



# Reconhecimento de características naturais para aplicação em sistemas de Realidade Aumentada

Gustavo Magalhães Moura

JUIZ DE FORA  
MARÇO, 2017

# Reconhecimento de características naturais para aplicação em sistemas de Realidade Aumentada

GUSTAVO MAGALHÃES MOURA

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Rodrigo Luis de Souza da Silva

JUIZ DE FORA

MARÇO, 2017

# RECONHECIMENTO DE CARACTERÍSTICAS NATURAIS PARA APLICAÇÃO EM SISTEMAS DE REALIDADE AUMENTADA

Gustavo Magalhães Moura

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Rodrigo Luis de Souza da Silva  
Doutor em Engenharia

Marcelo Caniato Renhe  
Mestre em Engenharia de Sistemas e Computação

Saulo Moraes Villela  
Doutor em Engenharia de Sistemas e Computação

JUIZ DE FORA  
16 DE MARÇO, 2017

## Resumo

Realidade Aumentada (RA) é uma tecnologia capaz de ampliar as interações do ser humano com o mundo real. Um dos campos de estudo desta área trata da utilização de objetos reais como marcadores para a geração do objeto virtual. Para a realização desta tarefa se faz necessário o reconhecimento de características do mundo real pelo sistema computacional. Este trabalho consiste na comparação e análise das combinações de algoritmos de detecção de características, descrição e correspondência presentes na biblioteca OpenCV que permitem a detecção de padrões pré-estabelecidos para determinar a melhor combinação para uma aplicação em RA. Para tal, foi avaliado o desempenho das combinações nas transformações de escala e rotação nos eixos X e Z em imagens e vídeos, utilizando como métricas a acurácia, o tempo de processamento e o *frame rate*, exclusivo para a avaliação de vídeo, na busca de uma imagem padrão em imagens teste. Os resultados mostram que, mesmo após refinamento das configurações das melhores combinações que utilizavam somente métodos livres, as combinações dos métodos analisados neste trabalho não são suficientes para a determinação da posição de um padrão em vídeo para aplicações de RA em tempo real.

**Palavras-chave:** Realidade aumentada, OpenCV, detecção de características.

# Abstract

Augmented Reality (RA) is a technology capable of amplifying the interactions of the human being with the real world. One of the fields of study in this area deals with the use of real objects as markers for the generation of the virtual object. To accomplish this task it is necessary the recognition of real world features by the computational system. This work consists of the comparison and analysis of the combinations of feature detection, description and correspondence algorithms present in the OpenCV library that allow the detection of pre-established patterns to determine the best combination for an application in RA. To accomplish this task, the performance of the combinations in the scale and rotation transformations in the X and Z axes in images and videos was evaluated, using accuracy, processing time and frame rate, exclusively for video evaluation, in search of a standard image in test images. The results show that, even after refinement of the configurations of the best combinations that use only free methods, the combinations of the methods analyzed in this work are not enough to determine the position of a video standard for RA applications in real time.

**Keywords:** Augmented Reality, OpenCV, feature detector.

## Agradecimentos

À minha esposa Marluce, com quem amo partilhar a vida. Obrigado pelo carinho e paciência na correria de cada semestre.

À todos os professores que buscaram transmitir seu conhecimento e ensinamentos ao longo da minha graduação. Em especial, ao Prof. Rodrigo Luis, responsável pela realização deste trabalho, pelas diversas orientações.

Aos colegas de curso com que compartilhei momentos de concentração, dedicação e de alegria ao longo destes anos.

Aos meus familiares, em especial aos meus avós maternos, que contribuíram, cada um à sua maneira, para a minha formação moral e ética.

*“If knowledge can create problems, it is not through ignorance that we can solve them.”*

*Isaac Asimov*

# Conteúdo

<b>Lista de Figuras</b>	<b>6</b>
<b>Lista de Tabelas</b>	<b>7</b>
<b>Lista de Abreviações</b>	<b>8</b>
<b>1 Introdução</b>	<b>9</b>
1.1 Problema abordado . . . . .	10
1.2 Objetivo . . . . .	11
1.2.1 Objetivos específicos . . . . .	11
1.3 Organização . . . . .	11
<b>2 Fundamentação teórica</b>	<b>12</b>
2.1 Detecção de características . . . . .	12
2.2 Descritores . . . . .	14
2.3 Algoritmos de correspondência . . . . .	15
2.4 Trabalhos Relacionados . . . . .	17
<b>3 Método proposto</b>	<b>23</b>
<b>4 Resultados</b>	<b>27</b>
4.1 Detectores de características . . . . .	27
4.2 Combinações . . . . .	29
<b>5 Conclusão e trabalhos futuros</b>	<b>35</b>
<b>Bibliografia</b>	<b>36</b>



## Lista de Figuras

1.1	Exemplos de marcadores fiduciais . . . . .	9
1.2	Exemplo de uso de marcador real . . . . .	10
3.1	Exemplo da construção das imagens testes. Dimensões em pixels. . . . .	24
3.2	Orientação dos eixos de rotação das transformações geométricas . . . . .	25
4.1	Distribuição dos pontos de interesse para cada algoritmo de detecção de características na imagem padrão. . . . .	28
4.2	Exemplo de correspondência de uma imagem padrão (a esquerda) em uma imagem teste . . . . .	30

## Lista de Tabelas

2.1	Tipos de características consideradas por cada método de detecção estudado	12
2.2	Resumo dos métodos citados e seus componentes . . . . .	16
2.3	Métodos de detecção de características e detecção presentes no OpenCV e estudos obtidos na revisão <i>quasi</i> -sistemática. . . . .	19
2.4	Transformações em imagens mais utilizadas em comparativos de detectores de características e descritores . . . . .	20
2.5	Métricas utilizadas nos estudos analisados. . . . .	21
2.6	Resumo dos principais resultados dos trabalhos avaliados conforme a transformação . . . . .	22
4.1	Desempenho dos detectores . . . . .	29
4.2	Acurácia dos testes com imagens de dimensões 1280 x 720 pixels conforme o eixo de rotação. . . . .	31
4.3	Resultados de acurácia e tempo médio de acerto para detecção de um padrão em imagens. . . . .	32
4.4	Resultados de acurácia e <i>frame rate</i> para detecção de um padrão em vídeo	33
4.5	Resultados das combinações do método ORB em imagens . . . . .	33
4.6	Resultados das combinações do método ORB em vídeos . . . . .	34

## Lista de Abreviações

BRIEF	Binary Robust Independent Elementary Features
BRISK	Binary Robust Invariant Keypoints
FAST	Features from Accelerated Segment Test
FREAK	Fast Retina Keypoint
GFTT	Good Features To Track
MSER	Maximally Stable Extremal Regions
OpenCV	Open Source Computer Vision
ORB	Oriented FAST and Rotated BRIEF
RA	Realidade Aumentada
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features

# 1 Introdução

Um sistema de Realidade Aumentada (RA) foi definido por Azuma et al. (2001) como o complemento do mundo real com objetos virtuais gerados por computador que aparentam coexistir no mesmo espaço que o mundo real. A geração destes objetos virtuais pode ser feita através de detecção e sobreposição de padrões, marcadores em imagens ou pontos georeferenciados.

Em sistemas de RA que utilizam marcadores para posicionar a imagem virtual gerada computacionalmente na cena, pode-se utilizar marcadores fiduciais ou marcadores reais. Marcadores fiduciais (Figura 1.1) são amplamente utilizados, pois possuem baixo custo e mínima manutenção. Contudo, estes marcadores devem estar em posições de interesse na cena, o que pode não ser viável em algumas situações, como em museus e áreas externas, onde podem poluir visualmente o ambiente. Assim, para contornar esta limitação, a utilização de imagens reais como marcadores se torna interessante.

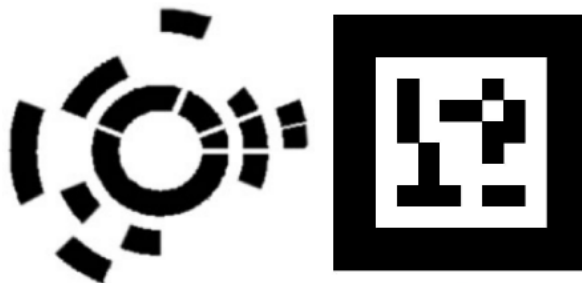


Figura 1.1: Exemplos de marcadores fiduciais

Para a utilização de marcadores reais em aplicações de RA (Figura 1.2), são necessários algoritmos para encontrar o marcador na cena, reconhecer sua posição relativa à câmera e projetar o objeto gerado computacionalmente na cena real. Especificamente ao uso de imagens reais como marcadores, para encontrar o marcador na cena são necessárias três etapas básicas: detecção de características (*feature detection*), descrição destas características em estruturas de dados (*description*) e correspondência do padrão com a cena (*matching*) utilizando os descritores.

Há bibliotecas que oferecem suporte para o desenvolvimento de aplicações de RA,



Figura 1.2: Exemplo de uso de marcador real

Fonte: (TAUB, 2009)

e entre elas destaca-se o OpenCV. O OpenCV (*Open Source Computer Vision*) é uma biblioteca gratuita e aberta criada em 1998 pela Intel, atualmente mantida pela Itseez, que disponibiliza ferramentas para o desenvolvimento de aplicações em Visão Computacional para uso acadêmico e comercial. Possui interfaces para C, C++, Python, Java e MATLAB e suporte para desenvolvimento em Linux, Windows, Mac OS, iOS e Android. O OpenCV possui mais de 2500 algoritmos que dão suporte para processamento básico e avançado de imagens, geometria computacional, detectores e descritores de características, rastreamento de objetos, fluxo óptico, calibração de câmera, entre outros. O OpenCV possui vasta aplicabilidade, desde controle de tráfego (RODRIGUES et al., 2013), à área de saúde (KURNIAWAN; DARMA PUTRA; SUDANA, 2014), e foi escolhido para este trabalho por ser gratuito, robusto, dar suporte a várias linguagens de programação.

## 1.1 Problema abordado

A biblioteca OpenCV possui 11 implementações de métodos para detecção de características (BRISK, Dense, FAST, GFTT, HARRIS, MSER, ORB, SIFT, SimpleBob, STAR e SURF), 6 de descrição (BRISK, BRIEF, FREAK, ORB, SIFT e SURF) e 5 de correspondência (BruteForce, BruteForce-L1, BruteForce-Hamming, BruteForce-Hamming(2) e FlannBased), que podem ser combinadas, conforme as particularidades

de cada método, para encontrar uma imagem padrão em uma outra imagem. E para uma aplicação de RA, a localização da imagem padrão deve ser rápida o suficiente para que, juntamente com a adição do objeto virtual, a fluidez da cena e a experiência do observador não sejam prejudicadas. Assim, há uma combinação de algoritmos presente no OpenCV que seja a melhor para uma aplicação RA?

## 1.2 Objetivo

O principal objetivo deste trabalho é pesquisar os algoritmos capazes de reconhecer características naturais em uma cena de forma a permitir o reconhecimento de objetos genéricos para utilização em sistemas de RA utilizando a biblioteca OpenCV.

### 1.2.1 Objetivos específicos

- Determinar as melhores combinações de algoritmos presentes na biblioteca OpenCV para reconhecer uma imagem padrão em uma cena;
- Determinar as melhores combinações de algoritmos presentes na biblioteca OpenCV para reconhecer uma imagem padrão em um vídeo.

## 1.3 Organização

Este trabalho contém mais 4 capítulos, além desta introdução, que apresenta uma breve descrição de Realidade Aumentada e o OpenCV, além do problema abordado e os objetivos deste trabalho. O próximo capítulo apresenta a fundamentação teórica, uma descrição de cada método utilizado e os trabalhos relacionados. O Capítulo 3 descreve detalhadamente o método proposto para tratar o problema descrito na Seção 1.1. O Capítulo 4 traz os experimentos e resultados obtidos na aplicação do método proposto. Por último, o Capítulo 5 conclui este trabalho indicando possibilidades de trabalhos futuros relacionados.

## 2 Fundamentação teórica

Este capítulo aborda os conceitos de detecção de padrões em imagens e um detalhamento dos métodos envolvidos neste trabalho.

### 2.1 Detecção de características

Segundo Suarez et al. (2014), detectores de características são métodos que extraem pontos de uma imagem que podem caracterizar seu conteúdo de forma robusta. O detector de características ideal extrai os mesmos pontos repetidamente sob quaisquer transformações que a imagem possa sofrer. Estes pontos, chamados também de características locais ou pontos de interesse, podem ser consideradas como áreas onde certas propriedades são aproximadamente constantes e podem ser classificados em bordas (*edges*), cantos (*corners*) ou regiões (*blobs*). Os métodos de detecção são especializados em um ou mais dos tipos de pontos citados. A Tabela 2.1 apresenta os tipos de características que cada método de detecção estudado considera.

Tabela 2.1: Tipos de características consideradas por cada método de detecção estudado

Detector	Tipo de característica		
	Borda	Canto	Região
BRISK		✓	✓
Dense	Detecta o tipo de característica que está distribuída regularmente na imagem.		
FAST		✓	✓
GFTT	✓	✓	
HARRIS	✓	✓	
MSER			✓
ORB		✓	✓
SIFT		✓	✓
SimpleBlob			✓
STAR	✓	✓	✓
SURF		✓	✓

A seguir, uma breve descrição dos métodos de detecção de características contidos no OpenCV e abordados neste trabalho em ordem cronológica:

- **HARRIS** combina o método GFTT com o método de detecção proposto por Harris e Stephens (1988). Este detector lida com cantos e bordas tolerando rotações, mas não é adequado a mudanças de escala.
- **GFTT** (*Good Features To Track*), proposto por Shi e Tomasi (1994), tenta detectar cantos e bordas que mais se destacam nas imagens, uma vez que são menos afetadas pela rotação ou escala da imagem.
- **SIFT** (*Scale Invariant Feature Transform*) foi apresentado por Lowe (1999) e é um dos detectores mais populares na literatura. O método processa a imagem pixel a pixel tendo um custo computacional elevado, mas tem a vantagem de possuir pontos de interesse invariantes a escala e a rotação. Este método é patenteado e não pode ser utilizado para fins comerciais sem autorização.
- **MSER** (*Maximally Stable Extremal Regions*), proposto por Matas et al. (2004) é um método direcionado para detecção de regiões, sendo tolerante a rotações, alterações de escala, variações de perspectiva e iluminação.
- **FAST** (*Features from Accelerated Segment Test*), proposto por Rosten e Drummond (2006), este detector é muito rápido na detecção de cantos. Entretanto, não é seletivo, detectando muitos pontos de interesse não ótimos.
- **SURF** (*Speeded Up Robust Features*) é um método proposto por Bay, Tuytelaars e Van Gool (2006), possuindo um detector de características robusto e tolerante a escala e rotação. É similar ao método SIFT na obtenção dos pontos de interesse, diferenciando no método utilizado para determinar os pontos de interesse válidos. Também possui patente, não podendo ser utilizado para fins comerciais.
- **BRISK** (*Binary Robust Invariant Keypoints*), proposto por Leutenegger, Chli e Siegwart (2011) foi inspirado no detector AGAST (MAIR et al., 2010), que por sua vez foi baseado no detector FAST. Foi apresentado como alternativa ao SIFT e ao SURF com a proposta de manter a robustez, mas com uma melhor performance.
- **ORB** (*Oriented FAST and Rotated BRIEF*) foi proposto por Rublee et al. (2011). Criado para ser uma alternativa ao SIFT e ao SURF, foi criado a partir do FAST



adicionando um componente de orientação. Os autores descrevem o método como sendo muito rápido e versátil.

- **Dense** é uma implementação do OpenCV e é um detector que escolhe os pontos de interesse nos nós de uma grade regular sobreposta à imagem, excluindo os dos limites da grade. Devido a esta característica, o processamento é custoso e os pontos de interesse escolhidos podem não ser relevantes.
- **SimpleBlob** também é uma implementação para o OpenCV que utiliza um método simples para detecção de regiões, como o nome sugere. O algoritmo converte a imagem original em duas ou mais cores gerando imagens em tons de cinza. A seguir os valores em tons de cinza são binarizados. As regiões comuns em ambas as imagens são então filtradas considerando diversos parâmetros. Este método é tolerante a rotações.
- **STAR** é uma implementação do OpenCV derivada do detector CenSurE (*Center Surround Extremas*), proposto por Agrawal, Konolige e Blas (2008). O CenSurE utiliza polígonos como quadrados, hexágonos e octógonos para delimitar a busca por regiões como alternativa menos custosa computacionalmente. O STAR utiliza como delimitador dois quadrados sobrepostos, sendo um rotacionado em  $45^\circ$ . A figura obtida é denominada octograma regular  $2\{4\}$ , que é uma estrela de oito pontas.

## 2.2 Descritores

Os algoritmos descritores transformam as informações dos pontos de interesse obtidas pelos algoritmos detectores em informações que são invariantes a diferenças de iluminação e pequenas deformações de perspectiva para serem utilizados pelos algoritmos de correspondência. Alguns algoritmos de detecção também tem seus componentes de descrição.

A seguir será apresentada uma breve descrição dos descritores utilizados neste trabalho em ordem cronológica.

- **SIFT**, proposto juntamente com seu detector, utiliza histograma de gradientes da região ao redor dos pontos de interesse dependente da localização deste ponto

na imagem. É um descritor robusto, invariante a escala, rotação e outras transformações nas imagens. Pela quantidade de informações que armazena para cada ponto de interesse, seu custo computacional é grande.

- **SURF** foi também proposto juntamente com seu detector, utiliza a orientação dominante da região quadrada em torno do ponto de interesse para determinar a orientação deste ponto.
- **BRIEF** (*Binary Robust Independent Elementary Features*) foi o primeiro descritor binário publicado e foi proposto por Calonder et al. (2010). Este descritor não possui um componente de orientação e não tem um padrão de amostragens dos pontos de interesse obtidos pelo detector.
- **BRISK** proposto juntamente com o detector de mesmo nome, é um descritor binário que utiliza um componente de orientação obtido pela análise dos pixels vizinhos aos pontos de interesse.
- **ORB** proposto juntamente com seu detector, o descritor foi baseado no descritor BRIEF, mas possui um cálculo mais eficiente da orientação dos pontos de interesse e uma análise da variância e correlação destes pontos para entregar os melhores como resultado.
- **FREAK** (*Fast Retina Keypoint*) é um descritor inspirado no sistema visual humano e foi proposto por Alahi, Ortiz e Vandergheynst (2012). É um descritor binário, utiliza pouca memória e é de rápido processamento.

Como apresentado, há métodos que somente são detectores ou descritores e há métodos que possuem componentes de detecção e de descrição. A Tabela 2.2 resume os componentes dos métodos já apresentados.

## 2.3 Algoritmos de correspondência

Os algoritmos de correspondência (*matchers*) são algoritmos que determinam quais características representadas nos descritores de duas imagens são semelhantes segundo seus

Tabela 2.2: Resumo dos métodos citados e seus componentes

Método	Detector	Descritor
BRISK	✓	✓
BRIEF		✓
Dense	✓	
FAST	✓	
FREAK		✓
GFTT	✓	
HARRIS	✓	
MSER	✓	
ORB	✓	✓
SIFT	✓	✓
SimpleBlob	✓	
STAR	✓	
SURF	✓	✓

critérios. Quanto maior o número de características semelhantes encontrados, maiores as chances de uma imagem estar contida em outra imagem ou duas imagens serem semelhantes.

Os algoritmos de correspondência por força bruta do OpenCV são simples. Cada característica do primeiro descritor é comparada com todas as características do segundo descritor segundo um cálculo de distância sendo o par mais próximo retornado. Um valor mínimo de distância determina se o par será considerado correspondente ou não.

A seguir será apresentada uma breve descrição dos algoritmos de correspondência utilizados e suas descrições por Suarez et al. (2014). Nas equações apresentadas a seguir,  $V_1$  e  $V_2$  são os vetores de características das duas imagens e  $M$  o tamanho dos vetores.

- **BruteForce-L1** utiliza a distância métrica L1, também chamada de *Manhattan* ou *City Block*, para determinar a distância entre descritores de ponto flutuante, como mostra a Equação 2.1:

$$d(V_1, V_2) = \sum_{i=1}^M |v_1[i] - v_2[i]| \quad (2.1)$$

- **BruteForce** é utilizado para descritores de ponto flutuante e a distância considerada é a L2, também chamada de distância Euclidiana. Este método requer mais processamento que o BruteForce-L1, já que é uma função quadrática, como mostra a Equação 2.2:

$$d(V_1, V_2) = \sqrt{\sum_{i=1}^M (|v_1[i] - v_2[i]|)^2} \quad (2.2)$$

- **BruteForce-Hamming** é utilizado para descritores binários. Sua equação é o somatório dos resultados de uma operação de XOR bit a bit entre os vetores de descrição das duas imagens, conforme a Equação 2.3:

$$d(V_1, V_2) = \sum_{i=1}^M v_1[i] \otimes v_2[i] \quad (2.3)$$

- **BruteForce-Hamming(2)** também é utilizado para descritores binários e utiliza dois bits ao invés de um bit na operação de XOR se comparado ao algoritmo anterior.
- **FlannBased** proposto por Muja e Lowe (2009), este algoritmo é utilizado para descritores de ponto flutuante. Utiliza uma estrutura para pré-processamento, sendo geralmente mais rápido que os algoritmos força bruta, ao custo de uma maior utilização de memória.

## 2.4 Trabalhos Relacionados

As imagens podem sofrer diversas transformações, que podem ser classificadas em geométricas e não geométricas. As transformações geométricas modificam a relação espacial entre os pixels de uma imagem (GONZALEZ; WOODS, 2002), como rotação, translação, mudanças de escala e perspectiva, e as transformações não geométricas consistem naquelas que são dependentes da captura de imagens e não dependem do ponto de vista (HEINLY; DUNN; FRAHM, 2012), como iluminação, contraste e compressão.

Os métodos de detecção de características e descrição são sensíveis a uma ou mais transformações que podem ser determinantes para o desempenho em sistemas de RA. Com o objetivo de verificar a suscetibilidade dos métodos de detecção de características e seus desempenhos relativos a estas transformações, foi realizado um conjunto de estudos baseados em revisão sistemática (KITCHENHAM, 2004), utilizando como base os passos contidos em (NEIVA; SILVA, 2016).

Assim, as questões formuladas para nortear a revisão *quasi*-sistemática foram:

- Questão 1: Quais métodos de detecção e descrição presentes no OpenCV são mais comparados?
- Questão 2: Quais transformações em imagens são mais avaliadas em comparativos nos métodos de detecção e descrição presentes no OpenCV?
- Questão 3: Quais as métricas utilizadas para realização dos comparativos?
- Questão 4: Qual a sensibilidade dos métodos de detecção de características e descrição presentes no OpenCV nas diferentes transformações em imagens?

As buscas por trabalhos para responder as questões formuladas foram realizadas em bibliotecas digitais que apresentaram ser as fontes mais relevantes no momento da pesquisa, IEEE Xplore, ACM, ScienceDirect e Springer, utilizando as seguintes palavras-chave: comparação (*comparison*) ou comparativo (*comparative*) no título do artigo, combinado com extração de características (*feature extraction*) ou detecção de características (*feature detection*) ou descritor (*descriptor*) e imagem (*image*) juntamente com algum dos nomes dos detectores de características e descritores presentes no OpenCV: BRISK, BRIEF, Dense, FAST, FREAK, GFTT, HARRIS, MSER, ORB, SIFT, SimpleBlob, STAR e SURF.

Os critérios para inclusão e exclusão foram fontes disponíveis na *Web* com utilização de pelo menos dois métodos de detecção de características ou descritores presentes no OpenCV e que foram abordados neste trabalho, utilização de pelo menos uma transformação em imagens comum à área de visão computacional, publicados após 01 de janeiro de 2006, em língua inglesa e que estavam disponíveis de forma gratuita.

Após a execução da busca utilizando as palavras-chave, os trabalhos resultantes foram analisados e selecionados aqueles que estavam dentro dos critérios pré-estabelecidos.

Relacionada à questão 1, a Tabela 2.3 apresenta os métodos de detecção de características e de descrição presentes no OpenCV e os estudos comparativos com transformações em imagens que os abordaram. A utilização dos métodos SIFT e SURF em maior quantidade de estudos se deve ao fato de serem considerados os métodos estado da arte pelos estudos que os utilizaram e serem a base de comparação na criação de novos

métodos, como em Agrawal, Konolige e Blas (2008), Leutenegger, Chli e Siegwart (2011) e Rublee et al. (2011). Não foram encontrados estudos que abordam todos os métodos presentes no OpenCV.

Tabela 2.3: Métodos de detecção de características e detecção presentes no OpenCV e estudos obtidos na revisão *quasi*-sistemática.

Método	Quantidade de estudos	Referências
BRISK	3	(CHAO et al., 2013) (HEINLY; DUNN; FRAHM, 2012) (MUKHERJEE; WU; WANG, 2015)
BRIEF	2	(HEINLY; DUNN; FRAHM, 2012) (MUKHERJEE; WU; WANG, 2015)
Dense	0	
FAST	3	(EL-GAYAR; SOLIMAN; MEKY, 2013) (HEINLY; DUNN; FRAHM, 2012) (MUKHERJEE; WU; WANG, 2015)
FREAK	1	(MUKHERJEE; WU; WANG, 2015)
GFTT	0	
HARRIS	2	(HEINLY; DUNN; FRAHM, 2012) (JIANG; YI, 2012)
MSER	4	(BOUCHIHA; BESBES, 2015) (CHAO et al., 2013) (HEINLY; DUNN; FRAHM, 2012) (MUKHERJEE; WU; WANG, 2015)
ORB	3	(HEINLY; DUNN; FRAHM, 2012) (HEINLY; DUNN; FRAHM, 2012) (MUKHERJEE; WU; WANG, 2015)
SIFT	7	(BOUCHIHA; BESBES, 2015) (CHAO et al., 2013) (EL-GAYAR; SOLIMAN; MEKY, 2013) (HEINLY; DUNN; FRAHM, 2012) (JIANG; YI, 2012) (MUKHERJEE; WU; WANG, 2015) (WU et al., 2013)
SimpleBlob	0	
STAR	1	(CHAO et al., 2013)
SURF	6	(BOUCHIHA; BESBES, 2015) (CHAO et al., 2013) (EL-GAYAR; SOLIMAN; MEKY, 2013) (HEINLY; DUNN; FRAHM, 2012) (MUKHERJEE; WU; WANG, 2015) (WU et al., 2013)

A Tabela 2.4 apresenta as transformações estudadas nos trabalhos analisados para responder à questão 2. Há uma preferência pelas transformações geométricas, escala e rotação, nos trabalhos encontrados.

A resposta da questão 3 é trazida pelos dados da Tabela 2.5. As métricas mais avaliadas são o tempo e o número de correspondências. É plausível a métrica de tempo, pois na maior parte das aplicações quer se chegar ao resultado no menor tempo possível. A segunda é uma medida para avaliar a combinação de detectores de características e des-

Tabela 2.4: Transformações em imagens mais utilizadas em comparativos de detectores de características e descritores

<b>Transformação</b>	<b>Quantidade de estudos</b>	<b>Referências</b>
Compressão JPEG	3	(CHAO et al., 2013) (HEINLY; DUNN; FRAHM, 2012) (MUKHERJEE; WU; WANG, 2015)
Embaçamento	3	(EL-GAYAR; SOLIMAN; MEKY, 2013) (HEINLY; DUNN; FRAHM, 2012) (WU et al., 2013)
Escala	6	(BOUCHIHA; BESBES, 2015) (EL-GAYAR; SOLIMAN; MEKY, 2013) (HEINLY; DUNN; FRAHM, 2012) (JIANG; YI, 2012) (MUKHERJEE; WU; WANG, 2015) (WU et al., 2013)
Exposição	2	(HEINLY; DUNN; FRAHM, 2012) (MUKHERJEE; WU; WANG, 2015)
Iluminação	6	(BOUCHIHA; BESBES, 2015) (EL-GAYAR; SOLIMAN; MEKY, 2013) (HEINLY; DUNN; FRAHM, 2012) (JIANG; YI, 2012) (MUKHERJEE; WU; WANG, 2015) (WU et al., 2013)
Oclusão	1	(EL-GAYAR; SOLIMAN; MEKY, 2013)
Perspectiva	2	(HEINLY; DUNN; FRAHM, 2012) (MUKHERJEE; WU; WANG, 2015)
Rotação	6	(BOUCHIHA; BESBES, 2015) (EL-GAYAR; SOLIMAN; MEKY, 2013) (HEINLY; DUNN; FRAHM, 2012) (JIANG; YI, 2012) (MUKHERJEE; WU; WANG, 2015) (WU et al., 2013)

critérios, uma vez que determina quantas características estão corretamente relacionadas entre as imagens testes.

Tabela 2.5: Métricas utilizadas nos estudos analisados.

Métrica	Descrição	Quantidade de estudos	Referências
Taxa correta de correspondência	Medida da proporção de imagens corretamente relacionadas com o padrão	1	(WU et al., 2013)
Entropia	Medida de propagação ou aleatoriedade na distribuição espacial de características na imagem	1	(HEINLY; DUNN; FRAHM, 2012)
Média das precisões médias ( <i>mAP</i> )	Média de todas as precisões médias de cada conjunto de imagens	1	(CHAO et al., 2013)
Número de correspondências	Número de características corretamente relacionadas entre o par de imagens	2	(EL-GAYAR; SOLIMAN; MEKY, 2013) (HEINLY; DUNN; FRAHM, 2012)
Precisão	Representa a precisão correspondente a um par detector+descriptor e o quão restritivo é o critério de correspondência	2	(HEINLY; DUNN; FRAHM, 2012) (MUKHERJEE; WU; WANG, 2015)
<i>Recall</i>	Quantifica quantas das possíveis correspondências corretas foram realmente encontradas	2	(BOUCHIHA; BESBES, 2015) (HEINLY; DUNN; FRAHM, 2012)
Repetibilidade	Número médio de regiões detectadas em imagens sob diferentes transformações	1	(HEINLY; DUNN; FRAHM, 2012)
Taxa de correspondência suposta	Representa a seletividade do descriptor	2	(HEINLY; DUNN; FRAHM, 2012) (MUKHERJEE; WU; WANG, 2015)
Tempo	Tempo entre o início e o fim de uma execução	4	(BOUCHIHA; BESBES, 2015) (EL-GAYAR; SOLIMAN; MEKY, 2013) (MUKHERJEE; WU; WANG, 2015) (WU et al., 2013)

Os dados para responder à questão 4 não se mostraram bem definidos para realizar uma ordenação unificada de sensibilidade dos métodos às transformações em imagens, uma vez que não são utilizados critérios comuns para a avaliação dos métodos frente às transformações. Por exemplo, em trabalhos diferentes são utilizados *datasets* e descritores diferentes para os métodos de detecção de características que não possuem este método de mesmo autor, FAST, HARRIS, MSER e STAR. Outro fator que influencia o desempenho dos métodos está relacionado à sua forma de implementação. Chao et al. (2013) cita que as implementações do OpenCV dos métodos SIFT e SURF obtiveram desempenhos inferiores que as implementações que obtiveram de outro trabalho. Dos trabalhos analisados, três utilizaram implementações do OpenCV de algum dos métodos para seus comparativos, (EL-GAYAR; SOLIMAN; MEKY, 2013; HEINLY; DUNN; FRAHM, 2012; MUKHERJEE; WU; WANG, 2015). Outro ponto a se considerar, é que os detectores têm forte influência no desempenho de transformações geométricas, principalmente, escala e rotação, assim a escolha de métodos para comparar o desempenho de descritores deve



Tabela 2.6: Resumo dos principais resultados dos trabalhos avaliados conforme a transformação

Tipo de método	Transformação	Resultado <sup>1</sup>	Observação	Referência
Descritor	Escala	Performances similares entre SIFT e SURF	Separou o desempenho do detector, mas não utilizou um detector único para a comparação	(BOUCHIHA; BESBES, 2015)
	Iluminação	SIFT > SURF		
	Rotação	SIFT > SURF em ângulos não maiores que 60°		
Detector+descritor	Compressão JPEG	MSER+SIFT	Fixou parâmetros para cada detector e utilizou as implementações dos autores de cada método	(CHAO et al., 2013)
Detector	Embaçamento	SIFT > SURF > FAST	Utilizou tempo e número de características como parâmetros	(EL-GAYAR; SOLIMAN; MEKY, 2013)
	Escala	SURF > SIFT > FAST		
	Iluminação	SIFT > SURF > FAST		
	Rotação	SIFT > SURF > FAST		
Detector+descritor	Compressão JPEG	HARRIS+BRIEF	Não utilizou combinação dos descritores SURF e SIFT com outros detectores, somente com os de mesmo nome e construiu o próprio <i>dataset</i> para algumas transformações	(HEINLY; DUNN; FRAHM, 2012)
	Embaçamento	SIFT+BRIEF		
	Escala	SIFT+SIFT		
	Exposição	FAST+BRIEF		
	Iluminação	MSER+BRIEF		
	Perspectiva	SIFT+SIFT		
Detector	Rotação	FAST+BRISK	Utilizou frames de vídeo com transformações Gaussianas para as avaliações	(JIANG; YI, 2012)
	Escala	SIFT > HARRIS		
	Iluminação	SIFT > HARRIS		
Detector+descritor	Rotação	SIFT > HARRIS	Utilizou brute-force para correspondência e os parâmetros padrões do OpenCV para os detectores	(MUKHERJEE; WU; WANG, 2015)
	Embaçamento	SURF+FREAK		
	Escala	SIFT+SIFT		
	Perspectiva	MSER+BRIEF		
Métodos com componentes de detecção e descrição	Rotação	SIFT+SIFT		(WU et al., 2013)
	Embaçamento	SIFT > SURF		
	Escala	SIFT > SURF		
	Iluminação	SIFT > SURF		
	Rotação	SIFT > SURF		

(1) Apresenta os resultados em escala, onde o símbolo > representa “melhor que” ou o melhor método ou combinação conforme a transformação.

ser criteriosa. A Tabela 2.6 apresenta um resumo dos principais resultados dos trabalhos analisados na revisão sistemática.

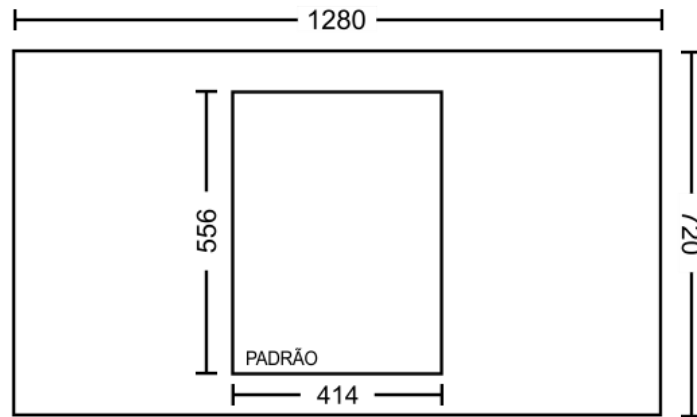
### 3 Método proposto

Este capítulo tem como objetivo descrever o método de comparação utilizado, construção dos padrões e imagens testes e as métricas utilizadas.

Uma aplicação de RA deve reconhecer um padrão em uma cena, para então utilizá-lo como referência no registro de um objeto virtual. Este padrão pode ser 2D ou 3D, mas normalmente padrões 2D, como uma imagem, são os mais utilizados. Esta imagem deve ser preferencialmente obtida frontalmente e sob condições favoráveis de iluminação para que a captura dos seus detalhes seja realizada de forma adequada. Ao executarmos a busca deste padrão em uma segunda imagem, ou em um vídeo, o mesmo pode sofrer distorções geométricas, como rotações e escala, que dificultam sua detecção. Além disso, uma sequência de imagens captadas em uma cena pode sofrer variação de luminosidade, contraste, foco, entre outras características visuais que podem atrapalhar a identificação de um determinado marcador ao mover a câmera de sua posição inicial. Desta forma, faz-se necessário utilizar uma metodologia sistemática de captura de imagens para que a variação das condições do ambiente não influencie na análise dos algoritmos descritos neste trabalho.

Com o objetivo de criar uma sequência de imagens com características mais controladas, foi utilizado um programa de *design* gráfico para construir a imagem padrão e as imagens testes, variando somente os parâmetros da transformação de estudo nas imagens testes. Para construção da imagem padrão, foi utilizada uma imagem em alta resolução de domínio público, obtida através de pesquisa na internet de um objeto do cotidiano com uma alta heterogeneidade de formas e cores. As imagens testes foram construídas utilizando a imagem padrão ao centro, com dimensões que correspondessem a, aproximadamente, 25% da área da imagem teste. Ao fundo foram dispostas outras imagens em alta resolução do mesmo tamanho do padrão posicionadas em ângulos diversos, a fim de se obter uma imagem teste com fundo heterogêneo. A Figura 3.1 exemplifica a construção das imagens testes. As dimensões das imagens testes foram reduzidas e uniformizadas conforme o padrão do experimento e o método de compressão utilizado foi o JPEG (*Joint*

*Photographics Experts Group*).



$$\frac{556 \times 414}{1280 \times 720} \approx 25\%$$

Figura 3.1: Exemplo da construção das imagens testes. Dimensões em pixels.

Para a construção dos vídeos testes, foi utilizado um programa de construção e edição de vídeos e as mesmas imagens em alta resolução utilizadas para construir as imagens testes. As dimensões dos vídeos testes foram as mesmas das imagens testes e o formato de compressão utilizado para os vídeos testes foi o AVI (*Audio Video Interleave*).

As transformações avaliadas foram a rotação no eixo X e eixo Z (Figura 3.2), transformações comuns em aplicações de RA, e resolução da mídia, que é um parâmetro da câmera utilizada para captação da cena. Para construção das imagens testes para a transformações de rotação, variou-se os ângulos de 0° a 75° para rotação no eixo X e de 0° a 345° para rotação no eixo Z em intervalos regulares de 15°, totalizando 6 e 24 imagens, respectivamente, para cada resolução avaliada. Para os vídeos testes, a taxa de variação do ângulo foi de 1 grau por quadro e a taxa de quadros por segundo foi de 24 *fps* para as duas transformações de rotação.

Para analisar os algoritmos citados, os parâmetros utilizados para comparação foram os seguintes:

- **Tempo de processamento:** tempo de processamento para que o algoritmo, ou conjunto de algoritmos, conclua seu propósito;
- **Entropia:** exclusiva para os detectores, segundo Heinly, Dunn e Frahm (2012), a entropia aborda a influência do detector em um descritor e penaliza detectores que

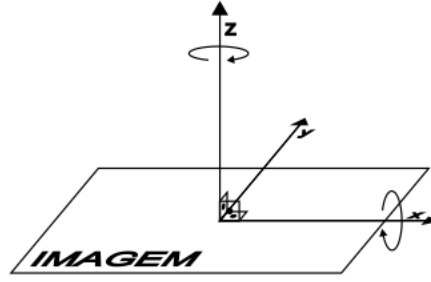


Figura 3.2: Orientação dos eixos de rotação das transformações geométricas

clusterizam seus pontos de interesse. Seus valores devem ser utilizados para ordenar os métodos de detecção envolvidos quanto à distribuição dos pontos de interesse na imagem. Para sua computação, o espaço de características 2D é uniformemente espaçado em regiões. As contribuições de cada ponto de uma região são ponderadas por uma Gaussiana relativa à distância do centro da região conforme abaixo:

$$r(c) = \frac{1}{Z} \sum_{p \in P} G(\|c - p\|) \quad (3.1)$$

onde  $r(c)$  é a região de centro  $c = (x_c, y_c)$ ,  $\frac{1}{Z}$  é uma constante calculada tal que a soma de todos  $r(c)$  seja igual a 1, assim  $Z = \sum_{c \in C} \sum_{p \in P} G(\|c - p\|)$ ,  $p$  é o ponto da característica no conjunto de características  $P$  da região  $r(c)$  e  $G$  é a Gaussiana da forma:

$$G(a) = k \cdot e^{-\frac{(a-l)^2}{2m^2}} \quad (3.2)$$

onde  $k$ ,  $l$  e  $m$  são constantes. A entropia do detector é calculada conforme a seguinte equação:

$$Entropia = \sum_c -r(c) \log r(c) \quad (3.3)$$

- **Tempo médio de acerto:** tempo médio de processamento de um conjunto de algoritmos nos casos onde houve resposta positiva;
- **Acurácia:** número de imagens teste onde a posição da imagem padrão foi corretamente identificada dividido pelo número total de imagens testes;

- **Taxa de quadros (*Frame rate*):** número de quadros processados por segundo (*fps*) pelo conjunto de algoritmos, exclusiva para a análise dos testes em vídeos.

As métricas de tempo foram utilizadas, pois a resposta da posição da imagem padrão na imagem teste deve ser rápida de forma que o sistema computacional possa sobrepor a imagem virtual sem que a fluidez da cena e a experiência do observador sejam prejudicadas. Optou-se pela acurácia, mesmo não ocorrendo em nenhum dos comparativos retornados pela revisão *quasi*-sistemática, pois a combinação de métodos deve trazer o maior percentual de resultados positivos, dentro das transformações impostas à cena para diminuir a limitação de movimentação do equipamento de captação de imagens.

## 4 Resultados

Este capítulo apresenta os experimentos e resultados obtidos afim de se obter a resposta do problema.

Os experimentos aqui descritos foram realizados em um *notebook* Asus S400CA com processador Intel® Core™ i5 3317U 1.70 GHz, 4Gb de memória RAM DDR3 1600 MHz SDRAM e placa gráfica Intel® HD Graphics 4000.

Os algoritmos dos métodos apresentados nas Seções 2.1, 2.2 e 2.3 foram executados utilizando a interface comum que o OpenCV, versão 2.4.11, suporta para cada tipo de algoritmo na linguagem C++ utilizando os valores *default* dos parâmetros. As aplicações desenvolvidas para os testes foram compiladas utilizando o GNU GCC Compiler 4.8.1. Entretanto, para os detectores ORB, GFTT e HARRIS, foi necessário especificar o número máximo de pontos a serem detectados, exigência da implementação do método. Desta forma, para equiparar todos os detectores ao seu número máximo de pontos de interesse computáveis, foram especificados para estes três detectores um valor suficientemente alto.

### 4.1 Detectores de características

A Figura 4.1 ilustra a distribuição dos pontos de interesse obtidos por cada algoritmo de detecção de características avaliados, onde é possível compará-las com a imagem padrão utilizada neste trabalho.

O número de pontos de interesse, o tempo de execução e a entropia de cada detector obtidos para a imagem padrão de 414 por 556 pixels e a imagem teste sem transformação com dimensões de 1280 x 720 pixels constam na Tabela 4.1. Para o cálculo da entropia, os valores das constantes da Gaussiana (Equação 3.2) foram obtidos variando-os nos intervalos  $[1,180]$  e  $[1,30]$  para  $l$  e  $m$ , respectivamente, afim de obter um intervalo entre os valores de entropia que possibilitasse classificar os métodos de detecção de forma segura, assim os valores escolhidos foram  $k = 1$ ,  $l = 150$  e  $m = 20$ .

Os detectores Dense, FAST e ORB se destacaram pelo grande número de pon-



Figura 4.1: Distribuição dos pontos de interesse para cada algoritmo de detecção de características na imagem padrão.

Tabela 4.1: Desempenho dos detectores

Detector	Imagem padrão			Imagem teste		
	Pontos de Interesse	Tempo (ms)	Entropia	Pontos de Interesse	Tempo (ms)	Entropia
BRISK	944	35,29	4,401	2518	102,42	4,289
Dense	6417	0,13	4,852	25680	0,50	5,187
FAST	6336	5,75	4,611	15415	15,45	4,577
GFTT	2972	19,89	4,491	6613	77,19	4,404
HARRIS	1431	18,68	4,262	2612	73,37	3,852
MSER	286	238,06	3,873	793	818,78	3,651
ORB	10885	37,07	4,362	34013	112,78	4,515
SIFT	1546	195,85	4,420	4513	735,89	4,639
SimpleBlob	11	90,18	0,662	37	234,85	1,845
STAR	481	14,04	3,997	1933	80,66	4,302
SURF	2243	317,95	4,640	6809	1120,66	4,899

tos de interesse detectados, sendo que os dois primeiros se destacaram ainda pelo baixo tempo de execução. O detector Dense obteve as maiores entropias nas duas imagens, ou seja, os pontos de interesse são mais distribuídos, o que está de acordo com sua estratégia de identificação dos pontos de interesse, conforme mostra a Figura 4.1c. Os detectores SIFT e SURF, (Figuras 4.1i e 4.1l, respectivamente) considerados métodos estado da arte, obtiveram altas entropias nas imagens, mesmo com um menor número de pontos de interesse detectados comparando-se com os detectores FAST, GFTT e ORB, demonstrando que possuem estratégias eficientes para determinar pontos de interesse com uma maior distribuição espacial nas imagens. A ordem dos métodos de detecção quanto ao seu resultado de entropia foi próxima à obtida no trabalho de Heinly, Dunn e Frahm (2012), considerando que a entropia é dependente da imagem avaliada.

## 4.2 Combinações

Como o objetivo inicial da maioria das aplicações de RA é a detecção de um padrão em uma imagem (Figura 4.2), utilizou-se todas as combinações dos algoritmos descritos anteriormente, 330 combinações ao todo, para detectar a imagem padrão em imagens testes de resoluções de 414 x 556 e 1280 x 720 pixels e com as transformações geométricas de rotação no eixo Z e rotação no eixo X.

A Tabela 4.2 apresenta a acurácia, em percentual, obtida para todas as com-





Figura 4.2: Exemplo de correspondência de uma imagem padrão (a esquerda) em uma imagem teste

Combinações de algoritmos nos testes de detecção do padrão em imagens com dimensões 1280 x 720 pixels e com as transformações geométricas de rotação no eixo Z e rotação no eixo X. Algumas combinações apresentaram erro durante os testes por incompatibilidade entre os métodos envolvidos e estão sem resultados ou não estão apresentadas. As combinações que obtiveram melhores acurácias considerando as duas transformações estão em destaque na Tabela 4.2, totalizando 36 combinações.

Apesar de o detector Dense ter obtido os maiores valores de entropia (Tabela 4.1), isto não se mostrou suficiente para obter resultados satisfatórios de acurácia, uma vez que estes pontos podem não ter sido relevantes e não tiveram correspondentes entre a imagem padrão e as imagens testes.

Para o restante do trabalho, a notação para os trios detector, descritor e algoritmo de correspondência detector+descritor+algoritmo de correspondência, para simplificar a representação.

A Tabela 4.3 apresenta os resultados das 36 melhores combinações nos testes com imagens de dimensões 1280 x 720, 960 x 540 e 480 x 360 pixels. Para cada combinação é apresentado a acurácia percentual e o tempo médio de acerto na detecção do padrão em segundos para as transformações geométricas de rotação no eixo Z e rotação no eixo X. Verificou-se que as combinações com maior percentual de acurácia associado a um menor tempo de processamento possuem como detector o método SIFT, sendo que a combinação SIFT+SIFT+BruteForce-L1 obteve melhores resultados de acurácia. As combinações SIFT+BRISK+BruteForce-Hamming e SIFT+BRISK+BruteForce-Hamming(2) obtiveram resultados satisfatórios de acurácia associados a menores tempos de processamento

Tabela 4.2: Acurácia dos testes com imagens de dimensões 1280 x 720 pixels conforme o eixo de rotação.

Eixo Z											Eixo X											
	BK	DE	FA	GF	HA	MS	OR	SI	SB	ST	SU	BK	DE	FA	GF	HA	MS	OR	SI	SB	ST	SU
BF	0,0	4,2	0,0	4,2	8,3	4,2	4,2	4,2	0,0	8,3	8,3	33,3	16,7	0,0	16,7	33,3	33,3	0,0	16,7	16,7	33,3	33,3
BK	0,0	4,2	62,5	54,2	37,5	16,7	41,7	91,7	0,0	50,0	70,8	0,0	66,7	33,3	16,7	0,0	16,7	33,3	50,0	0,0	50,0	66,7
FR	0,0	0,0	0,0	0,0	0,0	0,0	0,0	-	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
OR	0,0	4,2	41,7	33,3	20,8	0,0	66,7	-	0,0	8,3	79,2	0,0	16,7	16,7	16,7	16,7	0,0	33,3	-	0,0	16,7	16,7
SI	<b>100,0</b>	-	12,5	12,5	12,5	12,5	<b>100,0</b>	<b>100,0</b>	4,2	12,5	<b>100,0</b>	<b>66,7</b>	-	83,3	83,3	83,3	83,3	<b>50,0</b>	<b>83,3</b>	33,3	83,3	<b>66,7</b>
SU	<b>95,8</b>	-	<b>100,0</b>	25,0	29,2	<b>100,0</b>	<b>100,0</b>	<b>95,8</b>	16,7	<b>100,0</b>	<b>100,0</b>	<b>66,7</b>	-	<b>50,0</b>	0,0	0,0	<b>50,0</b>	<b>66,7</b>	<b>50,0</b>	0,0	<b>50,0</b>	<b>66,7</b>

(a) BruteForce

	BK	DE	FA	GF	HA	MS	OR	SI	SB	ST	SU	BK	DE	FA	GF	HA	MS	OR	SI	SB	ST	SU
BF	8,3	4,2	4,2	8,3	8,3	8,3	4,2	4,2	0,0	8,3	8,3	33,3	16,7	0,0	16,7	33,3	16,7	0,0	16,7	16,7	33,3	33,3
BK	4,2	8,3	87,5	91,7	91,7	33,3	41,7	66,7	0,0	62,5	50,0	0,0	66,7	50,0	50,0	66,7	33,3	33,3	66,7	0,0	50,0	50,0
FR	0,0	0,0	0,0	-	0,0	0,0	0,0	-	-	0,0	0,0	0,0	0,0	0,0	-	0,0	0,0	0,0	-	-	0,0	0,0
OR	0,0	0,0	45,8	37,5	50,0	8,3	45,8	-	0,0	45,8	66,7	0,0	16,7	16,7	16,7	33,3	16,7	66,7	-	0,0	16,7	16,7
SI	<b>100,0</b>	-	12,5	12,5	12,5	12,5	<b>100,0</b>	<b>95,8</b>	8,3	20,8	<b>100,0</b>	<b>66,7</b>	-	83,3	83,3	83,3	66,7	<b>66,7</b>	<b>83,3</b>	50,0	83,3	<b>66,7</b>
SU	<b>95,8</b>	-	<b>100,0</b>	62,5	62,5	<b>95,8</b>	<b>100,0</b>	91,7	16,7	<b>100,0</b>	<b>100,0</b>	<b>66,7</b>	-	<b>66,7</b>	0,0	0,0	<b>66,7</b>	<b>66,7</b>	50,0	0,0	<b>66,7</b>	<b>66,7</b>

(b) BruteForce-L1

	BK	DE	FA	GF	HA	MS	OR	SI	SB	ST	SU	BK	DE	FA	GF	HA	MS	OR	SI	SB	ST	SU
BF	8,3	4,2	8,3	8,3	8,3	8,3	8,3	8,3	0,0	8,3	8,3	83,3	50,0	33,3	50,0	50,0	83,3	50,0	33,3	50,0	33,3	50,0
BK	4,2	12,5	100,0	91,7	87,5	25,0	75,0	<b>95,8</b>	0,0	70,8	62,5	0,0	66,7	33,3	50,0	16,7	33,3	50,0	<b>66,7</b>	0,0	50,0	66,7
FR	0,0	-	0,0	0,0	0,0	0,0	0,0	-	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	-	0,0	0,0	0,0
OR	0,0	4,2	79,2	91,7	79,2	8,3	<b>95,8</b>	-	4,2	75,0	83,3	0,0	33,3	33,3	50,0	50,0	16,7	<b>50,0</b>	-	0,0	33,3	50,0

(c) BruteForce-Hamming

	BK	DE	FA	GF	HA	MS	OR	SI	SB	ST	SU	BK	DE	FA	GF	HA	MS	OR	SI	SB	ST	SU
BF	8,3	4,2	8,3	8,3	8,3	8,3	8,3	8,3	0,0	8,3	8,3	83,3	33,3	33,3	33,3	50,0	66,7	33,3	33,3	50,0	33,3	33,3
BK	4,2	12,5	100,0	91,7	79,2	20,8	75,0	<b>95,8</b>	4,2	62,5	62,5	0,0	66,7	33,3	50,0	16,7	16,7	50,0	<b>66,7</b>	0,0	50,0	66,7
FR	0,0	-	0,0	0,0	0,0	0,0	0,0	-	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	-	0,0	0,0	0,0
OR	0,0	12,5	79,2	91,7	75,0	25,0	<b>95,8</b>	-	4,2	70,8	83,3	0,0	33,3	33,3	50,0	50,0	16,7	<b>50,0</b>	-	0,0	33,3	50,0

(d) BruteForce-Hamming(2)

	BK	DE	FA	GF	HA	MS	OR	SI	SB	ST	SU	BK	DE	FA	GF	HA	MS	OR	SI	SB	ST	SU
SI	<b>100,0</b>	-	12,5	12,5	12,5	12,5	<b>100,0</b>	<b>100,0</b>	4,2	12,5	<b>100,0</b>	<b>66,7</b>	-	83,3	83,3	83,3	66,7	<b>50,0</b>	<b>83,3</b>	33,3	83,3	<b>66,7</b>
SU	<b>95,8</b>	-	<b>100,0</b>	29,2	37,5	<b>100,0</b>	<b>100,0</b>	<b>95,8</b>	16,7	<b>100,0</b>	<b>100,0</b>	<b>66,7</b>	-	<b>50,0</b>	0,0	0,0	<b>50,0</b>	<b>66,7</b>	<b>50,0</b>	0,0	<b>50,0</b>	<b>50,0</b>

(e) FlannBased

As combinações que obtiveram melhores acurácias considerando as duas transformações estão em destaque. As linhas são os descritores e as colunas os detectores: *BF* - BRIEF, *BK* - BRISK, *DE* - Dense, *FA* - FAST, *FR* - FREAK, *GF* - GFTT, *HA* - HARRIS, *OR* - ORB, *SB* - SimpleBlob, *SI* - SIFT, *ST* - STAR, *SU* - SURF.

para todos as dimensões de imagens testes. As combinações STAR+SURF+BruteForce, STAR+SURF+FlannBased e STAR+SURF+BruteForce-L1 obtiveram resultados satisfatórios de acurácia e menores tempos de processamento entre todas as combinações, exceto para a menor dimensão de imagem avaliada.

A Tabela 4.4 contém os resultados de acurácia e *frame rate* para as 18 melhores combinações nos testes de detecção de um padrão em vídeos de dimensões 960 x 540 e 480 x 360 pixels. Os métodos de detecção destas combinações apresentaram os melhores resultados de entropia, após o método Dense, constantes na Tabela 4.1, demonstrando que há uma relação entre entropia e acurácia para combinações que utilizaram estes detectores.

As combinações que obtiveram melhores resultados em relação ao *frame rate* foram SIFT+BRISK+BruteForce-Hamming(2) e SIFT+BRISK+BruteForce-Hamming, entretanto seus resultados de acurácia para os testes com as imagens de dimensões 960 x 540 pixels não foram satisfatórios. As combinações SIFT+SIFT+BruteForce, SIFT+SIFT

Tabela 4.3: Resultados de acurácia e tempo médio de acerto para detecção de um padrão em imagens.

Detector	Descritor	Algoritmo de correspondência	1280 x 720				960 x 540				480 x 360			
			Eixo Z		Eixo X		Eixo Z		Eixo X		Eixo Z		Eixo X	
			A	T	A	T	A	T	A	T	A	T	A	T
BRISK	SIFT	BruteForce	100,0	1,98	66,7	1,91	100,0	1,39	50,0	1,39	0,0	-	0,0	-
BRISK	SIFT	BruteForce-L1	100,0	1,95	66,7	1,88	100,0	1,37	66,7	1,37	12,5	0,55	0,0	-
BRISK	SIFT	FlannBased	100,0	1,89	50,0	1,85	100,0	1,36	66,7	1,36	0,0	-	0,0	-
BRISK	SURF	BruteForce	95,8	0,75	66,7	0,71	100,0	0,47	66,7	0,49	58,3	0,15	50,0	0,18
BRISK	SURF	BruteForce-L1	95,8	0,73	66,7	0,69	100,0	0,46	50,0	0,47	41,7	0,15	33,3	0,16
BRISK	SURF	FlannBased	95,8	0,7	66,7	0,66	100,0	0,46	50,0	0,47	58,3	0,16	33,3	0,17
FAST	SURF	BruteForce	100,0	6,04	50,0	6,96	100,0	2,93	66,7	2,92	95,8	0,63	66,7	0,61
FAST	SURF	BruteForce-L1	100,0	5,37	66,7	6,05	100,0	2,63	66,7	2,63	100,0	0,59	66,7	0,56
FAST	SURF	FlannBased	100,0	1,85	50,0	2,13	100,0	1,19	66,7	1,21	95,8	0,45	66,7	0,43
MSER	SURF	BruteForce	100,0	1,13	50,0	1,11	95,8	0,62	50,0	0,65	12,5	0,17	0,0	-
MSER	SURF	BruteForce-L1	95,8	1,13	66,7	1,1	91,7	0,62	66,7	0,63	8,3	0,16	0,0	-
MSER	SURF	FlannBased	100,0	1,13	50,0	1,12	95,8	0,63	50,0	0,66	12,5	0,17	0,0	-
ORB	ORB	BruteForce-Hamming	83,3	4,69	50,0	4,56	83,3	3,46	66,7	3,41	87,5	0,47	66,7	0,48
ORB	ORB	BruteForce-Hamming(2)	83,3	5,32	66,7	5,31	83,3	3,96	50,0	3,88	87,5	0,53	50,0	0,55
ORB	SIFT	BruteForce	100,0	22,14	33,3	22,06	70,8	19,81	33,3	19,85	0,0	-	0,0	-
ORB	SIFT	BruteForce-L1	95,8	21,05	50,0	21,09	79,2	19,08	16,7	19,05	0,0	-	0,0	-
ORB	SIFT	FlannBased	100,0	15,12	50,0	15,13	70,8	14,71	33,3	14,71	0,0	-	0,0	-
ORB	SURF	BruteForce	100,0	15,65	66,7	15,56	95,8	14,3	66,7	14,33	100,0	5,06	50,0	5,28
ORB	SURF	BruteForce-L1	100,0	15	66,7	14,95	100,0	13,84	66,7	13,86	100,0	5	50,0	5,22
ORB	SURF	FlannBased	100,0	11,72	66,7	11,66	95,8	11,43	66,7	11,49	100,0	4,8	50,0	5,01
SIFT	BRISK	BruteForce-Hamming	95,8	1,42	50,0	1,49	91,7	0,77	50,0	0,77	100,0	0,17	66,7	0,17
SIFT	BRISK	BruteForce-Hamming(2)	95,8	1,5	50,0	1,6	91,7	0,82	50,0	0,82	100,0	0,17	66,7	0,17
SIFT	SIFT	BruteForce	100,0	2,35	83,3	2,47	100,0	1,37	83,3	1,37	100,0	0,36	66,7	0,36
SIFT	SIFT	BruteForce-L1	95,8	2,26	83,3	2,38	100,0	1,33	83,3	1,32	100,0	0,36	83,3	0,36
SIFT	SIFT	FlannBased	100,0	1,93	66,7	2,04	100,0	1,18	83,3	1,19	100,0	0,36	66,7	0,37
SIFT	SURF	BruteForce	95,8	1,85	50,0	2,37	87,5	1,22	66,7	1,29	83,3	0,39	66,7	0,39
SIFT	SURF	FlannBased	95,8	1,62	50,0	2,05	83,3	1,11	66,7	1,18	83,3	0,39	50,0	0,38
STAR	SURF	BruteForce	100,0	0,43	50,0	0,43	100,0	0,25	50,0	0,26	0,0	-	0,0	-
STAR	SURF	BruteForce-L1	100,0	0,43	66,7	0,44	95,8	0,25	66,7	0,28	0,0	-	0,0	-
STAR	SURF	FlannBased	100,0	0,43	50,0	0,43	100,0	0,26	50,0	0,27	0,0	-	0,0	-
SURF	SIFT	BruteForce	100,0	13,59	66,7	12,7	100,0	7,8	66,7	7,48	8,3	2,06	16,7	2,03
SURF	SIFT	BruteForce-L1	100,0	13,4	66,7	12,5	100,0	7,74	66,7	7,39	25,0	1,99	50,0	2,04
SURF	SIFT	FlannBased	100,0	12,57	66,7	11,75	100,0	7,5	66,7	7,2	8,3	2,06	33,3	2,01
SURF	SURF	BruteForce	100,0	4,77	66,7	4,58	95,8	2,58	83,3	2,42	87,5	0,58	66,7	0,58
SURF	SURF	BruteForce-L1	100,0	4,65	66,7	4,49	100,0	2,54	66,7	2,51	95,8	0,58	66,7	0,58
SURF	SURF	FlannBased	100,0	4,19	66,7	4,06	95,8	2,41	83,3	2,28	87,5	0,58	50,0	0,61

Legenda: *A* - Acurácia (%), *T* - Tempo médio de acerto (s).

+FlannBased e SIFT+SIFT+BruteForce-L1 obtiveram os melhores resultados de acurácia, entretanto, os seus melhores resultados de *frame rate* foram entre 2,2 e 1,6 vezes menores que os resultados das combinações com melhor *frame rate*.

Das 18 combinações constantes na Tabela 4.4, 16 utilizam pelo menos um dos componentes dos métodos SIFT ou SURF, que são métodos patenteados, portanto, dependem de autorização dos detentores da patente para uso comercial. As duas combinações que utilizam somente métodos livres são as ORB+ORB+BruteForce-Hamming e ORB+ORB+BruteForce-Hamming(2) que obtiveram resultados intermediários de acurácia e *frame rate*.

Como o método ORB, em trabalho anterior (MUKHERJEE; WU; WANG, 2015), obteve desempenho superior a outros métodos considerando tempo de processamento,

Tabela 4.4: Resultados de acurácia e *frame rate* para detecção de um padrão em vídeo

Detector	Descritor	Algoritmo de correspondência	960 x 540				480 x 360			
			Eixo Z		Eixo X		Eixo Z		Eixo X	
			A	F	A	F	A	F	A	F
FAST	SURF	BruteForce	97,8	0,34	54,4	0,41	99,4	1,61	58,9	1,76
FAST	SURF	BruteForce-L1	98,6	0,38	57,8	0,45	100,0	1,72	60,0	1,87
FAST	SURF	FlannBased	97,8	0,85	54,4	0,95	99,4	2,29	60,0	2,36
ORB	ORB	BruteForce-Hamming	90,6	0,15	52,2	0,19	95,6	2,31	46,7	2,23
ORB	ORB	BruteForce-Hamming(2)	90,3	0,14	51,1	0,17	96,1	2,05	44,4	2,01
ORB	SURF	BruteForce	99,7	0,05	61,1	0,05	96,1	0,22	41,1	0,22
ORB	SURF	BruteForce-L1	100,0	0,05	62,2	0,05	96,7	0,22	37,8	0,22
ORB	SURF	FlannBased	99,7	0,06	55,6	0,07	96,1	0,23	43,3	0,23
SIFT	BRISK	BruteForce-Hamming	90,6	1,59	58,9	1,51	98,6	6,91	60,0	5,43
SIFT	BRISK	BruteForce-Hamming(2)	91,4	1,50	55,6	1,49	98,3	6,70	60,0	5,28
SIFT	SIFT	BruteForce	100,0	0,80	67,8	0,87	100,0	3,13	63,3	3,02
SIFT	SIFT	BruteForce-L1	99,7	0,83	66,7	0,90	99,7	3,20	67,8	3,09
SIFT	SIFT	FlannBased	100,0	0,92	66,7	0,97	100,0	3,11	65,6	2,97
SIFT	SURF	BruteForce	89,4	0,95	50,0	1,08	82,2	3,07	52,2	3,30
SIFT	SURF	FlannBased	88,9	1,04	50,0	1,16	81,7	3,03	50,0	3,26
SURF	SURF	BruteForce	100,0	0,43	60,0	0,47	91,4	1,94	53,3	1,85
SURF	SURF	BruteForce-L1	100,0	0,43	56,7	0,48	91,4	1,95	52,2	1,86
SURF	SURF	FlannBased	100,0	0,46	60,0	0,50	91,4	1,91	50,0	1,83

Legenda: *A* - Acurácia (%), *F* - *Frame rate* (fps).

por ser um método livre e sua implementação no OpenCV permitir até 8 parâmetros configuráveis, foram realizados testes variando o valor máximo de pontos de interesse que o detector ORB pode encontrar dentro do intervalo [10, 10.000] em busca dos melhores resultados de tempo de processamento e acurácia. Nas Tabelas 4.5 e 4.6 constam os menores limites de pontos de interesse que obtiveram os melhores resultados de acurácia e tempo de processamento nos testes com imagens e vídeos, respectivamente, de dimensões 480 x 360 pixels.

Tabela 4.5: Resultados das combinações do método ORB em imagens

Combinação	Eixo de rotação	Limite de pontos de interesse	A	T
ORB+ORB+BruteForce-Hamming	Z	1990	95,8	0,14
	X	910	66,7	0,09
ORB+ORB+BruteForce-Hamming(2)	Z	2400	95,8	0,19
	X	1840	66,7	0,18

Legenda: *A* - Acurácia (%), *T* - Tempo de médio de acerto (s).

Comparando os resultados das Tabelas 4.3 e 4.5, as combinações ORB+ORB+BruteForce-Hamming e ORB+ORB+BruteForce-Hamming(2) obtiveram resultados de acurácia e tempo de médio de acerto superiores quando o número de pontos de interesse

Tabela 4.6: Resultados das combinações do método ORB em vídeos

Combinação	Eixo de rotação	Limite de pontos de interesse	A	F
ORB+ORB+BruteForce-Hamming	Z	3640	95,0	4,18
	X	4500	51,1	3,05
ORB+ORB+BruteForce-Hamming(2)	Z	3640	94,7	4,26
	X	4260	51,1	2,84

Legenda:  $A$  - Acurácia (%),  $F$  - *Frame rate* (fps).

foi limitado a um valor baixo, se comparado aos resultados destas combinações com limite suficientemente alto. O resultado de tempo médio de acerto para a combinação ORB+ORB+BruteForce-Hamming foi superior ao resultado da melhor combinação da Tabela 4.3.

Comparando os resultados das Tabelas 4.4 e 4.6, as combinações ORB+ORB+BruteForce-Hamming e ORB+ORB+BruteForce-Hamming(2) obtiveram resultados de acurácia equivalentes e os resultados de *frame rate* superiores quando o número de pontos de interesse foi limitado a um valor baixo, se comparado aos resultados destas combinações com limite suficientemente alto. Entretanto, esses resultados não foram superiores aos resultados das melhores combinações na detecção de um padrão em um vídeo da Tabela 4.6.

A diminuição do tempo de acerto médio e o aumento do *frame rate* eram esperados, pois limitando o número de pontos de interesse detectáveis, a detecção de pontos de interesse se torna mais rápida, influenciando no resultado total da detecção do padrão. Mas os resultados obtidos não foram suficientes para se obter uma taxa de *frame rate* alta o suficiente para obter uma fluidez na cena.

## 5 Conclusão e trabalhos futuros

A principal contribuição deste trabalho foi fornecer a comparação e a análise de combinações dos algoritmos presentes na biblioteca OpenCV que permita a detecção de padrões pré-estabelecidos em imagens e vídeos e determinar as combinações mais adequadas na utilização em sistemas de Realidade Aumentada. Todas as combinações de algoritmos detectores, descritores e de correspondência disponíveis na documentação da biblioteca livre OpenCV foram comparados para a detecção de um padrão em imagens e vídeo com aplicação de transformações de rotação e de dimensão.

Após os testes, os resultados demonstraram que, sem prévia configuração, as combinações do detector SIFT, descritor BRISK e os algoritmos de correspondência binários foram superiores na detecção de um padrão em imagens e vídeo sob as transformações estudadas, especialmente de menores dimensões.

O método ORB com ajustes nos parâmetros, se mostrou uma alternativa aos métodos patenteados, obtendo resultados satisfatórios, principalmente, na detecção de um padrão em imagens. Uma proposta de trabalho futuro seria a implementação do método ORB, e todas as classes envolvidas, retirando-se estruturas não utilizadas para a detecção de uma imagem padrão em uma cena e limitando o número de pontos de interesse por quadrante da imagem, o que poderia melhorar o desempenho deste método.

As taxas de *frame rate* encontradas nos resultados das combinações, nas condições avaliadas, não são suficientes para a determinação da posição de um padrão em um vídeo para aplicações em Realidade Aumentada em tempo real. Para contornar este problema seria necessário utilizar uma abordagem híbrida, utilizando, por exemplo, técnicas de fluxo ótico após o reconhecimento dos objetos de interesse, para manter um *frame rate* mais adequado a esse tipo de aplicação. A utilização desta abordagem híbrida é outra proposta para trabalho futuro, o que pode ser útil na obtenção de uma configuração que possibilite a criação de sistemas de Realidade Aumentada com marcadores reais em tempo real, utilizando somente os algoritmos presentes na biblioteca OpenCV.

## Bibliografia

- AGRAWAL, M.; KONOLIGE, K.; BLAS, M. R. *CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. 102–115 p. ISBN 978-3-540-88693-8.
- ALAHY, A.; ORTIZ, R.; VANDERGHEYNST, P. Freak: Fast retina keypoint. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. [S.l.: s.n.], 2012. p. 510–517. ISSN 1063-6919.
- AZUMA, R. et al. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, v. 21, n. 6, 2001.
- BAY, H.; TUYTELAARS, T.; Van Gool, L. SURF: Speeded Up Robust Features. In: LEONARDIS, A.; BISCHOF, H.; PINZ, A. (Ed.). *Computer Vision – ECCV 2006*. [S.l.]: Springer Berlin Heidelberg, 2006. v. 3951, p. 404–417. ISBN 978-3-540-33832-1.
- BOUCHIHA, R.; BESBES, K. Comparison of local descriptors for automatic remote sensing image registration. *Signal, Image and Video Processing*, v. 9, n. 2, p. 463–469, 2015. ISSN 1863-1703.
- CALONDER, M. et al. BRIEF: Binary Robust Independent Elementary Features. In: DANIILIDIS, K.; MARAGOS, P.; PARAGIOS, N. (Ed.). *Computer Vision – ECCV 2010*. [S.l.]: Springer Berlin Heidelberg, 2010. v. 6314, p. 778–792. ISBN 978-3-642-15560-4.
- CHAO, J. et al. Performance comparison of various feature detector-descriptor combinations for content-based image retrieval with JPEG-encoded query images. In: *2013 IEEE International Workshop on Multimedia Signal Processing, MMSP 2013*. [S.l.: s.n.], 2013. p. 29–34. ISBN 9781479901258.
- EL-GAYAR, M. M.; SOLIMAN, H.; MEKY, N. A comparative study of image low level feature extraction algorithms. *Egyptian Informatics Journal*, v. 14, n. 2, p. 175–181, 2013. ISSN 11108665.
- GONZALEZ, R. C.; WOODS, R. E. *Digital image processing*. Third edit. [S.l.]: Prentice Hall, 2002. 190 p. ISSN 0024094X. ISBN 0201180758.
- HARRIS, C.; STEPHENS, M. A Combined Corner and Edge Detector. *Proceedings of the Alvey Vision Conference 1988*, p. 147–151, 1988. ISSN 09639292.
- HEINLY, J.; DUNN, E.; FRAHM, J.-M. Comparative evaluation of binary features. In: *Computer Vision–ECCV 2012*. [S.l.]: Springer, 2012. p. 759–773. ISBN 3-642-33708-2.
- JIANG, D.; YI, J. Comparison and Study of Classic Feature Point Detection Algorithm. In: *2012 International Conference on Computer Science and Service System*. [S.l.: s.n.], 2012. p. 2307–2309. ISBN 978-0-7695-4719-0.
- KITCHENHAM, B. Procedures for performing systematic reviews. *Keele, UK, Keele University*, v. 33, n. TR/SE-0401, p. 28, 2004. ISSN 13537776.

- KURNIAWAN, S. F.; DARMA PUTRA, I. K. G.; SUDANA, A. A. K. Bone fracture detection using OpenCV. *Journal of Theoretical and Applied Information Technology*, Asian Research Publishing Network (ARPN), v. 64, n. 1, p. 249–254, 2014.
- LEUTENEGGER, S.; CHLI, M.; SIEGWART, R. Y. BRISK: Binary Robust Invariant Scalable Keypoints. In: *Proceedings of the 2011 International Conference on Computer Vision*. [S.l.]: IEEE Computer Society, 2011. p. 2548–2555. ISBN 978-1-4577-1101-5.
- LOWE, D. G. Object recognition from local scale-invariant features. In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. [S.l.: s.n.], 1999. v. 2, p. 1150–1157 vol.2.
- MAIR, E. et al. Adaptive and Generic Corner Detection Based on the Accelerated Segment Test. *Proceedings of the European Conference on Computer Vision (ECCV)*, p. 1–14, 2010.
- MATAS, J. et al. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, v. 22, n. 10, p. 761–767, 2004.
- MUJA, M.; LOWE, D. G. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. In *VISAPP International Conference on Computer Vision Theory and Applications*, v. 2, 2009.
- MUKHERJEE, D.; WU, Q. M. J.; WANG, G. A comparative experimental study of image feature detectors and descriptors. *Machine Vision and Applications*, v. 26, n. 4, p. 443–466, 2015. ISSN 14321769.
- NEIVA, F. W.; SILVA, R. L. d. S. da. Revisão Sistemática em Ciência da Computação - Guia Prático. *Universidade Federal de Juiz de Fora*, 2016.
- RODRIGUES, V. et al. Automatic tracking of red blood cells in micro channels using OpenCV. In: *AIP Conference Proceedings*. [S.l.: s.n.], 2013. v. 1558, p. 594–597.
- ROSTEN, E.; DRUMMOND, T. Machine Learning for High-Speed Corner Detection. In: LEONARDIS, A.; BISCHOF, H.; PINZ, A. (Ed.). *Computer Vision – ECCV 2006*. [S.l.]: Springer Berlin Heidelberg, 2006. v. 3951, p. 430–443. ISBN 978-3-540-33832-1.
- RUBLEE, E. et al. ORB: An efficient alternative to SIFT or SURF. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. [S.l.: s.n.], 2011. p. 2564–2571.
- SHI, J.; TOMASI, C. Good features to track. In: *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*. [S.l.: s.n.], 1994. p. 593–600. ISSN 1063-6919.
- SUAREZ, O. et al. *OpenCV essentials: acquire, process, and analyze visual content to build full-fledged imaging applications using OpenCV*. Birmingham, UK: Packt Publishing Ltd., 2014. ISBN 978-1-78398-424-4.
- TAUB, E. A. *Webcam Brings 3-D to Topps Sports Cards*. New York: [s.n.], 2009. Disponível em: <http://www.nytimes.com/2009/03/09/technology/09topps.html>. Acesso em: 10 de outubro de 2016.
- WU, J. et al. A Comparative Study of SIFT and its Variants. *Measurement Science Review*, v. 13, n. 3, p. 122–131, 2013. ISSN 1335-8871.