



# Defesa de Dissertação

## Avaliação de Técnicas de Reconhecimento de Padrões em Ambientes Aeronáuticos

Bruno Duarte Corrêa <sup>1</sup>

<sup>1</sup>Programa de Pós-Graduação em Engenharia Aeronáutica e Mecânica

Mestrado Profissional em Engenharia Aeronáutica

Bruno Duarte Corrêa

Defesa de Dissertação

# Sumário

- 1 Introdução
- 2 Contexto
  - Cenário
- 3 Objetivos
- 4 Fundamentação Teórica
  - Realidade Aumentada
  - Características
  - Reconhecimento
- 5 Metodologia
  - Definição de Parâmetro
  - Análise de dados
- 6 Resultados
  - Taxa de Acertos

# Introdução

As diversas abordagens e peculiaridades adotadas pelos algoritmos atuais, fazem com que a seleção ideal não seja trivial, sendo portanto, necessária uma análise prévia das características apresentadas pela cena em questão.

## Contexto

O contexto dessa tese prevê o cenário de manutenção com o uso de realidade aumentada como uma ferramenta para auxílio nas tarefas rotineiras. Algumas variáveis devem ser consideradas para garantir a viabilidade de implantação da abordagem:

- Velocidade de reconhecimento;
- Qualidade do reconhecimento;
- Invariância à parâmetros ambientais.

Como caso de uso será adotado a janela de inspeção frontal, como mostrado na imagem 1, localizada na aeronave Embraer ERJ-190.



Figura: Posicionamento da LRU

# Objetivos

- Avaliar os algoritmos clássicos de reconhecimento
- Aplicar os algoritmos clássicos à situações reais
- Selecionar algoritmo mais adequado para o contexto

# Realidade Aumentada

A realidade aumentada é uma técnica de visão computacional em que valendo-se de artefatos do mundo real, tem por objetivo causar sensação de imersão do usuário em um ambiente aumentado por artefatos virtuais, ao contrário de ambientes puramente virtuais como é comum em aplicações de realidade virtual

# Realidade Aumentada Aplicada

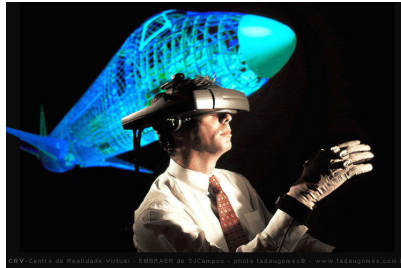


Figura: Ambiente Imersivo. Fonte CRV EMBRAER



# Realidade Aumentada Aplicada



**Figura:** Aplicação de Realidade Aumentada no meio automobilístico Fonte [www.metaio.com.br](http://www.metaio.com.br)

## Processo de Realidade Aumentada

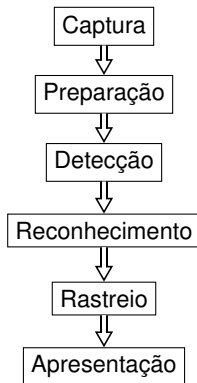


Figura: Processo Canônico de Realidade Aumentada

## Características



Figura: Reconhecendo características. Fonte [?]

Introdução  
Contexto  
Objetivos  
**Fundamentação Teórica**  
Metodologia  
Resultados  
Conclusão  
Propostas de Trabalhos Futuros  
Agradecimento

Realidade Aumentada  
**Características**  
Reconhecimento

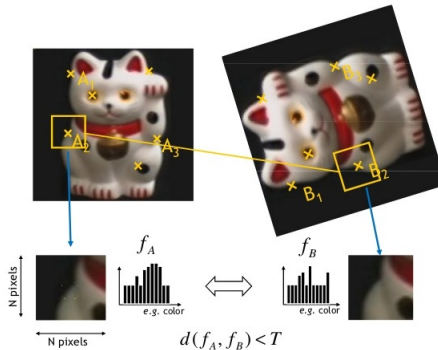
# Detector de característica

Introdução  
Contexto  
Objetivos  
**Fundamentação Teórica**  
Metodologia  
Resultados  
Conclusão  
Propostas de Trabalhos Futuros  
Agradecimento

Realidade Aumentada  
**Características**  
Reconhecimento

## Descritor de característica

# Reconhecimento



**Figura:** Ilustração do procedimento de reconhecimento com características locais. Fonte [2]

Processo de reconhecimento como ilustrado na imagem 6:

- Encontrar um grupo de *keypoints* distintos;
- Definir uma região em torno de cada *keypoint*;
- Extrair e normalizar o conteúdo da região;
- Calcular um descritor para a região normalizada;
- Encontrar correspondências de descritores.

# Metodologia

Os algoritmos são selecionados por meio de resultados de simulação feitos utilizando-se dos algoritmos: BRISK, FAST, FREAK, GFTT, MSER, ORB, STAR, SURF, SIFT em uma aplicação desenvolvida utilizando OpenCV e instrumentada para recuperar informações de qualidade de reconhecimento e tempo de execução





## Definição de Parâmetros

- **Variação de escala** para simular a aproximação dentro da janela de inspeção;
- **Variação de rotação** para simular a movimentação durante a manutenção;
- **Variação de iluminação** para simular manutenções feitas em horários do dia e iluminação diferente, como por exemplo, ambientes com neve com brilho muito maior;
- **Adição de *Blur*** para emular ambientes com muita poeira, esfumaçados ou ainda movimentos bruscos.

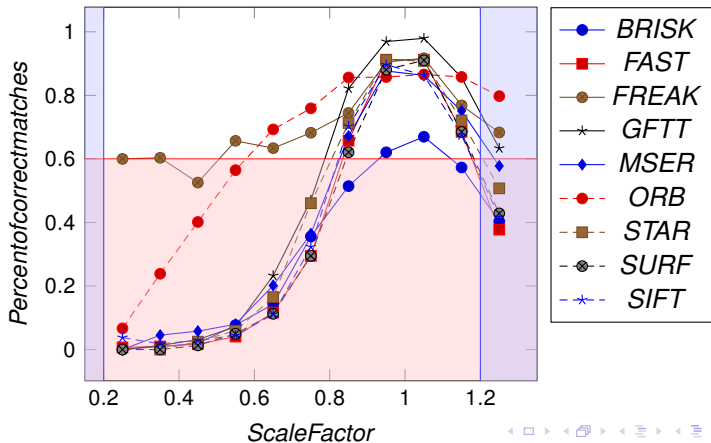
# Análise de dados

- Cadência  $> 5\text{fps}$  ou seja, tempo  $< 200\text{ms}$ ;
- Robustez à escala:  $0.5 < \text{escala} < 1.2$ ;
- Robustez à rotação:  $0^\circ < \text{rotação} < 30^\circ$ ;
- Robustez à iluminação:  $-50 < \text{iluminação} < 50$
- Robustez à *blur*:  $\text{kernelsize} < 4$

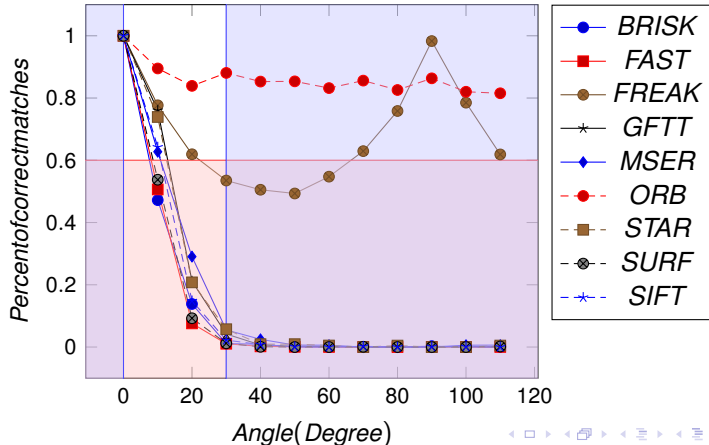
# Resultados

- Taxa de Acertos: a razão entre a quantidade de características corretamente localizadas e o número de características na imagem inicial
- Análise de Tempo: os usuários não toleram aplicações com menos do que 5fps

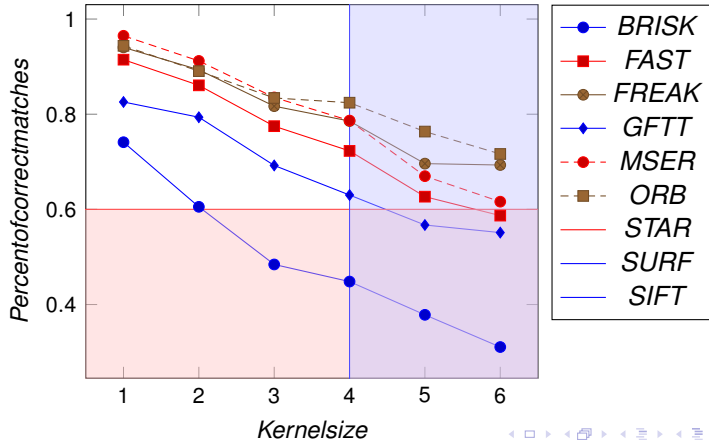
## Robustness to scaling

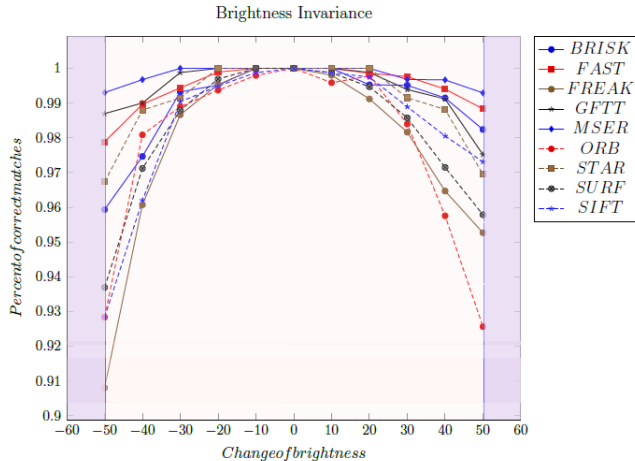


## Rotation Invariance

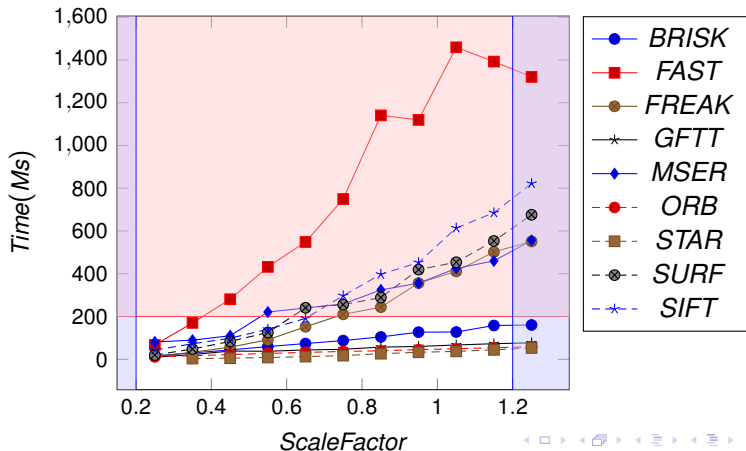


## Robustness to blur



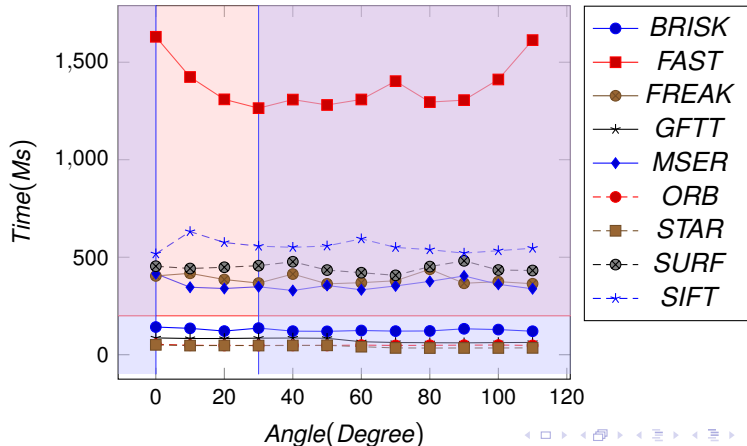


## Robustness to scaling

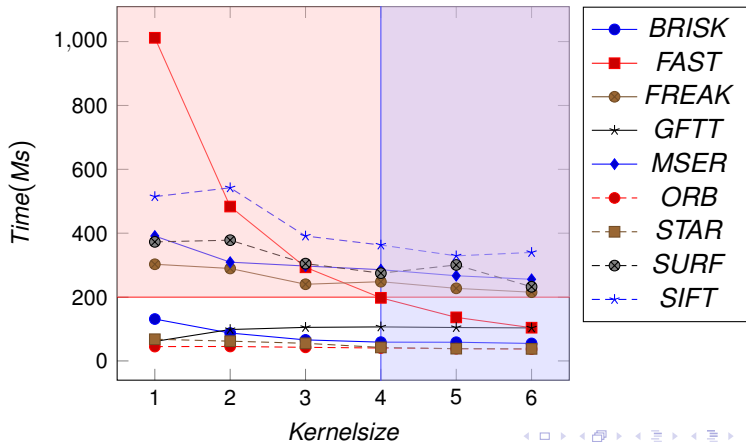




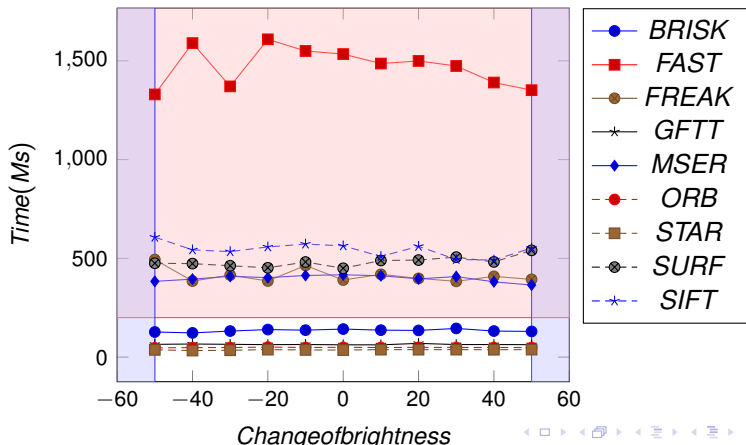
## Rotation Invariance



## Robustness to blur



## Brightness Invariance





# Conclusão

## **Avaliar os algoritmos clássicos de reconhecimento**

O trabalho apresentou os algoritmos clássicos, descrevendo seu funcionamento e realizando testes de desempenho com os mesmos.

## **Aplicar os algoritmos clássicos à situações reais**

Os algoritmos foram implementados em um protótipo desenvolvido em C++ e OpenCV, configurado para rodar simulações dos algoritmos BRISK, FAST, FREAK, GFTT, MSER, ORB, STAR, SURF, SIFT, gerando ao final de cada simulação resultados de precisão de reconhecimento e de tempo gasto.

# Conclusão

## **Selecionar algoritmo mais adequado para o contexto**

Análise comparativa entre os algoritmos utilizando as restrições e criado nos gráficos, janelas de decisão, gerando uma matriz de decisão com os resultados de todos os testes, tanto de qualidade de reconhecimento quanto de tempo de execução.

## Propostas de Trabalhos Futuros

- Adequar a aplicação para dispositivos móveis como tablets, celulares ou mesmo dispositivos HMD de forma a dar mais flexibilidade ao condutor da manutenção;
- Realizar o casamento de padrões com vídeos e imagens em tempo real, utilizando as técnicas identificadas, otimizando a aplicação para se tornar o mais tempo real e aceitável possível;

## Propostas de Trabalhos Futuros

- Adaptar a aplicação para utilizar processamento paralelo e processamento em GPU, visto os algoritmos serem recursivos e localmente independentes;
- Analisar por meio de testes em campo com possíveis usuários para abstrair parâmetros de usabilidade, como por exemplo determinar que tipo de informação seria útil ao usuário ou mesmo que tipo de configuração de dispositivo seria o mais adequado para uma aplicação desse porte.



# Obrigado

Bruno Duarte Corrêa

[bruno.duarte@gmail.com](mailto:bruno.duarte@gmail.com)