UNIVERSIDADE FEDERAL DA FRONTEIRA SUL
CAMPUS CHAPECÓ
CURSO DE CIÊNCIA DA COMPUTAÇÃO

RODRIGO LEVINSKI

DESENVOLVIMENTO DE UM SISTEMA PARA RECONHECIMENTO DE OBJETOS UTILIZANDO VISÃO COMPUTACIONAL

CHAPECÓ
2015

RODRIGO LEVINSKI

DESENVOLVIMENTO DE UM SISTEMA PARA RECONHECIMENTO DE OBJETOS UTILIZANDO VISÃO COMPUTACIONAL

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.

Orientador: Prof. Dr. Claunir Pavan

CHAPECÓ
2015

RODRIGO LEVINSKI

DESENVOLVIMENTO DE UM SISTEMA PARA RECONHECIMENTO DE OBJETOS UTILIZANDO VISÃO COMPUTACIONAL

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.

Orientador: Prof. Dr. Claunir Pavan

Este trabalho de conclusão de curso foi defendido e aprovado pela banca em: ____/____/____

BANCA EXAMINADORA:

_____
Dr. Claunir Pavan - UFFS

_____
Dr. Emílio Wuerges - UFFS

_____
Me. Adriano Sanick Padilha - UFFS

RESUMO

Visão computacional é a área da ciência da computação responsável por tornar possível a ideia de máquinas "enxergarem"e possui um grande número de aplicações. Na maioria dos casos são desenvolvidos algoritmos específicos para determinada abordagem, aplicados em ambientes controlados. Neste contexto, este trabalho apresentará o resultado de uma investigação sobre técnicas de visão computacional para correspondência(*Matching*), aplicadas a ambientes não controlados. Adicionalmente, será desenvolvida uma ferramenta computacional capaz de reconhecer objetos previamente constantes em uma base de dados. Para isso, são utilizadas técnicas baseadas em descritores locais invariantes a uma certa quantia de modificações no ambiente de captura. Este sistema fará uso do algoritmo *Scale Invariant Features Transform*(SIFT) para extração de tais descritores devido a robustez e invariância apresentada pelo mesmo em condições desfavoráveis. Será criada uma interface para possibilitar o cadastro de objetos que posteriormente poderão ser identificados pelo próprio sistema a partir da captura ou *upload* de uma imagem. Além disso serão realizados testes para validação da eficiência do mesmo a partir de métricas comuns na análise de desempenho de algoritmos deste tipo, tais como tempo de execução e porcentagem de acertos e erros.

Palavras-chave: Visão computacional. Reconhecimento. Pontos-chave. Descritores locais.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

1.1 Motivação

A visão computacional é o ramo da ciência da computação que faz uso de um grande número de outras áreas para possibilitar que máquinas possam interpretar situações e tomar decisões. Ou seja, por meio da utilização de técnicas de visão computacional torna-se possível a interpretação de imagens por sistemas computacionais artificiais implementados em computadores ou equipamentos de hardware desenvolvidos para tal finalidade.

Na prática, são desenvolvidos algoritmos e técnicas específicas para lidar com os diversos tipos de problemas que surgem nesse ramo da Ciência, pois não existe uma teoria padronizada ou suficientemente genérica para modelar todos os aspectos da percepção visual [Maia, 2010].

Devido ao fato da visão humana ser algo tão comum para maioria das pessoas, é fácil de ser enganado e pensar que a visão computacional é uma tarefa simples como tal. Nosso cérebro divide o sinal da visão em vários canais onde cada um assemelha um tipo diferente de informação para o mesmo, ele está sempre atento a qualquer variação do ambiente e é capaz de definir uma área de interesse para ser analizada enquanto ignora outras menos importantes.

Esta mesma tarefa que apresenta tanta simplicidade para nosso cérebro demanda de uma série de transformações gráficas dependendo do método envolvido e mais cálculos matemáticos baseados na imagem a ser identificada que precisa ser processada quadro a quadro para se conseguir um resultado que nem sempre é satisfatório.

Uma dos maiores obstáculos da visão computacional é a não existência de um método genérico para reconhecimento e tratamento de qualquer objeto ou ambiente, cada problema específico deve ser tratado através de uma abordagem específica desenvolvida especialmente para o problema em questão surgindo assim a necessidade de sempre se estar inovando e desenvolvendo aplicações nesta área que venham para facilitar a iteração entre humano e computador ou auxílio na realização de tarefas.

1.2 Tema

Este trabalho propõe o desenvolvimento de um sistema que seja capaz de reconhecer objetos e fazer o $matching$ com outros previamente cadastrados, para auxiliar em tarefas de

reconhecimento em atividades que envolvem um grande número de objetos. Pretende-se, com esse sistema, facilitar a tarefa de reconhecimento e identificação de um objeto em meio a outros com características parecidas.

1.3 Objetivos

1.3.1 Objetivo Geral

Desenvolver um sistema que utilize técnicas de visão computacional para reconhecimento de objetos em ambientes não controlados.

1.3.2 Objetivos Específicos

- Identificar técnicas para reconhecimento de objetos em ambientes não controlados;
- Identificar as restrições de cada técnica;
- Desenvolver uma ferramenta para reconhecimento de objetos a partir do processamento de imagens;
- Verificar e validar os resultados obtidos por meio de $benchmarks$ e gráficos.

1.4 Justificativa

Ao pensar em um sistema que possa reconhecer objetos logo surge a ideia de uma linha de produção ou de processos automatizados capazes de compreender seu ambiente de trabalho e efetuar tarefas repetitivas ou de alto risco. Hoje em dia cresce cada vez mais a iteração entre o homem e os computadores, $smartphones$, entre outros dispositivos.

Todavia ao se buscar por sistemas eficientes de reconhecimento acaba-se por se deparar com a dificuldade de encontrá-los ou de usá-los. Surge assim a ideia de um sistema que seja capaz de identificar objetos com um custo computacional aceitável, que seja simples de usar e funcional.

Segundo [Lowe, 1999], o reconhecimento de objetos em cenas reais desordenadas requer características locais na imagem que sejam invariantes mesmo pela desordem dos seus arredores ou oclusão parcial. Cenas como essas estão entre os principais casos de usabilidade

de ferramentas como essa, além do uso para distinção de objetos parecidos mas com características únicas que os diferem.

Para isso, o trabalho em questão propõe, além de um estudo sobre as técnicas de reconhecimento de objetos, o desenvolvimento de um sistema que utilize os conhecimentos adquiridos no curso, e especificamente sobre visão computacional, o qual possa ser utilizado em computadores pessoais ou até mesmo celulares como ferramenta de auxílio para outras aplicações ou diretamente em uma específica para reconhecimento de objetos.

O reconhecimento de objetos pode ser interpretado de diferentes formas de acordo com a abordagem necessária, por exemplo, reconhecer pode ser a resposta para uma pergunta do tipo: o objeto "X" encontra-se na imagem atual? Neste caso busca-se por uma segmentação do objeto quanto ao restante do contexto e uma identificação. Outra abordagem comum trata de responder a pergunta: que objeto é este? Onde um objeto "Y" é posto em frente a câmera responsável pela captura das imagens e busca-se efetuar o $matching$ do objeto com um banco de imagens de objetos pré cadastradas e caso encontre uma combinação o sistema retorna o nome do mesmo entre outras características, as quais também foram cadastradas previamente.

O sistema proposto, com base em técnicas de visão computacional, efetua a identificação de um objeto apresentado ao mesmo por meio de comparações com o banco de imagens e características pré cadastradas. Essa identificação servirá para facilitar tarefas do usuário, tal como, a identificação de um objeto que precisa ser encontrado em um almoxarifado com grande variedade de produtos, produzindo como saída a localização do mesmo em um depósito bem como sua especificação.

Pretende-se, com o desenvolvimento deste sistema, criar uma interface onde possam ser cadastrados os objetos a serem reconhecidos futuramente. Com os objetos já cadastrados será apresentada outra interface no mesmo programa que ofereça o reconhecimento de um objeto quando apresentado para câmera ou efetuado o carregamento de uma imagem do mesmo. Como resultado será apresentado uma saída positiva com a descrição do objeto reconhecido ou negativa com uma mensagem.

1.5 Estrutura do Trabalho

A estruturação deste trabalho se dá por meio de seis capítulos. No capítulo inicial é apresentada a motivação, seguida pela definição do tema, objetivos geral e específicos, e a justificativa. Nos capítulos 2 e 3 apresentam o referencial teórico, o capítulo 2 voltado a visão

computacional de forma geral, enquanto o capítulo 3 volta-se ao método escolhido para implementação. O capítulo 4 apresenta os trabalhos relacionados. No capítulo 5 apresenta-se a proposta de desenvolvimento. O capítulo final apresenta a metodologia utilizada na etapa de revisão bibliográfica e estudos bem como as etapas de desenvolvimento do projeto. Ao final destes seis capítulos são apresentadas as referências bibliográficas que serviram de base para construção deste projeto.

2 VISÃO COMPUTACIONAL

A visão computacional é o ramo da Ciência da Computação que reúne todas as teorias e tecnologias desenvolvidas com a finalidade de possibilitar que imagens sejam interpretadas por sistemas artificiais implementados em computadores [Maia, 2010]. Ou seja, várias são as ferramentas necessárias para se conseguir que um computador possa "enxergar"e interpretar, técnicas essas executáveis por meio de hardware e software com inúmeras abordagens e aplicações.

Neste capítulo serão abordados os conhecimentos necessários para a compreensão e validação do projeto. Apresenta-se aqui o intuito da visão computacional, aplicabilidades, principais conceitos e ao final deste capítulo algumas das técnicas mais desenvolvidas que fazem o uso de descritores locais.

A visão humana é sem dúvida um dos sentidos mais importantes e complexos de nosso corpo, ela nos permite perceber e entender o mundo ao nosso redor, tarefa que para nós parece ser tão simples, mas quase nunca paramos para pensar em todo processo cerebral envolvido neste ato. A visão computacional tenta replicar o processo efetuado em nosso cérebro por meio de processamento eletrônico, percebendo e entendendo a imagem [Sonka et al., 2007]. Mas fazer com que os computadores ganhem essa habilidade de "enxergar"não é uma tarefa tão simples.

Vivemos em um mundo tri-dimensional (3D), e quando um computador tenta analisar objetos em um espaço 3D nos deparamos com o fato de que as tecnologias disponíveis costumam trabalhar com imagens bi-dimensionais (2D),ou seja, uma projeção para um número menor de dimensões, o que acarreta em uma grande perca de informações. Cenas dinâmicas tais como estamos acostumados a ver, com objetos em movimento, mudança de cores, entre outras variações tornam a visão computacional ainda mais complicada [Sonka et al., 2007].

Com o intuito de melhorar os resultados das técnicas tornam-se necessários métodos matemáticos, inteligência artificial (IA) entre outras disciplinas científicas como já citado anteriormente. Com o intuito de simplificar este processo de compreensão da visão computacional, a mesma é comummente subdividida em dois níveis distintos; processamento de imagens: *baixo nível*, e compreensão de imagens: *alto nível*.

Métodos *baixo nível* costumam tomar pouco conhecimento sobre o contexto da imagem, geralmente preparados somente para efetuar uma tarefa específica, por exemplo, a descrição de

uma imagem 2D, a partir da captura da mesma, descrevendo seu brilho com a representação em uma matriz de sua escala cinza, onde cada elemento da matriz corresponde ao brilho em uma localização específica da imagem.

Por outro lado a visão computacional de processamento *alto nível* é baseada em conhecimentos específicos e aprofundados com objetivos finais bem definidos. Nesta parte métodos de inteligência artificial são utilizados da maioria dos casos, servindo como ferramenta principal no melhoramento dos resultados finais.

A visão computacional de *alto nível* tenta imitar ao máximo a cognição humana e nossa habilidade de tomar decisões de acordo com as informações contidas na imagem a ser processada. Neste nível tenta-se manter um modelo formal o mais parecido possível com o original, aplica-se aqui um laço de conhecimento interativo para compreensão da imagem, o qual eventualmente converge para o objetivo final.

Conceitos aplicados específicos são necessários para que se consiga um sistema de visão computacional funcional e efetivo, os quais serão apresentados nesta seção com o intuito de esclarecer o funcionamento do sistema citado relatando suas fases principais e seus respectivos objetivos.

2.1 Sistema de Visão Artificial (SVA)

2.1.1 Imagem Digital

Atualmente, efetuar a captura de uma imagem é um processo cada vez mais comum, menores e melhores sensores com funcionalidades robustas vem sendo desenvolvidas para uso em celulares, *smartphones* entre outros, possibilitando o uso em diversas aplicações tais como: biometria, vigilância, transmissão de vídeo, diagnóstico médico, etc.

O processamento de uma imagem digital consiste de um conjunto de algoritmos envolvendo operações matemáticas aplicadas sobre a captura para gerar uma imagem alvo, representação ou descrição de conteúdo, processos estes também realizados em um sistema de visão computacional.

Uma imagem monocromática pode ser descrita matematicamente por uma função $f(x, y)$ da intensidade luminosa, sendo seu valor, em qualquer ponto de coordenadas espaciais (x, y), proporcional ao brilho (ou nível de cinza) da imagem naquele ponto [Marques Filho and Vieira Neto, 1999]. A Figura 2.1 apresenta uma exemplo de representação de uma imagem

monocromática, onde são usados apenas tons de cinza para diferenciar a intensidade luminosa em cada pixel, a mesma demonstra também a convenção mais comum utilizada na representação de imagens no espaço 2D.

Figura 2.1: Imagem monocromática e a convenção mais comum utilizada para sua representação do par de eixos (x, y) [Marques Filho and Vieira Neto, 1999].

Imagens monocromáticas são as mais usadas no âmbito da visão computacional devido ao número reduzido, mas mesmo assim suficiente, de informações e variáveis contidas nas mesmas. Porém caso uma imagem possua informações em intervalos ou bandas distintas de frequência, é necessário uma função $f(x, y)$ para cada banda. É o caso das imagens coloridas do padrão *RGB* formado pela informação das cores primárias aditivas: vermelho (*Red*), verde (*Green*) e azul (*Blue*) [Marques Filho and Vieira Neto, 1999].

Ao efetuar a digitalização de uma imagem capturada a mesma assume um tamanho adimensional, em pixeis. Mas pode-se conhecer uma medida qualitativa da amostragem, conhecendo-se a razão entre o número de pixeis obtido e o tamanho da imagem real no filme ou equivalente. Isso se chama Resolução. A existência de 2 dimensões, permite definir uma resolução horizontal e uma vertical, as quais seguem uma razão de forma a manter a qualidade de uma imagem.

2.1.2 Estrutura de um Sistema de Visão Artificial

Um sistema de visão artificial (SVA), ou sistema de visão computacional, é constituído de diversas etapas, cada qual com suas peculiaridades e papéis definidos com objetivo de processar e interpretar imagens capturadas de cenas reais. Como parte do processo de contextualização, a seguir serão apresentadas as etapas principais de um SVA. Partindo da premissa de que cada problema prático possui uma abordagem específica e métodos diferentes de serem tratados.

De modo representativo, a Figura 2.2 apresenta as principais etapas de um sistema de visão artificial, as quais serão explanadas nas sub-seções seguintes.

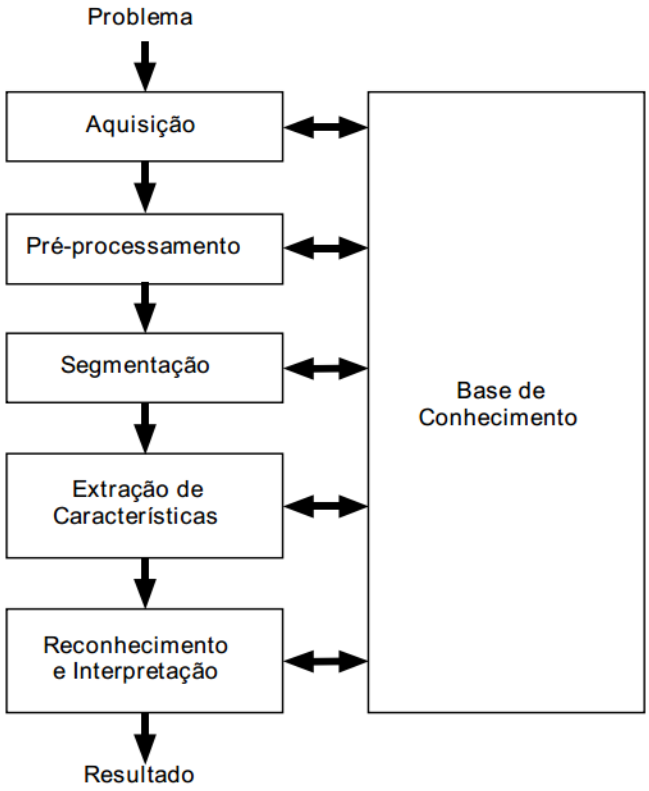

Figura 2.2: Principais etapas de um Sistema de Visão Artificial (SVA) [Marques Filho and Vieira Neto, 1999].

2.1.3 Definição do problema

A definição do problema, em um SVA, consiste em delimitar qual será o problema a ser tratado bem como a especificação dos resultados que pretende-se atingir. As premissas definidas nesta etapa servirão como base para toda construção do SVA, mas isso não significa que as mesmas serão imutáveis, dependendo do SVA mudanças podem ocorrer no decorrer do

processo de desenvolvimento.

Para fins de demonstração, pode se usar como exemplo a leitura da placa de um lote de veículos como problema a ser solucionado. O domínio do problema, neste caso, consiste no lote de placas e o objetivo do SVA é ler as letras e os números contidos em cada uma delas. Desse modo, o que espera como resultado é uma sequência formada por três letras e quatro dígitos correspondente a placa lida.

2.1.4 Aquisição da Imagem

Entrando definitivamente no processo computacional, o primeiro passo é efetuar a aquisição de imagens das placas. Para isso tornam-se necessários um sensor para capturá-las e um digitalizador para converter a imagem analógica, captada pelo sensor, em imagem digital.

Este processo apresenta aspectos de projeto que precisam ser definidos, dentre eles pode-se mencionar a escolha do tipo do sensor, as condições do ambiente onde serão efetuadas as capturas, a forma com que cada captura sera efetuada (definindo se seriam placas avulsas ou presas a veículos), a resolução necessária para uma boa representação entre outros. Após definidas essas condições, esta etapa produz como saída uma imagem digitalizada da placa.

2.1.5 Pré-processamento

Com a imagem do passo anterior em mãos, pode acontecer da mesma apresentar imperfeições derivadas do processo de digitalização. De modo a minimizar estas imperfeições a etapa de pré-processamento busca aprimorar a qualidade da imagem digitalizada para as etapas seguintes do SVA.

As técnicas de processamento devem ser escolhidas de acordo com sua finalidade e importância para o resultado esperado, em imagens coloridas, percebe-se com certa facilidade a importância da etapa de pré-processamento, pois em um reconhecimento de padrões desse tipo a cor pode ser um descritor fundamental do objeto simplificando assim sua segmentação.

Um exemplo de filtro de pré-processamento pode ser visto na Figura 2.3, o filtro em questão é chamado de Filtro da Média, cujo objetivo é diminuir os níveis de ruído de uma imagem, percebe-se na imagem a esquerda uma suavização entre as diferenças de cada pixel.

Figura 2.3: Aplicação do Filtro da Média. À esquerda a imagem original, à direita a imagem resultante da aplicação do filtro [Selhorst, 2014].

2.1.6 Segmentação

A etapa de segmentação tem como principal objetivo dividir a imagem em múltiplas regiões de interesse formadas por conjuntos de pixeis com características parecidas, com objetivo de encontrar uma representação que facilite a análise por meio do realce de objetos que a compõem. Esta tarefa, apesar de simples de descrever, é das mais difíceis de implementar [Marques Filho and Vieira Neto, 1999].

Especificamente no problema da identificação de placas, o problema pode ser representado pelas etapas de localização dos caracteres da placa e segmentação de cada caractere de forma individual. Processo este que gera uma sequência de imagens que representam cada caractere da placa e provavelmente algumas outras áreas da imagem que não são de interesse.

Na Figura 2.4 pode-se observar um exemplo de segmentação aplicando-se a técnica de Limiarização em duas intensidades de cores diferentes. A Figura 2.4(A) representa a imagem original, já na Figura 2.4(B) foi aplicado o algoritmo Limiarização para cores abaixo do tom 128, enquanto na Figura 2.4(C) o mesmo algoritmo foi aplicado para cores com tom acima de 128.

2.1.7 Extração de Características

O próximo passo em um SVA, após a segmentação dos objetos, é definir quais características úteis serão extraídas, e em seguida extraí-las através de descritores que permitam caracterizar de forma mais singular possível cada objeto identificado na imagem. No caso das

Figura 2.4: Exemplo processo de Limiarização com diferentes parâmetros (A) Imagem original (B) Limiarização abaixo da cor 128 (c) Limiarização acima da cor 128.

placas de veículos, busca-se caracterizar cada caractere com características que apresentem um bom poder de discriminação perante os demais.

Esta etapa ainda trabalha com uma imagem como entrada, porém como saída produz um conjunto de dados referentes àquela imagem. Para maior clareza, suponhamos que os descritores utilizados para descrever um caractere sejam as coordenadas normalizadas x e y de seu centro de gravidade e a razão entre sua altura e largura. Neste caso, um vetor de três elementos é uma estrutura de dados adequada para armazenar estas informações sobre cada dígito processado por esta etapa [Marques Filho and Vieira Neto, 1999].

2.1.8 Reconhecimento e Interpretação

Baseando-se nas características de um objeto traduzidas por seus descritores, na ultima etapa do SVA efetua-se o reconhecimento e a atribuição de um rótulo para o objeto, no processo denominado reconhecimento. Por outro lado a tarefa de interpretação, consiste em atribuir significado a um conjunto de objetos reconhecidos.

No contexto do reconhecedor de placas, o processo de interpretação seria responsável por efetuar a verificação se as placas reconhecidas são válidas, descobrindo assim se o resultado da extração de características faz sentido ou não.

2.1.9 Base de Conhecimento

Trabalhando sempre em conjunto com todas as etapas descritas acima pressupõem-se a existência de um conhecimento específico sobre o problema a ser resolvido, cuja complexidade

pode variar enormemente dependendo da aplicação. Idealmente, esta base de conhecimento deveria não somente guiar o funcionamento de cada etapa, mas também permitir a realimentação entre elas [Marques Filho and Vieira Neto, 1999].

Quanto maior for a iteração e realimentação entre as etapas, maiores são as chances de se alcançar melhores resultados. A integração entre as várias etapas ainda é um objetivo difícil de se alcançar e não está presente na maioria dos SVAs existentes atualmente [Marques Filho and Vieira Neto, 1999].

2.2 Descritores locais em SVAs

O desenvolvimento de um SVA envolve uma série de técnicas, aplicadas diferentemente para cada abordagem de problema. Cada etapa das citadas na seção anterior possui um leque diferente de métodos que se saem melhor conforme o objetivo do SVA. Neste projeto, serão utilizadas técnicas referentes ao uso de descritores locais, também conhecidos como pontos de interesse ou pontos-chave, o quais tem a finalidade de encontrar por pontos de um objeto registrado em imagem que sejam capazes de caracterizá-lo unicamente.

Nesta seção apresenta-se as técnicas mais difundidas e utilizadas em SVAs que utilizam a ideia de descritores locais e/ou detecção de pontos de interesse que sejam invariantes mesmo após algumas transformações em escala, ângulo de visão ou mudança de proporção.

2.2.1 Métodos *Harris-Laplace* e *Hessiano-Laplace*

Características locais tem se mostrado bem adaptadas as tarefas de correspondência e reconhecimento de imagens, uma vez que possuem robustez mesmo com visibilidade e desorganização parcial. A dificuldade ainda está na obtenção de invariância sob condições de visualização arbitrárias [Mikolajczyk and Schmid, 2002].

Neste contexto, [Mikolajczyk and Schmid, 2002], adaptaram os métodos baseados na métrica de Harris e no determinante da matriz Hessiana para que esses detectores fossem utilizados no contexto do espaço de escalas, de forma a obter métodos de detecção invariantes a escala, os quais são denominados pelos autores como *Harris-Laplace* e *Hessiano-Laplace*. Em ambos os casos trabalha-se com detecção de pontos-chaves de maneira não supervisionada com a proposta de um algoritmo iterativo para auto-adaptação de uma elipse ao formato local das estruturas detectadas.

2.2.1.1 Harris-Laplace

O detector de Harris, proposto por [Harris and Stephens, 1988], baseia-se na segunda matriz de momento, também chamada de matriz de auto-correlação, a qual é usada para detecção de características e para descrição de estruturas locais em uma imagem [Tuytelaars and Mikolajczyk, 2008]. Esta matriz descreve a distribuição do gradiente de cores na vizinhança local de um ponto, ou no caso de imagens de um pixel, por meio do cálculo de *eigenvalues*.

Seguindo essa mesma ideia o detector Harris-Laplace apresenta uma evolução do detector bidimensional original de Harris para o espaço e escala, a qual é uma das limitações mais conhecidas desse método. Basicamente, a matriz Hessiana é derivada para o caso da representação via L_t, obtendo-se uma métrica M_t adaptada á escala local considerando-se $\mathbf{H}_{L_t}$

$$M_t = det(\mathbf{H}_{L_t}) - \alpha Tr^2(\mathbf{H}_{L_t}) \tag{2.1}$$

A detecção de cantos ocorre da mesma maneira que o caso 2D mesmo após esta transformação, porém agora iterando sobre várias escalas L_{t_i} para efetuar a seleção automática de escala. O valor obtido de M_t é comparado com seus 8 vizinhos para efetuar a supressão de não-máximos restando apenas o ponto de máxima, quando existir, determinando que ali é uma extremidade [Tuytelaars and Mikolajczyk, 2008].

2.2.1.2 Hessiano-Laplace

Seguindo a ideia do método Harris-Laplace temos o detector de pontos Hessiano-Laplace, o qual funciona de forma semelhante ao mesmo, porém com uma diferença na métrica utilizada para detecção. No caso, em uma escala t, os extremos locais dos valores do Laplaciano e do Hessiano são detectados simultaneamente.

Ao maximizar o valor do Hessiano, essa abordagem penaliza estruturas muito longas com valores muito pequenos no resultado da expressão, caracterizando uma mudança de sinal. O detector de Hessiano-Laplace apresenta melhores resultados do que o Harris-Laplace [Mikolajczyk and Schmid, 2005]

2.2.2 Método *Speeded-Up Robust Features* (SURF)

Dentre o métodos mais comuns para extração de detectores e descritores locais para imagens, encontra-se o método SURF (Speeded-Up Robust Features), conhecido pelas carac-

terística de ser invariante a escala e rotação. SURF se aproxima ou até mesmo ultrapassa a performance de outros esquemas propostos anteriormente, devido a sua repetibilidade, distinção e robustez na computação e comparação de características em baixo tempo [Bay et al., 2008].

A detecção de pontos-chaves do método SURF explora o uso de imagens integrais para computar eficientemente uma aproximação do operador HoG (Haar of Gaussians) na qual *wavelets* de *Haar* são usadas para computar uma aproximação das derivadas de segunda ordem do núcleo Gaussiano) em diferentes escalas, o que lhe confere um desempenho, falado-se em tempo, de 3 a 7 vezes melhor do que o apresentado pelo método SIFT.

Apesar do alto desempenho deste método, os próprios autores em [Bay et al., 2008], afirmam que seu método reporta pontos-chaves tão estáveis quando os decorrentes do uso de SIFT, porém por se tratar de um modo aproximativo do espaço de escala sua taxa de identificação é inferior ao SIFT, sendo assim SIFT pode ser considerado mais robusto.

2.2.3 Introdução ao método SIFT (*Scale-Invariant Features Transform*)

Precursor e referência na área de detecção de características locais, tem-se o algoritmo SIFT [Lowe, 1999] que efetua a busca por pontos-chaves por meio de cálculos de diferença das Gaussianas (DoG), o mesmo é conhecido pela suas caraterísticas de invariância de escala, rotação, iluminação e ponto de visão.

Devido principalmente a essas características, este método foi selecionado para o desenvolvimento deste projeto, pois possuem grande poder de distinção entre objetos encaixando-se na ideia de ambientes não controlados ou com uma série de objetos distintos e com mudanças de visualização.

A partir da base de conhecimento montada no capítulo atual, onde foram apresentadas breves introduções sobre as principais características de um SVA e de algoritmos para detecção de pontos de interesse, será apresentado, no próximo capítulo, o método SIFT constituindo parte fundamental deste projeto e base para o desenvolvimento do restante do trabalho.

3 SCALE INVARIANT FEATURES TRANSFORM (SIFT)

O algoritmo para detecção e $matching$ de pontos de interesse em imagens SIFT foi proposto por *David G. Lowe* em 1999 e patenteado no EUA pela *Univesity of British Columbia*. As principais aplicações de SIFT incluem reconhecimento de objetos, mapeamento e navegação de robôs, modelagem 3D, reconhecimento de gestos, acompanhamento em vídeo, etc.

SIFT é composto por duas etapas distintas: o detector e o descritor. As quais serão apresentadas as seções seguintes ao decorrer deste capítulo

3.1 Etapas do Algoritmo SIFT

Como citado anteriormente, o detector e o descritor constituem as duas principais etapas do algoritmo de SIFT. O detector SIFT é baseado em cálculos de diferença de Gaussianas e o descritor utiliza histogramas de gradientes orientados para descrever a vizinhança local dos pontos de interesse [Lowe, 2004]. Ambas etapas tem o objetivo de encontrar e descrever pontos-chave, respectivamente.

No decorrer destas duas etapas principais pode-se subdividi-las em sub-estágios, de maneira a entender de forma individual os procedimentos realizados no decorrer do algoritmo. Cada um destes estágios será explicado nas sub sessões a seguir e podem ser observados na Figura 3.1.

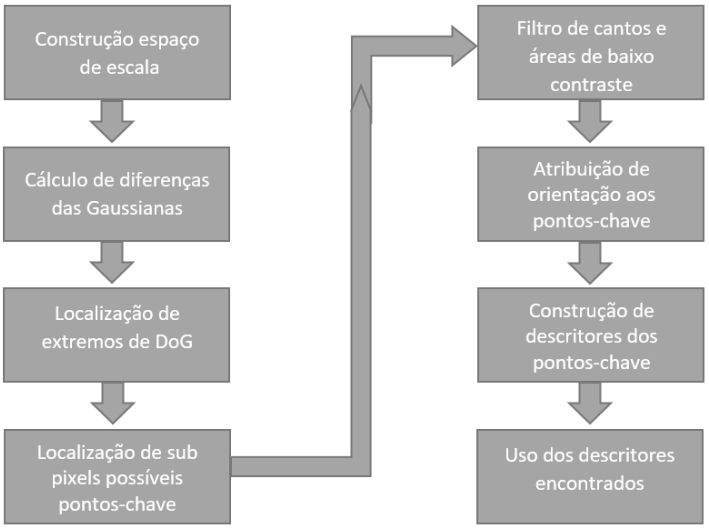

Figura 3.1: Principais estágios do algoritmo SIFT.

3.1.1 Detecção de extremos

O primeiro estágio do algoritmo SIFT, consiste em buscar por pontos que sejam invariantes a mudanças de escala da imagem, possibilitando a detecção de pontos com a câmera próxima ou distante do objeto de interesse. Tal objetivo é alcançado procurando características estáveis em diferentes escalas utilizando uma função no espaço de escala, que neste caso é a função Gaussiana.

Primeiramente uma imagem $I(x, y)$ passa a ser definida por uma função $L(x, y, \sigma)$, no espaço escala. Tal função é produzida pela convolução de uma função gaussiana $G(x, y, \sigma)$, com a Imagem, $I(x, y)$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{3.1}$$

onde $*$ é a operação de convolução em x e y, e

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} * e^{-(x^2+y^2/2\sigma^2)} \tag{3.2}$$

percebe-se que este filtro é variável à escala através do parâmetro σ [Lowe, 2004].

Para detectar eficientemente pontos-chave estáveis no espaço de escala utiliza-se uma função DoG(*"Difference if Gaussian*) formada pela diferença de imagens filtradas em escalas próximas, separadas por uma constante k. A definição da função DoG pode ser visto na equação 3.3, onde k é o fator da diferença entre duas escalas.

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma) \tag{3.3}$$

O resultado de efetuar a convolução de uma imagem com o filtro DoG é dado por

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \tag{3.4}$$

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{3.5}$$

A função DoG suaviza as imagens e pode ser calculada pela simples subtração de imagens borradas por um filtro Gaussiano em escalas σ e $k\sigma$.O objetivo da mesma é conseguir amostras de imagens onde detalhes indesejados e ruídos sejam eliminados e características fortes sejam realçadas [Lowe, 2004]. Variando σ torna-se possível encontrar tais características em diferentes escalas.

Durante os cálculos da função DoG utiliza-se uma representação que se parece com a computação de uma pilha de imagens contendo níveis de detalhe do espaço de escala linear em

um formato de pirâmide. Cada uma dessas pilhas em vários níveis de detalhe e é geralmente denominada como *oitavas de Gaussianas*. Cada nível f_i da pirâmide contém uma oitava obtida através da sub-amostragem sobre a oitava f_{i-1}, localizada no nível imediatamente inferior [Maia, 2010].

Este processo pode ser repetido indefinidamente até que se obtenha o nível de representação desejado, mas geralmente contém blocos de 8 x 8 pixeis. Esta estrutura é ilustrada pela Figura 3.2.

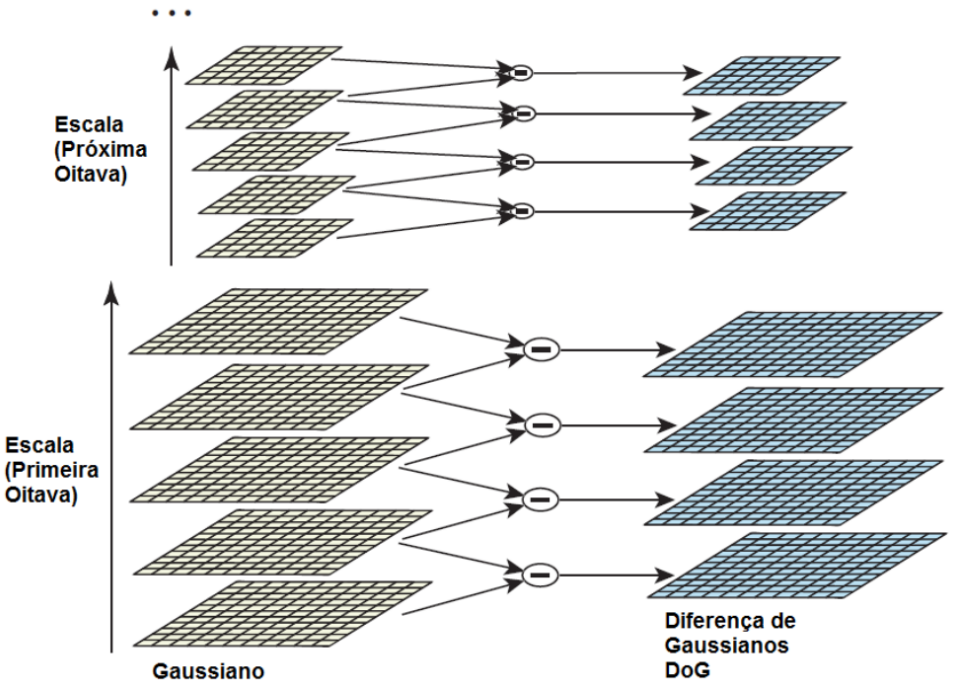

Figura 3.2: Representação do procedimento das Diferenças Gaussianas DoG para diversas oitavas de uma imagem. [Lowe, 2004]

Com o fim processamento das oitavas por meio da função DoG, parte-se para a etapa onde será feita a detecção de extremos em cada intervalo de cada oitava. Para um valor extremo pode-se tomar qualquer valor da função DoG de um pixel maior que todos os seus vizinhos no espaço-escala.

Tais extremos são dados por valores de máximo ou mínimo locais para cada $D(x, y, \sigma)$, que podem ser obtidos através da comparação de cada ponto, neste caso pixel, com seus oito vizinhos na sua escala e com os nove vizinhos na escala imediatamente inferior e superior, conforma representação da Figura 3.3.

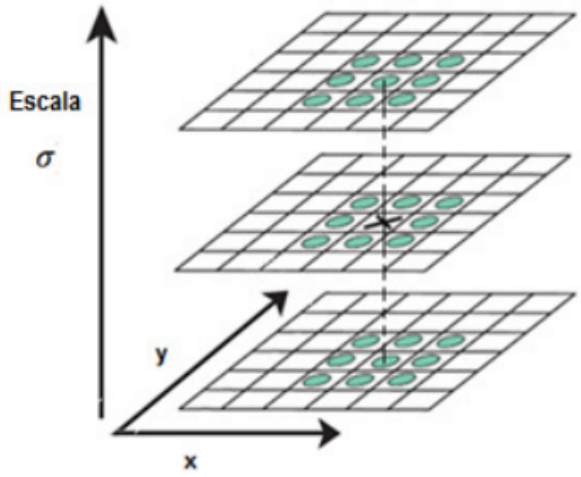

Figura 3.3: Detecção de máxima e minima da função DoG aplicada as imagens por meio da comparação do pixel X a seus 26 vizinhos da escala atual e das adjacentes [Lowe, 2004].

3.1.2 Localização de Pontos-Chave

Definidos os pontos de máxima e mínima, passa-se a etapa de definição de pontos-chave e descarte de pontos considerados instáveis. Toma-se como princípio a ideia de que todos os pontos detectados como extremos são candidatos a pontos-chave, aqui calcula-se a posição exata destes pontos e efetua-se a validação.

O método consiste em ajustar uma função quadrática 3D do ponto de amostragem local de modo a determinar uma localização interpolada do máximo. Esta abordagem utiliza as expansões de Taylor da função DoG aplicada a imagem, $D(x, y, \sigma)$, deslocada de modo que esteja localizada no ponto da amostragem [Brown and Lowe, 2002].

$$D(\bar{x}) = D + \frac{\partial D^T}{\partial \bar{x}}\bar{x} + \frac{1}{2}\bar{x}^T\frac{\partial^2 D}{\partial x^2}\bar{x}... \tag{3.6}$$

$$\bar{x} = (x, y, \sigma)^T \tag{3.7}$$

Onde o valor de D, a sua primeira e a segunda derivadas são calculadas no ponto de amostragem $\bar{x}$, representa o deslocamento deste ponto. A localização de $sub-pixeis$ do pondo de interesse é dada pelo extremo da função na equação 3.6. Esta localização, $\hat{x}$, é determinada ao ser calculada a derivada de $D(\bar{x})$ em relação a $\bar{x}$, e igualado o resultado a zero:

$$\frac{\partial D}{\partial \bar{x}} + \frac{\partial^2 D}{\partial \bar{x}^2}\hat{x} = 0 \tag{3.8}$$

Tem-se então a posição do extremo, dada por:

$$\hat{x} = -\frac{\partial^2 D^{T-1}}{\partial \bar{x}^2}\frac{\partial^2 D}{\partial \bar{x}} \tag{3.9}$$

O valor da função no extremo, $D(\bar{x})$, é útil para a rejeição de extremos instáveis com baixo constante, que seriam sensíveis a ruído. Substituindo-se a equação 3.9 na equação 3.6 obtém-se:

$$D(\hat{x}) = D + \frac{1}{2}\frac{\partial D^T}{\partial \bar{x}}\hat{x} \tag{3.10}$$

Segundo Lowe é aconselhável que se rejeitem valores de $|D(\hat{x})|$ inferiores a um determinado limiar, em [Lowe, 2004] tal limiar era 0.03, assumindo-se que os tons de cinza de cada pixel estejam normalizados em valores entre $[0, 1]$.

Mesmo com esse procedimento para eliminação de valores que não ultrapassam o limiar, a função DoG ainda apresenta valores altos ao longo de arestas fazendo com que estes pontos sejam escolhidos como pontos de interesse, o que não é desejável, tendo em vista que se busca por características invariantes as quais não se referem a arestas.

Para eliminação destes pontos-chave próximos à arestas Lowe propôs o uso de uma matriz Hessiana 2x2, H, computada na localização e escala dos pontos-chave na função D.

$$H(x, y) = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{3.11}$$

onde D_{xy} é a derivada de $D(x, y, \sigma)$ na localização e escala em relação a x e y; D_{xx} é a derivada segunda em relação a x; e D_{yy} é a derivada de segunda em relação a y. A matriz Hessiana representa assim uma segunda derivada, permitindo mensurar as magnitudes das curvaturas de D a partir de seus autovalores.

A partir da matriz H calcula-se a soma dos autovalores pelo traço de H e o produto pelo seu determinante:

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \tag{3.12}$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \tag{3.13}$$

Para o caso em que o determinante for negativo, as curvaturas possuem sinais diferentes, e o ponto é descartado, não sendo considerado um extremo.

Todo esse processo matemático acima citado tem como principal objetivo a eliminação de pontos coletados pela primeira etapa do algoritmo que se encontram em extremidades ou arestas, ou seja, que são fortes candidatos a sofrer influência de mudanças de escala, ponto de visão entre outras, as quais são a grande proposta do algoritmo final, assim a permanência destes pontos poderia comprometer os resultados.

3.1.3 Atribuição de Orientação dos Descritores

Entrando na parte do descritor do algoritmo de SIFT, faz-se a atribuição de uma orientação a cada ponto-chave encontrado até este estágio, que futuramente será utilizada para construção de descritores invariantes a rotação, e pode ser obtida através da análise de características locais da imagem.

Para cada amostragem da imagem na escala $L(x, y, \sigma)$, calcula-se a magnitude $m(x, y)$ e a orientação $\theta(x, y)$ do gradiente usando as diferenças de pixeis:

$$m(x,y) = \sqrt{((L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2)} \tag{3.14}$$

$$\theta(x,y) = tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right) \tag{3.15}$$

Monta-se então um histograma das orientações para pixeis em uma região vizinha ao redor do ponto-chave. O histograma possui 36 regiões, cobrindo todas orientações possíveis de $(0$ a $2\pi)$, como pode ser visto na representação de Figura 3.4

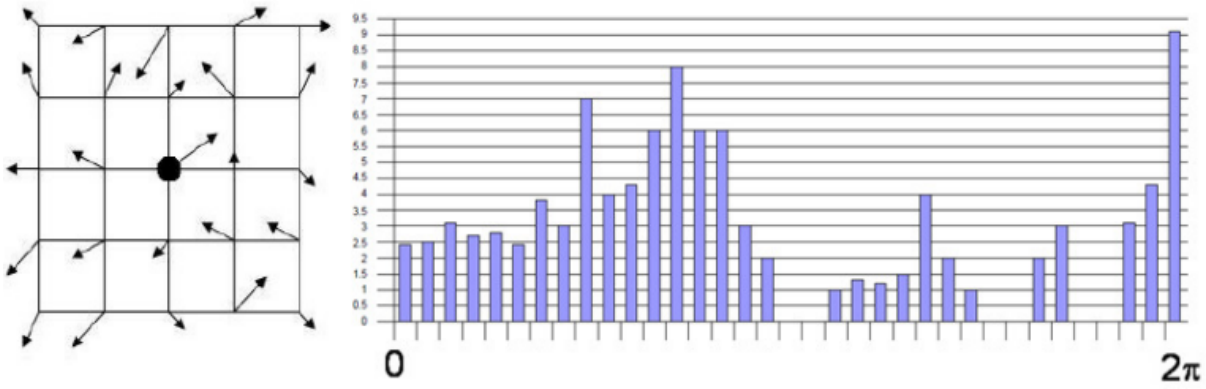

Figura 3.4: Histograma de orientações de um ponto-chave [Lowe, 2004].

Cada ponto da vizinhança do ponto-chave é adicionado ao histograma com um valor e um peso determinado, o primeiro é o valor da magnitude $m(x, y)$ de cada ponto adicionado, o segundo peso é dado por uma janela Gaussiana circular com σ' igual a 1,5 vezes maior que a escala do ponto chave, a qual pode ser definida pela seguinte equação:

$$g(\Delta x, \Delta y, \sigma') = \frac{1}{2\pi\sigma'^2} e^{-(\Delta x^2 + \Delta y^2)2\sigma'^2} \tag{3.16}$$

onde Δx e Δy são as distâncias entre cada ponto verificado e o ponto-chave.

Picos no histograma de orientação correspondem a orientações dominantes dos gradientes locais. Com base no máximo, mas não exclusivamente nele, define-se a orientação do ponto-chave. Picos que correspondem a pelo menos 80% do valor do pico máximo também são considerados, implicando que um ponto-chave poderá ter mais de uma orientação associada.

Quando a atribuição de muitas orientações acontece, o ponto-chave torna-se ainda mais estável para uma identificação futura. Ao final do processo uma parábola é usada para interpolar os três valores do histograma mais próximos do pico de forma a se obter uma melhor exatidão de sua posição [Lowe, 2004].

Ao final desta etapa cada ponto-chave contém 4 dimensões: sua posição x e y, magnitude e orientação. A partir de então pode-se partir para o próximo estágio do algoritmo.

3.1.4 Construção do Descritor Local

O próximo estágio do descritor de SIFT é efetuar a atribuição de um descritor invariante a iluminação e do ponto de vista 3D, tornando-os bem distinguíveis. Este descritor será calculado com base nos valores normalizados para cada ponto chave da imagem que passou por todas os estágios anteriores do algoritmo.

Com o intuito de montar descritores com invariância a rotação, é necessário que as orientações dos gradientes destes pontos sejam giradas de um ângulo correspondente à orientação do ponto-chave definida na seção anterior [Lowe, 2004].

O descritor do ponto-chave é então criado computando-se as magnitudes e orientações dos gradientes, que são amostrados ao redor da localização do ponto-chave. Tal procedimento pode ser visualizado na Figura 3.5.

Mesmo com a aplicação dessas técnicas duas imagens de um mesmo objeto podem possuir variações de luminosidade que modifiquem sensivelmente os descritores obtidos, este problema é solucionado efetuando a normalização do descritor, assim o mesmo terá característica de invariância à luminosidade.

Para cada imagem são construídos diversos descritores usando as ferramentas apresentadas até então, e cada um deles é referente a exclusivamente um ponto-chave. Tem-se como resultado, portanto, um conjunto de descritores robustos que podem ser utilizados para fazer a correspondência entre imagens, processo que será detalhado na próxima seção. Todos os detalhes sobre o processo de construção dos descritores SIFT e encontro de pontos-chave são encontrados em [Lowe, 1999] e [Lowe, 2004].

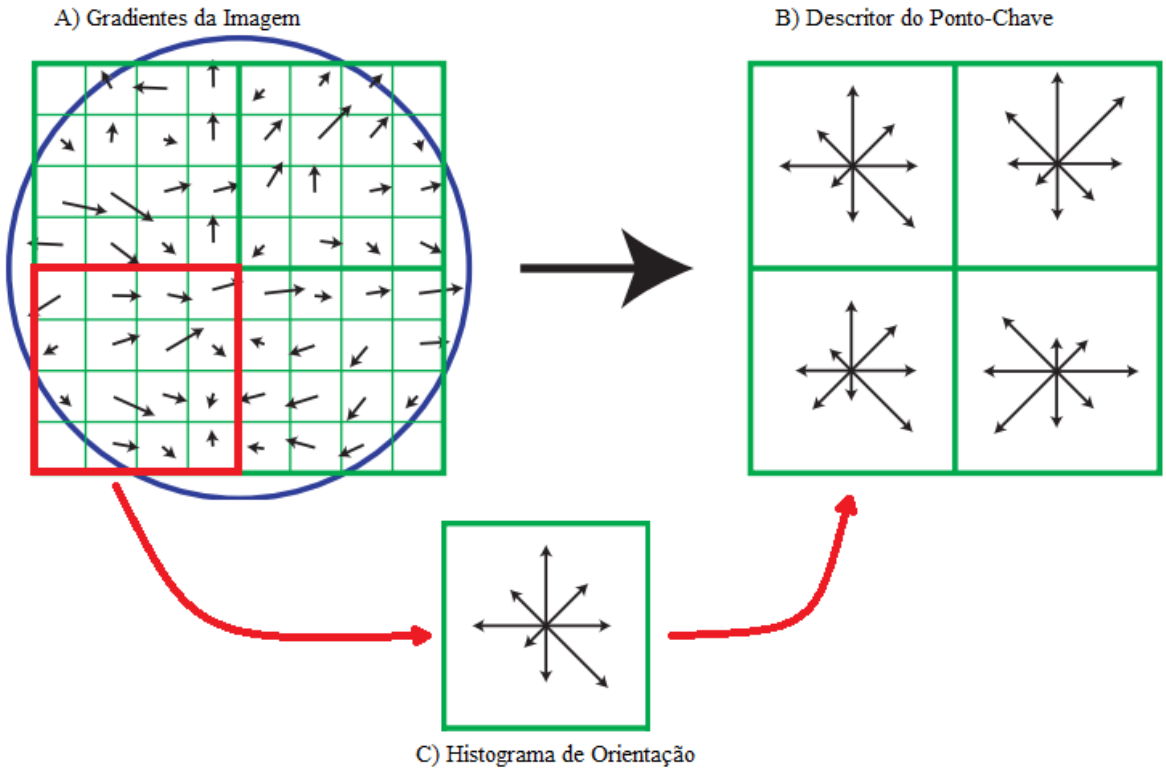

Figura 3.5: Um descritor de um ponto-chave B) é criado a partir do cálculo de uma função Gaussiana para dar peso à magnitude de cada ponto da vizinhança (representados pelas setas) do ponto-chave A). Neste exemplo são acumulados para cada histograma de orientação C) a soma da magnitude dos gradientes próximos aquela direção de uma das regiões 4x4 da imagem A). Esta figura demostra um vetor descritor de 2x2 computado de um conjunto 8x8 de gradientes.

3.2 Encontro de Pontos em Comum: *Matching*

Após a execução de toda parte de processamento de características, finalmente, é chegada a hora de utilizá-las para efetuar a correspondência entre duas imagens procurando por pontos correspondentes entre elas. Uma representação do processo de correspondência entre imagens pode ser visto na Figura 3.6. Tal correspondência entre os descritores da imagem baseia-se no quanto parecidos são os descritores.

A obtenção de uma solução robusta para o problema de comparação entre os vetores descritores pode ser considerada como um elemento chave na identificação de objetos, pois quanto melhor for essa solução mais rapidamente os objetos serão reconhecidos ou descartados.

Na abordagem SIFT, os vetores descritores podem ser comparados, por exemplo, utilizando a distância Euclidiana. Geralmente, pontos candidatos a melhor correspondência são pontos próximos dessa maneira o melhor candidato ao *matching* é o ponto com menor distância Euclidiana.

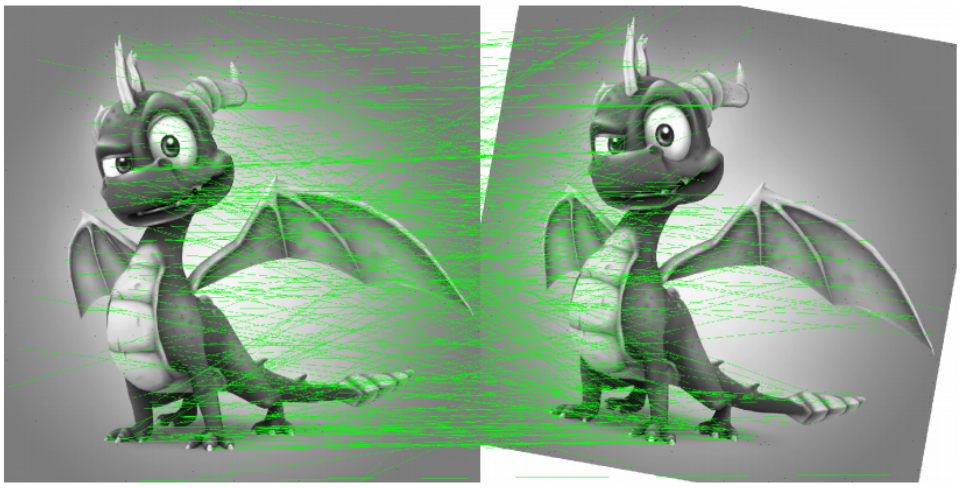

Figura 3.6: Processo de correspondência entre duas imagens através da técnica SIFT.

Em sua abordagem Lowe, [Lowe, 2004],utilizou uma modificação do algoritmo conhecido como Árvore k-d chamado de *Best-Bin-First*(BBF), que pode identificar vizinhos com elevada probabilidade, utilizando uma quantidade inferior de esforço computacional.

Porém por ser um método probabilístico, alguns pontos instáveis são detectados como corretos ao longo do processo, levando a falsas correspondências. para eliminação deste problema, Lowe usou além da comparação da menor distância a comparação da segunda melhor distância, selecionando apenas correspondentes próximos a um limiar. Utilizando um limiar de 0.8, como valor máximo em relação a distância de 2 pontos, é possível eliminar 90% das falsas correlações, porém apenas descartando 5% das correspondências corretas. Portanto, de maneira eficiente as correspondências são refinadas e os falsos pares são descartados.

A partir das explicações deste capítulo, pode-se formar uma ideia mais sólida do funcionamento de cada parte de um SVA, desde a captura das imagens e aplicação dos primeiros filtros de aperfeiçoamento até a execução de métodos sofisticados e com grande poder de reconhecimento como SIFT.

4 TRABALHOS RELACIONADOS

Com o crescimento da busca constante por melhorias e novos métodos de reconhecimento cada vez melhores, foram encontrados vários trabalhos relacionados. Praticamente todos eles foram encontrados a partir da etapa de investigação de técnicas proposta anteriormente. Neste processo, com a leitura pelo menos parcial de algumas publicações, foram encontradas combinações de métodos diferentes da área de visão computacional que na maioria dos casos se apresentavam com bom desempenho para a abordagem que foram desenvolvidos.

Dentre eles, foram selecionados cinco principais divididos em duas classes.Primeiramente referentes a afirmação do uso de SIFT para reconhecimento dos objetos de forma a consolidar o porque da escolha deste método para a abordagem proposta, foram escolhidos [Zahedi and Salehi, 2011], [Ruf et al., 2008], [Alonso-Fernandez et al., 2009],[Rublee et al., 2011]. Trabalhos estes que podem ser considerados recentes se levado em conta os primeiros estudos deste método, os quais datam do ano de 1999 [Lowe, 1999].

Em [Zahedi and Salehi, 2011] foi proposta uma abordagem de SIFT no reconhecimento de placas de automóveis, inspirado pelo sucesso do método na extração de características. A aplicação do método por si só não apresentou resultados 100% satisfatórios, pois foram encontrados problemas com diferentes fundos na imagem bem como a identificação de falsos pontos de interesse. Todavia, os resultados foram refinados com um tratamento da imagem com a aplicação de uma técnica para encontro de contornos parecida com a Limiarização, com o objetivo de eliminar estes objetos ao fundo que causavam falsas correspondências levando a resultados errados.

Neste caso, o diferencial da abordagem proposta em relação a última está em não utilizar nenhuma técnica para auxiliar no processamento das imagens dos objetos de teste, explorando ao máximo toda a robustez oferecida pelo SIFT. Já [Alonso-Fernandez et al., 2009], baseando-se na troca de alguns parâmetros do método SIFT, apresentou resultados melhorados para sua abordagem de reconhecimento da iris do olho.

Este último apresentou contribuições referentes ao grande número possível de aplicabilidade de SIFT, bem como ferramentas para melhor compreensão do impacto das alterações de dos parâmetros do mesmo, parâmetros esses como o tamanho de cada oitava e o fator de escala da função Gaussiana (σ). Em [Alonso-Fernandez et al., 2009], é mais uma vez ressaltado o poder de invariância de SIFT mesmo com a mudança de condições do cenário.

No processo de investigação sobre técnicas de visão computacional e reconhecimento foram encontrados métodos que apresentavam resultados melhores que SIFT, porém em condições limitadas ou com percas em outros aspectos. Em [Ruf et al., 2008], foram utilizados SIFT e SURF para identificação de pontos-chave e posterior correspondências entre imagens de obras de museus, tais métodos foram aplicados sobre diferentes resoluções das imagens e de diferentes perspectivas.

Os resultados obtidos por [Ruf et al., 2008] durante a comparação de SIFT e SURF evidenciaram que SURF possui um tempo de execução melhor e sua complexidade é menor em todas resoluções testadas, devido a isso o número de descritores encontrados por ele é menor, fato este que tem influência direta na performance do reconhecimento, ou seja, SURF apresentou-se inferior a SIFT em qualquer um dos experimentos realizados.

Como consequência do crescimento dos estudos nesta área da visão computacional surgem também novos métodos alternativos, muitos deles com boa eficiência se comparados aos mais conhecido. Neste contexto em [Rublee et al., 2011] é apresentado um descritor binário chamado de *Oriented Fast and Rotated BRIEF* (ORB) invariante a rotação e resistente a ruído, o qual apresenta uma velocidade de reconhecimento que pode alcançar até 2 vezes a velocidade de SIFT.

Mas toda essa velocidade deixa de ser tão importante se comparados a quantidade e o poder dos descritores encontrados por este método a SIFT. ORB possui possui menos características de invariância que SIFT, sendo esse o ponto chave da proposta deste trabalho, que é trabalhar com ambientes não controlados. Para isso quanto mais invariância o método oferecer, maior será a capacidade de reconhecimento nesses ambientes.

Esses dois últimos trabalhos citados foram de suma importância para a afirmação da proposta de uso do método SIFT. Voltando ao ponto de algoritmos específicos para cada abordagem, outros métodos como SURF e ORB podem ser extremamente suficientes, implicando assim que seu uso seja mais indicado, o que não ocorre dentro do escopo da proposta deste trabalho.

Durante o processo de investigação destes métodos ficou claro que os métodos de reconhecimento eram parte importante na maioria das aplicações, porém em complemento a eles sempre haviam algoritmos poderosos para efetuar a correspondência entre os descritores que os mesmos geravam. Portanto, foram necessárias leituras referentes aos principais encontrados que se mostravam mais viáveis para a esta abordagem.

Leituras de trabalhos como [Aly et al., 2009] serviram de base para investigações que ainda precisam ser finalizadas na etapa detre trabalho. Em [Aly et al., 2009] apresenta-se uma análise geral de algoritmos de correspondência que trabalham com grandes volumes de dados, tais quais que serão gerados pelo método SIFT na construção dos descritores.

Os algoritmos apresentados por [Aly et al., 2009] foram *Kd-trees e Kd-forests*, *Locality Sensitive Hashing (LSH)*, além de *Bag-of-Words Search*. Cada um destes apresentou-se como uma alternativa viável para bancos de dados com até um milhão de características. Porém, a performance dos métodos *Kd-trees* e *Bag-of-Words* decai conforme o banco de dados aumenta, o que não acontece com algoritmo de hash *LSH*, o qual permanece constante para todos os bancos testados por eles. *LSH* e *Kd-trees* requisitam a mesma quantidade de memória, fator este que pode tornar o *LSH* preferível para grandes bancos de dados. O número de acertos das correspondências com *LSH* diminui a medida que o banco de dados aumenta, mas mesmo assim sua performance se mantém constante, enquanto a dos outros dois métodos diminui.

Estes algoritmos ainda precisarão ser estudados mais a fundo antes de se efetuar o desenvolvimento da proposta final deste trabalho, assim como vários outros conceitos ainda deverão ser revistos principalmente com o surgimento das dúvidas no decorrer das próximas etapas. O próximo capítulo apresenta de forma resumida a metodologia seguida para realização deste projeto.

5 PROPOSTA

Após leituras sobre os métodos de reconhecimento, sempre levando em conta o objetivo desde trabalho, *Scale Invariant Feature Transform*(SIFT), concebido por David G. Lowe et al. em 1999 apresentou-se com o um método favorável e mais apropriado para o desenvolvimento de um sistema de reconhecimento para ambientes não controlados. Um método conhecido por sua robustez em mudanças de escala, iluminação e distorções de perspectivas. Mas seu grande benefício é o uso de características locais para descrição da imagem.

Durante uma investigação de técnicas foram descobertos vários métodos que eram parecidos em sua essência porém muito particulares para a abordagem que eram desenvolvidos, o que reforçou ainda mais a ideia e a compreensão de o porque de se falar que cara abordagem tem seu próprio estudo do problema bem como formas diferentes de tratá-lo, ainda mais se nestes forem envolvidos descritores locais.

Os algoritmos de reconhecimento que usam aspectos globias para reconhecimento tendem a deixar a desejar em resultados obtidos com essas variações no ambiente de reconhecimento, os mesmos precisam primeiramente de uma etapa de segmentação para serem extraídos os objetos de interesse e descartados todo e qualquer fundo. Processo este que ainda não possui algoritmos "baratos"capazes de executá-los com resultados satisfatórios em ambientes não controlados.

As imagens dos objetos utilizadas neste sistema serão capturadas em um ambiente não controlado qualquer, sendo assim, poderão conter partes extras da cena que não farão parte do objeto alvo ou até mesmo poderão estar em intensidades diferentes de luminosidade, o que pode tornar o reconhecimento um verdadeiro desafio. Mas a grande maioria deste problemas podem ser evitados fazendo uso de descritores locais de características da imagem, os quais são os pilares da proposta do sistema a ser desenvolvido.

Primeiramente, será oferecido uma interface para que o usuário seja capaz de cadastrar objetos a serem reconhecidos pelo sistema na etapa de *correspondência*, bem como sua descrição, caso considerar relevante.

Com as imagens cadastradas será criado uma base de dados que armazenará as informações fornecidas pelo usuário juntamente com os pontos-chave identificados na imagem, os quais são utilizados para identificar os objetos na etapa do reconhecimento. Como exemplo pode-se citar que o usuário queira cadastrar uma xícara, ele irá informar que o objeto sendo

identificado é uma xícara, vamos supor com capacidade de 350ml, e quem é o dono dela além de ouras características que achar relevante.

A ideia do uso de de informações fornecidas pelo usuário é a de contextualizar o objeto reconhecido, para que não apenas seja mostrado que o objeto foi reconhecido, mas que seja apresentado o que é este objeto de fato.

A principal funcionalidade do sistema será o reconhecimento de objetos contidos em fotos capturadas ou carregadas para o reconhecedor, juntamente com contextualização do objeto. Por exemplo, em um almoxarifado estão armazenadas peças de carros de diferentes modelos e marcas, todas devidamente cadastradas no sistema de reconhecimento. Ao se apresentar a imagem de uma peça, o sistema pode identificá-la apresentando a qual modelo de automóvel ela pertence, de qual peça se trata enfim todas informações fornecidas no momento da alimentação da base de dados, sem que haja necessidade do usuário saber todas essas informações por conta própria facilitando a realização da tarefa.

O grande desafio deste sistema é ser eficiente e ao mesmo tempo robusto, efetuando reconhecimento de objetos em um tempo hábil com o menor esforço computacional possível, mas mantendo a credibilidade dos resultados. Para isso, todas as etapas do mesmo serão validadas por meio de testes e métricas frequentemente utilizadas para este propósito de análise.

O resultado final será um algoritmo gratuito de código aberto, que sirva como um identificador versátil de objetos, que possa ser utilizado por qualquer pessoa, porém sem fins lucrativos devido a patente do método SIFT. Assim, facilitando o acesso a esse tipo de ferramenta não muito comum e disponível. Dando sequência, o próximo capítulo apresentará a metodologia utilizada durante o desenvolvimento deste projeto.

6 METODOLOGIA

Com a definição do tema de pesquisa e, a partir do mesmo, iniciou-se o processo de busca por referencias teóricas. Os primeiros resultados foram obtidos realizando uma consulta no *Google Scholar* utilizando a $string$ de busca *Object recognition techniques*, a qual filtrou uma grande quantidade de resultados, porém esta *string* não foi a única usada, algumas variações da mesma foram utilizadas para tentar melhorar os resultados mas essa foi que menos fugiu a ideia deste trabalho e apresentou resultados satisfatórios. A partir disso foram lidos uma certa quantidade dos títulos dos resultados da busca e posteriormente acessados os que pareciam ser de maior relevância.

Para os resultados que foram vistos nesta etapa foi efetuada leitura parcial ou total dos resumos para se obter uma perspectiva melhor do conteúdo abordado pelos mesmos. Com essa leitura grande parte dos resultados foram descartados pois apresentavam grandes diferenças do tema alvo restando em torno de dez publicações. Nesta etapa verificou-se que a maioria dos resultados derivavam da biblioteca digital *IEEE Explore*, então foi realizada uma busca diretamente no site da biblioteca.

Realizada a primeira busca foram retornados um grande numero de resultados, para os mesmos foram utilizados o mesmos critérios já citados anteriormente para selecionar quais poderiam contribuir com o trabalho em questão. Após esta seleção restaram em torno de dez publicações.

Com um número maior de publicações tornou-se necessário efetuar uma classificação em uma escala de 1 a 5, onde 1 continha publicações menos relevantes e 5 mais relevantes. O primeiro ranqueamento foi feito com base na leitura dos resumos, o qual foi aprimorado posteriormente com a leitura total ou parcial das publicações ou com a inserção de outras publicações relevantes encontradas no decorrer do desenvolvimento.

As primeiras leituras tiveram o intuito de ampliar o conhecimento sobre o estado-da-arte, bem como descobrir as etapas e o funcionamento das técnicas de um sistema de reconhecimento de objetos, o que apresentou surpresas devido a grande diversidade de técnicas e abordagens diferentes para cada necessidade.

Nesta etapa um método de reconhecimento e correspondência de objetos chamou a atenção pelas citações e pelo jeito como esses processos eram realizados, tratava-se do método *Scale Invariant Feature Transform (SIFT)*. Iniciou-se outro processo de busca por publicações e refe-

rências sobre o mesmo, fazendo uso de uma série de *strings* de busca tais como, *SIFT Object Recognition*, *SIFT Reconhecimento de objetos*, *Implementing SIFT algorithm* entre outras. Os resultados dessa obtidos foram mais concentrados e específicos, consequentemente mais objetivos e menos numerosos, após a leitura de algumas publicações optou-se por usar esta como principal técnica para o desenvolvimento deste trabalho.

Com a definição da técnica e das leituras mais aprofundadas foram efetuadas novas buscas conforme a necessidade para obter o melhor número possível de informações e referências sobre a mesma e seus conhecimentos derivados, esses pesquisas foram essências para a continuidade e compreensão da proposta do trabalho.

O projeto será dividido em quatro etapas principais: (i) Estudo e construção de um projeto do sistema (ii) Desenvolvimento do sistema (iii) Validação dos resultados $(iiii)$ Testes de desempenho (*Benchmarks*)

Na primeira etapa será feito um levantamento das ferramentas necessárias para a construção do sistema bem como a definição da linguagem e técnicas secundárias a serem utilizadas e a estrutura básica de como o mesmo vai funcionar.

Na segunda etapa será efetuado o desenvolvimento do sistema em si, essa provavelmente será a etapa mais longa, fazendo uso de bibliotecas de código aberto como auxílio no desenvolvimento.

A etapa de validação tem o intuito de observar os resultados obtidos e classificá-los conforme a necessidade do sistema garantindo assim que o mesmo funciona de fato. Será necessário o uso de métricas simples como porcentagem de acertos e tempo de execução para análise dos resultados válidos.

Na última etapa serão feitos testes de desempenho do sistema para por em prova sua efetividade e efetuar análises das taxas de reconhecimento por meio de gráficos.

REFERÊNCIAS

[1] F. Alonso-Fernandez, P. Tome-Gonzalez, V. Ruiz-Albacete, and J. Ortega-Garcia. Iris recognition based on sift features. In *Biometrics, Identity and Security (BIdS), 2009 International Conference on*, pages 1–8, 2009.

[2] M. Aly, P. Welinder, M. Munich, and P. Perona. Scaling object recognition: Benchmark of current state of the art techniques. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 2117–2124, Sept 2009. doi: 10.1109/ICCVW.2009.5457542.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008. ISSN 1077-3142. doi: 10.1016/j.cviu.2007.09.014. URL http://dx.doi.org/10.1016/j.cviu.2007.09.014.

[4] M. Brown and D. Lowe. Invariant features from interest point groups. In *In British Machine Vision Conference*, pages 656–665, 2002.

[5] C. Harris and M. Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[6] D. G. Lowe. Object recognition from local scale-invariant features. pages 1150–, 1999. URL http://dl.acm.org/citation.cfm?id=850924.851523.

[7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

[8] J. G. R. Maia. Detecção e reconhecimento de objetos utilizando descritores locais. Maio 2010.

[9] O. Marques Filho and H. Vieira Neto. *Processamento Digital de Imagens*. Editora Brasport, Rio de Janeiro, Brazil, 1999.

[10] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, ECCV '02, pages 128–142,

London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-43745-2. URL http://dl.acm.org/citation.cfm?id=645315.649184.

[11] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, Oct. 2005. ISSN 0162-8828. doi: 10.1109/TPAMI.2005.188. URL http://dx.doi.org/10.1109/TPAMI.2005.188.

[12] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2564–2571, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-1-4577-1101-5. doi: 10.1109/ICCV.2011.6126544. URL http://dx.doi.org/10.1109/ICCV.2011.6126544.

[13] B. Ruf, E. Kokiopoulou, and M. Detyniecki. Mobile museum guide based on fast sift recognition. In M. Detyniecki, U. Leiner, and A. Nürnberger, editors, *Adaptive Multimedia Retrieval*, volume 5811 of *Lecture Notes in Computer Science*, pages 170–183. Springer, 2008. ISBN 978-3-642-14757-9. URL http://dblp.uni-trier.de/db/conf/amr/amr2008.html#RufKD08.

[14] A. S. Selhorst. Utilização de visão computacional e detecção de características para auxiliar na navegação de pessoas com deficiência visual em ambientes internos. Technical report, Curso de Ciência da Computação. Universidade Federal da Fronteira Sul, Chapecó, 2014.

[15] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, 2007. ISBN 049508252X.

[16] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177–280, July 2008. ISSN 1572-2740. doi: 10.1561/0600000017. URL http://dx.doi.org/10.1561/0600000017.

[17] M. Zahedi and S. M. Salehi. License plate recognition system based on SIFT features. In A. Karahoca and S. Kanbul, editors, *First World Conference on Information Technology, WCIT 2010, Istanbul, Turkey, October 6-10, 2010*, volume 3 of *Procedia Computer Science*, pages 998–1002. Elsevier, 2011. doi: 10.1016/j.procs.2010.12.164. URL http://dx.doi.org/10.1016/j.procs.2010.12.164.