

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

ROZPOZNANIE TÓNOV HRY NA KLAVÍRI
BAKALÁRSKA PRÁCA

2020
ŠIMON BABÁL

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

ROZPOZNANIE TÓNOV HRY NA KLAVÍRI
BAKALÁRSKA PRÁCA

Študijný program: Aplikovaná informatika
Študijný odbor: 2511 Aplikovaná informatika
Školiace pracovisko: Katedra aplikovanej informatiky
Školiteľ: RNDr. Paula Štancelová

Bratislava, 2020
Šimon Babál



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Šimon Babál
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Rozpoznanie tónov hry na klavíri
Piano tones recognition

Anotácia:

1. Prehľad metód pre efektívnu detekciu stlačených kláves pri hre na klavíri v kombinácii s použitím rôznych existujúcich a dostupných technológií.
2. Analýza možností detekcie v reálnom čase.
3. Rozpoznanie tónov hry na základe detekcie stlačených kláves.
4. Export rozpoznaných tónov do štandardizovaných formátov MIDI/MusicXML.

Vedúci: RNDr. Paula Štancelová
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.
Dátum zadania: 03.10.2019

Dátum schválenia: 07.10.2019

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Pod'akovanie: Tu môžete pod'akovať školiteľovi, prípadne ďalším osobám, ktoré vám s prácou nejako pomohli, poradili, poskytli dáta a podobne.

Abstrakt

Slovenský abstrakt v rozsahu 100-500 slov, jeden odstavec. Abstrakt stručne sumarizuje výsledky práce. Mal by byť pochopiteľný pre bežného informatika. Nemal by teda využívať skratky, termíny alebo označenie zavedené v práci, okrem tých, ktoré sú všeobecne známe.

Kľúčové slová: automatická hudobná transkripcia,

Abstract

Abstract in the English language (translation of the abstract in the Slovak language).

Keywords:

Obsah

Úvod	1
1 Prehľad problematiky	3
1.1 Teória	3
1.1.1 Klavír	3
1.1.2 Tón	3
1.1.3 Homofónna hudba	3
1.1.4 Polyfónna hudba	4
1.1.5 Hudobná transkripcia	4
1.1.6 Diskrétna konvolúcia	5
1.1.7 Segmentácia obrazu	5
1.2 Možné riešenia problematiky	8
1.2.1 Na základe spracovávanía zvuku	8
1.2.2 Na základe spracovávanía obrazu	8
1.3 Technológie	12
1.3.1 Python	12
1.3.2 OpenCV	12
1.3.3 MIDI	12
1.3.4 MusicXML	12
1.3.5 Knižnica music21	14

Zoznam obrázkov

1.1	Pozícia klaviatúry na klavíri	4
1.2	Príklad výpočtu konvolúcie	5
1.3	Príklad segmentácie	6
1.4	Prewittov filter	7
1.5	Sobelov filter	7
1.6	Laplacian	7
1.7	Pozícia kamery pri natáčaní videa	9
1.8	Rozdiel medzi snímkami	9
1.9	Porovnanie úspešnosti pri rýchlejšej a pomalšej skladbe	10
1.10	Grafická vizualizácia detekcie stlačenia kláves	11
1.11	Tabuľka úspešnosti ClaVision	11
1.12	Názvy nôt, MIDI čísla a frekvencie	13

Zoznam tabuliek

Úvod

Cieľom tejto bakalárskej práce je tvorba softvéru pre automatickú hudobnú transkripciu hry na klavíri do formátu MIDI pomocou počítačového videnia s použitím programovacieho jazyka Python a knižnice OpenCV. Hlavná časť tejto práce bude venovaná popisom algoritmov na rozpoznávanie klaviatúry v rámci klavíra a na detekciu stlačených kláves.

Výsledná aplikácia bude umožňovať vložiť video a ako výstup z nej bude súbor v MIDI alebo MusicXML formáte, v ktorom bude zachytené, ako boli na videu stláčané klávesy na klavíri. Aplikácia po vložení videa najskôr deteguje, v ktorej časti obrazu sa nachádza klaviatúra, následne v nej segmentuje jednotlivé klávesy. Po tejto segmentácii bude hľadať rozdiely medzi jednotlivými snímkami a z týchto rozdielov bude aplikácia vedieť určiť, kedy bol nejaký kláves stlačený a kedy pustený.

Vo východiskovej kapitole budú vysvetlené základné pojmy, ktoré sa budú týkať tejto práce. Budú to pojmy z oblasti hudby a spracovania obrazu. V ďalšej časti tejto kapitoly bude prehľad možných riešení a podobných prác, ktoré rozoberali problém automatickej hudobnej transkripcie hry na klavíry. Na konci tejto kapitoly budú bližšie vysvetlené technológie, ktoré budú využívané v praktickej časti.

Kapitola 1

Prehľad problematiky

1.1 Teória

1.1.1 Klavír

Klavír je strunový hudobný nástroj, ktorého zvuk vzniká šírením vlnenia vyvolaného kmitaním strún. Klavírové struny sa rozkmitajú pomocou kladiviek, ktoré sú ovládané klávesmi. Klávesy sa nachádzajú na klaviatúre, a pozícia klaviatúry je znázornená na obrázku 1.1. Štandardný klavír obsahuje 88 kláves, pričom 52 z nich je bielych a 36 čiernych.

Okrem štandardných klavírov existujú však aj špeciálne koncertné alebo elektrické, pri ktorých sa môžu vyššie popísané vlastnosti mierne líšiť.

1.1.2 Tón

Pojem tón je ľubovoľné počuteľné vlnenie, ktoré bolo spôsobené rozkmitaním klavírových strún pomocou kladiviek. Pri elektrických klavíroch je tón generovaný digitálne a vlnenie sa šíri z reproduktora, keďže elektrický klavír nemá struny. Výška tónu je daná frekvenciou kmitania klavírových strún a mala by byť relatívne konštantná. Tón je základný prvok hudby. Počuteľné frekvencie pre ľudské ucho sú od 16 Hz až 20 kHz.

1.1.3 Homofónna hudba

Homofónna hudba je taká, pri ktorej súčasne hrá najviac jeden tón v jednom momente. Hovoríme jej taktiež jednohlasná. Hudobné nástroje, pre ktoré je homofónna hudba najtypickejšia sú väčšinou dychové, napríklad trubka, hoboje, trombón a rôzne druhy flaut. Homofónnu hudbu je možné vytvoriť pomocou ktoréhokoľvek hudobného nástroja. Pre tento typ hudby sa robí hudobná transkripcia najjednoduchšie, keďže obsahuje najmenej dát, ktoré je potrebné spracovať.



Obr. 1.1: Pozícia klaviatúry na klavíri

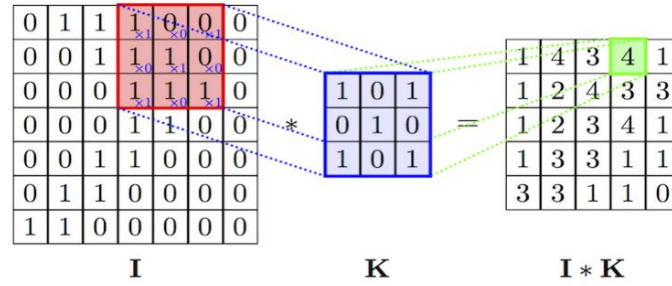
1.1.4 Polyfónna hudba

V tejto práci budeme polyfónnu hudbu definovať ako niekoľko poskladaných homofónnych hudieb a zvykne sa jej hovoriť viachlas. Polyfónna hudbu dokážu tvoriť zložitejšie hudobné nástroje, ako je napríklad klavír, harfa, organ a gitara. Pre tento typ hudby sa robí hudobná transkripcia zložito, pretože obsahuje veľké množstvo dát, ktoré treba spracovávať.

1.1.5 Hudobná transkripcia

Hudobná transkripcia je prepis záznamu znejúcej hudby do inej podoby. V tejto práci sa budeme venovať digitálnej transkripcii, pri ktorej máme 3 možnosti, ako ju je možné zrealizovať. Prvou a najviac používanou je transkripcia v reálnom čase hry na digitálnom klavíri cez rozhranie MIDI, pričom klavír komunikuje s počítačom cez kábel alebo Bluetooth. Pri tomto type transkripcie klavír posiela informácie o stlačených tónoch úplne priamo vo formáte MIDI a počítač ich môže následne uložiť do súboru, alebo vytvoriť notový záznam stlačených tónov. Ďalšou možnosťou je vykonať transkripciu využitím audio záznamu. Táto možnosť je však veľmi zložitá pri polyfonickej hudbe, pretože je veľmi ťažké oddeliť jednotlivé frekvencie tónov v tomto type záznamu. Poslednou možnosťou je transkripcia s použitím obrazového záznamu ako zdroja informácií, ktorej bude primárne venovaná táto práca. Pri tejto transkripcii ako vstup bude video súbor a ako výstup hudobnej transkripcie bude MIDI súbor, ktorý bude obsahovať dáta, ako zahrať skladbu, ktorá bola na danom videu.

Všetky vyššie popísané možnosti je však možné kombinovať a rovnako dnes existujú riešenia s využitím neurónových sietí. Všetke tieto riešenia sú bližšie popísané v ďalšej časti tejto kapitoly 1.2.



Obr. 1.2: Príklad výpočtu konvolúcie [7]. Pre skrátenej zápis operácie sa používa znak "hviezdička".

1.1.6 Diskrétna konvolúcia

Diskrétna konvolúcia je veľmi často využívaná na predspracovanie obrazu a môžeme ju nasledovne definovať podľa [9], kde $f(x, y)$ je funkcia obrazu a $h(x - m, y - n)$ je konvolučná maska:

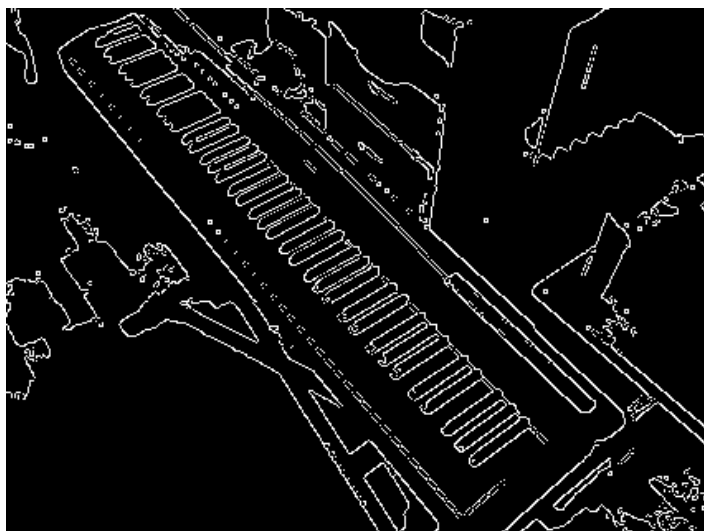
$$g(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n)$$

Operácie diskkrétnej konvolúcie sa počítajú podľa vyššie popísanej definície a dnes máme na tieto výpočty k dispozícii vysoko optimalizované implementácie, napríklad pre Intel procesory v knižnici Intel Integrated Performance Primitives (IPP), ktorú používajú Matlab, OpenCV a ďalšie softvérové nástroje s určením pre spracovanie obrazu [9].

Konvolučnú masku vždy definujeme tak, aby jej veľkosť bola nepárna, napríklad 3×3 alebo 5×5 . Robíme to z dôvodu, aby bola jednoznačná stredová pozícia, na ktorú budeme zapisovať výslednú hodnotu [9]. Tieto masky vieme využívať na vyhladzovanie, odstraňovanie šumu, potlačanie alebo zvýraznenie hrán. Pokiaľ nie je našim úmyslom obraz zosvetliť, alebo stmaviť, maska by mala byť navrhnutá tak, aby bol súčet všetkých hodnôt konvolučného jadra rovný jednej.

1.1.7 Segmentácia obrazu

Segmentácia obrazu je proces jeho transformácie, ktorého cieľom je rozdeliť obraz na objekty alebo oblasti a ďalšie informácie o nej budeme definovať podľa [4]. Metódy segmentácie môžeme zhrnúť do 3 kategórií, ktorými sú: prahovanie, segmentácia založená na hranách a segmentácia založená na oblastiach, pričom prvé dve budú viac objasnené v nasledovných odstavcoch.



Obr. 1.3: Príklad segmentácie: na obrázok bola aplikovaná cannyho detekcia hrán a následne otsuov algoritmus na optimálne prahovanie

Prahovanie

Prahovanie je najjednoduchšia a výpočtovo najmenej náročná, najrýchlejšia technika na segmentáciu. Používa sa pri nej jasová konštanta, ktorý nazývame prah. Prahovaním transformujeme obraz takým spôsobom, že prechádzame cez každý pixel obrazu. Ak je jasová hodnota pixlu menšia než prah, hodnotu jasu daného pixlu zmeníme na 0. V opačnom prípade zmeníme hodnotu jasu daného pixlu na maximálnu hodnotu. Toto sa taktiež nazýva binarizácia obrazu a môžeme ju vidieť na obrázku 1.3. Existuje však aj niekoľko modifikácií prahovania, ako napríklad čiastočné prahovanie, pri ktorom časť obrázku ostane bez zmeny a prahujú sa len pixle, ktorých hodnota jasu spadá do daného intervalu) alebo s viacerými prahmi. Prah môžeme určiť manuálne, alebo použiť na nájdenie prahu jeden z nasledovných prístupov:

1. Na základe údajov z histogramu. V tomto prípade sa jedná o globálne, keďže na celý obraz aplikujeme ten istý prah. V bimodálnych histogramoch sa zvykne vyberať prah ako minimum medzi dvoma najvyššími lokálnymi maximami.
2. Na základe metódy podielu, čiže ak vieme, akú časť obrazu pokrýva objekt, tak vyberáme prah takým spôsobom, aby objekt pokrýval dané percento obrazu a pozadie zvyšok.
3. Na základe optimálneho prahovania, ktoré určuje prah ako najbližšiu šedú úroveň zodpovedajúcu minimu pravdepodobnosti medzi maximami dvoch alebo viacerých normálnych rozdelení. Takéto prahovanie vedie k minimálnej chybe pri segmentácii.

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

(a) Horizontálna maska (b) Vertikálna maska

Obr. 1.4: Prewittov filter

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

(a) Horizontálna maska (b) Vertikálna maska

Obr. 1.5: Sobelov filter

Segmentácia založená na hranách

Segmentácia založená na hranách pracuje s lokálnymi hranami, ktoré boli detegované hranovými operátormi. Lokálne hrany sa nachádzajú na tých miestach obrazu, na ktorých dochádza k ostrému prechodu v úrovni šedej, farbe, textúre, alebo v inej vlastnosti. Najväčšími problémami v tejto metóde je detekcia falošných hrán alebo nedetegovanie niektorých hrán.

V tejto práci budeme hrany detegovať za pomoci ostrenia, ktoré budeme realizovať diskretnou konvolúciou. Keďže klavír obsahuje pravidelné hrany, ktoré zvyknú byť rovnobežné, budeme používať filtre slúžiace na vertikálne a horizontálne hrany. Podľa [9] si zadefinujeme niekoľko najznámejších konvolučných masiek na obrázkoch 1.4, 1.5 a 1.6.

Segmentačnou technikou využívajúcou detekciu hrán je taktiež houghova transfor-

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Obr. 1.6: Diskrétny Laplacian je definovaný ako suma druhej diferencie v danom pixeli vzhľadom na okolie pixelu. [9]

mácia. Používame ju vtedy, keď ideme detegovať objekty so známym tvarom hranice. Houghova transformácia dokáže detegovať hranice objektu, ak sú známe ich analytické vyjadrenia. Pôvodne bola houghova transformácia navrhnutá na detekciu rovných čiar a na tento účel bude využívaná aj v tejto práci.

Na segmentáciu sa okrem vyššie popísaných postupov používajú taktiež ucelené algoritmy, ktoré vykonávajú aj iné operácie okrem prahovania a ostrenia. Jeden z nich sa nazýva cannyho detektor hrán a bude využívaný v tejto práci. Tento algoritmus najskôr eliminuje šum a zvykne to robiť gaussovským filtrom. Následne aplikuje jeden z gradientných operátorov, najčastejšie sobelov filter. Po tomto kroku nasleduje stenčenie hrán. Táto funkcia vyberá z hodnôt gradientov len lokálne maximá. Posledným krokom tohto algoritmu je prahovanie, čiže binarizácia obrazu. Cannyho detektor hrán je v praxi veľmi užitočný, pretože je menej citlivý na šum na rozdiel od jednoduchších prístupov na detekciu hrán.

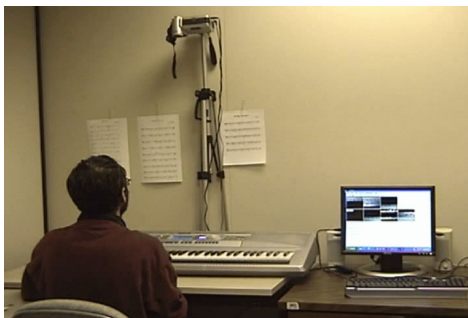
1.2 Možné riešenia problematiky

1.2.1 Na základe spracovávania zvuku

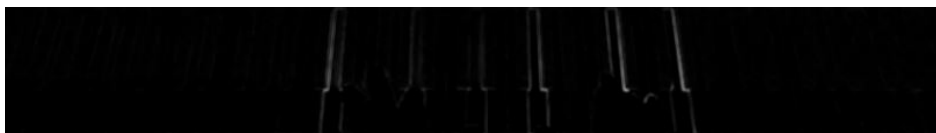
Automatická hudobná transkripcia hry na klavíry pomocou spracovania zvuku je predmetom skúmania približne od roku 1975 a pri homofónnej hudbe môžeme považovať tento problém za vyriešený v dnešnej. Komplikovanejšie je to však pri polyfónnej hudbe, ktorej transkripcia sa začala skúmať od roku 1990. Pri tomto type hudby je potrebné oddeliť jednotlivé homofónne zložky hudby, čo je mimoriadne zložité. Najkomplikovanejšie je oddeliť tóny, ktoré sa nachádzajú tesne pri sebe, pretože ťažko sa určuje, kde jeden tón končí a ďalší začína. Najrozsiahlejšiu analýzu v tejto oblasti spravil Klapuri v roku 2004 [5], podľa ktorého je jednotlivé tóny možné oddeliť viacnásobnými odhadmi F_0 z audio krivky, pričom F_0 je najnižšia vlnová dĺžka a najnižšia frekvencia v periodickom priebehu. Problém je však v tom, že týmto odhadovaním vieme detegovať iba výšku tónu, ale jeho časová dĺžka zostane neznáma. Na zistenie trvania tónu je preto potrebné analyzovať rytmus danej hudby, čiže nájsť pravidelnosti v striedaní dĺžok jednotlivých tónov. Tieto techniky však ani zďaleka nie sú dokonalé a postačujúce na dokonalú transkripciu, pretože nezaručujú správne fungovanie pri rôznych šumoch a zmene rytmu v hudbe.

1.2.2 Na základe spracovávania obrazu

Na hudobnú transkripciu hry na klavíri pomocou spracovania obrazu bez informácie o hĺbke je zameraná táto práca, čiže tento spôsob riešenia bude prezentovaný detailnejšie na rozdiel od zvyšných. Doteraz sa tejto problematike venovali jeden vedecký



Obr. 1.7: Pozícia kamery pri natáčaní videa



Obr. 1.8: Rozdiel medzi snímkou, na ktorej nie je stlačený žiaden kláves a snímkou, na ktorej je stlačený A-durový akord, po odstránení prstov zo snímok.

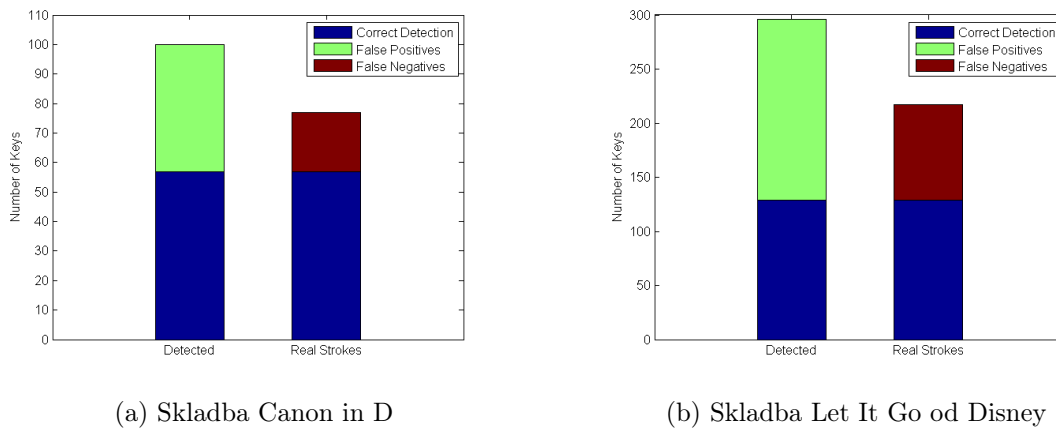
článok [10] jedna diplomová práca [3] a jedna dizertačná práca [6].

V každej z týchto prác sa riešia rovnaké problémy. Prvým problémom je extrahovanie klaviatúry z jednotlivých snímok a detekcia bielych a čiernych kláves. Ďalším krokom je odstránenie rúk hrajúceho na klavíri pomocou detekcie farby pokožky. Nasledujúcou úlohou je samotná detekcia stlačených kláves, čiže zisťovanie rozdielov medzi jednotlivými snímkami a počiatočnou snímkou pozadia, na ktorej nie je stlačený žiaden kláves. Tento rozdiel môžeme vidieť na obrázku 1.8. Posledným problémom je riešenie výstupu transkripcie, ktorý je v týchto prácach riešený buď výstupom v notovej reprezentácii alebo vo formáte MIDI. Vo všetkých týchto prácach je spracovávané video, ktoré bolo natáčané nad klavírom, čiže ideálne je, aby bol vektor kamery približne kolmý na klaviatúru, ako je znázornené na obrázku 1.7.

Detection of Piano Keys Pressed in Video

Ako prvý sa venoval automatickej hudobnej transkripcii hry na klavíri pomocou počítačového videnia Suteparuk v roku 2013 [10]. V tejto práci autor vytvoril program v programovacom jazyku Matlab, ktorý nebol síce určený na transkripciu, ale iba detegoval stlačenie jednotlivých kláves. Túto detekciu program používateľovi ukázal grafickým zvýraznením kláves, ktoré boli stlačené.

Program najskôr predspracuje video. V tomto predspracovaní ako prvé zvýrazní hrany sobelovým gradientným operátorom, ktorý bol autorom vybraný z dôvodu najrýchlejšej detekcie hrán s porovnaním s inými technikami. Pre ešte lepšie zvýraznenie hrán sa následne aplikuje prahovanie. Na extrahovanie klaviatúry bola použitá houghova transformácia.



Obr. 1.9: Porovnanie úspešnosti pri rýchlejšej a pomalšej skladbe

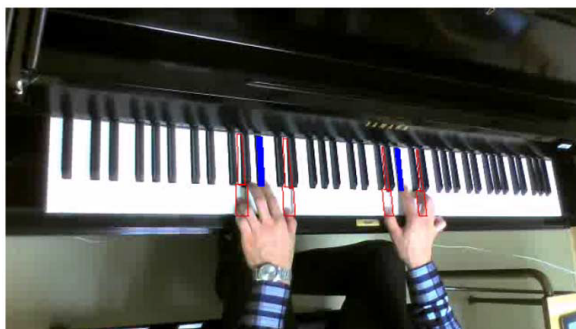
Najkomplikovanejšia časť detekcie je odstránenie rúk z klaviatúry. V tejto práci bola táto problematika riešená naučením sa farby pokožky na 10 snímkoch z videa. Následne bola aplikovaná maska, ktorá ruky z jednotlivých snímkov odstránila.

V programe sa následne detegujú čierne a biele klávesy, pričom čierne sa detegujú ako prvé, keďže je to jednoduchšie. V tomto kroku sa označilo, kde sa nachádzajú čierne klávesy, pričom bolo potrebné myslieť na okraje jednotlivých kláves, ktoré by mohli neskôr skresľovať detekciu kláves. Následne sa detegujú biele klávesy.

Detekcia stlačených kláves prebieha na základe zisťovania rozdielov medzi jednotlivými snímkami a snímkou pozadia. Autor uvažoval aj prípad, v ktorom by sa s kamerou mierne hýbalo počas nahrávania videa, preto nepoužíval len konštantnú snímku pozadia. Namiesto konštantnej snímky pozadia využíval V práci sú zverejnené výsledky z detekcie na dvoch videách. Obe boli malí snímkovú frekvenciu 30 snímkov za sekundu. Výsledky z testovania sú graficky znázornené na obrázku 1.9. Prvá skladba mala pomalšie tempo a mala názov Canon in D a druhá s rýchlejším tempom bola Let It Go od Disney. Pri prvom videu bolo správne detegovaných 74% stlačených a pri druhom to bolo 59.4%. Algoritmus však detegoval aj falošné stlačenia, čiže stlačenia, ku ktorým v skutočnosti nedošlo a pravdepodobne boli spôsobené šumom a tieňmi jednotlivých kláves. To znížilo jeho úspešnosť pri prvom videu na 63.3% a pri druhom na 43.6%. Podľa vyjadrenia autora [10] bolo možné o niečo zvýšiť úspešnosť detekcie využitím adaptívneho prahovania.

CLAVISION: VISUAL AUTOMATIC PIANO MUSIC TRANSCRIPTION

Ako ďalší sa tejto problematike venoval Akbari diplomovou prácou v roku 2015 [3]. Akbari vytvoril softvér v programovacom jazyku C-Sharp, ktorý robil automatickú hudobnú transkripciu hry na klavír v reálnom čase, pričom ako vstup mal video s minimálnym rozlíšením 320×240 pixelov a 24 snímkov za sekundu. Ako výstup poskytoval



Obr. 1.10: Grafická vizualizácia detekcie stlačenia kláves, pričom modrou farbou sú čierne klávesy a červenou farbou sú biele klávesy.

Video	Angle	Keys Detection %	Recall %	Precision %	F_1 Score
V9	0	100.0	74.6	85.4	0.796
V10	30	94.9	66.7	93.3	0.778
V11	45	80.3	63.0	100.0	0.773
V12	60	63.9	40.7	64.1	0.498
V13	75	38.6	15.7	28.6	0.165

Obr. 1.11: Tabuľka úspešnosti v závislosti od uhlu kamery, ktorá natáčala klavír.

MIDI alebo priamo notový záznam hudby. Popri automatickej hudobnej transkripcii tento softvér graficky zvýrazňoval stlačené klávesy, ako je znázornené na obrázku 1.10.

Akbari [3] použil na detekciu stlačených kláves veľmi podobný postup, ako Sutepa-ruk [10]. Hlavný rozdiel bol v tom, že na detekciu hrán použil cannyho metódu. Tá je výpočtovo náročnejšia, ale hrany dokáže detegovať spoľahlivejšie, čo sa ukázalo aj vo výsledkoch testovania a taktiež bola v tejto práci urobená analýza, ako sa líši úspešnosť detekcie klaviatúry a stlačených kláves v závislosti od uhlu kamery 1.11. Softvér mal dosahovať najpresnejšie výsledky, keď vektor pohľadu kamery a klaviatúra zvierali uhol 45 stupňov. Avšak pri tomto rozpoložení sa niektoré klávesy nepodarilo rozpoznať. Na druhej strane, klávesy boli najúspešnejšie rozpoznané, keď bol daný uhol 0 stupňov 1.7. Čiže najdokonalejšia presnosť by sa dala dosiahnuť použitím viacerých kamier, ktoré by klaviatúru snímali pod rôznymi uhlami [3].

Piano Music Transcription Based on Computer Vision

Najčerstvejšie bola tomuto problému venovaná dizertačná práca od Mccaffrey v roku 2017 [6]. Autor vytvoril softvér na automatickú hudobnú transkripciu hry na klavíri v programovacom jazyku C++ s použitím knižníc OpenCV na spracovanie obrazu a Midifile na vytváranie súborov vo formáte MIDI. Tento softvér mal priemernú úspešnosť detekcie stlačených kláves 93.57% a celkovú presnosť mal 78.72%. Algoritmus na detekciu stlačených kláves bol veľmi podobný algoritmu, ktorý používal Akbari [3], preto algoritmus dosahoval podobnú úspešnosť.

1.3 Technológie

1.3.1 Python

Python je interpretovaný a dynamicky typovaný programovací jazyk, ktorý vytvoril Guido van Rossum a jeho počiatky siahajú do roku 1990. Keďže je to interpretovaný jazyk, príkazy vykonáva priamo za sebou bez toho, aby sa celý program musel na začiatku skompilovať do strojového kódu. Tento jazyk je taktiež interaktívny, čiže program môžeme upravovať počas jeho behu. Python podporuje štruktúrované, funkcionálne a objektovo orientované programovanie. V tejto práci bude využívaný vo verzii Python 3.8.1.

1.3.2 OpenCV








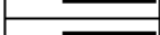




























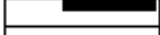





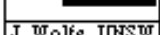
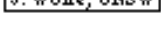








OpenCV (Open source computer vision) je knižnica s otvoreným zdrojovým kódom, ktorá obsahuje funkcie na spracovanie obrazu v reálnom čase a vznikla v roku 1999. Táto knižnica je napísaná v jazyku C++, ale je podporovaná aj v jazykoch Java a Python. V tejto práci bude táto knižnica využívaná vo verzii OpenCV 4.1.1.

1.3.3 MIDI

MIDI (Musical Instrument Digital Interface) je rozhranie, ktoré slúži na prenos a zaznamenávanie dát o hudbe a bolo štandardizované v roku 1983. Dáta o hudbe v MIDI formáte obsahujú informácie, ako bola nejaká melódia zahraná a následne je možné z MIDI záznamu danú melódiu opätovne zreprodukovat'. MIDI-protokol je v dnešnej dobe podporovaný vo väčšine elektrických hudobných nástrojoch a taktiež vo väčšine zvukových kariet. MIDI je taktiež podporované vo veľa softvéroch na prácu s hudbou, napríklad Sibelius alebo MuseScore. Na obrázku 1.12 je znázornený spôsob mapovania výšky jednotlivých klavírových tónov na reprezentáciu výšky tónov v MIDI formáte.

1.3.4 MusicXML

MusicXML je formát veľmi podobný MIDI s rozdielom, že je založený na XML (eXtensible Markup Language) a slúži na ukladanie informácií o postupe zahrania hudby. Je o niečo novší v porovnaní s MIDI a je taktiež podporovaný vo väčšine softvéru na prácu s hudbou.

MIDI number		Note name	Keyboard	Frequency	
21	22	A0		27.500	
23		B0		30.868	29.135
24	25	C1		32.703	
26		D1		36.708	34.648
28	27	E1		41.203	38.891
29		F1		43.654	
31	30	G1		48.999	46.249
33	32	A1		55.000	51.913
35	34	B1		61.735	58.270
36		C2		65.406	
38	37	D2		73.416	69.296
40	39	E2		82.407	77.782
41		F2		87.307	
43	42	G2		97.999	92.499
45	44	A2		110.00	103.83
47	46	B2		123.47	116.54
48		C3		130.81	
50	49	D3		146.83	138.59
52	51	E3		164.81	155.56
53		F3		174.61	
55	54	G3		196.00	185.00
57	56	A3		220.00	207.65
59	58	B3		246.94	233.08
60		C4		261.63	
62	61	D4		293.67	277.18
64	63	E4		329.63	311.13
65		F4		349.23	
67	66	G4		392.00	369.99
69	68	A4		440.00	415.30
71	70	B4		493.88	466.16
72		C5		523.25	
74	73	D5		587.33	554.37
76	75	E5		659.26	622.25
77		F5		698.46	
79	78	G5		783.99	739.99
81	80	A5		880.00	830.61
83	82	B5		987.77	932.33
84		C6		1046.5	
86	85	D6		1174.7	1108.7
88	87	E6		1318.5	1244.5
89		F6		1396.9	
91	90	G6		1568.0	1480.0
93	92	A6		1760.0	1661.2
95	94	B6		1975.5	1864.7
96		C7		2093.0	
98	97	D7		2349.3	2217.5
100	99	E7		2637.0	2489.0
101		F7		2793.0	
103	102	G7		3136.0	2960.0
105	104	A7		3520.0	3322.4
107	106	B7		3951.1	3729.3
108		C8		4186.0	

Obr. 1.12: Názvy nôt, MIDI čísla a frekvencie

1.3.5 Knižnica music21

Music21 je knižnica v jazyku Python, ktoré slúži na vytváranie a spracovávanie zvuku. Poskytuje funkcie na pomoc s prácou vo formátoch MIDI a MusicXML. Rovnako poskytuje vizualizáciu hudby pomocou knižnice Matplotlib.

Literatúra

- [1] Midi. https://nl.wikipedia.org/wiki/Musical_Instrument_Digital_Interface.
- [2] Yamaha gc 1 m pe grand piano. https://www.thomann.de/be/yamaha_gc1_stutzfluegel.htm.
- [3] Mohammad Akbari and Howard Cheng. Real-Time Piano Music Transcription Based on Computer Vision. *IEEE Transactions on Multimedia*, 17(12):2113–2121, 2015.
- [4] Milan Ftáčnik. Segmentácia obrazu. <http://sccg.sk/~ftacnik/IP-5.pdf>.
- [5] Anssi P. Klapuri. Automatic Music Transcription as We Know it Today. *Journal of New Music Research*, 33(3):269–282, 2004.
- [6] Robert McCaffrey. *Piano Music Transcription Based on Computer Vision*. PhD thesis, University of Dublin, Trinity College, 2017.
- [7] Ihab S. Mohamed. *Detection and Tracking of Pallets using a Laser Rangefinder and Machine Learning Techniques*. PhD thesis, 09 2017.
- [8] Mehmet Sezgin and Bülent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146 – 165, 2004.
- [9] Elena Sikudova, Zuzana Cernekova, Benešová Wanda, Zuzana Berger Haladova, and Julia Kucerova. *Počítačové videnie: Detekcia a rozpoznávanie objektov. - 1. vydanie*. Wikina, 2014.
- [10] Potcharapol Suteparuk. Detection of Piano Keys Pressed in Video. page 2006, 2013.