



Semestrální práce z KIV/AZS

Zvukové efekty

Dominik Zappe
(A23N0011P)

Únor 2024

Obsah

1	Zadání	1
2	Zvolené efekty	2
3	Implementace	5
4	Výsledky	7
5	Uživatelská příručka	11
6	Závěr	12

1 Zadání

1. Prostudujte algoritmy pro vytváření zvukových efektů (echo, reverb, chorus, flange)
2. Implementujte zvolené algoritmy a demonstруйте jejich vlastnosti.

2 Zvolené efekty

Celkem bylo zvoleno a implementováno 10 zvukových efektů:

- Ping Pong Delay
- Wah-Wah
- Flanger
- Phaser
- Overdrive
- Distortion
- Reverb
- Bit Crusher
- 8D Audio
- Vocal Doubler

Většina efektů, až na efekt *Vocal Doubler*, se spíše hodí pro aplikaci na hudební nástroje. Je možné je testovat na lidském hlase, ale hudební nástroje jsou vhodnější.

Ping Pong Delay

Ping pong delay je zvukový efekt využívající zpoždění (*Delay*), ale vytvářená ozvěna zvuku se opakuje s určitým zpožděním mezi levým a pravým stereo kanálem, čímž vzniká efekt pohybu mezi reproduktory, resp. sluchátky [2].

Wah-Wah

Zvukový efekt Wah-Wah se často používá v kytarové hudbě. Tento efekt se původně objevil ve 20. letech 20. století jako součást elektrických varhan, ale začal být populární v kytarové hudbě ve 60. letech. Wah-Wah mění charakter zvuku tak, aby vytvořil zvuk podobný tomu, který vydává hlasitý hlas člověka, když vydává „wah“ zvuk. Požadovaného efektu se dosahuje filtrací a změnou frekvenční odezvy [1].

Flanger

Tento efekt se opět často používá v hudbě. Podobně jako chorus a phaser patří flanger mezi tzv. *modulační* efekty, které mění zvuk tím, že vytváří kopii původního signálu a následně ji mírně posouvají v čase. Tento posun se mění dynamicky, což vytváří charakteristický zvukový efekt [1].

Phaser

Podobně jako flanger, phaser pracuje s kopií původního signálu, ale místo toho, aby byla kopie posunuta lineárně v čase, jsou různé frekvence signálu posunuty fázově vzhledem k sobě [1]. Tento efekt je opět často užíván v hudbě.

Overdrive

Efekt overdrive je zvukový efekt používaný v kytarové hudbě, který zvyšuje hlasitost a zkresluje zvuk elektrické kytary, aby vytvořil teplý, syrový a sytý zvuk [1]. Overdrive zesiluje harmonické složky signálu, což vytváří příjemný a plný zvuk s "hřejivou" kvalitou, která je oblíbená mezi kytaristy v mnoha žánrech hudby, jako je rock, blues, a metal [2].

Distortion

Distortion je další zvukový efekt pro elektrické kytary, který se používá k vytvoření zkresleného a agresivního zvuku. Podobně jako overdrive, distortion zesiluje signál kytary nad hranici lineárního zesílení, ale v tomto případě je zkreslení mnohem intenzivnější a výraznější [1, 2]. Efekt distortion mění tvar vlny signálu tak, že ho deformuje a vytváří tak charakteristickou "zakroucenou" a "přerostlou" zvukovou texturu.

Reverb

Reverb, zkráceně od anglického "reverberation", je zvukový efekt, který simuluje odrazy zvuku od různých povrchů a prostor ve skutečném životě. Tento efekt přidává do zvuku pocit prostoru, jako by byl produkován v reálném akustickém prostředí, jako je například katedrála, hala nebo malá místnost [2].

Bit Crusher

Bit crusher je zvukový efekt, který uměle snižuje rozlišení (bitovou hloubku) zvukového signálu, což vytváří charakteristický "rozbitý" zvuk. Tento efekt je často používán v

elektronické hudbě, zejména v žánrech jako je techno.

8D Audio

8D zvuk, někdy označovaný jako „binaurální zvuk“, je zvukový efekt, který vytváří iluzi prostorového zvuku, který se pohybuje kolem posluchače. Tento efekt se často dosahuje prostřednictvím speciálního zpracování zvuku, které využívá posun zvuku mezi levým a pravým kanálem, různé úrovně hlasitosti a filtraci, aby vytvořil dojem, že zvuk putuje kolem posluchače, často ve smyčce kolem hlavy.

Vocal Doubler

Vocal Doubler je efekt, který se používá k rozšíření a zesílení hlasového vokálu v mixu. Jeho hlavním účelem je vytvořit dojem, že zpěvák zpívá vícekrát nebo že zpěvák doprovází další vokalist. Oproti zvukovému efektu „Chorus“ se *Vocal Doubler* spíše zaměřuje na „dvojitost“ hlasu, než na prostorovost [2].

3 Implementace

Na veškerou implementaci byl využit jazyk Python ve verzi 3.9. Pro všechny zvukové efekty byly využity především knihovny *NumPy* a *SciPy*. Implementace je obohacena o grafické uživatelské rozhraní vytvořené pomocí knihoven *imgui* a *wxWidgets* s pomocí API *OpenGL*.

Převážná většina zvuků, na kterých byla implementace testována, byly staženy z webu <https://pixabay.com/sound-effects>.

Ping Pong Delay

Pro každý vzorek signálu je vypočtena hodnota zpožděného signálu pro oba kanály (levý a pravý). Ke každému vzorku levého kanálu je přičten zpožděný vzorek z pravého. Podobně je to provedeno i pro pravý kanál. Na závěr dochází k míchání vstupního signálu se zpožděným signálem.

Wah-Wah

Nejprve se inicializuje výstupní signál a filtry. Poté je vytvořeno pole „centerfrekvencí“ pro efekt Wah-Wah. Následně se aplikuje efekt na vstupní signál pomocí filtrů a modulační funkce sinusového tvaru.

Flanger

Na začátku se vytvoří nízkofrekvenční oscilátor, který se používá k modulaci zpoždění signálu. Dále se aplikuje efekt Flanger na vstupní signál pomocí zmiňovaného oscilátoru.

Phaser

Nejprve se vypočítá maximální počet vzorků zpoždění na základě maximálního zpoždění a vzorkovací frekvence. Následuje aplikace efektu na vstupní signál pomocí výpočtu zpoždění a modulace amplitudy signálu na základě zvolené frekvence.

Overdrive

Signál je aplikován pomocí několika podmínek, které zesilují signál podle definovaného prahu a provedou složitější výpočet pro určení výstupní hodnoty signálu.

Distortion

Tento efekt je ještě jednodušší než Overdrive, nejprve se vezme znaménko vstupního signálu, poté se vypočte koeficient zkreslení ze zadaného parametru *gain*. Následně se aplikuje efekt zkreslení na vstupní signál pomocí exponenciální funkce.

Reverb

Efekt je implementován pomocí konvoluce vstupního signálu s danou impulzní odezvou. Výsledný signál je mix původního vstupu s příslušným poměrem výsledku konvoluce. Impulzní odezvy byly staženy z internetu – v grafickém uživatelském rozhraní je na výběr mezi čtyřmi různými impulzními odezvami. Za zmínku nejvíce stojí impulzní odezva kostela – <https://afewthingz.com/impulseresponse>. Zbylé impulzní odezvy byly získány z webu <https://pixabay.com/sound-effects>.

Bit Crusher

Implementace tohoto efektu je prakticky jednoduchá kvantizace – hodnoty vzorků vstupního signálu jsou zaokrouhleny na nejbližší celé číslo a následně vyděleny maximální hodnotou.

8D Audio

V tomto efektu se vstupní signál se konvertuje na mono. Poté vypočítá panoramatické zesílení pro přední, zadní, levý a pravý kanál. Vstupní signál je následně vynásoben těmito zesíleními a rozdělen na čtyři kanály (směry). Nakonec jsou tyto kanály sečteny, aby se získal výstupní signál (stereo).

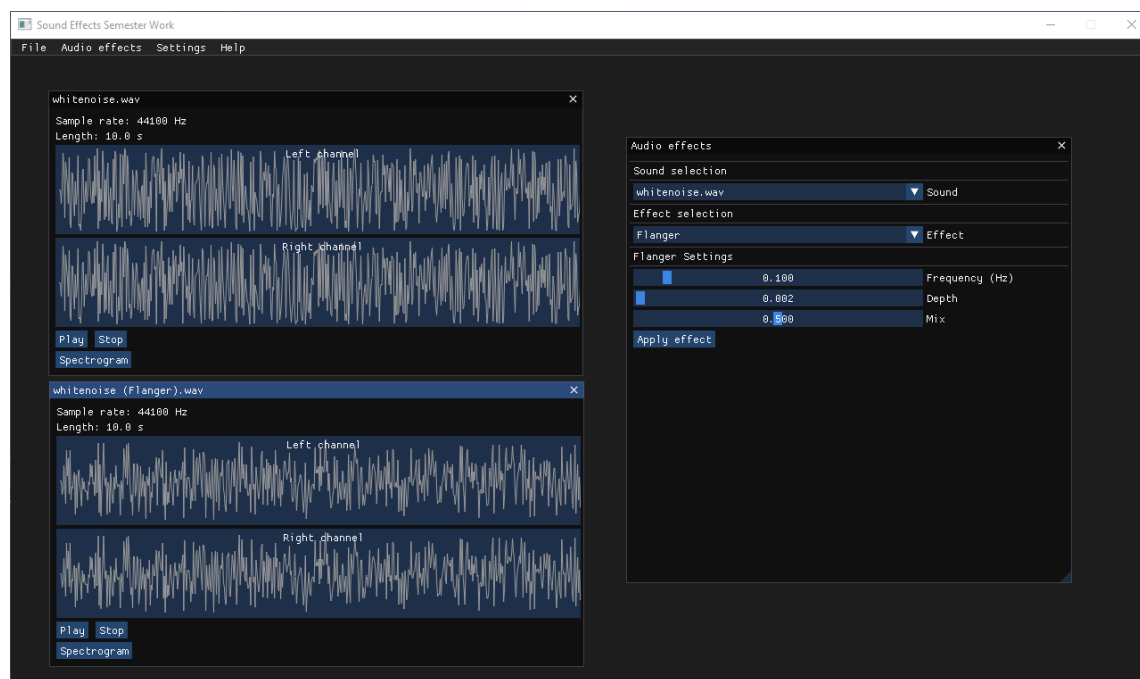
Vocal Doubler

Nejprve se vytvoří kopie vstupního signálu a spočítá se počet vzorků zpoždění. Poté se vytvoří duplikát vstupního signálu s posunutím v čase a s detuningem pomocí interpolace. Detuning je prakticky jednoduchý *Pitch Shift* – změna výšky tónu. Nakonec se oba signály (původní a detunovaný) smíchají dohromady v poměru 1:1.

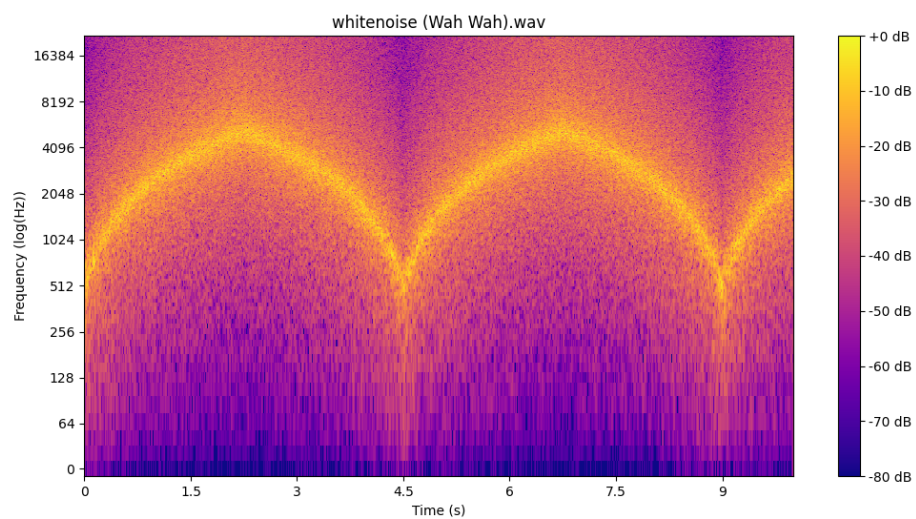
4 Výsledky

Pro ověření výsledků je vhodný poslech upravovaných zvuků, to však v dokumentaci lze stěží – grafické rozhraní vykresluje závislost amplitudy na čase každého načteného zvuku, je navíc možné si nechat vykreslit i spektrogram (závislost frekvence na čase).

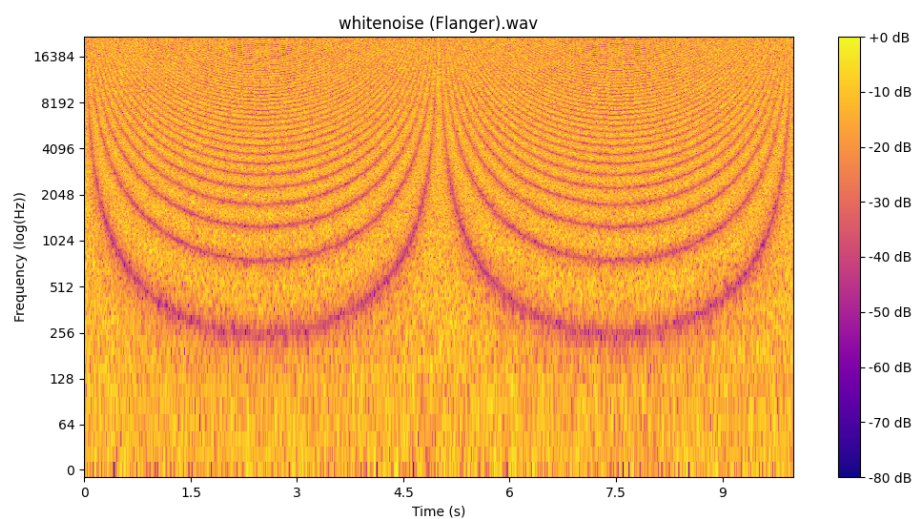
Na následujících obrázcích je demonstrace výsledků aplikace. Na Obrázku 1 je vidět výsledné grafické rozhraní aplikace. Na Obrázcích 2, 3 a 4 jsou spektrogramy efektů Wah-Wah, Flanger a Phaser – ve spektrogramech jsou vidět krásné rozdíly jednotlivých efektů (Flanger a Phaser jsou si obecně podobné efekty a vhodným nastavením parametrů jednoho z efektů lze jednoduše docílit podobného zvuku efektu druhého). Na obrázcích 5 a 6 jsou vidět výsledky efektů na grafech amplitud v závislostech na čase – je vidět, že Distortion je silnější efekt než Overdrive; zároveň je také vidět výsledek Bitcrusher efektu, tj. výsledný signál je hranatý.



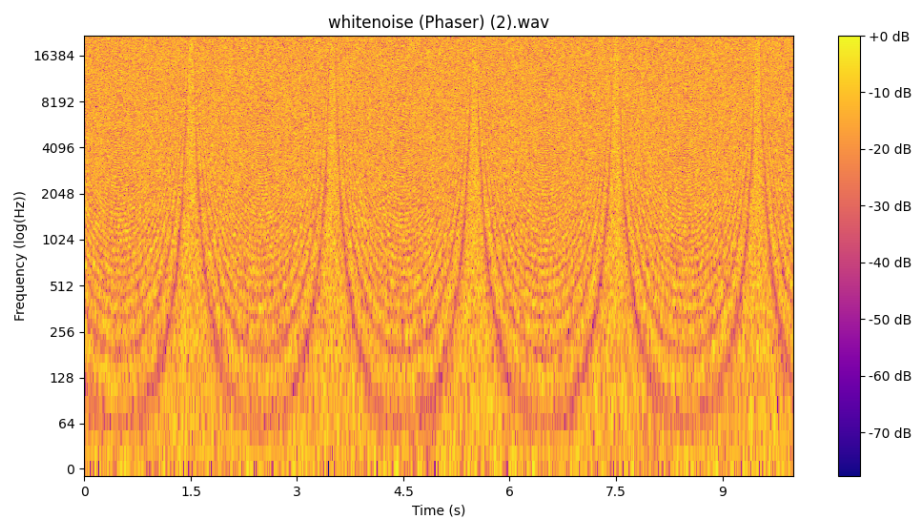
Obrázek 1: Výsledné grafické rozhraní aplikace



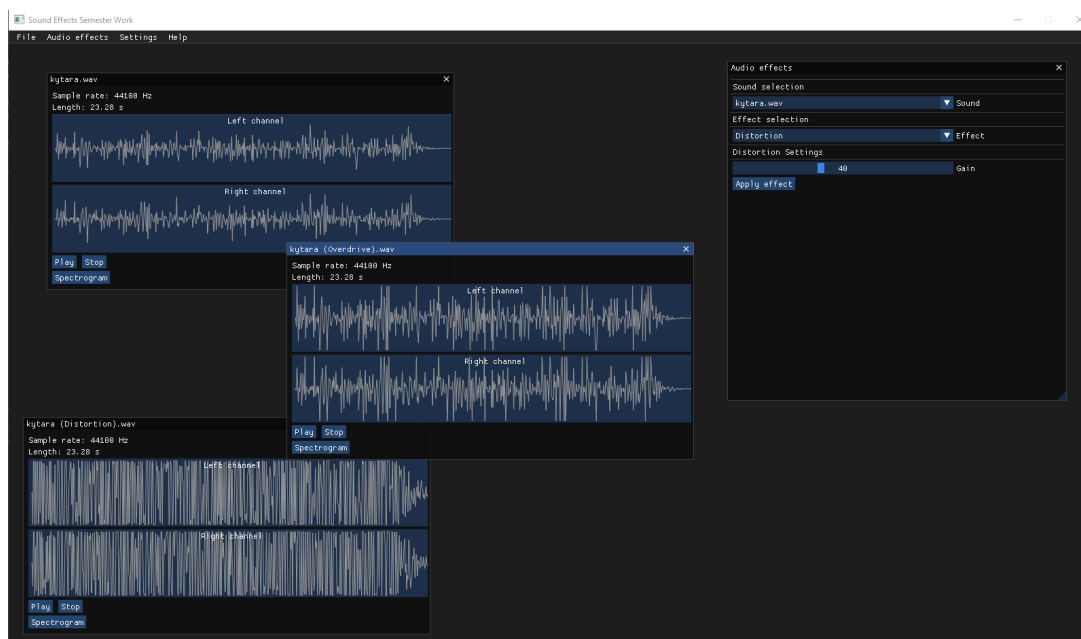
Obrázek 2: Spektrogram Wah-Wah efektu na bílém šumu



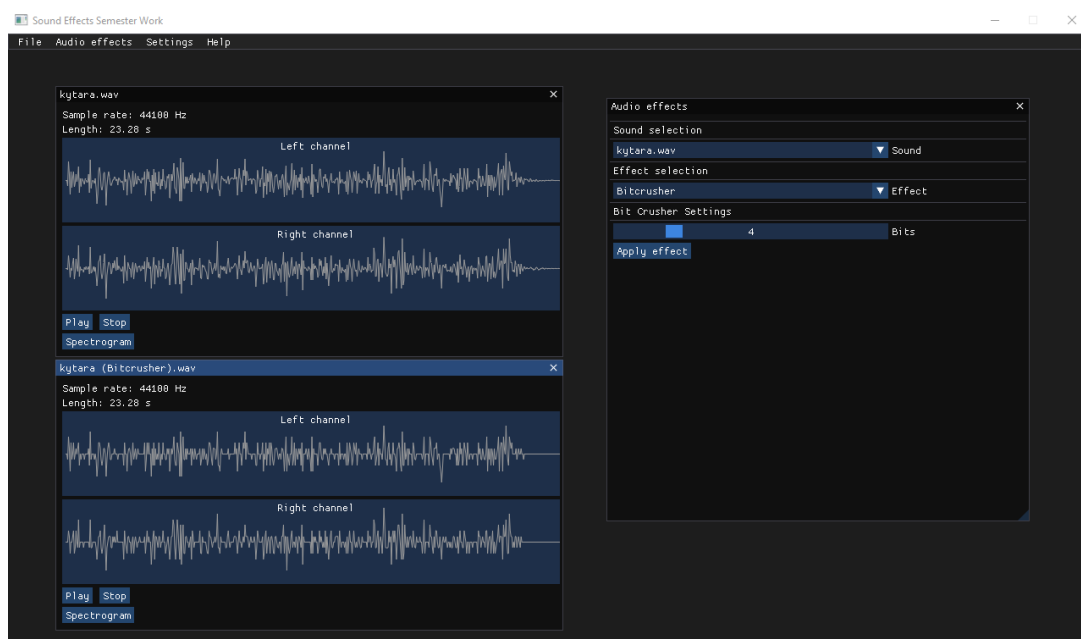
Obrázek 3: Spektrogram Flanger efektu na bílém šumu



Obrázek 4: Spektrogram Phaser efektu na bílém šumu



Obrázek 5: Overdrive a Distortion efekt aplikovaný na kytara.wav



Obrázek 6: Bitcrusher efekt aplikovaný na kytara.wav

5 Uživatelská příručka

Nejprve je nutné vyřešit veškeré závislosti aplikace, z kořenového adresáře projektu je možné pro tento účel použít příkaz:

```
pip install -r requirements.txt
```

Poté je možné samotnou aplikaci spustit z adresáře `/src/` pomocí příkazu:

```
python main.py
```

Po spuštění aplikace se objeví hlavní okno aplikace viz Obrázek 1. Hlavní menu nabízí možnosti File, Audio effects, Settings a Help. Pod nabídkou File se schovávají možnosti jako vyresetování pracovního prostředí, načtení zvuku, uložení zvuku a ukončení aplikace. Pod nabídkou Audio effects je pouze jedna jediná možnost a to jest otevření podokna se zvukovými efekty (toto okno je otevřeno automaticky při startu aplikace, ale pokud by si ho uživatel z nějakého důvodu zavřel, je možné si ho znovu otevřít takto). Settings obsahuje položku Window settings, tedy pouze nastavení grafiky okna (například velikost okna nebo barevné schéma). Pod nabídkou Help je pouze About, který neříká nic zajímavého – pouze jméno autora.

Každé načtení nového zvuku, či vygenerování zvuku pomocí aplikace efektu, otvírá nové podokno obsahující informace o daném zvuku (vzorkovací frekvence, délka zvuku, graf levého a pravého kanálu (závislost amplitudy na čase) a možnost přehrání, zastavení přehrávání a možnost vykreslení spektrogramu).

Podokno zvukových efektů je poměrně přímočaré, v horní části je rozbalovací seznam pro výběr zvuku, následuje rozbalovací seznam pro výběr kýženého zvukového efektu. Po výběru specifického efektu se zbytek okna modulárně mění na základě daného efektu – každý zvukový efekt má jiné parametry. Úplně na konci v tomto podokně je tlačítko Apply effect, jak již název napovídá, tímto tlačítkem se daný efekt s danými parametry aplikuje na daný zvuk.

6 Závěr

V rámci této semestrální práce byla řešena analýza a implementace různých zvukových efektů pomocí digitálního zpracování signálů. Cílem bylo porozumět principům a algoritmům jednotlivých zvukových efektů a realizovat je v programovacím jazyce Python. Každý efekt byl implementován jako samostatná funkce s přehledným a komentovaným kódem.

V budoucnu by bylo možné rozšířit tuto práci o další zvukové efekty, rozšířit již existující efekty o více parametrů nebo optimalizovat stávající implementaci.

Bibliografie

- [1] Jean-Michel Reveillac. *Musical sound effects: Analog and digital sound processing*. ISTE, 2018.
- [2] Eric Tarr. *Hack Audio: An introduction to computer programming and Digital Signal Processing in MATLAB*. Routledge, 2019.