



**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. Emilio Wuerges - UFFS

Ma. Adriano Sanik Padilha - UFFS

RESUMO

Palavras-chave: Visão computacional. Reconhecimento. .

LISTA DE FIGURAS

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].	13
Figura 2.2 – Principais etapas de um Sistema de Visão Artificial (SVA) [Marques Filho and Vieira Neto, 1999].	14
Figura 2.3 – Aplicação do Filtro da Média. À esquerda a imagem original, à direita a imagem resultante da aplicação do filtro [Selhorst, 2014].	16
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.	16
Figura 3.1 – Os 4 estágios do algoritmo SIFT. A) Detecção de extremos B) Localização de pontos-chave C) Definição de orientação e D) Descrição dos pontos-chave	22
Figura 3.2 – Representação do procedimento das Diferenças Gaussianas DoG para diversas oitavas de uma imagem. [Lowe, 2004]	23

SUMÁRIO

1 INTRODUÇÃO	7
1.1 Motivação	7
1.2 Tema	7
1.3 Objetivos	8
1.3.1 Objetivo Geral	8
1.3.2 Objetivos Específicos	8
1.4 Justificativa	8
1.5 Estrutura do Trabalho	9
2 VISÃO COMPUTACIONAL	11
2.1 Sistema de Visão Artificial	12
2.1.1 Imagem Digital	12
2.1.2 Estrutura de um Sistema de Visão Computacional	13
2.1.3 Definição do problema	14
2.1.4 Aquisição da Imagem	15
2.1.5 Pré-processamento	15
2.1.6 Segmentação	15
2.1.7 Extração de Características	16
2.1.8 Reconhecimento e Interpretações	17
2.1.9 Base de Conhecimento	17
2.2 Descritores locais em SVAs	18
2.2.1 Métodos <i>Harris-Laplace</i> e <i>Hessiano-Laplace</i>	18
2.2.1.1 Harris-Laplace	18
2.2.1.2 Hessiano-Laplace	19
2.2.2 Método <i>Speeded-Up Robust Features</i> (SURF)	19
2.2.3 Introdução ao método SIFT (<i>Scale-Invariant Features Transform</i>)	20
3 SCALE INVARIANT FEATURES TRANSFORM (SIFT)	21
3.1 Etapas do Algoritmo SIFT	21
3.1.1 Detecção de extremos	21
4 TRABALHOS RELACIONADOS	24
5 PROPOSTA	25
6 METODOLOGIA	26
REFERÊNCIAS	28

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 analisada 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 interação 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 objetos previamente cadastrados, para auxiliar em tarefas de

reconhecimento em atividades que envolvem um grande número de objetos com características parecidas. 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. Porém hoje em dia cresce cada vez mais a interação entre o homem e os computadores, *smartphones* entre outros dispositivos no dia-a-dia.

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 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 a especificação do mesmo.

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 conseguir que um computador possa "enxergar" e interpretar, técnicas essas executáveis por meio de hardware e software que tem inúmeras aplicações.

Neste capítulo serão abordados os conhecimentos necessários para a compreensão e validação do projeto. Apresenta-se aqui a intuição da visão computacional, aplicabilidades, principais conceitos e ao final deste capítulo algumas das técnicas mais comuns utilizadas.

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 as tecnologias disponíveis costumam trabalhar com imagens bi-dimensionais (2D), esta projeção para um número menor de dimensões acarreta em uma grande perda de informações. Cenas dinâmicas tais como estamos acostumados a ver, com objetos em movimentos, 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 são necessárias 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 comumente 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-a de 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 aprofundados e em objetivos finais. 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 mais o parecido possível com o original, aplica-se aqui um laço de conhecimento interativo para compreensão da imagem a qual eventualmente converge para o objetivo final.

Conceitos aplicados 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

2.1.1 Imagem Digital

Atualmente, efetuar a captura de uma imagem é um processo cada vez mais comum, melhores e menores sensores com funcionalidades robustas vem sendo desenvolvidas para uso em celulares e *smartphones* possibilitando 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 imagem para gerar uma nova 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.

Imagens monocromáticas são as mais usadas no âmbito da visão computacional devido



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].

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 pixels. Mas, pode-se conhecer uma medida de qualidade da amostragem, conhecendo-se a razão entre o número de pixels 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 Computacional

Um sistema de visão artificial (SVA), ou sistema de visão computacional, é constituído de diversas etapas, cada qual com suas peculiaridades e papel definido 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 formas 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.

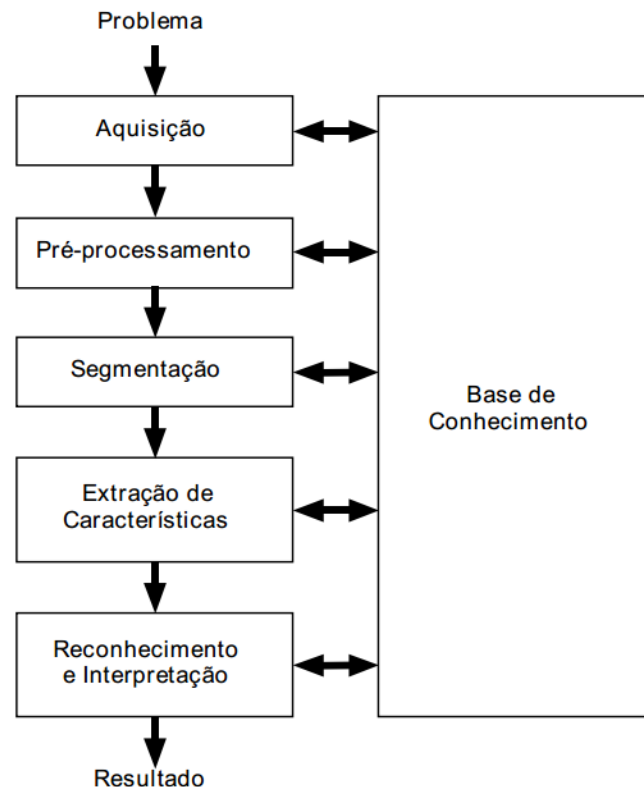


Figura 2.2: Principais etapas de um Sistema de Visão Artificial (SVA) [Marques Filho and Vieira Neto, 1999].

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.

Para fins de demonstração, pode se usar como exemplo a leitura da placa de um lote de carros 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 pode ser visto na Figura 2.3, o filtro em questão é chamado de Filtro da Média, cujo objetivo é diminuir o nível de ruído em uma imagem, percebe-se na imagem a esquerda uma suavização entre as diferenças de cada pixel.

2.1.6 Segmentação

A etapa de segmentação tem como principal tarefa dividir a imagem em múltiplas regiões de interesse formadas por conjuntos de pixels 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



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

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.



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.

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 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 Interpretações

Baseando-se nas características de um objeto traduzidas por seus descritores, na última 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, a interpretação seria 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 tarefas 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, 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 técnicas mais estudadas 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 *matching* e reconhecimento, uma vez que possuem robustez mesmo com visibilidade e desorganização parcial. A dificuldade é a 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 de *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 autoadaptação 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 também é usada para detecção de características e para descrição de estruturas locais em 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 detec-

tor 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}

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

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 um canto [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 característica 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 os 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 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 [?] que efetua a busca por pontos-chaves por meio de cálculos de diferença das Gaussianas, o mesmo é conhecido pela suas características de invariância de escala, rotação, iluminação e ponto de visão.

Devido principalmente a essas características, este método foi selecionado para o desenvolvimento deste projeto, tais características 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 sendo este o método 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 partes 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 formam duas principais partes 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].

No decorrer destas duas etapas principais pode-se subdividi-las em dois estágios cada, de maneira a entender de forma clara o funcionamento do algoritmo. Cada um destes estágios será explicado nas sub sessões a seguir e podem ser observados na Figura 3.1.

3.1.1 Detecção de extremos

O primeiro estágio do método SIFT, Figura 3.1 A), 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 denominada espaço de escala, que neste caso é a função Gaussiana.

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

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (3.1)$$

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

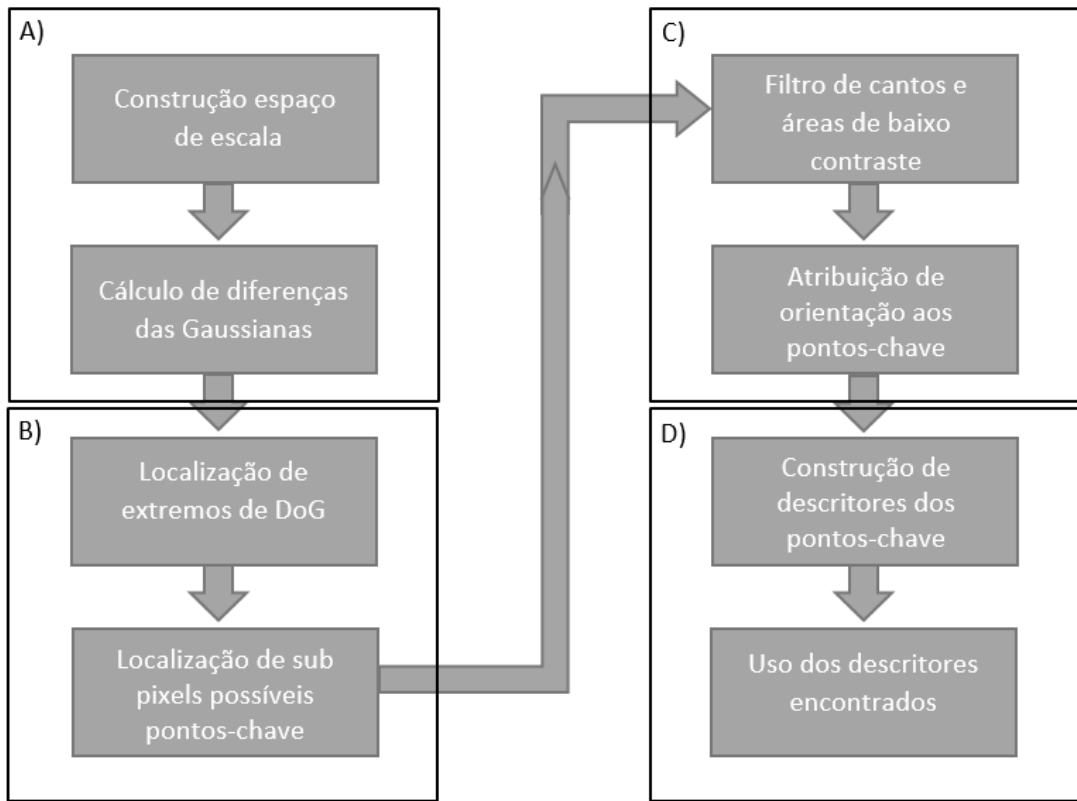


Figura 3.1: Os 4 estágios do algoritmo SIFT. A) Detecção de extremos B) Localização de pontos-chave C) Definição de orientação e D) Descrição dos pontos-chave

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

percebe-se que este filtro é variável à escala através do parâmetro σ , onde $\sqrt{2}$ é considerado um bom valor para o mesmo [Lowe, 2004].

Para detectar eficientemente pontos-chave estáveis no espaço de escala utiliza-se uma função DoG("Difference of Gaussian") formada pela diferença de imagens filtradas em escalas próximas, separadas por uma constante k . A definição da função DoG é definida por

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma) \quad (3.3)$$

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

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

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (3.5)$$

Esta função DoG suaviza as imagens e pode ser calculada pela simples subtração de imagens borradas por um filtro Gaussiano em escalas σ e $k\sigma$. Tal função tem o objetivo de conseguir

amostras de imagens onde detalhes indesejados e ruídos sejam eliminados e características fortes sejam realçadas [Lowe, 2004].

Durante os cálculos da função DoG utiliza-se uma representação que combina a computação de uma pilha de imagens contendo níveis de detalhe do espaço de escala linear com pirâmides de imagens. Esta pilha de imagens em vários níveis de detalhe é geralmente denominada *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 pixels. Este processo é ilustrado pela Figura 3.2.

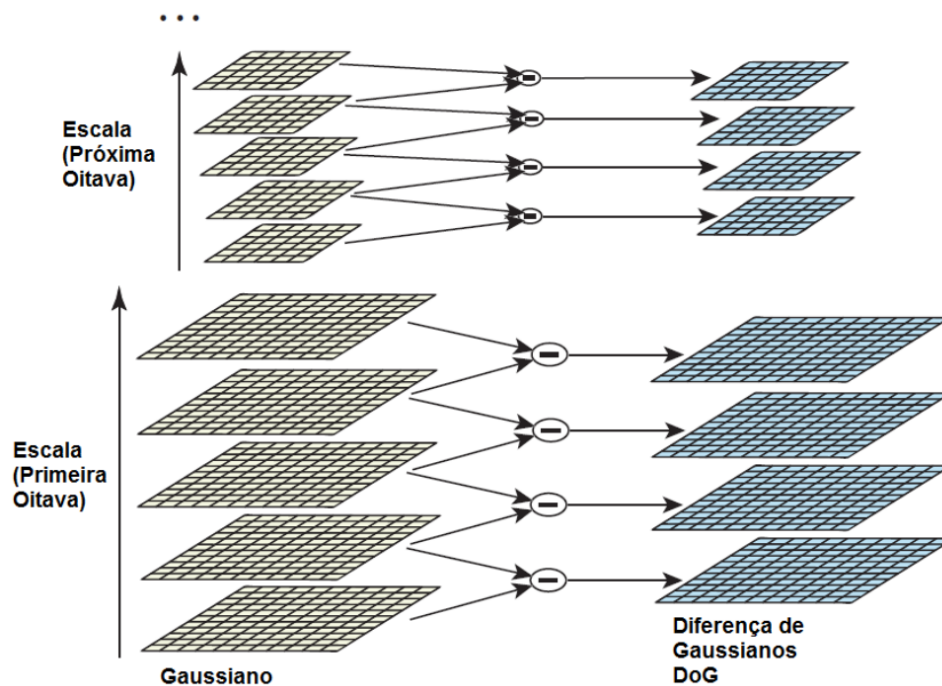


Figura 3.2: Representação do procedimento das Diferenças Gaussianas DoG para diversas oitavas de uma imagem. [Lowe, 2004]

4 TRABALHOS RELACIONADOS

[Chaturvedi and Prof, 2013]

[Ruf et al., 2008]

[Zahedi and Salehi, 2011]

[Alonso-Fernandez et al., 2009]

[Ruble et al., 2011]

5 PROPOSTA

6 METODOLOGIA

Com a definição do tema de pesquisa e, a partir do mesmo, iniciou-se o processo de busca por referências 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 número de resultados, para os mesmos foram utilizados os 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 *matching* 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, essas 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] 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>.
- [3] A. Chaturvedi and N. S. Prof. Automatic license plate recognition system using surf features and rbf neural network, 2013.
- [4] C. Harris and M. Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.
- [5] D. G. Lowe. Object recognition from local scale-invariant features. 1999.
- [6] D. G. Lowe. Distinctive image features from scale-invariant keypoints. Janeiro 2004.
- [7] J. G. R. Maia. Detecção e reconhecimento de objetos utilizando descritores locais. Maio 2010.
- [8] O. Marques Filho and H. Vieira Neto. *Processamento Digital de Imagens*. Editora Brasport, Rio de Janeiro, Brazil, 1999.
- [9] 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>.
- [10] 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>.
- [11] 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>.
- [12] 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>.
- [13] 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.
- [14] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, 2007. ISBN 049508252X.
- [15] 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: 10.1561/06000000017. URL <http://dx.doi.org/10.1561/06000000017>.
- [16] 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>.