
Sistema para localização robótica de veículos
autônomos baseado em visão computacional por
pontos de referência

Leandro Nogueira Couto

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

**Sistema para localização robótica de veículos autônomos
baseado em visão computacional por pontos de referência**

Leandro Nogueira Couto

Orientador: Prof. Dr. Fernando Santos Osório

Dissertação apresentada ao Instituto de Ciências
Matemáticas e de Computação - ICMC-USP, como parte
dos requisitos para obtenção do título de Mestre em
Ciências - Ciências de Computação e Matemática
Computacional. *VERSÃO REVISADA*

USP – São Carlos

Julho de 2012

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

C871s Couto, Leandro Nogueira
 Sistema para localização robótica de veículos
 autônomos baseado em visão computacional por pontos
 de referência / Leandro Nogueira Couto; orientador
 Fernando Santos Osório. -- São Carlos, 2012.
 82 p.

 Dissertação (Mestrado - Programa de Pós-Graduação em
 Ciências de Computação e Matemática Computacional) --
 Instituto de Ciências Matemáticas e de Computação,
 Universidade de São Paulo, 2012.

 1. Robótica Móvel. 2. Visão Computacional. I.
 Osório, Fernando Santos, orient. II. Título.

Resumo

A integração de sistemas de Visão Computacional com a Robótica Móvel é um campo de grande interesse na pesquisa. Este trabalho demonstra um método de localização global para Robôs Móveis Autônomos baseado na criação de uma memória visual, através da detecção e descrição de pontos de referência de imagens capturadas, com o método SURF, associados a dados de odometria, em um ambiente interno. O procedimento proposto, associado com conhecimento específico sobre o ambiente, permite que a localização seja obtida posteriormente pelo pareamento entre quadros memorizados e a cena atual observada pelo robô. Experimentos são conduzidos para mostrar a efetividade do método na localização robótica. Aprimoramentos para situações difíceis como travessia de portas são apresentados. Os resultados são analisados, e alternativas para navegação e possíveis futuros refinamentos discutidos.

Abstract

Integration of Computer Vision and Mobile Robotics systems is a field of great interest in research. This work demonstrates a method of global localization for Autonomous Mobile Robots based on the creation of a visual memory map, through detection and description of reference points from captured images, using the SURF method, associated to odometer data in a specific environment. The proposed procedure, coupled with specific knowledge of the environment, allows for localization to be achieved at a later stage through pairing of these memorized features with the scene being observed in real time. Experiments are conducted to show the effectiveness of the method for the localization of mobile robots in indoor environments. Improvements aimed at difficult situations such as traversing doors are presented. Results are analyzed and navigation alternatives and possible future refinements are discussed.

Sumário

RESUMO	I
ABSTRACT	III
SUMÁRIO.....	V
LISTA DE FIGURAS	VI
1 INTRODUÇÃO.....	1-1
1.1 MOTIVAÇÃO.....	1-5
1.2 OBJETIVO	1-7
1.3 ESTRUTURA DA MONOGRAFIA	1-10
2 ROBÓTICA MÓVEL.....	2-11
2.1 SENsores e ATUADORES	2-12
2.2 MAPAS e LOCALIZAÇÃO	2-15
2.2.1 <i>Métodos Probabilísticos Para Localização.....</i>	2-20
2.3 A NAVEGAÇÃO e CONTROLE ROBÓTICO	2-22
2.4 A ALTERNATIVA DOS SENsores VISUAIS	2-26
3 VISÃO COMPUTACIONAL E PROCESSAMENTO DE IMAGENS	3-28
3.1 REPRESENTAÇÃO DIGITAL DE IMAGENS.....	3-28
3.2 VISÃO COMPUTACIONAL APLICADA À ROBÓTICA.....	3-31
3.3 MÉTODOS DETECTORES-DESCRITORES BASEADOS EM ESPAÇO DE ESCALA	3-34
3.3.1 <i>SIFT.....</i>	3-34
3.3.2 <i>SURF</i>	3-39
3.3.3 <i>Outros métodos similares</i>	3-40
4 VISÃO COMPUTACIONAL APLICADA À ROBÓTICA: FERRAMENTAS E MÉTODOS	4-42
4.1 PLATAFORMAS ROBÓTICAS e PERIFÉRICOS	4-42
4.2 OPENCV	4-44
4.3 A FERRAMENTA PLAYER/STAGE.....	4-45
4.4 DETALHAMENTO DA PROPOSTA E MÉTODOS UTILIZADOS	4-47
4.4.1 <i>A Construção do Mapa.....</i>	4-49
4.4.2 <i>A Localização Autônoma.....</i>	4-50
4.4.3 <i>Escolha e Aplicação do Método SURF</i>	4-52
4.4.4 <i>Travessia de Portas</i>	4-56
5 RESULTADOS E CONCLUSÕES	5-58
5.1 TESTES e RESULTADOS DA LOCALIZAÇÃO GLOBAL.....	5-58
5.2 TESTES e RESULTADOS DA TRAVESSIA DE PORTAS.....	5-64
6 DISCUSSÃO.....	6-67
6.1 CONSIDERAÇÕES FINAIS e TRABALHOS FUTUROS.....	6-70
REFERÊNCIAS BIBLIOGRÁFICAS.....	6-72

Lista de Figuras

<i>Figura 1.1: Exemplo de um braço robótico Unimate (Robot Hall of Fame).</i>	1-2
<i>Figura 1.2: Robô humanóide Asimo, desenvolvido pela Honda.</i>	1-3
<i>Figura 1.3: Exemplos de pareamento entre imagens. Perspectivas mais próximas geram mais pares de pontos, e os pares são mais alinhados horizontalmente.</i>	1-9
<i>Figura 2.1: O ciclo de percepção e ação do robô.</i>	2-11
<i>Figura 2.2: Exemplo de mapa métrico (S. Thrun).</i>	2-19
<i>Figura 2.3: Um mapa métrico pode ser decomposto em nós ou células, se tornando um mapa topológico, representado na figura pelo grafo (Siegwart e Nourbakhsh).</i>	2-19
<i>Figura 2.4: O funcionamento da localização Monte Carlo. No quadro A, a nuvem de partículas representa a incerteza quanto à posição do robô. No quadro B, o robô andou um metro para a frente. Como não sabemos sua orientação inicial, as partículas se espalham como no círculo. No quadro C, o robô observa um ponto de referência próximo no canto superior direito, e a likelihood das partículas ali aumenta. O quadro D mostra as novas partículas após o resampling. A partir daí o ciclo reinicia.</i>	2-22
<i>Figura 3.1: Exemplos de métodos de extração de características visuais. Do topo, da esquerda para a direita: imagem original; imagem com Canny; imagem com Sobel; Laplaciano do Gaussiano.</i>	3-33
<i>Figura 3.2: O primeiro passo do algoritmo SIFT. Primeiro calcula-se o Gaussiano das imagens por convolução, e através da subtração dos resultados adjacentes obtém-se as diferenças-do-Gaussiano, onde serão buscadas as características invariantes. A escala da imagem é reduzida e o processo é realizado novamente (Lowe, 2004).</i>	3-37
<i>Figura 3.3: Exemplo de keypoints obtidos com o algoritmo SIFT. Em (a) temos a imagem original, em (b) o primeiro conjunto de keypoints obtidos, em (c) o conjunto reduzido após aplicação de um threshold de contraste mínimo e (d) o conjunto final após um threshold de razão máxima de principais curvaturas (Lowe, 2004).</i>	3-38
<i>Figura 3.4: O método SURF é resistente a alterações na imagem. Neste caso, a presença de uma pessoa na cena não prejudica pares de pontos em outras regiões da imagem.</i>	3-40
<i>Figura 4.1: Robôs Móveis do Laboratório de Robótica Móvel na USP (LRM – ICMC – USP). Da esquerda para a direita, o Pioneer 3-AT, o Pioneer 3-DX e os dois robôs Erratic.</i>	4-43
<i>Figura 4.2: Robô Pioneer 3-AT equipado para uso, com a câmera e notebook.</i>	4-44
<i>Figura 4.3: Uma captura de tela do ambiente de simulação Stage.</i>	4-46
<i>Figura 4.4: A VSRR, composta de um percurso de gravação, posteriormente usado na localização do robô (adaptado de Matsumoto et al., 1996).</i>	4-48
<i>Figura 4.5: Erro de comparação do SURF em imagens com poucas características. O critério de escolha de imagens por proximidade espacial desclassifica estes resultados.</i>	4-55
<i>Figura 4.6: O robô observa a porta com a câmera lateral, buscando parear a imagem observada com a imagem memorizada, que representa o ponto ótimo para manobra.</i>	4-57
<i>Figura 5.1: Diversos dos tipos de quadros encontrados no ambiente interno percorrido.</i>	5-59
<i>Figura 5.2: Estes dois pares mostram diferentes resultados (imagens da direita) obtidos pelos critérios diferentes (primeiro critério em cima, segundo critério embaixo), para a mesma imagem (a imagem da esquerda é a mesma nos dois casos).</i>	5-60
<i>Figura 5.3: Pares de resultados obtidos pelos dois critérios em um ambiente diferente.</i>	5-61
<i>Figura 5.4: Número de pares (matches) obtidos pelo método SURF em relação a diferentes imagens, comparado a uma mesma base de imagens.</i>	5-62
<i>Figura 5.5: O Mosaico mostra a diferença que variações sutis de perspectiva causam no número pares de características visuais em diversas situações.</i>	5-63
<i>Figura 5.6: Exemplo de resultado da travessia de porta, baseado na distância entre características no eixo X.</i>	5-66
<i>Figura 6.1: Fluxo óptico. À esquerda, um exemplo do trabalho de Sokolov et al., (2010). À direita, o fluxo estimado através do OpenSURF.</i>	6-68

1 Introdução

Com o avanço acelerado da tecnologia nas últimas décadas a robótica tem atraído crescente atenção de pesquisadores, empresas, investidores e do público consumidor em geral. De fato, é um campo da tecnologia que oferece diversas possibilidades de aplicação, e conforme desenvolvimentos tecnológicos tornam possíveis avanços na robótica, naturalmente as atenções têm se voltado para a área. Através da pesquisa científica e da engenharia, o ser humano sempre buscou técnicas e ferramentas que trouxessem eficácia, eficiência e segurança para a realização de suas tarefas. Portanto, a perspectiva de robôs ou máquinas que possam trabalhar autonomamente, com pouca ou nenhuma intervenção humana, é uma ideia atraente e atual, e devido ao grande interesse e investimento em projetos e soluções na área, a robótica já está profundamente inserida em muitos setores da sociedade, em aplicações comerciais e industriais. Ramos da ciência e tecnologia que se relacionam com esta área incluem a mecânica, engenharia elétrica e eletrônica e a computação.

Apesar de sua popularização acelerada, a robótica é uma ciência relativamente recente. É difícil apontar uma data para o surgimento da robótica, já que a própria definição de o que constitui um robô possui diferentes versões. Apesar disso, máquinas que podem ser consideradas totalmente autônomas, com algum tipo de propósito inteligente, só surgiram em meados do século XX, sendo que o primeiro robô industrial, o Unimate, foi introduzido em 1961, na forma de um braço robótico de quase duas toneladas que obedecia a seqüências de comandos gravados para tarefas repetitivas¹.

Robôs como o Unimate (Figura 1.1), cujas tarefas pouco exigiam em termos de ação inteligente, não requeriam características muito elaboradas de *hardware*. Desde então, houve grande evolução, e hoje existem novas aplicações para robôs que necessitam de alto poder de processamento, *software* adequado e *hardware* robusto, com sensores elaborados e atuadores precisos, frequentemente embarcados em robôs de tamanho muito reduzido, para atender a aplicações cada vez mais desafiadoras.

¹ <http://www.prsrobots.com/unimate.html>, acesso em 01/2012



Figura 1.1: Exemplo de um braço robótico Unimate (Robot Hall of Fame).

Apesar de ser classificado como um robô, o Unimate era apenas um braço mecânico que repetia comandos pré-estabelecidos. O robô era pouco mais que uma ferramenta sem inteligência e capacidade de tomar decisões. Aos poucos, porém, juntamente com a evolução da computação, evoluiu o conceito de robô inteligente, que é o robô que incorpora elementos que o capacitam a perceber e interagir com o ambiente. Robôs capazes de locomover-se e explorar o ambiente surgiram, com habilidades que procuram emular o que pode ser alcançado pela inteligência humana.

Robôs bípedes, por exemplo, como o Asimo da Honda (Figura 1.2), que é capaz de reconhecer faces, comandos de voz e até correr, requerem atuadores mecânicos precisos para manterem-se em pé, além de sensores de equilíbrio como os acelerômetros ou giroscópios, integrados por um sistema de controle que deve operar praticamente em tempo real para evitar uma queda. Robôs como o Spirit e Opportunity, enviados a Marte, precisam de autonomia suficiente para passar um longo tempo sem manutenção ou troca de baterias, e exigem uma interface de comunicação potente que funcione a grandes distâncias. Através desses exemplos é possível notar a importância da autonomia na robótica móvel, já que estes robôs devem ser independentes o suficiente para tomar decisões de forma rápida e sem intervenção humana, em contraste com muitos robôs industriais e de linhas de montagem. Nestes casos mais recentes, já se pode falar em comportamento robótico, e todos estes avanços são indissociáveis dos grandes avanços do potencial dos computadores e das tecnologias que estes robôs utilizam.

Campos como a Inteligência Artificial exploram maneiras de possibilitar que computadores e robôs repliquem ou emulem o comportamento humano ou animal, com capacidade de raciocínio, abstração, associação de conceitos, entre outras habilidades.

Enquanto inteligência é um conceito difícil de definir, a ciência tem proporcionado avanços significativos neste sentido.



Figura 1.2: Robô humanóide Asimo, desenvolvido pela Honda².

A robótica móvel é uma das sub-áreas da robótica, e relaciona-se a robôs que não são fixos, mas podem se deslocar em relação ao ambiente, de modo tele-operado, semi-autônomo ou totalmente autônomo. Esta é uma aplicação de grande abrangência, e referente a robôs que podem navegar tanto em ambientes terrestres como aquáticos e aéreos. A robótica móvel oferece grandes perspectivas para aplicações no futuro, e robôs móveis já têm sido aplicados na indústria, com pesquisas e projetos também em atividades agrícolas (Ollis e Stentz, 1997), militares³ ou mesmo robôs domésticos⁴.

² <http://asimo.honda.com>, acesso em 01/2012.

³ <http://www.washingtonpost.com/wp-dyn/content/article/2011/01/01/AR2011010102690.html>, acesso em 01/2012.

⁴ <http://www.irobot.com/>, acesso em 01/2012.

Recentemente, temos visto avanços no desenvolvimento de veículos capazes de navegar autonomamente. Um marco digno de nota é o DARPA Grand Challenge de 2005⁵, evento organizado pelo DARPA (agência de pesquisas do Departamento de Defesa dos Estados Unidos); no evento, carros autônomos competiram navegando um percurso de 11.78 km no deserto. Cinco veículos completaram a prova, e o evento obteve resultados gerais muito superiores aos da edição do evento no ano anterior. Em 2007 ocorreu o DARPA Urban Challenge, um evento semelhante, no qual o percurso consistia de um ambiente urbano simulado. Investimentos concentrados em viabilizar veículos autônomos em aplicações práticas e comerciais estão sendo realizados por empresas como Google⁶ ou BMW⁷. Aplicações de robótica móvel críticas como estas demonstram os desafios que a robótica móvel enfrenta; os robôs precisam de respostas rápidas e precisas, em ambientes dinâmicos e altamente complexos, em situações sensíveis a contexto e muitas vezes imprevisíveis. Sistemas assim exigem robôs seguros, resistentes a falhas e capazes de realizar interpretações corretas do ambiente.

No Brasil, a relevância da Robótica Móvel pode ser percebida quando se considera iniciativas como a do INCT-SEC, o Instituto Nacional de Ciência e Tecnologia em Sistemas Embarcados Críticos⁸, em cujo contexto de pesquisa se insere este trabalho. Um dos objetivos primordiais do instituto é justamente o desenvolvimento de veículos autônomos para uso em áreas estratégicas. Uma das frentes de pesquisa do INCT-SEC consiste do desenvolvimento de Veículos Aéreos Não Tripulados (VANTs). Outro grupo de trabalho visa o desenvolvimento de um veículo terrestre autônomo, capaz de auxiliar e assegurar o motorista contra falhas humanas ou situações de risco, ou direção assistida (Tönnis *et al.*, 2008), ou realizar a condução do carro de forma autônoma. Finalmente, um dos grupos de trabalho aborda o desenvolvimento de um robô tático para monitoramento e segurança de ambientes internos (como prédios, instalações civis ou militares), visando ainda compor esquadrões para execução dessas tarefas em conjunto, de forma colaborativa e ordenada com outros robôs. Individualmente ou em esquadrões, esses robôs podem realizar o monitoramento do ambiente,

⁵ <http://www.scientificamerican.com/article.cfm?id=innovations-from-a-robot-2006-01&sc=I100322>, acesso em 01/2012.

⁶ <http://www.nytimes.com/2010/10/10/science/10google.html?ref=science>, acesso em 01/2012.

⁷ <http://www.autoblog.com/2012/01/26/bmw-takes-its-autonomous-5-series-onto-the-autobahn/>, acesso em 01/2012.

⁸ <http://www.inct-sec.org/>, acesso em 01/2012.

detecção e reação a incidentes como invasões ou situações de perigo como focos de incêndios (Pessin, 2008). A navegação com robôs de menor porte também pode ser aplicada em tarefas de navegação em regiões desconhecidas, de difícil acesso ou onde haja perigo ao ser humano.

O projeto de pesquisa desta dissertação de mestrado se enquadra principalmente no último grupo, sendo uma aplicação voltada para robótica móvel em ambientes internos. Os resultados obtidos, porém, podem ser facilmente aplicados nas outras duas frentes de trabalho (VANTs e Veículos Autônomos Terrestres). Portanto este trabalho está contextualizado junto às atividades do INCT-SEC, que têm sido desenvolvidas junto ao LRM (Laboratório de Robótica Móvel do ICMC-USP), onde as aplicações citadas anteriormente servem de motivação e demonstram a importância do presente trabalho.

1.1 Motivação

A visão é um dos principais sentidos do ser humano. A riqueza das informações oferecidas pela visão é imensa. O ser humano não possui um órgão de sentido que se possa considerar equivalente a um sensor *laser* ou sonar, que estimam distâncias pela reflexão de ondas no ambiente (como possuem os morcegos, ou baleias), no entanto pode realizar feitos parecidos e ainda outros com o sentido da visão. O ser humano é capaz de perceber distâncias, reconhecer elementos de uma cena, perceber movimento, cores e texturas e solucionar problemas extremamente complexos e variados de localização e navegação através da visão, utilizando informações visuais obtidas por seus olhos (visão estereoscópica). Na realidade, mesmo usando apenas um dos olhos e associações semânticas e contextuais, o cérebro humano é capaz de realizar reconhecimentos praticamente instantâneos e avaliações corretas de profundidade a partir da visão⁹ (Oliva e Torralba, 2007).

Entre as habilidades cognitivas que o sistema visual humano permite, é possível citar a capacidade de lembrança de um trajeto previamente percorrido ou o reconhecimento de um rosto familiar através do uso da memória visual. Esta capacidade cognitiva é extremamente robusta a deformidades, e efeitos semelhantes podem ser identificados em muito outros animais. Até mesmo abelhas são capazes de realizar essa tarefa com proficiência (Vladusich

⁹ <http://webvision.med.utah.edu/book/part-viii-gabac-receptors/perception-of-depth/>, acesso em 01/2012.

et al., 2005), memorizando sinais como árvores, acidentes geográficos e orientando-se pela posição do Sol (Von Frisch e Seeley, 1993)

A computação da informação visual e de seu conteúdo muitas vezes exige grande poder de processamento. A evolução tecnológica dos computadores, porém, tem permitido o avanço das pesquisas em visão computacional, que é o uso de métodos computacionais envolvendo imagens (Forsyth e Ponce, 2003; Ballard e Brown, 1982). O grande potencial da informação visual faz com que a visão computacional e sua aplicação na robótica móvel sejam campos convidativos a serem exploradas de forma mais detalhada. A robótica móvel busca com frequência inspiração nos seres vivos, portanto é natural explorar uma analogia entre a visão em robôs móveis e o sentido da visão animal. Diversas abordagens existem para simular a capacidade humana de guardar, reconhecer e relembrar informação visual. Entre elas, existem os métodos para extração de pontos de referência de imagens, que surgiram com detectores de características distintas como bordas e cantos, e evoluiu para métodos bastante robustos que escolhem pontos de referência estáveis, ou seja, que possam ser identificados mesmo com alterações na cena em questão. Enquanto o funcionamento da cognição humana ainda não foi completamente compreendido, existem teorias (Ross e Oliva, 2010) que defendem que a memória visual humana e reconhecimento de cenas envolvem a seleção de pontos de referência e atributos globais robustos da cena de forma semelhante a estes métodos computacionais.

Não menos relevante é o fato de que sensores como câmeras de vídeo são muito baratos em relação a outros sensores também usados em pesquisas com robôs móveis como *lasers* do tipo LIDAR (*Light Detection And Ranging*). Existem, é claro, tipos diversos de câmeras com particularidades, qualidade e custo diferentes, mas é fato que na atualidade câmeras fotográficas ou de vídeo são tão acessíveis que são acessórios comuns embarcados em telefones celulares e *notebooks*. Mesmo câmeras simples apresentam qualidade satisfatória para algumas aplicações. Por estas características, as câmeras são sensores de relevância inestimável para a robótica móvel. O uso deste tipo de sensor é essencial para que a robótica possa se tornar cada vez mais ubíqua, comercialmente viável e acessível a uma grande quantidade de pessoas.

Este trabalho é, portanto, motivado pela possibilidade da adoção do uso de câmeras de vídeo como principal sensor no robô, de modo a dispensar equipamentos caros em favor de

dispositivos comuns e de baixo custo, que possam eventualmente ser usados em aplicações cotidianas. Pretende-se também explorar a analogia entre sistemas detectores e descritores de características e a visão humana, e o trabalho busca demonstrar uma aplicação prática da robótica móvel, desenvolvendo um sistema útil em uma situação real de navegação autônoma de ambiente interno.

1.2 *Objetivo*

O objetivo deste trabalho foi a proposta, pesquisa, projeto e desenvolvimento de um sistema robótico móvel autônomo capaz de realizar localização utilizando como principal sensor uma câmera de vídeo monocular. O sistema deveria ser geral o bastante para localizar-se globalmente com a câmera de forma independente de contexto, e não ser excessivamente custoso em termos de processamento realizado em tempo de execução. A proposta envolvia também testar o sistema em um conjunto de imagens possível de ser encontrado em uma aplicação prática. Para estes fins, foi escolhido um método de extração e descrição de pontos distintos, ou características visuais distintas (em inglês, *features*) para localização. Buscou-se avaliar se o método é robusto o suficiente, em relação à possível aplicação em um sistema de patrulha de ambientes internos através de uma rota pré-determinada.

O projeto inspira-se no modo como uma pessoa usualmente se localiza em um ambiente onde já esteve previamente, reconhecendo elementos presentes neste ambiente que o caracterizam e servem de referência. Assim, a localização foi baseada em uma "memória visual" de lugares que definem trajetórias a serem novamente percorridas. Através da aquisição prévia de imagens espacialmente relacionadas, que definem uma trajetória e o espaço onde o robô pode se encontrar, procura-se identificar pontos de referência para localização associando as imagens da memória com a imagem atual percebida pelo robô.

Métodos chamados detectores-descritores de características de imagens podem ser usados para realizar esta associação de imagens através de características visuais distintas. Sabemos que o pareamento destes pontos de referência entre imagens (ou *frames*) permite que o robô identifique objetos ou cenários previamente conhecidos no ambiente, mesmo com mudanças de perspectiva, iluminação, rotação e diversas outras perturbações. Através do uso

do popular método SURF, ou *Speeded Up Robust Features* (Bay *et al.*, 2008), para detecção e descrição de características, espera-se que ao identificar uma mesma cena corretamente a partir de duas perspectivas ou escalas diferentes, o melhor número de pareamentos positivos (em inglês, *matches*) seja maior quando as perspectivas das imagens sejam mais próximas. Chamaremos pareamentos positivos simplesmente de pares. Este trabalho se propõe a utilizar essa relação e aplicar a diferença no número de pares entre diferentes perspectivas de uma mesma cena como critério para escolher a imagem que melhor representa a localização do robô em relação a uma cena, além de determinar a precisão de tal método. O trabalho também propõe um método alternativo para eleição da melhor imagem além do número de pareamentos. Em perspectivas mais próximas, os pareamentos em geral estão espacialmente mais próximos em suas respectivas imagens. O trabalho pretendeu testar este critério alternativo em comparação ao número de pareamentos. Ambos os critérios são exemplificados na Figura 1.3.

Para testar esta proposta, o sistema de localização e navegação descrito foi projetado utilizando apenas os resultados do método SURF com uma câmera monocular e a informação do sensor de odometria para estimativa de localização. O trabalho também estudou uma possível configuração do sistema com mais uma câmera acrescentada ao conjunto, visando avaliar os benefícios da adição do sensor em situações difíceis. O exemplo usado foi a travessia de portas, um desafio comum em ambientes internos, que exige precisão.

A tarefa de navegação interna aplica-se à pesquisa do INCT-SEC relacionada a robôs móveis de pequeno porte para ambientes internos. Para o trabalho proposto, o ambiente foi considerado imutável, e a patrulha foi definida como o percurso de um caminho pré-determinado, pelo qual o robô já passou anteriormente, conduzido por um ser humano em uma volta de gravação onde o robô pode gravar imagens do ambiente e a odometria associada. Acredita-se que o requisito de que haja um percurso prévio pelo ambiente seja uma exigência razoável, possível de se realizar na maioria das situações práticas de patrulha de instalações internas.

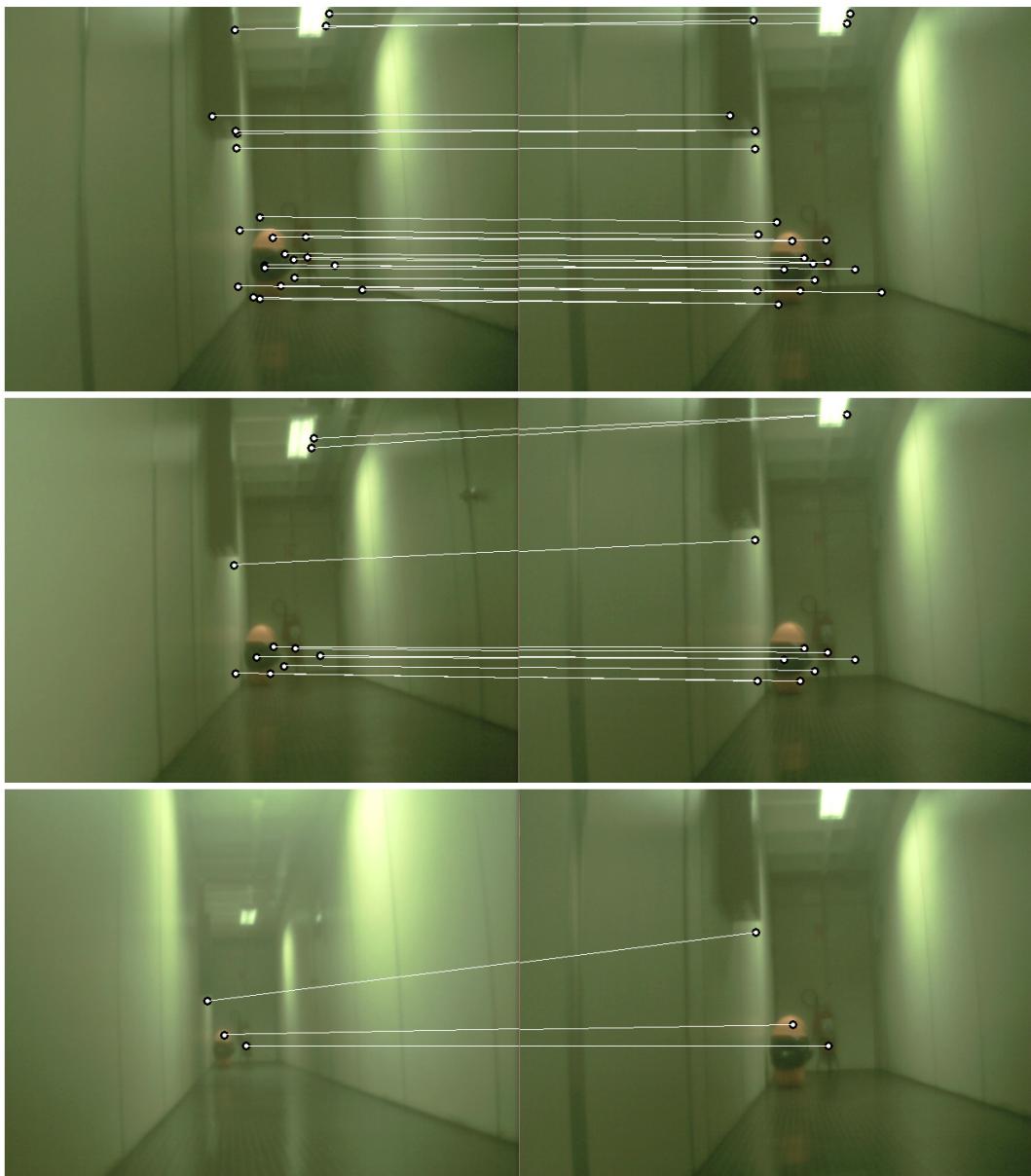


Figura 1.3: Exemplos de pareamento entre imagens. Perspectivas mais próximas geram mais pares de pontos, e os pares são mais alinhados horizontalmente.

Em resumo, seguem listados os principais objetivos específicos desta dissertação de mestrado:

- Pesquisa e desenvolvimento de um sistema eficiente de localização e navegação através apenas de informação visual e odometria;
- Teste da capacidade de localização de um robô através da avaliação dos resultados de pareamento do método SURF através de dois critérios, comparação do número de pareamentos e distância entre os pontos pareados;

-
- Desenvolvimento de uma aplicação do sistema voltada para patrulha de ambientes internos, para tarefas de monitoramento e segurança;
 - Validação através de simulações e testes em ambientes internos.

1.3 Estrutura da Monografia

O conteúdo dos capítulos seguintes desta dissertação pode ser dividido da seguinte maneira:

Capítulo 2: trata de fundamentos e de bases teóricas da robótica, especificamente da robótica móvel e das técnicas de localização, controle e navegação de robôs autônomos.

Capítulo 3: aborda a área de processamento de imagens, introduzindo o assunto e descrevendo o método SURF e alguns dos outros métodos importantes relacionados à extração de pontos de referência e detecção de elementos relevantes de uma imagem, e suas aplicações na navegação robótica.

Capítulo 4: descreve as ferramentas e métodos que foram usadas no desenvolvimento do projeto proposto, com detalhes do desenvolvimento.

Capítulo 5: apresenta os resultados dos testes e conclusões que podem ser obtidas a partir destes.

Capítulo 6: avalia os resultados obtidos, discute outras abordagens que podem ser utilizadas para aperfeiçoar os métodos e seus efeitos esperados, e possibilidades de trabalhos futuros.

2 Robótica Móvel

Um robô móvel pode ser considerado um agente inteligente autônomo, dotado de sensores e atuadores. Sendo um agente, o robô pode interagir com o meio ambiente, ou seja, realizar ações sobre este. Não apenas isso, mas um robô realmente inteligente deve levar em conta sua capacidade de percepção do ambiente e de seu próprio estado para tomar decisões sobre suas ações. Isso resulta, portanto, em um ciclo que envolve a percepção do ambiente, tomada de decisão através de um sistema de controle inteligente, realização de ação e finalmente a percepção da nova configuração do ambiente após a ação (Figura 2.1). Essa estrutura rege toda a interação entre o agente e o ambiente que o cerca, para qualquer que seja a função desempenhada. Podemos perceber daí a importância dos sensores e atuadores, e também do algoritmo de controle embarcado no robô.



Figura 2.1: O ciclo de percepção e ação do robô.

2.1 Sensores e Atuadores

O retorno, ou *feedback*, que o robô recebe do ambiente, é obtido através dos sensores. Um sensor pode ser definido como um dispositivo que realiza uma medida no ambiente, e fornece o resultado. O sensor *laser*, por exemplo, realiza uma medida de distância através da emissão de um feixe de *lasers*, retornando o valor numérico da distância medida por cada raio emitido em uma unidade apropriada. Além do *laser*, sensores modernos frequentemente usados na robótica móvel incluem câmeras de vídeo, sistema de GPS (*Global Positioning System*), sensor de iluminação, de contato (como *bumpers*), infravermelho, sonar, sensores iniciais (giroscópios e acelerômetros), etc. Além destes sensores externos (também chamados de exteroceptivos), existem os chamados sensores proprioceptivos, que se referem à capacidade do robô de perceber seu próprio estado (Bekey, 2005). Em um veículo, por exemplo, pode-se citar entre sensores proprioceptivos comuns o medidor de odometria, medidor de combustível ou bateria, contador de giros, entre outros.

O sensor adequado depende do objetivo do robô, ou da tarefa que se deseja realizar. Sensores como o sonar, o *laser* e o LiDAR (*Light Detection and Ranging*, um tipo de sensor *laser*) são chamados de sensores de distância (ou sensores de *range*), pois fornecem diretamente medidas de distância entre o emissor e obstáculos, mas possuem alcance limitado. São amplamente difundidos e muitos oferecem medidas de distância com alta precisão. Enquanto isso, por exemplo, sensores como a IMU (do inglês, Unidade de Medida Inercial) podem ser úteis para aeronaves autônomas manterem sua orientação em relação ao solo, caso no qual o sensor *laser* seria de pouca valia.

Os sistemas de visão podem ser úteis em diversas aplicações. Em relação aos exemplos citados, sensores visuais podem ser usados para detectar obstáculos e estimar distâncias com câmeras estéreo (Bankman, 2006; Mendes e Wolf, 2011), da mesma forma que o cérebro humano é capaz de ter noções de profundidade através da diferença na imagem captada por cada um dos olhos. Câmeras estereoscópicas promovem robustez por redundância de informações (Jones *et al.*, 1997). Em certos contextos, a profundidade pode ser calculada até mesmo com câmeras monoculares (Delage *et al.*, 2005). Sensores visuais também podem ser usados em veículos aéreos para estimativa de movimento e orientação em relação ao

horizonte (Dusha *et al.*, 2007; Todorovic, *et al.*, 2003). Câmeras omnidirecionais são alternativas interessantes por proverem ângulos de observação amplos, facilitando a localização de marcos no ambiente e tornando possível manter elementos de interesse no seu campo de visão mais facilmente. Ainda assim, configurações estéreo de câmeras necessitam de calibração precisa, e câmeras omnidirecionais são itens mais especializados e custosos que câmeras monoculares. Com o aperfeiçoamento de métodos e técnicas que façam uso inteligente da riqueza de informações provida, câmeras monoculares têm se mostrado uma opção viável na prática.

É importante destacar que sensores não são dispositivos perfeitos, e possuem erros associados às medições, que fazem com que os dados observados nem sempre correspondam exatamente ao valor real. Podem ocorrer desvios de medida causados pelo truncamento do valor medido (devido à resolução máxima limitada do sensor ou no processo de digitalização do valor observado) ou mesmo imperfeições na detecção. Devido à natureza suscetível a erros dos sensores, é possível também que os erros das leituras não sejam uniformemente distribuídos (o sensor possui um viés), ou mesmo que as faixas de erro sejam variáveis, como no caso do GPS, em que o erro das medições depende do número de satélites rastreados pelo aparelho, altitude e obstáculos físicos, de acordo com experimentos realizados no Laboratório de Robótica Móvel do ICMC (LRM - ICMC) por Gustavo Pessin. A aplicação deve ser considerada para desenvolver um controle inteligente que incorpore esses erros na tomada de decisões. O erro pode ainda ser relativo a cada medida específica de maneira pontual, ou cumulativo. Um exemplo clássico de sensor com erro cumulativo é o sensor de odometria, que mede a distância do deslocamento do robô e os ângulos de suas curvas medindo o giro e orientação das rodas e eixos. Neste caso, o erro de uma medição é somado aos erros de medições anteriores, o que, mesmo em um sensor sem viés maior, faz com que o erro em relação à origem aumente conforme a distância percorrida aumenta. A odometria por si só é pouco utilizada por essa razão. É habitual na robótica, portanto, que outros sensores sejam considerados juntamente com a odometria para corrigir o erro odométrico, e de fato, muitos trabalhos de pesquisa utilizam essa técnica (Siegwart e Nourbakhsh, 2004).

A fusão de sensores é uma prática comum, e não só em relação à odometria. A fusão de sensores consiste da obtenção de leituras do ambiente através de mais de um tipo de sensor, e da combinação dos resultados adquiridos visando compensar deficiências de cada

sensor e tirar proveito de uma maior quantidade de informação. Tem-se deste modo uma melhor aferição do ambiente, assim como a percepção humana consiste de um conjunto de informações de diversos sentidos, e não apenas de um. É necessário que haja um sistema de controle que considere essas informações e utilize as informações obtidas de maneira integrada, como em (Sim e Dudek, 2003) e (Kidono *et al.*, 2002).

Os atuadores, por sua vez, são os dispositivos através dos quais o robô realiza ações sobre o ambiente e altera seu próprio estado. Em robôs móveis terrestres, os atuadores mais comuns são os motores e rodas do robô, embora existam muitos robôs que se movimentem com pernas mecânicas, esteiras, ou mesmo propulsores, turbinas e hélices. O conhecimento dos atuadores do robô é importante para projetar um sistema que considere as limitações e possibilidades do robô, já que atuadores também podem apresentar imprecisões. De especial importância para a locomoção do robô é o conhecimento do modelo cinemático e dinâmico do robô, que descreve sua trajetória e seus movimentos.

No caso de robôs móveis terrestres, é possível classificar os robôs em holonômicos e não-holonômicos. Robôs holonômicos são robôs que podem se movimentar livremente sobre seu próprio eixo, realizando rotação e mudanças de orientação sem precisar se deslocar (Dudek e Jenkin, 2000). Este é um conceito comumente aplicado a robôs móveis, mas que também pode se referir a um braço mecânico, por exemplo. Robôs não-holonômicos, por outro lado, não possuem essa liberdade, e precisam se deslocar para virar (Laumond, 1986). Um automóvel como um carro é o exemplo clássico de um veículo não-holonômico, que só pode se deslocar no sentido em que seus eixos estão alinhados. Para alterar sua orientação, o veículo precisa mover suas rodas, alterando a direção do eixo normal a elas, e então se deslocar. Essa geometria é conhecida como cinemática de Ackerman, ou *Ackerman steering* (Osório *et al.*, 2009). Em robôs não-holonômicos, é preciso considerar a distância em relação a obstáculos e o vetor velocidade do robô para que a navegação possa evitar obstáculos adequadamente. Como esta cinemática replica o funcionamento de veículos tradicionais, seu estudo é de grande interesse, e trabalhos que exploram navegação de um automóvel autônomo incorporam as restrições deste modelo no planejamento de trajetória.

2.2 Mapas e Localização

Muitas tarefas mais elaboradas de robótica móvel requerem o uso de um mapa, alguma ferramenta que ofereça informações ao robô sobre o ambiente além do alcance instantâneo de seus sensores, e através da qual o robô possa conhecer sua posição relativa. Para ir de um ponto A (origem) para B (destino) é necessário que o robô saiba onde está e onde está seu objetivo. Assim, para que o controle da navegação possa ser realizado, é necessária a determinação da localização do robô, e portanto, pressupõe-se o uso de algum tipo de mapa ou referência que permita manter este controle da posição atual e do destino a ser alcançado. Não pode haver planejamento global de trajetória sem que haja um conhecimento do ambiente que se estenda além das redondezas imediatas do robô. O mapa é uma forma de descrição do ambiente, uma ferramenta que permite que o robô tenha conhecimento do perfil da área. Existem diversas técnicas de navegação robótica em que o próprio robô realiza o mapeamento do ambiente (Bonin-Font *et al.*, 2008), e inclusive métodos para que o robô possa mapear o ambiente e se localizar ao mesmo tempo, como o SLAM¹⁰, ou *Simoultaneous Localization and Mapping* (Leonard e Durrant-Whyte, 1991).

O uso de um mapa é de pouca valia se o robô não souber onde está localizado no ambiente para que possa planejar sua trajetória. Logo, a determinação e a manutenção de informação sobre a pose do robô no ambiente são elementos importantes para sistemas de controle de navegação mais complexos. Define-se a pose do robô não necessariamente somente como sua posição, mas também sua orientação, e em alguns casos até o estado de outros atuadores.

Já mencionamos o erro cumulativo intrínseco da odometria. Uma das soluções amplamente utilizadas para corrigir o erro incrementalmente é associar informações provindas de outros sensores ao odômetro para aumentar a precisão da localização (Thrun, 1998) (Siegwart e Nourbakhsh, 2004). O uso conjunto de sensores, como a câmera ou o *laser*, e o odômetro, permite que as observações atuais e as passadas sejam usadas para se estimar as possíveis posições do robô. Utilizando métodos desse tipo, com informação suficiente o robô pode se localizar com alto grau de certeza. Em geral, as percepções passadas são os próprios

¹⁰ <http://www.cas.kth.se/SLAM/slam-papers.html>, acesso em 01/2012.

mapas em si. Com isso, é essencial que os mapas não contenham erros para evitar falhas no processo de localização. De uma forma geral, o problema de localização pode ser classificado em dois principais tipos (Thrun *et al.*, 2005):

- **Rastreamento da Posição (Position Tracking):** Neste problema, considera-se que a posição inicial do robô em relação ao mapa (posição global) é conhecida. Com isso, o desafio consiste em determinar localmente a posição do robô enquanto ele se desloca e mantê-la, levando-se em conta a localização prevista pelo movimento, com o tratamento de erros dos sensores de odometria acumulados. Em geral, o *position tracking* é um método iterativo, baseado em pequenos e constantes incrementos à estimativa de posição. Com uma frequência suficiente de incrementos, pressupõe-se que esses erros sejam pequenos entre cada iteração. Para controlar o erro odométrico, um sensor como a câmera pode ser usada para ajustar a posição do robô. Se com esta fusão de sensores o erro de odometria puder ser anulado (ou ao menos limitado) a cada iteração, ele não sairá do controle. Os erros de posição normalmente podem ser aproximados por uma distribuição unimodal.

- **Localização Global (Global Localization):** Ao contrário do *position tracking*, nos problemas que envolvem localização global não se conhece a posição inicial do robô no mapa. Dessa forma, procuram-se estratégias para obter a localização global do robô, considerando-se como no caso anterior, a possibilidade de erros na estimativa da posição. Na localização global usualmente busca-se determinar qual é a posição atual do robô em relação a todo o mapa (toda a área útil onde este possa estar localizado).

Também vale mencionar o que é referenciado na literatura como o problema do robô sequestrado (*kidnapped robot problem*). Este na realidade é um caso particular da localização global. No caso geral da localização global, é possível que o robô tenha acesso a informações anteriores à perda de localização, mas no problema do robô sequestrado o robô não possui informação alguma sobre sua posição antes do início da execução (Choset *et al.*, 2005). Assume-se que o robô poderá de um instante para outro ser levado para outro lugar por influências externas, sem que seus sensores percebam, ou que ele possa ser ativado em um local qualquer do ambiente. Por causa disso, o robô deverá reconhecer quando a sua posição foi alterada, para que ele possa novamente calcular a sua referência no mapa. A análise deste problema é importante para localização inicial do robô quando ele é inicializado em uma posição completamente desconhecida no ambiente, ou para que o robô possa se recuperar de

situações durante a navegação onde ocorre falha no rastreamento de posição para manter sua localização local (quando o ambiente observado não corresponde ao esperado).

Existem técnicas variadas para estimativa de posição, gerais ou específicas, simples ou complexas. Em relação aos métodos de localização global, muitos focam no uso de métodos estatísticos (Thrun *et al.*, 1999). Entre eles destaca-se o uso do Filtro de Kalman, métodos de Monte Carlo (Dellaert *et al.*, 1999) e processo de Markov (Fox *et al.*, 1999).

Seja pelo uso de um mapa da região criado pelo próprio robô em uma tarefa de mapeamento, ou seja, através de mapa conhecido previamente, o uso de mapas pode ser de grande auxílio para a navegação e localização, mesmo considerando que aplicações como o SLAM utilizam mapas incompletos, com erros ou com elementos desconhecidos. Sendo uma representação de um ambiente muitas vezes dinâmico e imprevisível, o mapa está sujeito a não ser uma representação completamente fiel da realidade. Os algoritmos de mapeamento, localização e controle devem ser robustos para incorporar tais diferenças, e muitas vezes serem capazes de incorporar o dinamismo e as alterações do ambiente ao mapa como em (Wolf e Sukhatme, 2003).

Muitas aplicações procuram resolver estes mesmos problemas através do uso de câmeras. No âmbito da localização local, podemos citar técnicas de *visual servoing* (Kragic e Christensen, 2002) e fluxo óptico (Horn e Schunck, 1993; Lucas e Kanade, 1981). Câmeras também podem ser aplicadas em técnicas de mapeamento, por exemplo, na reconstrução tridimensional de ambientes ou objetos¹¹ (Moravec, 1988; Wooden, 2006; Saxena *et al.*, 2007). Aplicações práticas de métodos gerais de detecção de características como o SURF podem ser vistas em (Chen *et al.*, 2007), um sistema com uma implementação de SURF para celulares capaz de detectar marcos previamente memorizados do ambiente. Entre trabalhos envolvendo localização e mapeamento simultâneo (SLAM) baseados em visão podemos citar (Kaess e Dellaert, 2009) e (Sim *et al.*, 2006; Sim e Little, 2006). Mapas formados por composições de fotografias do ambiente ou por sequências de imagens também são usados.

Existem muitos tipos de mapas, e através do uso de um mapa apropriado à aplicação e aos sensores, aperfeiçoa-se o conhecimento do robô em relação ao ambiente. Muitos mapas apresentam informações semânticas, associando significado e função aos elementos no mapa, para que o robô possa interagir com esses elementos de maneira lógica e contextualizada. O

¹¹ <http://homes.esat.kuleuven.be/~visit3d/webservice/v2/>, acesso em 01/2012.

conhecimento semântico permite ao robô níveis mais elaborados de autonomia e planejamento (Calisi *et al.*, 2007).

Em relação a robôs na forma de veículos terrestres, o mais comum é o uso de mapas bidimensionais (2D), aplicações em terrenos externos acidentados podem considerar o relevo do ambiente, mas em áreas internas pode ser desnecessário. No caso de veículos aéreos e aquáticos/submarinos, raramente é possível ignorar qualquer uma das três dimensões do espaço. Isto adiciona uma carga extra de complexidade a esses tipos de aplicação. Nesse caso, é comum o uso de mapas tridimensionais (3D).

Existem diversas formas de representar o ambiente através de um mapa. Na robótica móvel, é possível citar dentre estes os tipos de mapa dignos de nota:

- **Mapas métricos:** definem o ambiente por um sistema único de coordenadas, baseado na representação de distâncias em escala;

- **Mapas topológicos:** o mapa topológico, em contraste com o mapa métrico, não contém informação geométrica ou geográfica, mas é um mapa qualitativo que representa o ambiente através de locais determinados distintivamente e como estes se conectam relativamente uns aos outros. Desta forma, um grafo pode representar o mapa topológico de algum lugar. Em geral esse tipo de mapas não apresenta distâncias absolutas ou coordenadas. Pode ser representado muitas vezes por um grafo, e é também chamado de mapa qualitativo (Kortenkamp *et al.*, 1988). Mapas topológicos são simples e compactos, menos custosos computacionalmente e são apropriados para navegação qualitativa de longa distância e planejamento de trajetória (DeSouza e Kak, 2002; Sales *et al.*, 2011);

- **Mapas em grid de ocupação:** uma variante específica do mapa métrico, o mapeamento por grade de ocupação (ou *occupancy grid*) (Elfes, 1989; Moravec, 1988) baseia-se na decomposição do espaço 2D ou 3D em várias células independentes, sendo que cada célula possui uma estimativa probabilística do seu estado (e.g. há 90% de probabilidade de o robô estar localizado em determinada célula);

- **Mapas sensoriais:** são basicamente mapas construídos a partir dos dados capturados do ambiente pelos sensores, de forma crua. Podem ser, por exemplo, um mapa feito pela combinação de nuvens de pontos capturadas por *lasers* ou mapas feitos pela união de fotografias da área, fornecendo uma visão panorâmica do ambiente. O SLAM em sua forma mais básica produz esse tipo de mapas, mas mapas sensoriais não estão limitados a esta

técnica. Um mapa sensorial pode ter sido criado previamente pelo robô, ou mesmo com o sensor independentemente.

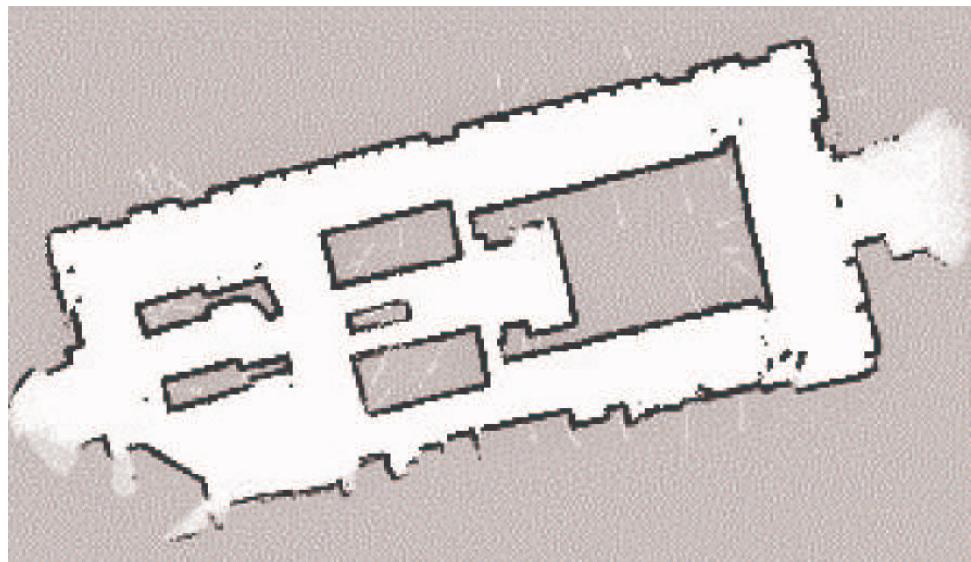


Figura 2.2: Exemplo de mapa métrico (S. Thrun).

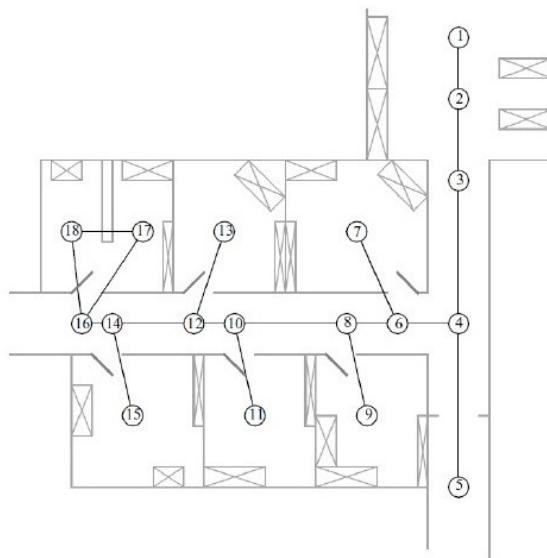


Figura 2.3: Um mapa métrico pode ser decomposto em nós ou células, se tornando um mapa topológico, representado na figura pelo grafo (Siegwart e Nourbakhsh).

É importante mencionar os diversos trabalhos baseados na integração e uso simultâneo de diferentes tipos de mapas (Dudek e Marinakis, 2007; Bazeille e Filliat, 2009; Thrun, 1998, Thrun *et al.*, 1998), pois estes representam formas de se usufruir das qualidades de diferentes

mapas, o que implica em maior riqueza de informações. Diferentes tipos de mapas oferecem diferentes representações de informação e dados diferentes, e tem custos associados (computacional, de *hardware*, de complexidade). A escolha do mapa tem impacto posteriormente na localização e navegação.

2.2.1 Métodos Probabilísticos Para Localização

Já foi estabelecido que a localização autônoma do robô no ambiente é tarefa de extrema importância. Ela precede e habilita a navegação, e um robô que não conhece sua pose em relação ao ambiente não será capaz de realizar tarefas mais elaboradas de movimentação ou dependentes do contexto de uma área específica do ambiente. Como mencionado, se a localização inicial do robô for conhecida (uma base de recarga de bateria ou algum tipo de *checkpoint*, por exemplo) o robô deverá manter-se localizado localmente ao navegar a partir desse ponto. Se a localização inicial não for conhecida, existem diversos métodos pra obter a localização global do robô. Uma classe de métodos que podem ser usados para localização são os algoritmos de estimativa probabilística de estado. O propósito básico destes algoritmos é recuperar as variáveis relevantes de estado do robô (que consistem em geral de posição, orientação e velocidade) ou mesmo do ambiente (para ambientes dinâmicos) a partir da leitura de dados do sensor e análise estatística.

O Filtro de Kalman, por exemplo, é uma das primeiras implementações do chamado Filtro de Bayes e é amplamente usado na estimativa de estado (e consequentemente, aplicável à estimativa de localização) (Kalman, 1960). É um algoritmo geral para cálculo de confiança, ou *belief*, sobre as variáveis de estado do robô, usando dados captados do ambiente (Thrun *et al.*, 2005). O Filtro de Kalman divide-se em um passo de predição e um passo de atualização, ou correção das medidas, e é um filtro com diversas aplicações de controle em muitos ramos da engenharia. De maneira geral, o fluxo da aplicação do Filtro de Kalman baseia-se na estimação do estado inicial, a estimação (ou predição) de um novo estado resultante de uma ação de controle, e finalmente a atualização deste estado predito a partir de uma nova observação. Assim, o filtro combina o estado esperado, ou predito, com o estado observado. O filtro obtém resultados ótimos para sistemas lineares, com variáveis com erros de média zero

e independentes entre si, mas pode funcionar quando essas hipóteses não forem respeitadas, e existem variantes e extensões do filtro (como o EKF, *Extended Kalman Filter*) que deixam a optimalidade em segundo plano para acomodar melhor não-linearidades e possíveis falhas de leitura no sensor, ou mesmo para incorporar densidades de probabilidade multi-modais, habilitando a localização global por Filtro de Kalman (Lefebvre *et al.*, 2001).

Outro método de localização que vem recebendo grande notoriedade é o método de Monte Carlo – MCL (*Monte Carlo Localization*) (Dellaert *et al.*, 1999). É uma variação do método conhecido como Filtro de Partículas, e baseia-se na distribuição aleatória de possíveis estados em um espaço de configurações, no caso, as áreas navegáveis do mapa. Cada estado é representado por uma partícula, e é usado um número grande de partículas, fator que influencia na precisão e velocidade de convergência, mas também no peso computacional do algoritmo. A quantidade de partículas deve ser suficiente para uma cobertura adequada do mapa. A partir disso, cada ação de controle do robô (como movimentar-se no ambiente) é aplicada às partículas a partir do modelo estatístico de movimentação (de forma semelhante ao que ocorre no filtro de Kalman). Além disso, cada observação real do estado dos sensores é comparada à observação das partículas (estados hipotéticos). Dependendo da proximidade entre a leitura do sensor e a leitura estimada para cada partícula, essa partícula recebe um peso, indicando sua probabilidade (ou *likelihood*) de ser representativa da posição real do robô. A partir disso, realiza-se uma nova seleção aleatória de amostras das possíveis configurações, desta vez levando-se em conta os pesos das partículas. Quanto maior a *likelihood* de uma partícula, maior a probabilidade de que ela seja escolhida novamente, o que com o tempo acarreta na aproximação da real posição do robô, representada pela partícula com a melhor estimativa da posição. Uma vantagem do sistema é que se o robô for transladado de sua posição ele pode se localizar de novo naturalmente, já que as partículas se espalharão pelo mapa em seleções (ou *samplings*) posteriores. Na Figura 2.4 temos um exemplo visual do funcionamento de uma iteração do algoritmo de Monte Carlo.

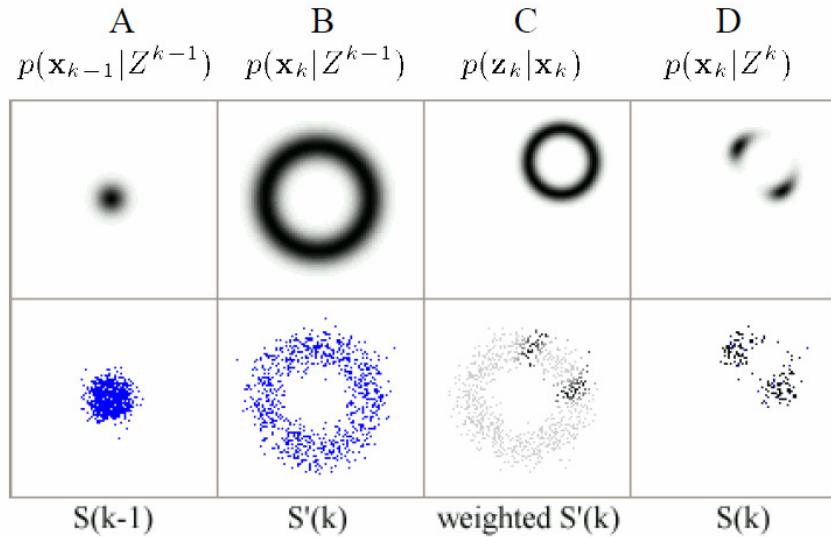


Figura 2.4: O funcionamento da localização Monte Carlo. No quadro A, a nuvem de partículas representa a incerteza quanto à posição do robô. No quadro B, o robô andou um metro para a frente. Como não sabemos sua orientação inicial, as partículas se espalham como no círculo. No quadro C, o robô observa um ponto de referência próximo no canto superior direito, e a *likelihood* das partículas ali aumenta. O quadro D mostra as novas partículas após o *resampling*. A partir daí o ciclo reinicia.

Uma conhecida e pioneira aplicação prática do algoritmo foi realizada com o robô MINERVA, um robô usado como guia no Smithsonian National Museum of American History, com bons resultados (Thrun *et al.*, 1999). O algoritmo também pode ser aplicado em técnicas de SLAM, com bons resultados (Yuen e MacDonald, 2003). O método, no entanto, possui algumas limitações, como a já mencionada necessidade de um mapa previamente fornecido do ambiente, que deve ser razoavelmente preciso, e a impossibilidade de se localizar adequadamente (localização global) em um ambiente muito dinâmico.

O desafio da localização é um processo constante, que por esse motivo deve ser preciso e eficiente. É um assunto importante que está diretamente relacionado a diversas aplicações, sendo o foco principal deste trabalho.

2.3 A Navegação e Controle Robótico

Uma vez localizado, o robô tem os recursos necessários para alcançar seu objetivo, mas ainda resta a realização do percurso. Há um espectro variado de possíveis soluções para este passo (Bonin-Font *et al.*, 2008).

O algoritmo de controle inteligente do robô é o sistema pelo qual o robô transforma a leitura dos sensores (entrada) em conclusões inteligentes sobre o ambiente e ações (saída). No caso de um robô móvel movendo-se de um ponto A para um ponto B, a ação é o deslocamento do robô, a navegação no ambiente. A atuação de um robô no ambiente influenciará seu estado e as leituras posteriores de seus sensores. O sistema deverá incorporar esta nova informação para tomar novas decisões contextualizadas. Este ciclo, portanto, pode ser considerado um sistema de controle em malha fechada (Castrucci e Sales, 1990; Åström e Murray, 2008). Existem inúmeros métodos e técnicas de controle de um robô móvel para navegação enquanto mantém a localização, e muitas variantes dependentes de cada situação, já que dependendo da aplicação do robô outras variáveis podem estar envolvidas. Bons resultados no controle robótico de navegação dependem do projeto cuidadoso da malha (Åström *et al.*, 1993) e calibragem de seus parâmetros cinemáticos (Kümmerle *et al.*, 2011). Um robô móvel com alta velocidade linear não será muito manobrável se sua velocidade angular não for também alta para esterçá-lo a contento, por exemplo.

Os tipos de arquiteturas de controle para robôs móveis podem ser divididos em controle reativo, controle deliberativo e controle híbrido (Szabo, 2003). No controle reativo, o robô realiza a tomada de decisão baseando-se nas características percebidas localmente, reagindo a obstáculos ou estímulos de forma imediata. O uso isolado do controle reativo, sem qualquer planejamento de futuros movimentos, rejeita toda a informação fora das redondezas imediatas do robô (região percebida pelos sensores), e assim pode restringir as possibilidades de locomoção inteligente. Há também o risco de que o robô caia em um mínimo local (Piaggio *et al.*, 1997; Luh e Liu, 2008), sem capacidade de perceber este problema, ficando muitas vezes bloqueado neste tipo de situação. Entretanto, a importância do controle reativo está em evitar movimentos locais incorretos, como batidas não-intencionais em paredes ou objetos, por exemplo, e é um método que em geral apresenta baixo custo computacional. É importante que o robô tenha um sistema de segurança reativo que tenha prioridade sobre o controle comum para evitar acidentes ou quaisquer movimentos indesejados. O controle reativo também é bastante usado em sistemas multirrobóticos ou de robótica de enxame (*swarm robotics*), baseados no comportamento de colônias de insetos, para fazer emergir comportamentos globais complexos a partir de comportamentos isolados simplificados (Parker e Zhang, 2006).

Já aludimos ao controle deliberativo quando mencionamos a necessidade de planejamento de trajetória em muitas aplicações. Neste caso, o robô realiza previamente todo o cálculo necessário de trajetória e estabelece um percurso, ou uma sequência de ações determinada antes da execução. Um robô não-holonômico, sujeito ao modelo cinemático de Ackerman, por exemplo, necessita deste planejamento para evitar colocar-se em situações em que deverá realizar manobras redundantes ou delicadas. Neste tipo de arquitetura de controle, assume-se usualmente a utilização de um mapa do ambiente pelo robô, sendo um método de controle mais custoso computacionalmente. A maior deficiência da forma de controle estritamente deliberativa é a não tolerância a alterações no ambiente e a presença de obstáculos não previstos. O controle reativo e controle deliberativo também são chamados de métodos locais e globais de desvio de obstáculos, respectivamente (Siegwart e Nourbakhsh, 2004).

Devido às deficiências dos controles reativo e deliberativo quando adotados isoladamente, são comuns técnicas de controle que reúnam os dois controles de forma simultânea, compondo uma arquitetura de controle hierárquica. Nas abordagens híbridas há o planejamento global, deliberativo, e também o desvio de obstáculos local, este último normalmente utilizado para evitar acidentes e reagir a alterações do ambiente não previstas no planejamento de trajetória.

A navegação pode ser definida como a escolha e o percurso de um caminho adequado entre a posição atual de um robô e seu alvo (Choset *et al.*, 2005). A localização simplifica e, muitas vezes, habilita a navegação (Thrun *et al.*, 2005). Tarefas robóticas dinâmicas, que exijam um grau de autonomia elevado, assim como tomada de decisão e tolerância a falhas, comumente requerem que o robô seja capaz de manter-se localizado no ambiente. O algoritmo de navegação é altamente dependente do objetivo do robô e da calibragem de diversos parâmetros de controle, e muitas vezes é projetado através de heurísticas, adaptando-se do melhor modo a uma tarefa específica. É possível citar, entre possíveis tarefas que exigem navegação, atividades como o patrulhamento, em que o robô deve percorrer de maneira otimizada as áreas de interesse, realizando um percurso cíclico, tarefas como o deslocamento ou transporte de objetos como em robôs empilhadeira em armazéns, ou navegação em formação para esquadrões de múltiplos robôs.

Existem diversas técnicas para planejamento de trajetória e para encontrar caminhos em um mapa. Em mapas topológicos, diretamente associados a grafos, podem ser usadas técnicas de busca por caminhos em grafos, das quais a mais tradicional é provavelmente o algoritmo de Dijkstra (Dijkstra, 1959). Mapas métricos podem ser simplificados com grades (*grids*) de ocupação. A partir da grade de ocupação, pode-se aplicar o algoritmo A* (A-Star ou A-Estrela)¹² ou uma de suas variantes¹³ como o D* (D-Estrela) (Stentz, 1995) para encontrar caminhos ótimos (*path finding*). Alternativamente, é possível transformar o mapa métrico em um grafo de visibilidade (De Berg *et al.*, 2008) e aplicar um algoritmo como Dijkstra. O grafo de visibilidade é um grafo que compõe o mapa do ambiente em locais (os nós) e os relaciona pelo critério de visibilidade (uma aresta significa que os nós ligados têm visada livre entre si, o que quer dizer que o deslocamento de um nó para outro pode ser feito de forma trivial e direta, sem desvio significativo de obstáculos). Técnicas como Campos de Força Virtuais (Virtual Force Fields) podem ser aplicadas ao mapa para direcionar a navegação (Kim e Nevatia, 1999). Estes métodos citados consideram a existência de um mapa do ambiente, e a também o conhecimento da localização (*pose*) do robô para fins de navegação.

Independentemente da aplicação, é interessante mencionar o uso difundido de Redes Neurais Artificiais (RNAs) (Haykin, 1999) na robótica. Buscando simular a estrutura e funcionamento das redes neurais biológicas, as RNAs são importantes para a robótica por serem bastante adaptativas, capazes de detectar padrões com eficácia e eficientes na resolução de problemas ambíguos. O uso das RNAs na robótica móvel pode englobar desde algoritmos de navegação (Shinzato, 2010; Souza *et al.*, 2011) até algoritmos de reconhecimento de padrões de leituras nos sensores, reconhecimento de gestos ou objetos (Bonato, 2004). Além disso, outras técnicas de aprendizado de máquina e lógica *fuzzy* são bastante usadas. Algoritmos Genéticos, que se baseiam na teoria da evolução para comparar a qualidade de soluções, podem ser usados para encontrar soluções ótimas para problemas em sistemas de controle robótico cujo funcionamento (as funções matemáticas que regem o sistema) é muito complexo ou desconhecido (Simões, 2000). Métodos como estes podem ser usados para

¹² <http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html#S3>, acesso em 01/2012.

¹³ <http://www.policyalmanac.org/games/aStarTutorial.htm>, acesso em 01/2012.

reconhecer situações e elementos do ambiente, assim como para decidir a melhor ação a ser realizada em relação a uma determinada situação.

2.4 A Alternativa dos Sensores Visuais

Como é possível notar, os métodos e trabalhos apresentados trazem conceitos e soluções importantes para localização. Tais métodos são bastante focados em sistemas com sensores de alcance, ou *range*, e precisão, como laser, infravermelho e sonar, devido à relação direta entre a distância, dado facilmente obtido através desses sensores, e o mapa métrico. Apesar disso, é possível encontrar trabalhos que apliquem os métodos descritos usando marcas de referência obtidas com câmeras, como a aplicação do método de localização de Monte Carlo em sistemas com câmera usados no campeonato de futebol de robôs, a RoboCup (Menegatti *et al.*, 2004; Heinemann *et al.*, 2006). Em relação aos sensores de distância, o alto erro de medição dos sonares faz com que eles estejam caindo em desuso especialmente para aplicações em ambientes internos devido à imprecisão. O *laser*, por sua vez, de fato se apresenta como uma boa solução, a princípio, para diversas aplicações. Existem, porém, as dificuldades já citadas, como o alto custo do *laser*, e as dificuldades intrínsecas de alguns dos métodos expostos, tais como a possibilidade de não-convergência. Além disso, há o fato de que um feixe pode atravessar obstáculos intransponíveis como vidros ou redes e não pode atravessar obstáculos facilmente transponíveis como folhas de um ramo de árvore, e todos estes são fatores difíceis de contornar apenas com a informação de distância. Sensores *laser* também apresentam alto consumo de energia e sua potência deve ser controlada para não se tornar perigoso ao ser humano. Há a alternativa do uso de câmeras de vídeo no projeto da localização e navegação, através de técnicas de processamento e manipulação de imagens. Muitas das técnicas com câmeras para localização usam câmeras estereoscópicas ou omnidirecionais. Este trabalho porém, insere-se e concentra-se no conjunto de técnicas de visão e robótica com câmeras monoculares, dentre as quais é possível citar: SLAM visual monocular (Davison *et al.*, 2007); métodos de reconstrução estrutural a partir de imagens monoculares (*Structure From Motion*) sem necessidade de calibragem; odometria visual monocular, onde se recupera o movimento da câmera observadora a partir da sequência de

imagens obtida (Fitzgibbon e Zisserman, 1998). O uso de uma única câmera ou de câmeras monoculares não calibradas é um desafio, mas é uma abordagem de mais baixo custo e, desse modo, realista e de grande importância do ponto de vista comercial e prático.

3 Visão Computacional e Processamento de Imagens

O processamento de imagens é a área da computação que se refere à manipulação e operações com imagens. Ao processamento e transformação de dados de imagens aplicado à computação de forma inteligente, interpretativa, de modo a extrair informações, novas representações e/ou conclusões a partir dos dados da imagem, dá-se o nome de visão computacional (Forsyth e Ponce, 2003). Não apenas importante na robótica móvel, encontramos aplicações de visão computacional em diversos tópicos da ciência, desde sistemas de reconhecimento de movimentos ou rostos, sistemas de realidade aumentada (Azuma, 1997), automatização de análise de imagens (útil para grandes bancos de imagens, como em registros de câmera de segurança), até áreas interdisciplinares como a interpretação de imagens médicas e biológicas.

Imagens são consideradas sinais bidimensionais. São também sinais no domínio espacial, assim classificados porque sinais de imagem são variantes no espaço (podendo ou não variar com o tempo), contrastando com sinais como o som ou impulsos elétricos, que variam no tempo (Haykin e Van Veen, 2002). Como qualquer sinal, computacionalmente é impossível representar o sinal análogo, que é infinito, sendo que apenas imagens digitais pertencem ao escopo deste trabalho. Imagens digitais podem ser definidas como funções bidimensionais compostas por um conjunto de elementos finitos, os *picture elements*, normalmente abreviados como *pixels*, que possuem localização e valor particulares (Gonzales e Woods, 2008). Imagens fotográficas com características similares à visão humana são as mais comuns e de especial interesse para o trabalho, mas outras aplicações podem evolver imagens sintéticas, omnidirecionais, imagens térmicas (ou em geral imagens representando luz de fora do espectro visível ao ser humano), entre outras.

3.1 Representação Digital de Imagens

Quando uma fotografia é obtida através de uma câmera digital, em algum momento a informação do ambiente foi transferida do formato analógico para o formato digital (a versão

da fotografia que pode ser visualizada em um computador ou aparelho digital). O sinal analógico, ou a luz que incide através do obturador da câmera, precisa ser convertido em uma sequência de *bits* que possa ser lida por um computador. O mesmo processo é necessário quando uma fotografia revelada de um filme de câmera analógica é digitalizada. Esse processo envolve duas etapas principais, presentes em qualquer conversão A/D (análogo para digital) em processamento de sinais: a amostragem e a quantização (Haykin, 2002). Na amostragem, o sinal contínuo obtido do ambiente, no caso a luz, deve ser discretizado. A discretização consiste da divisão do sinal em um número finito de valores computacionalmente viáveis. Em sinais de imagem, este é o processo que define a resolução da imagem resultante, e portanto a quantidade e valor dos *pixels* na matriz de *pixels*.

A etapa seguinte é a quantização, que envolve a aproximação da faixa contínua de valores obtidos do sinal analógico para um conjunto, ou uma escala, de valores finitos. Esse processo determina a profundidade (*depth*) da imagem digital resultante, que é o valor individual de cada *pixel*. Um valor de quantização comumente usado em imagens de tons de cinza é a quantização em 8 *bits*. Com cada *bit* podendo assumir 2 valores diferentes, 0 ou 1, temos 2^8 possíveis níveis, então podemos representar, no caso, 256 níveis de cinza, de 0 a 255. Via de regra, 0 e 255 representam o preto e o branco, respectivamente, com tons de cinza entre eles. Como outro exemplo, uma imagem binária, que possui profundidade de apenas 1 *bit*, será quantizada em apenas dois níveis, normalmente preto e branco. Nesse caso, apenas 1 *bit* é suficiente para descrever o valor de um *pixel*, já que 2^1 resulta em 2 níveis possíveis. Assim, para a representação adequada de imagens na forma digital, é preciso que haja um padrão, uma representação da informação dos *pixels* da imagem no computador, que associe a relação entre o valor do *pixel* e a cor que ele representa.

Quando se trata de imagens coloridas, essa relação nem sempre é imediata, e a representação digital de cor pode ser um pouco mais complexa. A quantidade de informação a ser representada, nesse caso, é maior que em sistemas espaços de cor monocromáticos. Existem diversos sistemas que regem a representação do valor do *pixel* em imagens coloridas, e a compreensão desses sistemas é de grande importância. A sensação de cor é a resposta do sistema visual humano a um estímulo causado pela incidência de radiação eletromagnética visível, a luz. Existem diversos padrões para representar a cor na computação, os chamados sistemas de cores, modelos de cores ou espaços de cores. Via de regra, a representação

moderna de cor na computação discretiza a informação de cor em um espaço de 24 bits (3 bytes). Isso nos fornece 2^{24} níveis de cor e, portanto, o sistema permite representar 2^{24} cores diferentes, o que é chamado de profundidade de cor de 16 *megabits*. Entre os sistemas mais comumente utilizados estão:

- RGB: Talvez o mais comum espaço de cores, o sistema RGB baseia-se na fisiologia da visão humana. Determinou-se experimentalmente que o sistema de visão humano apresenta três tipos diferentes de células foto-receptoras, cada uma com sensibilidade diferente a diferentes frequências de luz (Wyszecky e Stiles, 1982). Para cada uma das células, as frequências que incitam a resposta mais intensa representam as cores vermelho, verde e azul. O sistema RGB, portanto, usa as componentes vermelha (*Red*), verde (*Green*) e azul (*Blue*) em diferentes intensidades para compor cada uma das cores que podem ser representadas. Cada componente é representada por um *byte*, e o valor do *byte* determina a intensidade da respectiva componente.

- HSV: O sistema HSV procura dividir o espaço de cores de forma mais intuitiva ao ser humano, e divide o valor do *pixel* nas componentes matiz (ou *hue*), saturação e valor (*value*), normalmente chamado de intensidade (como no sistema análogo HSI). A matiz representa a informação básica da cor. A fraca correlação entre os componentes faz com que o sistema HSV seja uma opção atraente quando se busca independência a variações de contraste para isolar uma cor, ou isolar um nível de iluminação, por exemplo.

A maioria dos sistemas segue o padrão de três componentes de intensidade. Apesar disto, existem sistemas que seguem outros esquemas de cor, como o RGBA, um sistema de 32 bits, sendo que os 24 primeiros bits determinam a cor e os 8 últimos bits são usados para representar efeitos como transparência e translucidez, no chamado canal alfa da imagem. O modelo de cores CMYK, por sua vez, é definido pelas componentes azul ciano, magenta, amarelo e preto, que sendo um modelo subtrativo de cores é usado em muitas impressoras.

O conhecimento da estrutura de imagens digitais, sua representação e padrões utilizados é importante para o tratamento computacional das mesmas. A visão computacional tem grande potencial de aplicações em diversas áreas do conhecimento, de especial interesse para este trabalho na robótica e robótica móvel.

3.2 Visão Computacional Aplicada à Robótica

A visão computacional é uma ciência que muito pode beneficiar a robótica móvel. Uma sequência de quadros ou mesmo uma única imagem são sinais complexos e de grande variedade. Não existe um método ideal para extração de informações de uma imagem; cada aplicação requer uma abordagem específica, e como a visão humana, contextualizada. Entre as pesquisas envolvendo aplicações de métodos e técnicas de visão computacional na robótica, temos trabalhos como (Zhou *et al.*, 2003) que usa análise de histogramas de imagens para localização com ótimos resultados. Os histogramas podem ser usados para pareamento com certa eficiência, como demonstrado em (Ulrich e Nourbakhsh, 2000), que compara perfis de histogramas para identificar cenas e localizar um robô móvel no ambiente. Uma desvantagem de histogramas é que estes não contêm informações sobre posições e relações espaciais entre *pixels* (Shen e Hu, 2006).

Operações deste tipo se enquadram na categoria de operações chamadas de operações espaciais. Operações espaciais referem-se a operações realizadas diretamente nos *pixels* de uma dada imagem, trabalhando no domínio espacial. Estas técnicas ainda se dividem em operações pontuais, realizadas *pixel a pixel* (*point operators*) ou operadores estatísticos (Nixon e Aguado, 2007).

Entre os operadores pontuais podemos citar operações com histogramas de imagens, como limiarização (*thresholding*) ou funções de alteração de níveis de cor. Entre os operadores não-pontuais temos operadores estatísticos, morfológicos, entre outros, e é possível citar transformações geométricas (rotação, escala e translação), interpolação, erosão, dilatação e filtros como o de média, em que o valor do *pixel* é dado pela média aritmética dos valores dos *pixels* vizinhos, ou o filtro de mediana (Davies, 1988), eficaz na substituição de *pixels* de valores muito díspares da imagem, sendo muito usado para remoção de ruído (Jähne, 2005; Couto, 2008).

Analizando-se as derivadas do sinal, ainda mais informações podem ser extraídas de uma imagem. A derivada fornece informações de variação, de contraste. Em uma imagem, ela enfatiza extremos entre regiões, e portanto, operadores que detectam bordas, cantos e imagens em geral baseiam-se em avaliar e interpretar as derivadas. Uma variação contrastante no sinal

implicará em uma derivada alta naquela região de transição, um valor de pico localizado na borda, e para intervalos pequenos o suficiente a primeira derivada pode ser aproximada com pouco erro pela tangente entre pontos adjacentes.

De modo semelhante, a segunda derivada pode ser averiguada, já que esta cruza o eixo zero nos pontos máximos e mínimos da primeira derivada. Assim, os pontos onde a segunda derivada alterna de sinal são os pontos de pico da primeira derivada, que são os pontos de maior variação da imagem original (Nixon e Aguado, 2007).

Operadores baseados nestas análises diferenciais de primeira e segunda ordem permitem detecção de contornos e bordas. Neste nível já se caracteriza a detecção de pontos de interesse em uma imagem, o que pode ser de grande interesse para a visão computacional e robótica móvel. Métodos para detecção de bordas com derivada de primeira ordem incluem Canny (Canny, 1986), Sobel e usando derivadas de segunda ordem temos o Laplaciano. Métodos similares permitem a detecção de curvaturas e de cantos (Moravec, 1980; Chen *et al.*, 1995), como no *Harris Corner Detector* (Harris e Stephens, 1988), ou FAST, *Features from Accelerated Segment Test* (Rosten e Drummond, 2006).

Existem também detectores de características que envolvem o uso conjunto de operadores matemáticos diferenciais e transformações (trabalhando temporariamente com sinais no domínio da frequência, como na análise de Fourier), em métodos como LoG (Laplaciano do Gaussiano) e DoG (Diferença de Gaussianos), entre outros (Figura 3.1).

Outra categoria de detectores de características são os detectores de forma (*shape detectors*), dentre os quais pode-se citar a Transformada Hough. Esta transformada é usada, especificamente, para a extração de formas como retas, círculos e elipses. O método foi desenvolvido em (Hough, 1959), mas apenas em (Duda e Hart, 1972) ele foi aplicado em processamento de imagens pela primeira vez (Bradski e Kaehler, 2008).

Além destes métodos que detectam características genéricas (livres de contexto), aplicações mais específicas podem utilizar informações sobre o problema em particular para desenvolver métodos especializados de detecção através de câmeras, como o sistema de proteção de privacidade usado no carro do Google Street View¹⁴ para detectar rostos e placas de carro e obscurecê-los por questões de privacidade (Frome *et al.*, 2009) ou sistema de

¹⁴ <http://maps.google.com/help/maps/streetview>, acesso em 01/2012.

autenticação biométrica, onde características particulares físicas de uma pessoa (rosto, íris, impressão digital, etc) podem ser usadas para detectar sua identidade (Jain *et al.*, 2000).



Figura 3.1: Exemplos de métodos de extração de características visuais. Do topo, da esquerda para a direita: imagem original; imagem com Canny; imagem com Sobel; Laplaciano do Gaussiano.

Marcas artificiais que sejam de fácil detecção podem ser adicionadas ao ambiente para auxiliar a detecção (Saripalli *et al.*, 2002). Identificadores de Radio-Frequência (RFIDs) podem ser usados como marcas inseridas no ambiente para navegação, como proposto em (Khubitza *et al.*, 1997), e mais recentemente em (Gueaieb e Miah, 2009) e em (Park e Hashimoto, 2009). Marcas artificiais podem ser combinadas com características SIFT. Em (Forssén *et al.*, 2008), a marca artificial funciona como um mecanismo para “atrair a atenção” do robô a um objeto; em seguida, o método SIFT é usado para reconhecimento.

Outros métodos utilizam Redes Neurais Artificiais. Desta forma, o sistema pode promover resultados eficientes para detecção de elementos específicos para os quais ele foi treinado, adicionando assim semântica à escolha das regiões de interesse da imagem.

3.3 *Métodos Detectores-Descritores Baseados em Espaço de Escala*

A capacidade de um método de visão computacional ser capaz de extrair pontos de interesse de uma imagem é bastante vantajosa. Tais pontos, porém, devem ser escolhidos de forma a serem possíveis de rastrear e recuperar futuramente. É interessante que a informação usada para descrever um ponto de referência seja rica, mas ao mesmo tempo é preciso que a característica visual descrita seja recuperada no futuro em outro sinal (imagem) diferente, então deve haver equilíbrio neste aspecto.

Um grupo de detectores de pontos de referência que oferecem bons resultados de detecção e recuperação de pontos de referência são os detectores-descritores de características baseados em espaço de escala. A inspiração para trabalhos do tipo veio do método SIFT, *Scale Invariant Feature Transform*, desenvolvido por Lowe (Lowe *et al.*, 2004), e desde então muitas versões e aprimoramentos foram criados. Uma comparação extensa de diversos métodos computacionais para geração de descritores pode ser encontrada em (Mikolajczyk e Schmid, 2005). Estes métodos são também chamados de *blob detectors*, pela tendência de pontos de referência nestes casos serem comumente detectados em *blobs*, ou seja, áreas delimitadas da imagem distintas por cor e forma.

A seguir, faremos uma apresentação e descrição dos principais métodos detectores-descritores.

3.3.1 SIFT

Um dos trabalhos seminais da área de métodos detectores-descritores de características baseados em espaço de escala, o SIFT foi desenvolvido e publicado por David Lowe (Lowe *et al.*, 2004). Por este motivo, embora este trabalho tenha usado o método SURF, é válido

analisar primeiramente o método sobre o qual ele se fundamentou. Detectores baseados em espaço de escala baseiam-se em dois elementos distintos; primeiramente o detector, que analisa a imagem e seus pontos de referência candidatos, selecionando os mais significativos e estáveis para compor uma assinatura. Dá-se o nome de assinatura ao conjunto de características visuais detectadas que descreve uma determinada cena, ou determinado elemento da cena. O segundo elemento é o descritor, que oferece parâmetros que descrevam cada ponto de forma que possam ser posteriormente comparados e associados a pontos de outras assinaturas. É desejável que o detector possua boa repetibilidade (ou seja, seja capaz de extrair os mesmos pontos de referência mesmo com alterações na imagem), enquanto o descritor deve oferecer descrições distintas para os diferentes pontos de referência.

A importância do SIFT se deve ao fato de o método ter oferecido uma alternativa aplicável em casos gerais, rápida e confiável para obter e comparar pontos importantes de uma imagem. O algoritmo SIFT é aplicado em imagens em escala de cinza, sem restrições de resolução ou conteúdo, e foi introduzido como um processo para extrair da imagem e descrever pontos de interesse de maneira invariante a variações de escala e rotação, e que também possuam a propriedade de serem estáveis a graus substanciais de interferências como distorções, alterações no ponto de vista (mudanças de perspectiva), mudanças de iluminação, oclusão parcial e adição de ruído (Lowe, 2004).

Cada ponto obtido na aplicação do algoritmo deve ser bem definido: cada ponto de interesse, ou *keypoint*, ou característica visual, possui parâmetros de magnitude e orientação próprias determinados pelo descritor. Assim é possível caracterizar a assinatura de uma imagem, definida por um conjunto de pontos de referência. A comparação de assinaturas de diferentes imagens pode ser usada para associar objetos e cenas iguais, mesmo em tempos e contextos diferentes. A taxa de acerto particularmente boa deste tipo de algoritmo fez com que os detectores-descritores baseados em espaço de escala se tornassem populares em diversas aplicações de visão computacional e robótica, por exemplo como ferramenta para localização, SLAM, algoritmos de *Structure from Motion*¹⁵, estimativa de movimento, *tracking* de objetos ou até reconhecimento de rostos (Majumdar e Ward, 2009).

Aplicações modernas também incluem o reconhecimento de sinais de trânsito (Kus *et al.*, 2008) e criação de fotografias compostas ou fotografias panorâmicas através da detecção

¹⁵ http://homepages.inf.ed.ac.uk/s0346435/projects/sfm/sfm_sift.html, acesso em 01/2012.

das áreas correspondentes de sobreposição nas imagens, como no software AutoStitch¹⁶ (Brown e Lowe, 2007). O fato de que o software nem ao menos necessita de qualquer orientação sobre posições relativas das fotos do conjunto de imagens de entrada é uma demonstração da robustez das características visuais SIFT.

O algoritmo SIFT se divide em quatro estágios: a detecção de extremos do espaço de escala, a detecção e localização dos pontos de referência, a atribuição de orientações e o descriptor dos pontos.

O primeiro passo do algoritmo SIFT, a detecção do espaço de escala, consiste em localizar os pontos da cena que são invariantes à mudança de escala. A detecção destes pontos pode ser realizada através da busca por características estáveis através de todas as possíveis escalas, utilizando uma função de escala chamada espaço de escala, a função Gaussiana (Witkin, 1983). A repetida convolução da imagem original em *grayscale* com a função Gaussiana gera um conjunto de Gaussianas da imagem. A subsequente subtração dos resultados das Gaussianas uns dos outros (Figura 3.2) resulta na chamada diferença-do-Gaussiano, ou DoG¹⁷. Em seguida, as imagens Gaussianas são quantizadas pela metade (*downsampling*) e o processo se repete por algumas iterações. Este procedimento garante invariância à escala.

Comparações experimentais (Mikolajczyk e Schmid, 2005) mostraram que os extremos locais da diferença-do-Gaussiano (máximos e mínimos locais) são pontos estáveis, e bons candidatos a serem características visuais da assinatura SIFT da cena. Neste primeiro estágio, a maior demanda computacional se deve à construção em si do espaço de escala Gaussiano, e não à detecção dos extremos do mesmo (Zhang *et al.*, 2008).

O segundo estágio do algoritmo é a detecção e localização dos pontos de referência, eleitos a partir dos pontos candidatos obtidos na fase anterior. Isto ocorre porque deseja-se rejeitar pontos pouco estáveis, como pontos de baixo contraste ou pontos localizados em bordas. Assim, um *threshold* de contraste é aplicado aos pontos. Pontos definidos em bordas são suscetíveis a ruído, e são detectados na diferença-do-Gaussiano por terem curvatura principal alta na direção da borda, mas não na direção perpendicular.

¹⁶ <http://www.autostitch.net>, acesso em 01/2012.

¹⁷ <http://fourier.eng.hmc.edu/e161/lectures/gradient/node11.html>, acesso em 01/2012.

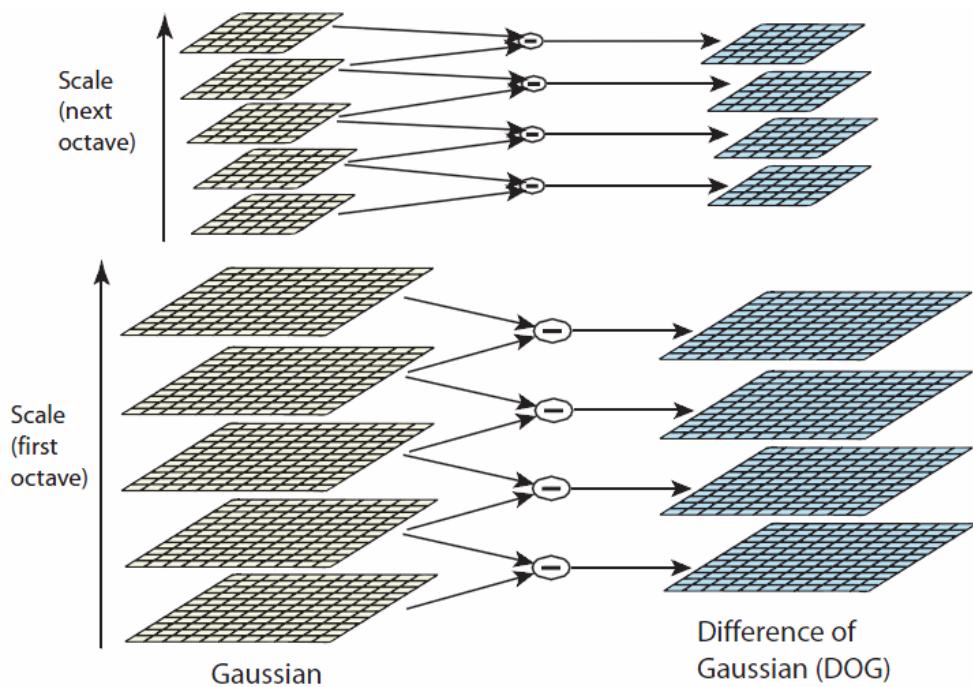


Figura 3.2: O primeiro passo do algoritmo SIFT. Primeiro calcula-se o Gaussiano das imagens por convolução, e através da subtração dos resultados adjacentes obtém-se as diferenças-do-Gaussiano, onde serão buscadas as características invariantes. A escala da imagem é reduzida e o processo é realizado novamente (Lowe, 2004).

O estágio que segue é a atribuição das orientações aos *keypoints* escolhidos. A representação das orientações torna os pontos invariantes a alterações de rotação. A orientação é obtida a partir da análise do gradiente em pontos da região ao redor do *keypoint*, a fim de determinar a orientação dominante. Finalmente, no quarto e último estágio é criado um descriptor local da característica. Neste passo o SIFT gera uma descrição altamente distinta para a região da imagem, que ainda seja estável a outras deformidades geométricas e fotométricas na imagem. Isto é realizado através do cálculo do histograma de gradientes de magnitudes e orientações ao redor do ponto detectado. Uma explicação para os bons resultados obtidos com o uso dos descritores baseados em histogramas, comuns em outros métodos além do SIFT, está na propriedade dos histogramas de serem uma representação da distribuição de dados independente de informação espacial, o que simplifica computacionalmente as comparações e promove estabilidade (Mikolajczyk e Schmid, 2005). Este processo é biologicamente inspirado em uma teoria sobre o funcionamento de grupo de neurônios complexos do córtex visual primário que permitem que seres humanos reconheçam

objetos tridimensionais a partir de perspectivas diferentes (Serre *et al.*, 2007). O conjunto dos pontos detectados e descritos forma a assinatura da imagem (Figura 3.3).

A assinatura ou descriptor SIFT de qualquer cena pode ser obtido de forma repetível. O pareamento é feito usando uma técnica simples de cálculo da distância euclidiana do descriptor do ponto encontrado em relação ao da base de dados. A partir dos pares entre diversos pares de pontos, a assinatura é identificada.

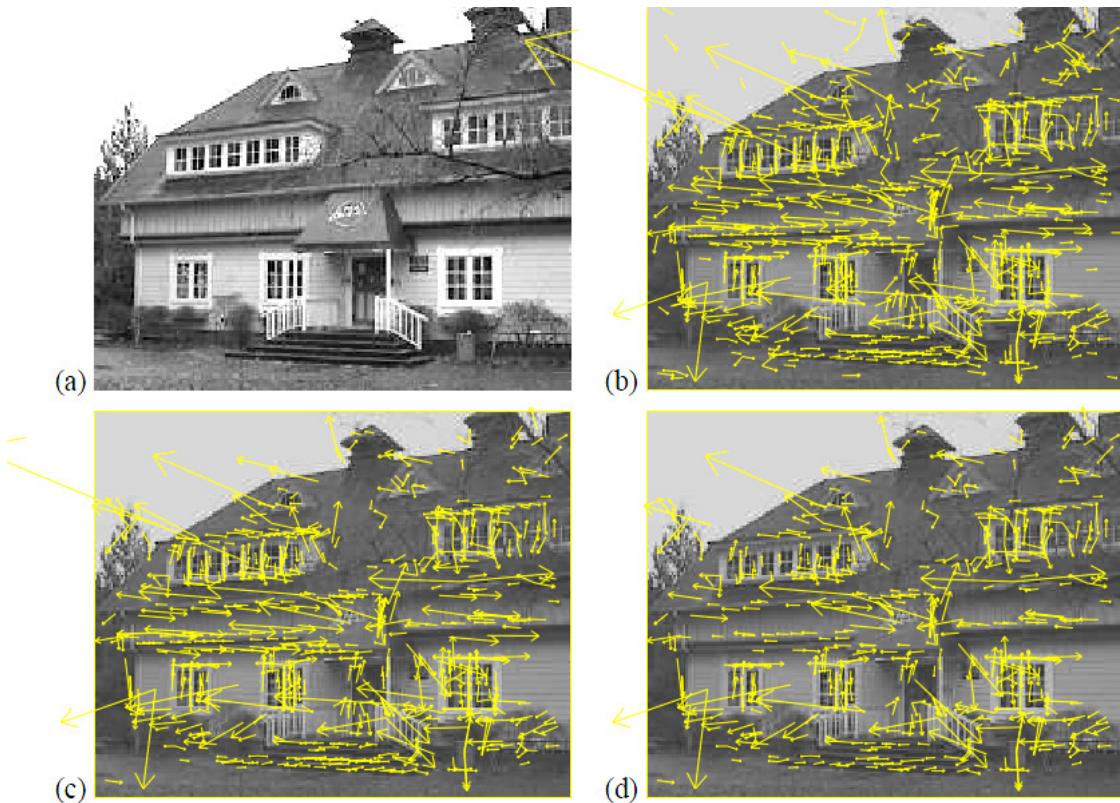


Figura 3.3: Exemplo de *keypoints* obtidos com o algoritmo SIFT. Em (a) temos a imagem original, em (b) o primeiro conjunto de *keypoints* obtidos, em (c) o conjunto reduzido após aplicação de um *threshold* de contraste mínimo e (d) o conjunto final após um *threshold* de razão máxima de principais curvaturas (Lowe, 2004).

No caso de um robô móvel, essa informação pode ser usada para localização, já que uma assinatura pode ser associada a uma localização ou orientação. Pode-se também adicionar contexto às assinaturas, o que capacita o robô a reconhecer alvos, objetivos e itens importantes do ambiente. Outro método de uso das *Scale-Invariant Features* na robótica móvel foi descrito por Lowe *et al.* (2001). Neste método, que usa uma câmera estereoscópica, as características SIFT de uma imagem são computadas para cada uma das lentes da câmera.

Os resultados podem então ser comparados para estimativa de distâncias e construção de uma mapa 3D do ambiente, pois *keypoints* SIFT mais próximos da câmera possuirão equivalentes com maior disparidade em cada uma das lentes.

3.3.2 SURF

Uma alternativa baseada no método SIFT, que foi o método usado neste trabalho, é o SURF, *Speeded-Up Robust Features* (Bay *et al.*, 2008). O método SURF possui desempenho que aproxima ou até mesmo supera métodos similares como o SIFT, e seus pontos de referência podem ser computados e comparados mais rapidamente, mesmo em alguns casos sacrificando robustez a algumas variações. Comparações entre SIFT e SURF indicam que o algoritmo SURF pode ser mais rápido que o SIFT, e pode também ser mais resistente a alterações de iluminação e *blurring*, ou imagem tremida (Huynh *et al.*, 2009). Isso faz com que o SURF seja uma alternativa viável para determinadas aplicações (mais detalhes sobre a escolha do método no Capítulo 4). Existe uma implementação *open source* do método, o OpenSURF¹⁸, e o método também está incluso na biblioteca de visão computacional OpenCV (Bradski e Kaehler, 2008).

O método SURF aplica diversos aperfeiçoamentos desenvolvidos ao longo de pesquisas na área aos métodos do SIFT. Por exemplo, o SURF usa imagens integrais para simplificar o cálculo das convoluções (Viola e Jones, 2001). Em contraste com o SIFT, o SURF usa o Laplaciano do Gaussiano (LoG) no lugar da diferença-de-Gaussianos (DoG), para construir o espaço de escalas. O uso de imagens integrais permite que, durante o processo de geração de Gaussianas em diversas escalas, o filtro seja aumentado (*upsampling*), no lugar de se reduzir a escala da imagem como no SIFT (*downscaling*). Isso permite ganhos de velocidade no processo (como mencionado, a cálculo do espaço de escala é custoso no SIFT), e evita o *aliasing* causado pela redução da imagem. O determinante da matriz Hessiana (matriz composta pelas segundas derivadas parciais da função imagem) pode então ser usado para classificar os máximos e mínimos da função pelo teste da derivada de segunda ordem (Evans, 2009).

¹⁸ <http://code.google.com/p/opensurf/>, acesso em 01/2009

Em relação ao descriptor do SURF, ele opera de forma análoga ao SIFT, com algumas diferenças. Na atribuição de orientações e cálculo do descriptor, o SURF utiliza a informação de gradiente pela soma das respostas da vizinhança do ponto à *wavelet* Haar, o que de acordo com os autores aumenta a velocidade e robustez do processo. Da mesma forma que o SIFT, a informação utilizada é um histograma de distribuições de gradiente da região do ponto de referência, mas no SURF a resposta à *wavelet* Haar é somada e independe das orientações individuais dos gradientes da região. Desta forma, o algoritmo é mais resistente a ruído, enquanto perde pouca capacidade discriminativa (Figura 3.4).

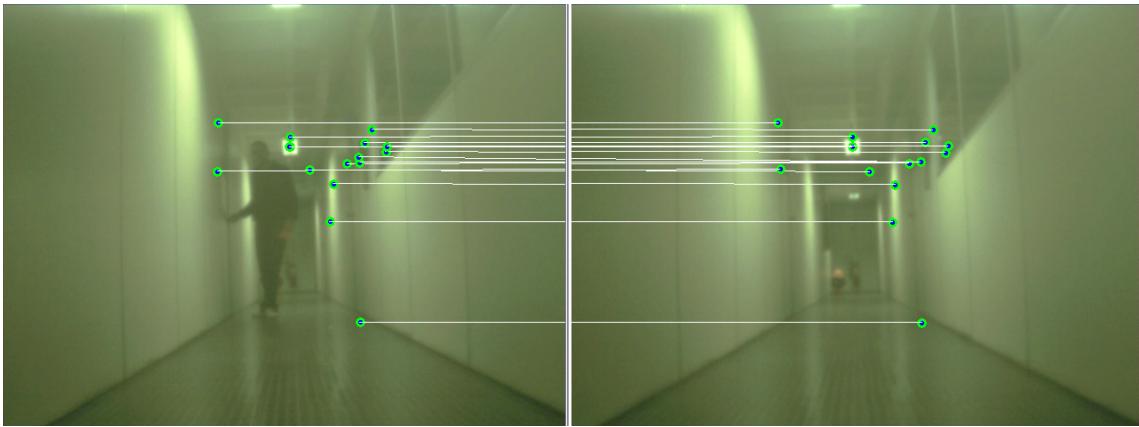


Figura 3.4: O método SURF é resistente a alterações na imagem. Neste caso, a presença de uma pessoa na cena não prejudica pares de pontos em outras regiões da imagem.

Uma outra otimização digna de menção acrescentada pelo método é que o sinal do operador Laplaciano é considerado durante o pareamento de pontos, e duas características só são comparadas se o sinal de seus Laplacianos (calculado na criação do espaço de escala, durante a determinação dos pontos de interesse) for o mesmo. Isto reduz o número de comparações realizadas e aumenta a capacidade discriminativa.

3.3.3 Outros métodos similares

Recentemente a pesquisa em métodos detectores-descritores tem gerado alternativas aos métodos SIFT e SURF. A maioria destas ferramentas mantém a estrutura original do método SIFT, acrescentando inovações e alternativas que em geral possuem vantagens e

desvantagens. Assim, é importante conhecer os métodos e suas características para escolher o mais adequado para a aplicação desejada.

Uma variante direta do algoritmo SIFT é o algoritmo Color SIFT, ou CSIFT (Abdel-Hakim e Farag, 2006; Burghouts e Geusebroek, 2009). No lugar de imagens monocromáticas, este método é adaptado para diferenciar cores. A discriminação de cores do CSIFT pode ser aplicada em situações específicas em que a informação de cor é relevante para aumentar o poder discriminativo do sistema (Van de Sande *et al.*, 2010) e o tempo de processamento não seja um requisito crítico. No CSIFT, o método SIFT é aplicado para cada um dos 3 canais da imagem, e 3 grupos de características são gerados. Os canais componentes da imagem usados no caso são: intensidade, azul-amarelo e verde-vermelho. Os ganhos não são significativos e, como esperado, o tempo de execução do algoritmo CSIFT é significativamente maior que o do SIFT comum.

O PCA-SIFT, por sua vez, é uma modificação do SIFT que aperfeiçoa o descritor aplicando PCA (*Principal Component Analysis*) nos histogramas de gradiente (Ke1 e Sukthankar, 2004). Como na modificação aplicada pelo SURF, o novo detector é mais distinto e resistente a deformações. O PCA (Jolliffe, 2002) é uma técnica de redução de dimensionalidade através da projeção de amostras de dimensões maiores em espaços de dimensões menores.

Um dos métodos mais recentes baseados em detecção de características visuais em espaço de escala é o CenSurE (Agrawal *et al.*, 2008). Como o SURF ele utiliza imagens integrais para ganho de desempenho

Baseado no CenSurE, surgiu o SUSurE, *Speeded Up Surround Extremas* (Ebrahimi e Mayol-Cuevas, 2009). Voltado para execução em dispositivos portáteis. Para reduzir a carga de transmissão, o SUSurE se propõe em compactar a quantidade de informação utilizada. Isto ocorre através da realização de amostragem esparsa na detecção e descrição.

4 Visão Computacional Aplicada à Robótica: Ferramentas e Métodos

Este capítulo apresenta as ferramentas e recursos utilizados no desenvolvimento desta dissertação. Os testes foram realizados e todos os materiais foram disponibilizados junto ao LRM, o Laboratório de Robótica Móvel do ICMC-USP. Para o desenvolvimento do projeto proposto foram usados os robôs móveis, computadores e periféricos do LRM, bem como as ferramentas de controle e simulação robótica Player/Stage, e o *software* de processamento de imagens e visão computacional OpenCV, que são descritos a seguir nas seções 4.1 a 4.3. A seção 4.4 descreve a proposta de pesquisa e os experimentos que foram desenvolvidos.

4.1 Plataformas Robóticas e Periféricos

Os testes práticos do projeto foram realizados principalmente sobre a plataforma robótica móvel para ambientes *indoor* Pioneer 3-AT. Alguns testes durante o projeto foram realizados sobre os robôs móveis Pioneer 3-DX e Erratic ERA-MOBI, plataforma similar ao P3-DX (Figura 4.1). Os robôs Pioneer estão entre as plataformas mais utilizadas na atualidade e pesquisas em robótica, e apesar da popularidade do Pioneer 3-DX, durante os testes e experimentos o Pioneer 3-AT mostrou-se bastante estável, havendo sido usado em todos os experimentos finais cujos resultados são aqui apresentados. Estes robôs podem ser equipados com diferentes sensores e atuadores, como sonares, laser Sick LMS e Hokuyo URG, bem como suportes móveis do tipo *pan-tilt* sobre os quais podem ser adaptadas câmeras de vídeo (monocular simples ou de visão estéreo). No caso deste trabalho, a câmera de vídeo foi acoplada sobre o robô e permaneceu fixa na direção frontal. O uso do *pan-tilt* pode ser útil em trabalhos futuros para a proposta de localização, mas este aspecto não foi explorado neste trabalho, que se concentrou nas possibilidades de localização com a câmera fixa. Diversas câmeras foram testadas durante o projeto, sendo que a câmera usada nos testes e resultados apresentados neste trabalho foi uma câmera de barramento FireWire da marca Videre, modelo Videre DCSG Color.



Figura 4.1: Robôs Móveis do Laboratório de Robótica Móvel na USP (LRM – ICMC – USP). Da esquerda para a direita, o Pioneer 3-AT, o Pioneer 3-DX e os dois robôs Erratic.

O Pioneer 3-AT (ou P3-AT ou simplesmente Pioneer) é um robô móvel desenvolvido pela empresa MobileRobots Inc. Versátil e robusto, é uma das plataformas mais reconhecidas e difundidas na academia e em pesquisas na atualidade. Existe a opção de embarcar um PC no robô ou acoplar um computador *laptop* a este, o que o torna adequado para diversas tarefas que demandam maior poder de processamento. O PC oferece grande liberdade para o desenvolvimento de projetos com o Pioneer, tendo sido a opção escolhida para o projeto (Figura 4.2).

A comunicação entre o PC embarcado e os sensores e atuadores se dá através da interface da ferramenta Player. Todos os robôs da família Pioneer são suportados pela ferramenta Player/Stage¹⁹, com *drivers* bem estabelecidos e suportados para comunicação entre o robô e o PC no Player, e com modelos para simulação dos robôs Pioneer no caso do Stage.

A base do Pioneer 3-AT, sem periféricos, apresenta medidas de 50,8 cm x 49,7 cm x 27,7 cm, o que o tornam adequado para navegação em ambientes internos (*indoor*), como salas e corredores. O Modelo AT também foi desenvolvido com o propósito de suportar a navegação por ambientes mais acidentados. O robô pode se mover em velocidades de até 0,8 metros por segundo, pode suportar uma carga de até 12 kg, e possui autonomia de até 4 horas de operação com suas 3 baterias de 7,2 Ah cada, o que foi suficiente para os testes realizados.

¹⁹ <http://www.mobilerobots.com/ResearchRobots/ResearchRobots.aspx>, acesso em 01/2012.

Com baterias reserva a carga pode ser suficiente para a maioria das aplicações de localização e navegação já que as baterias são *hot-swappable*, ou seja, as baterias podem ser trocadas sem que haja necessidade de desligar o robô.

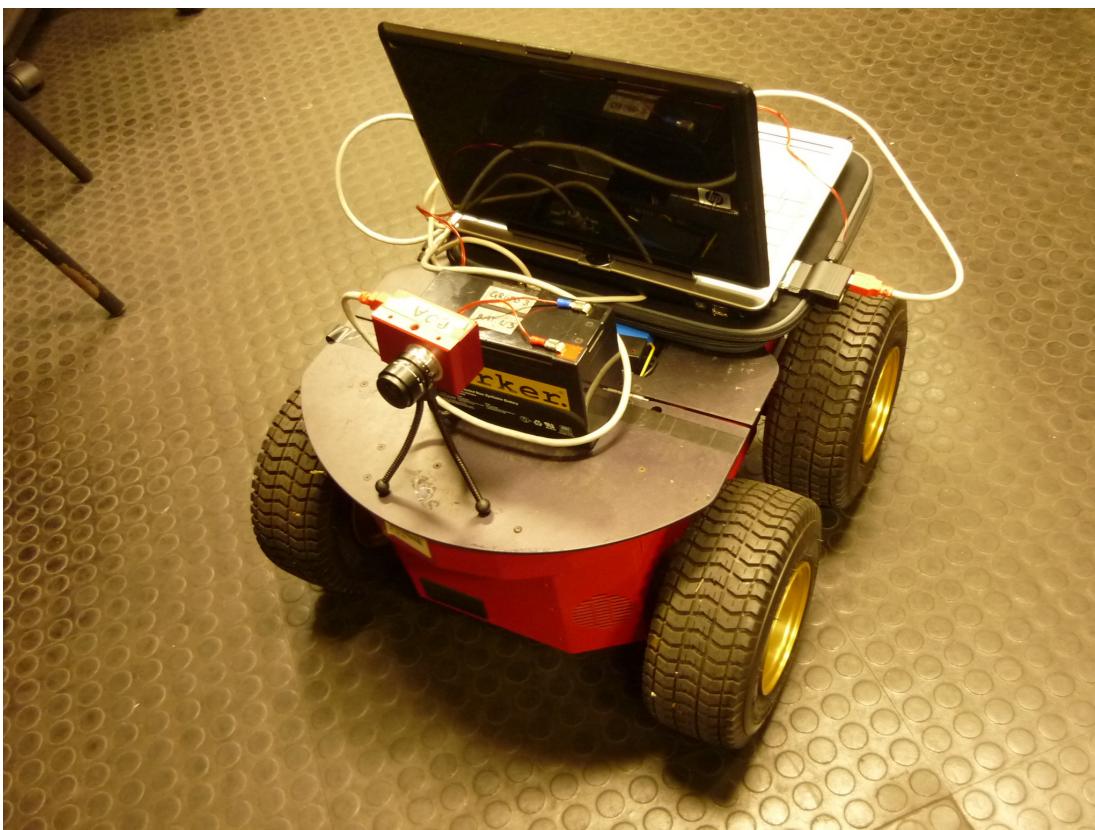


Figura 4.2: Robô Pioneer 3-AT equipado para uso, com a câmera e *notebook*.

4.2 OpenCV

O processamento de imagens foi implementado em linguagem C, usando como base a biblioteca de Visão Computacional OpenCV²⁰ (Bradski e Kaehler, 2008). A OpenCV é uma biblioteca abrangente e estabelecida, comumente utilizada em aplicações de Interação Homem-Máquina (HCI), reconhecimento de características visuais, rostos e gestos, seguimento (ou *tracking*) de movimento e robótica móvel. Como tal, a biblioteca oferece implementações de diversas transformações, técnicas e operações sobre imagens, suporte de

²⁰ <http://opencv.willowgarage.com/wiki/>, acesso em 01/2012.

leitura e escrita para os principais formatos de vídeo e imagem, estruturas de dados auxiliares específicas para trabalho com imagens e funções de aprendizado de máquina, como suporte a redes neurais. A documentação existente é extensa e abrangente, e a OpenCV pode ser usada em ambientes Linux ou Windows, sendo compatível com a plataforma PC embarcada nos robôs móveis Pioneer e Erratic.

A OpenCV apresenta-se como uma importante ferramenta para o desenvolvimento de algoritmos e a realização de experimentos e testes em visão computacional. Suas aplicações na área de visão computacional aplicada à robótica móvel têm sido extensas. Além disto, é uma ferramenta que se integra ao sistema embarcado de controle e navegação dos robôs móveis, tendo um desempenho bastante adequado para uma execução de algoritmos de controle e navegação em tempo de execução.

Nos estágios preliminares do trabalho, a OpenCV foi extensivamente usada para testes dos diversos detectores de características, incluindo detectores de bordas e cantos, transformada Hough e variantes dos métodos SIFT e SURF (Laganière, 2011).

4.3 A Ferramenta Player/Stage

Uma das alternativas de projeto do sistema de um robô é o caso em que um robô opera como um sistema embarcado, onde o *software* projetado é específico para o *hardware* daquele robô. Por outro lado, quando se usa um computador de propósito geral como um PC para se controlar o robô, com software de alto-nível sendo executado, é preciso que haja uma camada intermediária de *software* que permita a interação entre o *software* de controle e os dispositivos do robô. Existem diversas alternativas para isso, e como já mencionado, neste projeto a comunicação entre o robô e o PC se deu através da ferramenta Player/Stage²¹, uma das ferramentas mais difundidas para controle robótico, interface e simulação. As ferramentas do Project Player, que inclui o Player e o Stage, são compatíveis com sistemas POSIX e são *software* livre.

O Player (Collett *et al.*, 2005) funciona baseado em um modelo cliente-servidor de rede, sendo que nele a comunicação ocorre através do protocolo TCP/IP. No caso, o robô atua

²¹ <http://playerstage.sourceforge.net/>, acesso em 01/2012.

como o servidor, enquanto um programa cliente no PC pode realizar requisições, enviar comandos e acessar os dispositivos do robô, por intermédio dos *drivers* fornecidos pelo Player. Alternativas mais recentes incluem sistemas como o ROS, Robot Operating System (Quigley *et al.*, 2009).

A ferramenta Stage (Gerkey *et al.*, 2003), por sua vez, é um simulador para robôs móveis. A simulação é uma alternativa atraente na robótica móvel devido muitas vezes às dificuldades de se realizar testes em campo. O Stage pode, por exemplo, simular dezenas de robôs simultaneamente no ambiente, ou mesmo acelerar a execução de um teste de muitas horas de duração que poderia ser inviável na prática (Figura 4.3). O ambiente controlado também oferece a possibilidade de testes e experimentos livres de perturbações do ambiente, e pode ser útil em testes para identificar a origem de problemas, através da comparação do desempenho do sistema no ambiente real com o simulado. A desvantagem dessa abordagem é que os modelos em simulação não representam todas as variáveis do ambiente real, o que acarreta muitas vezes em resultados contrastantes entre simulação e testes em robôs reais.

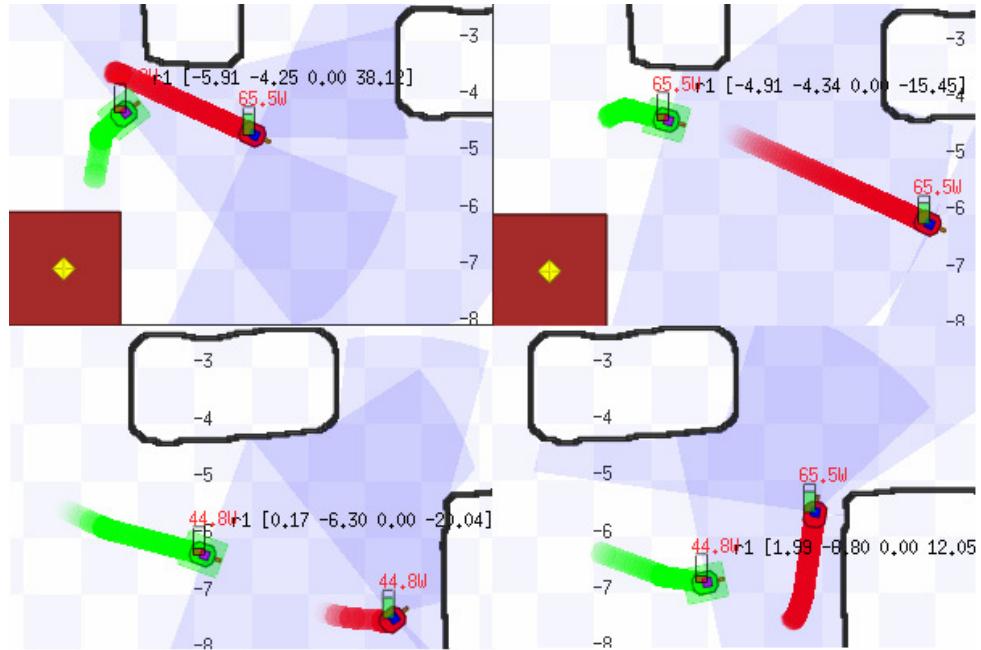


Figura 4.3: Uma captura de tela do ambiente de simulação Stage.

Considerando as disparidades e vantagens da simulação em relação aos testes de campo, a ferramenta Stage será usada especialmente para avaliação do algoritmo de controle e

navegação autônoma do robô móvel. As atuais versões (Stage 3.2.x) possuem opção para visualização da simulação de forma 2D, através de vista superior (*bird's-eye view*), ou através de visão 3D. O ponto de vista 3D pode ser livre, ou pode-se optar por visualizar a informação 3D simulada da câmera embarcada de determinado robô da simulação. A visão 3D do Stage é também chamada de 2.5D²², já que é apenas um modo de visualização dos dados, e a simulação propriamente não envolve a terceira dimensão, diferente do que ocorre em outros simuladores para robótica como o Gazebo²³ ou Microsoft Robotics Studio²⁴, que incorporam simulação de física tridimensional.

O simulador Stage é carregado com um arquivo *world*, que representa o mapa do ambiente (extensão .world), incluindo descritores para os robôs que serão usados, que explicitam características de hardware, bem como os módulos que serão usados para controle dos mesmos. Além disso, o arquivo *world* traz representações de elementos de interesse do mapa que denotam as possíveis interações entre esses elementos e os robôs. Todas essas informações são descritas em uma sintaxe própria do Stage.

O simulador Stage foi utilizado em experimentos preliminares para testes da interação do Player com o robô simulado, antes que os resultados fossem aplicados ao robô real.

4.4 Detalhamento da Proposta e Métodos Utilizados

O presente trabalho demonstra o projeto de um sistema de localização global e navegação autônoma utilizando como sensor apenas uma câmera. A pesquisa mostra como um método de extração de assinaturas de pontos de interesse de imagens, associado à memória visual formada pelo armazenamento de imagens de um caminho percorrido e à informação odométrica, pode ser usado para construir no robô um mapa de memória do ambiente. Este servirá de referência para que o robô recupere sua posição em uma situação futura, através da eleição da imagem mais adequada pela associação entre as assinaturas observadas e as cenas armazenadas em sua memória.

²² <http://playerstage.sourceforge.net/index.php?src=stage>, acesso em 01/2012.

²³ <http://gazebosim.org/>, acesso em 01/2012.

²⁴ <http://www.microsoft.com/Robotics/> acesso em 01/2012.

A proposta consiste da capacitação do robô para se localizar dentro de uma rota. A rota é descrita de forma semelhante à VSRR (*View Sequenced Route Representation*), proposta por Matsumoto *et al.* (1996). No VSRR, um percurso de gravação é realizado pelo robô enquanto este é guiado por um ser humano (Figura 4.4). Técnicas baseadas na recuperação de imagens armazenadas em uma etapa prévia de gravação são chamadas de *appearance-based techniques* (Jafar *et al.*, 2009). Durante este percurso guiado, o robô realiza a captura de imagens do ambiente, onde as imagens capturadas estão associadas à informação odométrica (posição e orientação do robô). Quando a localização autônoma for realizada, o robô pode associar a imagem atual observada com as cenas armazenadas no percurso de gravação, e assim pode aproximar sua pose atual relativamente às poses gravadas.

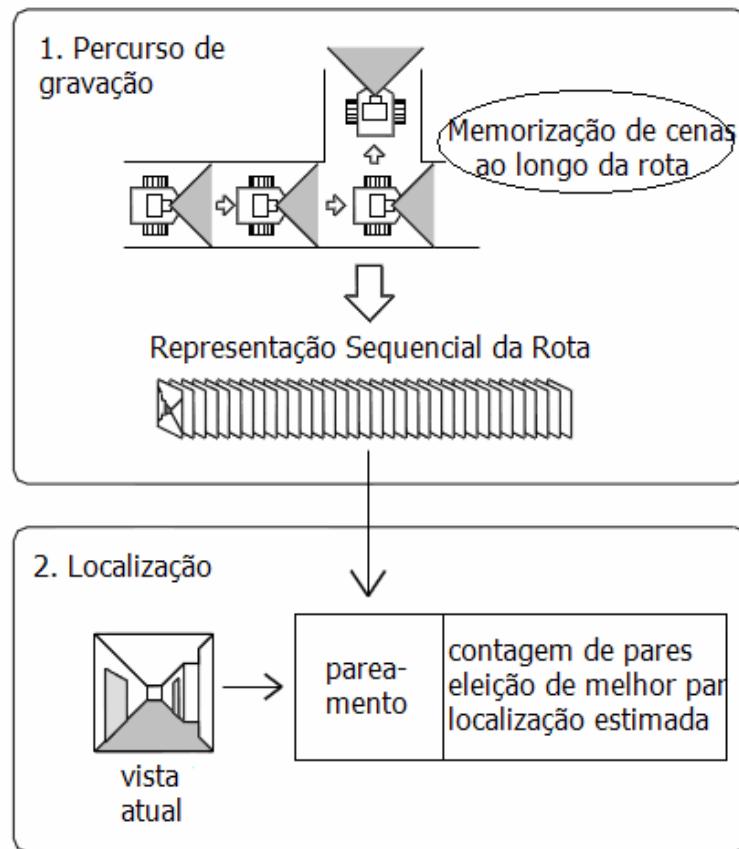


Figura 4.4: A VSRR, composta de um percurso de gravação, posteriormente usado na localização do robô (adaptado de Matsumoto *et al.*, 1996).

No trabalho original da VSRR, o pareamento entre a imagem observada e as imagens da memória se deu através de uma técnica de correlação (Crowley e Chenavier, 1992). Nesta

proposta, utilizamos o método SURF de extração de pontos de referência para gerar as assinaturas de cada imagem, e o descritor do SURF para a comparação entre diferentes imagens. Após a comparação e pareamento, a informação odométrica nos permitiu medir o erro de odometria entre as imagens pareadas. Note que a medida odométrica das imagens memorizadas carrega em si um erro odométrico cumulativo. Para aplicações práticas, o importante neste caso é que a informação visual esteja correta, pois ela pode ser usada durante a localização e navegação pra mitigar o erro odométrico, realizando refinamentos no ângulo de navegação do robô e distância percorrida, para que ele possa se manter próximo da rota desejada. Com isso, não é necessário considerar a odometria de uma cena em relação à origem (erro acumulado), mas a localização relativa às imagens pode manter o erro dentro de uma margem limitada.

4.4.1 A Construção do Mapa

O mapa do ambiente armazenado é a referência através da qual o robô se localiza e navega. O robô móvel foi equipado com um *notebook*, ao qual uma câmera foi ligada. O percurso foi conduzido por um ser humano, guiando o robô com um *joystick*. Durante todo este percurso manual, o robô realizou a captura de imagens do ambiente, na freqüência de um *frame* por segundo de execução. Esta frequência resultou em uma densidade de imagens considerada suficiente para oferecer a precisão de localização necessária para uma grande gama de aplicações de navegação de espaços internos. Com base nos dados das imagens obtidas no percurso guiado, calculamos que a distância entre *frames* adjacentes no mapa de memória foi de no máximo 10 cm, para os diferentes percursos capturados. Este valor representa a resolução da localização do robô. Queremos neste caso testar a precisão do algoritmo de pareamento para diferenças pequenas de perspectiva, portanto uma alta definição é importante, comparativamente, por exemplo, à definição usada no trabalho de Matsumoto (um metro entre uma imagem e outra). Simultaneamente à captura das imagens, a informação odométrica foi gravada em um registro de *log*; especificamente os parâmetros de posição (*x,y*) e a rotação (*z*), parâmetros suficientes para posicionamento de um robô móvel em terreno considerado plano (uma suposição razoável para os ambientes internos onde ocorrerá a

navegação). Os parâmetros são considerados relativamente à coordenada onde o robô iniciou sua execução, que convencionamos como sendo $(x, y, z) = (0, 0, 0)$. Cada imagem capturada no percurso de gravação foi associada à informação odométrica da posição daquele local, de forma que um mapa de imagens do ambiente foi obtido. As assinaturas SURF de cada imagem foram calculadas e armazenadas nesta etapa, antecipando esta tarefa para otimizar o desempenho durante a execução.

4.4.2 A Localização Autônoma

A localização do robô no ambiente é essencial para a maioria das tarefas de navegação robótica, e um dos elementos mais importantes que devem ser abordados na robótica móvel (Thrun *et al.*, 2005). Existem sistemas robóticos de navegação reativos ou simplificados que podem ser funcionais sem que haja necessidade de que o robô saiba sua localização precisa, mas a capacidade de se localizar representa um conjunto maior de atividades possíveis que um robô pode realizar. De fato, como mencionado, a localização é importante para qualquer tarefa de navegação mais complexa, que exija planejamento de trajetória ou qualquer tipo de conhecimento da posição do robô em relação ao ambiente.

O sistema proposto no projeto não exige o conhecimento da posição inicial do robô, então ambos os tipos de localização, global e local, são necessários. É um problema semelhante ao do robô sequestrado (Howie *et al.*, 2005), já que a única verdade assumida é que o robô está em algum ponto de sua rota memorizada, mas sem conhecimento do ponto exato. A câmera de vídeo é um sensor que traz dados ricos sobre o ambiente, mas são informações dependentes de interpretação. Métodos como o SURF oferecem reconhecimento de imagens, necessário para a aplicação desejada.

A localização global requer que exista um mapa do ambiente previamente conhecido, relativo ao qual a localização se dará. No caso do presente trabalho, este mapa é a sequência de imagens obtida no trajeto inicial do robô (memória visual do caminho), portanto a localização se dará por um modelo não-métrico (sem informação explícita de distância). Posteriormente a odometria será considerada para medir a acurácia do método, mas esta informação não é usada na localização. Uma vez posto em funcionamento, o robô não tem

conhecimento de sua posição em todo o ambiente, portanto a localização deverá ser global. A única informação sobre seu estado atual no percurso que o robô possui ao ser ligado é a imagem imediatamente à sua frente capturada pela câmera. Como assumimos que o robô foi posicionado em algum lugar do caminho memorizado, a localização consiste em percorrer a base de dados de imagens em busca do quadro mais condizente com a captura atual da câmera. Esta busca completa do mapa de imagens só deverá ser realizada no momento da inicialização ou caso o robô perca sua posição durante a localização local incremental.

A comparação dos quadros é realizada através do método SURF. O SURF é um método especializado em identificação de cenas similares, a partir de perspectivas distintas (entre outros parâmetros). O SURF, no entanto, não é um método binário. O algoritmo de identificação obtém, para cada par de cenas, uma certa quantidade de pares de pontos de referência identificados. Cenas iguais em perspectivas muito diferentes ainda são pareadas com sucesso, mas cenas iguais em perspectivas apenas levemente diferentes oferecem um número maior de pares de pontos associados, como é intuitivamente esperado. A quantidade destes pares pode ser usada como critério para avaliação da similaridade entre as cenas. Será mostrado que em ambientes internos como, por exemplo, um corredor, o robô observa frequentemente a mesma cena com leves variações de perspectiva entre cada quadro (conforme o robô se move). A aplicação do SURF em um caso como este fornece medidas gradativas do nível de similaridade entre cenas diferentes.

Além do número de pares, que é o critério padrão de métodos detectores-descritores para determinar similaridade entre imagens, propomos também um método alternativo para a pareamento. Em casos próximos, o método da contagem do número de pares pode não ser tão preciso, já que a diferença entre as perspectivas não é necessariamente suficiente pra causar redução significativa no número de características obtidas. Testes também mostraram que diferenças de iluminação, *motion blur* e foco da câmera podem causar variações no número de pares obtidos, o que pode ser suficiente para que a imagem mais adequada seja ultrapassada em número de pareamentos por alguma imagem próxima. Para contornar este problema, é possível utilizar um critério diferente para a decisão da melhor candidata entre as imagens escolhida, que não seja aquela com o maior número de pareamentos. O critério proposto é a variação em *pixels* entre as características visuais nas diferentes imagens. Com a câmera fixa em uma posição, testes mostraram que características visuais pareadas que são próximas em

número de *pixels* no eixo Y apresentam maior chance de terem sido obtidas a partir da mesma distância. Neste caso e no restante do trabalho nos referiremos às coordenadas (X,Y) da imagem em *pixels*, sendo X e Y respectivamente os eixos abscissa e ordenada. Isto equivale de certa forma a parear as assinaturas que possuem escalas similares, e é útil em ambientes como os corredores onde há pouca oclusão ou efeitos de paralaxe (*parallax*). Para a localização em ambientes como esses, portanto, pode-se tratar apenas a dimensão em que ocorre o deslocamento do robô (ao longo do corredor ou da rota). Esta é a coordenada mais significativa, pois espera-se que variações de rotação e translação lateral na posição do robô sejam pequenas (devido à pouca largura do corredor e conformidade do robô à rota). Estas duas coordenadas se confundem e podem ser tratadas e corrigidas indissociadamente em uma possível aplicação de navegação (Basri *et al.*, 1999). Nos testes, portanto, comparamos o critério do número de pareamentos com o critério da distância em *pixels* entre as características pareadas, utilizando como referência para comparação apenas a dimensão mais significativa, de deslocamento ao longo da rota no sentido do movimento do robô.

4.4.3 Escolha e Aplicação do Método SURF

O método escolhido para detecção de pontos de referência com características distintas foi o SURF (Bay *et al.*, 2008). Sendo um método detector-descritor, ele gera boas assinaturas de pontos de referência (uma assinatura é um conjunto de características destacadas da imagem), relativamente resistentes a mudanças (variações de perspectiva, rotação, escala, luminosidade, oclusão, ruído), o que é de grande valia em uma aplicação real em um ambiente pouco controlado. Os métodos detectores-descritores são comumente usados para identificação de objetos e cenas iguais em contextos diferentes. Técnicas de mapeamento usando tais métodos existem, mas comumente utilizam câmeras estéreo para a tarefa. Técnicas que utilizam câmeras monoculares normalmente necessitam de algoritmos custosos para realizar a estimativa de profundidade. Este projeto mostra que havendo um mapa associado às assinaturas, é possível realizar localização com métodos detectores-descritores usando câmeras monoculares; assim elimina-se a necessidade de estimativa de profundidade e movimento e reduz-se a carga de processamento durante a execução. A pesquisa também

avalia se o método detector-descritor SURF é robusto o suficiente para a aplicação em ambientes internos com poucas características e na análise de quadros parecidos.

A escolha do método não foi arbitrária, optando-se pela alternativa mais adequada às prioridades do projeto. Métodos alternativos de detecção de pontos de interesse como o FAST (Rosten e Drummond, 2006) e o Harris (Harris e Stephens, 1988), ambos baseados em detecção de cantos (conhecidos como *corner detectors*) são pouco estáveis a perturbações (especialmente a variações de escala, um ponto importante no presente projeto), e foram descartados devido à dependência do projeto de detecção de assinaturas em diferentes escalas. O detector deve possuir boa repetibilidade, enquanto o descritor deve oferecer descrições distintas para os pontos de referência.

O método SURF para detecção de características distintas apresenta bom desempenho quando comparado com outros métodos de detecção e descrição de características que usam espaço de escala. Um aspecto onde o SURF com frequência supera outros métodos similares como o SIFT é na velocidade de processamento, em termos, por exemplo, de pareamentos por segundo na etapa de comparação de assinaturas (Bauer *et al.*, 2007), e o algoritmo utiliza menor quantidade de memória RAM, sendo o SURF a escolha de preferência para diversas aplicações embarcadas em celulares onde a economia de recursos de *hardware* é essencial, como visto em (Takacs *et al.*, 2008) ou (Chen *et al.*, 2007). Como o presente trabalho consiste da avaliação de quantidades relativamente grandes de imagens em tempo de execução, estes fatores foram determinantes na escolha do SURF. Outros trabalhos comparativos (Juan e Gwun, 2010) demonstram que o SURF tem taxa de acerto semelhante ao método SIFT, com uma tendência maior de ser prejudicado por variações de rotação, mas superior ao lidar com outros tipos de deformidades na cena.

O projeto concentra-se em localização de robôs em ambientes internos, nos quais espera-se que o robô se encontre, via de regra, em terrenos planos. Isso significa que é válido assumir que a navegação e a localização ocorrerão em um plano 2D, o que reduz o número de variáveis envolvidas e simplifica o problema. Não menos importante, esta simplificação aplicada ao algoritmo detector-descritor permite considerar que as assinaturas não sofrem alteração na sua rotação referente à linha do horizonte. Em termos de coordenadas tridimensionais, nesta aplicação específica a única variável de rotação que se altera é a azimutal, sendo que a inclinação (ou elevação) e o giro (*roll*) da câmera permanecem

constantes. Para refletir esta informação, realizamos uma alteração no método SURF para que o passo que garante invariância a rotação fosse ignorado, uma variante conhecida como Upright-SURF, ou U-SURF (Bay *et al.*, 2006). Com o U-SURF, a eficiência do método aumenta em termos de velocidade e capacidade discriminativa; ao mesmo tempo em que se reduz o tempo gasto pelo método eliminando um passo do processamento, evita-se potenciais falsos positivos, pois quaisquer pares de características com diferenças significativas em seu atributo de rotação serão descartados.

Note que o U-SURF não realiza a mesma triagem que o segundo critério proposto para escolha da imagem de perspectiva mais similar. O U-SURF rejeita pontos baseado na rotação destes em relação à região onde estão inseridos. O critério proposto rejeita assinaturas com muitos pares que apresentem disparidades de distância no eixo Y.

A pesquisa relacionada a estes descritores é recente e bastante ativa, portanto existem métodos mais novos que tem apresentado bons resultados, como o CenSurE (Agrawal *et al.*, 2008) e algumas variantes. Estas alternativas, no entanto, não se mostram definitivamente superiores e requerem testes comparativos mais aprofundados para justificar sua escolha, sendo o SURF (e sua versão modificada U-SURF) uma opção mais estabelecida.

O programa desenvolvido para a navegação consiste de um algoritmo comparativo entre assinaturas. O pareamento é realizado entre a assinatura calculada em tempo real e as assinaturas memorizadas previamente no percurso de gravação. O número de associações positivas entre cada par de assinaturas indica o grau de semelhança entre estas, e foi então o parâmetro usado como determinante da similaridade entre as cenas representadas. Quanto maior o número de pareamentos positivos entre as características visuais, melhor candidata é a imagem. Como já discutimos no Capítulo 3, cada ponto de uma assinatura possui diversos parâmetros descritores, entre eles sua coordenada (a,b) na imagem, contada em *pixels* a partir do canto superior esquerdo. Assim, cada par de pontos associados em duas imagens diferentes pode ser referenciado como o par de coordenadas $[(a,b),(c,d)]$.

É importante notar que mesmo nos testes em que o U-SURF foi utilizado, houve casos em que falsos positivos ocorriam em que um par de quadros não correspondente gerava uma quantidade alta de pares. Neste caso uma possível solução seria analisar os resultados das imagens vizinhas (que deveriam ter uma grande quantidade de pares no caso de um positivo verdadeiro), mas verificou-se que o problema ocorria especialmente quando uma cena era

particularmente desprovida de características marcantes e gerava um único ponto que se associava a diversas características da imagem alvo (Figura 4.5). Assim, foi necessário introduzir uma regra que identifica características associadas a muitas posições iguais em outras cenas, pois o critério de contagem de número de pares classificava estas assinaturas incompatíveis como boas candidatas. O uso do critério de distância média entre os pontos das assinaturas (segundo critério), por sua vez, elimina estas características e naturalmente contorna este problema, pois dá classificações inferiores a pares de pontos com posições inconsistentes.

O software desenvolvido deve ser capaz de aplicar a extração de características a um conjunto grande de imagens, obtendo assinaturas de características específicas de cada uma delas, através do SURF. O primeiro programa, que trata do percurso de gravação, recebe como entrada o conjunto de imagens obtidas pelo robô no percurso guiado manualmente por um humano. As assinaturas das imagens são então obtidas e gravadas em uma base de referência de assinaturas. A assinatura utilizada foi de 64 bits, pois a assinatura de 128 bits não resulta em melhorias significativas na qualidade dos descritores e acrescenta ao tempo de processamento que é significativo após muitas repetições (Juan e Gwun, 2010). O segundo programa, executado no percurso autônomo do robô, compara a assinatura da cena atual da câmera com as assinaturas previamente salvas. Testes *offline* preliminares, utilizando a câmera real e o simulador Stage no lugar do robô real, verificaram o funcionamento de ambas as etapas do robô. a qualidade da associação de pontos de referência entre imagens, tanto em ambientes ricos em detalhes visuais, com muitos elementos distintos, quanto em áreas vazias.

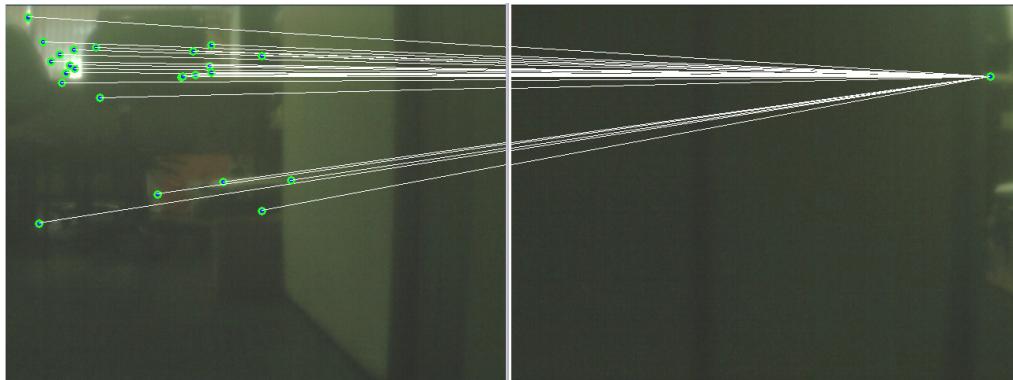


Figura 4.5: Erro de comparação do SURF em imagens com poucas características. O critério de escolha de imagens por proximidade espacial desclassifica estes resultados.

4.4.4 Travessia de Portas

Um dos desafios fundamentais da navegação de robôs em ambientes internos envolve lidar com os espaços estreitos e a pouca liberdade para manobras. Uma situação que resume muito bem este tipo de problema é a travessia de portas. É razoável assumir que em um ambiente interno um robô móvel irá se deparar com portas e passagens. Além disso, curvas fechadas em corredores exigem manobras de precisão semelhante.

É importante, assim, que um algoritmo de localização seja capaz de lidar com estas situações. O sucesso de uma possível manobra para travessia de porta dependerá da precisão do posicionamento do robô e da capacidade do mesmo de evitar erros de odometria. Este é um problema comum especialmente quando não se usa sensores de distância, e isso é ainda mais evidente nas situações em que o robô precisa dobrar uma esquina com um ângulo pequeno; a falta de espaço impede que o robô realize uma manobra mais suave, o que significa que na configuração com câmera fixa o robô deve se posicionar em relação à porta com esta fora do campo de visão limitado da câmera. Se houver uma parede ou obstrução próxima à frente do robô o problema fica mais difícil, pois a câmera frontal fixa estará muito próxima do obstáculo para obter uma quantidade suficiente de pontos distintos.

Existem diversas abordagens para garantir o alinhamento correto do robô em situações que demandam tal precisão. A abordagem sugerida neste trabalho é a adição de uma câmera extra no robô que auxilie o sistema nesta tarefa. As câmeras seriam posicionadas perpendicularmente uma à outra, sendo que uma das câmeras permanece voltada para a frente do robô como no sistema original, e a outra para os lados. Isso reduz o problema do ponto cego enquanto o robô se alinha à passagem, já que uma porta aberta de uma sala ou corredor são cenas mais ricas em pontos de referência. Para demonstrar o funcionamento de um sistema com câmera extra e para exemplificar como ele pode ser usado para solução do problema de travessia de portas, foram feitos testes *offline* com duas câmeras utilizando o algoritmo de memória proposto e o método de SURF para pareamento.

O princípio da demonstração da localização proposta foi simular uma situação real de travessia de porta. Uma sequência de imagens foi tomada mostrando o alinhamento do robô com a porta. Em uma aplicação prática, a presença de uma porta lateral ou alguma outra curva de ângulo fechado pode ser inferida pela odometria associada ao mapa; se houver grandes

variações angulares (z) com poucas variações de posição (x, y) entre as imagens do mapa de memória do robô, a câmera lateral deve ser usada para alinhamento e pareamento. A câmera lateral deve possuir o próprio mapa de memória, obtido no percurso de gravação. Este mapa pode ser reduzido e só incluir áreas próximas a portas para diminuir a quantidade de imagens armazenadas. Na prática, o método SURF deverá parear a cena que o robô captura pela câmera lateral à cena mais adequada da memória, que indique o melhor alinhamento com a porta (Figura 4.6). Quando o movimento é concluído, na aplicação sugerida o controle de localização pode retornar à câmera frontal. Da mesma forma que na proposta do critério de média de distâncias entre características visuais para a localização global, aplicamos aqui um critério alternativo para o pareamento entre imagens: utilizamos neste caso a variação de distância da posição em *pixels* no eixo X de características pareadas em diferentes imagens como critério para determinar o pareamento, já que esta é a direção do movimento da cena em relação à câmera lateral. O deslocamento das características visuais é acentuado, portanto, no eixo X (eixo horizontal).

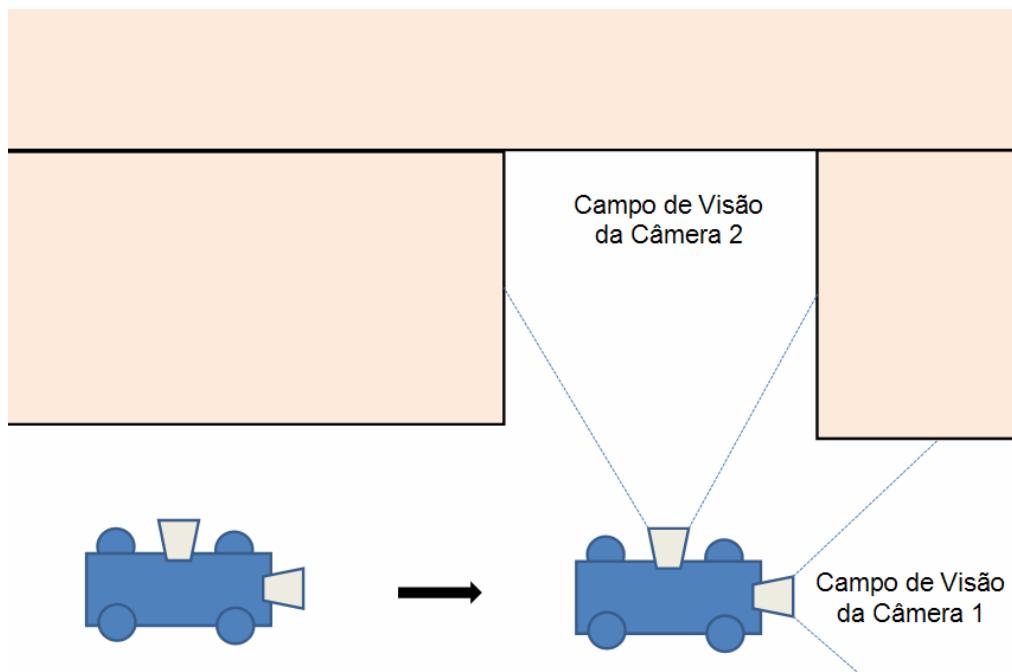


Figura 4.6: O robô observa a porta com a câmera lateral, buscando parear a imagem observada com a imagem memorizada, que representa o ponto ótimo para manobra.

5 Resultados e Conclusões

Nas seções seguintes, detalharemos os resultados dos experimentos realizados. Primeiramente, avaliaremos os resultados do teste de localização global. Os testes relativos à travessia de portas serão relacionados e serão apresentados a seguir. Finalmente, discutiremos a aplicação prática da metodologia no sistema de navegação de ambientes internos proposto. Em cada seção, os resultados destes experimentos serão analisados e as conclusões tiradas serão apresentadas.

5.1 Testes e Resultados da Localização Global

Para o teste da localização global, o percurso de gravação foi realizado em dois percursos distintos, em ambientes internos. A Figura 5.1 mostra exemplos de imagens capturadas de cada um dos trajetos realizados com o robô. O robô foi guiado por um ser humano realizando a gravação dos percursos. Cada percurso usado cobriu um trajeto de cerca de 20 metros, com cerca de 250 imagens em cada percurso. Em cada um dos percursos, realizamos duas repetições do trajeto, cada uma gerando uma base de imagens diferente. Isso possibilitou realizar a localização em um mesmo ambiente sob condições diferentes (principalmente variações de dia e noite e pequenas alterações em elementos da cena).

Para este teste de localização, o valor considerado para cálculo do erro foi o deslocamento na direção do movimento do robô apenas. Translações e deslocamentos angulares causados por imperfeições no movimento do robô foram desconsiderados. Além de ser a coordenada mais significativa, a coordenada da direção do deslocamento do robô é, via de regra para robôs terrestres, menos afetada por erros de odometria que o deslocamento angular.

De acordo com o primeiro critério, a imagem do percurso da memória com o maior número de características visuais pareadas com a imagem do percurso atual é a melhor candidata a representar a posição atual do robô. De acordo com o segundo critério, a melhor candidata a ser selecionada entre as imagens é aquela cuja disparidade média de distâncias no eixo Y entre as imagens é menor.



Figura 5.1: Diversos dos tipos de quadros encontrados no ambiente interno percorrido.

Cada característica visual extraída pelo SURF possui coordenadas (X, Y), contadas em *pixels*, que denotam sua posição na imagem. Duas características visuais pareadas entre duas imagens possuem cada uma sua respectiva coordenada (X, Y). Subtraindo-se as coordenadas das características uma da outra, temos a diferença de posição entre os pontos em suas respectivas imagens. Por exemplo, a diferença de distância entre duas imagens A e B, com coordenadas (X_a, Y_a) e (X_b, Y_b) , respectivamente, será dada no eixo X por $|X_a - X_b|$ e por $|Y_a - Y_b|$ no eixo Y. A diferença representa a distância e portanto é tomada em módulo; o sinal não é importante. O segundo critério propõe que se a média aritmética das diferenças de posição das características de duas imagens é pequena, as perspectivas das duas cenas são próximas. Como no caso queremos estimar a localização na coordenada da direção de movimento do robô, tomamos a diferença de distâncias no eixo Y, ou seja, os valores de $|Y_a - Y_b|$ para cada característica (ignorando assim variações de translação e rotação). Pares de imagens que apresentaram um número baixo de pares (testes indicaram que um mínimo de cinco pares é um bom pré-requisito para as imagens candidatas) eram eliminados, para evitar falso-

positivos no cálculo deste segundo critério. Ambos os critérios foram testados sobre os mesmos conjuntos de imagens.

A Figura 5.2 e a Figura 5.3 mostram exemplos dos resultados do pareamento de assinaturas entre duas imagens, e mostram exemplos da diferença de resultados obtidos com o uso dos dois critérios a partir da mesma imagem observada (imagens da esquerda).

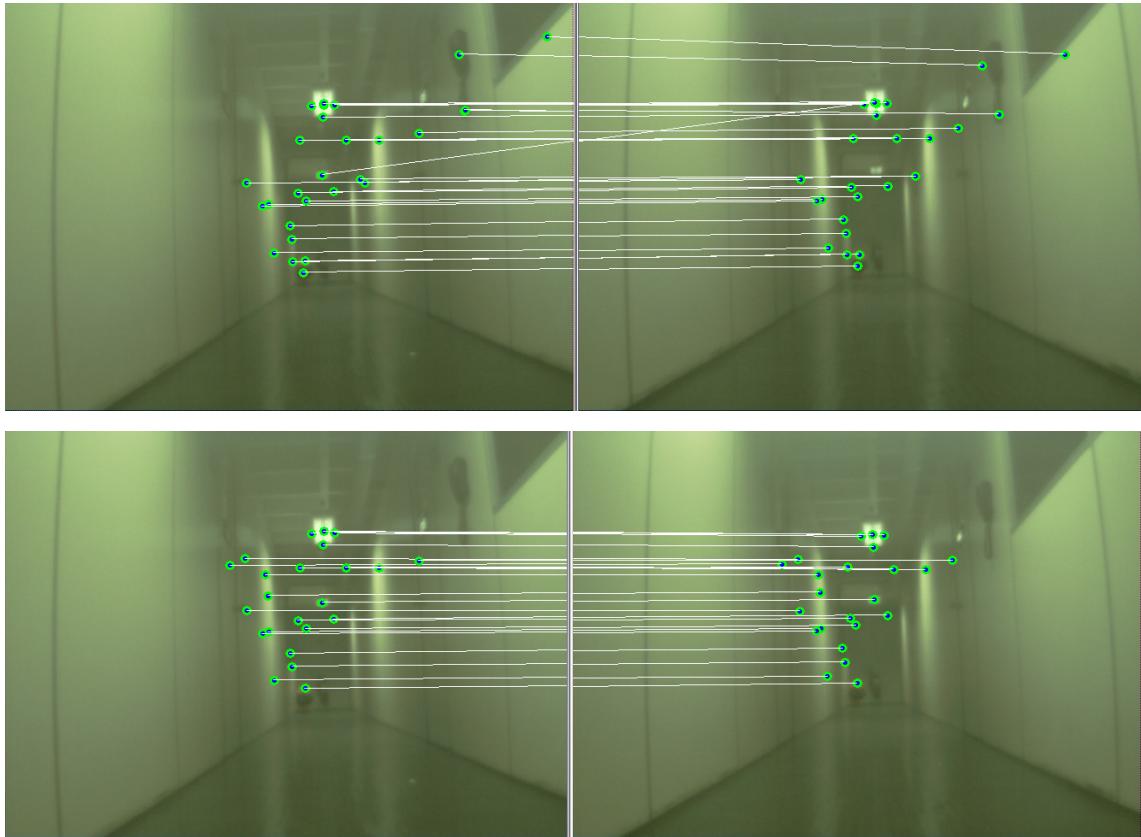


Figura 5.2: Estes dois pares mostram diferentes resultados (imagens da direita) obtidos pelos critérios diferentes (primeiro critério em cima, segundo critério embaixo), para a mesma imagem (a imagem da esquerda é a mesma nos dois casos).

Podemos notar na figura a diferença entre os critérios. Nas duas figuras, pelo primeiro critério (par de imagens de cima), vê-se um número maior de pares. No segundo critério (par de imagens de baixo) vemos um número menor de pares, mas características espacialmente próximas em sua posição no eixo vertical. A diferença de erro entre a imagem observada e a imagem eleita da memória foi medida como a diferença entre os dados odométricos associados às duas imagens. Para isso é essencial que o robô comece os dois percursos no mesmo lugar pois a odometria é relativa à origem do movimento. Analisando todas as

comparações realizadas, calculamos que para o primeiro critério, o erro médio da localização estimada em relação à localização armazenada adquirida pela odometria, na coordenada da direção do deslocamento, foi de 23.98 cm, e o desvio padrão calculado foi de 40.52 cm. Para o segundo critério, com o mesmo conjunto de dados, o erro médio da localização estimada em relação à localização real foi de 16.26 cm. O desvio padrão neste caso foi de 25.13 cm.

Em um computador Intel Core 2 Duo 2.20GHz com 2 GB de RAM, o tempo médio de processamento para aquisição das características da imagem observada (o que só precisa ser feito uma vez por imagem de entrada) foi de 330 ms. O tempo médio obtido para comparação entre duas assinaturas, que foi realizado para todas as imagens do conjunto, foi de 85 ms por imagem. Isto mostra que especialmente o descriptor do método SURF permite comparação rápida de imagens, adequado para uso em tarefas em tempo real, dependendo da aplicação.

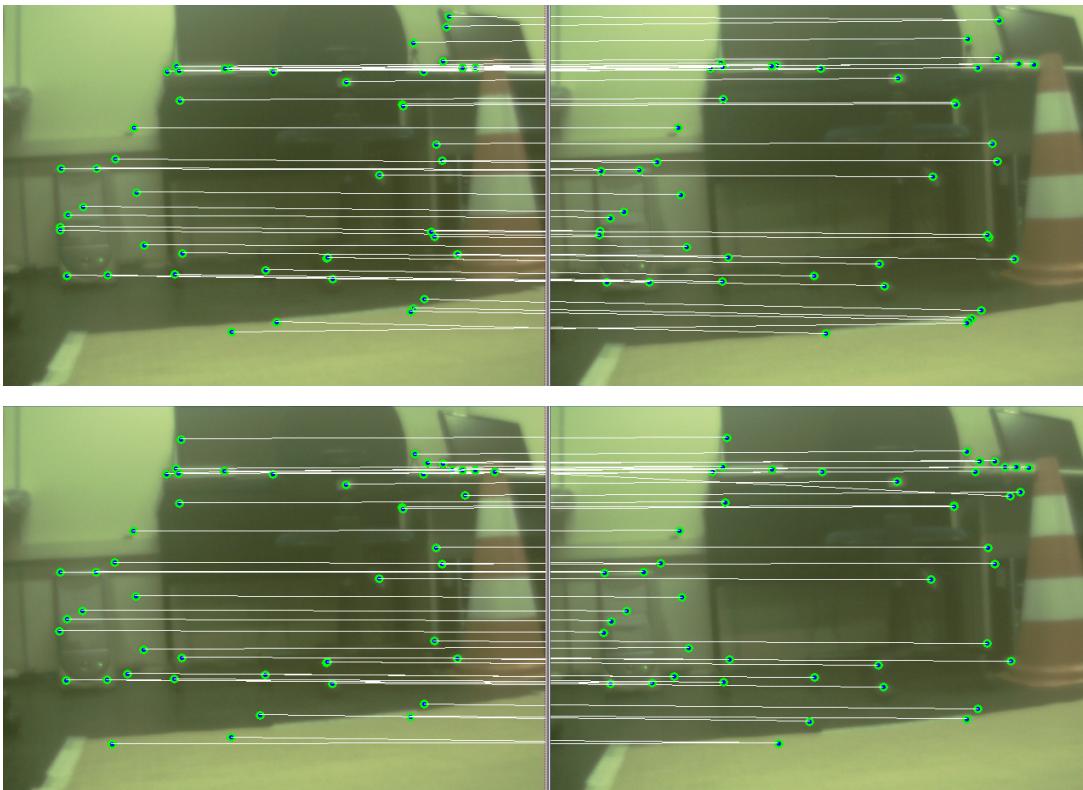


Figura 5.3: Pares de resultados obtidos pelos dois critérios em um ambiente diferente.

Para reduzir o número de comparações, em uma situação prática para manter a localização, pode-se ainda usar o conhecimento sobre a posição do robô para eliminar comparações irrelevantes, buscando-se apenas imagens próximas à posição anterior conhecida do robô.

A Figura 5.4 mostra gráficos dos resultados de três imagens diferentes obtidas em pontos distintos do trajeto em relação à base de dados. O círculo vermelho representa o índice onde houve maior número de pareamentos para cada imagem. A imagem usada na linha rosa do gráfico foi um corredor longo e vazio, a imagem usada na linha azul foi um corredor curto (próximo à parede) e a imagem usada na linha verde foi dentro de uma sala, com muitas características visuais complexas. A partir disto podemos ver o perfil das cenas: corredores longos apresentam variações suaves no número de pares, e devido à falta de características marcantes, apresentam menos pares no geral; a sala, por outro lado, apresentou grandes quantidades de características visuais devido ao alto número de características de destaque no ambiente, e apresentou variações mais bruscas no número de características devidas à variação mais acentuada de perspectiva. A imagem tomada no corredor curto apresenta um meio termo entre estes dois casos.

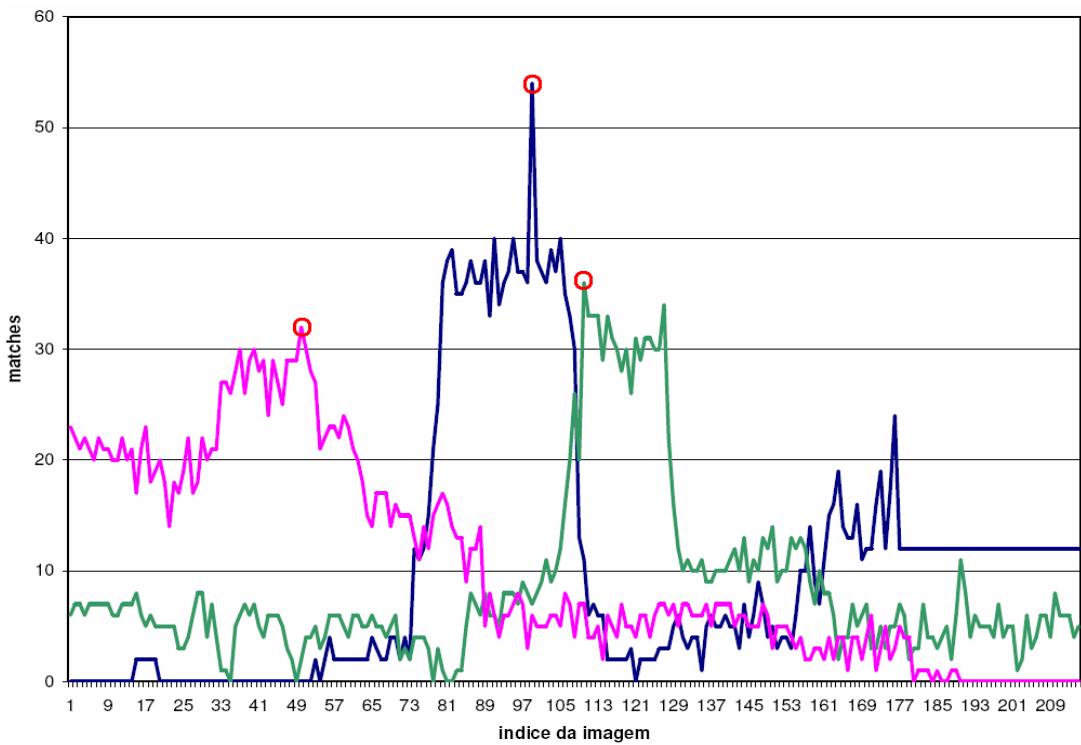


Figura 5.4: Número de pares (matches) obtidos pelo método SURF em relação a diferentes imagens, comparado a uma mesma base de imagens.

A Figura 5.5 mostra diversos exemplos dos resultados de pareamento obtidos entre diversas assinaturas. É possível ver como números maiores de pares associados de

características visuais aparecem em imagens de perspectivas mais próximas, e onde há mais características evidentes.

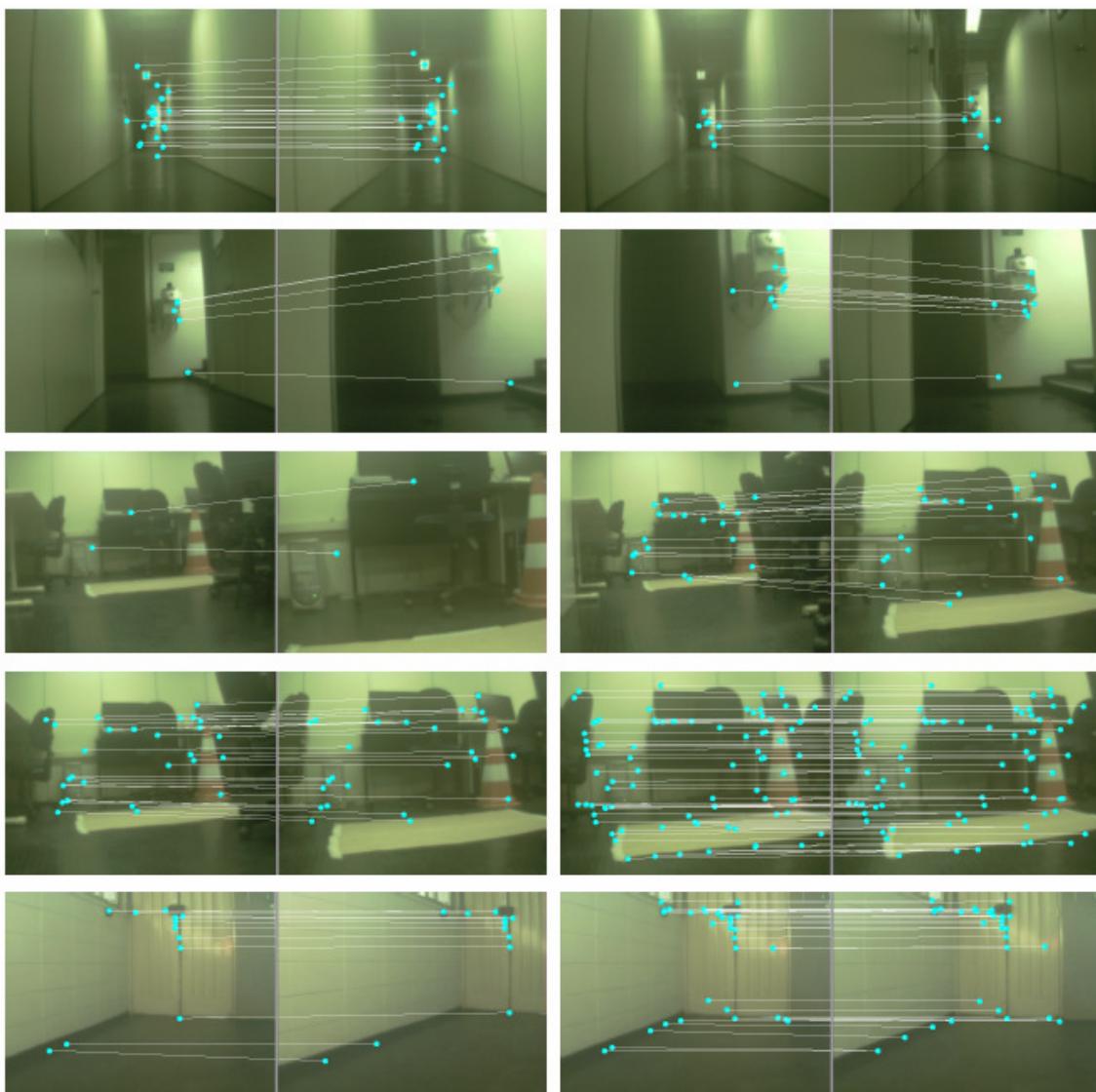


Figura 5.5: O Mosaico mostra a diferença que variações sutis de perspectiva causam no número pares de características visuais em diversas situações.

A Tabela 1 apresenta dois exemplos da comparação odométrica realizada para verificar a precisão do pareamento. Ela mostra os pareamentos em dois pequenos percursos de aproximadamente 5 m, apresentando o número de pares obtidos e as diferenças de odometria calculadas. Note que no primeiro exemplo o melhor número de pares foi 24, em imagem que apresentou erro de 24 cm. O valor está destacado em negrito na tabela.

Tabela 1: Exemplo de pareamentos em trecho - Número de pares e diferença odométrica (m)

10	-3.47	11	-2.84	13	-2.36	7	-0.92	24*	-0.24*
3	-3.47	7	-2.84	10	-2.36	9	-0.91	18	-0.22
6	-3.47	9	-2.84	15	-2.34	7	-0.87	19	-0.21
4	-3.47	12	-2.81	13	-2.30	10	-0.84	21	-0.19
9	-3.47	7	-2.81	15	-2.22	16	-0.84	18	-0.04
8	-3.47	8	-2.81	17	-2.12	19	-0.84	15	+0.03
5	-3.45	8	-2.81	9	-1.69	18	-0.84	17	+0.19
6	-3.43	8	-2.80	14	-1.65	16	-0.84	22	+0.27
6	-3.39	4	-2.77	17	-1.51	16	-0.81	23	+0.37
6	-2.35	3	-2.74	11	-1.41	18	-0.79	19	+0.45
4	-3.27	6	-2.72	15	-1.30	21	-0.77	21	+0.51
11	-3.13	4	-2.70	15	-1.27	18	-0.75	21	+0.61
8	-3.05	4	-2.69	15	-1.23	15	-0.72	17	+0.70
7	-2.99	9	-2.67	16	-1.14	17	-0.60	19	+0.80
6	-2.95	7	-2.59	12	-1.08	16	-0.52	22	+0.88
9	-2.89	8	-2.49	11	-1.04	21	-0.42	20	+0.98
9	-2.85	7	-2.40	4	-0.98	18	-0.36	18	+1.05
6	-2.85	8	-2.39	10	-0.95	18	-0.29	14	+1.12

5.2 Testes e Resultados da Travessia de Portas

Em um conjunto de imagens organizadas topologicamente como o utilizado neste trabalho, a localização relativa a marcos significativos do ambiente como portas e cruzamentos é muito importante. A seguir apresentamos os testes de um sistema para detecção de portas, e analisamos a precisão da localização para uma possível manobra de travessia da porta.

Os testes da travessia de portas se deram de forma semelhante aos testes de localização global, mas em um escopo reduzido. Para detecção e alinhamento em relação à porta, não é necessário que longos percursos sejam gravados, apenas o percurso nas redondezas imediatas da porta à qual o robô irá se alinhar. Assim evita-se que erros maiores de odometria afetem as medições. Para o teste, foi considerada uma situação comum em ambientes internos, em que o robô navega em um corredor e deverá realizar uma curva de 90° para alinhar-se com a porta e entrar.

Como dito, neste teste a câmera foi posicionada lateralmente ao robô, de forma perpendicular à câmera frontal. A medida de largura do robô é de 49.7 cm, e a medida da porta em questão 80.0 cm, o que permite ao robô uma margem de erro de cerca de 15 cm para qualquer dos lados para realização da manobra. Cinco imagens do ponto em que o robô está alinhado ao centro da porta (o ponto ideal para realização da manobra) foram obtidas, em distâncias variadas em relação à porta. A distância destes pontos em relação à porta foi medida. O robô foi orientado paralelamente ao centro do corredor, e posicionado em uma marcação no chão para que a partida em todos os percursos ocorresse da mesma posição. Em cada percurso gravado, a informação odométrica foi calculada. Assim, o percurso gravado consistiu da passagem do robô por toda a extensão da porta. Por fim, as imagens tomadas pela câmera lateral do robô em cada um dos 8 percursos foram comparadas aos cinco quadros centrais obtidos, e buscou-se determinar qual das imagens de cada um dos conjuntos obtidos (cada conjunto possuindo entre 150 e 300 imagens) melhor representava a imagem central da porta. Assim, foram obtidos 40 resultados. Sabendo-se o valor da odometria da imagem em questão, foi possível estimar o erro entre a posição real do centro da porta, ideal para a manobra, e a posição da imagem escolhida para a manobra do robô.

As comparações entre as imagens base e os percursos não se deram apenas pela contagem do número de características SURF obtidas pelo pareamento de cada par de imagens. Analogamente ao segundo critério proposto no teste de localização global (a média das distâncias entre as características), utilizou-se neste caso a média da distância das características no eixo X (no caso, $|X_a - X_b|$). Com a câmera lateral, o robô se desloca perpendicularmente à direção da câmera, e a variação de perspectiva é mais acentuada no eixo X. Neste caso, portanto, o método usado foi o cálculo da distância entre as assinaturas das imagens. O número de pares foi usado para que fosse possível eliminar casos com números muito pequenos de pares (um mínimo de 5 pares foi considerado requisito razoável para consideração de um quadro candidato). Considerando-se apenas casos em que o pareamento oferece 5 ou mais pares positivos, o dx das assinaturas (a distância no eixo X, dada pela diferença em *pixels* entre cada característica visual de cada imagem) foi calculado. Quanto menor o dx , mais próxima a posição das assinaturas relativamente às suas respectivas imagens, e maior a proximidade das perspectivas. Assim, o pareamento com o menor valor de dx foi considerado o melhor candidato. Este sistema obteve um erro médio de odometria de

8.63 cm, com desvio padrão de 6.72 cm. Em 90% dos casos o erro foi pequeno o suficiente (menor que 15 cm) para que a manobra fosse realizada com sucesso. A Figura 5.6 mostra um exemplo de resultado do método.

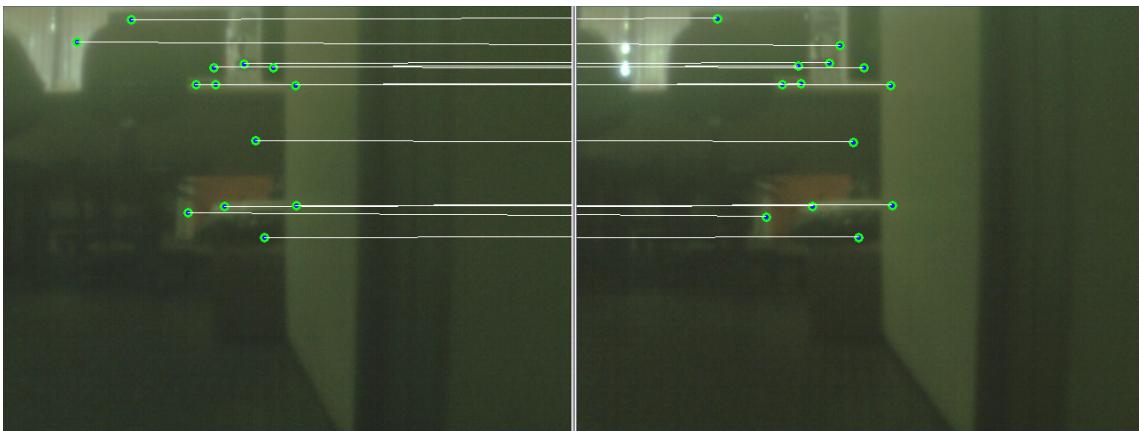


Figura 5.6: Exemplo de resultado da travessia de porta, baseado na distância entre características no eixo X.

Estes resultados mostram que a câmera lateral permite a localização da porta com boa precisão. Em uma situação real, uma configuração semelhante a essa com *pan-tilt* pode ser usada para que as câmeras possam observar ambos os lados do robô (não simultaneamente).

Devido ao fato de que a detecção de portas consiste de manobras precisas relativas a obstáculos próximos, os pontos de referência gerados pelo SURF pertencem a características reais que estão mais próximas à câmera. Assim, o deslocamento do robô acarreta em deslocamentos mais evidentes destas características relativamente ao observador (câmera). Este aspecto favorece o resultado do método SURF baseado no critério de distância entre características visuais, pois a distinção entre imagens é maior sem que haja necessidade de grandes deslocamentos, ao contrário da localização global em que muitas imagens da base de dados consistem de longos corredores com características distantes. O fato de o deslocamento do robô ser perpendicular à orientação da câmera também favorece a percepção de movimento. Estes fatores explicam os melhores resultados obtidos neste caso, em relação à localização global com o deslocamento estimado através da câmera frontal.

6 Discussão

O projeto proposto teve por objetivo desenvolver um sistema de baixo custo de localização a partir de visão computacional, através do uso de câmera monocular como principal sensor. Visou-se determinar a possibilidade do uso de métodos detectores-descritores de características em situações complexas, na localização do robô em ambientes internos, onde há poucos elementos de destaque no ambiente e com variações sutis de perspectiva.

De acordo com a análise em (Mikolajczyk e Schmid, 2005), métodos detectores-descritores baseados em espaço de escala, como o SURF, são especialmente adequados a perceber variações de textura. Um ambiente interno típico como os utilizados no projeto possui diversidade de texturas menor que ambientes externos. Ainda assim, com parâmetros de *threshold* padrão, o método resultou em quantidades suficientes de pontos detectados, exceto nos casos discutidos de proximidade a paredes e áreas desprovidas de características distintas. Assim, espera-se que o método funcione tão bem ou até melhor em ambientes externos.

Nos testes, o SURF gerou boa precisão de localização com ambos os critérios individualmente. Acreditamos que o número de pares seja uma boa opção para localização global mais “grossa” (*coarse*), enquanto o método de distância em *pixels* entre as características seja uma boa alternativa para refinar os resultados e encontrar a melhor imagem em uma vizinhança de imagens próximas. O critério de distâncias médias é eficaz no ambiente proposto, mas baseia-se em assumir que características visuais com posições próximas na imagem 2D observada são próximas na realidade tridimensional. Casos onde isso não é verdade (o efeito paralaxe) podem acarretar em falso-negativos. Com ajustes paramétricos estima-se que seja possível utilizar este sistema em ambientes externos. Ainda assim, os resultados mostram que o conhecimento da aplicação e do contexto do ambiente permitiu que o critério original fosse aperfeiçoado através da adição de uma nova forma de classificação das comparações entre assinaturas. O novo critério contextualizado apresentou melhores resultados que o critério mais geral.

Pretende-se realizar testes para verificar possíveis combinações entre os dois critérios e explorar a interação entre eles. Em aplicações que necessitam de mais robustez, porém, é possível obter-se melhores resultados sem abrir mão da configuração com apenas uma câmera. Outros métodos de localização que podem ser usados com câmera monocular incluem técnicas de odometria visual como desenvolvido por Levin e Szeliski, (2004) e fluxo óptico como em Sokolov *et al.*, (2010). O OpenSURF, implementação usada do SURF, oferece métodos de estimativa de movimento por fluxo óptico, que pode ser estimado comparando-se a posição dos pontos obtidos na imagem atual com a posição dos pontos em imagens anteriores (Figura 6.1).

Uma alternativa para tornar o sistema mais resistente a alterações no ambiente é a atualização do mapa do ambiente. Isto pode ser realizado se a localização do robô for confiável no momento e se a perspectiva for adequada. Durante a navegação, o robô gera um novo conjunto de imagens do ambiente para substituir uma região do conjunto de imagens memorizadas anteriormente, ou para gerar um novo conjunto a ser usado juntamente ao anterior, em ambos os casos assimilando alterações no ambiente. Neste caso, técnicas complementares de navegação como as já mencionadas devem ser usadas para garantir a precisão odométrica e compensar dificuldades do método de extração de características visuais causadas por variações no ambiente.

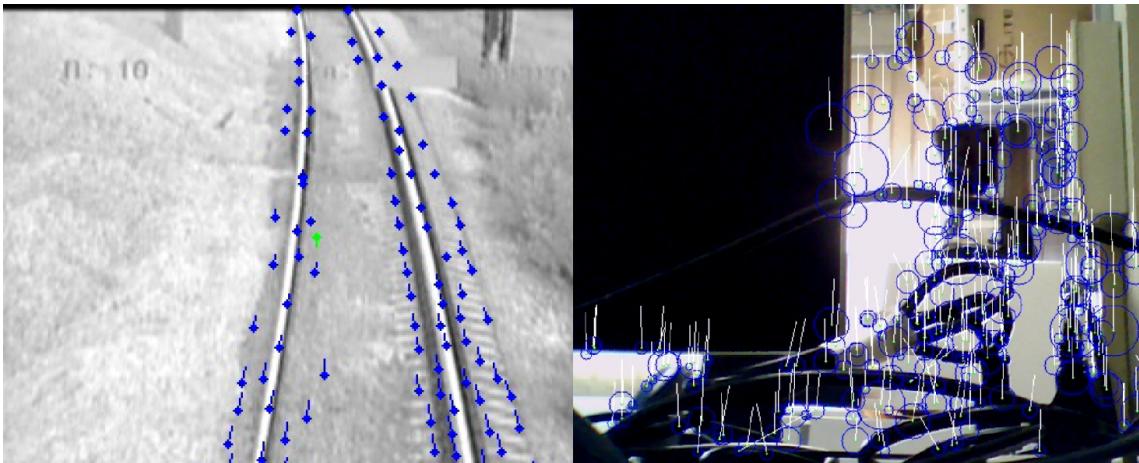


Figura 6.1: Fluxo óptico. À esquerda, um exemplo do trabalho de Sokolov *et al.*, (2010). À direita, o fluxo estimado através do OpenSURF.

A localização global proporcionou bons resultados com custo computacional reduzido, e robustez a diferenças na imagem. A simetria natural de certos ambientes internos torna muitas vezes necessário que métodos probabilísticos como o Filtro de Partículas utilizando um *laser* iterativamente aproximem sua posição durante a navegação. Nos testes realizados com o SURF, a localização pode ocorrer com precisão através da comparação de apenas um quadro, sendo que é possível a adaptação de métodos como o Filtro de Partículas para a tarefa de visão. De forma semelhante ao que foi realizado na navegação desenvolvida, é possível também aperfeiçoar a localização em relação às características observando as distorções na assinatura observada, de escala e de rotação e translação, que denotam variações de perspectiva (Jafar *et al.*, 2010) de modo similar às abordagens probabilísticas para localização com sensores de *range*. Isto ofereceria uma estimativa de localização em relação à assinatura a partir de pontos de vista distintos.

A sugestão de travessia de portas com duas câmeras é uma alternativa viável para muitas aplicações. O custo e a carga computacional de um projeto ainda seriam inferiores aos de um sistema com câmeras estereoscópicas. Dependendo dos requisitos de precisão, a câmera secundária da configuração proposta pode ser utilizada apenas em situações críticas como em locais próximos a portas, locais que exijam manobras complexas ou locais que no mapa de memória produzem poucos pontos de interesse na câmera principal. Configurações de câmeras estereoscópicas necessitam de calibragem que requer precisão e gastos computacionais elevados, além de não proporcionar o aumento no campo de vista obtido com a configuração proposta neste projeto.

A câmera lateral pode ser utilizada com a configuração completa em *pan-tilt*, ou apenas a câmera secundária. No caso de uma câmera frontal fixa e câmera secundária em um *pan-tilt*, é possível que a câmera secundária tenha liberdade de apontar para a área onde mais pontos de referência são obtidos naquele momento, inclusive para trás, o que a torna útil em uma configuração de corredor. Todas estas configurações alternativas proporcionam informação adicional ao sistema em relação à apenas uma câmera, com o custo de uma câmera extra e aumento no processamento necessário, então a aplicação deve ser considerada na decisão.

Os aprimoramentos citados, porém, aumentam o custo computacional do método. Outras aplicações também podem exigir tempos de processamento menores que os aqui obtidos no projeto proposto. Para estes casos, existem implementações de SURF realizadas

em hardware como FPGAs, que reduzem muito o tempo para computação do algoritmo (Se *et al.*, 2004).

No que se refere à aplicação em patrulha e monitoramento de robôs móveis para ambientes internos, relacionada com o INCT-SEC, acreditamos que os resultados mostram que o projeto pode ser aplicado em um sistema real. É possível integrar o método usado neste trabalho com as alternativas apresentadas anteriormente e com diversas outras técnicas de acordo com a aplicação. Além do sistema de localização proposto, por exemplo, é possível adicionar-se um sistema de detecção de pessoas. Tal funcionalidade seria útil em um sistema de monitoramento interno, que pode incluir segurança contra invasões. Detecção de pessoas também oferece ao sistema robustez ao dinamismo do ambiente (tráfego de pessoas), e pode ser alcançado através de métodos usando fluxo óptico ou detecção de movimento (Dalal *et al.*, 2006, Lin *et al.*, 2010, Zhao e Nevatia, 2004). A técnica de fluxo óptico pode até mesmo ser aplicada para detecção de obstáculos e pessoas usando os próprios pontos obtidos com um detector de características visuais, semelhante à proposta usada em (Campbell *et al.*, 2005).

Mesmo com esses aprimoramentos, acreditamos que seja possível o projeto de um sistema de navegação simples e eficiente utilizando os resultados obtidos. Apresentaremos a seguir a proposta de um sistema de navegação e manutenção de localização baseado nos resultados obtidos para ambientes internos.

6.1 Considerações Finais e Trabalhos Futuros

A intersecção entre as áreas de visão computacional e a robótica móvel é ampla e diversa em aplicações e métodos. Entre as possíveis abordagens, o uso na robótica de câmeras monoculares simples é de especial nota por causa do baixo custo aliado à riqueza de informações oferecidas. Este trabalho apresentou um método de localização global e local utilizando detecção de pontos de referência, através de um método detector-descritor baseado em espaço de escala, em um mapa visual memorizado pelo robô. Apresentou-se também como este método pode ser usado em um caso particular como a travessia de portas. Os experimentos e resultados obtidos nesta dissertação foram publicados em (Couto e Osório, 2012).

Observou-se que ambos os critérios de pareamento usados possuem vantagens e desvantagens dependendo da situação. Para trabalhos futuros, propõe-se integrar os dois critérios propostos para pareamento entre imagens. Acredita-se que o critério de contagem de pares seja mais eficiente na localização global e o critério de distâncias médias se mostrou mais eficiente em ajustes finos. Pretende-se testar o uso em conjunto dos critérios e sua aplicação em ambientes externos, onde o detector-descritor SURF retorna melhores resultados e efeitos de paralaxe podem causar problemas para o critério de distâncias médias. Em curto prazo, testes mais detalhados do sistema de navegação e localização local proposto serão realizados. Pretende-se também integrar o sistema com técnicas de odometria visual para maior robustez em situações onde a extração de características visuais não é suficiente. Entre os objetivos de médio e longo prazo para a continuidade dos trabalhos, pretende-se explorar o desenvolvimento de um sistema que seja capaz de realizar SLAM monocular através dos pontos de referência obtidos.

Trabalhos no LRM (Laboratório de Robótica Móvel) vêm sendo desenvolvidos com sensores como o Kinect²⁵. Sensores de *range* de baixo custo e alcance limitado como o Kinect ou um sonar podem ser úteis integrados com a câmera. Estes podem garantir a localização local do robô e a navegação reativa, enquanto a câmera, que não possui limitações de alcance, pode ser usada na localização global, mantendo-se ainda uma configuração de custo relativamente baixo. Outros trabalhos do laboratório como (Sales *et al.*, 2011), relativo à navegação por mapas topológicos, também podem se aproveitar da localização realizada com um sensor de custo mais baixo que o *laser*.

Robôs móveis têm sido aplicados para automatização de diversas tarefas, conferindo segurança e eficiência à realização das mesmas. Conforme a robótica se torna cada vez mais ubíqua na sociedade e seu papel torna-se cada vez mais determinante, as pesquisas na área crescem em importância e atualidade. Espera-se que o trabalho realizado contribua com objetivos de longo prazo das áreas nas quais está inserido, como a melhor compreensão e replicação do comportamento do sistema visual humano na visão computacional, bem como o desenvolvimento de veículos mais autônomos e seguros na robótica móvel.

²⁵ <http://www.xbox.com/pt-BR/kinect>, acesso em 01/2012.

Referências Bibliográficas

- ABDEL-HAKIM, E.; FARAG A. A. *CSIFT: A SIFT Descriptor with Color Invariant Characteristics*. Computer Vision and Image Processing Laboratory (CVIP) University of Louisville, Louisville, KY, 2006.
- AGRAWAL, M.; KONOLIGE, K.; BLAS, M.R. *Censure: Center surround extrema for realtime feature detection and matching*. In D. A. Forsyth, P. H. S. Torr, and A. Zisserman, editors, *ECCV* (4), volume 5305 of *Lecture Notes in Computer Science*, p. 102–115. Springer-Verlag New York, Inc., 2008.
- ANGUELOV, D.; KOLLER, A.; PARKER, E.; THRUN, S. Detecting and Modeling Doors with Mobile Robots. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2004.
- ÅSTRÖM, K. J; HÄGGLUND, T.; HANG, C. C.; HO, W. K. Automatic Tuning and Adaptation for PID Controllers - A Survey. *Control Engineering Practice*. v. 1, n. 4, p. 699-714, 1993.
- ÅSTRÖM, K. J.; MURRAY R. M. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008.
- AZUMA, R. T. A Survey of Augmented Reality. *Teleoperators and Virtual Environments* v. 6, n. 4, p. 355-385, 1997.
- BALLARD, D. H.; BROWN, C. M.: *Computer Vision*. Prentice Hall, 1982.
- BANKMAN, D. J.: Study of compact stereoscopic system for target distance estimation. *Proceedings of SPIE, the International Society for Optical Engineering*, v. 6238, p. 62380E.1 - 62380E.7, 2006.
- BASRI, R.; RIVLIN, E., SHIMSHONI, I. Visual Homing: Surfing on the Epipoles. *International Journal of Computer Vision*. v. 33, n. 2, p. 117–137, 1999.
- BAUER, J.; SUNDERHAUF, N.; PROTZEL, P. Comparing several implementations of two recently published feature detectors (Published Conference Proceedings style). In *Proc of the International Conference on Intelligent and Autonomous Systems, IAV*, 2007.
- BAZEILLE, S.; FILLIAT, D. Combining Odometry and Visual Loop-Closure Detection for Consistent Topo-Metrical Mapping. *Procedding of the conference on COGnitive systems with Interactive Sensors (COGIS2009)*. 2009.
- BAY, H.; FASEL, B.; Van GOOL, L. Interactive Museum Guide: Fast and Robust Recognition of Museum Objects. *Proceedings of the first international workshop on mobile vision*, May 2006.
- BAY, H.; TUYTELAARS, T.; Van GOOL; L. SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding (CVIU)*, v. 110, n. 3, p. 346-359, 2008.

-
- BEKEY, G. A. *Autonomous robots: From Biological Inspiration to Implementation and Control*. MIT Press, 2005, 577 p.
- BONATO, V. *Projeto de um módulo de aquisição e pré-processamento de imagem colorida baseado em computação reconfigurável e aplicado a robôs móveis*. Dissertação de Mestrado, ICMC - Universidade de São Paulo, USP, Brasil, 2004.
- BONIN-FONT, F.; ORTIZ, A.; OLIVER, G. Visual Navigation for Mobile Robots: A Survey. *Journal of Intelligent and Robotic Systems*, v. 53, p. 263-296, 2008.
- BRADSKI, G.; KAEHLER, A.: *Learning OpenCV: Computer Vision with the OpenCV Library*. Cambridge, MA: O'Reilly Media, Inc., 2008.
- BROWN, M.; LOWE, D.: Automatic Panoramic Image Stitching using Invariant Features. *International Journal of Computer Vision*, v.74, n.1, p. 59-73, 2007.
- BURGHOUTS, G. J.; GEUSEBROEK, J. M.: Performance Evaluation of Local Colour Invariants. *Computer Vision and Image Understanding*, v. 113, p. 48–62, 2009.
- CALISI, D.; FARINELLI, A.; GRISETTI, G.; IOCCHI, L.; NARDI, D.; PELLEGRINI, S.; TIPALDI, D.; ZIPARO, V. A. Contextualization in Mobile Robots. *ICRA-07 Workshop on Semantic Information in Robotics*. Italy, 2007.
- CAMPBELL, J.; SUKTHANKAR, R.; NOURBAKHSH, I.; PAHWA, A. A. Robust Visual Odometry and Precipice Detection System Using Consumer-grade Monocular Vision. *Robotics and Automation, ICRA 2005. Proceedings of the 2005 IEEE International Conference*, p. 3421-3427, 2005.
- CANNY, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. 8, n. 6, p. 679-698, 1986.
- CASTRUCCI, P.; SALES, R. M. *Controle Digital*, v. 3. Editora Edgard Blücher, 1990.
- CHEN, C-H; LEE, J-S; SUN, Y-N. Wavelet transformation for gray-level corner detection. *Pattern Recognition*. v. 28, n. 6, p. 853-861, 1995.
- CHEN, W-C; XIONG, Y.; GAO, J.; GELFAND, N.; GRZESZCZUK, R. Efficient Extraction of Robust Image Features on Mobile Devices. In *ISMAR 2007. Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, Washington, DC, USA, p 1-2, 2007.
- CHOSET, H.; LYNCH, K.; HUTCHINSON, S.; KANTOR, G.; BURGARD, W.; KAVRAKI, L.; THRUN, S. *Principles of Robot Motion*. Cambridge, Massachussets. The MIT Press, 2005.
- COLLETT, T. H. J.; MacDONALD, B. A.; GERKEY, B. A. Player 2.0: Toward a Practical Robot Programming Framework". In *Proceedings of the Australasian Conference on Robotics and Automation (ICRA 2005)*, Sydney, Australia, December 2005.

-
- COUTO, L. N. *Manipulação de imagens e análise do processamento de imagens utilizando hardware reconfigurável*. Trabalho de Conclusão de Curso de Engenharia de Computação - Universidade de São Paulo, USP, Brasil, 2008.
- COUTO, L. N.; OSÓRIO, F. S. Autonomous Self-Localization of Mobile Robots through Reference Point Based Computer Vision. *Proceedings of CBSEC - Second Brazilian Conference on Critical Embedded Systems*, 2012. p. 1-5.
- CROWLEY, J. L.; CHENAVIER, F. Position Estimation for a Mobile Robot Using Vision and Odometry. *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, p. 2588-2593, 1992.
- DALAL, N.; TRIGGS, B.; SCHMID, C. Human Detection Using Oriented Histograms of Flow and Appearance. *Proceedings of ECCV*, v.2, p. 428-441, 2006.
- DAVIES, E. R. On the noise suppression and image enhancement characteristics of the median, truncated median and mode filters. *Pattern Recognition Letters*, v. 7, p. 87-97, 1988.
- DAVISON, A. J.; REID, I. D.; MOLTON, N. D.; STASSE, O. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, v. 29, n. 6, p. 1052—1067, 2007.
- De BERG, M.; CHEONG, O.; KREVELD, M.; OVERMARS, M. *Computational Geometry: Algorithms and Application*, 3rd ed. Springer-Verlag, 2008.
- DELAGE, E.; LEE, H.; NG, A.Y. Automatic Single-Image 3d Reconstructions of Indoor Manhattan World Scenes. *In Proceedings of ISRR*, p. 305-321, 2005.
- DELLAERT, F.; FOX, D.; BURGARD, W.; THRUN, S. Monte Carlo Localization for Mobile Robots. *IEEE International Conference on Robotics and Automation (ICRA99)*, 1999.
- DESOUZA, G. N.; KAK, A. C. Vision for Mobile Robot Navigation: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, p. 237-267, 2002.
- DIJKSTRA, E. W. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*. v. 1, p. 269-271, 1959.
- DUDA, R.O.; HART, P. E. Use of Hough transform to detect lines and curves in picture. *Communications of the ACM*, v.15, n.1, p. 11-15, 1972.
- DUDEK, G.; JENKIN, M. *Computational Principles of Mobile Robotics*. Cambridge: Cambridge University Press, 2000.
- DUDEK, G.; MARINAKIS, D. Topological Mapping with Weak Sensory Data. *InProceedings of AAAI*, p. 1083-1088, 2007.

-
- DUSHA, D.; BOLES, W.; WALKER, R. Attitude Estimation for a Fixed-Wing Aircraft Using Horizon Detection and Optical Flow. In *DICTA*, p. 485-492, 2007.
- EBRAHIMI, M.; MAYOL-CUEVAS, W.W. SUSurE: Speeded Up Surround Extrema Feature Detector and Descriptor for Realtime Applications. In *Computer Vision and Pattern Recognition Workshops. CVPR Workshops 2009. IEEE Computer Society Conference*, p. 9-14, 2009.
- ELFES, A. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer Society Press*, v. 22, p. 22:46–57, 1989.
- EVANS, C. *Notes on the OpenSURF Library*. University of Bristol, 2009.
- FITZGIBBON, A.W.; ZISSEMAN, A. Automatic Camera Recovery for Closed or Open Image Sequences. *European Conference on Computer Vision*, p. 311-326, 1998.
- FORSSÉN, P. E.; MEGER, D.; LAI, K.; HELMER, S.; LITTLE, J. J.; LOWE, D. G. Informed Visual Search: Combining Attention and Object Recognition. *2008 IEEE International Conference on Robotics and Automation Pasadena, CA, USA*, 2008.
- FORSYTH, D. A.; PONCE, J. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.
- FOX, D.; BURGARD, W. THRUN, S. Markov Localization for Mobile Robots in Dynamic Environments. *Journal of Artificial Intelligence Research* v. 11, p.391-427, 1999. 391-427
- FROME, A.; CHEUNG, G.; ABDULKADER, A.; ZENNAROL, M.; WUL, B.; BISSACCO, A.; ADAM, H.; NEVEN, H.; VINCENT, L. Large-scale Privacy Protection in Google Street View. *IEEE 12th International Conference on Computer Vision Issue: ICCV, Publisher*, p. 2373-2380, 2009.
- GERKEY, B.; VAUGHAN, R.T.; HOWARD, A. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In *Proceedings of the 11th International Conference on Advanced Robotics (ICAR 2003)*, p. 317-323, Coimbra, Portugal, June 2003.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*, 3rd Ed. Prentice Hall, 2008.
- GUEAIEB, W.; MIAH, S. A Modular Cost-Effective Mobile Robot Navigation System Using RFID Technology. *Journal of Communications*, vol. 4, n. 2, 2009.
- HARRIS, C.; STEPHENS, M. J. A combined corner and edge detector. *Plessey Research Roke Manor, United Kingdom. The Plessey Company plc.*, 1988.
- HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. McMaster University, Ontario Canada. Prentice Hall, 1999, 842 p.
- HAYKIN, S.; Van VEEN, B.: *Signals and Systems, 2nd Edition*. John Wiley & Sons, 2002.

-
- HEINEMANN, P.; HAASE, J.; ZELL, A. A Combined Monte-Carlo Localization and Tracking Algorithm for RoboCup. *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing, China, 2006.
- HORN, B. K. P.; SCHUNCK, B. G. Determining optical flow: A Retrospective. *Artificial Intelligence*. v. 59, n. 1-2, p. 81-87, 1993.
- HOUGH, P. V. C. Machine Analysis of Bubble Chamber Pictures. *International Conference on High Energy Accelerators and Instrumentation, CERN*, 1959.
- HOWIE, M. C.; LYNCH, K. M.; HUTCHINSON, S.; KANTOR, A.; BURGARD, W.; KAVRAKI, L. E.; THRUN, S. Principles of robot motion: theory, algorithms, and implementation. 2005, 302 p.
- HUYNH, D. Q.; SAINI, A.; LIU, W. Evaluation of three local descriptors on low resolution images for robot navigation. *Image and Vision Computing New Zealand, 2009. IVCNZ '09. 24th International Conference*, p. 113-118, 2009.
- JAFAR, F. A.; SUZUKI, Y.; TATENO, Y. Localization Method for Autonomous Mobile Robot using Visual Features in Environment. *International Journal of Intelligent Information Technology Application*, v. 2, n. 4, p. 169-176, 2009.
- JAFAR, F. A.; SUZUKI, Y.; TATENO, Y.; TABATA, T.; YOKOTA, K. An environmental visual features based navigation for mobile robot in a corridor environment. In *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference*, p. 1612-1617, 2010.
- JÄHNE, B. *Digital Image Processing*. 6th revised and extended. Springer-Verlag New York, Inc., 2005.
- JAIN, A.; HONG, L.; PANKANTI, S. Biometric Identification. *Communications of the ACM*. v. 43, n. 2, p. 90-98, 2000.
- JOLLIFFE, I. T. *Principal Component Analysis*. Springer-Verlag New York, Inc., 2002, 487 p.
- JONES, S. D.; ANDERSEN, C.; CROWLEY, J. L. Appearance Based Processes for Visual Navigation. *IROS '97, IEEE International Conference on Intelligent Robots and Systems, IROS 97*, Grenoble, 1997.
- JUAN, L.; GWUN, O. A Comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)*, v. 3, n. 4, p.143-152, 2010.
- KAESS, M.; DELLAERT, F. Probabilistic Structure Matching for Visual SLAM with a Multi-Camera Rig. *Computer Vision and Image Understanding*, 2009.
- KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, v. 82 (Series D): p. 35-45, 1960.

-
- KEL, Y.; SUKTHANKAR, R. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In *IEEE Computer Society Conference on Image Processing and Pattern Recognition*, p. 506–513, 2004.
- KHUBITZ, O.; BERGER, M. O.; PERLICK, M. Application of radio frequency identification devices to support navigation of autonomous mobile robots. *47th IEEE Vehicular Technology Conference*, 1997.
- KIDONO, K.; MIURA, J.; SHIRAI, Y. Autonomous visual navigation of a mobile robot using a human-guided experience. *Robotic Autonomous Systems*, v. 40, n.2–3, p. 121–130, 2002.
- KIM, D.; NEVATIA, R. Symbolic Navigation with a Generic Map. *Journal Autonomous Robots Archive*, v. 6, n. 1, 1999.
- KORTENKAMP, D.; BONASSO, R. P.; MURPHY, R. *Artificial Intelligence and Mobile Robots*. Massachusetts: The MIT Press, 1988.
- KRAGIC, D.; CHRISTENSEN, H. Survey on Visual Servoing for Manipulation. Tech. Rep. ISRN KTH/NA/P, 02/01 – SE, Jan. 2002, CVAP 259, 2002, 58 p.
- KÜMMERLE, R.; GRISETTI, G.; STACHNISS, C.; BURGARD, W. Simultaneous Parameter Calibration, Localization, and Mapping for Robust Service Robotics. In *Proc. of the IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*. Half Moon Bay, CA, USA, 2011.
- KUS, M. C.; GÖKMEN, M.; ETANER-UYAR, S. Traffic Sign Recognition using Scale Invariant Feature Transform and Color Classification. *23rd International Symposium on Computer and Information Sciences*, ISCIS '08, 2008.
- LAGANIÈRE, R. *OpenCV 2 Computer Vision Application Programming*. Cookbook. Packt Publishing Ltd, 2011, 287 p.
- LAUMOND, J. P. Feasible Trajectories for Mobile Robots with Kinematic and Environment Constraints. *International Conference on Intelligent Systems*, Amsterdam, p. 346-354, 1986.
- LEFEBVRE, T.; BRUYNINCKX, H.; SCHUTTER, J. D. Kalman Filters for nonlinear systems: a comparison of performance. *IEEE Transactions on Automatic Control*, 2001.
- LEONARD, J. J.; DURRANT-WHYTE, H. F. Simultaneous Map Building and Localization for an Autonomous Mobile Robot. *International Workshop on Intelligent Robots and Systems*, IROS'91. Osaka, Japan, 1991.
- LEVIN, A.; SZELISKI, R. Visual Odometry and Map Correlation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'2004), p. 611-618, 2004.

-
- LIN, Y.; GUO, F.; LI, S. Improvement of optical flow in pedestrian detection based on pictorial structure. In *Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference*, v. 3, p. 917-921, 2010.
- LOWE, D. G. Distinctive Image Features from Scale-Invariant Keypoints. *Computer Science Department of University of British Columbia*. Vancouver, B.C., Canada, 2004.
- LOWE, D. G.; SU, S.; LITTLE, J. Vision-based Mobile Robot Localization And Mapping using Scale-Invariant Features. *Computer Science Department of University of British Columbia*. Vancouver, B.C., Canada, 2001.
- LUCAS, B. D.; KANADE, T. An Iterative Image Registration Technique with an Application to Stereo Vision (IJCAI). *7th International Joint Conference on Artificial Intelligence (IJCAI)*, p. 674-679, April, 1981.
- LUH, G.; LIU, W. An immunological approach to mobile robot reactive navigation. *Applied Soft Computing*, v. 8, p. 30-45, 2008.
- MAJUMDAR, A.; WARD, R. K. Discriminative SIFT Features for Face Recognition. *Department of Electrical and Computer Engineering, University of British Columbia*. Vancouver, B.C., Canada, 2009.
- MARTIN, J.; CROWLEY, J. L. Experimental Comparison of Correlation Techniques. *IAS-4, International Conference on Intelligent Autonomous Systems*, 1995.
- MATSUMOTO, Y.; IKEDA, K.; INABA, M.; INOUE, H. Visual navigation using omnidirectional view sequence. *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, 1999. IROS '99*, v. 1, p. 317-322, 1999.
- MATSUMOTO, Y.; MASAYUKI, I.; INOUE, H. Visual Navigation Using View-Sequenced Route Representation. *International Conference on Robotics and Automation (ICRA '96)*, 1996.
- MENDES, C.; WOLF, D. Desvio de Obstáculos Utilizando um Método Estéreo Semi-global. *ENIA - VIII Encontro Nacional de Inteligência Artificial. XXXI Congresso da Sociedade Brasileira de Computação*. Natal (RN), 2011.
- MENEGATTI, E.; PRETTO, A.; PAGELLO, E. Testing omnidirectional vision-based Monte-Carlo Localization under occlusion. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings*. v. 3, p. 2487-2493, 2004.
- MIKOLAJCZYK, K.; SCHMID, C. A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 27, p. 1615-1529, 2005.
- MORAVEC, H.P. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. Robotics Institute, Carnegie Mellon University & Doctoral Dissertation, Stanford University, 1980.

-
- MORAVEC, H.P. Sensor fusion in certainty grids for mobile robots. *AI Magazine* v. 9, p. 61-74, 1988.
- NIXON, M.S.; AGUADO, A.S. *Feature Extraction & Image Processing*, 2nd ed. Academic Press, 2007.
- OLIVA, A.; TORRALBA, A. The role of context in object recognition. *Trends in Cognitive Sciences*. v.11, n.12, p. 520-527, 2007.
- OLLIS, M.; STENTZ, A. Vision-Based Perception for an Automated Harvester Robotics. In *Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference*, v. 3, p. 1838-1844, 1997.
- OSÓRIO, F. S.; PESSIN, G.; WOLF, D. F.; SIMÕES, E. V.; BRANCO, K. C. Simulação Virtual de Carros em Jogos e Aplicações de I.A. *VIII SBGames – Simpósio Brasileiro de Jogos e Entretenimento Digital*, Rio de Janeiro, 2009.
- PARK, S.; HASHIMOTO, S. An approach for mobile robot navigation under randomly distributed passive RFID environment. *IEEE International Conference on Mechatronics, 2009. ICM 2009*, p 1-6, 2009.
- PARKER, C. A.; ZHANG, H. Collective Robotic Site Preparation. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, v. 14, p. 5-19, 2006.
- PESSIN, G. *Evolução de Estratégias e Controle Inteligente em Sistemas Multi-Robóticos Robustos*. Dissertação de Mestrado – PPG em Computação Aplicada da Unisinos: São Leopoldo, RS, 2008.
- PIAGGIO, M.; SGORBISSA, A.; VERCELLI, G.; ZACCARIA, R. Autonomous Robot Navigation Using a Reactive Agents. In *Proc. of the 5th Congress of the Italian Association for Artificial Intelligence on Advances in Artificial Intelligence*, p. 96-105, 1997.
- QUIGLEY, M.; GERKEYY, B.; CONLEYY, K.; FAUSTY, J.; FOOTEY, T.; LEIBSZ, J.; BERGERY, E.; WHEELERY, R.; NG, A. ROS: an open-source Robot Operating System. In *Proceedings of the Open-Source Software workshop at the International Conference on Robotics and Automation (ICRA)*, 2009.
- ROSS, M. G.; OLIVA, A. Estimating perception of scene layout properties from global image features. *Journal of Vision*. v. 10, n. 1, p 1–25, 2010.
- ROSTEN, E.; DRUMMOND, T. Fusing Points and Lines for High Performance Tracking. *IEEE International Conference on Computer Vision*. v. 2, p. 1508–1511, 2005.
- SALES, D.O.; OSÓRIO, F.S.; WOLF, D. F. Topological Autonomous Navigation for Mobile Robots in Indoor Environments using ANN and FSM. In *CBSEC - Conferência Brasileira de Sistemas Embarcados Críticos, 2011, São Carlos. Proceedings of CBSEC - Brazilian Conference on Critical Embedded Systems*. São Paulo : EPUSP, 2011, v. 1. p. 14-19, 2011.

-
- SARIPALLI, S.; MONTGOMERY, J.F.; SUKHATME, G.S. Vision-based Autonomous Landing of an Unmanned Aerial Vehicle. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on Issue*, p. 2799 – 2804, 2002.
- SAXENA, A.; CHUNG, S. H.; NG, A. Y. 3-D Depth Reconstruction from a Single Still Image. *International Journal of Computer Vision*. v. 76, n. 1, Publisher: Springer, p. 53-69, 2007.
- SE, S.; NG, H-K.; JASIOBEDZKI, P.; MOYUNG, T-J. Vision based Mod-eling and Localization for Planetary Exploration Rovers. In *Proceedings of International Astronautical Congress*, Vancouver, Canada, 2004.
- SHEN, J.; HU, H. Visual Navigation of a Museum Guide Robot. In *Proceedings of the 6th World Congress on Control and Automation*, 2006.
- SHINZATO, P.Y. *Sistema de Identificação de Superfícies Navegáveis Baseado em Visão Computacional e Redes Neurais Artificiais*. Dissertação de Mestrado, ICMC - Universidade de São Paulo, USP, Brasil, 2010.
- SIEGWART, R.; NOURBAKHS, I. *Introduction to Autonomous Mobile Robotics*. Cambridge, Massachusetts: MIT Press, 2004.
- SIM, R.; DUDEK, G. Learning generative models of scene features. In *Proc. of IEEE Intl Conf. Computer Vision and Pattern Recognition (CVPR)*, v. 1, p. 406–412, 2001.
- SIM, R.; DUDEK, G. Effective exploration strategies for the construction of visual maps. In *Proc. of 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, v. 3, p. 3224–3231, 2003.
- SIM, R.; ELINAS, P.; GRIFFIN, M.; SHYR, A., LITTLE, J. J. Design and analysis of a framework for realtime vision-Based SLAM using Rao-Blackwellised particle filters. In *Proc. of the 3rd Canadian Conference on Computer and Robotic Vision*, 2006.
- SIM, R.; LITTLE, J. J.: Autonomous vision-based exploration and mapping using hybrid maps and Rao-Blackwellised particle filters. In *Proc. of IEEE Int'l Conf. on Intelligent Robots and Systems (IROS)*, 2006.
- SIMÕES, E. V. Development of an Embedded Evolutionary Controller to Enable Collision-Free Navigation of a Population of Autonomous Mobile Robots. *Ph.D. thesis, University of Kent at Canterbury*, 2000.
- SOKOLOV, S. M; BOGUSLAVSKY, A. A.; KUFTIN, F. A. Vision System for Relative Motion Estimation from Optical Flow. *Journal of Systemics, Cybernetics and Informatics*. v. 8, n. 4, p 65-70, 2010.
- SOUZA, J. R.; PESSIN, G. ; OSORIO, F. S. ; WOLF, D. F. Vision-based Autonomous Navigation Using Supervised Learning Techniques. In *12th Engineering Applications of Neural Networks, 2011, Corfu, Greece. 12th Engineering Applications of Neural Networks*

(EANN). Amsterdam: Springer Verlag : IFIP Advances in Information and Communication Technology, 2011, v. 363. p. 11-20.

STENTZ, A. Optimal and Efficient Path Planning for Unknown and Dynamic Environments. *International Journal of Robotics and Automation*, v. 10, n. 3, 1995.

SZABO, S. Research of local and global navigation methods for autonomous mobile robot with multiprocessor control system. *2003 IEEE International Conference on Industrial Control*, v.1, p. 217-220, 2003.

TAKACS, G.; CHANDRASEKHAR, V.; GELFAND, N.; XIONG, Y., CHEN, W-C.; BISMPIGIANNIS, T.; GRZESZCZUK, R.; PULLI, K.; GIROD, B. Outdoors Augmented Reality on Mobile Phone using Loxel-Based Visual Feature Organization. In *The Proc of the 1ST ACM International Conference on Multimedia information retrieval (MIR '08)*, p. 427-434, 2008.

THRUN, S. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, v. 99, n.1, p. 21-71, 1998.

THRUN, S.; BENNEWITZ, M.; BURGARD, W.; CREMERS, A.; DELLAERT, F.; FOX, D. MINERva: a second generation museum tour-guide robot. In *Proc. of IEEE Int'l Conf. on Robotics and Automation (ICRA)*, v. 3, p. 1999–2005, 1999.

THRUN, S.; FOX, D.; BURGARD, W. *Probabilistic Robotics*. Cambridge, Massachussets. MIT Press, 2005.

THRUN, S.; GUTMANN, S.; FOX, D.; BURGARD, W.; KUIPERS, B.J. Integrating Topological and Metric Maps for Mobile Robot Navigation: A Statistical Approach. *Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, 1998.

TODOROVIC, S.; NECHYBA, M. C.; IFJU, P.G. Sky/ground modeling for autonomous MAV flight. In *Proceedings of ICRA*, p. 1422-1427, 2003.

TÖNNIS, M.; FISCHER, J-G.; KLINGER, G. From Sensors to Assisted Driving – Bridging the Gap. *Journal of Software*, v. 3, n. 3, p. 71-82, 2008.

ULRICH, I.; NOURBAKHS, I. Appearance-Based Place Recognition for Topological Localization. *IEEE International Conference on Robotics and Automation*, San Francisco, CA. p. 1023-1029, 2000.

Van DE SANDE, K. E. A.; GEVERS, T.; SNOEK, C. G. M. Evaluating Color Descriptors for Object and Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 32, n. 9, p. 1582-1596, 2010.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Vision and Pattern Recognition*, v.1, p.511-518, 2001.

-
- VLADUSICH, T.; HEMMI, J. M.; SRINIVASAN, M.V.; ZEIL, J. Interactions of visual odometry and landmark guidance during food search in honeybees. *Journal of Experimental Biology* 208. p. 4123-4135, 2005.
- Von FRISCH, K.; SEELEY, T. D. *The Dance Language and Orientation of Bees*. Belknap Press, Dec 1993.
- WITKIN, A. Scale space filtering. *Proc. Int. Joint Conference Artificial Intelligence., held at Karsruhe, West Germany, 1983, published by Morgan-Kaufmann, Palo Alto, California, 1983.*
- WOLF, D.F.; SUKHATME, G. S. Towards Mapping Dynamic Environments, In *Proceedings of the International Conference on Advanced Robotics*, p. 594-600, 2003.
- WOODEN, D. A guide to vision-based map building. *IEEE Robot. Automata Magazine* v. 13, p. 94–98, 2006.
- WU, P.; KONG, L.; ZHAO, F.; LI, X. Particle filter tracking based on color and SIFT features. In *Audio, Language and Image Processing, 2008. ICALIP*, p. 932-937, 2008.
- WYSZECKI, G.; STILES, W. S. *Color science: Concepts and methods, quantitative data and formulae*, 2nd ed. New York: NY. John Wiley & Sons, 1982.
- YUEN, D. C.; MacDONALD, B. A. Line-based SMC SLAM Method in Environment with Polygonal Obstacles. In *Proc. of the Australasian Conference on Robotics and Automation*, 2003.
- ZHANG, Q.; CHEN, Y.; ZHANG, Y.; XU, Y. SIFT Implementation and Optimization for Multi-Core Systems. *Dept. of Computer Science, Univ. of Science and Technology of China Intel China Research Center, Intel Corporation*, 2008.
- ZHAO, T.; NEVATIA, R. Tracking multiple humans in complex situations. Pattern Analysis and Machine Intelligence, In *IEEE Transactions*, v. 26, n. 9, p. 1208-1221, 2004.
- ZHOU, C.; WEI, Y.; TAN, T. Mobile robot self-localization based on global visual appearance features. In *Proc. of IEEE Int'l Conf. on Robotics and Automation (ICRA)*, p. 1271–1276, 2003.