

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ANALÝZA AUDIO SIGNÁLU V REÁLNÉM ČASE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MARTIN ŘEZÁČ

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ANALÝZA AUDIO SIGNÁLU V REÁLNÉM ČASE

REAL-TIME ANALYSIS OF AUDIO SIGNALS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MARTIN ŘEZÁČ

VEDOUCÍ PRÁCE
SUPERVISOR

Doc. Dr. Ing. Jan Černocký

BRNO 2012

Abstrakt

Cílem této práce je vytvořit program, který bude schopen na základě příchozích audio vzorků rozpoznávat v reálném čase základní frekvenci tónu, na základě určené frekvence vytvářet MIDI informace a ty posílat na zvolené MIDI rozhraní. V úvodní části je čtenář uveden do problematiky detekce základních frekvencí tónu. Následuje popis jednotlivých metod pro získávání základní frekvence s důrazem na metodu založenou na spektrální analýze zvuku a popis použitých technologií. Součástí je také popis implementace a vyhodnocení provedených testů včetně uvedení názorů některých hudebníků na použití této aplikace. Závěrem je celá práce zhodnocena. Součástí závěru je i zamýšlení nad dalším vývojem této aplikace.

Abstract

The goal of this thesis is creation an application, which can perform real-time fundamental frequency tracking of incoming audio samples. According to detected frequencies, the program generates MIDI messages, which are sent to chosen MIDI device. First, the reader is introduced to the issue of fundamental frequency tracking. The following part describes individual methods, especially the one based on spectral analysis of a tone. A description of used technologies is also a included in this part of this thesis. In the following part, the implementation and testing of application are described including opinions of several musicians about this product. At the end, the whole work is concluded and the possible further development is outlined.

Klíčová slova

Analýza, audio, signál, vzorky, vzorkování, MIDI, VST, VSTi, Steinberg, plugin, rozhraní, virtuální MIDI, informace, DSP, syntetizér, syntéza, reálný čas, FFT, Fourierova transformace, frekvence, základní frekvence, sledování, nota, tón, výška tónu

Keywords

Analysis, audio, signal, samples, sampling, MIDI, VST, VSTi, Steinberg, plugin, interface, virtual MIDI, information, DSP, synthesiser, synthesis, real-time, FFT, Fourier transform, frequency, fundamental frequency, tracking, note, tone, pitch

Citace

Řezáč Martin: Analýza audio signálu v reálném čase, bakalářská práce, Brno, FIT VUT v Brně, 2012

Analýza audio signálu v reálném čase

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Doc. Dr. Ing. Jana Černockého a konzultanta z Výzkumného centra JAMU, pana Ing. Lubora Přikryla. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Martin Řezáč
9.5.2012

Poděkování

Chtěl bych poděkovat především vedoucímu Doc. Janu Černockému za pomoc při tvorbě algoritmu pro rozpoznávání a připomínky, které přispěly ke zkvalitnění této práce. Dále bych chtěl poděkovat panu Ing. Luboru Přikrylovi za pomoc při návrhu výsledné aplikace. Velký dík patří také programátorovi z výzkumného centra JAMU – panu Ing. Vladimírovi Tichému za pomoc při programování VST rozhraní aplikace.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah	1
1 Úvod	3
2 Aplikace na trhu	4
2.1 Digital Ear	4
2.2 IntelliScore	5
2.3 Solo Explorer	5
2.4 WiDi Recognition System	6
3 Základní frekvence tónu	7
3.1 Vztahy mezi jednotlivými tóny	7
3.2 Detekce založená na spektrální analýze	8
3.2.1 Počet vzorků v signálovém rámci	8
3.2.2 Získávání základní frekvence	9
3.2.3 Rozpoznávání not se stejnou základní frekvencí	10
3.3 Ostatní metody pro rozpoznávání	10
4 Použité technologie	11
4.1 MIDI	11
4.1.1 MIDI informace	11
4.1.2 Struktura kanálové MIDI informace	11
4.1.3 Využití v tomto projektu	12
4.2 Steinberg VST	12
4.2.1 Verze VST	12
4.2.2 VST & VSTi	12
4.2.3 Využití v projektu	13
4.3 Implementační jazyk a prostředí	13
5 Návrh aplikace	14
5.1 Koncept návrhu	14
5.1.1 Samostatná aplikace s podporou VSTi	14
5.1.2 Samostatná aplikace s MIDI výstupem	15
5.1.3 VST plugin s MIDI rozhraním	15
5.2 Grafické uživatelské rozhraní	16
6 Implementace	18
6.1 Struktura aplikace	18
6.1.1 Modul s VST rozhraním	19
6.1.2 Modul pro posílání MIDI dat	19

6.1.3	Modul pro frekvenční analýzu	19
6.2	Uživatelské rozhraní aplikace	20
7	Testování	21
7.1	Návrh testů	21
7.1.1	Typy chyb při analýze zvuku	21
7.1.2	Automatizované testování	22
7.1.3	Testování s použitím reálných nástrojů	22
7.1.4	Uživatelské testování	22
7.2	Vyhodnocení testů	22
7.2.1	Vyhodnocení automatizovaných testů	23
7.2.2	Vyhodnocení testů s reálnými nástroji	26
7.2.3	Uživatelské testy	28
8	Závěr	29
	Příloha A – Příklad testovacích souborů	32
	Příloha B – Uživatelské hodnocení aplikace	33
	Příloha C – Návod k použití aplikace	35
	Příloha D – Obsah příloženého CD	36

1 Úvod

Zpracování zvuku je v dnešní době velmi rychle se rozvíjející odvětví hudebního i IT průmyslu. Nejenže je zpracování zvuku prakticky nezbytné pro televizní, rádiové přenosy, živá vystoupení apod., ale především ze strany hudebníků a skladatelů neustále rostou požadavky na kvalitu zpracování zvukového signálu a na věrohodnost syntetizovaných zvuků.

V této práci se budu zabývat jedním z méně rozšířených, avšak přesto velmi perspektivních oborů IT průmyslu, a to oblastí převodu reálného zvukového signálu na MIDI¹ informaci a komunikací programů přes rozhraní VST².

VST bylo vyvinuto firmou Steinberg pro komunikaci hudebních aplikací se zásuvnými moduly - tzv. pluginy. V dnešní době je rozhraní VST natolik rozšířené, že prakticky každá běžně používaná aplikace, která s MIDI informacemi pracuje, podporuje výměnu dat pomocí tohoto rozhraní a absence VST by měla při prodeji aplikace z finančního hlediska znatelné důsledky. Dalo by se tedy hovořit o jakémsi unifikovaném protokolu pro výměnu MIDI/audio dat mezi programy, nebo mezi moduly programů.

Na úvod této práce bude vytvořen krátký přehled některých aplikací, které jsou dostupné pro převod zvukových souborů na MIDI. Součástí této kapitoly bude i základní srovnání funkčnosti a ovládání zvolených aplikací.

Druhá část bude věnována detekci jednotlivých tónů ve zvukovém signálu. Bude zde rozebrán zvolený způsob detekce hlavních frekvencí tónu. Součástí kapitoly bude také krátká zmínka o ostatních metodách pro detekci hlavní frekvence a jejich stručný popis, výhody a nevýhody.

Hlavním tématem této práce bude tvorba funkční aplikace, která bude schopna pomocí vybraného postupu v reálném čase analyzovat vstup zvukového zařízení počítače, určit jednotlivé tóny, jejich výšku a délku a průběžně posílat naměřená data přes uživatelem zvolený MIDI výstup do externích hudebních syntetizérů, nebo jiných aplikací. Součástí bude testování výsledné aplikace jak na syntetizovaných vzorcích audio signálu, tak na reálných nástrojích.

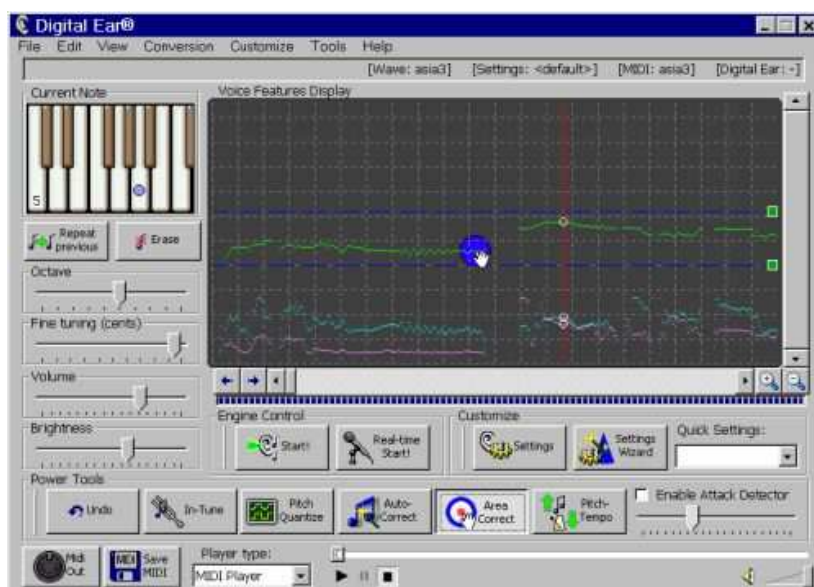
¹ Musical Instrument Digital Interface – rozhraní pro výměnu hudebních dat mezi hudebními nástroji. Zároveň standard definující formát těchto dat.

² Virtual Studio Technology – jedná se o speciální rozhraní pro komunikaci aplikace a přídatných modulů pro zpracování zvuku.

2 Aplikace na trhu

Tato kapitola byla částečně inspirována zdrojem [5]. Součástí této kapitoly je testování a hodnocení jednotlivých dostupných produktů pro frekvenční analýzu zvuku. Všechny uvedené produkty pracují nad již vytvořenými hudebními soubory, přítomen je často i modul pro zpracování audia v reálném čase (dále jen RT). Bohužel, latence při RT analýze byla často tak velká, že dělala program pro použití při živém hraní prakticky nepoužitelným. Pro práci v reálném čase se podařilo nalézt pouze jeden použitelný produkt od firmy Roland. Jedná se však o specializovaný hardwarový analyzátor, tedy jde o trochu jinou kategorii produktů. Absence softwarových produktů pro „on-line“ analýzu zvuku by se dala považovat za určitou výhodu této práce oproti ostatním komerčním nebo volně dostupným produktům.

2.1 Digital Ear



Obrázek 2.1 – software Digital Ear, převzato z [12]

Digital Ear je produkt firmy Epinoisis Software³. Cena tohoto produktu činí 129.95 USD. K dispozici je zdarma i demoverze, která je omezena celkovou délkou analyzovatelného souboru. Pro testování byla použita právě tato demoverze. Tento produkt se specializuje na monofonní zvuk. Výrobce uvádí i přítomnost modulu pro analýzu zvuku v reálném čase, při testování se však tuto funkci nepodařilo provozovat.[12]

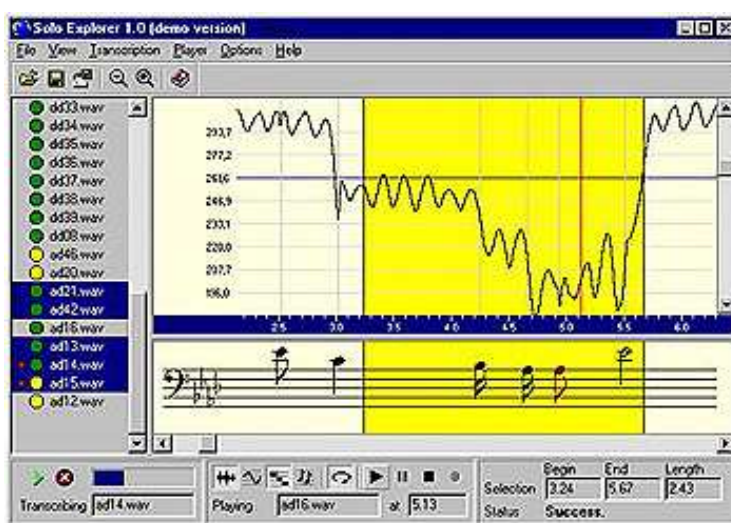
Program má poměrně složité až matoucí ovládání. Rozmístění ovládacích prvků je rovněž neintuitivní. Co se kvality týče, při vývoji tohoto produktu zůstala kvalita a přesnost analýzy až na druhém místě, což by se u komerčních produktů stávat nemělo. Jako pozitivum lze uvést fakt, že program dosahoval podobné přesnosti u analýzy zvuku hudebních nástrojů i hlasu, což je poměrně neobvyklé.

³ Produkt dostupný z URL: <http://www.digital-ear.com/digital-ear/buy.asp?tab=7>

2.2 IntelliScore

„Nástroj IntelliScore⁴ podporuje vstup více formátů (wav, mp3, wma), dále pak i realtime zpracování.“ [5] Modul pro RT zpracování ovšem neobsahuje podporu pro hostování ASIO ovladačů (Jedná se o ovladače s velmi nízkou latencí). Program dokáže analyzovat jak monofonní, tak polyfonní vstup. U monofonního vstupu je přesnost, dle očekávání, daleko vyšší. Před samotným převodem je možné nastavit několik parametrů, mezi nimi i nástroj, který byl použit pro nahrávání zvuku. Pro polyfonii obsahuje program i několik předdefinovaných hudebních těles. Vhodně zvolenými parametry se dá podstatně zvýšit přesnost detekce správných frekvencí. Program také disponuje modulem pro notový zápis, což mohou ocenit především lidé znalí hudební teorie.

2.3 Solo Explorer



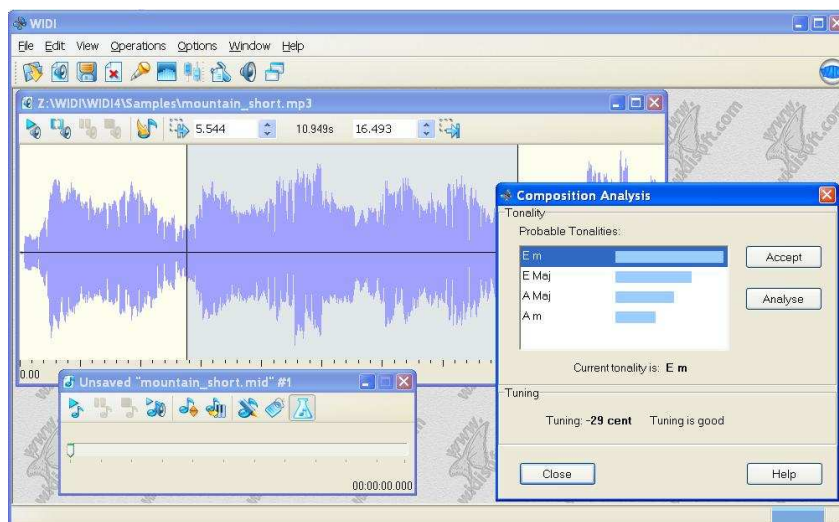
Obrázek 2.2 – program Solo Explorer od společnosti Recognisoft, převzato z [14]

Jedná se o software, který je primárně určen pro zpracování monofonního zvuku. Opět jde o komerční produkt, k dispozici je volně stažitelná demoverze⁵. Omezením u této demoverze je doba používání a délka vstupního audia. Uživatelské rozhraní tohoto programu je velmi intuitivní a dobře se tak s programem pracuje. Nevýhodou jsou poměrně chudé možnosti v oblasti nastavení rozpoznávání. Program také dokáže pracovat v RT režimu, avšak zde je pojetí RT zpracování poněkud nešťastné. Program ve zmíněném režimu pouze nahrává zvuk do souboru, analýzu poté provádí nad celým souborem, tedy až po dokončení nahrávání.

⁴ Produkt dostupný z URL: <http://www.intelliscore.net/product.html>

⁵ Produkt dostupný z URL: <http://www.recognisoft.com/>

2.4 WiDi Recognition System



Obrázek 2.3 – program Recognition system od firmy WiDiSoft, převzato z [11]

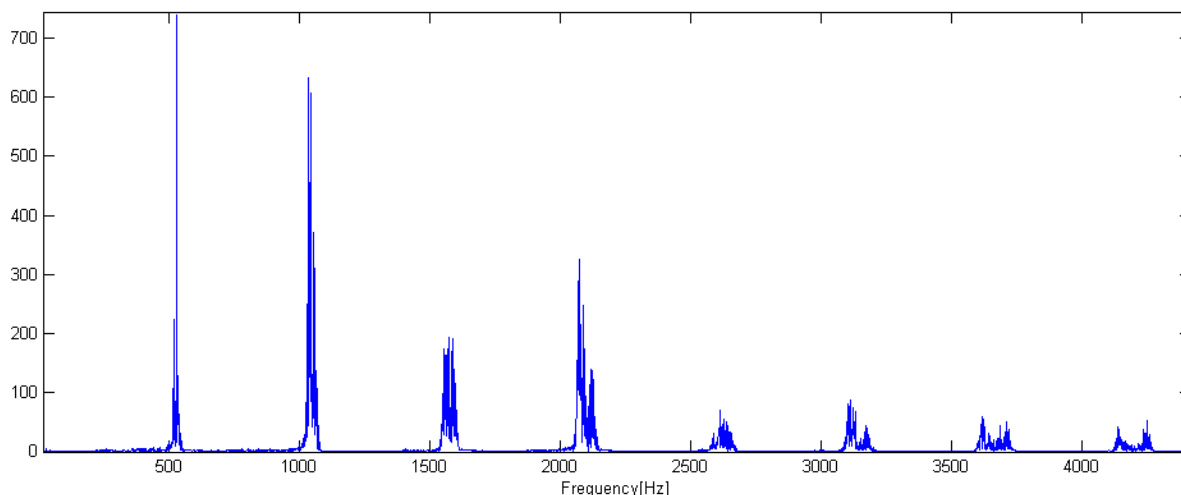
Tento program byl vyvinut ruskou firmou WiDiSoft. Opět se jedná o komerční produkt, jeho cena je 199 USD. K dispozici je zdarma trial verze⁶. Tato verze je mimo ukládání souborů plně funkční, její funkčnost je však časově omezena. Pro testování byla použita právě tato trial verze. Ze všech testovaných produktů dosahoval právě tento software nejlepších výsledků. Jako jediný obstojně zvládal analýzu polyfonního zvuku. I zde má uživatel možnost nastavit poměrně mnoho parametrů. Za zmínku stojí určitě volba tóniny a možnost výběru z několika rozpoznávacích algoritmů pro monofonní nebo polyfonní vstup.

„Rozpoznaný part lze editovat pomocí intuitivního editoru, který umožňuje například úpravu vlastností jednotlivých tónů, vkládání a mazání jednotlivých tónů nebo ruční rozdělení taktů za účelem zvýšení čitelnosti notového zápisu.“ [5] Program rovněž disponuje RT režimem. Latence je v tomto případě daleko nižší než u předchozích produktů, stále by se ale o použitelnosti tohoto programu pro živé hraní dalo diskutovat.

⁶ Produkt dostupný z URL: <http://www.widisoft.com/english/products.html>

3 Základní frekvence tónu

Každý tón je určen svou výškou, délkou, hlasitostí a barvou. Výška tónu je určena právě základní frekvencí. Pokud se tón pomocí příslušných metod rozloží do frekvenčního spektra (viz obrázek 3.1),



Obrázek 3.1 – Spektrum tónu C5 hraného houslemi

je patrné, že mimo hlavní frekvenci jsou zde slyšitelné i další frekvence. Tyto frekvence určují zmíněnou barvu tónu. Cílem této práce je správně detekovat základní frekvenci tónu, tedy separovat výšku tónu a frekvence, které určují barvu tónu na dvě oddělené složky.

3.1 Vztahy mezi jednotlivými tóny

Základní tón, tzv. *komorní A* má frekvenci 440 Hz. V MIDI informacích je tón reprezentován jako *A4*. Frekvence ostatních tónů se dá spočítat pomocí rovnice 3.1.

$$f_n = 440 \times 2^{i_n/12} \quad (3.1)$$

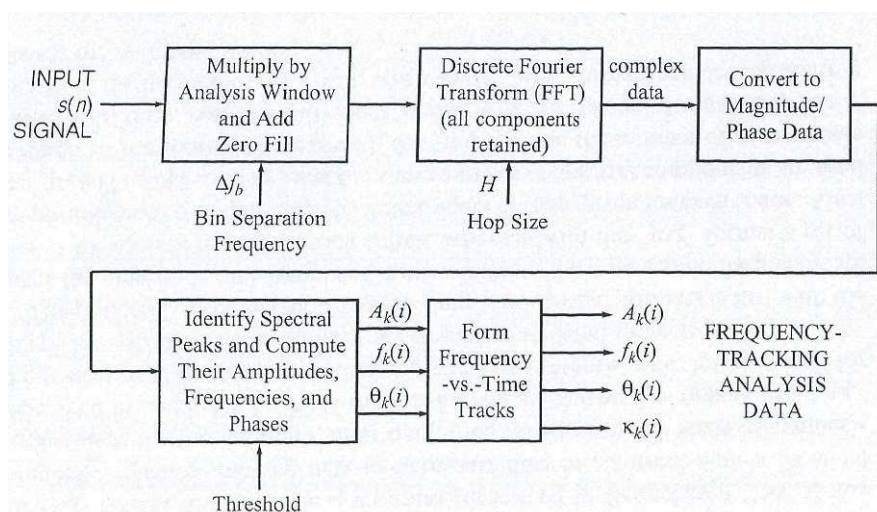
kde i_n je půltónová vzdálenost zvoleného tónu od *komorního A*. Pokud je tón položen níže, resp. má nižší frekvenci, tato vzdálenost je záporné celé číslo. V opačném případě se jedná o kladné číslo.

Pro účel rozpoznávání potřebujeme především zjistit tzv. *čtvrttóny*. Čtvrttón je jakousi pomyslnou hranicí, která dělí interval mezi dvěma tóny na dvě části. Při určování čtvrttónů musí být ovšem respektován základní vztah mezi dvěma sousedními tóny.

$$f_{quarter} = f_{low} \times 2^{0.5/12} \quad (3.2)$$

Rovnice 3.2 popisuje vztah mezi tónem, resp. půltónem, jehož frekvenci zastupuje proměnná f_{low} , a jeho vyšším *čtvrttónem*, tedy *čtvrttónem*, jenž rozděluje hranici mezi zmíněným tónem a vyšším sousedním půltónem.

3.2 Detekce založená na spektrální analýze



Obrázek 3.2 – schéma detekce základní frekvence založené na spektrální analýze, převzato z [1]

Tato metoda je založená na hledání maximální hodnoty koeficientů Fourierovy transformace (dále jen FT) daného signálu. V prvním kroku je tedy třeba signál transformovat pomocí FT. Vzhledem ke skutečnosti, že se jedná o digitální zpracování zvuku, tedy pracujeme s diskretním signálem, musí být použit odpovídající typ FT, konkrétně Diskrétní FT (dále jen DFT). Ta je popsána vzorcem:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}nk} \quad k = 0, \dots, N-1 \quad (3.3)$$

Za pozornost stojí fakt, že tato metoda je velmi náročná na výkon počítače, její složitost je $O(N^2)$. Pokud by tato metoda byla použita, znamenalo by to další zvětšení latence. Pro naše potřeby bude tedy použita alternativa pod zkratkou *FFT* (Fast Fourier transform – rychlá Fourierova transformace). Složitost této metody je $O(N \log N)$, pro naše potřeby tedy daleko přijatelnější. Algoritmů pro výpočet FFT existuje celá řada, popis těchto algoritmů je nad rámec této práce.

3.2.1 Počet vzorků v signálovém rámci

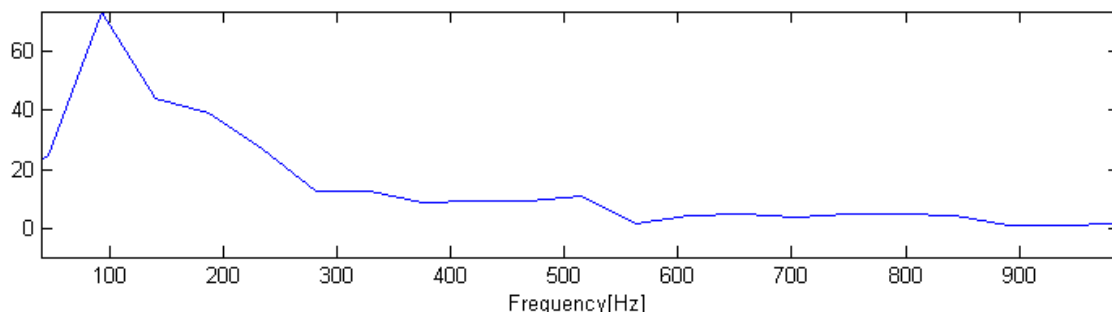
Vstupní signál je reprezentován reálnými čísly, je nutné provést FFT z reálných na komplexní čísla. V takovém případě jsou použitelné pouze koeficienty od 0 do $N/2$, kde N je počet vzorků signálu ve vstupním rámci, resp. velikost použitého rámce vstupního signálu⁷. Výsledný počet koeficientů je tedy oproti počtu vstupních vzorků přibližně poloviční. Zde nastává problém v přesnosti přepočtu indexu koeficientu transformovaného úseku na frekvenci. Tato frekvence se počítá podle vzorce 3.4.

$$f_z = \frac{C_i * F_s}{N} \quad (3.4)$$

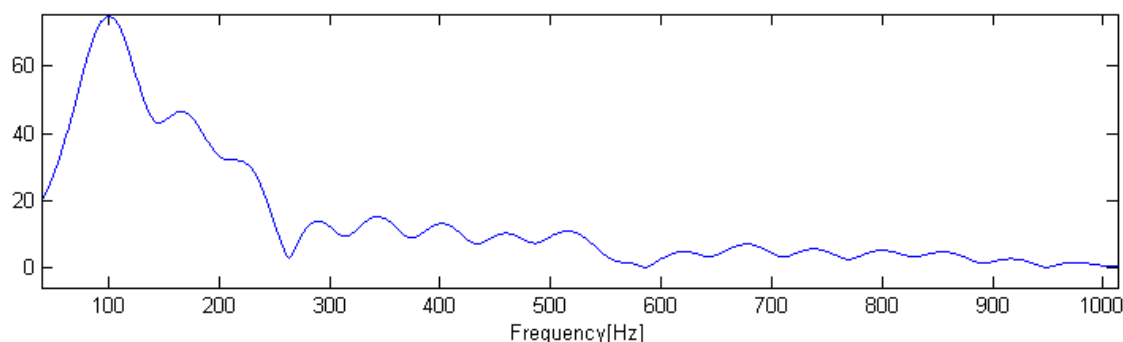
C_i zde reprezentuje index koeficientu Fourierovy transformace, N vstupní počet hodnot FFT, F_s je vzorkovací frekvence příslušného zvukového toku. Je patrné, že čím více vzorků se na vstupu FFT objeví, tím více koeficientů je možno získat. Přímá úměrnost platí také ve vztahu počtu koeficientů FFT a frekvenčního rozlišení.

⁷ Výstupem FFT z použitých knihoven v aplikaci je pouze $N/2 + 1$ koeficientů, zbytek je zanedbáván.

Malé frekvenční rozlišení se potom projevuje neschopností detekovat správně některé pultóny. V nižších frekvencích, kdy rozdíly frekvencí jsou poměrně malé (přibližně 2-5 Hz) dochází často k poměrně závažnému jevu, kdy se do intervalu frekvencí s použitím sousedních koeficientů vejde i více tónů. To se uživateli projevuje způsobem, že při hraní určitých hlubokých tónů se uživateli ozývají tóny sousední, tedy analyzátor je schopen rozpoznat každý třetí tón.



Obrázek 3.3 – výsledek FFT nad 1024 vzorky signálu



Obrázek 4.3 – výsledek FFT nad 65536 vzorky signálu

Na předchozích dvou obrázcích jsou zobrazeny části grafu výsledků dvou FFT, z nichž první obrázek obsahuje graf výsledku FFT nad rámcem o velikosti 1024 vzorků, druhý 65536 vzorků. U druhého grafu je vidno, že obsahuje více bodů (na křivce nejsou znatelné zlomy), než první. Toto jen potvrzuje fakt, že při použití více vzorků je detekce, resp. přepočítání na frekvenci daleko přesnější.

Je tedy vhodné zajistit pro Fourierovu transformaci více hodnot. Aby frekvenční rozlišení bylo dostačující, je dobré použít přibližně stejný nebo vyšší počet vzorků, než je vzorkovací frekvence signálu. Avšak pokud by program analyzoval větší úsek signálu, mělo by to za následek vysokou latenci. Uživateli by se vysoká latence jevila tím způsobem, že by program reagoval opožděně, a to je velmi nežádoucí jev.

Je třeba vyřešit tento problém. Potřebujeme dostatečně velký rámec signálu, ale zároveň, jelikož se jedná o zpracování v reálném čase, si nemůžeme dovolit zvýšit latenci, protože tím by se stal produkt nepoužitelným. Řešením tohoto problému je tzv. *zero padding*⁸. [2] Nulové vzorky prakticky simulují ticho a nijak signál dále neovlivňují. Tímto se dá získat více koeficientů, potažmo dosáhnout cíleného vyššího frekvenčního rozlišení bez vedlejšího efektu prodlužování odezvy programu.

3.2.2 Získávání základní frekvence

Z výsledného spektra je poté nutné vybrat koeficient s největší energií. Protože výsledkem Fourierovy transformace je pole komplexních čísel, pro porovnávání je z těchto čísel nutné získat jejich absolutní hodnotu (*magnitude*). Absolutní hodnotu komplexního čísla lze spočítat:

⁸ Technika, kdy se za rámec signálu přidávají nulové vzorky.

$$x = \sqrt{r^2 + j^2} \quad (3.5)$$

Proměnná r zde reprezentuje reálnou složku komplexního čísla, proměnná j imaginární složku. Na základě získaných absolutních hodnot jednotlivých koeficientů je vybrána největší hodnota, resp. koeficient Fourierovy transformace s největší energií. Index vybraného koeficientu je poté nutné přepočítat na frekvenci podle vzorce 3.4.

3.2.3 Rozpoznávání not se stejnou základní frekvencí

V této fázi je možno detekovat různé noty, nikoliv však sekvenci tónů se stejnou frekvencí. Tento problém se dá řešit pomocí sledování energie koeficientu. Pokud jsou detekovány dva tóny se stejnou frekvencí, je nutné porovnat jejich hodnotu (zatím jsme pracovali pouze s indexy). Logicky, pokud je hodnota aktuálního koeficientu větší než hodnota předchozího se stejným indexem, pravděpodobně se jedná o začátek nového tónu.

3.3 Ostatní metody pro rozpoznávání

Existují i další metody pro rozpoznávání frekvence. Jedná se tedy především o metody, které jsou založené na zpracovávání čistého signálu, tedy neprovádí se předem žádné FFT ani jiné transformace. Jedna z metod je založena na filtrování signálu bankou filtrů a následném porovnávání získaných hodnot z jednotlivých filtrů.[1] Tato metoda má jeden velký nedostatek, a to obrovskou náročnost na operační paměť a procesor. Dalším negativem je její poměrně složitá implementace.

Další metodou, resp. rodinou metod je určování základní frekvence pomocí korelačních funkcí.[3] V této rodině existuje hned několik velice podobných postupů k získání základní frekvence, bohužel vyskytuje se zde velký problém s přesností. Tyto metody pracují rovněž nad čistým signálem a jsou založeny na určování podobnosti úseků signálů. Tyto metody mohou fungovat s vysokou přesností nad spojitými signály, nikoliv však nad diskrétními. Často se stává, že metoda nedetekuje správně jednu periodu, ale například polovinu periody, nebo její násobky. Analyzátor by tak často detekoval tóny ležící o *oktávu*⁹ výše nebo níže.

⁹ Interval, který obsahuje 12 půltónů.

4 Použité technologie

Pro širší možnosti použití produktu této práce bylo použito několik technologií. Výsledkem je tak relativně univerzální aplikace, kterou lze integrovat prakticky do jakéhokoliv programu pro zpracování hudby.

4.1 MIDI

Musical Instrument Digital Interface, zkráceně MIDI, je mezinárodní specifikace vydaná pro hudební průmysl, která umožňuje v reálném čase široké škále hudebních nástrojů spolu komunikovat. Tato technologie byla uvedena na trh v roce 1982 a dodnes je spravována organizací MIDI Manufacturers Association. Důvodem zavedení této technologie byla absence jakéhokoliv standardu pro komunikaci mezi různými hudebními nástroji, především tedy syntetizéry.[6]

Zavedení této technologie přineslo řadu výhod. Asi největší výhodou je znatelná redukce kabeláže mezi jednotlivými zařízeními, a s tím spojená menší náročnost na různé další převodníky, které zajišťovaly kompatibilitu mezi zařízeními různých značek. Další nespornou výhodou je univerzalita použití. V dnešní době se MIDI rozhraní používá také pro ovládání různých efektových procesorů.

MIDI rozhraní je sériové a je možné vybírat mezi 16 kanály. Je velmi důležité si uvědomit, že zařízení, která jsou propojena pomocí MIDI rozhraní, mezi sebou posílají MIDI informace, nikoliv výsledný zvuk.

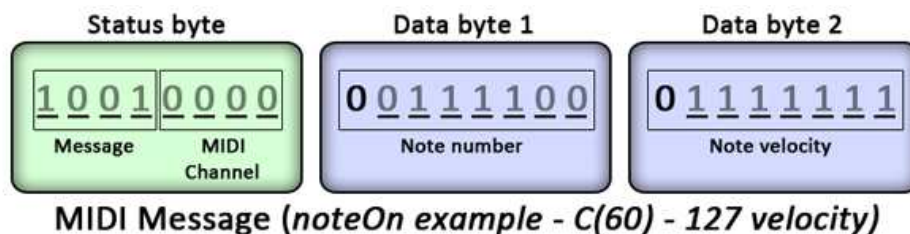
4.1.1 MIDI informace

Informací přenášených přes MIDI rozhraní je několik typů. Za zmínku určitě stojí tzv. SysEx (System Exclusive) zprávy. Tyto zprávy slouží především k nastavování pokročilých vlastností mezi zařízeními, např. nastavení celkové hlasitosti, mikroladění, požadavek pro výpis textu na displej apod. Do rodiny SysEx informací se zahrnují i speciální zprávy sloužící pro nastavení konkrétního parametru u specifického zařízení. Takové zprávy ovšem nejsou nijak standardizovány, většinou jsou dodefinovány výrobcem daného zařízení.[4]

Další rodinou informací jsou tzv. systémové zprávy (MIDI system messages). Tyto zprávy jsou určeny pro nastavování hodnot, které jsou společné pro všechny kanály. Typickým příkladem těchto zpráv je volba kanálu, nebo žádost o naladění syntetizéru.

Poslední skupinou MIDI informací jsou tzv. kanálové MIDI zprávy. Tyto informace jsou určeny k posílání požadavků na přehrání noty, zastavení přehrávání, ohýbání not apod.

4.1.2 Struktura kanálové MIDI informace



Obrázek 4.1 – Struktura kanálové MIDI informace s příkladem zprávy noteOn tónu C s maximální hlasitostí na prvním kanálu¹⁰

¹⁰ V MIDI zprávách se kanály číslovají od 0, resp. první kanál je reprezentován binární hodnotou 0000.

Kanálová MIDI informace má velikost 3 bajty. Na obrázku výše je uvedena struktura této informace s příkladem zprávy pro zahrání noty C s maximální hlasitostí na prvním kanále. Ve status bajtu se nachází dvě čtyřbitové informace. První udává typ požadavku, druhá udává kanál. Následují dva datové bajty, které nesou informace o čísle noty (první bajt) a její síle (druhý bajt). Podobné schéma mají i systémové MIDI zprávy, liší se však číslem požadavku a významem hodnot datových bajtů.[9]

4.1.3 Využití v tomto projektu

Jak již bylo uvedeno výše, pomocí MIDI informací lze ovládat většinu syntetizérů. Stejně tak existují tzv. virtuální MIDI rozhraní, která se tváří jako hardwarové zařízení. Ve skutečnosti se jedná pouze o aplikaci, která přijatá data posílá na rozhraní, které se tváří jako MIDI vstup. MIDI data můžeme tedy posílat v rámci jednoho počítače, ale také můžeme posílat analyzovaná data do externího zařízení, čímž se dále rozšiřují možnosti použití aplikace, jež je předmětem této práce.

4.2 Steinberg VST

Virtual Studio Technology, zkráceně VST je rozhraní vyvinuté firmou Steinberg.[7] Pomocí tohoto rozhraní lze vytvářet různé moduly – též nazývané jako VST pluginy. Tyto moduly se dají použít v různých hostitelských aplikacích, které samozřejmě musí podporovat technologii VST. Takové aplikace jsou potom nazývány VST hosty. Mezi nejznámější VST hosty lze zařadit např. Cubase od firmy Steinberg, programy Fruity Loops a FL Studio od firmy Image-Line nebo program Ableton Live.

Firma Steinberg dala pro programátorské účely volně k dispozici tzv. VST SDK¹¹. Jde o sadu tříd napsaných v jazyce C++ včetně detailní dokumentace a různých příkladů použití. Součástí tohoto balíku je také soubor tříd pro popis grafického uživatelského rozhraní pluginu – VST GUI, kde jsou implementované třídy pro tlačítka, posuvníky, displeje, otočné potenciometry a další. Technologie VST je kompatibilní se systémy Mac OS a Microsoft Windows. Existuje sice i modifikace tohoto balíku, která je použitelná v systémech typu UNIX, nejedná se ale o oficiální produkt firmy Steinberg.

4.2.1 Verze VST

V poslední době se na trhu objevila nová verze této technologie – verze 3. Nejedná se však o rozšíření starší verze 2.4, která je v dnešní době asi nejvíce rozšířena, ale jde o úplně nové přepracování celé technologie VST. Základní rozdíl mezi verzemi VST 2.4 a VST 3 je mj. ve způsobu definice průběhu tónu. Zatímco technologie VST 2.4 umožňuje jednomu tónu definovat pouze sílu, výšku a délku, s využitím verze VST 3 je možnost definovat celý průběh tónu včetně zesílení, zeslabení a jiných jeho atributů, což je nesporná výhoda VST 3 oproti VST 2.4¹². Bohužel jen málo VST hostů podporuje tuto novější verzi, proto využití starší verze v pluginech stále převažuje.

4.2.2 VST & VSTi

Existuje několik typů VST pluginů. Tyto typy jsou navzájem nekompatibilní, tedy nelze použít syntetizér tam, kde je místo určené pro efekt a obráceně. VST pluginy dělíme na 3 kategorie:

- První kategorií jsou tzv. VST nástroje (VSTi – VST Instrument). Jedná se o pluginy, kterým jsou na vstup přiváděna MIDI data, na jejichž základě je produkován zvuk. Prakticky se jedná o softwarové syntetizéry. Tento druh VST pluginů je na trhu asi nejrozšířenější.

¹¹ VST SDK lze stáhnout z URL: <http://www.steinberg.net/en/company/developer.html>

¹² Novinky verze VST 3 jsou popsány zde: <http://www.steinberg.net/en/company/technologies/vst3.html>

- Druhou kategorií jsou VST efekty (VST – VST Effect). Tyto pluginy dostávají na vstup audio signál, který určitým způsobem zpracovávají. Na jejich výstupu je opět audio signál. Mezi tyto efekty můžeme řadit prakticky jakékoliv zvukové procesory. Většinou se jedná o efekty typu *delay*, *flanger*, *reverb*, *chorus* apod.
- Třetí kategorií tvoří MIDI efekty. Princip těchto efektů spočívá v analýze MIDI dat a jejich modifikaci. Na výstupu těchto pluginů jsou opět MIDI data. Typickým MIDI efektem je tzv. *pitch shifter*. Jedná se o typ efektu, jenž dokáže upravovat výšku tónů, které jsou efektem zpracovávány.

4.2.3 Využití v projektu

Důvodem použití právě této technologie je její rozšířenost v oblasti hudebního průmyslu. Program bude implementován jako VST efekt. Použití této technologie zvyšuje potenciál využití aplikace v praxi. Vzhledem k tomu, že tato aplikace nijak nepracuje s průběhem tónu, bude použita verze VST 2.4. Dalším důvodem, proč použít zrovna tuto verzi této technologie je již zmíněná velmi rozšířená podpora v aplikacích.

4.3 Implementační jazyk a prostředí

Vzhledem k nárokům na rychlost výpočtu bude program vytvořen v jazyce C++. V projektu budou využity některé funkce pro práci s MIDI zařízením z prostředí Windows API[15], tedy nejvhodnější vývojové prostředí je Microsoft Visual Studio s kompilátorem MS Visual C++. Při implementaci bude použita šablona pro VST pluginy od firmy Steinberg, v souladu s licenčními podmínkami proto na obrázku 4.2 uvádím logo této technologie.



Obrázek 4.2 – logo technologie Steinberg VST

Pro výpočet Fourierovy transformace bude použita knihovna FFTW 3¹³. Důvodem k použití právě této knihovny je její vysoká rychlost a vysoká míra optimalizace. Tato knihovna je mj. použita i v programech pro výpočet matematických formulí Maple a Matlab.

¹³ Knihovna FFTW 3 je stažitelná ze stránky: <http://www.fftw.org/download.html>

5 Návrh aplikace

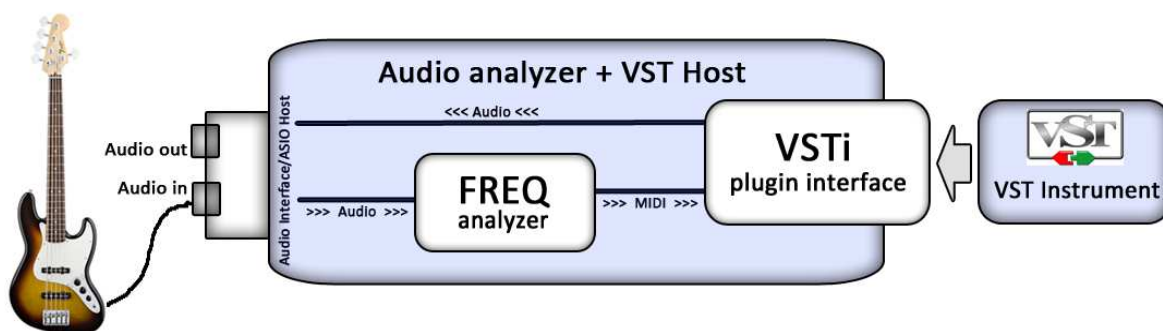
Předem je důležité sdělit, že cílem tohoto projektu není syntetizovat zvuk, pouze jej analyzovat, proto ke zpětné syntéze použijeme již hotových VSTi produktů.

Od původního návrhu došlo k velkým změnám ve struktuře programu. V této kapitole budou detailně specifikovány jednotlivé koncepty aplikace, včetně rozvedení důvodů, proč jsou nevhodné nebo naopak – proč byl daný koncept vyhodnocen jako nejlépe použitelný.

5.1 Koncept návrhu

5.1.1 Samostatná aplikace s podporou VSTi

Jedná se o původní koncept celé aplikace. Princip tohoto modelu je relativně jednoduchý a výsledný produkt nevyžaduje téměř žádnou podpůrnou programovou výbavu, snad jen ASIO ovladače pro zvukovou kartu uživatele.



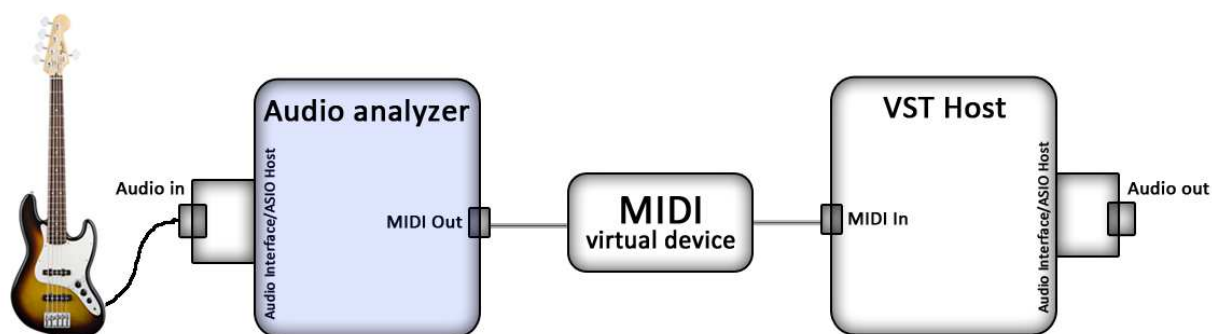
Obrázek 5.1 – Původní návrh aplikace

Funkce tohoto modelu je následující. Model se pomocí ASIO nebo jiných ovladačů připojí ke zvukové kartě, odkud bere vstupní signál z připojeného hudebního nástroje, nebo podobného zdroje. Následně dochází ke zpracování vstupního audio signálu pomocí algoritmů popsanych v kapitole 2. Na základě zjištěného tónu je vytvořena MIDI informace a ta je přes rozhraní VST poslána do speciálního modulu – VSTi pluginu. Tento modul na základě přijatých MIDI informací je schopen syntetizovat určitý zvuk. Volba VSTi pluginu závisí pouze na uživateli. Zde je potřeba sesbírat výsledný zvuk z výstupu použitého VSTi a poslat na výstup.

Na první pohled působí tento koncept jako nejlépe použitelný. Dalo by se říci, že toto je ideální model funkčnosti aplikace. Avšak jsou tu hned dva problémy. Tvorba tohoto modelu by znamenala implementovat minimálně rozhraní pro VSTi pluginy, což je poměrně komplikovaná záležitost i pro velké firmy působící v tomto oboru několik let. Navíc, ne všechny VSTi pluginy se chovají korektně a ošetření případných chyb v těchto zásuvných modulech je poměrně nelehké. Další složitostí by byla implementace hostování ASIO ovladačů, tj. ovladačů které zdatelně snižují latenci při práci se zvukem.

5.1.2 Samostatná aplikace s MIDI výstupem

Další návrh je o něco jednodušší, než předchozí. Cílem tohoto návrhu bylo eliminovat co nejvíce problémů souvisejících s implementací. Proto byla zvolena možnost spolupráce s externím programem, který je schopen VSTi hostovat. Tímto odpadají veškeré problémy související s tvorbou VSTi hosta, zůstává pouze implementace hostování ASIO ovladačů.



Obrázek 5.2 – Návrh aplikace s MIDI výstupem

Funkce tohoto modelu je poněkud složitější na pochopení a je také náročnější na programovou vybavu uživatele. Modul, jenž je předmětem této práce, je připojen ke zvukové kartě, odkud bere zvuková data. Ta dále zpracovává na základě zmíněných algoritmů. Opět probíhá tvorba MIDI informací na základě určeného tónu. Tyto informace jsou poté odesílány na interní MIDI rozhraní počítače, odkud jsou dále posílány do dalšího programu, který se stará o veškeré další zpracování od VSTi hostingu až po posílání výsledného zvuku na výstup.

Tento model má oproti předchozímu jednu nespornou výhodu. Vzhledem k implementaci MIDI rozhraní je možné pomocí tohoto modelu posílat MIDI data i do jiných hardwarových zařízení připojených k počítači, kde může být zvuk syntetizován mnohdy ve vyšší kvalitě. Předpokladem k použití tohoto typu syntézy je samozřejmě dostupný hardwarový MIDI port u obou zařízení.

Bohužel kvůli faktu, že ASIO ovladače jsou pouze *single-host*, tedy jedno audio zařízení může být pomocí ASIO připojeno pouze k jednomu programu, a tento koncept obsahuje dva programy, které komunikují se zvukovým zařízením, nelze je v tomto modelu použít.[8] V tomto případě by nezbyla jiná možnost, než použít standardní systémové ovladače (*WDM*). Je tu však další velmi kritický faktor, a to je latence (zpoždění zvuku). *WDM* ovladače běžně zpožďují zvuk o 70-200ms při jednom průchodu, v našem případě tohoto modelu by mohlo celkové zpoždění dosáhnout až 400-500ms, což už je za hranicí použitelnosti.

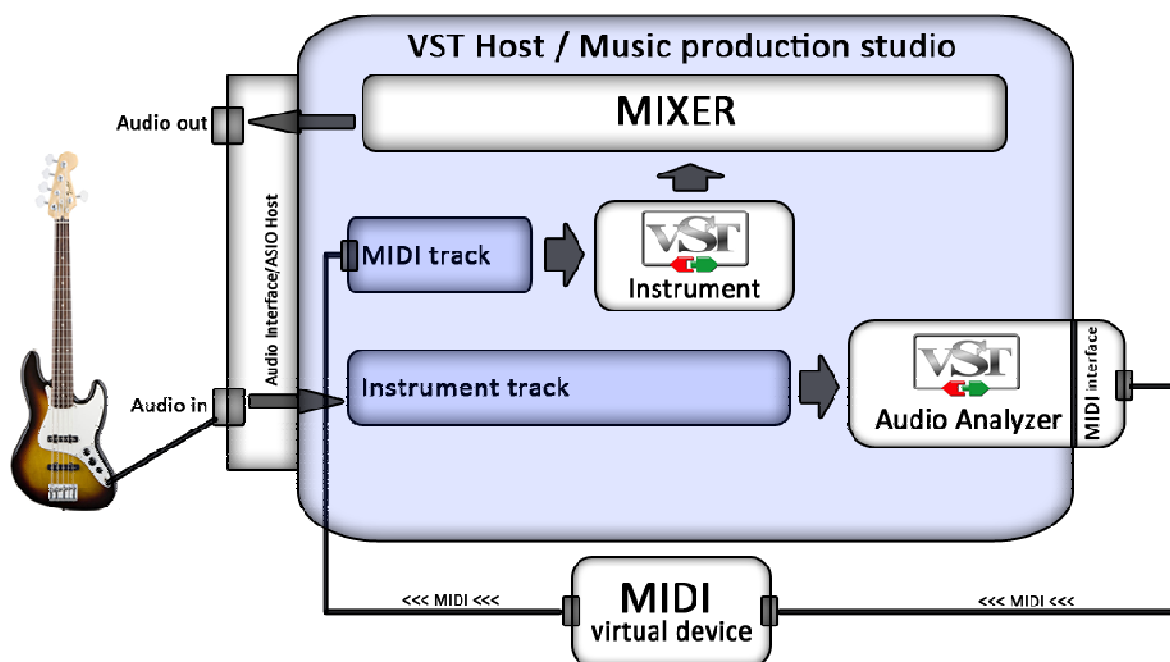
5.1.3 VST plugin s MIDI rozhraním

Konečný model byl navržen s ohledem na použitelnost, přenositelnost, latenci a implementační náročnost. Výsledkem je VST plugin s MIDI rozhraním, na které jsou odesílána získaná MIDI data. Výhodou tohoto konceptu je skutečnost, že je možné posílat MIDI data jak na virtuální rozhraní, z kterého může hostující aplikace tato data sbírat, tak na fyzické MIDI rozhraní zvukové karty, pomocí kterého lze tato data přenášet do externích syntetizérů.

Tento model má velmi podobné nároky na programové vybavení jako předchozí model, avšak zatímco u předchozího modelu se jednalo o samostatný program, tento model je zkompileován pouze jako dynamická knihovna (*DLL*).

Zásadní rozdíl také spočívá v náročnosti na implementaci. Zatímco u předchozích dvou konceptů bylo nutno složitě integrovat rozhraní pro hostování ASIO ovladačů, zabezpečit plynulý tok dat ze vstupu i na výstup nebo v horším případě integrovat složité rozhraní pro hostování VSTi, v tomto modelu nic z toho není třeba řešit, protože se o vše stará hostující aplikace.

S tím souvisí i daleko širší použití. Vzhledem k tomu, že se jedná o VST plugin, dá se použít prakticky v jakémkoli programu, který umí používat VST, což jsou dnes prakticky všechny profesionální programy pro tvorbu a editaci hudby.



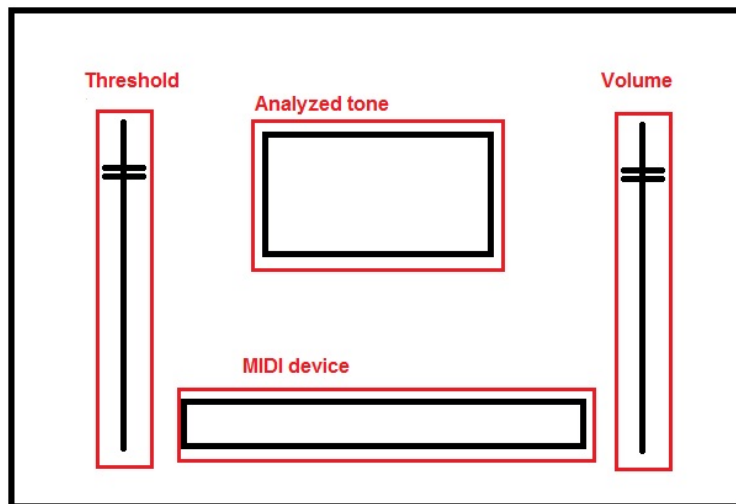
Obrázek 5.3 – návrh aplikace jako VST pluginu

Na obrázku 5.3 je zobrazeno detailní schéma propojení hostující aplikace s tímto modulem. Zmíněný modul – *Audio Analyzer* zde slouží opravdu jen k účelu analýzy zvuku a posílání MIDI dat, o vše ostatní se stará hostující aplikace.

Na obrázku se vyskytují dvě různé stopy – MIDI stopa (*MIDI Track*) a Audio stopa (*Instrument track*). MIDI stopě jsou přiváděny na vstup MIDI informace, tedy nejedná se o reprodukovatelný zvuk. Tato data jsou potom posílána do příslušného VST Nástroje (*VSTi*), kde se podle typu nástroje mění zmíněné informace na vzorky audio signálu. Audio stopa má na vstupu diskretní audio signál, tento signál lze potom upravovat pomocí VST Efektů, mezi které se řadí i zmíněný modul *Audio Analyzer*, jenž je předmětem této práce.

5.2 Grafické uživatelské rozhraní

Důvodem zavedení grafického uživatelského rozhraní (dále jen GUI) je fakt, že některé parametry programu by mělo být snadné za běhu měnit. GUI je prakticky jedinou schůdnou možností. Otevírání konzole by bylo příliš složité a pro běžného uživatele nepohodlné.



Obrázek 5.4 – Návrh GUI aplikace

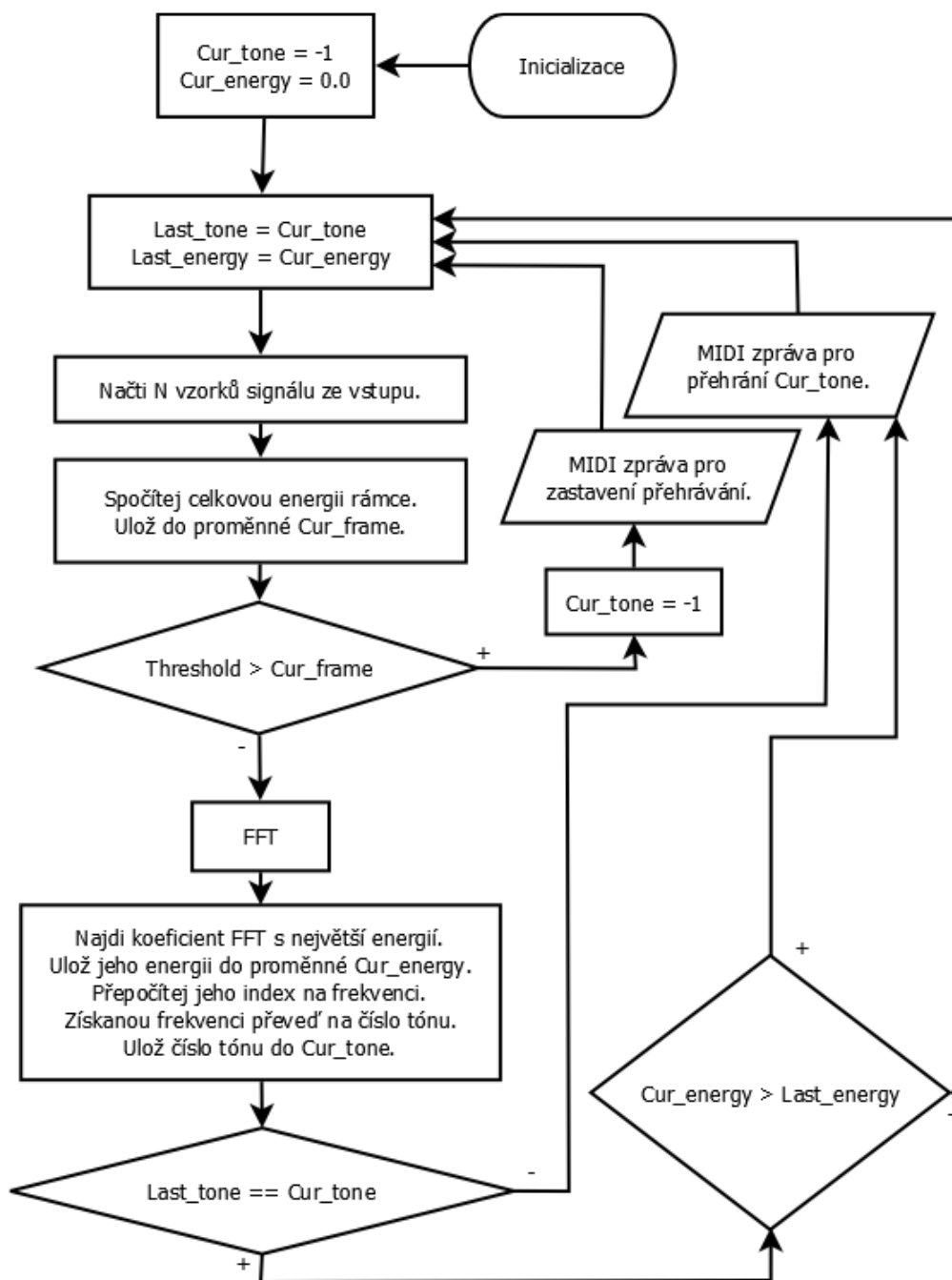
Na obrázku 5.4 je zobrazen přibližný náčrt GUI se zvolenými ovládacími prvky. Prvek *Analyzed tone* je pouze informativní a slouží výhradně k zobrazování aktuálně rozpoznávaných not. Nelze jej tedy řadit mezi ovládací prvky. Mezi ovládací prvky patří následující:

- **Threshold** - Tento ovladač je určen k nastavení hladiny mezi zvukem a šumem, tedy pomocí něj lze nastavit úroveň hlasitosti, jakou musí daný signál dosáhnout, aby byl analyzován. Důvodem zavedení tohoto prvku jsou odlišné vlastnosti různých zvukových zařízení a taktéž různá síla signálu produkovaného hudebními nástroji.
- **Volume** - Důvod zavedení tohoto ovládacího prvku je poměrně zřejmý, jde tedy o ovládání výstupní hlasitosti, v tomto případě se jedná o korekci MIDI parametru *velocity*.
- **MIDI device** - Jak bylo již řečeno v předchozí kapitole, vybraný koncept návrhu aplikace umožňuje posílat data do externích zvukových zařízení. Tento ovladač slouží k výběru zařízení, na které chceme analyzovaná data posílat – interní virtuální MIDI rozhraní (počítač jím však musí disponovat), nebo fyzické, které disponuje použitelným MIDI portem.

6 Implementace

6.1 Struktura aplikace

Aplikace je členěna do tří základních modulů (objektů), které mají specifické vlastnosti a funkce. Tyto objekty fungují zcela autonomně. Funkce aplikace je popsána vývojovým diagramem na obrázku 6.1.



Obrázek 6.1 – Algoritmus převodu zvuku na MIDI

6.1.1 Modul s VST rozhraním

Hlavní modul je realizován jako objekt. Tento modul je navržen dle požadavků pro kompatibilitu s VST Effect rozhraním, resp. splňuje požadavky pro použití aplikace jako VST Efektu, nikoli VSTi (VST Nástroje). Tento modul má na starost veškerou synchronizaci s hostující aplikací, včetně získávání informací nutných pro správnou detekci tónů, např. *sample rate* (počet audio vzorků za sekundu).

Součástí tohoto modulu je také grafické uživatelské rozhraní, které je realizováno speciální třídou pomocí frameworku VST GUI, taktéž dodávaného firmou Steinberg. Na tento modul jsou přímo připojeny dva moduly, první pro analýzu zvuku, druhý pro posílání MIDI dat.

6.1.2 Modul pro posílání MIDI dat

Tento modul je určen k posílání analyzovaných dat na určené výstupní MIDI rozhraní. Jedná se prakticky o objektovou nastavbu nad funkcemi z rozhraní Windows API pro přístup k MIDI zařízení, která je speciálně upravená pro monofonii. Součástí modulu je automatická kontrola, která zabráňuje přehrávání více not současně. V případě, že se objeví požadavek pro posílání tzv. *noteOn* informace (událost udávající počátek přehrávání určitého tónu) a modul je ve stavu, kdy nebylo přehrávání předchozího tónu ukončeno, je nejprve poslána informace pro ukončení předchozí noty, až poté je proveden zmíněný příchozí požadavek. Modul poskytuje také základní funkce pro otevření a uzavření výstupního MIDI zařízení a funkce pro získání názvů zařízení.

6.1.3 Modul pro frekvenční analýzu

Tento modul zahrnuje celý rozpoznávací proces. Vstupním bodem tohoto modulu je funkce *addData(double *data)*. Parametrem této funkce je ukazatel na pole předem definovaného počtu vzorků. Jejím výstupem je struktura, která obsahuje informace o požadovaném úkonu pro modul starající se o posílání MIDI dat (dále jen MIDI modul). Informace mohou být trojího typu. *MIDI_PLAY* je typ informace sdělující MIDI modulu, že má poslat požadavek na zahrání určité noty. *MIDI_STOP* je typ požadavku pro zastavení přehrávání, a *MIDI_NOACTION* je informace, při které se neprovádí nic.

Než dojde k jakémukoliv rozpoznávání frekvence, je rámec prvotně analyzován. Cílem prvotní analýzy je zjistit, zda se jedná o zvukový rámec, s kterým lze dále pracovat, nebo se jedná pouze o šum. Princip spočívá ve spočítání celkové energie rámce podle vzorce 6.1, kde x je hodnota zpracovávaného vzorku v rámci a N je velikost rámce.

$$E_{celk} = \sum_{n=0}^N x^2 [n] \quad (6.1)$$

Celková energie je poté porovnána s hraniční hodnotou – tzv. *threshold*. Pokud je celková energie rámce menší než zmíněná hodnota, rámec je zahozen a MIDI modulu je poslána zpráva *MIDI_STOP*. V opačném případě modul pokračuje v analýze.

Analýza je založena na metodě uvedené v kapitole zabývající se teorií rozpoznávání. Pokud rámec projde úspěšně prvotní analýzou, je dále doplněn o nulové vzorky na délku 65536 vzorků. Počet vzorků byl záměrně zvolen jako násobek dvou, důvodem je rychlejší průběh FFT. Na základě dat upraveného rámce je provedena FFT, výsledkem je pole komplexních čísel. Pro reprezentaci komplexních čísel byl zvolen speciální datový typ *fftw_complex*. Tento datový typ je součástí použité knihovny FFTW 3.

Modul obsahuje tabulku frekvencí jednotlivých čtvrttónů. Každé sousední dva čtvrttóny v tabulce tvoří interval, který reprezentuje nějaký tón. Po dokončení FFT následuje výběr koeficientu s největší absolutní hodnotou. Index koeficientu je poté přepočten na frekvenci. Na základě příslušnosti vypočtené frekvence do jednoho z intervalů je vypočten tón. Tento tón je poté porovnán s předchozím analyzovaným tónem. V případě, že se tyto tóny neshodují, je odeslán signál

MIDI_PLAY s příslušnými informacemi. Pokud se tyto dva tóny shodují, probíhá kontrola intenzity těchto dvou tónů. V případě, že aktuální tón má větší intenzitu než předchozí analyzovaný, pravděpodobně se jedná o novou notu a MIDI modulu je odeslán signál *MIDI_PLAY* s příslušnými daty. V opačném případě je signalizována akce *MIDI_NOACTION*. Vzhledem ke skutečnosti, že může docházet k drobným nepřesnostem v intenzitě koeficientů, resp. frekvencí, je zde použita drobná korekce, kdy analyzátor vyhodnotí rámeček jako novou notu pouze v případě, že je intenzita příslušného koeficientu o 20% vyšší, než intenzita stejného koeficientu v předchozím rámečku.

6.2 Uživatelské rozhraní aplikace

Primárním cílem bylo rozmístit a zobrazit ovládací prvky tak, aby na první pohled bylo patrné, k čemu mají být určeny. Jelikož cílová skupina uživatelů jsou především lidé hudebně založení a lze tedy předpokládat, že tito lidé mají aspoň základní zkušenosti v oblasti zvukové techniky, ovládací prvky byly vytvořeny tak, aby připomínaly ovládací prvky reálných komponent pro mixáž a zpracování zvuku. Výsledné GUI je zobrazeno na obrázku 6.2.



Obrázek 6.2 – Konečný vzhled aplikace

Ovládací prvky nebyly oproti návrhu nijak pozměněny, upravena byla pouze funkce tahového potenciometru *Output gain* (v návrhu byl tento prvek uveden jako *Volume*). Tento potenciometr, jak již bylo zmíněno v předchozí kapitole, slouží k ovládání výstupní hlasitosti. Avšak vzhledem k tomu, že tato aplikace nemá prakticky žádný zvukový výstup, ale jejím výstupem jsou MIDI informace posílané na zvolené zařízení, jedná se pouze o nastavení hodnoty síly úderu u posílaných not, tzv. *velocity*. Logický důsledek tohoto faktu je skutečnost, že při pohybu jezdcem během přehrávání libovolné noty se výsledná hlasitost projeví až na následujícím tónu po změně hodnoty potenciometru.

7 Testování

7.1 Návrh testů

Testování tohoto produktu je nedílnou součástí vývoje. Pro testování byly použity tři metody. Jedna metoda je založena na rozdílech dvou MIDI sekvencí, tedy jde o automatizované testování, druhá metoda je částečně automatizována s rozdílem, že jsou použity reálné hudební nástroje. Třetí metoda, resp. její výsledky jsou poněkud subjektivnější, jde o hodnocení několika uživatelů, kterým byl tento produkt představen.

7.1.1 Typy chyb při analýze zvuku

Během výsledného porovnávání může dojít k detekci následujících chyb:

Insertion error – Jedná se o jednu ze dvou méně závažných chyb, podstatou této chyby je vložení tónu, který v originále chybí. Tato chyba se nejčastěji vyskytuje při hraní not s prudkým atakem, nebo také na konci dlouhých tónů, kdy se hlavní i vedlejší frekvence tónu začínají slévat dohromady. Na obrázku 7.1 je tato chyba graficky znázorněna.



Obrázek 7.1 – Insertion error

Deletion error – Taktéž jedna z méně závažných chyb. Její podstatou je zanedbání některých tónů originálu. Tato chyba se nejčastěji vyskytuje při posloupnostech několika tónů se stejnou výškou, obzvláště když jsou tóny hrány mdlé nebo s malým atakem. Druhým nejčastějším důvodem výskytu je skutečnost, že hlasitost hraných tónů nedosahuje nastavené hranice mezi šumem a zvukem. Na obrázku 7.2 jsou opět znázorněny tyto typy chyb.



Obrázek 7.2 – Deletion error

Substitution error – Závažnější chyba, která je způsobena špatnou analýzou zvuku. Podstatou této chyby je záměna noty originálu za notu jiné výšky. Pokud nastává tato chyba, tak se jedná v lepším případě o špatně naladěný nástroj, v horším případě se jedná o chybu analyzátoru.



Obrázek 7.3 – Substitution error

7.1.2 Automatizované testování

Tento druh testování je založen na porovnávání dvou MIDI souborů, resp. sekvencí, kdy první soubor slouží jako předloha pro přehrávání (dále jen originál). Druhý soubor je generován při přehrávání originálu na základě rozpoznávaných tónů. Vzhledem k tomu, že při přehrávání, posílání MIDI dat i zpětné analýzu vzorků signálu dochází ke zpoždění, drobné časové nepřesnosti jsou zde zanedbávány.

Kvalita rozpoznávání bývá rozdílná na vysokých a nízkých frekvencích, bylo tedy provedeno několik testů, z nichž každý byl zaměřen na jiný frekvenční rozsah. Abychom co nejvíce eliminovali lidský faktor, byly pro toto testování použity různé VSTi pluginy pro simulaci reálných nástrojů.

Pro vyhodnocení přesnosti rozpoznávání byl použit program HResults, který je součástí balíku HTK¹⁴. Příklad zprávy programu je zobrazen na obrázku 7.4.

```
===== HTK Results Analysis =====  
Date: Sun May 13 02:48:15 2012  
Ref : ./music/ref.mlf  
Rec : ./music/test.mlf  
----- Overall Results -----  
WORD: %Corr=75.00, Acc=62.50 [H=6, D=1, S=1, I=1, N=8]  
=====
```

Obrázek 7.4 – Příklad výstupu programu HResults

Ukazatel *Corr* značí správnost a je uveden v procentech. *Acc* je koeficient přesnosti. Při velké míře chyb typu *Insertion error* může tento koeficient nabývat záporných hodnot. V hranatých závorkách jsou uvedeny další symboly. Symbol *H* značí počet správných not, *D* značí počet chyb typu *Deletion error*, *S* počet chyb typu *Substitution error* a *I* chyby typu *Insertion error*. *N* je celkový počet tónů v originálním souboru. Příklad testovacích souborů¹⁵ je uveden v příloze A.

7.1.3 Testování s použitím reálných nástrojů

Tato sada testů je zaměřena na rozpoznávání frekvence u různých reálných hudebních nástrojů, především elektrických strunných nástrojů. Důvodem volby je přítomnost elektronického snímače, pomocí kterého se dá zvuk snadno vést do počítače. Prakticky jde o testování podobné předchozímu uvedenému. Součástí jednotlivých testů je i poslechový test a hodnocení posluchače.

7.1.4 Uživatelské testování

Tento test je spíše zaměřen na užitečnost a prodejnost tohoto produktu. Podstatou tohoto testování je předvedení produktu cílové skupině uživatelů, tedy hudebníkům. Tito uživatelé měli možnost vlastního vyzkoušení produktu, poté byli požádáni o krátké zhodnocení aplikace a napsání krátké zprávy, ve které se měli zaměřit především na ovládání a použitelnost produktu v praxi.

7.2 Vyhodnocení testů

Veškeré testování probíhalo na operačním systému Microsoft Windows 7. Jako hostitelská aplikace pro testování byl použit produkční software FL Studio 8.

¹⁴ Produkt je stažitelný z URL: <http://htk.eng.cam.ac.uk/download.shtml>

¹⁵ Jedná se o speciální testovací soubory s příponou *mlf*.

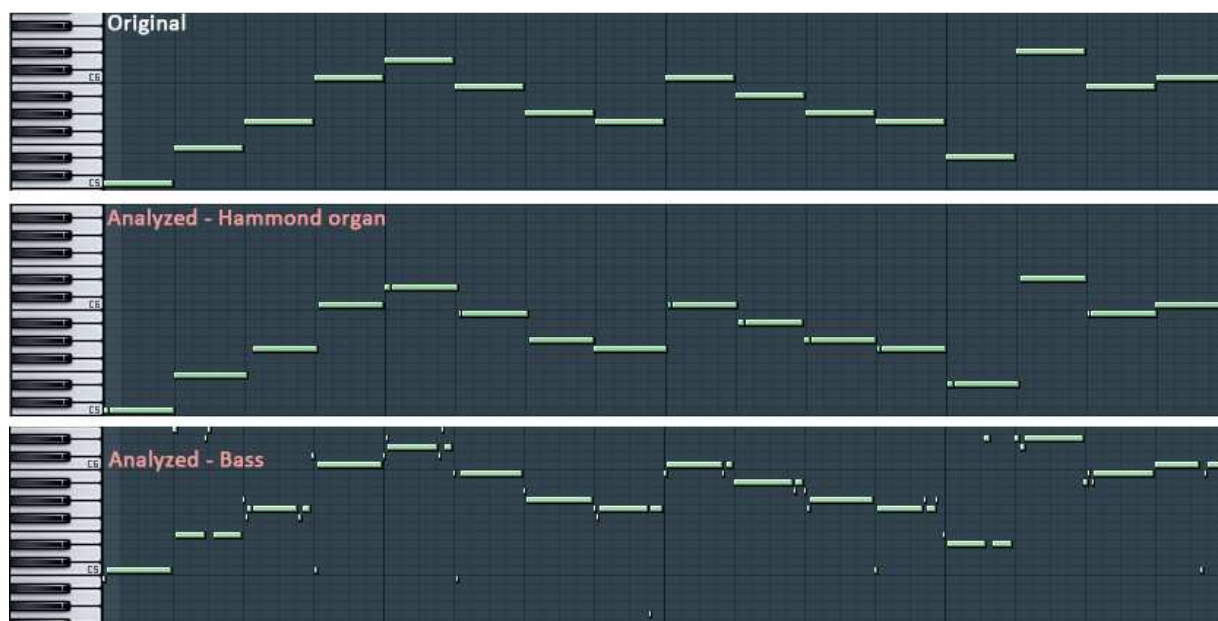
7.2.1 Vyhodnocení automatizovaných testů

Pro simulaci reálných nástrojů byl použit modul Hammondových varhan VST pluginu KeyRig od firmy M-Audio. Druhým použitým nástrojem byl VST plugin *BooBass*, který je součástí použitého produkčního softwaru a simuluje zvuk elektrické baskytary.

Tyto nástroje byly zvoleny na základě frekvenční složitosti výsledného syntetizovaného zvuku. Zatímco modul Hammondových varhan dokáže syntetizovat téměř čistý tón, tedy výsledná analýza by měla být přesnější, modul *BooBass* produkuje zvuk o poměrně složité frekvenční charakteristice, jednotlivé tóny mají navíc ostřejší atak, proto je rozpoznávání tónů zde daleko složitější.

Test vysokých frekvencí

Tento test byl zaměřen na detekci hlavních frekvencí v oblasti tónů C5-E6, frekvenčně přibližně mezi 500-1500Hz. Na obrázku 7.5 je zobrazen graf detekovaných tónů.



Obrázek 7.5 – Výsledky testů detekce vysokých frekvencí

Již z grafu je patrné, že analýza zvuku produkovaného modulem *BooBass* (část Analyzed – Bass) obsahuje daleko více chyb, než analýza zvuku Hammondových varhan. Jedná se především o chyby vložení (*Insertion error*). Důvodem je fakt, že analyzátor detekoval vedlejší frekvenci, která v daný moment svou intenzitou předčila hlavní frekvenci tónu, analyzátor tedy mylně detekoval nový tón. Skutečnost, že se jedná především o chyby vložení, potvrzuje i níže uvedená zpráva z programu HResults (viz obr. 7.6).

Hammond organ

```
===== HTK Results Analysis =====  
Date: Thu May 10 14:59:54 2012  
Ref : ./music/hi/ref.mlf  
Rec : ./music/hi/organ.mlf  
----- Overall Results -----  
WORD: %Corr=100.00, Acc=43.75 [H=16, D=0, S=0, I=9, N=16]
```

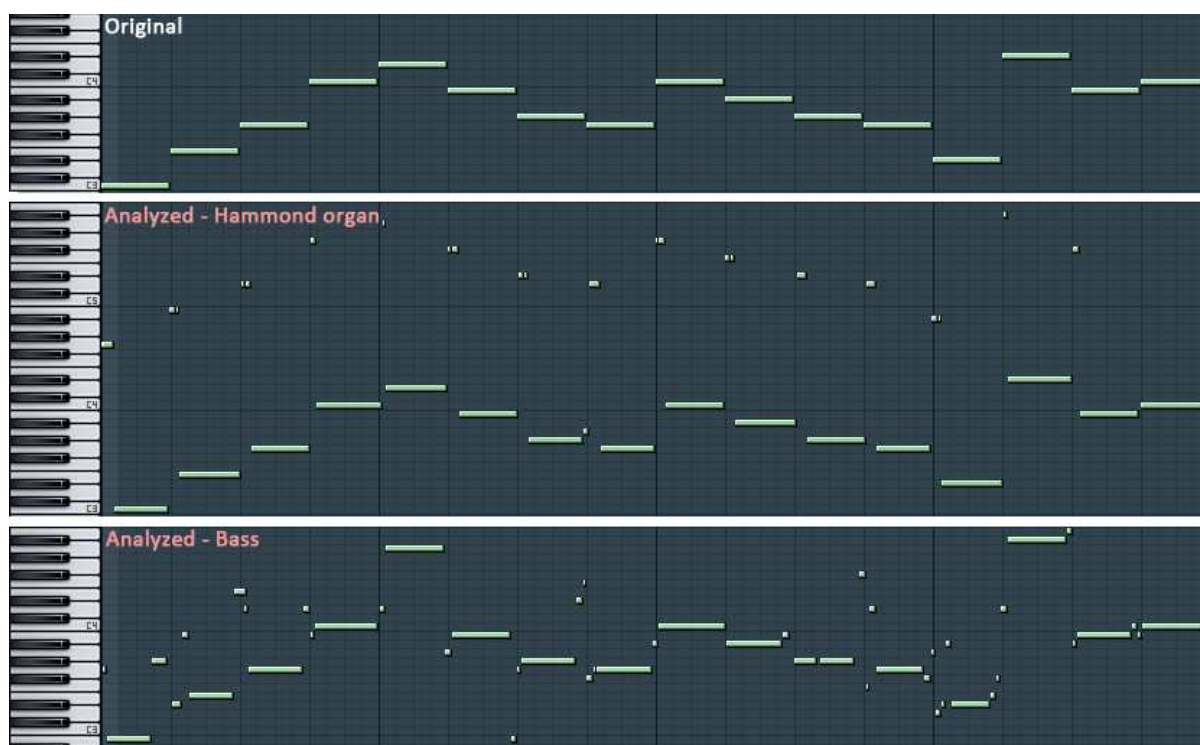
BooBass

```
===== HTK Results Analysis =====  
Date: Thu May 10 15:10:04 2012  
Ref : ./music/hi/ref.mlf  
Rec : ./music/hi/bass.mlf  
----- Overall Results -----  
WORD: %Corr=100.00, Acc=-156.25 [H=16, D=0, S=0, I=41, N=16]
```

Obrázek 7.6 – Výstup HResults pro test vysokých frekvencí

Test středových frekvencí

Tento test byl zaměřen na analýzu tónů v rozmezí C3 – F4, tedy frekvenčně přibližně mezi 130 – 400 Hz. Z níže obrázku 7.7 je patrné, že v nižších frekvencích má analyzátor daleko větší problémy s přesností detekce hlavní frekvence.



Obrázek 7.7 – Výsledky testů detekce středových frekvencí

Ačkoliv se chyb vložení objevilo více, procento ostatních chyb je pořád malé. Oproti předchozímu testu se v testu s využitím Hammondových varhan objevily drobné chyby na začátku téměř každého tónu. Tento nástroj umožňuje různá nastavení, kde je možno přidávat další hlasy. Tyto chyby jsou pravděpodobně způsobeny skutečností, že nástroj nebyl nastaven na čistý zvuk, ale měl do výsledného zvuku přimíchané další harmonické frekvence.

Hammond organ

```
===== HTK Results Analysis =====  
Date: Thu May 10 15:13:24 2012  
Ref : ./music/mid/ref.mlf  
Rec : ./music/mid/organ.mlf  
----- Overall Results -----  
WORD: %Corr=100.00, Acc=-37.50 [H=16, D=0, S=0, I=22, N=16]
```

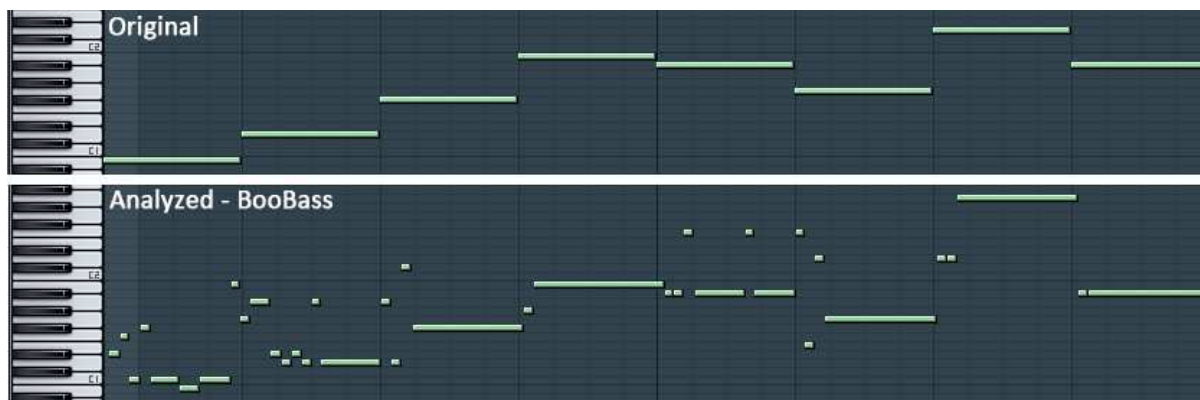
BooBass

```
===== HTK Results Analysis =====  
Date: Thu May 10 15:14:22 2012  
Ref : ./music/mid/ref.mlf  
Rec : ./music/mid/bass.mlf  
----- Overall Results -----  
WORD: %Corr=87.50, Acc=-106.25 [H=14, D=0, S=2, I=31, N=16]
```

Obrázek 7.8 – Výstup HResults pro test středových frekvencí

Test nízkých frekvencí

Tato část testování byla zaměřena především na nejnižší používané tóny. Jedná se o rozsah tónů H0 – F2. Prakticky se jedná o nižší frekvence rozsahu pětistrunné baskytary. Frekvenčně se jedná přibližně o rozmezí 30 – 80 Hertzů. Pro tento test byl použit pouze modul *BooBass*.



Obrázek 7.9 – Výsledek testu detekce nízkých frekvencí

Na obrázku 7.9 již lze poznat, že v oblasti nízkých frekvencí má analyzátor nemalé problémy s detekcí. Důvodů může být hned několik. Může se jednat buď o nedokonalou syntézu použitého VSTi nástroje, nebo třeba o fakt, že rámec, který je analyzován, neobsahuje celou periodu hlavní frekvence, ale pouze její část. Pro správnou funkci detekce nízkých frekvencí je tedy třeba algoritmus dále optimalizovat. Na obrázku 7.9 jsou uvedeny výsledky z programu HResults.

```
===== HTK Results Analysis =====  
Date: Thu May 10 15:16:38 2012  
Ref : ./music/low/ref.mlf  
Rec : ./music/low/bass.mlf  
----- Overall Results -----  
WORD: %Corr=87.50, Acc=-262.50 [H=7, D=0, S=1, I=29, N=8]
```

Obrázek 7.10 – Výstup HResults pro test nízkých frekvencí – modul BooBass

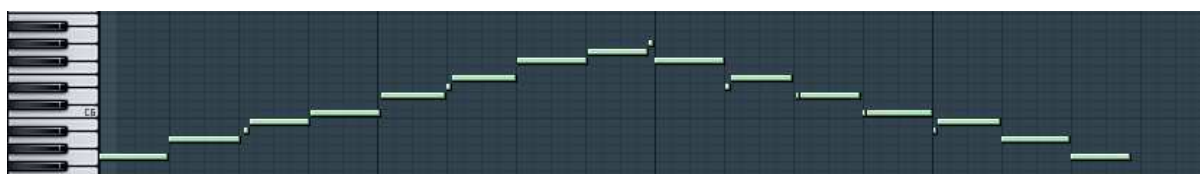
7.2.2 Vyhodnocení testů s reálnými nástroji

Během tohoto testování bylo vyzkoušeno několik hudebních nástrojů – elektrická kytara, pražcová baskytara, bezpražcová baskytara a housle. Na nástroje byla poté zahrána stupnice G-dur (viz obr. 7.11) v různých oktávách.



Obrázek 7.11 – stupnice G-dur

Dle očekávání byla největší přesnost dosažena během testů s houslemi. Úspěšnost rozpoznávání byla v tomto případě velmi vysoká.



Obrázek 7.12- Test reálných nástrojů - Housle

```
===== HTK Results Analysis =====  
Date: Thu May 10 17:02:32 2012  
Ref : ./music/real/violin/ref.mlf  
Rec : ./music/real/violin/test.mlf  
----- Overall Results -----  
WORD: %Corr=100.00, Acc=53.33 [H=15, D=0, S=0, I=7, N=15]
```

Obrázek 7.13- Výsledky z HResults - test reálných nástrojů - Housle

Analyzátor během hraní smyčcem bez použití pokročilejších technických prvků hry prakticky neměl problém s rozpoznáváním. V tomto případě by se tedy dal program bez větších problémů použít i k živému vystoupení. Problém nastával v momentě, kdy hráč použil vibrato a podobné pokročilé techniky. Výjimka byla u techniky zvané *pizzicato*¹⁶, kde opět rozpoznávání bylo téměř bezproblémové.

Druhým použitým nástrojem byla bezpražcová baskytara. I zde se dá mluvit o slušné kvalitě rozpoznávání, drobným problémem je však zjištění, že ne každá bezpražcová baskytara produkuje stejně kvalitní zvuk, co se týče konkrétnosti. Často se člověk může setkat s nástroji, kdy je rozpoznávání výšky tónu problém pro cvičené lidské ucho, natož pro programy tohoto typu. Pro co největší přiblížení k objektivním výsledkům byla tedy použita baskytara střední třídy¹⁷.



Obrázek 7.14 - Test reálných nástrojů – Bezpražcová baskytara

¹⁶ Technika hraní prsty, při které se ozývají pouze krátké tóny s výrazným atakem.

¹⁷ Jedná se o model baskytary *WB 4 OB* výrobce *Tune*. Cena tohoto modelu se pohybuje okolo 20 000, tudíž se dá mluvit o nástroji střední třídy.

```

===== HTK Results Analysis =====
Date: Thu May 10 17:05:53 2012
Ref : ./music/real/fretless/ref.mlf
Rec : ./music/real/fretless/test.mlf
----- Overall Results -----
WORD: %Corr=100.00, Acc=-53.33 [H=15, D=0, S=0, I=23, N=15]
=====

```

Obrázek 7.15 - Výsledky z HResults - test reálných nástrojů – Bezpražcová baskytara

Nutno podotknout, že při hraní vyšších tónů dosahoval analyzátor daleko větší přesnosti, než v nižších polohách, kde se občas stávalo, že aplikace detekovala správný tón, ale ve špatné oktávě.

Posledními dvěma nástroji byla pražcová baskytara a elektrická kytara. Výsledky testování u těchto dvou nástrojů byly velmi podobné, proto je lze sloučit do jedné kapitoly. Obecně u všech nástrojů se na začátku tónu vyskytují i další frekvence spojené s ostřejším atakem, u pražcových se ale tento problém projevuje nejvíce. Do frekvenčního spektra jsou často přimíchávány frekvence, které odpovídají drnčení struny o pražec, cinkání pražců apod. Toto způsobuje analyzátoru nemalé problémy.



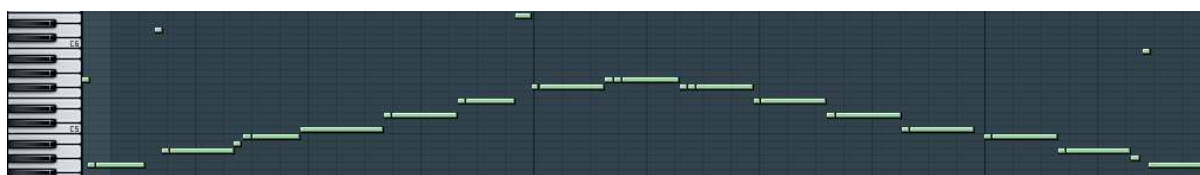
Obrázek 7.16 - Test reálných nástrojů – Baskytara

```

===== HTK Results Analysis =====
Date: Thu May 10 17:04:36 2012
Ref : ./music/real/bass/ref.mlf
Rec : ./music/real/bass/test.mlf
----- Overall Results -----
WORD: %Corr=100.00, Acc=-153.33 [H=15, D=0, S=0, I=38, N=15]
=====

```

Obrázek 7.17 - Výsledky z HResults - test reálných nástrojů – Bezpražcová baskytara



Obrázek 7.18 - Test reálných nástrojů – Elektrická kytara

```

===== HTK Results Analysis =====
Date: Thu May 10 17:03:44 2012
Ref : ./music/real/guitar/ref.mlf
Rec : ./music/real/guitar/test.mlf
----- Overall Results -----
WORD: %Corr=100.00, Acc=-40.00 [H=15, D=0, S=0, I=21, N=15]
=====

```

Obrázek 7.19 - Výsledky z HResults - test reálných nástrojů – Bezpražcová baskytara

Z výsledků je také patrné, že rozpoznávání tónů kytary bylo přesnější než rozpoznávání tónů baskytary. Důvodem je pravděpodobně fakt, že rozpoznávání vyšších frekvencí je díky jejich kratší periodě daleko snadnější a přesnější.

Podobně jako v předchozí sadě testů, i zde se chyba vložení objevovala nejčastěji. Přestože hodnocení přesnosti vyhodnocené programem HResults nebylo příliš dobré, dle obrázků s grafy jednotlivých hraných not lze posoudit rozpoznávání jako poměrně úspěšné. Chyby se objevovaly především na začátku jednotlivých not, další rozpoznávání již bylo téměř bezproblémové. Co se týče výsledného zvuku po zpětné syntéze, zmíněné chyby sejevily jako krátké přeskočení not na začátku některých tónů, což je daleko přijatelnější, než kdyby analyzátor detekoval chyby v průběhu tónů.

7.2.3 Uživatelské testy

K testování a hodnocení bylo pozváno několik uživatelů, každý z nich s určitou zkušeností z hudební branže. Při výběru lidí pro testování produktu byl kladen důraz především na různorodé žánrové zaměření hudebníků, širší věkové rozpětí a v neposlední řadě také různou míru zkušeností s hudbou. Do testování byl také zařazen člověk, který má zkušenosti s hudbou především po komerční stránce, resp. prodejce hudebních nástrojů a producent. Po krátkém čase, kdy měli tito lidé možnost si analyzátor vyzkoušet, byli požádáni o sepsání krátkého hodnocení, kde měli zhodnotit především použitelnost, ovládání a funkčnost aplikace.

Téměř všichni účastníci testování se shodli na faktu, že v hudebním průmyslu takové věci pořád ještě scházejí, jedná se tedy o produkt s velkým potenciálem. *„Jsem rád, že se konečně někdo odváží vstoupit do těchto relativně neznámých vod. Je to přesně to, co na našem trhu, i mezi muzikanty, chybí...“ Stanislav Ferko.* Ovládání, resp. GUI programu hodnotili všichni účastníci jako intuitivní a přehledné, velmi kladně byla hodnocena možnost posílání MIDI dat do externího syntetizéru.

Funkčnost byla hodnocena jako průměrná. Muzikanti často během testování naráželi na nepřesnosti při použití pokročilých herních technik, u baskytaristů se jednalo především o slap, u houslisty se jednalo o vibrato. Navzdory očekávání, že největším problémem bude fakt, že produkt si nedokáže poradit s polyfonií, byla často zmiňována absence rozpoznávání dynamiky tónu. *„Jedinou věc, kterou bych asi jako muzikant požadoval navíc, je nějaká dynamika... Je docela škoda, že tak důležitá složka zůstala ošizena...“ Karel Pech*

Z dostupných hodnocení také vyplývá fakt, že většina muzikantů by produkt používala především pro sólové motivy, doprovodné (akordové) hraní zůstává až na druhé koleji. Veškerá hodnocení včetně jmen a stručných profilů hodnotitelů jsou k dispozici v příloze B. Všichni uvedení hodnotitelé dali svolení k publikaci jejich hodnocení včetně jejich jména.

8 Závěr

Cílem této práce bylo vytvořit určitý druh softwarového analyzátoru/syntetizéru. Dle hodnocení lidí, kteří se účastnili testování lze usoudit, že aplikace splnila svůj účel, i když s drobnými nedostatky. Během testování se také ukázalo, že program není jen další z řady zbytečných nástrojů pro úpravu zvuku, ale dalo by se mluvit o produktu s velkým komerčním potenciálem. Samozřejmě před uvedením produktu na trh by byly další úpravy a optimalizace.

V rámci rozšíření bakalářské práce by bylo, po určitých testech, vhodné dodělat analýzu dynamiky tónu, stejně tak by bylo vhodné provést určité optimalizace algoritmu pro eliminaci chyb na začátku tónů.

Pokud by na tomto projektu pracoval větší tým lidí a byl by dostatek času, určitě by bylo vhodné modifikovat aplikaci tak, aby byla přenositelná minimálně mezi systémy Mac OS a Microsoft Windows. Zajímavé by bylo i vyzkoušet kombinaci několika metod pro rozpoznávání, implementovat je a na základě jejich výsledků aproximovat cílovou hodnotu tak, aby se co nejvíc blížila realitě.

Literatura

- [1] BEAUCHAMP, James W. *Analysis, synthesis, and perception of musical sounds: the sound of music*. Editor James W. Beuchamp. New York: Springer Science Business Media, 2007, 325 s. ISBN 978-0387-32496-8.
- [2] GOLD, Bernard, Nelson MORGAN a Dan ELLIS. *Speech and audio signal processing: processing and perception of speech and music*. 2nd ed. Hoboken, N.J.: Wiley, c2011, 661 s. ISBN 978-047-0195-369.
- [3] BROWN, Judith C. Musical fundamental frequency tracking using a pattern recognition method. *The Journal of the Acoustical Society of America*. 1992, roč. 92, č. 3, s. 1394-. ISSN 00014966. DOI: 10.1121/1.403933.
Dostupné z: <http://link.aip.org/link/JASMAN/v92/i3/p1394/s1>
- [4] JIRSÁK, Martin. Seri(ál)ová MIDI komunikace. *MUZIKUS: Pro muzikanty* [online]. 2010 [cit. 2012-05-08].
Dostupné z: <http://www.muzikus.cz/tagy/serialova-midi-komunikace/>
- [5] MAREŠOVÁ, Marcela a Ondřej MAŇÁK. *PROJEKT DO PŘEDMĚTU GZN: I05 - Software pro převod WAV to MIDI*. 2011 [cit. 2012-05-08].
- [6] MIDI. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-05-08].
Dostupné z: <http://en.wikipedia.org/wiki/MIDI>
- [7] Virtual Studio Technology. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-05-09].
Dostupné z: http://en.wikipedia.org/wiki/Virtual_Studio_Technology
- [8] Audio Stream Input/Output. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-05-09].
Dostupné z: http://en.wikipedia.org/wiki/Audio_Stream_Input/Output
- [9] MIDI Messages. [online], [cit. 2010-05-17]
Dostupné z <http://www.midi.org/techspecs/midimessages.php>
- [10] *Steinberg: 3rd party developer area* [online]. 2012 [cit. 2012-05-10].
Dostupné z: <http://www.steinberg.net/en/company/developer.html>
- [11] *WiDiSoft: Products* [online]. 2011 [cit. 2012-05-10].
Dostupné z: <http://www.widisoft.com/english/products.html>
- [12] *Digital Ear: Information* [online]. 1999, 2012-05-05 [cit. 2012-05-10].
Dostupné z: <http://www.digital-ear.com/digital-ear/wav-to-midi.asp>

- [13] *IntelliScore: Product info* [online]. 1999 [cit. 2012-05-10].
Dostupné z: <http://www.intelliscore.net/product.html>
- [14] *Solo Explorer: Way to MIDI conversion software* [online]. 2002 [cit. 2012-05-10].
Dostupné z: <http://www.recognisoft.com/>
- [15] MIDI Reference. MICROSOFT. *Windows Desktop Development* [online]. 2012 [cit. 2012-05-10]. Dostupné z:
msdn.microsoft.com/en-us/library/windows/desktop/dd757277%28v=vs.85%29.aspx
- [16] *VST Plug-Ins SDK Documentation* [online]. 2003 [cit. 2012-05-10].
Dostupné z: <http://www.gersic.com/vstsdk/>

Příloha A

Příklad testovacích souborů

Original	Analyzed
#!MLF!#	#!MLF!#
"data/a.lab"	"data/a.rec"
c5	c5
e5	c5
g5	e5
c6	g5
d6	c6
h5	d6
g#5	d6
g5	h5
c6	h5
a#5	g#5
g#5	g5
g5	c6
d#5	c6
d#6	a#5
h5	a#5
c6	g#5
	g#5
	g5
	g5
	d#5
	d#5
	d#6
	h5
	h5
	c6

Obrázek 9.1 – Příklad testovacích souborů pro HResults

Příloha B

Uživatelské hodnocení aplikace

Stanislav Ferko – 33 let, hudební skladatel, autor dvou muzikálů, hráč na kytaru

„Jsem rád, že se konečně někdo odvážil vstoupit do těchto relativně neznámých vod. Je to přesně to, co na našem trhu, i mezi muzikanty, chybí. Chápu, že analýza zvuku a zjišťování frekvence je asi složitý proces, ale určitě nic není nezvládnutelné, když se v dnešním světě stále otevírají nové a nové možnosti. Proto si velmi vážím toho, že se konečně někdo z muzikantů do toho pustil.

Ohledně ovládání programu, musím říct, že je to přesně podle mých představ, je to jednoduché, snadno pochopitelné, pěkně to vypadá. Také velmi oceňuji univerzalitu použití, ovládat kytaru klávesy je určitě nejen zábavné, ale i atraktivní pro diváky.

Celkově program sice funguje dobře, ale přece jenom se najdou nějaké mouchy. Kdyby se to ještě trochu doladilo, myslím si, že z toho může být velmi slušný produkt, který by našel uplatnění jak v živém hraní, tak ve studiu. Také je docela škoda, že aplikace nezvládá více tónů najednou.“

Karel Pech – 49 let, hráč na basketu

„Když jsem slyšel o tomto projektu poprvé, říkal jsem si, že to bude zase něco, co se stane tradiční šuplíkovou záležitostí. Nikdy jsem nebyl moc nakloněn věcem, které vyžadují více než nástroj, kabel a zesilovač. Když jsem tedy dostal pozvánku k nějakému testování, mé pocity byly spíše negativní. Nabídku jsem ale přijal, a nelituju toho.

Přestože funkcionálna a kvalita rozpoznávání není nijak závratná, je to rozhodně zajímavá práce, u které by bylo škoda, kdyby zůstala ležet někde pod záplavou papírů. Zrovna tohle je věc, u které se můžete nejen náramně pobavit, ale taky pokud na podium vylezete s basketu, která hraje jako flétna, musí to diváky zaujmout. A o to v muzice jde.

Jedinou věc, kterou bych asi jako muzikant požadoval navíc, je nějaká dynamika... Je docela škoda, že tak důležitá složka zůstala ošizená. Všiml jsem si taky, že spousta kolegů se ptala na polyfonii, bylo by to sice fajn, ale pro mě jako basketaristu to zas tak důležité není.“

Nikola Dus – 23 let, skladatel moderní hudby, hráč na kytaru a kontrabas

„Audio Analyzer si sympatie svého uživatele získá hned při prvním spuštění. Příjemné grafické zpracování je adekvátní zaměřením produktu, příliš nápadná vizuální stránka by byla spíše ke škodě. Barva je zvolena vhodně, ovládání se uživateli neztrácí na monitoru, zvláště uvážíme-li programy, se kterými bude analyzátor pravděpodobně spuštěn. Přesto by byla, v rámci budoucích verzí, pěkným zpestřením možnost měnit barevné kombinace.

Z hlediska náročnějších uživatelů se ovládání jeví jako příliš zjednodušené, první setkání však může být matoucí. Po několika hodinách práce se zvuky běžný (a snad i náročnější) uživatel shledá, že více než ony dvě ovládací možnosti nejsou příliš zapotřebí, a to i s ohledem na možnosti běžně používaných programů. Ovládání je tedy intuitivní a plně pochopitelné. Spuštění nečiní potíže a uživatel nemusí čelit prodlevám.

Samotná práce se zvukem je uspokojující, navíc, uvážíme-li šíři zvukových knihoven, otevírají se netušené možnosti, pro nahrávání, ale i hraní naživo, v tomto ohledu by bylo na místě eliminovat šum, zpestřit dynamiku a zkrátit odezvu.

Jedná se o velmi ambiciózní produkt, jehož absence, především v posledních letech, minimálně mezi muzikanty, byla smutným faktem. Nezbyvá než vřele doporučit!“

Marek Dufek – 24 let, student jazzového oddělení LK v Ostravě, hráč na baskytaru a kontrabas

„Přiznám se, nejsem moc velký fanoušek digitálního zpracování zvuku, zvláště ne pro živé hraní. Přesvědčit mě, abych používal produkt takového typu, to je nelehký úkol. Na druhou stranu, musím říct, že se určitě jedná o program, který má svůj potenciál, zvláště u hudebníků technicky zběhlejších, než jsem já.

Ovládání je, dalo by se říci, intuitivní. Nic nechybí, nic nepřebývá, přesně tak by to mělo být. Zpracování... no, tady to stále ještě pokulhává, ale věřím tomu, že kdyby na to byl čas a lidi, tak to může fungovat docela dobře. Největší problém vidím asi v rozpoznávání dynamiky.

Myslím si, že pokud by programátoři dokázali takový produkt pořádně doladit a nacpat jej do malé krabičky, aby člověk s sebou nemusel tahat počítač a další věci, určitě by to přispělo k prodejnosti programu a možná i já bych uvažoval o koupi takového efektu.“

Příloha C

Návod k použití aplikace

Aplikace je určena především pro operační systémy MS Windows XP a vyšší. Pro bezproblémový překlad je nutné nejdříve vytvořit *lib* soubor knihovny FFTW. Tento krok se dá provést pomocí aplikace *lib*, která je součástí MS Visual Studia 2010. Soubory potřebné k vytvoření zmíněného souboru jsou umístěny na CD v adresáři *libfft*. Projekt byl vytvořen v prostředí MS Visual Studia 2010 Professional, pro bezproblémový překlad je tedy doporučeno právě toto vývojové prostředí. Soubor s projektem se nachází v adresáři se zdrojovými kódy na CD.

Aby bylo možné používat aplikaci, je nutné mít nainstalován nějaký software, který může fungovat jako VST Host, dokáže přijímat MIDI data a podporuje rovněž zpravování více stop. Doporučenými programy jsou např. Cubase, FL Studio, Cakewalk Sonar nebo Ableton Live. Případně lze použít jednostopý program s posláním do externího syntetizéru. Vzhledem k tomu, že aplikace byla vytvořena především pro analýzu zvuku hudebních nástrojů, je více než vhodné mít k dispozici hudební nástroj se zvukovým výstupem.

V rámci práce je přiložen volně stažitelný program LoopBe1. Jedná se o aplikaci, která v počítači vytvoří virtuální MIDI rozhraní, které se dá použít pro přeměrování MIDI dat zpátky na vstup hostitelské aplikace.

Před spuštěním samotného programu je nutné zkopírovat knihovnu *libfft3-3.dll* z CD do adresáře, kde se nachází spustitelný soubor zvolené hostující aplikace. V opačném případě by produkt této práce nešel načíst, systém by hlásil chybějící knihovnu.

Pro správné chování aplikace je postup následující. Spustit hostitelskou aplikaci, na vstup jedné zvukové stopy připojit výstup hudebního nástroje. Poté je nutné importovat VST plugin AAnalyzer jako efekt na výstup zmíněné stopy. Dále je nutné vytvořit MIDI stopu, její výstup poslat do jakéhokoli VSTi. Pozor! Výstup zmíněného VST nástroje nesmí směřovat na stejný kanál jako výstup stopy s hudebním nástrojem. Posledním krokem je zvolení příslušného MIDI zařízení v Uživatelském rozhraní importovaného VST efektu AAnalyzer. Nyní stačí už jen povolit zachycování MIDI dat ze zvoleného rozhraní v hostitelské aplikaci.

V případě posílání dat do externího zařízení je postup mírně odlišný. Opět je nutné spustit hostitelskou aplikaci. V této aplikaci propojit vstup jakékoliv zvukové stopy s výstupem hudebního nástroje. Na tuto stopu potom nasadit efekt AAnalyzer a v něm zvolit příslušné MIDI rozhraní, kam posílat zprávy.

Aplikaci lze rovněž použít pro analýzu již nahraných zvukových souborů. Pro tento typ použití je nutné vložit na vstup zvukové stopy hostitelské aplikace, místo hudebního nástroje, soubor, který má být analyzován.

Příloha D

Obsah přiloženého CD

Na CD se nacházejí následující adresáře:

- *bin* - v tomto adresáře se nachází výsledná aplikace (dynamická knihovna)
- *src* - zde se nacházejí veškeré zdrojové texty aplikace.
 - *project/AAalyzer/win* - adresář se souborem projektu pro MS Visual Studio 2010 (*.sln*)
- *libfft* - v tomto adresáři se nachází použitá knihovna *libfftw3-3* spolu s definičním souborem
- *loopbe* - adresář obsahuje instalační soubor programu pro vytvoření virtuálního MIDI rozhraní
- *pdf* - zde je umístěn text této práce ve formátu *pdf*
- *text* - adresář obsahuje text bakalářské práce ve formátu *docx*
- *matlab* - adresář obsahuje textový soubor s algoritmem pro matlab
- *tests* - adresář obsahuje soubory pro HResults, na základě kterých byly prováděny testy
 - *readme.txt* - soubor s vysvětlením jednotlivých adresářů a názvů souborů