

Dissertação apresentada à Pró-Reitoria de Pós-Graduação e Pesquisa do Instituto Tecnológico de Aeronáutica, como parte dos requisitos para obtenção do título de Mestre em Engenharia no Curso de Mestrado Profissional em Engenharia Aeronáutica no Programa de Pós-Graduação em Engenharia Aeronáutica e Mecânica., Área de Informática.

Bruno Duarte Corrêa

**AVALIAÇÃO DE TÉCNICAS DE
RECONHECIMENTO DE PADRÕES EM
AMBIENTES AERONAUTICOS**

Dissertação aprovada em sua versão final pelos abaixo assinados:

Prof. Dr. Luís Gonzaga Trabasso
Orientador

Eng. Nelson José Issa de Macedo
Coorientador

Prof. Dr. Luiz Carlos Sandoval Góes
Pró-Reitor de Pós-Graduação e Pesquisa

Campo Montenegro
São José dos Campos, SP - Brasil
2015

Dados Internacionais de Catalogação-na-Publicação (CIP)
Divisão Biblioteca Central do ITA/CTA

Duarte Corrêa, Bruno

Avaliação de técnicas de reconhecimento de padrões em ambientes aeronauticos / Bruno Duarte Corrêa.

São José dos Campos, 2015.

63f.

Dissertação de Mestrado – Curso de Mestrado Profissional em Engenharia Aeronáutica no Programa de Pós-Graduação em Engenharia Aeronáutica e Mecânica.. Área de Informática – Instituto Tecnológico de Aeronáutica, 2015. Orientador: Prof. Dr. Luís Gonzaga Trabasso. Coorientador: Eng. Nelson José Issa de Macedo.

1. Augmented reality. 2. Opencv. 3. Computer vision. 4. Maintenance. 5. Feature matching. 6. Descriptors. 7. Markerless. I. Centro Técnico Aeroespacial. Instituto Tecnológico de Aeronáutica. Divisão de Engenharia da Computação. II. Título.

REFERÊNCIA BIBLIOGRÁFICA

DUARTE CORRÊA, Bruno. **Avaliação de técnicas de reconhecimento de padrões em ambientes aeronauticos**. 2015. 63f. Dissertação de Mestrado – Instituto Tecnológico de Aeronáutica, São José dos Campos.

CESSÃO DE DIREITOS

NOME DO AUTOR: Bruno Duarte Corrêa

TÍTULO DO TRABALHO: Avaliação de técnicas de reconhecimento de padrões em ambientes aeronauticos.

TIPO DO TRABALHO/ANO: Dissertação / 2015

É concedida ao Instituto Tecnológico de Aeronáutica permissão para reproduzir cópias desta Dissertação e para emprestar ou vender cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta Dissertação pode ser reproduzida sem a autorização do autor.

Bruno Duarte Corrêa

Av. Juscelino Kubistcheck, 6701 ap 14 bl 21

CEP 12.220-000 – São José dos Campos–SP

AVALIAÇÃO DE TÉCNICAS DE RECONHECIMENTO DE PADRÕES EM AMBIENTES AERONAUTICOS

Bruno Duarte Corrêa

Composição da Banca Examinadora:

Prof. Dr.	Luís Gonzaga Trabasso	Presidente	-	ITA
Prof. Dr.	Luís Gonzaga Trabasso	Orientador	-	ITA
Eng.	Nelson José Issa de Macedo	Coorientador	-	EMBRAER
Prof. Dr.	Ricardo Bedin França	Membro Externo	-	EMBRAER
Prof. Ph.D.	Osamu Saotome	Membro	-	ITA

ITA

Dedico esse trabalho primeiramente a Deus e a todas as pessoas que me apoiaram, acreditaram e incentivaram, sem as quais, com toda certeza não teria obtido os resultados que esperava.

"Persistence is the shortest path to success."

— CHARLES CHAPLIN

Resumo

O reconhecimento de objetos em uma cena para posterior uso em realidade aumentada depende de diversas variáveis, causando a necessidade do uso de técnicas específicas para cada cenário, sendo portanto, um estudo de fronteiras para a melhor escolha do algoritmo de reconhecimento, de acordo com a aplicação em questão de grande valia para o meio acadêmico. Esta tese se propõe a pesquisar, categorizar e traçar fronteiras das técnicas conhecidas, tendo como caso de uso a manutenção de aeronaves feita dentro de centros fechados, utilizando as técnicas BRISK, FAST, FREAK, GFTT, MSER, ORB, STAR, SURF, SIFT em uma análise aplicada com imagens reais de janelas de inspeção do Embraer ERJ-190 para reconhecimento de objetos e posteriores aplicações em manutenção. Comparando todas as técnicas quanto à cadência e à precisão de reconhecimento de características, é possível selecionar GFTT e ORB como técnicas mais apropriadas ao contexto, por terem seus resultados de variação de rotação, escala, brilho e *blur* dentro de uma faixa esperada para o contexto de manutenção.

Abstract

Recognition of objects in a scene for later use in augmented reality depends on several variables, causing the need to use several specific techniques for each scenario, and therefore a border analysis to the best choice of the registration algorithm according to application in question, of great value to academia. This thesis aims to examine, categorize and draw boundaries of the known techniques having as use case maintenance of aircraft made in closed center, using the techniques BRISK,FAST,FREAK,GFTT,MSER,ORB,STAR,SURF,SIFT in an applied analysis with images of the Embraer ERJ-190 to recognition of objects and future usage in maintenance. Comparing all of the techniques, using cadency and recognition precision, it is possible to chose GFTT and ORB as the most appropriate ones because its results to the variation of rotation, scale, brightness and blur fulfils the constraints needed to the maintenance context.

Sumário

LISTA DE FIGURAS	xi
LISTA DE TABELAS	xiii
LISTA DE ABREVIATURAS E SIGLAS	xiv
1 INTRODUÇÃO	15
1.1 Motivação	15
1.2 Objetivos	16
1.3 Justificativa	17
1.4 Contextualização	18
1.4.1 Cenário	18
1.4.2 Variáveis de contorno	19
1.5 Recursos e Métodos	19
1.6 Estrutura do Texto	19
2 FUNDAMENTAÇÃO TEÓRICA	21
2.1 Realidade Virtual	21
2.2 Realidade Aumentada	22
2.3 Cadência	23
2.4 Filtros Lineares	24

2.5	Reconhecimento	27
2.6	Características Locais	28
2.6.1	Características	28
2.6.2	Propriedades da Característica Local ideal	30
2.6.3	Detecção de características	31
2.6.4	Correspondência de características	31
2.7	Imagem Integral	32
2.8	Algoritmos de Reconhecimento	33
2.8.1	FAST - <i>Features from Accelerated Segment Test</i>	33
2.8.2	BRISK - <i>Binary Robust Invariant Scalable Keypoints</i>	34
2.8.3	FREAK - <i>Fast Retina Keypoint</i>	35
2.8.4	GFTT - <i>Good Features To Track</i>	37
2.8.5	MSER - <i>Maximal Stable Extremal Regions</i>	38
2.8.6	ORB - <i>Oriented Fast and Rotated Brief</i>	39
2.8.7	SIFT - <i>Scale-Invariant Feature Transform</i>	39
2.8.8	SURF - <i>Speeded Up Robust Feature</i>	41
2.8.9	STAR	43
3	APLICAÇÃO E SELEÇÃO DE ALGORITMOS	45
3.1	Metodologia	45
3.1.1	Definição de parâmetros	45
3.2	Análise de dados	45
3.2.1	Protótipo	46
4	ANÁLISE DE RESULTADOS	49
4.1	Taxa de Acertos	49
4.2	Análise de Tempo	53

4.3	Seleção da Técnica	57
5	CONCLUSÃO	59
5.1	Atendimento dos objetivos	59
5.2	Proposta de Trabalhos Futuros	60
	REFERÊNCIAS	61

Lista de Figuras

FIGURA 1.1 – Posicionamento da janela de inspeção. Fonte http://www.aero-news.net/	18
FIGURA 2.1 – Aplicação de realidade virtual. Fonte(KIRNER; SISCOUTO, 2007)	22
FIGURA 2.2 – Processo de Realidade Aumentada	23
FIGURA 2.3 – Kernel para Convolução do filtro de médias com vizinhança 3x3	25
FIGURA 2.4 – Kernel para Convolução do filtro gaussianos simétrico. Fonte (FORSYTH; PONCE, 2012)	26
FIGURA 2.5 – Aproximação discreta do kernel gaussiano. Fonte (FORSYTH; PONCE, 2012)	26
FIGURA 2.6 – Ilustração do procedimento de reconhecimento com características locais. Fonte (TUYTELAARS; MIKOLAJCZYK, 2007)	27
FIGURA 2.7 – Reconhecimento de padrões. Fonte (OPENCV, 2014)	29
FIGURA 2.8 – Regiões de reconhecimento de padrões. Fonte (OPENCV, 2014)	29
FIGURA 2.9 – Região da Figura integral. Fonte (VIOLA; JONES., 2001)	32
FIGURA 2.10 –Detector FAST. Fonte (ROSTEN; DRUMMOND, 2006)	33
FIGURA 2.11 –Padrão de amostras no BRISK. Fonte (LEUTENEGGER <i>et al.</i> , 2011)	35
FIGURA 2.12 –Padrão de amostragem do descritor FREAK. Fonte (ALAHÍ <i>et al.</i> , 2012)	36
FIGURA 2.13 –Distribuição de receptores na retina. Fonte (ALAHÍ <i>et al.</i> , 2012)	36
FIGURA 2.14 –Pares selecionados para calcular a orientação. Fonte (ALAHÍ <i>et al.</i> , 2012)	37
FIGURA 2.15 –Regiões MSER. Fonte (MATAS <i>et al.</i> , 2002)	38

FIGURA 2.16 –Variação de escala. Fonte (LOWE, 2004)	39
FIGURA 2.17 – <i>Difference of Gaussian</i> . Fonte (LOWE, 2004)	40
FIGURA 2.18 –Formação do descritor SIFT. Fonte (LOWE, 2004)	41
FIGURA 2.19 – <i>Box Filtering</i> . Fonte (BAY <i>et al.</i> , 2008)	42
FIGURA 2.20 –Janela do Descritor. Fonte (BAY <i>et al.</i> , 2008)	43
FIGURA 2.21 –Filtro bi-level aplicado à formas de n lados. Fonte (AGRAWAL <i>et al.</i> , 2008)	44
FIGURA 3.1 – Imagem utilizada no protótipo	47
FIGURA 4.1 – Resultado de performance de variação de escala	50
FIGURA 4.2 – Resultado de performance de variação de rotação	51
FIGURA 4.3 – Resultado de performance de variação de <i>blur</i>	52
FIGURA 4.4 – Resultado de performance de variação de iluminação	53
FIGURA 4.5 – Análise de tempo de variação de escala	54
FIGURA 4.6 – Análise de tempo de variação de rotação	55
FIGURA 4.7 – Análise de tempo de variação de blur	56
FIGURA 4.8 – Análise de tempo de variação de iluminação	57

Lista de Tabelas

TABELA 1.1 – Resumo de Recursos e Métodos	19
TABELA 4.1 – Decisão de técnica a utilizar	57

Lista de Abreviaturas e Siglas

AR	Augmented Reality
BRIEF	Binary Robust Independent Elementary Features
BRISK	Binary Robust Invariant Scalable Keypoints
DoG	Difference of Gaussians
FAST	Features from Accelerated Segment Test
FREAK	Fast Retina Keypoint
GFTT	Good Features To Track
GPU	Graphics Processing Unit
HMD	Head-Mounted Display
LoG	Laplacian of Gaussians
MSER	Maximally Stable Extremal Regions
ORB	Oriented Fast and Rotated Brief
SIFT	Scale-Invariant Feature Transform
SURF	Speed Up Robust Feature
STAR	Não é um acrônimo
VR	Virtual Reality

1 Introdução

1.1 Motivação

O reconhecimento de estruturas e sistemas de forma automática no campo da manutenção pode propiciar a construção de ferramentas de capacitação dentre várias outras possibilidades, auxiliando na garantia de maior confiabilidade no diagnóstico de problemas, na agilidade e precisão de execução das tarefas. Um dos mais básicos problemas atualmente, limitando o ramo da Realidade Aumentada, é a etapa de reconhecimento. A Realidade Aumentada prevê imersão entre o mundo virtual e o mundo real e por isso, para que a experiência seja coerente, é necessário que os dois mundos estejam bem sincronizados e propriamente alinhados. Em algumas situações, tal sincronia aumenta a experiência, entretanto o alinhamento é primordial, por exemplo, em aplicações médicas como uma aplicação de biópsia. Se o objeto não estiver no espaço e tempo real, a informação fornecida ao cirurgião poderá por em risco a vida do paciente. Na maioria das aplicações de tempo real, problemas de reconhecimento podem invalidar o uso da Realidade Aumentada. Um outro problema que pode ocorrer com falhas de reconhecimento é a tendência do cérebro em capturar com mais qualidade estímulos visuais, do que qualquer outro sentido e isso é acentuado por um fenômeno conhecido por *visual capture* (WELCH, 1978). Neste caso o sentido da visão tende a sobrepor os outros sentidos. Assim como um ventríloquo consegue enganar quem assiste um *show*, acreditando que o som sai da boca do boneco, o usuário de uma aplicação de realidade aumentada tende a acreditar no que vê, mesmo que esteja defasado no espaço/tempo. No caso do erro se tornar sistemático o usuário tende a se acostumar e inconscientemente adaptar-se ao erro, corrigindo o efeito. Erros de reconhecimento são difíceis de controlar adequadamente devido à grande precisão requerida das diversas fontes de erro, que podem ser divididas em estáticas e dinâmicas

sendo as estáticas contornadas com calibração prévia de sensores, entretanto os erros dinâmicos são mais difíceis porque são susceptíveis a instantes e intervalos de tempo entre o real e o apresentado na tela. O reconhecimento de objetos na cena permeia também:

- O contexto da cena facilitado pelo prévio conhecimento;
- O material do qual os objetos são constituídos;
- Informações suficientes em cada objeto, garantindo precisão de reconhecimento;
- Transformações afins como rotação, escala
- Variação de iluminação por meio de adição de intensidade, pixel a pixel.

Portanto para que as diversas fontes de erros dinâmicos não sejam um impeditivo para o reconhecimento, de acordo com a cena, algoritmos diferentes devem ser selecionados por terem peculiaridades e características que garantam sucesso de acordo com o tipo de desafio. 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.

1.2 Objetivos

O presente trabalho tem como objetivo geral avaliar o ambiente de manutenção aeronáutico, no contexto de janelas de inspeção, traçando estratégias de reconhecimento de itens de manutenção. Para a consecução do objetivo geral, foram definidos os seguintes objetivos específicos:

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

Esta dissertação propõe o uso da realidade aumentada no cenário de manutenção de aeronaves, tendo como objetivo determinar a melhor estratégia de reconhecimento de características dos objetos no contexto, para que posteriormente seja utilizado em ferramentas de auxílio na manutenção.

1.3 Justificativa

Muitas são as abordagens de comparação entre detectores de características ultimamente. Uma análise de desempenho entre descritores locais aponta a metodologia, bem como a métrica utilizada nesse trabalho ([MIKOLAJCZYK; SCHMID, 2005](#)).

Outra análise comparativa feita com três descritores binários (ORB, BRIEF e BRISK) utilizando os mais conhecidos detectores (ORB, MSER, SIFT, SURF, FAST e BRISK) é realizada comparando-se o efeito de transformações geométricas e fotométricas ([HEINLY *et al.*, 2012](#)).

Também um estudo comparativo de desempenho apresenta resultados para FAST-SIFT comparando variações de *blur*, iluminação, escala e rotação ([EL-GAYAR; SOLIMAN, 2013](#)).

Um estudo do desempenho de vários descritores foi feita no contexto de navegação robótica, decidindo pelo melhor detector e descritor para navegação de robôs ([SCHMIDT *et al.*, 2012](#)).

O desempenho dos detectores (FREAK vs. SURF vs. BRISK) é examinado no contexto de detecção de pedestres ([SCHAEFFER, 2013](#)). Há também estudos sobre descritores locais, descrevendo e comparando os principais descritores quanto à suas peculiaridades ([TUYTELAARS; MIKOLAJCZYK, 2007](#)).

Uma análise teórica de desempenho de sete combinações de conhecidos detectores e descritores, a saber, SIFT com SIFT, SURF com SURF, MSER com SIFT, BRISK com FREAK, BRISK com BRISK, ORB com ORB e FAST com BRIEF mostra uma abordagem de seleção da técnica adequada, concluindo que ORB com ORB e MSER com SIFT são preferíveis para quase todas as abordagens ([ISIK; OZKAN, 2014](#)).

O presente trabalho apresenta a comparação de nove combinações de detectores e descritores, a saber, BRISK com BRISK, STAR com BRIEF, MSER com SIFT, ORB com ORB, FAST com BRIEF, FREAK com FREAK, GFTT com BRIEF, SURF com SURF e SIFT com SIFT e propõe um estudo de caso no ambiente aeronáutico com um método pré estabelecido de seleção de limites restritivos com janelas decisivas de tal forma a eliminar as combinações que não estejam dentro da região desejada.

1.4 Contextualização

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 para que a aplicação seja utilizada pelo usuário com uma taxa aceitável, garantindo assim uma experiência ;

Qualidade do reconhecimento de objetos para que sejam encontrados pela técnica com resultados comparáveis a um mecânico;

Invariância a parâmetros ambientais para que seja possível emular situações reais do dia a dia como alteração entre momentos do dia, ambientes esfumados, etc ...

1.4.1 Cenário

O uso da realidade aumentada em manutenção de aeronaves pode trazer ganho no fornecimento de informações de procedimentos, na previsão de falhas ou no reconhecimento de regiões com falha.

Como caso de uso será adotado a janela de inspeção frontal, como mostrado na Figura 1.1, localizada na aeronave Embraer ERJ-190.



FIGURA 1.1 – Posicionamento da janela de inspeção. Fonte <http://www.aero-news.net/>

1.4.2 Variáveis de contorno

O cenário de reconhecimento de objetos dentro da aeronave traz alguns desafios que devem ser contornados:

- Pouca iluminação em ambientes internos;
- Objetos muito parecidos entre si;
- Alguns objetos com textura;
- Objetos brilhantes.

1.5 Recursos e Métodos

Com uma abordagem pragmática, temos a tabela 1.1, que exemplifica para cada um dos objetivos os recursos utilizados e os métodos com os quais serão atingidos.

TABELA 1.1 – Resumo de Recursos e Métodos

Objetivo	Recurso	Método
Avaliar os algoritmos clássicos de reconhecimento	Artigos, Livros e Sites	Pesquisa Bibliográfica
Aplicar os algoritmos clássicos à situações reais	Recursos fornecidos pelo OpenCV	Simulações
Selecionar algoritmo mais adequado para o contexto	Dados coletados de simulação	Análise de cadência e qualidade de reconhecimento

1.6 Estrutura do Texto

Este trabalho fundamenta-se em 5 capítulos, conforme descritos abaixo:

- Capítulo 1 apresenta a motivação do presente trabalho, levantando as necessidades e limitações impostas pelo ambiente, o fato de haver diversas técnicas de reconhecimento de características, a necessidade de selecionar a adequada para o contexto, bem como descreve o escopo do trabalho e o contexto em que os testes são feitos;

- Capítulo 2 tem informações suficientes para o entendimento das análises, descrevendo conceitos básicos e técnicas de reconhecimento que foram comparadas;
- Capítulo 3 descreve a metodologia de análise adotada bem como o protótipo desenvolvido;
- Capítulo 4 descreve os resultados obtidos comparando as técnicas, selecionando qual a mais adequada para reconhecimento de características para o caso de uso descrito;
- Capítulo 5 conclui o trabalho apresentando como os objetivos foram atingidos e possíveis trabalhos futuros.

2 Fundamentação Teórica

Para compreender a presente tese, se faz necessário elucidar fundamentos básicos, fornecendo assim, subsídios necessários para o leitor avaliar com mais propriedade os resultados apresentados. Os conceitos seguem uma ordem crescente de conhecimento para que seja construído um arcabouço de conhecimento suficiente para avaliar os resultados obtidos. O capítulo se inicia com uma explanação sobre realidade virtual e realidade aumentada e elucida o uso de dispositivos para imersão. Para a realização de aplicações de realidade aumentada, o uso de sensores para abstrair informações do ambiente é fundamental, sendo as câmeras de baixo custo os mais comuns, portanto, são descritos no capítulo os problemas de distorção inerentes a tais dispositivos e a modelagem matemática adotada para as câmeras utilizadas em todas as abordagens presentes nesse trabalho. A abordagem aqui utilizada é de reconhecimento de características locais, diferente do conceito comum de reconhecer padrões como retângulos, círculos, ou contornos, por isso conceituar o reconhecimento de características se faz tão importante. É também apresentada a conceituação básica de cada um dos algoritmos e parâmetros utilizados para a análise dos resultados.

2.1 Realidade Virtual

A Realidade Virtual é uma "interface avançada do usuário" para acessar aplicações executadas no computador, propiciando a visualização, movimentação e interação do usuário, em tempo real, em ambientes tridimensionais gerados por computador, como mostra a Figura 2.1. O sentido da visão costuma ser preponderante em aplicações de realidade virtual, mas os outros sentidos, como tato, audição podem ser usados para enriquecer a experiência do usuário (KIRNER; SISCOUTO, 2007).

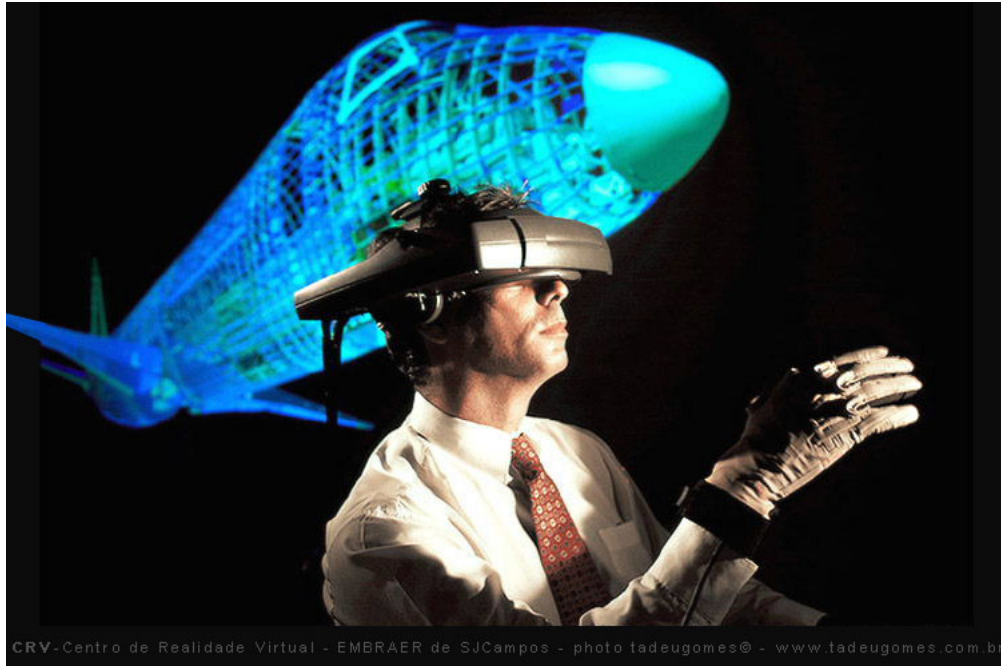


FIGURA 2.1 – Aplicação de realidade virtual. Fonte([KIRNER; SISCOUTO, 2007](#))

2.2 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 ([AZUMA, 1997](#)). Idealmente o mundo virtual se torna imersivo o suficiente para que o usuário não consiga distinguir o real do virtual. Alguns autores definem AR como tendo a necessidade de utilizar-se de interfaces visuais portáteis, como por exemplo celulares ou tablets, para que a usabilidade tenha mais coerência com a proposta inicial de garantir uma experiência imersiva. As imagens são obtidas por câmeras e o resultado apresentado em dispositivos como projetores ou displays como monitores, *tablets* ou *head-mounted display* (HMD). Realidade aumentada pode ser realizada com ou sem marcadores para facilitar o reconhecimento e posicionamento de entidades. No presente trabalho é utilizada a abordagem sem marcadores para que aplicações no cenários de manutenção se torne mais flexível no que tange à aplicabilidade e configuração inicial, não sendo necessário modificar o ambiente. A Figura 2.2 apresenta um processo básico de realidade aumentada. Esta tese trata de aspectos até a etapa de reconhecimento, que são representadas por:

- **Captura:** Etapa de obtenção de imagens, feita por sensores como câmeras;
- **Preparação:** Etapa de preparação da Figura, aplicando filtros preparando para a etapa de detecção;
- **Detecção:** Etapa de detecção de padrões, em que são removidas informações das imagens;
- **Reconhecimento:** Etapa de reconhecimento de padrões e comparação das características reconhecidas na etapa anterior;
- **Rastreio:** Etapa que garante que a Figura reconhecida continua no contexto, apesar de movimentações ou outras variações;
- **Apresentação:** Etapa em que são desenhadas na tela representações dos objetos reconhecidos de acordo com a aplicação.

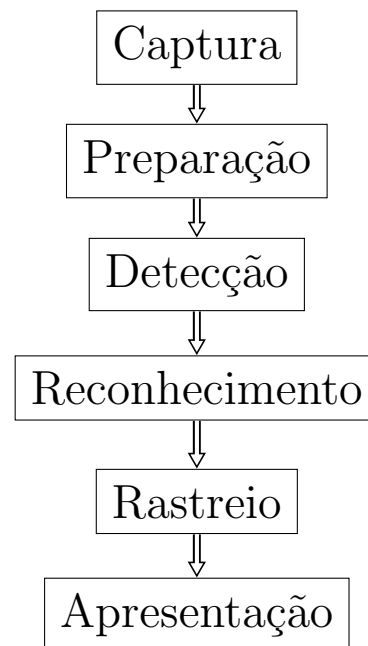


FIGURA 2.2 – Processo de Realidade Aumentada

2.3 Cadência

É a medida do número de quadros individuais que um determinado dispositivo óptico ou eletrônico processa e exibe por unidade de tempo. Em geral a cadência é medida em

fps. Em cinema, a cadência de projeção padrão desde 1929 foi fixada em 24fps, sendo no período do cinema mudo a maioria dos filmes eram rodados com cadência entre 16 e 20fps. Em vídeo, os principais sistemas lidam com cadência entre 25fps(PAL) e 30fps(NTSC). As aplicações devem ter cadência tolerável dependendo de seu uso, segundo (TANG *et al.*, 1993) para aplicações interativas o mínimo tolerável é de 5fps enquanto para aplicações de animações fluidas de 30fps. Sendo a cadência a frequência entre frames, deve ser contabilizado o tempo de geração de informações e o tempo de dispor a informação no dispositivo óptico. O tempo de cada frame é calculado o inverso do número de fps como mostrado na equação 2.1

$$t_{frame} = \frac{1}{fps} \quad (2.1)$$

No caso de cadência mínima de 5fps, temos quadros com tempo menor que 200ms, portanto as análises devem ser balizadas a tempos menores.

2.4 Filtros Lineares

Filtros são técnicas utilizadas para modificar or melhorar uma imagem, sendo responsável por enfatizar ou mesmo remover algumas características (FORSYTH; PONCE, 2012). Em geral os filtros são operações realizadas pixel a pixel utilizando informações dos pixels vizinhos para realizar transformações pontuais. Tais operações podem ser modeladas com operações básicas como somas ou subtrações considerando que a imagem em questão é uma matriz de pixels em que sua cor e intensidades atuais são considerados seu peso e seu valor é calculado somando-se o peso de pixels ao seu redor. Diferentes formas de considerar sua vizinhança ou mesmo formas de somar os valores de seus pesos. Por exemplo, podemos somar os pesos de tal forma a adicionar valores maiores no centro do que nas extremidades, modelando o tipo de suavização que ocorre em sistemas de lentes desfocadas. O resultado do processo depende muito mais do padrão da soma realizada do que da posição do pixel em torno do qual será feita a operação, sendo uma transformação linear no que diz respeito ao resultado obtido, pois o resultado da soma de duas imagens cujo filtro foi aplicado é o mesmo que aplicar o filtro separadamente em duas imagens e somá-las posteriormente. Colocando as nomenclaturas em padrões dormais, o padrão

de soma utilizado no filtro linear é chamado de **kernel** e o processo pelo qual o filtro é aplicado, chamado de **convolução**. Em particular dado um filtro kernel H , a convolução com a imagem F é a imagem R como se segue na equação 2.2, sendo i, j as coordenadas da imagem.

$$R_{ij} = \sum_{u,v} H_{i-u,j-v} F_{u,v} \quad (2.2)$$

Os limites de aplicação do kernel geralmente não são definidos pois o seu limite nem sempre é o tamanho da imagem, dependendo do tipo do **kernel**.

A Figura 2.3 mostra um kernel de tamanho 3x3 utilizado em filtros de média em que cada pixel é calculado como uma média dos seus pontos vizinhos.

1	1	1
1	1	1
1	1	1

FIGURA 2.3 – Kernel para Convolução do filtro de médias com vizinhança 3x3

Entretanto o efeito que um filtro de médias aplica em uma imagem não é uma modelagem fiel ao efeito gerado por embaçamento de câmeras pois resulta em imagens com um embaçamento não uniforme, sendo perceptível as transições. Uma abordagem atenuações menos abruptas na intensidade das imagens é utilizar um decaimento exponencial partindo da intensidade máxima no centro e mínima nas bordas como mostrado na Figura 2.4, caracterizando os **filtros Gaussianos**.

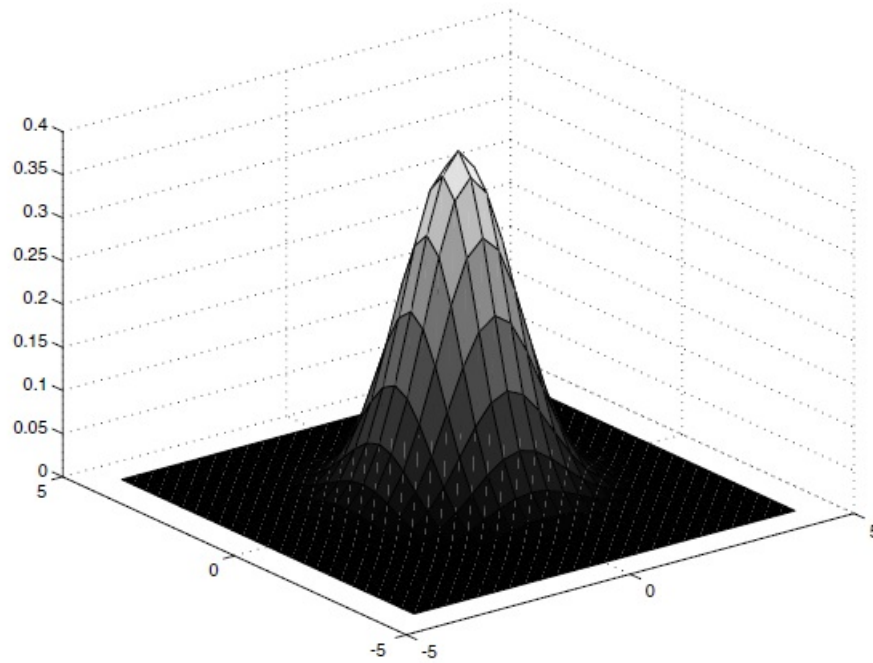


FIGURA 2.4 – Kernel para Convolução do filtro gaussianos simétrico. Fonte ([FORSYTH; PONCE, 2012](#))

O kernel do filtro gaussiano pode ser descrito pela equação 2.3

$$G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (2.3)$$

Entretanto como as imagens sob as quais o filtro será aplicado estão no espaço discreto, uma aproximação se faz necessária, como a apresentada na matriz da Figura 2.5.

	1	4	7	4	1
	4	16	26	16	4
$\frac{1}{273}$	7	26	41	26	7
	4	16	26	16	4
	1	4	7	4	1

FIGURA 2.5 – Aproximação discreta do kernel gaussiano. Fonte ([FORSYTH; PONCE, 2012](#))

2.5 Reconhecimento

É a etapa em que padrões são identificados e comparados para posteriormente identificar objetos (TUYTELAARS; MIKOLAJCZYK, 2007).

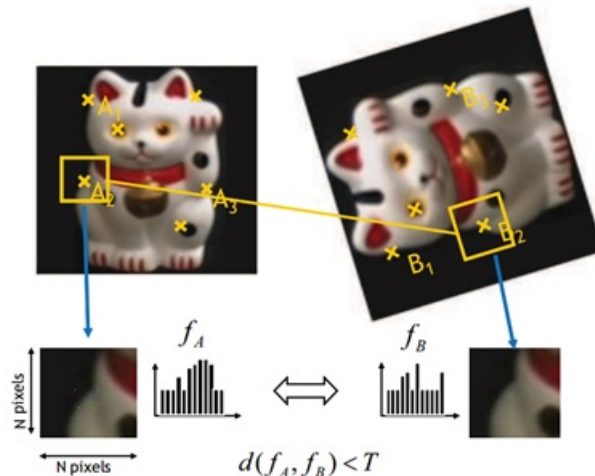


FIGURA 2.6 – Ilustração do procedimento de reconhecimento com características locais. Fonte (TUYTELAARS; MIKOLAJCZYK, 2007)

Processo de reconhecimento como ilustrado na Figura 2.6:

Encontrar um grupo de *keypoints* distintos

Etapa em que são encontrados pontos de interesse, selecionando-se, arestas, pontos ou vértices

Definir uma região em torno de cada *keypoint*

Em torno do ponto selecionado é delimitada uma janela de informação.

Extrair e normalizar o conteúdo da região

A região selecionada deve então ser extraída e abstraída, tornando-se em informação por meio de uma função

Calcular um descritor para a região normalizada

Uma vez a informação da região recuperada e normalizada é importante por meio de modelagem matemática transformar a informação em uma função capaz de ser comparada com outra região e decidir se a informação pertence ou não a um mesmo objeto.

Encontrar correspondências de descritores

A ultima etapa do reconhecimento, não abordada nessa dissertação, recupera todos os descritores e por meio de distâncias Hamming determinar se as informações podem ser consideradas iguais.

2.6 Características Locais

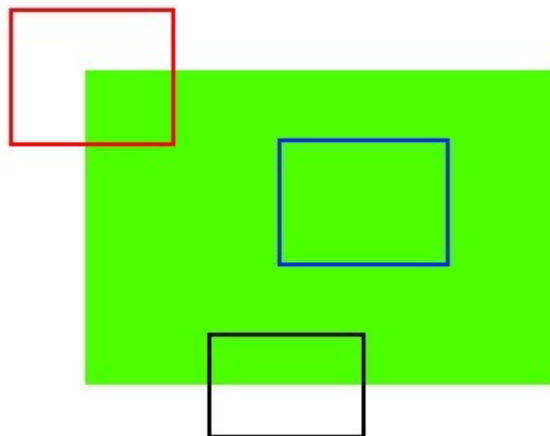
Características locais são padrões em imagens que diferenciam padrões de seu vizinho imediato, sendo que as mais comuns são intensidade, cor e texturas.

2.6.1 Características

Antes de compreender como é feito o reconhecimento e registro de imagens é importante nos perguntar como nós conseguimos reconhecer objetos em uma cena, como conseguimos comparar facilmente objetos em duas imagens distintas. Somos treinados desde cedo a diferenciar formas geométricas, perceber escalas diferentes ou mesmo reconhecer o mesmo objeto independente de como está posicionado na cena buscando padrões que categorizem e diferenciem o objeto. Instintivamente conseguimos reconhecer boas características e localizar objetos. Na Figura 2.7 temos uma Figura de um prédio e seis recortes, dos quais conseguimos facilmente reconhecer com precisão a de letra E e F, as de letra A,B,C,D podemos identificar possíveis localizações mas não podemos dizer com certeza onde estão na Figura.

FIGURA 2.7 – Reconhecimento de padrões. Fonte ([OPENCV, 2014](#))

As características E e F são o que consideramos boas características, pois o nível de certeza é bem alto.

FIGURA 2.8 – Regiões de reconhecimento de padrões. Fonte ([OPENCV, 2014](#))

A Imagem 2.8 ilustra tipos de características. A região azul não possibilita diferenciar onde está na Figura, a região preta pode ser confundida com qualquer uma das regiões

ao deslocarmos horizontalmente, a Figura vermelha nos possibilita diferenciar e reconhecer o canto da Figura verde com precisão milimétrica. Podemos então concluir que uma característica é boa para ser utilizada como parâmetro de entrada para algoritmos de reconhecimento, quanto maior foi o nível de certeza da sua localização, o que facilita o ***Feature Detection***. Para localizar o mesmo objeto em outra Figura é necessário identificarmos a região onde se encontra, caso contrário no exemplo da Figura 2.7 seria impossível localizar uma janela específica. Tal descrição de contexto é chamada de ***Feature Description***. Uma vez de posse da característica e do seu contexto é possível reconhecer o objeto de fato.

2.6.2 Propriedades da Característica Local ideal

Algoritmos de reconhecimento baseam-se em comparações de características recuperadas da cena. A recuperação e comparação de pontos tem um custo computacional relevante perto do tempo de execução da aplicação, portanto selecionar o menor número possível de características aumenta o desempenho e diminui o tempo de resposta da aplicação. Garantir que são selecionadas boas características pode ser crucial na eficácia do reconhecimento. Segundo (TUYTELAARS; MIKOLAJCZYK, 2007) , boas características devem ter as seguintes propriedades:

- **Repetibilidade:** Dadas duas imagens do mesmo objeto ou cena, tomadas em condições ou pontos de vista diferentes, uma porcentagem alta de características deve ser reconhecida se estiverem visíveis.
- **Distinção:** Os padrões reconhecidos têm de ser possíveis de serem distinguidos entre si para facilitar o casamento.
- **Localidade:** As características devem ser locais para reduzir a probabilidade de oclusão.
- **Quantidade:** O número de pontos detectados tem que ser o suficiente para que mesmo objetos pequenos tenham minimamente características que possam ser localizados e para que o objeto possa sofrer oclusão e ainda assim ser reconhecido.

- **Exatidão:** As características detectadas tem que ser localizadas com o máximo de exatidão possível com respeito tanto referente à posição quanto à escala.
- **Eficiência:** De preferência a detecção deve ser o mais rápido possível.

2.6.3 Detecção de características

O primeiro passo para o reconhecimento de objetos é a detecção de características. Para que possa ser feita a comparação da Figura referência com a Figura sentida, a abstração de informações a ser reconhecidas tem que ser suficiente para lidar com escalas diferentes, rotações entre as imagens e pequenas distorções. Para que o reconhecimento seja eficiente e eficaz é necessário que uma massa de pontos mínima seja selecionada e feita a correspondência entre as imagens sentidas e a Figura referência. As características utilizadas são em geral cantos, linhas, curvas, padrões ou regiões. O tipo de característica selecionada é dependente do tipo de Figura provida. Imagens de cenas feitas a mão geralmente são compostas de segmentos de retas enquanto imagens de satélite são geralmente compostas de contornos e regiões. Quanto mais invariantes, ou seja, que as variações não influenciem nos resultados encontrados, forem as características, mais robusto e preciso é o processo de comparação.

2.6.4 Correspondência de características

A fase de correspondência de características é feita tanto selecionando características na Figura referência e procurando a correspondente na Figura sentida ou mesmo selecionando características nas duas imagens independentemente e procurando a correspondência entre elas. Quando a característica selecionada não for do tipo ponto, é importante para cada par de correspondências que pelo menos um ponto seja determinado para que seja utilizado para determinar posteriormente os parâmetros de transformação. Por exemplo, se forem selecionados padrões como tipo de característica, o centro do padrão é considerado o ponto, se for selecionado uma região, o centro de massa da região é o ponto de apoio, se linhas forem tomadas como tipo de característica, devem ser tomadas intersecções como ponto de apoio e finalmente se forem selecionadas curvas, os máximos locais são considerados os pontos correspondentes.

2.7 Imagem Integral

Uma representação intermediária da Figura (VIOLA; JONES., 2001), pode ser calculada rapidamente a partir de uma Figura de referência. Dada uma Figura I e o ponto (x, y) , a Figura integral é calculada como a soma dos valores entre o ponto dado e a origem. Formalmente pode ser definida pela equação 2.4:

$$I_{\Sigma(x,y)} = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(x, y) \quad (2.4)$$

Utilizando a abordagem da Figura integral, a area pode ser calculada com quatro operações. Se considerarmos uma região retangular constituída pelos vértices A,B,C e D como na Figura 2.9, a soma dos pixels é aproximada pela equação 2.5:

$$\sum(x, y) = A + D - (C + B) \quad (2.5)$$

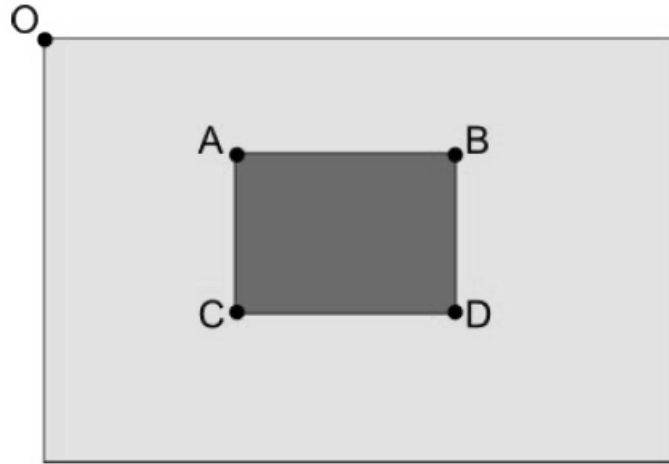


FIGURA 2.9 – Região da Figura integral. Fonte (VIOLA; JONES., 2001)

Como o gasto computacional é invariante ao tamanho da Figura, essa abordagem é particularmente útil quando são utilizadas imagens grandes.

2.8 Algoritmos de Reconhecimento

Existem diversos algoritmos de reconhecimento de características, entretanto nesse artigo os testes serão restritos aos algoritmos BRISK, FAST, FREAK, GFTT, MSER, ORB, STAR, SURF e SIFT por serem as técnicas mais utilizadas pelo meio de visão computacional no contexto de reconhecimento de características. Este capítulo se propõe a dar uma visão geral dos algoritmos, pois o foco do presente trabalho está na análise comparativa dos resultados obtidos no contexto e não nas implementações, visto existir implementações conceituadas no OpenCV, *framework* utilizado, apresentado na Seção 3.2.1.3.

2.8.1 FAST - *Features from Accelerated Segment Test*

FAST é um método de reconhecimento baseado em detecção de arestas originalmente desenvolvido por Edward Rosten e Tom Drummond ([ROSTEN; DRUMMOND, 2006](#)). A maior promessa do método é a eficiência computacional. O método considera um círculo de dezesseis pixels ao redor da aresta considerada p . O detector original ([ROSTEN; DRUMMOND, 2005](#)), classifica p como uma aresta se existirem n pixels contíguos em um círculo que são mais brilhantes do que o pixel candidato de intensidade I_p mais um threshold t ou mais escuros do que $I_p - t$

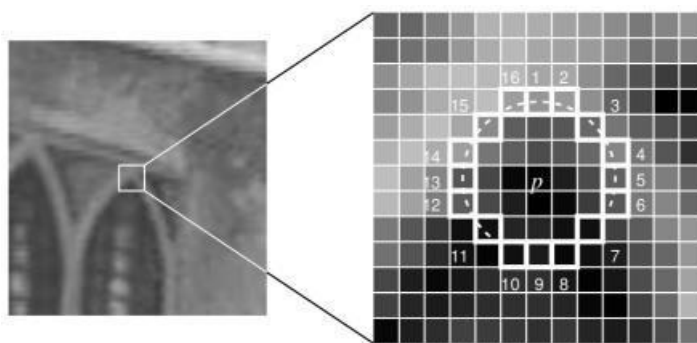


FIGURA 2.10 – Detector FAST. Fonte ([ROSTEN; DRUMMOND, 2006](#))

Na Figura 2.10 foi escolhido $n=12$ e seguido o seguinte algoritmo:

1. Selecionar o ponto e testar primeiro as extremidades. No caso da Figura, escolhido o ponto p

2. Comparar os pontos 1 e 9, e verificar se o ponto p tem intensidade com diferença de módulo t, ou seja os pontos 1 e 9 são mais claros ou mais escuros do que o ponto p pelo fator de t
3. Avaliar se o ponto p continua sendo um candidato considerando os pontos 5 e 13
4. Analisar se p é uma aresta, sendo que para isso, pelo menos 3 desses pontos devem ser mais brilhantes ou mais escuros do que p, para que então o teste pode ser feito nos demais pontos.

2.8.2 BRISK - *Binary Robust Invariant Scalable Keypoints*

Como descrito em (LEUTENEGGER *et al.*, 2011), ao contrário dos descritores vetoriais, como SURF e SIFT, BRISK utiliza descritores no espaço binário o que para dispositivos com restrição de recursos e poder computacional como dispositivos *mobiles* pode ser interessante. Baseado no detector da técnica FAST, o processo consiste de três partes.

Amostragem de padrões

Retirando um padrão de amostras ao redor do *keypoint* referentes a pontos espalhados em círculos concêntricos, que são usados para determinar se o ponto deve ou não ser selecionado como em um detector FAST. As amostras são separadas em pares de dois *subsets*, curta distância e longa distância.

Compensação de Orientação

Para atingir invariância à rotação, a direção de cada *keypoint* é determinada tomando a soma dos gradientes locais calculados entre pares de longa distância e rotacionando os pares de curta distância baseados na orientação obtida.

Comparação de pares de amostragem

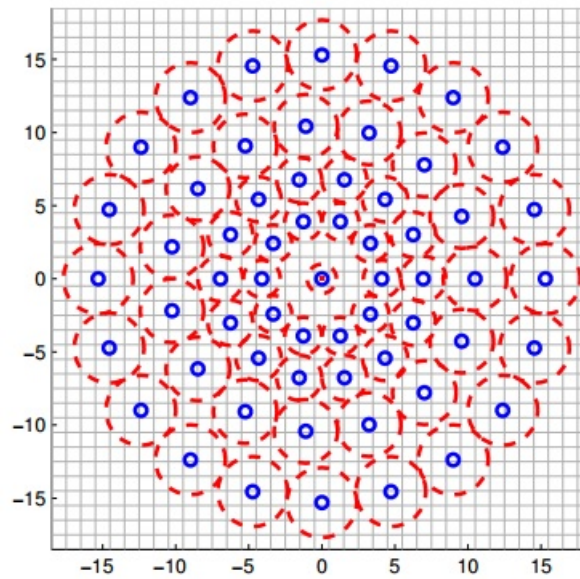


FIGURA 2.11 – Padrão de amostras no BRISK. Fonte (LEUTENEGGER *et al.*, 2011)

Como mostrado na Figura 2.11, BRISK possui um descritor binário de 512 bits que calcula a média ponderada utilizando uma Gaussiana sobre um padrão de pontos mais próximos do ponto selecionado como mostrado na Figura 2.11 no ponto (0,0). Ao longo do padrão descrito, são aplicadas suavizações Gaussianas sendo que os círculos vermelhos representam o tamanho do desvio padrão do filtro Gaussiano aplicado a cada ponto. É baseado em comparações entre pares de janelas de Gaussianas, resultando em 1 ou 0, dependendo de qual janela no par for maior. Os pares então são pré selecionado no BRISK, criando descritores binários que são posteriormente utilizados para casamento de padrão, utilizando distâncias Hamming.

2.8.3 FREAK - *Fast Retina Keypoint*

FREAK (ALAHY *et al.*, 2012) é um descritor binário, composto por três etapas:

Amostragem de padrão

Propõe uma abordagem biológica para o reconhecimento de características, emulando o funcionamento da retina para amostragem de padrões, como demonstrado na Figura 2.12, sendo um padrão circular com maior densidade de pontos próximo do centro, decrescendo exponencialmente.

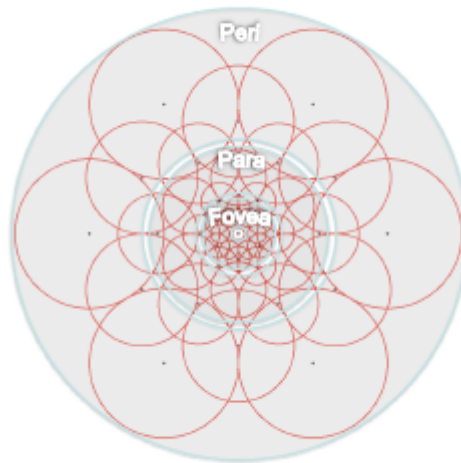


FIGURA 2.12 – Padrão de amostragem do descritor FREAK. Fonte (ALAHÍ *et al.*, 2012)

Cada amostra é suavizada por um filtro kernel Gaussiano em que o raio do círculo ilustra o tamanho do desvio padrão do kernel. Como pode ser observado na Figura 2.13, o padrão de amostragem corresponde à distribuição de receptores na retina.

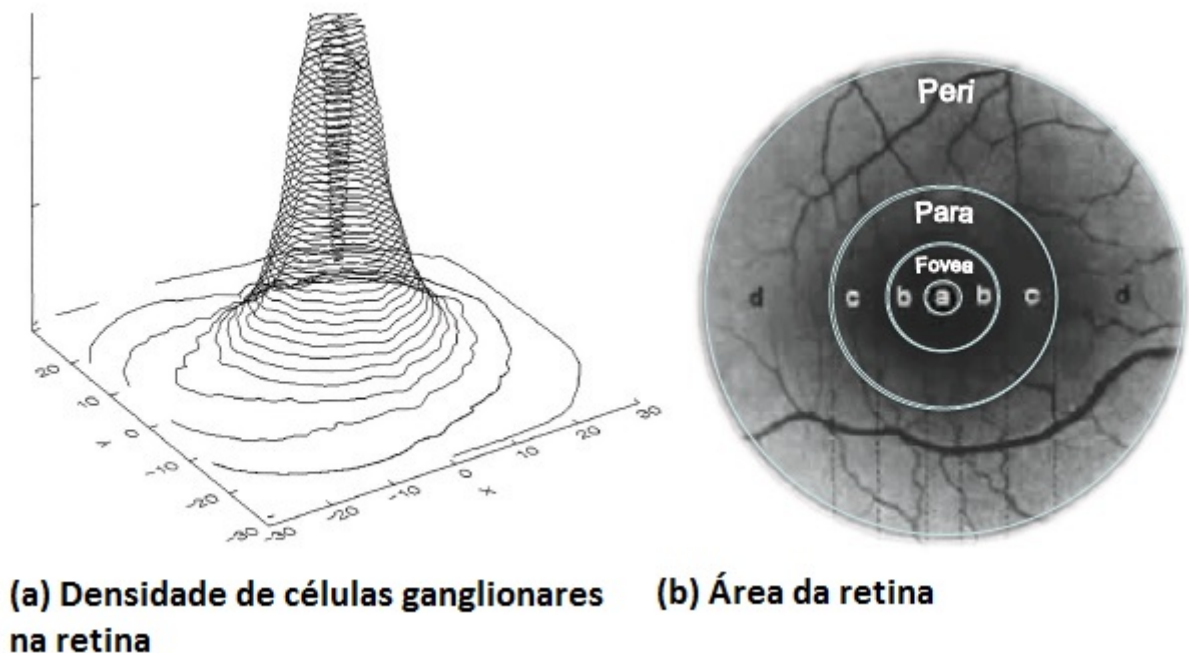


FIGURA 2.13 – Distribuição de receptores na retina. Fonte (ALAHÍ *et al.*, 2012)

Compensação de Orientação

Para estimar a rotação dos *keypoints*, são somados os gradientes locais assim como no

BRISK, entretanto ao invés de considerar os pontos de longa distância, é considerado um padrão de 45 pontos como mostrado na Figura 2.14

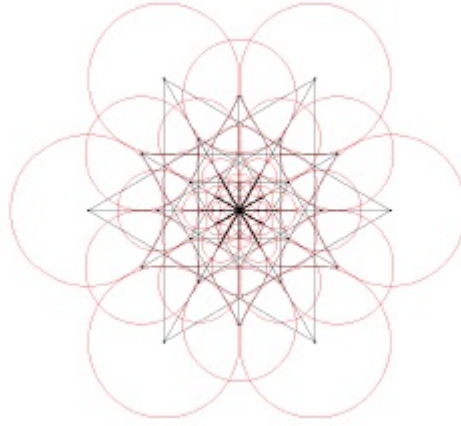


FIGURA 2.14 – Pares selecionados para calcular a orientação. Fonte (ALAHY *et al.*, 2012)

Apesar de ter menos precisão para recuperar informações de rotação, como o número de pontos, o descritor é bem menor do que BRISK e a quantidade de memória armazenada é em geral 5 vezes menor.

Comparação de pares de amostragem

Os pares de pontos são selecionados considerando a densidade maior no centro, como podemos observar na Figura 2.13 (a). Os pares começam a ser comparados pelas extremidades e para dentro do centro, dessa forma otimizamos o reconhecimento pois com menos pontos podemos descartar casos em que a distância estiver maior do que um *threshold*, caso contrário, devemos recuperar a informação integral do descritor e utilizar os outros 128 bits do descritor.

2.8.4 GFTT - *Good Features To Track*

O método é desenvolvido para detectar padrões sem bordas (SHI; TOMASI, 1994). A matriz de gradientes G é calculada para cada pixel como mostrado na equação 2.6

$$G = \sum_{\Omega} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (2.6)$$

Sendo o valor de intensidades $I(x, y)$ de uma Figura de tons de cinza e suas derivadas

parciais I_x, I_y de uma determinada região Ω . A matriz de gradientes é implementada por meio de uma Figura integral para I_x^2, I_y^2 e $I_x I_y$. Devido ao uso de imagens integrais, a complexidade computacional da matriz de gradientes é constante e independente do tamanho de Ω . Uma boa característica pode ser identificada pelo menor autovalor de G . Pontos fortes aparecem em geral nas bordas, onde problemas com movimento são mais comuns, o que leva para o problema de abertura geral. Pós processamento é aplicado por uma supressão de não máximos com um *threshold* em $q * \max(\lambda(x, y))$, em que q é uma constante para garantir a qualidade.

2.8.5 MSER - *Maximal Stable Extremal Regions*

O detector MSER é composto por regiões de todos os pixels conectados considerando um *threshold*. Em outras palavras, as regiões selecionadas são padrões que não mudam e que a binarização local é estável ao longo de uma faixa de *thresholds*. Podemos fazer uma analogia com o processo de formação de poças de água para compreendermos como são formadas as regiões MSER, que são descritas em imagens em tons de cinza representados pela função $I : \Omega \rightarrow [0 \dots 255]$ em que $\Omega = [1 \dots W] \times [1 \dots H]$. O método garante a localização de objetos que estão mais próximos de nossa realidade como pode ser observada na Figura 2.15 (MATAS *et al.*, 2002).



FIGURA 2.15 – Regiões MSER. Fonte (MATAS *et al.*, 2002)

Selecionado um *threshold* de intensidades, a Figura é dividida em dois grupos, P(pretos) e B(brancos). É observada que a quantidade de regiões varia dependendo do *threshold* aplicado, de 0 a 255. A área de cada componente conectado é então armazenada como uma função. Dentre as regiões, as mais estáveis são selecionadas analisando as funções para as quais cada região em potencial mantém seu estado com a mesma função independente da variação de *threshold*. As regiões “máximamente estáveis” são chamadas regiões MSER, que mudaram apenas em tamanho variando-se pelo menos alguns níveis de *threshold*.

2.8.6 ORB - *Oriented Fast and Rotated Brief*

Como citado em (RUBLEE, 2011), é uma combinação de FAST e BRIEF. Para extrair *keypoints*, modifica o detector do FAST construindo uma pirâmide de escalas das Figuras. Em cada escala *keypoints* são detectados e a distância de Harris aplicada para selecioná-los, sendo somente os N melhores pontos selecionados de acordo com um *threshold*. Para obter invariância à rotação, momentos de primeira ordem são utilizados para calcular a orientação local através de um centróide que referencia a média da magnitude dos pixels em um trecho local. Posteriormente são calculados os descritores BRIEF nos trechos rotacionados e armazenadas as informações em um descritor ORB.

2.8.7 SIFT - *Scale-Invariant Feature Transform*

SIFT (LOWE, 2004) é um método baseado em detecção de arestas tendo como proposta garantir a invariância à escala, um problema que nos métodos de reconhecimento de arestas se não tratado pode causar diminuição na robustez do algoritmo. A Figura 2.16 ilustra bem o efeito que a mudança de escala pode fazer.

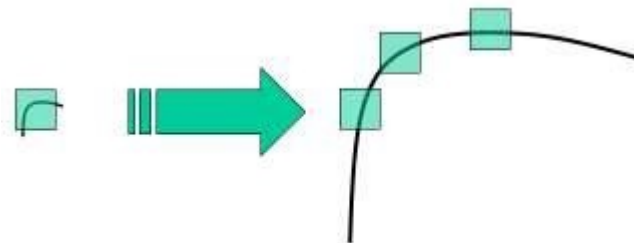


FIGURA 2.16 – Variação de escala. Fonte (LOWE, 2004)

Observando a Figura 2.16 é possível notar que não podemos utilizar a mesma janela de inspeção independente da escala do objeto. Para objetos maiores temos que utilizar janelas maiores. Nesse contexto, o filtro de espaço de escalas é utilizado e são calculados Laplacianos de Gaussianos com diversos valores de σ . Utilizar Laplacianos de Gaussianos é uma abordagem custosa computacionalmente, como uma forma de aproximação são utilizados diferenças de Gaussianas.

Uma vez que as diferenças de gaussianas são calculadas, é necessário procurar por máximos entre espaços de escala diferentes, como mostrado na Figura 2.17. Um pixel na

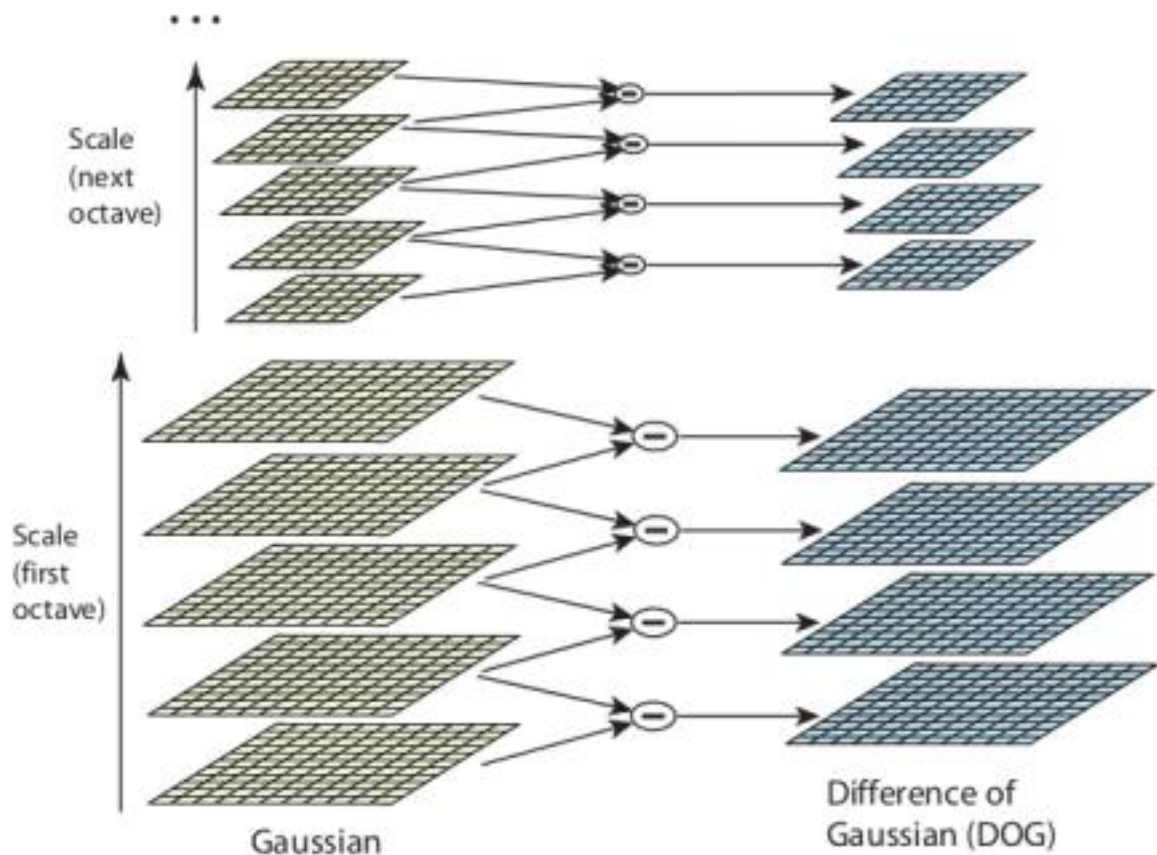
FIGURA 2.17 – *Difference of Gaussian*. Fonte (LOWE, 2004)

Figura é comparado com seus 8 vizinhos e com níveis de escala próximos e anteriores. O que representa que o *keypoint* encontrado é melhor representado naquela escala.

2.8.7.1 *Keypoint* Localization

Uma vez que *keypoints* potenciais são localizados, são selecionados pontos de interesse com contraste alto. A localização dos pontos é refinada utilizando uma expansão de Taylor e caso as intensidades dos máximos forem menores do que um *threshold*, rejeitados. Uma característica do DoG é a alta resposta a arestas gerando falsos positivos. Portanto é necessário eliminar algumas arestas identificadas erroneamente.

2.8.7.2 Orientation Assignment

Para obter descritores que são invariantes a rotação, um histograma de orientações é gerado a partir do gradiente de orientações de cada maximo local do DoG ao redor dos *keypoints*.

2.8.7.3 *Keypoint Descriptor*

O descritor é construído por um vetor de características, considerando a direção do *keypoint* cujo peso no gradiente é máximo. Como mostrado na Figura 2.18, tipicamente, uma região de adjacentes 16x16 é determinada partindo do *keypoint* no centro. Depois da região ser selecionada, SIFT divide em sub regiões de 4x4 com 8 representações de orientações em cada. Como obtemos $4 \times 4 = 16$ histogramas, cada um com 8 items, o vetor tem 128 elementos, o que torna o descritor significativo, compacto e ainda assim robusto o suficiente para mudanças de iluminação e variações de câmera.

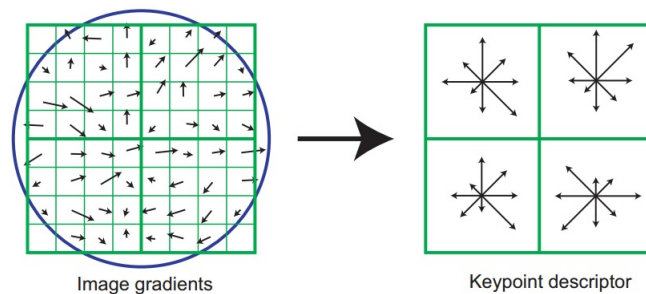


FIGURA 2.18 – Formação do descritor SIFT. Fonte (LOWE, 2004)

2.8.7.4 *Casamento de Padrões*

O casamento entre dois pontos é feito identificando pontos próximos. Entretanto em algumas situações existem pontos muito próximos que podem ser causados por ruídos na detecção de pontos de interesse, nesse caso é calculada uma razão de distância entre o ponto de interesse com o mais próximo, e com o segundo mais próximo, se a razão for maior do que 80%, são rejeitados. Tal abordagem elimina cerca de 90% de falsos positivos e 5% de pontos corretos.

2.8.8 *SURF - Speeded Up Robust Feature*

O descritor de características de SURF (BAY *et al.*, 2008) é bem parecido com o descrito em SIFT na Seção 2.8.7, sendo SURF desenvolvido para tentar suprir as principais fraquezas: alta complexidade computacional e baixa velocidade de execução. A detecção de SURF utiliza aproximações baseadas no determinante da matriz hessiana da equação 2.7

mostrado na equação 2.8.

$$H(f(x, y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \quad (2.7)$$

$$\det(H) = \frac{\partial^2 f}{\partial x^2} \frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 \quad (2.8)$$

Aproximando as segundas derivadas, sendo muito mais arrojado do que o algoritmo SIFT que aproxima Laplacianas de Gaussianas(LoG) por Diferenças de Gaussianas(DoG), aproximando LoG por *Box-type Filter*, como mostrado na Figura 2.19, não sendo utilizado nenhum tipo de suavização entre escalas, o que garante mais agilidade nos resultados porque as convoluções com *box filters* são muito mais rápidas com o uso de imagens integrais.

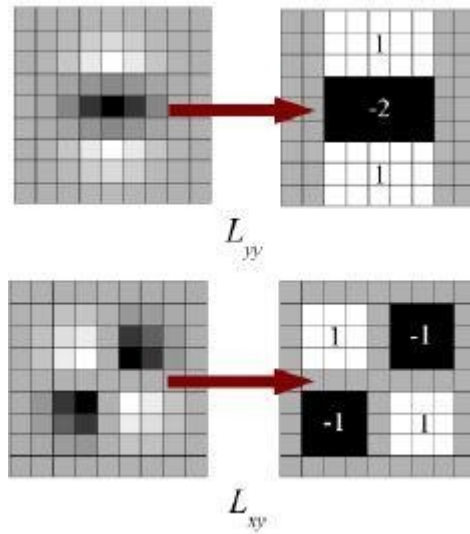


FIGURA 2.19 – *Box Filtering*. Fonte (BAY *et al.*, 2008)

O descritor usa imagens integrais para interpretar a distribuição de intensidade de pixels na vizinhança da característica detectada considerando a escala das características. O cálculo do descritor de uma característica em uma dada escala inicia adotando-se uma orientação dominante para fazer o descritor invariante à rotação.

O processo inicia calculando-se as respostas à *Haar wavelet* em duas direções dominantes para cada ponto em um raio próximo da característica, sendo que o tamanho da máscara adotada é ajustada de acordo com o tamanho da característica. Os resultados

são então pesados segundo uma gaussiana centralizada na característica encontrada. Cada uma das respostas é responsável por um ponto no vetor espacial, com x respostas nas abcissas e y respostas nas ordenadas. Em seguida, um segmento de círculo de $\frac{\pi}{3}$ é rotacionado em torno da origem da característica. A resposta do segmento é somada e formado assim um vetor de resultantes.

O ângulo de rotação corresponde à orientação dominante. O cálculo do descritor em si, inicia por posicionar uma janela retangular de tamanho $20s$ centralizada na posição da característica e orientá-la como foi calculado no passo anterior como pode ser observado no exemplo da Figura 2.20.

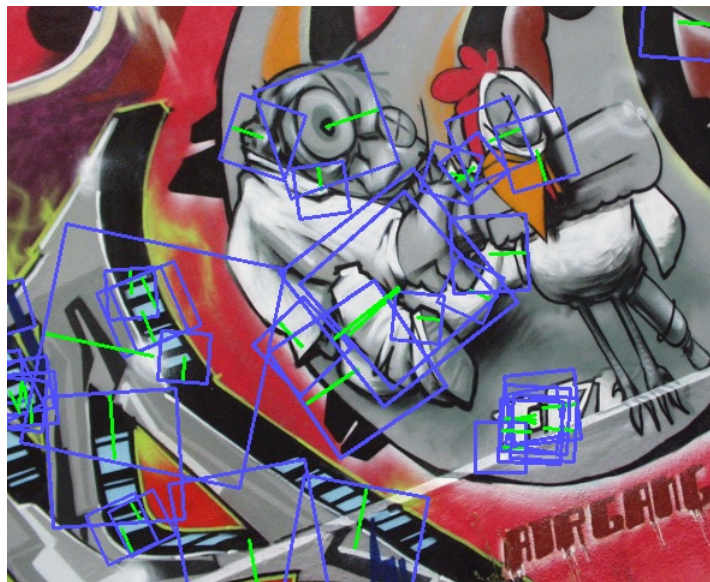


FIGURA 2.20 – Janela do Descritor. Fonte (BAY *et al.*, 2008)

A janela é então dividida em 4×4 sub regiões regulares. Cada sub-região é dividida em 5×5 pontos amostrais distribuídos uniformemente. Para cada ponto, Haar *wavelets* são calculados para as duas principais direções. Cada sub região contribui para o descritor com componentes: a soma das respostas nas duas principais regiões e seus valores absolutos. A resposta das 16 sub-regiões é então pesada com uma gaussiana. Para 16 sub-regiões, o tamanho do descritor é 64.

2.8.9 STAR

Derivado de CenSurE (*Center Surround Extrema*), STAR (AGRAWAL *et al.*, 2008), assim como SURF citado na Seção 2.8.8 é baseado em *box filters*. Entretanto, enquanto DoB

não é invariante à rotação, é introduzido o filtro *center surround* que são bi-level. A Figura 2.21 mostra o padrão do filtro bi-level em vários níveis, sendo o quanto mais circular, mais preciso, entretanto também mais difícil de se calcular.

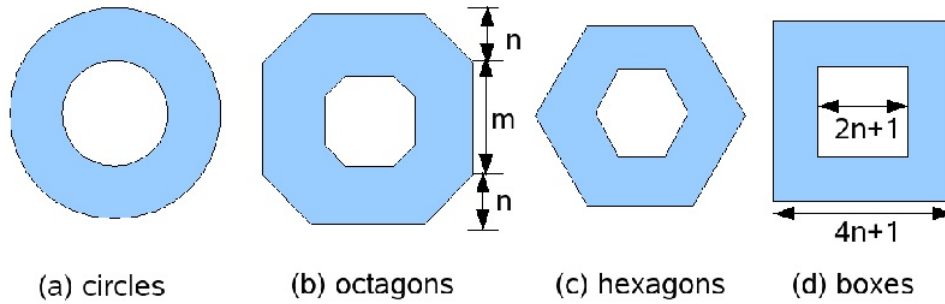


FIGURA 2.21 – Filtro bi-level aplicado à formas de n lados. Fonte ([AGRAWAL et al., 2008](#))

O detector de características de STAR em contrapartida à proposta do CenSurE, usa um filtro composto de dois quadrados rotacionados. A resposta do filtro é calculada para sete escalas em cada pixel da Figura. Em contraste com SIFT e SURF, o tamanho da amostra é constante em cada escala e tende à resolução total em cada escala. Um passo de pós processamento é realizado para suprimir os não máximos e as linhas. Características que estão ao longo de linhas são detectadas devido à matriz de gradientes, como apresentado na equação 2.6.

3 Aplicação e Seleção de Algoritmos

3.1 Metodologia

3.1.1 Definição de parâmetros

Os testes são realizados adicionando variáveis de forma artificial por meio de transformações afins, de forma a emular comportamentos ou situações encontradas no ambiente de manutenção:

- 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 ou esfumados.

A definição das faixas de parâmetros utilizadas foi feita com inspeção em campo.

3.2 Análise de dados

A análise de desempenho de cada técnica deve ser balizada por parâmetros que garantam que a aplicação seja viável para o contexto citado na Seção 1.4. Para tal, deve-se considerar os parâmetros estimados abaixo definidos por inspeção:

- Cadência $> 5\text{fps}$ ou seja, tempo $< 200\text{ms}$ ([TANG et al., 1993](#));

- 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*: $\text{kernelsize} < 4$

Uma análise mais precisa dos parâmetros não faz parte do escopo desse trabalho pois envolve pesquisa de campo em situações adversas. Portanto, tais parâmetros foram escolhidos para validar o método de seleção das técnicas, sendo assim, identificados por experiência como usuais para o contexto, entretanto, não há perda de generalidade no método, posto que pode ser executado da mesma maneira com outros valores.

3.2.1 Protótipo

O escopo de desenvolvimento dessa dissertação propõe traçar um método de escolha do algoritmo a ser utilizado dependente de aspectos descritos na Seção 1.4.2. Os testes são feitos avaliando os algoritmos mais conhecidos no contexto aeronáutico.

3.2.1.1 Testes de reconhecimento de padrões

Utilizando os parâmetros definidos na Seção 3.2, serão executados todos os algoritmos para avaliar qual se encaixa melhor em cada uma das situações. O protótipo para testes será realizado utilizando o OpenCV por já disponibilizar os algoritmos mais utilizados para o reconhecimento de padrões e de visão computacional. Nesse protótipo serão avaliados os algoritmos: BRISK, FAST, FREAK, GFTT, MSER, ORB, STAR, SURF e SIFT utilizando a Figura 3.1 nos testes



FIGURA 3.1 – Imagem utilizada no protótipo

Segundo os seguintes critérios:

Velocidade - para garantir que a aplicação poderá ser utilizada como uma ferramenta de tempo real, é imprescindível buscar uma aplicação que execute naturalmente a uma taxa aceitável. Tal realidade é facilmente atingida em computadores com vários núcleos como os core i7, mas é importante lembrar que o cenário de realidade aumentada, para ter um contexto coerente com a proposta de imersão, é composto por dispositivos portáteis que via de regra não têm poder de processamento tão alto.

Qualidade - características detectadas são geralmente utilizadas posteriormente em etapas de rastreamento e casamento de padrão e contém informações que combinadas garantem o sucesso do algoritmo, portanto informações relevantes e o mais precisas possíveis são importantes por proporcionar o reconhecimento com o menor número possível de características e portanto melhor desempenho.

Invariância à transformações - detectores de características devem ter a habilidade de reconhecê-las independente das seguintes transformações:

- **Rotação:** robustez à rotação também é uma característica importante, pois nem sempre encontraremos situações em que o dispositivo de AR é posicionada na mesma orientação o tempo todo, garantindo uma maior autonomia a quem executa a ma-

nutenção;

- **Brilho:** a variação de brilho pode emular tanto variações temporais, como dia e noite, mas também situações de muita luminosidade como neve;
- ***Blur*:** a robustez à *Blur* auxilia na resposta do algoritmo frente à movimentos bruscos;
- **Escala:** a robustez à escala do objeto garante que o mesmo é reconhecido tanto ao longe quanto em situações bem próximas.

3.2.1.2 Ambiente

Os testes têm por objetivo garantir as restrições descritas na Seção 3.2. Os resultados obtidos descritos nessa tese foram executados em uma máquina com a seguinte configuração:

Processador: Core 2-duo 2.2GHz

Memória: 4GB

Placa de Vídeo: NVidia GForce 9300M GS

Biblioteca: OpenCV 2.4.10

Sistema Operacional: Windows 7

3.2.1.3 OpenCV

Esta biblioteca é distribuída sob licença BSD e portanto, livre para uso acadêmico e comercial. Possui interfaces para C++, C, Python e Java suportando Windows, Linux, Mac OS, iOS e Android. OpenCV foi desenvolvido para resultados eficientes e com foco em aplicações de tempo real, podendo tomar vantagem de processamentos paralelos, utilizando-se de OpenCL, aproveitando-se da aceleração de *hardware* ou mesmo de plataformas heterogêneas. Adotado ao redor do mundo em várias pesquisas com uma comunidade de mais de 47 mil pessoas com mais de 9 milhões de *downloads*. Adotado desde plataforma de desenvolvimento para aplicações de arte interativa a aplicações de robótica avançada.

4 Análise de Resultados

A abordagem de testes e análise de desempenho é feita como descrito em (MIKO-LAJCZYK; SCHMID, 2005). Na Seção 3.2, existem restrições para garantir que a técnica utilizada responda segundo as necessidades da aplicação. O protótipo, apresentado na Seção 3.2.1, é instrumentado para auxiliar a análise de tempo gasto para reconhecer características a cada frame de execução e para fornecer informações da razão entre o número de características encontradas e o número de características boas encontradas, gerando a partir das informações, gráficos.

Sob os gráficos traçados, são adicionadas as restrições definidas na Seção 3.2, gerando janelas de restrição.

Os valores que se encontram nas regiões dentro da janela de restrição, ou seja, as que estiverem nas regiões não coloridas, são considerados adequados segundo cada um dos critérios selecionados.

As técnicas têm suas implementações no OpenCV com a possibilidade de configurar alguns parâmetros, entretanto para garantir que a comparação inicial seja independente das peculiaridades da implementação ou mesmo das diversas possibilidades de configuração, foi feita a comparação utilizando os construtores padrão, com as configurações iniciais.

4.1 Taxa de Acertos

Foi avaliada a razão entre o número de características corretamente localizadas e o número total de características na Figura inicial. Idealmente, o valor dessa medida deve se aproximar de 100%. Inicialmente, foram considerados os resultados que se demonstraram

dentro de pelo menos mais de 50% da janela delimitada, pois a análise deve ser feita considerando tanto o parâmetro de taxa de acertos quanto o tempo gasto para realizá-lo.

Como mostrado na Figura 4.1, para os testes de robustez à escala estão dentro da janela, as técnicas FREAK, ORB e GFTT. Apesar de todas as técnicas serem invariantes à escala, o que é possível notar que em torno da escala unitária, ou seja objeto em tamanho real, podemos dizer que aproximadamente 10% para mais e para menos, entretanto para variações maiores nem todas as abordagens respondem bem à variação de escala, o que no contexto da manutenção significa que uma aproximação brusca do objeto em questão não garanta uma boa resposta o que provavelmente resultaria em perda de reconhecimento.

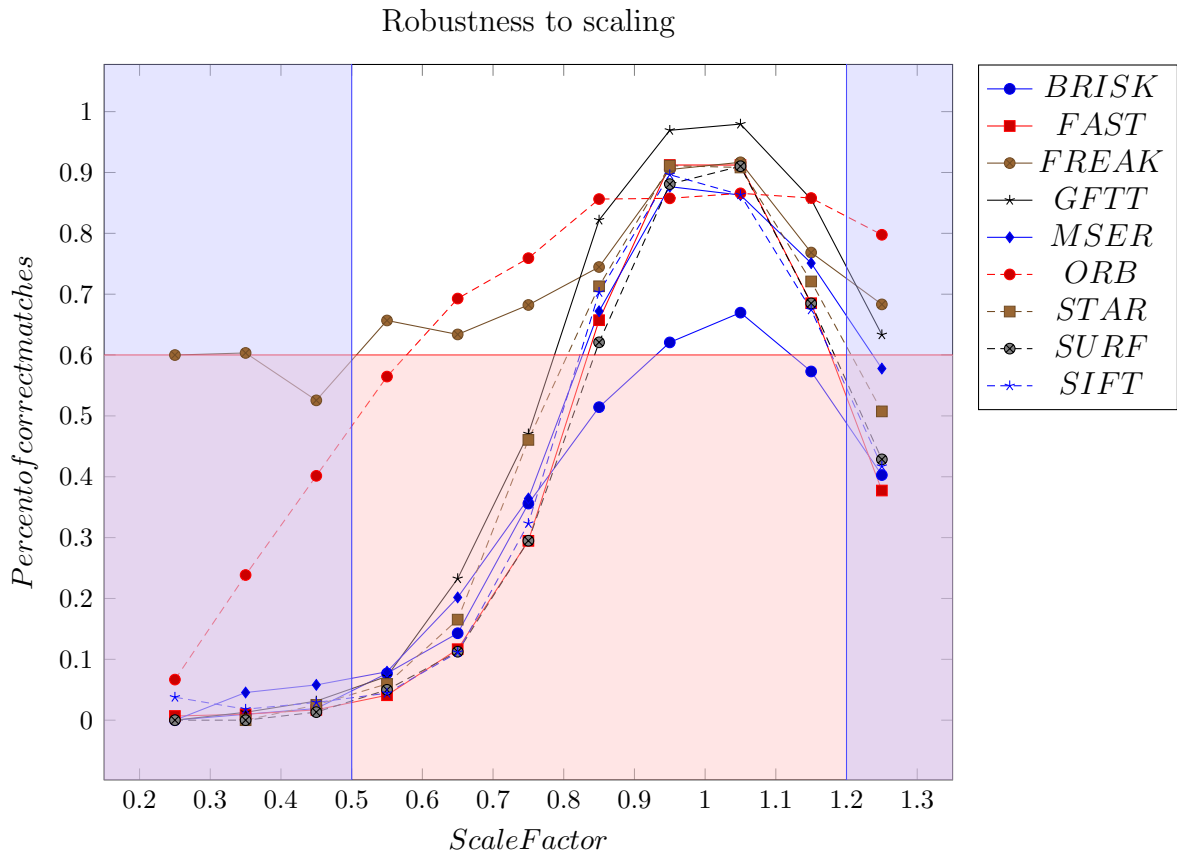


FIGURA 4.1 – Resultado de performance de variação de escala

Como mostrado na Figura 4.2, para os testes de Rotação estão dentro da janela as técnicas BRISK, FAST, FREAK, GFTT, MSER, ORB, STAR, SURF e SIFT. O que demonstra que as informações de orientação armazenadas pelos descritores são suficientes para garantir robustez à rotação dentro da faixa esperada, sendo assim no contexto de manutenção, garantido que a movimentação de dispositivos móveis ou mesmo vibrações inesperadas não tragam perda de reconhecimento do objeto em questão.

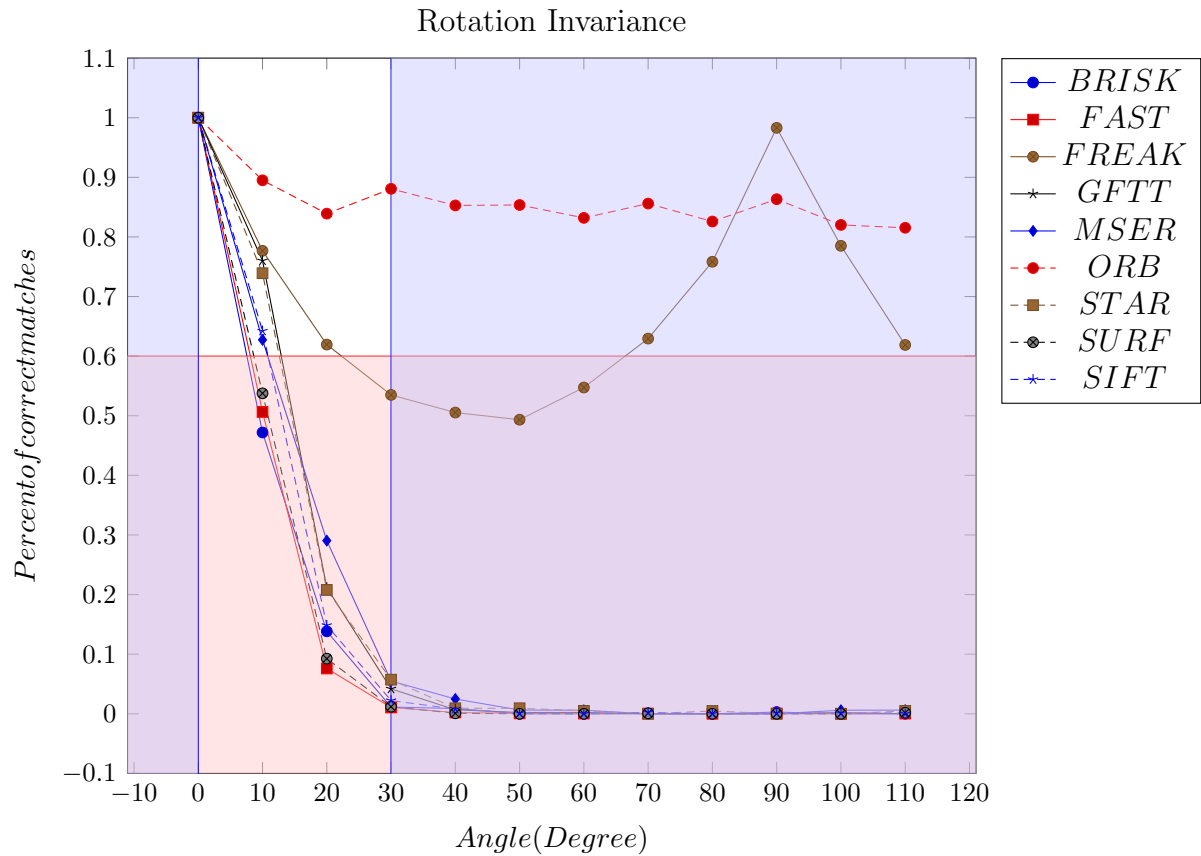
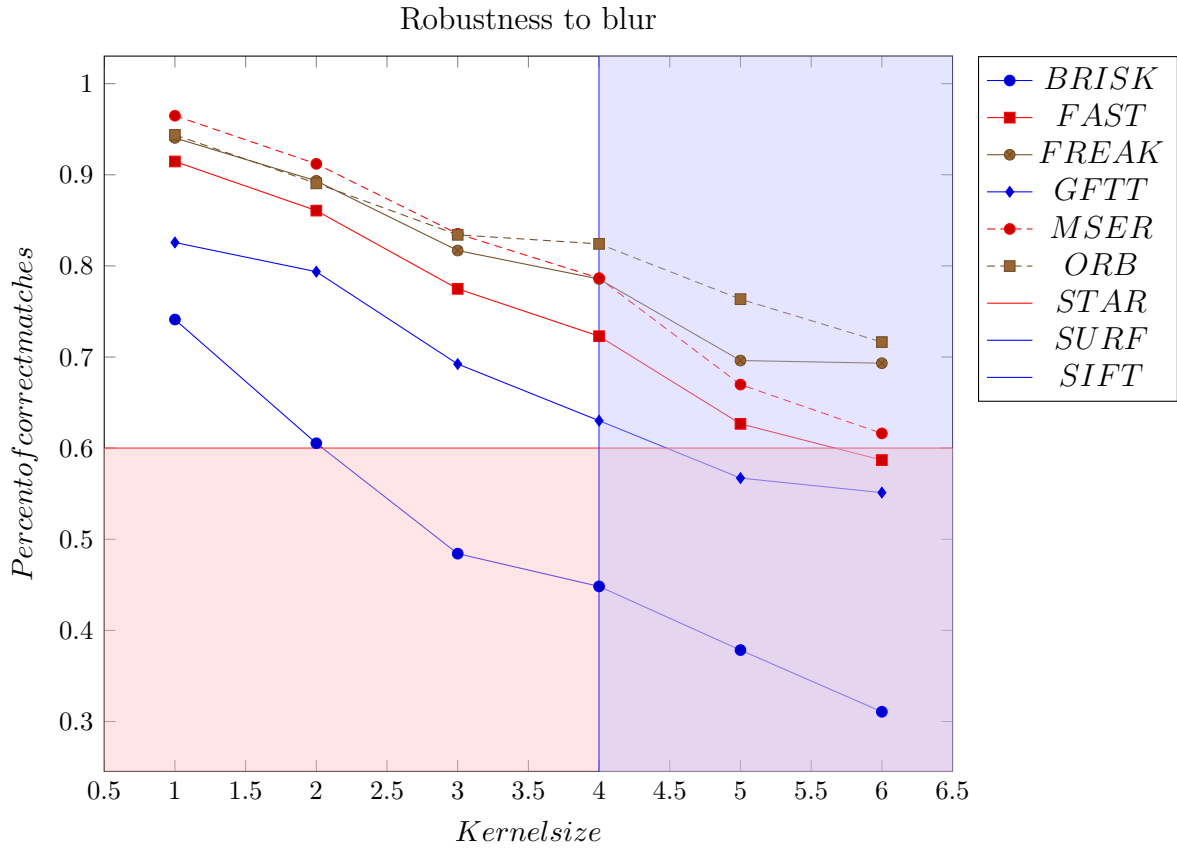


FIGURA 4.2 – Resultado de performance de variação de rotação

Como mostrado na Figura 4.3, para os testes de Blur estão dentro da janela as técnicas FAST, FREAK, GFTT, MSER, ORB, STAR, SURF, SIFT. A técnica BRISK tem um decréscimo acentuado da taxa de acerto frente à *blur* pois seu descritor é baseado em gaussianas em torno de um ponto central, comparando as intensidades com a região no entorno, portanto adicionando o parâmetro de *blur* geral os pontos próximos se tornam cada vez mais homogêneos e portanto descartados segundo o algoritmo de detecção.

FIGURA 4.3 – Resultado de performance de variação de *blur*

Como mostrado na Figura 4.4, para os testes de Iluminação estão dentro da janela as técnicas BRISK, FAST, FREAK, GFTT, MSER, ORB, STAR, SURF, SIFT. O que demonstra que para o contexto de manutenção, sob os valores estipulados na Seção 1.4.2 todas as técnicas garantem a variação de iluminação sem perda significativa da taxa de acertos.

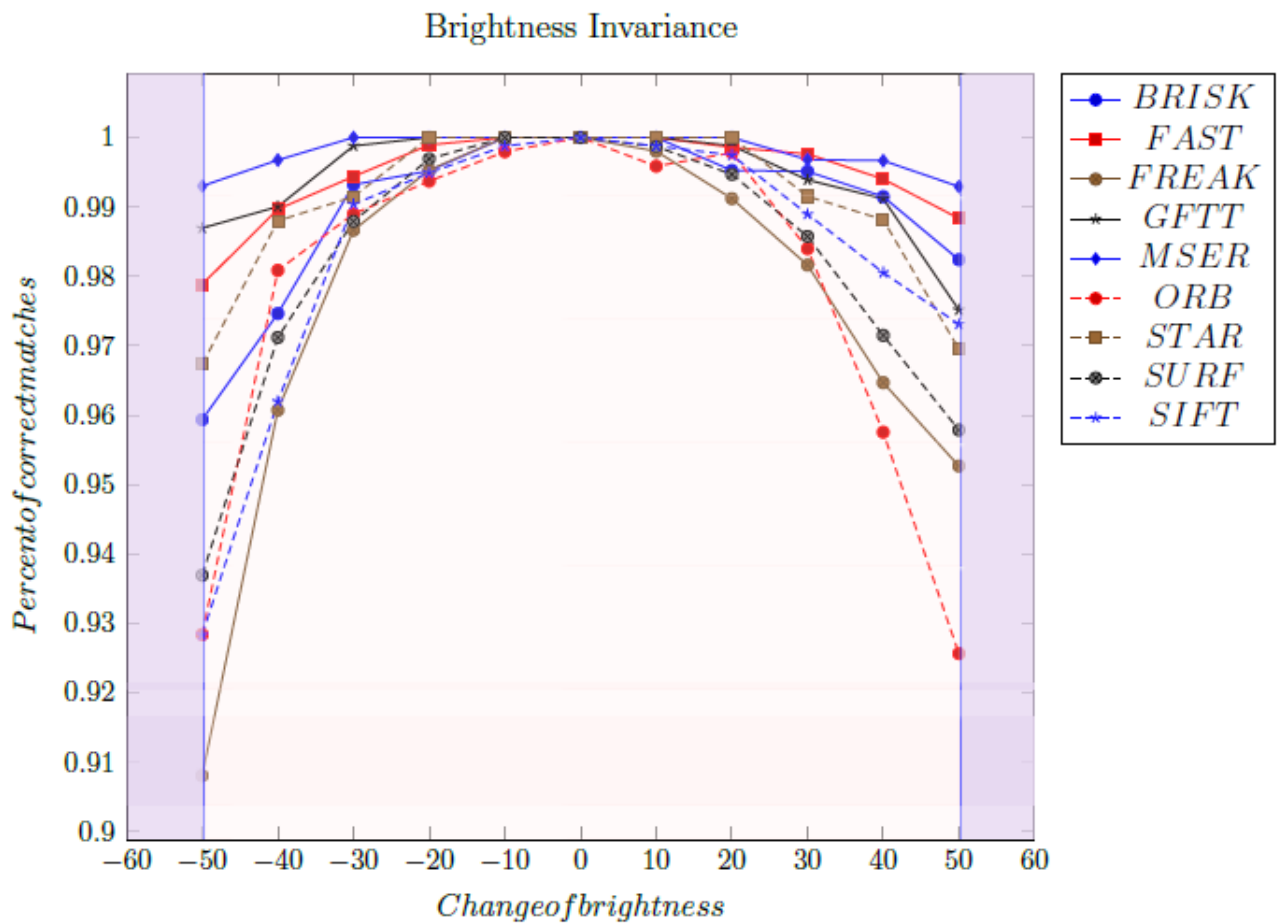


FIGURA 4.4 – Resultado de performance de variação de iluminação

4.2 Análise de Tempo

A análise de tempo, ao contrário da análise de taxa de acertos, é excludente, pois como descrito em (TANG *et al.*, 1993), os usuários não toleram aplicações com menos do que 5fps.

Como mostrado na Figura 4.5, para os testes de Escala, estão dentro da janela, as técnicas BRISK, GFTT, ORB e STAR. Os algoritmos SIFT e SURF apesar de serem os mais utilizados no contexto de visão computacional, não tem desempenho esperado. FAST tem o pior resultado e notoriamente segundo a comunidade OpenCV não tem uma implementação otimizada. Entorno do ponto de escala unitária, ou seja do objeto em questão em tamanho real, os algoritmos BRISK, GFTT, ORB e STAR não variam muito, em contrapartida, FAST, FREAK, MSER, SURF e SIFT além de variar muito em torno

do ponto de escala unitária, tem variação acentuada em todas as escalas, portanto não se mantém na faixa de valores aceitáveis.

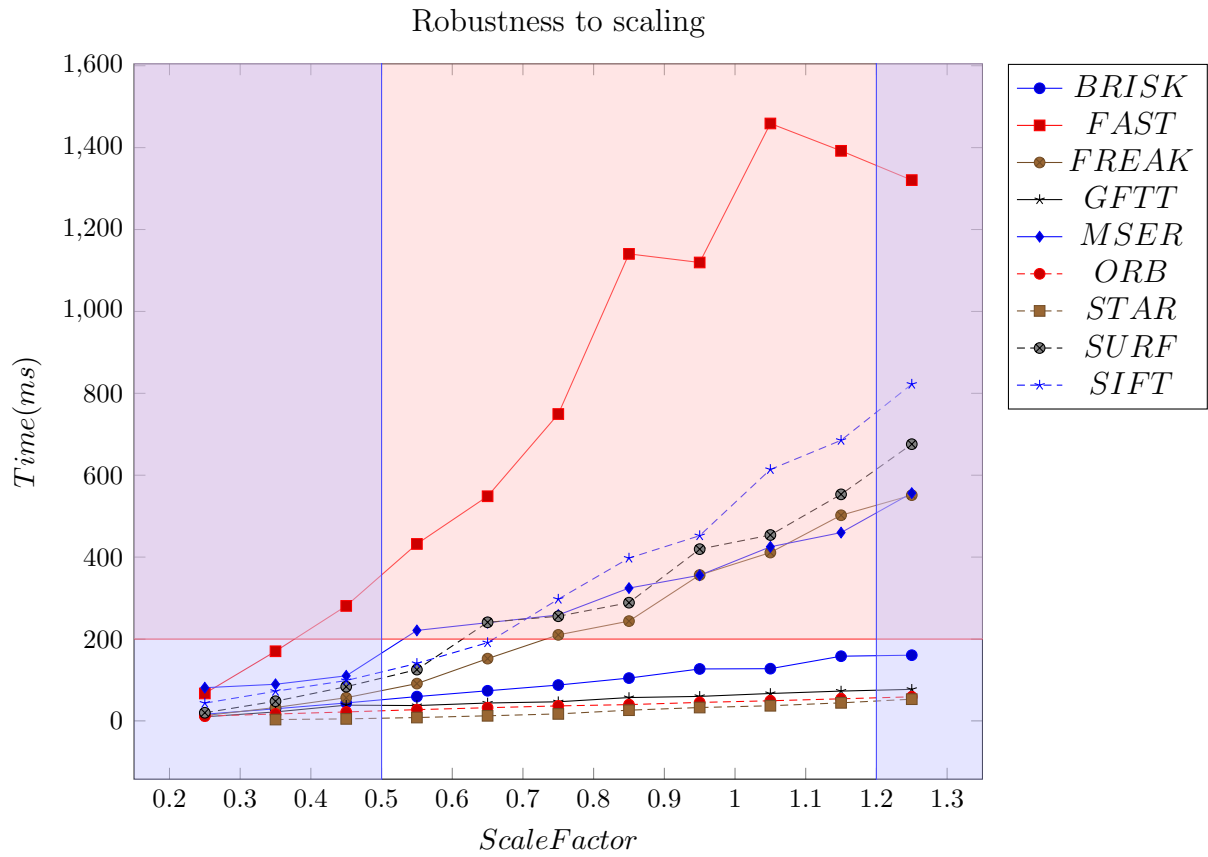


FIGURA 4.5 – Análise de tempo de variação de escala

Como mostrado na Figura 4.6, para os testes de Rotação estão dentro da janela, as técnicas BRISK, GFTT, ORB, STAR. Apesar de as técnicas FAST, SIFT, SURF, FREAK e MSER apresentarem resultados satisfatórios no quesito taxa de acertos, tem um desempenho abaixo do esperado, não conseguindo recuperar as características em tempo hábil.

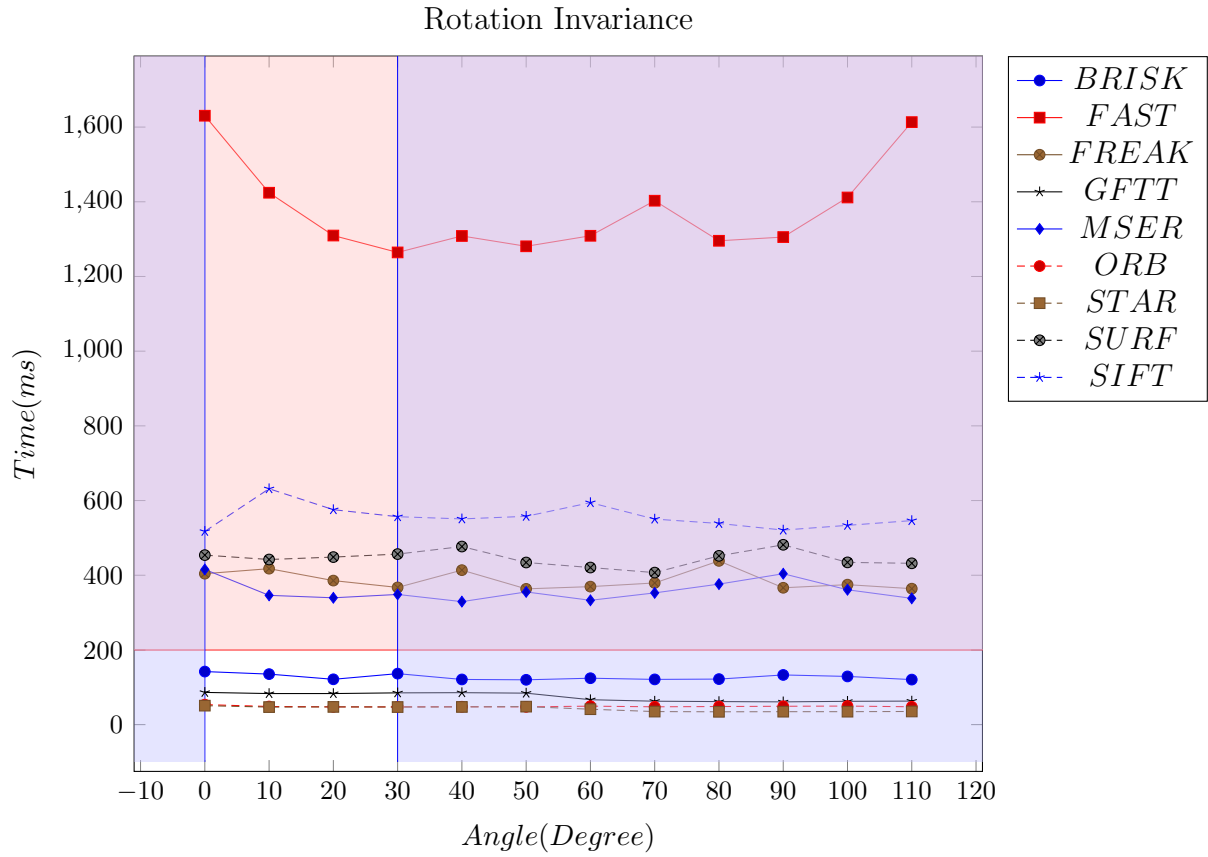


FIGURA 4.6 – Análise de tempo de variação de rotação

Como mostrado na Figura 4.7, para os testes de *Blur* estão dentro da janela, as técnicas BRISK, GFTT, ORB, STAR. FAST e SIFT demonstram ser muito sensíveis à *blur* e as técnicas FAST, SIFT, SURF, MSER e FREAK ficam fora de toda a janela de restrições. O algoritmo mais eficiente nesse quesito é o ORB pois tem sua implementação otimizada para lidar com parâmetros de orientação.

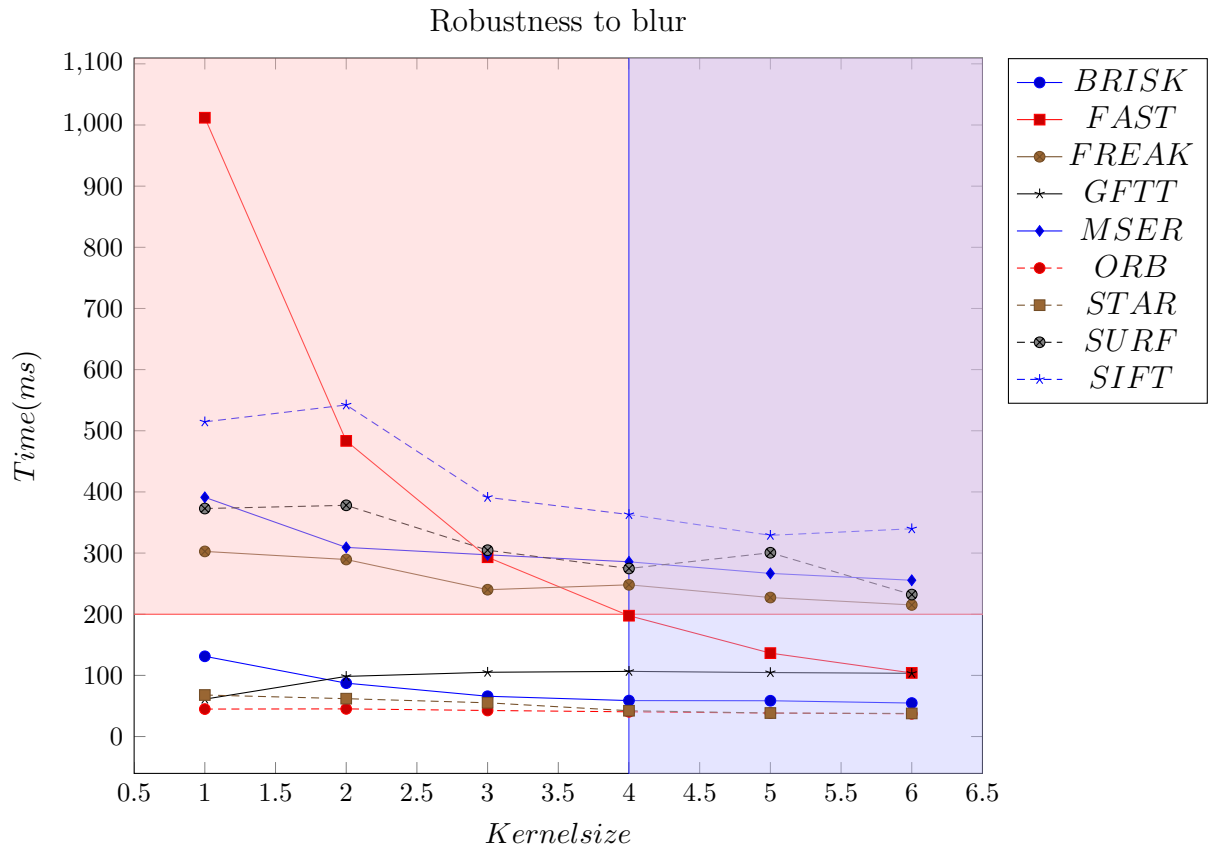


FIGURA 4.7 – Análise de tempo de variação de blur

Como mostrado na Figura 4.8, para os testes de Iluminação estão dentro da janela, as técnicas BRISK, GFTT, ORB, STAR. Apesar de todas as técnicas apresentarem resultados satisfatórios para o quesito variação à brilho, FAST, SIFT, SURF, FREAK e MSER tem um desempenho muito inferior à BRISK, GFTT, ORB e STAR.

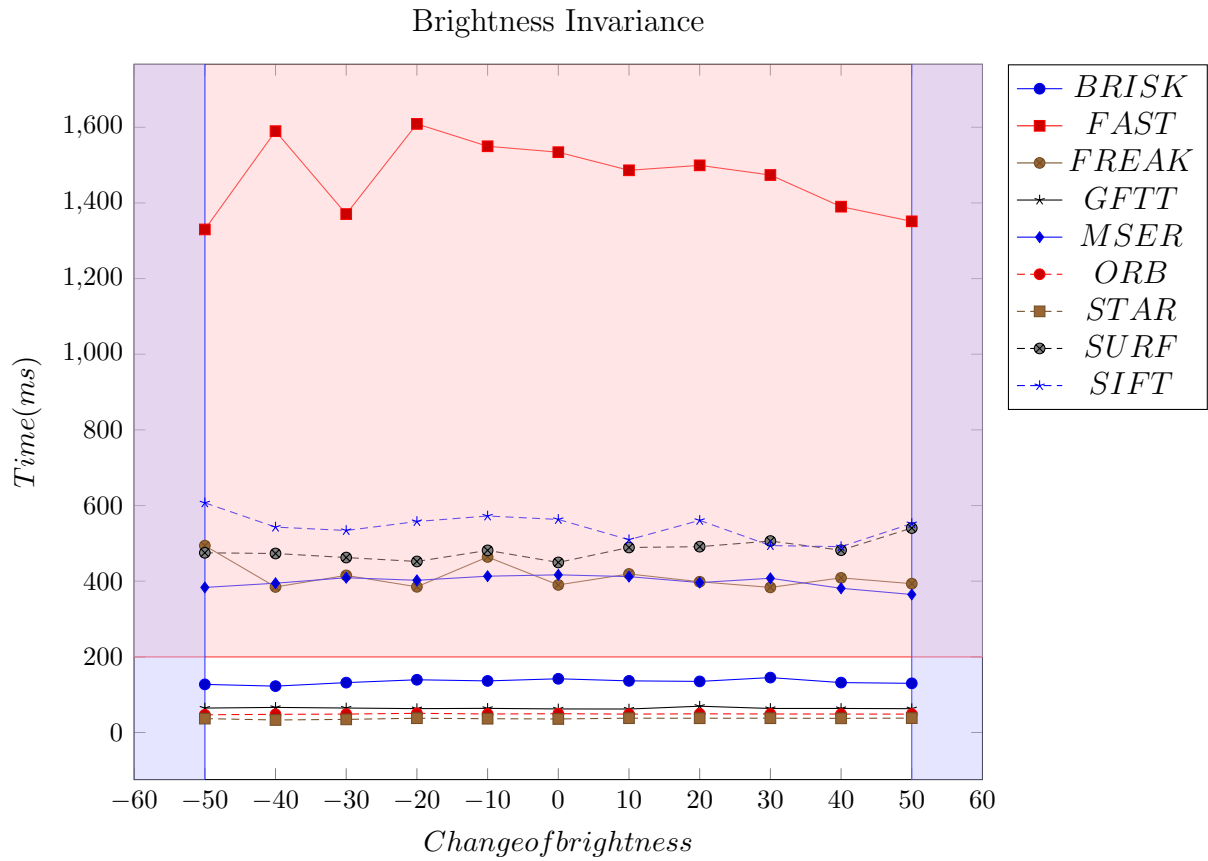


FIGURA 4.8 – Análise de tempo de variação de iluminação

4.3 Seleção da Técnica

A avaliação pontualmente feita em cada uma das análises e resumida na tabela 4.1, nos leva a escolher GFTT e ORB por serem os únicos que garantem resultados dentro da janela esperada para todos os testes:

TABELA 4.1 – Decisão de técnica a utilizar

	BRISK	FAST	FREAK	GFTT	MSER	ORB	STAR	SURF	SIFT
Precisão Escala			X	X		X			
Precisão Rotação	X	X	X	X	X	X	X	X	X
Precisão <i>Blur</i>		X	X	X	X	X	X	X	X
Precisão Iluminação	X	X	X	X	X	X	X	X	X
Tempo Escala	X			X		X	X		
Tempo Rotação	X			X		X	X		
Tempo <i>Blur</i>	X			X		X	X		
Tempo Iluminação	X			X		X	X		

Os resultados de precisão de escala, precisão de rotação, precisão de *blur* e precisão de iluminação foram avaliados considerando que as técnicas que estivessem pelo menos 50% dentro da janela de restrição, enquanto que os gráficos tempo escala, tempo rotação, tempo *blur* e tempo iluminação foram avaliados como excludentes pois caso a aplicação execute com cadência menor do que 5fps o usuário descartaria seu uso.

5 Conclusão

Este trabalho apresentou uma metodologia de comparação de algoritmos de reconhecimento de padrões, seguindo a mesma abordagem que apresentada em (KHVEDCHENYA, 2014a), (KHVEDCHENYA, 2014b) e (KHVEDCHENYA, 2014c), tendo como caso de uso o contexto aeronáutico, mais especificamente no reconhecimento de peças do interior da aeronave. Os conceitos básicos de Realidade Aumentada e dos algoritmos utilizados foram apresentados para fornecer meios ao conhecimento dessa tecnologia, que são fundamentais para compreender melhor a proposta do trabalho. Com o objetivo de traçar uma metodologia de decisão imparcial, foram descritas algumas restrições impostas à análise, que posteriormente foram aplicados nos resultados das simulações. Foram feitos testes emulando situações de variação ambiental, pois fazer os testes em ambientes e situações reais demandaria um custo muito elevado, além dos resultados serem semelhantes. Dessa forma concluiu-se que GFTT Seção 2.8.4 e ORB Seção 2.8.6 são os mais favoráveis para o contexto proposto no trabalho.

5.1 Atendimento dos objetivos

1. Avaliar os algoritmos clássicos de reconhecimento

O trabalho apresentou os algoritmos clássicos na Seção 2.8, descrevendo seu funcionamento e realizando testes de desempenho com os mesmos.

2. 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 re-

sultados de precisão de reconhecimento e de tempo gasto.

3. Selecionar algoritmo mais adequado para o contexto

Foi feita uma análise comparativa entre os algoritmos com as mesmas imagens de teste e conFigurando as técnicas com suas configurações padrão definidas pelo OpenCV, para que a única variável no momento fosse a técnica em questão. Foram aplicadas restrições através de parâmetros empíricos definidos por análise de engenharia, com os valores mais prováveis no contexto. Para a seleção das técnicas mais adequadas, foram utilizadas 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.

5.2 Proposta de Trabalhos Futuros

A utilização de Realidade Aumentada no campo da manutenção pode trazer muitos ganhos no que tange à usabilidade levando ao usuário uma quantidade de informações que, da maneira tradicional, por meio da inspeção e consulta em manuais, seria inviável. Este trabalho teve como foco o reconhecimento de padrões em um cenário aeronáutico específico. Como próximos passos temos:

- Adequar a aplicação para dispositivos móveis como *tablets*, celulares ou óculos de realidade aumentada 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 quanto ao desempenho;
- 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.

Referências

AGRAWAL, M.; KONOLIGE, K.; BLAS, M. R. Censure: Center surround extremas for realtime feature detection and matching. **Computer Vision - ECCV**, 2008.

ALAH, A.; ORTIZ, R.; VANDERGHEYNST, P. Freak: Fast retina keypoint. **Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland**, 2012.

AZUMA, R. A survey of augmented reality. **Hughes Research Laboratories**, aug 1997.

BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf : Speeded up robust features. **ETH Zurich**, 2008.

EL-GAYAR, M.; SOLIMAN, N. m. H. A comparative study of image low level feature extraction algorithms. **Egyptian Informatics Journal of image low level feature extraction algorithms**, 2013.

FORSYTH, D.; PONCE, J. **Computer Vision: A Modern Approach**. [S.l.]: Pearson, 2012. (Always learning).

HEINLY, J.; DUNN, E.; FRAHM, J.-M. Comparative evaluation of binary features. **Computer Vision - ECCV**, 2012.

ISIK, S.; OZKAN, K. A comparative evaluation of well-known feature detectors and descriptors. **International Journal of Applied Mathematics, Electronics and Computers**, 2014.

KHVEDCHENYA, E. **Comparison of the OpenCV feature detection algorithms**. 2014. Disponível em: <<http://computer-vision-talks.com/articles/2011-01-04-comparison-of-the-opencv-feature-detection-algorithms/>>. Acesso em: 19 jan. 2015.

- KHVEDCHENYA, E. **Comparison of the OpenCV feature detection algorithms - II**. 2014. Disponível em: <<http://computer-vision-talks.com/articles/2011-07-13-comparison-of-the-opencv-feature-detection-algorithms/>>. Acesso em: 19 jan. 2015.
- KHVEDCHENYA, E. **Feature descriptor comparison report**. 2014. Disponível em: <<http://computer-vision-talks.com/articles/2011-08-19-feature-descriptor-comparison-report>>. Acesso em: 19 jan. 2015.
- KIRNER, C.; SISCOUTO, R. **Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicacoes**. [S.l.]: Editora SBC, 2007.
- LEUTENEGGER, S.; CHLI, M.; SIEGWART, R. Brisk: Binary robust invariant scalable keypoints. In: **In Computer Vision (ICCV), 2011 IEEE International Conference on**. [S.l.: s.n.], 2011. p. 2548–2555.
- LOWE, D. Distinctive image features from scale-invariant keypoints. **Computer Science Department University of British Columbia Vancouver, B.C., Canada**, 2004.
- MATAS, J.; CHUM, O.; URBAN, M.; PAJDLA, T. Robust wide baseline stereo from maximally stable extremal regions. **British Machine Vision Conference**, 2002.
- MIKOLAJCZYK, K.; SCHMID, C. A performance evaluation of local descriptors. feb 2005.
- OPENCV. **Understanding Features**. 2014. Disponível em: <http://docs.opencv.org/trunk/doc/py_tutorials/py_feature2d/py_features_meaning/py_features_meaning.html>. Acesso em: 19 jan. 2015.
- ROSTEN, E.; DRUMMOND, T. Fusing points and lines for high performance tracking. **IEEE International Conference on Computer Vision**, vol. 2, 2005.
- ROSTEN, E.; DRUMMOND, T. Machine learning for high-speed corner detection. In: **In European Conference on Computer Vision**. [S.l.: s.n.], 2006. p. 430–443.
- RUBLEE, E. Orb: an efficient alternative to sift or surf. **Computer Vision (ICCV), 2011 IEEE International Conference on**, 2011.

- SCHAEFFER, C. **A Comparison of Keypoint Descriptors in the Context of Pedestrian Detection: FREAK vs. SURF vs. BRISK.** [S.l.]: Stanford University CS Department, 2013.
- SCHMIDT, A.; KRAFT, M.; FULARZ, M.; DOMAGALA, Z. The comparison of point feature detectors and descriptors in the context of robot navigation. **Workshop on Perception for Mobile Robots Autonomy**, 2012.
- SHI, J.; TOMASI, C. Good features to track. **IEEE**, 1994.
- TANG, J. C.; ISAACS, E. A.; INC, S.; TANG, J. C.; ISAACS", E. "why do users like video". In: **Computer Supported Cooperative Work.** [S.l.]: Kluwer Academic Publisher, 1993. p. 163–196.
- TUYTELAARS, T.; MIKOLAJCZYK, K. Local invariant feature detectors: A survey. **Int Comp. Graphics and Vision**, 2007.
- VIOLA, P.; JONES., M. Rapid object detection using a boosted cascade of simple features. 2001.
- WELCH, R. B. Perceptual modification: Adapting to altered sensory environments. **Academic Press**, 1978.

FOLHA DE REGISTRO DO DOCUMENTO

1. CLASSIFICAÇÃO/TIPO DP	2. DATA 17 de março de 2015	3. DOCUMENTO Nº DCTA/ITA/DP-005/2015	4. Nº DE PÁGINAS 63
5. TÍTULO E SUBTÍTULO: Avaliação de técnicas de reconhecimento de padrões em ambientes aeronauticos			
6. AUTOR(ES): Bruno Duarte Corrêa			
7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES): Divisão de Engenharia da Computação – ITA/IEC			
8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR: augmented reality,opencv,computer vision,maintenance,feature matching,descriptors,markerless			
9. PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO: Manutenção de aeronaves; Realidade virtual; Visão por computadores; Algoritmos; Sistemas homem- máquina; Engenharia aeronáutica; Computação.			
10. APRESENTAÇÃO: (X) Nacional () Internacional ITA, São José dos Campos. Curso de Mestrado Profissional em Engenharia Aeronáutica. Programa de Pós-Graduação em Engenharia Aeronáutica e Mecânica. Orientador: Luís Gonzaga Trabasso; coorientador: Nelson José Issa de Macedo. Defesa em 03/03/2015. Publicada em 2015.			
11. RESUMO: O reconhecimento de objetos em uma cena para posterior uso em realidade aumentada depende de diversas variáveis, causando a necessidade do uso de técnicas específicas para cada cenário, sendo portanto, um estudo de fronteiras para a melhor escolha do algoritmo de reconhecimento, de acordo com a aplicação em questão de grande valia para o meio acadêmico. Esta tese se propõe a pesquisar, categorizar e traçar fronteiras das técnicas conhecidas, tendo como caso de uso a manutenção de aeronaves feita dentro de centros fechados, utilizando as técnicas BRISK,FAST,FREAK,GFTT,MSER, ORB,STAR,SURF,SIFT em uma análise aplicada com imagens reais de janelas de inspeção do Embraer ERJ-190 para reconhecimento de objetos e posteriores aplicações em manutenção. Comparando todas as técnicas quanto à cadência e à precisão de reconhecimento de características, é possível selecionar GFTT e ORB como técnicas mais apropriadas ao contexto, por terem seus resultados de variação de rotação, escala, briho e <i>blur</i> dentro de uma faixa esperada para o contexto de manutenção.			
12. GRAU DE SIGILO: (X) OSTENSIVO () RESERVADO () CONFIDENCIAL (X) SECRETO			