



**POSICIONAMENTO E EXIBIÇÃO DE IMAGENS 3D UTILIZANDO
ÓCULOS DE REALIDADE AUMENTADA PARA APLICAÇÃO
CIRÚRGICA**

Relatório final na modalidade de auxílio à iniciação científica, submetido à
Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

Processo: 2020/15835-4

Pesquisador Responsável: Dr. Glauco Augusto de Paula Caurin

Coorientador: Paulo Henrique Polegato

Beneficiário: Calvin Suzuki de Camargo

Informações gerais do projeto

- Título do projeto:

Posicionamento e exibição de imagens 3D utilizando óculos de realidade aumentada para aplicação cirúrgica

- Pesquisador responsável:

Dr. Glauco Augusto de Paula Caurin

- Coorientador:

Paulo Henrique Polegato

- Beneficiário:

Calvin Suzuki de Camargo

- Número do processo do projeto:

2020/15835-4

- Instituição sede do projeto:

Escola de Engenharia de São Carlos da Universidade de São Paulo

- Período de vigência proposto:

01 de março de 2021 a 28 de fevereiro de 2022

- Período coberto por este relatório científico:

01 de março de 2021 a 28 de fevereiro de 2022

Resumo

Trata-se do relatório final do projeto de pesquisa de iniciação científica que compreende os trabalhos iniciados em março de 2021 até o final de fevereiro de 2022. O projeto visa estudar a exibição de objetos 3D em óculos inteligentes (*Smart glasses*) com realidade aumentada, que podem servir como um dispositivo auxiliar para aplicações cirúrgicas. Para isso, pretende-se conceituar as relações entre visão computacional e computação gráfica no campo da realidade aumentada. Com o apoio do Laboratório Aeronáutico de Tecnologias (AeroTech), estudaremos uma forma de elaborar uma aplicação que sobreponha a cabeça de um paciente com um modelo 3D indicando pontos de implantação de eletrodos durante o procedimento neurocirúrgico orientado por estereoeletroencefalografia (*SEEG*).

Palavras-chaves: *Smart glasses*, Aplicação cirúrgica, Visão computacional, Realidade aumentada.

Sumário

Informações gerais do projeto	i
Resumo	ii
1 Resumo do projeto	1
1.1 Objetivo	1
1.2 Metodologia	1
1.3 Histórico do projeto	1
2 Realizações	3
2.1 Estudos em desenvolvimento Android Studio	3
2.2 Estudos de desenvolvimento Unity	5
2.3 Arquitetura do sistema	7
Referências bibliográficas	9

1 Resumo do projeto

1.1 Objetivo

Pretende-se exibir informações e posicionar modelos tridimensionais em uma região do espaço com *AR*, de forma que facilite o acesso do cirurgião à informação durante a cirurgia; estudar e registrar a resposta dos equipamentos utilizados no quesito de qualidade gráfica e latência de resposta do sistema. Tudo isso, com o objetivo central de aumentar a proximidade do cirurgião com a tecnologia de *AR* como apoio durante os procedimentos cirúrgicos.

1.2 Metodologia

Consiste na listagem de possíveis soluções, técnicas ou ferramentas; o estudo e a discussão sobre elas, em seguida, sua implementação. Paralelamente a isso, a busca bibliográfica é constantemente realizada com o objetivo de esclarecer dúvidas sobre os meios imaginados e discutidos com o orientador e coorientador. Essa busca dá ênfase nos resultados encontrados pelos artigos, o objetivo disto é caracterizar os prós e contras das diversas opções encontradas na listagem de técnicas e soluções. Essa pesquisa de artigos desenvolve um discernimento que é refletido em uma noção de funcionamento dos métodos, impactando muito na escolha da implementação para o projeto de pesquisa.

1.3 Histórico do projeto

Desde o início dos trabalhos no projeto, foram experimentados diversos tipos de contato com a elaboração de *softwares* para o sistema operacional *Android* (figura 1.1a); testes das ferramentas da documentação dos óculos de realidade aumentada *SEIKO EPSON Moverio BT-350* (figura 1.1b e 1.1c); e a elaboração de aplicativos que ilustram o objetivo do *VCranium* (figura

1.1d). Assim, como foi explicado no relatório parcial da pesquisa, pretendíamos prosseguir o desenvolvimento estabelecendo uma arquitetura composta por computador, câmera (*webcam*) e óculos para capturar os dados necessários para a projeção em realidade aumentada, mais detalhes serão descritos no capítulo das realizações.



(a) Primeiro aplicativo AR para *Android*



(b) Tentativa de calibração do *Moverio*



(c) Aplicativo de exemplo fornecido pela *EPSON*



(d) Primeira versão do *VCranium*

Figura 1.1: Histórico de realizações da pesquisa até a primeira entrega parcial. Fonte: Autor.

2 Realizações

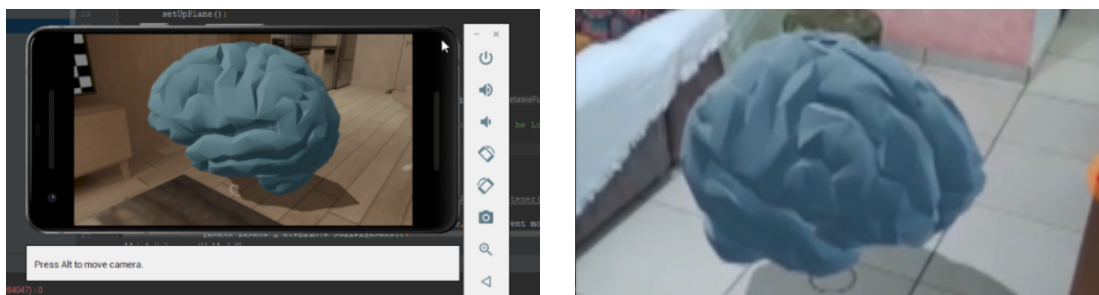
2.1 Estudos em desenvolvimento Android Studio

O desenvolvimento de aplicativos para *Android* foi estudado nos primeiros meses da pesquisa. Nessa parte do projeto, a pesquisa teve um aspecto mais técnico, que foi necessário para a programação das futuras aplicações que estariam por vir. A *IDE* (ambiente de desenvolvimento integrado) utilizada foi o *Android Studio*, que foi sugerida pelo curso adquirido da *Udemy* (UDEMY..., 2021).

Após as primeiras semanas de aprendizado *Android*, foi iniciado uma pesquisa sobre o desenvolvimento de aplicativos voltados a realidade aumentada. Porém, antes de irmos diretamente nisso, um interessante teste foi realizado com *OpenCV*, que consistiu em realizar as primeiras manipulações das imagens da câmera (ABOUT..., s.d.). A experiência foi proveitosa para o aprendizado da integração de ferramentas externas ao projeto padrão da plataforma e que, ademais, será feito muitas vezes até a conclusão da pesquisa.

A primeira biblioteca de realidade aumentada a ser testada foi o *Google Sceneform* (SCENEFORM..., 2021). Muitos problemas foram encontrados na integração dos *plugins* com o *Android Studio*, pois este precisava estar em uma versão antiga específica para funcionar. Após a instalação, foi possível ver a primeira projeção em realidade aumentada em um simulador de *smartphone* no computador (figura 2.1a). Restava testar o aplicativo para um *smartphone* real, porém o afastamento dos integrantes do laboratório pela pandemia, e a falta de dispositivos compatíveis à minha disposição, causaram um atraso nos testes. Felizmente, foi gerado um arquivo que permite a instalação à distância e o aplicativo funcionou com sucesso nos celulares da equipe (figura 2.1b).

Prosseguindo os estudos de aplicativos AR, percebemos que os problemas de versão tidos com o *Sceneform* estavam sendo reparados pelo *Google* adaptados em um novo programa: *Google ARCore Services* (GOOGLE..., 2021). Sua proposta é que uma biblioteca seja instalada no



(a) Simulação da projeção AR no *Android Studio* (b) *Smartphone* com o aplicativo funcionando

Figura 2.1: Resultados adquiridos com o *Google Sceneform*. Fonte: Autor.

celular para que os aplicativos tenham acesso, trazendo a vantagem da redução do tamanho das aplicações produzidas. No entanto, somente uma lista restrita de *smartphones* modernos podem instalar essa biblioteca, a justificativa dos desenvolvedores é a compatibilidade com o sistema (ARCORE..., 2021).

Seguido disso, durante a pesquisa foi encontrado uma nova ideia *open-source*, também do *Google*, chamado *Mediapipe*, que tinha a proposta de entregar muitas ferramentas de visão computacional com ML (*machine learning*) integrado (MEDIAPIPE..., 2022). O projeto foi iniciado em julho de 2019, e tem ganhado mais popularidade por ser gratuito, multi-plataforma e facilitar muito a aplicação de ML para *Android*, *iOS*, e PC. No momento da descoberta, não era conhecido a compatibilidade do *Mediapipe* com *API* antigas de *Android* e muito menos o seu comportamento em óculos de realidade aumentada, por isso, a ideia foi reservada.

Um sumário das opções testadas com breves comentários abaixo:

Google Sceneform É um *plugin* para *Android Studio* que introduz muitas ferramentas para o desenvolvimento de aplicativos AR. Funciona somente em *Android* com a *API* 24 ou superior. Seu projeto foi arquivado em meados de 2020.

Google ARCore Services Pode se considerar o sucessor das ideias do *Sceneform*. Ele fornece uma biblioteca de ferramentas sofisticadas para AR e atualizações frequentes. Funciona somente em uma lista estrita de *smartphones* modernos.

Mediapipe Fornece uma grande quantidade de ferramentas baseadas em *Machine Learning* para visão computacional. Projeto iniciado em junho de 2019 e ganhando mais força recentemente. Pela recente criação, não é conhecido seu comportamento em *API* antigas (28 ou anterior).

O que podemos concluir do desenvolvimento de aplicações em realidade aumentada com *Android Studio* é que mesmo existindo ferramentas robustas de criação de *apps* em AR, eles são restritos às novas versões de *Android*, que por sua vez, estão presentes somente em dispositivos de nova geração, i.e, de lançamento posteriores a 2018. Por fim, a incompatibilidade das bibliotecas com o sistema e a construção dos óculos AR foram os motivos para a procura de um novo ambiente de desenvolvimento, que seja mais versátil e comporte bem com os recursos do *Moverio BT-350*.

2.2 Estudos de desenvolvimento Unity

Unity é uma plataforma voltada para o desenvolvimento de jogos em múltiplas plataformas, sendo o *Android* uma delas, e com o apoio da USP, temos acesso a mais recursos na plataforma (UNITY..., 2021). A documentação da *EPSON* forneceu um programa de exemplo para *Unity* que pode reconhecer um certo carrinho de papel e sobrepor com animações em AR.

Foi sugerido a impressão e a montagem do carrinho de papel pela documentação, porém, como os óculos possuem somente uma *webcam* simples para a detecção, supomos que ele não pode diferenciar o carrinho real de uma fotografia do carrinho, então, ele funcionaria normalmente. De fato, isso ocorreu e o teste foi registrado nas figuras 2.2a e 2.2b.



(a) Modelo virtual do carrinho de papel



(b) Sobreposição em AR com *Moverio*

Figura 2.2: Resultados adquiridos dos testes da documentação *Moverio* np *Unity*. Fonte: Autor.

Este foi o único programa disponibilizado na documentação, o projeto foi elaborado no *Unity* 2017, mesmo com alguns alertas de incompatibilidade, o projeto pôde ser compilado com êxito e funcionou nos óculos. Novamente, a falta do suporte da *EPSON* deixou incerto se era uma boa opção construir um aplicativo somente baseado na documentação disponível. Por isso, iniciamos novamente uma pesquisa sobre as ferramentas para suporte de AR nessa plataforma.

O *Vuforia* foi o primeiro *plugin* a ser testado no ambiente do *Unity* (VUFORIA..., 2021). Atualmente, ele apresenta mais opções rastreo para AR, mas no momento em que foi encontrado pela pesquisa, as opções eram de usar imagens e algumas formas 3D como marcadores para suas projeções. Foi testado a detecção de imagem com um pedaço da logo da *Coca-cola™* como alvo, a figura 2.3 ilustra os resultados.



Figura 2.3: Testes com diferentes modelos de diferentes complexidades, constatando performance acima de 15 quadros por segundo em todos os casos. Fonte: Autor.

As formas de detecção tridimensionais não eram compatíveis com a detecção da posição da cabeça de um paciente, por isso esse recurso não foi testado. O *Vuforia* é uma ferramenta que trabalha com *Android* de *API 23* ou superior, infelizmente, também incompatível com o *Moverio BT-350* que possui uma *API 22* e sem chances de atualização. Outras ferramentas foram encontradas como o *Wikitude™* que também era para *API 23* (WIKITUDE..., 2022), porém uma outra ferramenta denominada *ARFoundation* que estima a posição da face humana chamou a atenção por termos a oportunidade de criar uma demonstração da ideia final do projeto nos óculos (AR-FOUNDATION..., 2022).

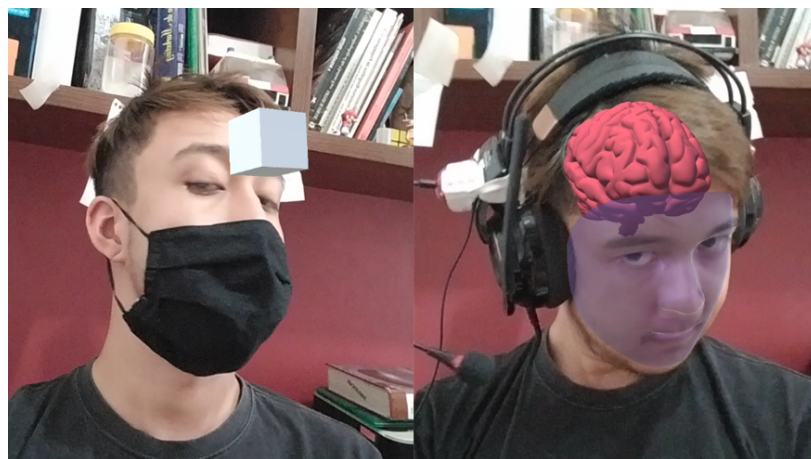


Figura 2.4: A primeira imagem foi o primeiro teste com a projeção de um cubo na região da testa. A segunda já detalha a superfície estimada da face e a projeção do cérebro. Fonte: Autor.

A instalação do *ARFoundation* foi feita no *Unity 2018* e apresentou compatibilidade com celulares de *API 28* ou superior. Os testes consistiram em tentar utilizar a posição encontrada pelo *plugin* para projetar um objeto em referência desses dados. A aplicação teve um ótimo resultado e foi possível apresentar o objetivo principal para professores e médicos da área pela aproximação visual dos testes com o resultado final esperado, visto na figura 2.4. Nesse momento, criamos um nome fantasia para o projeto: *VCranium*.

Retornando ao objetivo prático de encontrar um meio de desenvolver o projeto, concluímos que temos muitas opções disponíveis de desenvolvimento, mas sem muito suporte para o *Moverio BT-350*. O *Unity* foi escolhido para seguir a pesquisa pela sua facilidade de elaboração de cenários tridimensionais e compatibilidade com o sistema de outros óculos de AR do mercado. Dessa maneira, podemos garantir que os próximos trabalhos que serão desenvolvidos, não dependem do equipamento que temos hoje, i.e, se futuramente precisarmos trocar para um óculos mais moderno, uma adaptação será feita e aproveitaremos o material produzido até o momento.

Dado o histórico de testes do projeto, foi decidido prosseguir sem depender de *plugins* e bibliotecas *closed-source*, isso garantirá um maior controle do *software* elaborado e também exigirá um estudo mais aprofundado de como os algoritmos de visão computacional funcionam. Essa mudança trará um contato maior com problemas discriminados de programação, auxiliando o processo do estudo e apresentação de dúvidas em seminário para os estudantes e professores dessa área de pesquisa na universidade.

2.3 Arquitetura do sistema

Para a definição de uma arquitetura adequada para o projeto foram analisados os exemplos que a literatura pôde nos dar como

[Fazer Tabela de arquiteturas]

Para esse fim, a equipe realizou reuniões e debates para estabelecer uma solução que seja compatível para um período de seis meses e respeitando as medidas de prevenção por afastamento imposto pela pandemia de COVID-19. Decide-se criar sistema que envolve um computador executando um servidor em *Python*; uma

Foram estudados diversos tipos de arquitetura do sistema de projeção em realidade aumentada

A escolha da arquitetura do sistema foi baseada na aplicação dos conceitos menos complexos da visão computacional: a detecção da posição de um marcador no espaço. Essa aplicação foi o ponto de partida de estudos das projeções em realidade aumentada em imagens capturadas por uma câmera.

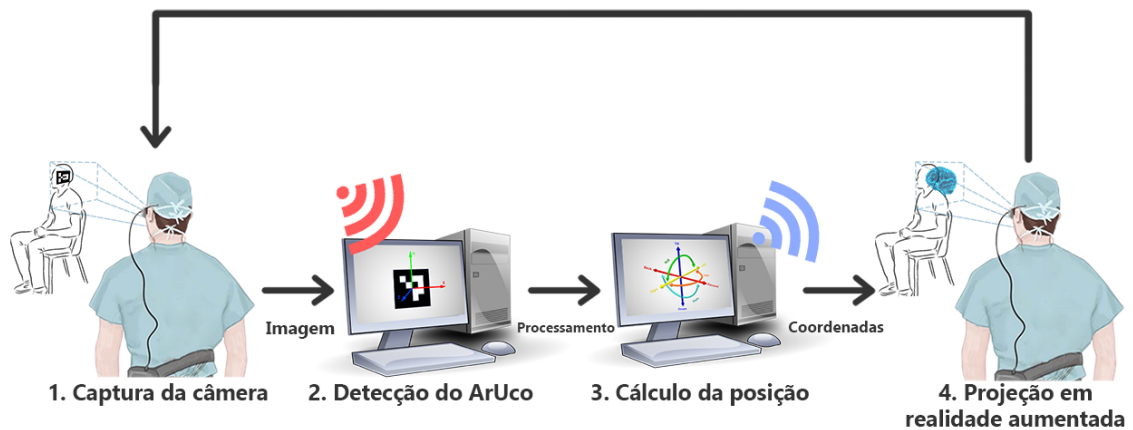


Figura 2.5: A figura representa o funcionamento do sistema. (1) Captura a imagem do paciente e envia para o computador. (2) Faz uma varredura na imagem e identifica o marcador ArUco. (3) Calcula a posição do marcador e envia as coordenadas para os óculos. (4) Recebe as informações e exibe a projeção para o usuário e então retorna para o passo 1. Fonte: Autor.

Referências bibliográficas

ABOUT OpenCV. [S.l.: s.n.]. <https://opencv.org/about/>. Acessado em 17 de fev. de 2020.

ARCORE: supported device. [S.l.: s.n.], 2021. <https://developers.google.com/ar/devices>. Acessado em: 7 de agosto de 2021.

ARFOUNDATION: Documentação do *plugin*. [S.l.: s.n.], 2022. <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/index.html>. Acessado em: 26 de fevereiro de 2022.

GOOGLE Play: Google Play *Services* para RA. [S.l.: s.n.], 2021. <https://play.google.com/store/apps/details?id=com.google.ar.core>. Acessado em: 5 de agosto de 2021.

MEDIAPIPE: *README* do projeto no *Github*. [S.l.: s.n.], 2022. <https://google.github.io/mediapipe/>. Acessado em: 26 de fevereiro de 2022.

SCENEFORM: Quickstart for Android. [S.l.: s.n.], 2021. <https://developers.google.com/sceneform/develop/android-quickstart>. Acessado em: 5 de agosto de 2021.

UDEMY: Desenvolvimento *Android* Completo 2021. [S.l.: s.n.], 2021. <https://www.udemy.com/course/curso-de-desenvolvimento-android-oreo/>. Acessado em: 5 de agosto de 2021.

UNITY: Arquivo de download Unity. [S.l.: s.n.], 2021. <https://unity3d.com/pt/get-unity/download/archive>. Acessado em: 10 de agosto de 2021.

VUFORIA: Developer Portal. [S.l.: s.n.], 2021. <https://developer.vuforia.com>. Acessado em: 9 de agosto de 2021.

WIKITUDE: Augmented Reality. [S.l.: s.n.], 2022. <https://www.wikitude.com/>. Acessado em: 1 de março de 2022.