



---

**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 comprehende 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 em aplicações cirúrgicas. Para isso, conceituamos as relações entre visão computacional e computação gráfica no campo da realidade aumentada. Com o apoio do Laboratório Aeronáutico de Tecnologias (AeroTech), definimos uma arquitetura de sistema que permite a sobreposição de um modelo 3D na cabeça de um paciente, indicando pontos de implantação de eletrodos durante o procedimento neurocirúrgico orientado por estereoeletroencefalografia (SEEG). Essa arquitetura foi baseada na comunicação de computador e óculos pela rede local: O computador recebe as imagens da câmera dos óculos e utiliza o método de detecção por marcadores fiduciais (*ArUco*) para estimar a posição do paciente e envia as coordenadas aos óculos, indicando onde a projeção deve ser exibida. Os resultados parciais da aplicação apresentam uma rápida responsividade com o método de estimativa escolhido.

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

# Sumário

<b>Informações gerais do projeto</b>	<b>i</b>
<b>Resumo</b>	<b>ii</b>
<b>1 Resumo do projeto</b>	<b>1</b>
1.1 Enunciado do problema . . . . .	1
1.2 Objetivo . . . . .	2
1.3 Metodologia . . . . .	2
1.4 Breve histórico do projeto . . . . .	3
<b>2 Realizações</b>	<b>4</b>
2.1 Calibração . . . . .	4
2.1.1 Rotação . . . . .	4
2.1.2 Translação . . . . .	5
2.1.3 <i>Field of View</i> . . . . .	5
2.1.4 Distorção . . . . .	6
2.2 Desenvolvimento Unity . . . . .	8
2.2.1 Ajuste dos intrínsecos . . . . .	8
<b>3 Proposta de extensão</b>	<b>10</b>
3.1 Objetivo . . . . .	11

3.2	Metodologia	11
3.3	Cronograma	13
<b>4</b>	<b>Participação de eventos acadêmicos</b>	<b>14</b>
	<b>Referências bibliográficas</b>	<b>15</b>

# 1 Resumo do projeto

## 1.1 Enunciado do problema

Atualmente na medicina, as cirurgias têm o objetivo de causar a mínima agressão ao organismo do paciente, isso implica em um menor tempo de hospitalização; menor incidência de complicações pós-operatórias; menos dor; e recuperação mais rápida (IPC..., 2019). Contudo, entre os diversos desafios de sua realização, a habilidade visual do médico, que envolve a capacidade de localizar e identificar os tecidos sensíveis, é essencial para redução dos danos ao paciente durante um procedimento cirúrgico, nos levando à questão: Como podemos potencializar a visão do médico para aumentar a qualidade dessas operações?



Figura 1.1: Exemplo da utilização de neuronavegador em cirurgia. Fonte: (SERVIÇO..., 2019).

Para esse fim, um *software* que informa e assiste o médico em tempo real na operação denota-se como um ótimo meio de aumentar as chances de sucesso dessas cirurgias. Atualmente, esses *softwares* são muito presentes no planejamento e execução das operações, entretanto, toma-se como exemplo os sistemas de neuronavegação, cuja função é permitir visualização, de forma precisa, das estruturas cerebrais do paciente. O neurocirurgião precisa virar a cabeça para ler os monitores e perde o foco na atividade conduzida, colaborando para um maior tempo de operação e favorecendo a exaustão do profissional, de modo que aumente seu potencial de erros durante os procedimentos cirúrgicos (Figura 1.1).

Para isso, a tecnologia de realidade aumentada (*AR*) oferece o potencial de reduzir es-

sas limitações, em razão de que figuras tridimensionais podem sobrepor a visão do médico para facilitar a visualização e a localização dos objetivos da operação enquanto mantém seus olhos direcionados ao paciente (CHO et al., 2020). O que foi feito nesse projeto é uma pesquisa que envolve visão computacional e computação gráfica no campo do *AR* e uma aplicação para os óculos da empresa *Seiko Epson Corporation*<sup>®</sup>, modelo *Moverio BT-350™*. Recebendo suporte físico e técnico da equipe do Laboratório Aeronáutico de Tecnologias (Aerotech) da Universidade de São Paulo, esse estudo corrobora com o objetivo do grupo que é desenvolver uma plataforma colaborativa de multi-aplicação neurocirúrgica.

## **1.2 Objetivo**

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 neurocirurgia; 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.

## **1.3 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 enfatiza os resultados encontrados pelos artigos, o objetivo disto é caracterizar os prós e contras das diversas opções encontradas. Essa pesquisa de artigos trás uma evolução do discernimento dos assuntos do estado da arte, refletindo em uma noção de funcionamento dos métodos científicos que constrói um novo pesquisador no desenvolvimento dessa iniciação científica.

## 1.4 Breve 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 1.2a); testes das ferramentas da documentação dos óculos de realidade aumentada *Moverio BT-350* (Figuras 1.2b e 1.2c); e a elaboração de aplicativos que ilustram o objetivo do *VCranium* (Figura 1.2d). Assim, como foi explicado no relatório parcial da pesquisa, pretendíamos prosseguir o desenvolvimento estabelecendo uma arquitetura composta por computador e óculos para capturar os dados necessários para a projeção em realidade aumentada, os detalhes serão descritos no capítulo das realizações.



Figura 1.2: Histórico de realizações da pesquisa até a primeira entrega parcial. Fonte: Autor.

## 2 Realizações

### 2.1 Calibração

O processo de calibração do *VCranium* foi complexo por abranger diversos conceitos de visão computacional e sua aplicabilidade na plataforma do *Unity* (UNITY..., 2021). Decidimos fragmentar o problema e estudá-los separadamente:

**Rotação:** Realizado com o *RVEC* de *Rodrigues* de cada *frame*;

**Translação:** Realizado com o *TVEC* de *Rodrigues* de cada *frame*;

**Field of View:** (*FOV*) Calculado com as informações intrínsecas da calibração da câmera;

**Distorção:** Calculado com as informações de distorção da calibração da câmera;

#### 2.1.1 Rotação

A informação da rotação do modelo virtual do cérebro é adquirida da comunicação *socket* com o servidor em *Python*. Abaixo está estruturada o formato da mensagem enviada a cada iteração do computador para a aplicação:

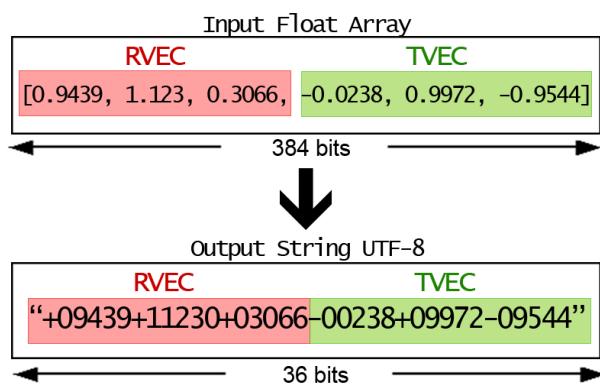


Figura 2.1: Estrutura da mensagem via *socket*. Fonte: Autor.

Um método foi implementado para a formatar cada mensagem gerada pela detecção da posição do marcador, transformando uma lista do *Python* em uma *string* com o formato de *UTF-8*, como é mostrado no *RVEC* da Figura 2.1. Assim que a aplicação recebe os três números, eles representam o vetor de rotação com as coordenadas X, Y e Z, respectivamente. Para descobrirmos a magnitude da rotação neste eixo, deve-se calcular o módulo desse vetor de rotação, o resultado é dado em radianos.

### 2.1.2 Translação

Da mesma forma que a rotação, a translação é recebida do *TVEC* da mensagem da Figura 2.1. Nesta, a informação é recebida em milímetros, portanto, devemos ajustar a escala da translação manualmente para corrigir a projeção do marcador no mundo virtual do *Unity*. Com a rotação e translação completas, temos a calibração extrínseca da projeção. Assim como foi explicado no último relatório, a projeção deve estar visualmente boa, especialmente nas regiões centrais da tela, representado na Figura 2.2.

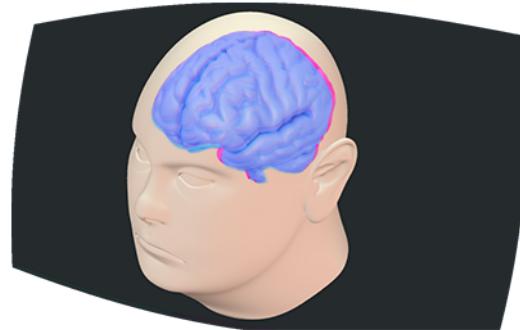


Figura 2.2: Exemplo de sobreposição sem calibração intrínseca. Fonte: Autor.

### 2.1.3 *Field of View*

O *Field of View* (*FOV*) é calculado da calibração da câmera com o *chessboard* que calcula a distorção e os intrínsecos da câmera, nomeados respectivamente de *DIST* e *MTX*. Para determinar o *FOV* da câmera observamos a matriz *MTX* [FONTE]:



Figura 2.3: Calibração dos intrínsecos da câmera com o *chessboard*. Fonte: Autor.

$$MTX = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Utilizamos as distâncias focais da câmera,  $f_x$  ou  $f_y$ , e a informação das dimensões da imagem em *pixels* para obter o *FOV* em X e Y:

$$FOV_x = 2 \arctan \left( \frac{w}{2f_x} \right); \quad FOV_y = 2 \arctan \left( \frac{h}{2f_y} \right)$$

Sendo  $w$  e  $h$  a largura e a altura em *pixels*, respectivamente. Não existe diferença em utilizar qualquer um dos dois, pois o *Unity* suporta ambas as configurações no objeto da câmera virtual.

#### 2.1.4 Distorção

A documentação do *OpenCV* baseia-se na câmera *pinhole* e modela duas principais distorções: radial e tangencial.

**Radial:** A distorção radial é percebida como retas se curvando em relação ao centro. Causada pela fabricação da forma circular das lentes, o que contribui para um maior campo de visão.

$$x_{distorted} = x(1 + k_1r^2 + k_2r^4 + k_3r^6)$$

$$y_{distorted} = y(1 + k_1r^2 + k_2r^4 + k_3r^6)$$

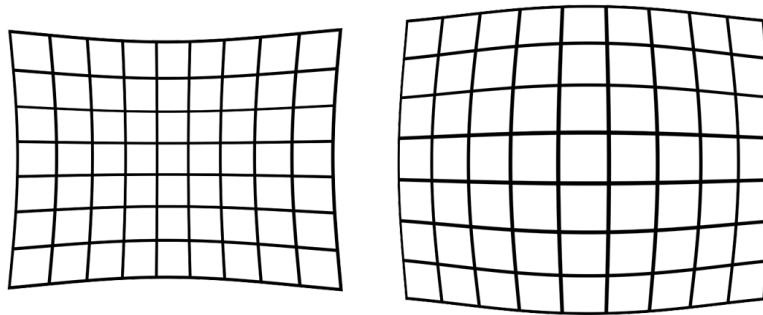


Figura 2.4: Exemplos de distorção radial. Fonte: Autor.

**Tangencial:** A distorção tangencial é percebida em regiões da imagem que parecem mais próximas ou mais distantes da câmera de forma não uniforme. Causada pelas tolerâncias de paralelismo da montagem do sensor com as lentes internas do equipamento.

$$x_{distorted} = x + 2p_1xy + p_2(r^2 + 2x^2)$$

$$y_{distorted} = y + p_1(r^2 + 2y^2) + 2p_2xy$$

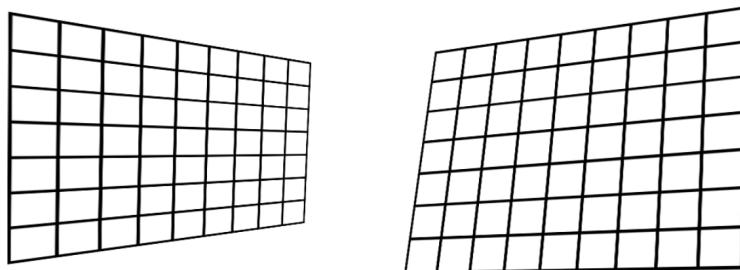


Figura 2.5: Exemplos de distorção tangencial. Fonte: Autor.

Utilizando as fórmulas apresentadas acima e os parâmetros intrínsecos  $f_x$   $f_y$   $c_x$   $c_y$  e distorção  $k_1$   $k_2$   $k_3$   $p_1$   $p_2$ , tem-se todas as variáveis necessárias para a modelagem da câmera com *OpenCV*:

$$DIST = [k_1 \ k_2 \ p_1 \ p_2 \ k_3]; \quad MTX = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix};$$

## 2.2 Desenvolvimento Unity

Trazendo a teoria apresentada na Seção 2.1 para a realidade no ambiente de desenvolvimento do *Unity*. Para isso, deve-se organizar de "cenas", onde o usuário pode transitar utilizando a interface, como na Figura 2.6, e utilizar a linguagem C# como *back-end* para a criação de *scripts*.



(a) Menu principal do programa

(b) Tela de conexão com computador

Figura 2.6: Imagens da interface do *VCranium*. Fonte: Autor.

### 2.2.1 Ajuste dos intrínsecos

A Seção 2.1 também explicita que obtemos todos os parâmetros intrínsecos e de distorção de uma câmera após a calibração com o método do *chessboard* (Figura 2.3). Porém, é necessário um método que realize a calibração da câmera em que o *VCranium* está sendo executado, para isso, deve-se criar um método auxiliar que capture uma série de imagens e envie para o computador calibrar e encontrar os parâmetros.

A principal ideia que se tem ao criar esse sistema é receber uma imagem assim que o usuário pressionar o botão de captura. Assim como foi estudado em relatórios anteriores, é necessário construir um *header* simples que antecede o envio da imagem para o servidor preparar a sua recepção, e então, guardar a informação de cada imagem em uma lista auxiliar. O próximo passo é usar o comando *calibrateCamera* da biblioteca do *OpenCV* e o programa retorna as informações necessárias ( $f_x$ ,  $f_y$ ,  $c_x$ ,  $c_y$ ,  $k_1$ ,  $k_2$ ,  $k_3$ ,  $p_1$ ,  $p_2$ ).

O desenvolvimento desse programa auxiliar não apresenta maiores desafios se comparado com o histórico do projeto desde 2021. No entanto, assim que o aplicativo *Unity* receber todos os parâmetros, não é trivial a distorção do modelo para a sobreposição em uma aplicação de realidade do tipo *VST* (*Video See-Through*).

Visto que seria necessário um estudo mais aprofundado no assunto de calibração, elaboração de conceitos da disciplina de Computação Gráfica para a vista em perspectiva, e a aplicação desse trabalho seriam restritos somente ao modelo *VST*, foi decidido privar a calibração intrínseca do projeto no dado momento. A Figura 2.2 mostra que ao fim da calibração extrínseca, a sobreposição é visualmente boa, i.e., em boa parte de ilustração, o erro da sobreposição é insignificante e, com isso, não apresenta problemas para demonstração de conceitos.

### 3 Proposta de extensão

No decorrer dos dois anos de desenvolvimento do projeto, buscou-se a formação no campo de desenvolvimento de aplicações, comunicação via *socket*, aprendizado de *Unity*, aprofundamento em visão computacional e, principalmente, pesquisa bibliográfica em aplicação de realidade aumentada em neurocirurgias. Contudo, toma-se a abordagem da aplicação dos óculos *Moverio BT-350* como ferramenta auxiliar de visualização anatômica do paciente em tempo real durante uma neurocirurgia. Para isso, foi proposto uma arquitetura compatível com equipamentos disponíveis e com tempo de resposta adequado.

No momento, o projeto funciona da maneira esperada, porém com muito espaço para ser aperfeiçoado antes de ser levado para a avaliação dos médicos. Os quesitos que podem ser melhorados são:

**Método da Detecção de Posição:** O projeto utiliza somente um marcador *ArUco* para estimar sua posição, o que traz principalmente o problema do usuário perder o marcador de vista por causa do ângulo de observação.

**Otimização da latência do sistema:** Os fatores que limitam o tempo de resposta do sistema é principalmente a conexão sem fio dos óculos com o *access point* e do roteador até o servidor.

A solução desses fatores torna a pesquisa apresentável para os médicos do Centro de Cirurgia de Epilepsia (CIREP) do Hospital das Clínicas da Faculdade de Medicina de Ribeirão Preto-USP. A parceria aproxima dos integrantes do laboratório dos pesquisadores do CIREP em forma de reuniões *online* e eventuais visitas até o hospital das clínicas. Seguindo essa oportunidade, se forem preparadas demonstrações do *VCranium*, o projeto pode receber sugestões diretas dos médicos e compiladas em um artigo científico.

### 3.1 Objetivo

Otimizar os processos do sistema de reconhecimento de posição da cabeça do paciente; definir um método de reconhecimento mais robusto e que permita demonstrações. Por fim, apresentar o projeto aos pesquisadores do Centro de Cirurgia de Epilepsia do Hospital das Clínicas da Faculdade de Medicina de Ribeirão Preto-USP.

### 3.2 Metodologia

O projeto já possui uma arquitetura que permite a alteração da forma de final de reconhecimento da pose de uma cabeça humana. Dessa forma, alterar o método de detecção de *ArUco* pode ser feito sem problemas. A confiabilidade comprovada dos métodos dos marcadores pode ser ampliada com o uso de mais marcadores, criando uma maior cobertura de atuação do sistema.

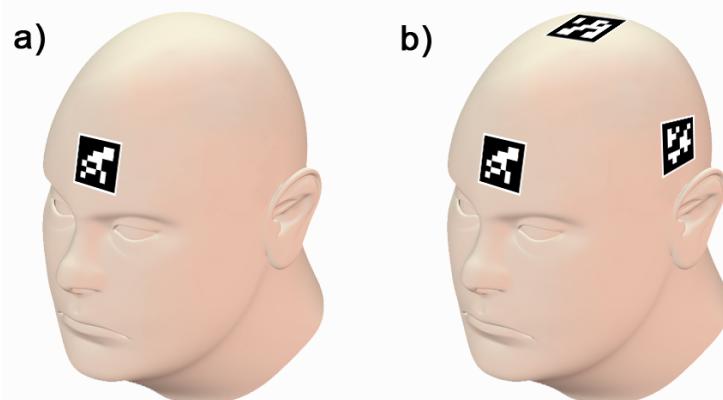


Figura 3.1: a) Aplicação atual de marcadores no projeto. b) Exemplo de utilização de múltiplos marcadores. Fonte: Autor.

A aplicação atual de marcadores do projeto pode ser vista na Figura 3.1a. Podemos observar que o sistema pode estimar a posição da cabeça em somente sua parte frontal, portanto, o sistema não funciona se observamos as laterais e atrás da cabeça. A Figura 3.1b traz uma solução a esse problema posicionando 5 marcadores em todos os lados da cabeça, assim o sistema pode estimar a posição do cérebro individualmente (Figura 3.2).

Esse método utiliza transformações de base, fundamentados da disciplina de Geometria Analítica e Álgebra Linear, para a estimar a posição central da cabeça e sua orientação. Para isso, deve-se saber a posição e orientação de cada marcador em relação ao centro da cabeça e, visto que

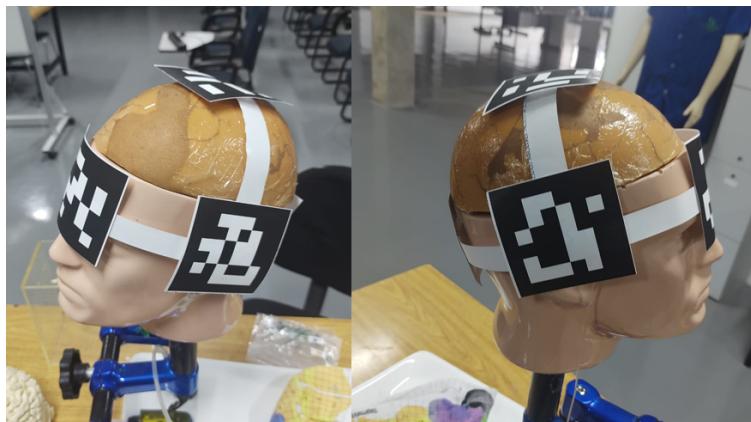


Figura 3.2: Esboço de uma tiara que suporta um marcador em cada lado da cabeça. Fonte: Autor.

os tamanhos da cabeça variam, encontra-se a necessidade de uma calibração para cada paciente. Pensando na impossibilidade do contato direto do marcador com a pele do paciente e a possível oclusão do marcador por ferramentas e as mãos do médico, esse método se torna uma opção menos viável para situações cirúrgicas.

Uma solução alternativa seria utilizar múltiplos marcadores distantes do paciente, mas ainda visíveis na câmera dos óculos como na Figura 3.3. Os marcadores em formato de cubo facilitam a montagem a calibragem de cada marcador em relação ao centro da cabeça.

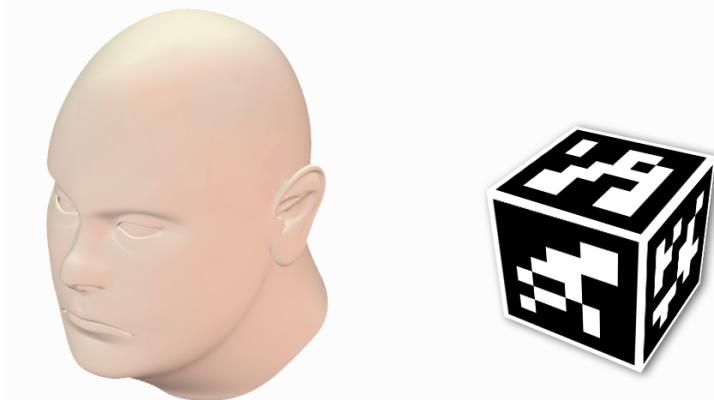


Figura 3.3: Solução de estimação de posição com cubo de marcadores. Fonte: Autor.

Por fim, o projeto vai aproveitar o tempo de extensão para fazer um pedido de Bolsa Estágio de Pesquisa no Exterior (BEPE) para buscar formação com grupos de pesquisa do exterior. O intercâmbio, além de colocar a comunicação acadêmica em prática, tem a oportunidade de receber críticas aprofundadas e criativas no assunto de realidade aumentada, visto que muito desses equipamentos não são vendidos no Brasil.

### **3.3 Cronograma**

O controle de atividades vai ser feito com encontros mensais e semanais com o coorientador, orientador e integrantes do laboratório. Um cronograma foi elaborado na tabela 3.1 para servir de guia para o progresso do projeto.

<b>Atividade</b>	<b>Bimestre</b>	<b>Execução</b>					
		1	2	3	4	5	6
Programação dos novos métodos com marcadores							
Montagem dos protótipos de tiara e cubo							
Estudos no exterior (BEPE)							
Produção de relatório final e relatório BEPE							
Apresentação final do trabalho*							

Tabela 3.1: \*A apresentação para os pesquisadores da parceria do Centro de Cirurgia de Epilepsia (CIREP) do Hospital das Clínicas da Faculdade de Medicina de Ribeirão Preto-USP

## 4 Participação de eventos acadêmicos

Novamente, os docentes e discentes do laboratório Aerotech se orgulham de todos terem a oportunidade promover nossas pesquisas no 30º Simpósio Internacional de Iniciação Científica e Tecnológica da USP (SIICUSP). Dentre os integrantes, somente um deles foi convidado para a segunda fase que, felizmente, também foi condecorado com uma menção honrosa.

Durante a apresentação, o projeto recebeu algumas provocações que ampliaram as possibilidades de aperfeiçoamento em aspectos de otimização e aplicabilidade. Durante uma apresentação *online* foram respondidas as perguntas:

**É possível mudar de Python para C/C++ para otimizar processo?** Resposta: É possível, porém isso traz problemas no quesito da velocidade do desenvolvimento, disponibilidade de documentação, bibliotecas e ferramentas. No entanto, é uma ótima opção quando o projeto ter uma versão funcional para testes, tendo um potencial reduzir a latência de cada iteração.

**O quanto longe o projeto está de um teste prático?** Resposta: No momento, o projeto consegue fazer demonstrações básicas com as funções de detectar e sobrepor um modelo virtual de um cérebro um marcador em tempo real (aproximadamente 15 vezes por segundo). Porém, o projeto abre muitas possibilidades de ser aperfeiçoados e, por isso, é desejável um trabalho sobre o programa atual antes da apresentação final aos médicos.

As provocações dos avaliadores trouxeram a mensagem que o projeto ainda possui espaço para desenvolver seu algoritmo na questão da otimização e robustez que foram refletidos na Seção 3. Espera-se que a extensão do projeto tenha um perfil voltado a construção de ferramentas para testes práticos do sistema.

## Referências bibliográficas

- CHO, J. et al. Enhancing Reality: A Systematic Review of Augmented Reality in Neuronavigation and Education. **World Neurosurgery**, p. 186–195, mai. 2020. DOI: 10.1016/j.wneu.2020.04.043.
- IPC Instituto de Patologia da Coluna. [S.l.: s.n.], 2019. <https://neurocirurgia.com/cirurgia-minimamente-invasiva/>. Acessado em: 18 de março de 2022.
- SERVIÇO de Neurocirurgia retira lesão cerebral com paciente acordado: Youtube. [S.l.: s.n.], 2019. <https://youtu.be/-n1x25yu04w>. Acessado em: 5 de março de 2022.
- 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.