Luciano Rodrigues Lucio Neto

Desenvolvimento de um *drum pad* usando visão artificial

Brasil

24 de maio de 2019

Luciano Rodrigues Lucio Neto

Trabalho de Conclusão de Curso Submetido à Coordenação do Curso de Engenharia de Computação e Automação do Centro de Tecnologia da Universidade Federal do Rio Grande do Norte, como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Computação.

Universidade Federal do Rio Grande do Norte - UFRN

Coordenação do Curso de Engenharia de Computação e Automação - DCA

Graduação em Engenharia de Computação

Orientador: Agostinho de Medeiros Brito Júnior

Brasil

24 de maio de 2019

Resumo

Apresenta o desenvolvimento de um *drum pad* usando visão artificial capaz de controlar sintetizadores musicais criando uma sequência de notas musicais em repetição. Instrumentos assim são muito usados por músicos amadores que precisam criar acompanhamentos de bateria ou baixo para suas composições e não dispõem de músicos auxiliares para fazê-lo. A ferramenta criada permite que, usando apenas uma webcam, uma folha de papel e software livre, um músico amador seja capaz de criar efeitos semelhantes aos de um drum pad físico desenhando ou sobrepondo pequenas fichas coloridas na folha de papel.

Palavras-chaves: Drum pad, sequenciador, controlador midi, OpenCV, Visão artificial

Abstract

abstract in english

Keywords: translate-as

Lista de Figuras

Sumário

1 Introdução **1**

2 Materiais e Métodos **3**

2.1 Modelo para Interpretação 3

2.2 Algoritmo 3

2.3 Protocolo MIDI 4

2.4 Marcadores Aruco 6

2.5 Transformação de Perspectiva 7

3 Resultados **8**

4 Conclusões **9**

Capítulo 1

Introdução

Um *drum pad* é um periférico utilizados por diferentes segmentos da sociedade de diferentes formas. Sua principal função é reproduzir sons escolhidos pelo usuário ao apertar de botões de sua interface e, por isso, é comumente utilizado principalmente pela comunidade de músicos sendo eles profissionais ou amadores. Utilizado também como uma forma de se programar uma sequência de sons que será reproduzida em um intervalo de tempo específico, um *drum pad* serve como acompanhamento musical com diversas aplicações sejam elas para guiar um músico, acompanhar o ritmo com alguma percussão ou até reproduzir detalhes específicos que o usuário pode não conseguir no momento correto.

Apesar de ser um periférico de fácil utilização e de rápida adaptabilidade, ainda depende da capacidade do usuário em reproduzir as notas no tempo correto e principalmente da capacidade que o usuário tem em investir num equipamento do tipo, variando de cerca de 50 dólares, podendo chegar a mais de 600 dólares, dependendo do seu tamanho, sua funções, caracterísitcas, inovações e acabamento.

A proposta do projeto apresentado neste trabalho de conclusão de curso é de facilitar a programação de uma sequência de sons que um usuário venha querer utilizar para qualquer propósito que seja, principalmente os descritos anteriormente. Com apenas uma webcam de cerca de 60 dólares, uma folha de papel e um programa de computador que utiliza conceitos de visão artificial, o usuário é capaz de programar, de forma interativa, uma sequência de sons provenientes de um sintetizador midi, que será reproduzida em um tempo determinado sem depender, por exemplo, de suas próprias habilidades rítmicas, a partir de marcações feitas na

folha.

No capítulo XX serão apresentados..., no capítulo XX será mostrado ... . O capítulo XX discorre sobre.... blá, blá, blá...

Capítulo 2

Modelo para Interpretação

Como muitas aplicações que envolvam conceitos de visão artificial, é necessário um ambiente controlado de onde se possa extrair as informações necessárias para o funcionamento do programa de computador. Para isso, foi necessária a criação de um *layout* com elementos específicos que façam com que o ambiente seja bem interpretado pelo programa.

IMAGEM DO LAYOUT

As principais características do *layout* para que haja a interação com o programa de computador são os marcadores nos cantos da folha e o código de barras. Tendo em vista que, nesse modelo, sempre haverão 13 notas distintas que podem ser reproduzidas, o código de barras é utilizado para guardar a informação de quantas notas o programa pode reproduzir, sem haver a necessidade do usuário especificá-la já que não faria sentido o modelo ter, por exemplo, uma área quadriculada de 13x8 e o usuário configurar que o essa área tem uma distribuição diferente da real. Além do código de barras, são utilizados marcadores nos cantos da folha que, ao serem interpretados, mostram a imagem final que o programa utilizará como fonte de interpretação.

Capítulo 3

Algoritmo

Partindo da imagem capturada da câmera do usuário, o programa procura os marcadores dos cantos da folha, o código de barras e os interpreta guardando as informações necessárias. Essas verificações acontecem a cada frame capturado pela câmera para manter a integridade com a imagem de tempo real. Com pontos capturados a partir dos marcadores, é selecionada a área de interesse da folha que o programa deverá interpretar ao sinal do usuário.

IMAGEM DA ÁREA DO MEIO DA FOLHA

Com a informação de quantas colunas existem na área quadriculada, adquirida do código de barras presente na folha, a próxima etapa executada pelo algoritmo é de verficiar, em cada um dos retângulos da área de interesse, se há dois tipos de marcações diferentes: uma representa um son que só tocará naquele momento e a outra representa a continuidade do som, permitindo que ela seja reproduzido durante o tempo desejado pelo usuário.

IMAGEM DA MARCAÇÃO SOZINHA E IMAGEM DA MARCAÇÃO CONTINUADA (LADO A LADO)

A partir das informações que o programa consegue da etapa anterior, ele envia mensagens MIDI, utilizando o protocolo MIDI, para se comunicar com uma fonte de áudio selecionada pelo usuário, a qual será responsável pela reprodução dos sons em sí.

A ordem de disposição das notas foi escolhida da maneira como mostrada na imagem (REFERENCIA A IMAGEM) pois é o mais próximo de como uma "pista"MIDI é representada em softwares profissionais de gravação de músicas, denominados DAW (*Digital Audio Workstation*),quando o usuário utiliza recursos MIDI, como se pode ver na imagem abaixo:

IMAGEM DO REAPER DE UMA MIDI TRACK

Também podemos interpretar o layout como um gráfico num simples plano cartesiano. O grá-

fico representa frequência $\times$ $tempoedigitalediscreto$, $fazendocomqueaposi$ção$odamarca$ção$orepresenteumaf$

IMAGEM DO LAYOUT COM MARCAÇÕES E GRÁFICO LADO A LADO

Capítulo 4

Protocolo MIDI

O Protocolo MIDI é o principal responsável pela reprodução dos sons de sintetizadores e sequenciadores MIDI e, sendo um protocolo, é universal, ou seja, independe do fabricante do sintetizador ou sequenciador uma vez seguido de maneira correta. Apesar de bastante extenso e complexo, serão explicadas aqui apenas conceitos que foram utilizados no desenvolvimento do projeto.

Esse protocolo consiste na troca de "mensagens MIDI"entre o software ou hardware controlador, que controla os sons, e do software ou hardware sintetizador, que sintetiza os sons. É utilizado para transmissão de informação em tempo real entre controlador e sintetizador a fim de gerar sons quando o usuário desejar.

Essas mensagens baseam-se em um conjunto de um ou mais *bytes*, onde o primeiro *byte* se classifica como *STATUS byte*, o qual define o tipo da mensagem, e é geralmente seguido por outros *bytes* chamados *DATA bytes*, os quais dão as características desejadas para a mensagem.

Existem diversos tipos de mensagens, mostradas na tabela (REFERENCIA TABELA) e de forma expandida na tabela (REFERENCIA TABELA), extraídas diretamente da *The MIDI Association* (www.midi.org), principal portal com informações do protocolo MIDI, porém, para o desenvolvimento do projeto descrito nesta tese de conclusão de curso, foram utilizadas bascicamente as mensagens chamadas *Note ON* e *Note OFF*, que especificam quando uma nota deve ser ativada e desativada, além de mensagens padrão de configuração disponibilizadas pela biblioteca rtmidi que foi escolhida a fim de facilitar a comunicação do software com sintetizadores livres.

As mensagens de ativação e desativação do som são compostas por um conjunto de 3 *bytes* os quais são representados da seguinte forma:

- A mensagem *Note ON*:
  - *Status byte*: 1001 AAAA
  - *Data byte*: 0BBB BBBB
  - *Data byte*: 0CCC CCCC

Onde AAAA representa o canal que está mensagem controlará, variando entre 16 canais distintos, BBBBBBB representa o *pitch* que será reproduzido e CCCCCCC representa a velocidade que a nota será reproduzida que, em notação musical, temos essas velocidades da seguinte maneira:

IMAGEM DAS VELOCIDADES

Essa velocidade de reprodução, dependendo do tipo de instrumento selecionado, também varia a dinâmica que o som é reproduzido como quando uma nota de um piano é pressionada de forma brusca ou mais suave.

- A mensagem *Note OFF*:
  - *Status byte*: 1000 AAAA
  - *Data byte*: 0BBB BBBB
  - *Data byte*: 0DDD DDDD

Para as mensagens *Note OFF*, os *bits* AAAA e BBBBBBB representam o mesmo da mensagem de ativação da nota sendo diferente apenas no terceiro *byte*, o DDDDDDD, que representa a velocidade de *release* do som, ou seja, a suavização até ele ser desligado.

Capítulo 5

Marcadores ArUco

A funcionalidade do software projetado nesta tese depende intrinsecamente da AOI (*area of interest*) do *layout* e, para se adquirir essa área de interesse da imagem inicial, uma das soluções mais comuns é a utilização de marcadores fiduciais quadrados binários. O principal benefício em se utilizar marcadores do tipo é de se obter, a partir dos quatro cantos de um único marcador, a *camera pose*, ou seja, tanto a orientação quanto a posição da câmera no ambiente.

No desenvolvimento deste projeto foi escolhido o módulo Aruco da biblioteca de visão artificial OpenCv para geração e identificação desses marcadores.O módulo Aruco é baseado na biblioteca ArUco, uma popular biblioteca utilizada para detecção de marcadores fiduciais quadrados, desenvolvida por Rafael Muñoz e Sergio Garrido, em aplicações de realidade aumentada.

A partir do módulo Aruco, pode-se gerar marcadores ArUco que consistem em marcadores quadrados compostos por uma larga borda da cor preta e uma matriz binária intríseca ao marcador, que determina seu identificador. As bordas existem para facilitar a detecção do marcador em uma imagem e a codificação da matriz binária para permitir a detecção desse marcador com intuíto de serem aplicadas outras técnicas de visão artifial. Abaixo temos alguns exemplos de marcadores ArUco:

IMAGEM MARCADORES ARUCO

Com uma detecção rápida e precisa desses marcadores, feita pela biblioteca OpenCv, foram escolhidos quatro identificadores que simbolizam os cantos da folha modelo e o software, ao decodificar os quatro marcadores, dependendo do seu identificador, são selecionados 4 pontos, um de cada marcador, a fim de adquirir a área de interesse da folha para o software.

IMAGEM DOS QUATRO PONTOS IDENTIFICADOS

Com a área de interesse adquirida, é aplicada uma transformação de perspectiva nessa nova imagem a fim de desprezar distorções causadas pelo posicionamento da câmera, uma vez que esta nova imagem que será analizada pelo software precisa ser paralela à folha, o que é possível a partir de diferentes ângulos, isso se os quatro marcadores forem visíveis, graças a transformação de perspectiva, fazendo o usuário manusear o software de forma mais confortável.

Outra forma de identificação que foi testada no projeto foi por cores em círculos coloridos posicionados no canto da folha. Devido à diversas variáveis de ambiente, essas cores poderiam variar de usuário para usuário, local para local, câmera para câmera entre outros, o que tornava necessária a marcação das cores ser feita manualmente pelo usuário com o propósito de haver um cálculo da média da cores dos círculos para sua identificação e, devido a essa inconsistência ao utilizar cores, a ideia foi descartada. O modelo anterior pode ser visto abaixo:

IMAGEM DO MODELO COM OS CÍRCULOS

Assuntos a serem abordados:

Capítulo 6

Transformação de Perspectiva

- Descrever a MATEMÁTICA envolvida nessa transformação

- Descrever COMO o usuário deve usar o quadriculado para selecionar as notas a serem tocadas. A propósito, um pouquinho de teoria musical explicando um bê-a-bá do uso do modelo é bem vinda. :)))

- Descrever o uso da classe RtMidi, porque foi escolhida e como se dá sua utilização. Para a utilização do protocolo MIDI e para a comunicação entre sintetizadores, foi escolhida a biblioteca RtMidi para a linguagem de programação C++. Toda a transmissão e criação das mensagens foi abstraída durante o desenvolvimento do projeto graças a funcionalidades da biblioteca.

- Descrever o sintetizador usado nos experimentos (QSynth).

- Descrever como se dá, no Linux, a interligação entre seu software Controlador e o Sintetizador. Um diagrama legal feito no inkscape cairia bem nesse canto. O diagrama a seguir representa como o software controlador se comunica com um sintetizador DIAGRAMA (MIDI INPUT, OUPUT, SOFTWARE CONTROLADOR E SINTETIZADOR)

Capítulo 7

Resultados

Apresentar exemplos de uso da ferramenta...

Tem um software bem legal de composição chamado rosegarden. Ele também funciona como um "sintetizador m MIDI", pois aceita entradas do controlador para permitir composições. Prepare um loop de exemplo e conecte a saida do seu controlador na entrada do rosegarden. Observe a sequencia gerada no software. Salve a sequencia e verifique se é possível, usando o rosegarden, repetir a sequencia de loops conectando agora rosegarden->qsynth.

Se funcionar legal (acredito que funcionará sem problemas), terás um resultado muito bom, pois poderá mixar várias combinações possíveis usando o rosegarden.

Capítulo 8

Conclusões

Revise em linhas gerais o algoritmo desenvolvido e mostre os progressos que obteve, comentando resultados e dificuldades enfrentadas.

Proponha melhorias para a sua criação.