COLÉGIO TÉCNICO DE CAMPINAS DA UNICAMP
INFORMÁTICA

LEONARDO MASSUHIRO SATO RA: 20141
LUCAS COIMBRA DA SILVA RA: 20144

CMOV - Projeto de Captura de Movimento

Orientador: Sérgio Luiz Moral Marques
Co-orientador: Simone Pierini Facini Rocha

Campinas, SP
2022

LEONARDO MASSUHIRO SATO RA: 20141
LUCAS COIMBRA DA SILVA RA: 20144

CMOV - Projeto de Captura de Movimento

Relatório de Pesquisa Final do Trabalho de Conclusão de Curso de Informática Integrado ao Ensino Médio do Colégio Técnico de Campinas da Unicamp.
Orientador: Sérgio Luiz Moral Marques
Co-orientador: Simone Pierini Facini Rocha

Campinas, SP
2022

Agradecimentos

Agradeço meus amigos e familiares pelo apoio que proporcionam.
Agradeço ao Orientador Professor Sérgio Luiz Marques, pela ajuda passada durante o desenvolvimento do projeto.
Agradeço aos trabalhadores do Google, ao proporcionar uma boa documentação e diversos serviços gratuitamente.
Agradeço a Deus, por manter minha integridade em momentos de dificuldade e me ajudar a superá-las.

“O futuro dependerá daquilo que fazemos no presente.”- **Mahatma Gandhi**

RESUMO

Facilitar a captura de movimento pode ser de grande importância para a utilização da captura em outras áreas que não seja em animações. **Objetivo:** Facilitar a captura de movimento com a criação de um software que não necessita de equipamentos com preço elevado para a captura, mostrando de exemplo o seu uso em um robô. **Método:** Estudo dos dados retirados pela câmera traseira do celular, utilizando a API Vision de detecção de poses do ML Kit (*Machine Learning Kit*), disponibilizada gratuitamente pela Google. As imagens podem ser captadas separadamente (foto à foto, utilizando uma foto capturada na hora ou uma foto da galeria) ou em conjunto (em tempo real). Foram comparados os valores de posição em relação à foto presentes nos pontos da pose para o cálculo da angulação das juntas corporais, utilizando a lei dos cossenos para calcular e definir a angulação dos motores do robô Poppy Humanoid, disponibilizado pelo COTUCA para o projeto. **Resultado:** A precisão da API Vision de detecção de poses é acima de 99% em fotos com uma boa iluminação e sem obstruções que impedem que o corpo apareça por completo. A precisão é acima de 98% em fotos com obstrução parcial do corpo. A aplicação Poppy teve êxito em simular um robô virtual, porém fracassou em mover um Poppy real. **Conclusão:** Houve a incapacidade de mostrar a exemplificação do projeto utilizando o robô Poppy Humanoid devido às limitações de tempo, hardware e referências anteriores. A utilização de Aprendizagem de Máquina para a captura de movimento em imagens estáticas ou múltiplas imagens (tempo real) pode ser considerada como uma alternativa viável para facilitar e baratear a captura de movimento atual devido a sua precisão, mas ainda é necessário melhorias, como a captura completa dos movimentos dos dedos da mão.

Palavras-chaves: Captura de movimento, Inteligência Artificial, Smartphone, Poppy, Robô.

SUMÁRIO

INTRODUÇÃO

A captura de movimento é o processo de gravação de movimento e transposição do movimento para um modelo digital. É muito utilizada em animações 3D, para torná-la mais fluida e realista comparada com uma animação que não a utiliza.

Essa tecnologia é muito complexa e custosa para utilizá-la com eficiência em áreas que não sejam estúdios profissionais de captura de movimento. A tecnologia de mocap (Motion Capture) gira em torno de 250.000 a 900.000 dólares, contando apenas o software utilizado para processá-la. Os custos tendem a aumentar considerando a compra de roupas de mocap, câmeras profissionais, sensores, iluminação, planos verdes, o próprio estúdio, entre outros.

Isso dificulta o trabalho de animadores 3D, entusiastas em tecnologia, pesquisadores, programadores e outras pessoas que podem usufruir da captura de movimento para facilitar seu trabalho, já que parte desses grupos não são de empresas que podem arcar com os custos ou simplesmente trabalham sozinhas sem a disponibilidade de um complexo estúdio especializado na captura.

O projeto visa encontrar uma forma de simplificar a captura de movimento, com o objetivo principal de barateá-la para, não só facilitar o trabalho de animadores sem condições de estúdios caríssimos, como também mostrar que é possível a utilização da captura em outras áreas da tecnologia, se possível simplificá-la.

Uma das tecnologias que podem ser utilizadas para simplificar a captura seria o uso do kinect do console Xbox One, o qual tem um custo médio de R$500,00 e já é utilizado para capturar o movimento em jogos eletrônicos. Nesse caso, também seria necessário um computador compatível com o kinect, com uma média de custo de R$3.000,00 , além de um local com espaço suficiente para ter uma distância de 1 metro do sensores.

Outra tecnologia, a qual foi testada nesse projeto, seria o uso de Inteligências artificiais para capturar as poses de uma pessoa baseado em uma foto. O preço médio seria apenas de um celular relativamente recente (cerca de R$1.500,00), já que os dados da foto são processados pela própria API da Inteligência Artificial. Algumas tecnologias pesquisadas foram: APIs da Microsoft Azure de captura corporal (as quais há custos mensais) e um grupo de APIs da Google chamado ML Kit (*Machine Learning Kit*), que foi a escolhida por ser disponibilizada gratuitamente, como uma boa documentação e com maior facilidade para a

sua aplicação em celulares. Essas tecnologias, além do celular, necessitam de ao menos 1 metro de distância da câmera e um local bem iluminado.

Recentemente, a Google disponibilizou gratuitamente em seu Kit de Aprendizado de Máquina uma API para detecção de poses. Oferecida em versão Beta, essa tecnologia foi aplicada em nosso projeto devido a sua facilidade de aplicação, e por ser feita especialmente para aplicações móveis.

METODOLOGIA

O projeto se dividiu em etapas semestrais, primeiro semestre consistiu em uma etapa de pesquisa sobre fontes e referências que seriam utilizadas no projeto, já a segunda consistiu na realização prática do projeto. O processo foi documentado como um todo pela ferramenta do diário de bordo, relatando erros, hipóteses, práticas desenvolvidas, ideias e rotinas. Além disso, na realização do projeto, foi utilizado o recurso do plano de pesquisa formal, feito nos padrões da ABNT.

Para a captura de movimento, foi aplicado a API Vision de detecção de poses do ML Kit (*Machine Learning Kit*) em uma aplicação em Dart, utilizando o framework Flutter. A aplicação consiste em duas funcionalidades:

- Captura de poses de uma foto estática (da galeria ou uma foto capturada pelo usuário);
- Captura de poses em tempo real (utilizando obrigatoriamente a câmera frontal).

Na captura de poses de uma foto estática, foi utilizado o plugin *ImagePicker*, responsável por capturar imagens e inserir imagens da galeria na aplicação.

Figura 1
Captura de posse de uma imagem da galeria, feita pelo aplicativo.

Na captura em tempo real, no primeiro teste foi utilizado um código simples o qual utilizava o plugin *Camera* para controlar a câmera e tirar fotos com o método padrão de captura de foto. Porém, utilizando esse método, havia uma demora de cerca de 5s para a captura de cada foto, sendo devagar demais para a proposta de "captura em tempo real".

Assim sendo, foi pesquisado outras formas de capturar imagens para utilizar na API. A solução mais encontrada durante a pesquisa foi usar um ImageStream (função do *Camera* responsável por lidar com o frame atual da câmera, ou então, a imagem atual) e utilizar a imagem gerada por ele na API. Mas a dificuldade encontrada nessa solução seria converter a imagem capturada pelo ImageStream para um formato que a API do ML Kit aceite (no caso de um celular android, por exemplo, seria converter imagem codificada em yuv420 para JPEG).

Para converter a imagem, seria necessário converter o binário da imagem para um binário JPEG, porém fazer isso utilizando Dart também afeta de forma negativa o processamento. Assim, na primeira solução, seria utilizado a biblioteca ffi, responsável por rodar nativamente código em C na aplicação flutter, apenas para converter o binário de yuv420 para JPEG (no caso de um celular android).

A primeira solução não funcionou de forma satisfatória, já que antes da finalização da primeira solução foi encontrada outra solução, a qual é mais simples que converter uma imagem utilizando código em C. Pesquisando mais à fundo, foi encontrado em uma thread do GitHub (já que a documentação do ML Kit *pose detector* não está completa por estar em beta) uma forma de passar o binário da imagem capturada pelo ImageStream diretamente para a API, mesmo que a codificação da imagem seja diferente de yuv420 (como é o caso de smartphones da marca Xiaomi e Apple).

Com a aplicação dessa solução, houve uma melhora que reduziu o tempo de processamento da imagem em 4 segundos. Atualmente, as únicas coisas que afetam no processamento seria a demora de resposta do servidor da Google, a qual há um delay médio de 0.3s segundos entre a captura da foto e a captura da pose em uma internet de 250 mega.

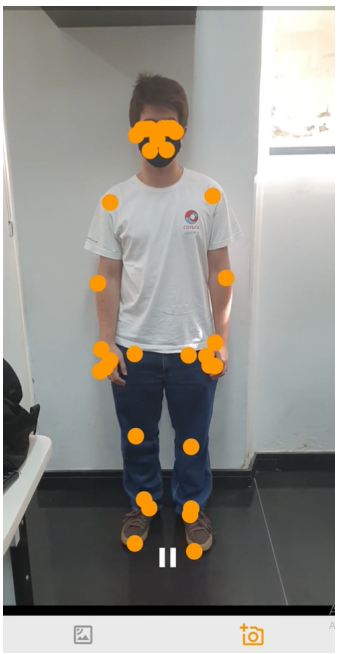


Figura 2
Captura de movimento em tempo real, utilizando a solução de processamento citada anteriormente.

O plugin do ML Kit responsável pela captura da pose em uma foto retorna uma lista de poses, a qual cada pose contém os valores de nome do ponto e *x, y* e *z* do ponto em relação a foto, como apresentado na figura 3. São 33 pontos capturados em cada imagem passada para a API.

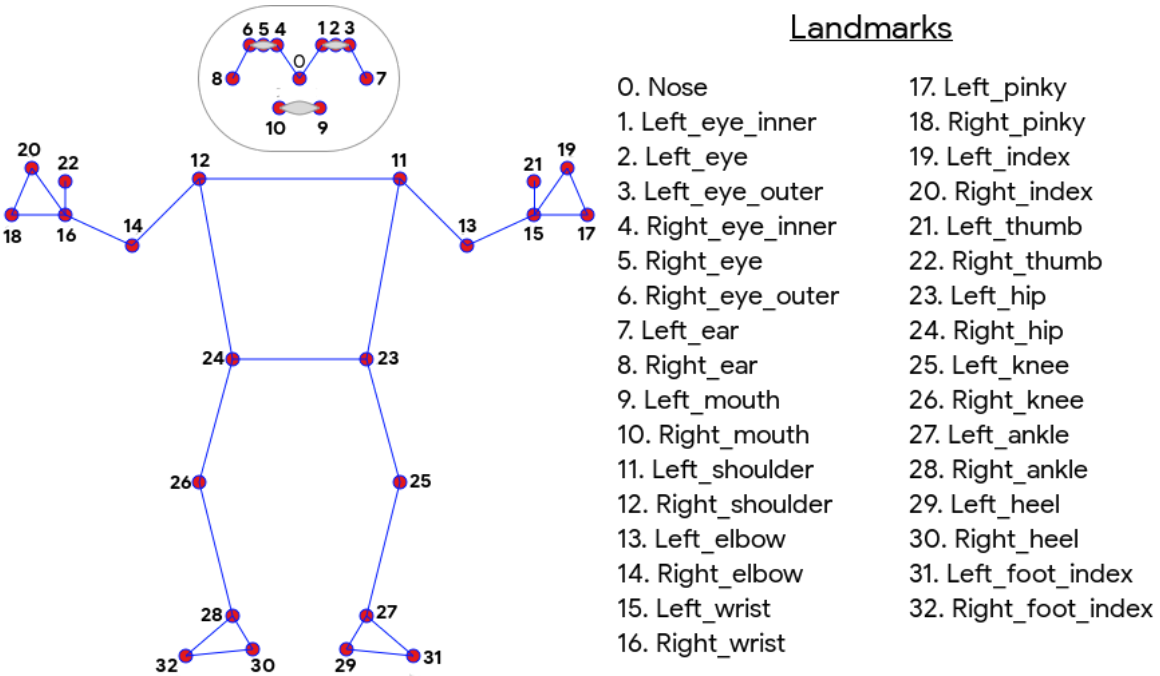


Figura 3
DOCUMENTAÇÃO ML KIT. Detecção de poses. Disponível em: https://developers.google.com/ml-kit/vision/pose-detection. Acesso em: 23 nov. 2022.

A API também apresenta um sistema de previsão de pose no caso de parte do corpo da pessoa na foto estar obstruído por outro objeto. Para evitar problemas nessa previsão, há a existência de uma variável chamada “likelihood”, a qual varia de 0 à 1, definindo a precisão do ponto de cada parte do corpo, sendo possível ignorar uma variável no código caso esse valor não atenda uma precisão mínima.

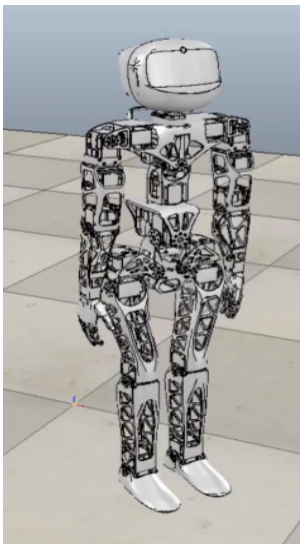
De acordo com essa mesma variável, a precisão da captura de movimento sem a obstrução do corpo por algum objeto e em um local bem iluminado é acima de 99%. Já, a precisão é acima de 98% com a obstrução parcial do corpo. A aplicação não consegue capturar as poses de um corpo com obstrução significativa (obstrução de mais de 60% do corpo). A aplicação consegue prever o rosto de um corpo de costas para a câmera com uma precisão acima de 98% de acordo com os testes (claro que a precisão é menor caso o objetivo da captura seja expressões faciais, já que não é possível prevê-las quando uma pessoa está de costas). Infelizmente a variável z, nos testes, tende a ser um pouco imprecisa comparado com a profundidade real, como dito pela própria documentação.

Para a transferência dos dados da captura de movimento para o robô, foi utilizado como backend o banco de dados NoSQL *Realtime Database* do Firebase (pertencente à Google). Foi escolhido esse backend pela facilidade de passar os dados do celular para o robô, sem a necessidade de lidar com servidores, portas, ou outros tipos de conexão, apenas a internet (necessária para o ML Kit também).

No backend, apenas os dados necessários para os cálculos do robô estarão disponíveis. No caso, esses dados são os pontos dos ombros esquerdo e direito, os pontos dos cotovelos direito e esquerdo e os pontos dos pulsos esquerdo e direito (6 pontos). Apenas esses são necessários já que apenas os braços do robô iriam se mover. Para definir como o robô deve se mover, seria usado lei dos cossenos para definir o ângulo entre 3 pontos, e usar isso como base da angulação do motor (ignorando pontos com likelihood abaixo de 98%, ou então, 0.98).

Figura 4
Banco de dados depois da execução da captura de movimento.

Para a realização da aplicação de expressão de movimento, foi utilizado um Poppy-Humanoid real e um Poppy-Humanoid e Poppy-Torso virtual. Para a demonstração dos resultados da aplicação virtual, houve a implementação de um código feito para ser executado na plataforma Jupyter dentro do Visual Studio Code. Entretanto, na aplicação com o robô real, houve uma implementação diferente, utilizando vias padrões de se executar código em python.



Figuras 5 e 6
5- Foto do robô Poppy-Humanoid utilizado no projeto.
6 - Foto do robô Poppy-Humanoid utilizado na simulação virtual.

RESULTADOS

Em relação a captura de movimento e transferência dos dados para um backend, os objetivos se deram por completo. A captura do movimento, quando em um ambiente bem iluminado e a pelo menos meio metro de distância da câmera, sempre teve uma precisão acima de 98% nos pontos da posse.

A eficiência se mostrou possível de melhora com o melhor envolvimento por parte da Google para a melhora do ML Kit, mas 0,3 segundo é um valor muito razoável para uma captura de movimento feita pelo celular . Para o banco de dados, a demora é de 1s para chegar os dados. Porém, retirando o delay dos servidores, não houve nenhum problema de eficiência no celular no resultado final.

Devido a eficiência da captura de movimento pelo celular com a utilização de tecnologias de IA, é plausível que um celular consiga ser um substituto para os complexos equipamentos de captura de movimento de estudios, desde que seja para ações mais simples (como a captura de movimento do braço), já que a API não se mostrou muito eficiente em capturas mais precisas (como a captura dos dedos).

Não foi possível fazer testes com o Kinect do Xbox One para comparar os dados com os resultados do ML Kit, já que a documentação é confusa, os programas para programar com o Kinect estão descontinuados e houve falta de tempo por parte do grupo para focar em um projeto para o Kinect.

Em relação ao robô, a aplicação virtual que simulava o robô Poppy foi completa, porque a partir dela era possível a movimentação dos membros, ou seja, a expressão do movimento de forma eficiente. Entretanto, já o programa do robô real, não teve o mesmo grau de eficácia, porque não houve a real expressão do movimento nessa aplicação.

O resultado do projeto CMOV se deu por completo na área de captura de movimento via ML Kit, porém as demais áreas demonstraram incompletas devido a fatores como escassez de tempo, falta de referência, entre outros.

CONCLUSÃO

Com a finalização da aplicação de captura de movimento utilizando Flutter e o ML Kit, apesar da não finalização da conexão com o robô, fica claro a capacidade do celular fazer a captura de movimento de uma pessoa, de forma a facilitar o processo de captura de movimento e diminuir drasticamente o preço dessa tecnologia.

A utilização de Inteligência Artificial para baratear a tecnologia de Captura de Movimento é plausível e sujeita a ser, no futuro, uma alternativa viável comparada a todos os equipamentos atuais necessários para fazer a captura. Porém, ainda é necessário melhorias para essas IAs chegarem próximas do nível de precisão de um equipamento profissional, como a captura completa dos movimentos dos dedos da mão.

REFERÊNCIAS

DOCUMENTAÇÃO ML KIT. Detecção de poses. Disponível em: https://developers.google.com/ml-kit/vision/pose-detection. Acesso em: 23 nov. 2022.

DOCUMENTAÇÃO DO FIREBASE. Realtime Database. Disponível em: https://firebase.google.com/docs/database/. Acesso em 23 nov. 2022.

DOCUMENTAÇÃO DA BIBLIOTECA POPPY-HUMANOID E POPPY TORSO. Expressão do movimento. Disponível em: https://docs.poppy-project.org/en/. Acesso em 23 de nov. 2022.

DOCUMENTAÇÃO DYNAMIXEL SDK. Software dos motores servo. Disponível em: https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel_sdk/overview/. Acesso em 23 nov. 2022