_v(1 - t_v) &= 2ht_u(1 - t_u) \\ {t_v}^2 &= h{t_u}^2 + k \end{aligned} \quad (8)$$

Po eliminaciji t_v dobimo kvadratno enačbo za t_u (9).

$$h(1 - h){t_u}^2 - 2hkt_u + k(1 - k) = 0 \quad (9)$$

Rezultat kvadratne enačbe t_u (med 0 in 1) vstavimo v enačbo (4) za $B(t_u)$, da dobimo kontrolno točko P_1 (10):

$$P_1 = \frac{P_u - (1 - t_u)^2 P_0 - {t_u}^2 P_2}{2t_u(1 - t_u)} \quad (10)$$

3.2 Izbiranje najustreznejše krivulje

Kot rečeno, ima v obravnavanem vzorcu signala kodirnik na izbiro 16 napovedi, 8 za prileganje k daljicam in 8 za prileganje k Bézierovim krivuljam. Izbere tistega z najnižjim povprečnim številom bitov na vzorec ter se po zapisu kode pomakne za ustrezno število vzorcev naprej. Prvi vzorec v avdiokanalu vedno prilegamo daljici dolžine 1, pri nadaljnjih blokih pa je treba upoštevati, da zapis daljice vsebuje amplitudo v končni točki in zaporedje lokalnih preostankov, zapis krivulje pa poleg tega še vmesno kontrolno točko P_1.

3.3 Kodiranje preostankov

Preostanke kodiramo s prilagojeno metodo BASC [9], ki omogoča kodiranje negativnih števil. Stiskanje je najbolj optimalno, če so razlike med interpoliranimi in dejanskimi vrednostmi razporejene po gručah. Algoritem predpostavi, da bo za zapis danega števila potreboval enako število bitov kot za zapis njegovega predhodnika. To predpostavko nato popravimo s predložnimi kodami.

Za dano število x moramo najprej izračunati minimalno število bitov, ki jih potrebujemo za njegov zapis. Temu številu bomo rekli b'. Velja enačba (11):

$$b' = \begin{cases} 1 + \lceil \log_2(-x) \rceil; & x < 0 \\ 0; & x = 0 \\ 2 + \lfloor \log_2(x) \rfloor; & x > 0 \end{cases}, x \in \mathbb{Z} \quad (11)$$

Za zapis števila potrebujemo tudi b, ki je b' prejšnjega števila v zaporedju. Število x zapišemo v načinu dvojiškega komplementa. Če velja $b' \leq b$, potem zapišemo x kot $0\ \underbrace{zapis\ x}_{b}$. Če velja $b' > b$, pa zapišemo x kot $\underbrace{1 \ldots 1}_{b'-b}\ 0\ \underbrace{zapis\ x\ brez\ MSB}_{b'-1}$. Najpomembnejši bit MSB pri tem zapisu ni potreben, saj velja $x > 0 \Rightarrow MSB = 1$; $x < 0 \Rightarrow MSB = 0$. Še vedno pa moramo zapisati predznak števila x.

4 Primerjava metod

Prototip našega algoritma smo primerjali s formati, obravnavanimi v poglavju 2, in sicer: shorten version 3.6.1 (shn), flac 1.3.4 (flac), WAVPACK Win64 Version 5.4.0 (wv) in MPEG-4 Audio Lossless Coding (ALS), Reference Model Codec Version 23.

Za testne vzorce smo izbrali glasbo iz različnih žanrov, vsi posnetki imajo vzorčno frekvenco 44,1 kHz, so 16-bitni in nestisnjeni. To so bili zvok iz filma Elephants Dream (dolžina 10 min 59 s) [10], rock skladba Images (dolžina 4 min 9 s) [11], orkestralna zasedba the Nymphaeum part I & II (dolžina 11 min 1 s) [12] in klavirska balada What a beautiful Sunset (Extended edition) (dolžina 4 min 45 s) [13]. Ker naš prototip trenutno še ne vključuje koreliranosti med stereokanaloma, smo opravili tudi primerjavo stopnje stiskanja z mono verzijami obravnavanih formatov.

Tabela 1. Stopnja stiskanja

Žanr	naš	shn	flac	wv	ALS
Film (mono)	1,81	1,97	2,09	2,10	2,11
Film (stereo)	1,78	1,94	2,14	2,13	2,16
Rock (mono)	1,36	1,45	1,51	1,52	1,53
Rock (stereo)	1,34	1,44	1,54	1,54	1,56
Orkester (mono)	1,57	1,70	1,81	1,81	1,84
Orkester (stereo)	1,55	1,68	1,83	1,82	1,85
Balada (mono)	1,59	1,73	1,81	1,81	1,84
Balada (stereo)	1,57	1,70	1,80	1,78	1,83

Tabela 1 prikazuje stopnje stiskanja obravnavanih formatov. $Stopnja\ stiskanja = \frac{Razširjena\ velikost}{Stisnjena\ velikost}$, zato so višje stopnje boljše. Naš algoritem se po pričakovanju obnese nekoliko slabše od ostalih, saj gre šele za prvi prototip in že zdaj vidimo precej možnosti za izboljšave, ki jih omenjamo v zaključnem poglavju 5. Kljub implementaciji, ki je še daleč od optimalne, pa gre za odstopanje zgolj za nekaj odstotkov.

Tabela 2. Čas kodiranja v sekundah (stereo)

Žanr	naš	shn	flac	wv	ALS
Film	28,53	3,19	1,33	2,39	6,52
Rock	11,56	1,30	0,53	0,94	2,48
Orkester	29,46	3,71	1,99	3,00	7,35
Balada	12,89	1,66	0,72	1,40	3,44

Občutno slabše se naš algoritem obnese glede na čas kodiranja (tabela 2), saj zaostaja za obstoječimi metodami za faktor 4-19. Čas kodiranja je pribl. 4,4 % časa predvajanja pri vzorčni frekvenci 44100 Hz. Čeprav to ni idealno, smatramo čas kodiranja za najmanj pomemben kriterij ocene potenciala metod stiskanja.

Tabela 3. Čas dekodiranja v sekundah (stereo)

Žanr	naš	shn	flac	wv	ALS
Film	11,08	2,01	0,85	1,65	3,15
Rock	4,46	0,80	0,35	0,80	1,20
Orkester	11,23	2,41	0,94	2,87	3,17
Balada	4,85	0,96	0,45	1,22	1,59

Tudi pri času dekodiranja (tabela 3) naš algoritem zaostaja za drugimi. Kljub temu pa deluje realnočasovno, saj dekodira sekundo stereo avdia v povprečno 17 ms. Algoritem ima potencial pretočnega formata, saj že prototip podpira prepletanje avdiokanalov.

5 Zaključek

Obstoječe metode brezizgubnega stiskanja avdia so si med seboj dokaj podobne, saj vse poskušajo napovedati naslednji vzorec iz linearne kombinacije prejšnjih, le redke metode pa se osredotočajo na opisovanje prisotnih frekvenc v avdiu. Tudi algoritmi za kodiranje razlik od napovedi so dokaj uniformni, skoraj vedno se uporabi variacija Golombovega kodiranja. Predlagali smo drugačno alternativo, ki namesto posameznih vzorcev napoveduje daljše sekvence s pomočjo prileganja daljicam ali Bézierovim krivuljam. Metoda je še v zgodnji fazi razvoja, a že zdaj dosega dokaj primerljivo stopnjo stiskanja in hitrost dekodiranja z obstoječimi metodami, hitrost kodiranja pa je za zdaj občutno slabša od obstoječih metod. Rezerve so še pri izbiri vmesnih interpoliranih točk P_u in P_v, kakor tudi pri izbiri nabora možnih dolžin krivulj. V intervalih z višjimi frekvencami, takšnih z zgoščenimi lokalnimi ekstremi signala, zaenkrat uporabljamo daljice, saj so Bézierove krivulje s kratkim intervalom med P_0 in P_2 neekonomične. Morda bi lahko po zgledu FLAC tukaj uporabili še kakšno drugačno napoved, pa tudi morebitne ponovitve, statistično porazdelitev preostankov in njihovih dolžin bi bilo smiselno raziskati. Smiselno bi bilo namesto BASC poskusiti tudi z Golomb-Riceovimi kodami, ki jih uporabljajo tudi primerjane metode.

Zahvala

Raziskave je sofinancirala Javna agencija za raziskovalno dejavnost Republike Slovenije v sklopu raziskovalnega projekta CEUS N2-0181.

Literatura

[1] T. Robinson, „SHORTEN: Simple lossless and near-lossless waveform compression“. University of Cambridge, Department of Engineering, 1994.

[2] D. Salomon in G. Motta, *Handbook of Data Compression*, 5. izd. Springer, 2009.

[3] J. Coalson, *FLAC - Free Lossless Audio Codec*. https://xiph.org/flac/ [dostop 29. 6. 2022].

[4] B. Iser, W. Minker, in G. Schmidt, *Bandwidth extension of speech signals*. Springer, 2008.

[5] D. Byrant, *Wavpack*. https://www.wavpack.com/ [dostop 17. 6. 2022].

[6] T. Liebchen, „MPEG-4 ALS – The Standard for Lossless Audio Coding“, *Journal of the the Acoustical Society of Korea*, let. 28, št. 7, str. 618–629, 2009.

[7] N. Guid, *Računalniška grafika*. Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko, 2001.

[8] M. A. Lachance in A. J. Schwartz, „Four point parabolic interpolation“, *Computer Aided Geometric Design*, let. 8, št. 2, str. 143–149, 1991, doi: 10.1016/0167-8396(91)90039-E.

[9] A. Moffat in V. N. Anh, „Binary codes for locally homogeneous sequences“, *Information Processing Letters*, let. 99, št. 5, str. 175–180, 2006, doi: 10.1016/j.ipl.2006.04.014.

[10] Orange Open Movie Team, *Elephants dream*. https://orange.blender.org/ [dostop 17. 6. 2022].

[11] Lost European, *Images*. freemusicpublicdomain.com [dostop 17. 6. 2022].

[12] Angelwing, *The Nymphaeum Part I and II*. freemusicpublicdomain.com [dostop 17. 6. 2022].

[13] Angelwing, *What A Beautiful Sunset (Extended)*. freemusicpublicdomain.com [dostop 17. 6. 2022].