

8 Conclusões e Trabalhos Futuros

A presente dissertação apresentou uma proposta de algoritmo para exploração e mapeamento completa para robôs móveis na seção 6.1. Ao longo da dissertação os componentes deste algoritmos foram descritos e investigados. O capítulo 7 mostrou alguns experimentos que ilustraram estes componentes, suas capacidades e suas deficiências.

Ao longo do projeto, todos os métodos propostos foram implementados e estudados para diversas configurações possíveis. Entretanto, não foi apresentado aqui a integração dos componentes propostos de modo a se fazer a exploração do ambiente como proposto na seção 6.1. Esta integração fica como proposta para um trabalho futuro.

Serão apresentadas aqui conclusões relativas aos experimentos vistos tal como conclusões gerais observadas ao longo do projeto. Este capítulo está dividido em seções devido ao fato da presente dissertação apresentar uma série de técnicas diferentes. Cada seção apresenta as conclusões relativas ao tópico específico tal como propostas para trabalhos futuros. Por fim são apresentadas conclusões e propostas gerais sobre o presente trabalho.

8.1. Transformações Invariáveis

Nos experimentos apresentados na seção 7.1 foi possível verificar visualmente que a implementação realizada gerou resultados dentro do domínio discreto de imagens bitmap compatíveis com as invariâncias esperadas e verificadas matematicamente para o domínio contínuo. Entretanto, algumas pequenas variações foram visualizadas. Estas variações são consequentes da aplicação ser feita no domínio contínuo e do fato de que algumas transformações (tal como escala) serem feitas sobre as imagens utilizando técnicas de aproximação como interpolação.

As pequenas variações encontradas mostraram que algumas Transformadas possuem grande sensibilidade a determinadas variações na imagem.

Os experimentos apresentados mostraram que o uso das transformações propostas foi adequado para se fazer comparação entre imagens que sofreram variações em escala, rotação e translação, sendo que cada uma das transformadas é mais adequada para algumas destas variações.

Pode-se observar também que o uso de correlação é superior ao uso de distâncias euclidianas para se fazer a avaliação das comparações.

8.2. Window Growing

Não só pelas experiências apresentadas, mas também pela série de testes que foram feitos ao longo do projeto, pode-se perceber que os melhores resultados foram verificados com o uso da Transformada de Fourier Mellin do tipo 1 e da do tipo 2. O uso da comparação por correlação após escala das imagens também demonstrou resultados próximos aos obtidos com as transformações invariáveis.

Porém, as comparações globais utilizadas têm algumas limitações e são extremamente sensíveis. As limitações encontradas são:

- O método não funciona para imagens que possuam objetos a diferentes distâncias do robô e que impliquem em nenhuma imagem centralizada ser parecida com a imagem de referência;
- Para o caso da câmera não estar bem nivelada, a imagem de referência pode não aparecer centralizada quando vista à distância. O método é sensível ao nivelamento da câmera;
- O método não funciona caso ocorram oclusões de áreas na imagem;
- O método não funciona caso apareçam informações visuais diferentes das verificadas na imagem de referência dentro da área centralizada que seria referente à imagem de referência quando o robô está à distância;

Portanto, devido as limitações apresentadas, o uso de comparações locais utilizando o método SIFT mostrou ser uma opção mais adequada dentro do contexto do presente trabalho.

A aplicação das comparações globais apresentadas para auxílio na navegação foi desenvolvida e testada em um primeiro momento. Ao serem observadas suas deficiências optou-se por trabalhar com a técnica SIFT que demonstrou ser uma boa escolha.

8.3. SIFT

Não só dentro dos exemplos apresentados mas durante uma série de testes feitos, a aplicação de descritores SIFT para se encontrar pontos em comum entre imagens se mostrou extremamente eficaz e poderosa. Mesmo em casos onde ocorre oclusão, ou em casos em que se tem objetos a diferentes distâncias dentro da visão do robô, a técnica mostrou não ter encontrado grandes problemas, com poucas exceções.

As dificuldades encontradas foram em se trabalhar com imagens a grandes distâncias onde os objetos vistos variavam muito em escala. Este problema pode ser facilmente resolvido trabalhando-se com resoluções superiores ou então limitando os percursos entre nós durante o processo de exploração.

Apesar da transformação SIFT ser adequada a uma série de aplicações, exige alto custo computacional. Isto implica em ser necessário utilizá-la somente em aplicações mais críticas e trabalhar com técnicas mais simples durante tarefas que necessitam de processamento em tempo real (5 a 10 Hz nos experimentos realizados).

8.4. Panorâmicas

Apesar de o algoritmo não garantir que as imagens sejam alinhadas corretamente, isto só deve vir a ocorrer em imagens com baixa quantidade de informação, sem elementos visuais que se distingam do resto da imagem para que possam ser utilizados ao se fazer o alinhamento. Outro possível caso de problemas de alinhamento é quando a área em comum entre as duas imagens é muito pequena.

Um modo de garantir maior perfeição de alinhamento seria seguir alguma outra estratégia de estimação de parâmetros que não pela pseudoinversa. A

estratégia obtida garante que a transformação levará ao menor erro possível para os pontos de controle obtidos e tem como vantagem baixíssimo tempo de processamento. Existem estratégias diversas de estimação de parâmetros que buscam minimizar o erro entre a imagem transformada e a imagem de referência.

Quando as duas imagens possuem muitos objetos com diferentes profundidades, podem surgir imperfeições de alinhamento devido à paralaxe. Um modo de corrigir estas imperfeições seria trabalhando com transformações locais.

8.5 *Quadtree*

Pode-se verificar que com o uso de parâmetros adequados, a segmentação realmente aponta áreas com maior detalhamento na imagem, ou, com maior quantidade de informação. A técnica demonstrou conseguir detectar regiões com maior quantidade de informação nas imagens sendo viável de ser usada dentro da proposta do presente projeto. É interessante ressaltar que a escolha de regiões com maior quantidade de informação facilita a navegação do robô pois os métodos utilizados ao longo da navegação dependem de áreas com bastante informação para serem efetivos. Isto acontece tanto para o uso de *features* SIFT quanto para a comparação de transformações invariáveis.

Muitas vezes o número de regiões correspondentes a menor divisão *quadtree* é muito pequeno. Quando isto ocorre, independente do método escolhido, os resultados não são muito bons e é de interesse utilizar também outros tamanhos de divisões *quadtree* que não somente o tamanho menor encontrado.

Se forem usadas regiões de segundo menor tamanho ou até mais, pode-se conseguir resultados ainda melhores. Portanto é aconselhável que se faça uma verificação do número de regiões mínimas após o *quadtree* de modo a se determinar quantos níveis de tamanho para as regiões é interessante de se trabalhar.

8.6 Visual Tracking

O método apresentado, apesar de simples, consegue apresentar os seguintes resultados dentro do sistema experimental utilizado:

- O robô consegue fazer o acompanhamento de pontos na tela desde que sua posição não seja alterada por fatores externos;
- Pequenas variações luminosas são compensadas pela técnica;
- O acompanhamento dos pontos observados é correto ao longo da movimentação executada pelo próprio robô;
- A técnica consegue lidar com oclusões temporárias (alguns segundos), mesmo que para toda a imagem;

Apesar de métodos mais complexos poderem ser usados, estes implicariam em perda de eficiência (quanto a velocidade de processamento) e poderiam não ser viáveis para o sistema usado.

8.7 Navegação auxiliada por *Visual Tracking*

Ao longo do projeto a navegação auxiliada por *Visual Tracking* demonstrou ser melhor que a navegação auxiliada por comparações baseadas em transformações invariáveis nos devidos aspectos:

- Maior robustez quanto a variações na iluminação, no posicionamento da câmera (inclinações por exemplo), na composição das cenas observadas, entre outros;
- Maior robustez quanto a oclusões temporárias na imagem. Isto permitiria o uso dos algoritmos propostos em ambientes não estáticos;
- Melhor funcionamento de um modo geral. A navegação auxiliada por comparações baseadas em transformações invariáveis possui sérias restrições que impossibilitam seu uso em determinadas situações;

O uso de descritores SIFT mostrou ser extremamente adequado, permitindo reencontrar os pontos de referência caso o robô por algum motivo não tenha pontos suficientes para se localizar e corrigir sua direção. A restrição em trabalhar

com descritores SIFT está em limitar a distância entre os nós para permitir que pontos sejam encontrados entre as diferentes visões do robô. Esta restrição pode ser balanceada escolhendo uma resolução para as imagens adequada à distância máxima entre os nós desejada.

Outra restrição do uso de SIFT está em seu custo computacional. Este problema foi contornado quando não se trabalhou com SIFT ininterruptamente, ou apenas somente em determinados momentos dos procedimentos propostos.

Quanto à estratégia de *Visual Tracking* proposta, pode-se dizer que esta se mostrou adequada para as tarefas em questão. A configuração apresentada no experimento visto mostrou-se bastante adequada. Mesmo assim ainda são propostas técnicas como Hough e RANSAC que possibilitariam melhora dos resultados com a desvantagem de maior custo computacional. As ferramentas utilizadas possibilitam configurações diversas de modo a adequar o *Visual Tracking* às necessidades encontradas.

Por fim, cabe citar que um problema observado é devido ao fato de que os movimentos mínimos do robô ER1 são grandes e de que o processamento no programa Matlab é relativamente lento, e portanto pode-se ter alguma instabilidade na correção da direção.

Fica a sugestão de se trabalhar com outros métodos de navegação auxiliados por *Visual Tracking*, porém sem deixar de se utilizar descritores SIFT dado que estes demonstraram ser uma boa adição aos métodos propostos.

8.8 Refinando o Modelo Criado Utilizando A.G.

O problema de adicionar novas adjacências a um grafo é tratado através de um Algoritmo Genético gerando bons resultados. Percebe-se que quanto maior o número de nós, maior a dificuldade associada ao problema, e melhores os resultados obtidos pelo Algoritmo Genético em relação à busca aleatória. A função de avaliação utilizada permitiu ainda o uso do A.G. para tentar resolver o problema de escolher a melhor combinação para um número mínimo de adjacências. Porém, os resultados obtidos só foram bons para um número pequeno de nós. Para resolver o problema de um grande número de nós, seria necessário

fazer modificações na função de avaliação ou semear a população inicial com indivíduos válidos.

São propostas diversas sugestões visando resultados ainda melhores, entre elas:

- Uso de *Steady State* sem duplicados;
- Trabalhar com diversos experimentos, semeando novas populações com os melhores indivíduos de populações anteriores;
- Interpolar os parâmetros do A.G.;
- Trabalhar com operadores diferentes dos utilizados;

Verificou-se a possibilidade da aplicação do método implementado não somente na aplicação proposta em robótica mas também no planejamento de rodovias, rotas aéreas ou redes de computadores, como alguns exemplos.

8.9

Conclusões e Propostas Gerais para Trabalhos Futuros

O presente trabalho apresentou uma variedade de técnicas e possibilidades para se fazer a exploração e mapeamento de um ambiente como proposto na seção 6.1. Como já observado, não foi feita a integração das componentes estudadas e desenvolvidas. Esta é portanto a principal proposta para trabalhos futuros de um modo geral.

As componentes desenvolvidas apresentaram bons resultados individualmente demonstrando estarem preparadas para a integração em um trabalho futuro.

Outra proposta que acrescentará muito às implementações feitas é se trabalhar com um sistema experimental mais adequado e menos limitado que o utilizado.

A primeira carência do sistema experimental está no robô utilizado. Como apontado diversas vezes este possui uma série de limitações, dentre as quais:

- Movimento restrito a rotações e trajetórias em linha reta. Não é possível acessar os motores do robô diretamente para se controlar as velocidades individuais de cada roda. Portanto, é impossível descrever trajetórias curvilíneas.

- Não há acesso direto às contagens referentes aos *encoders* de cada roda. Só é possível saber a posição do robô através das coordenadas (x,y) . Deveria ser possível adquirir também a coordenada θ relativa à direção do robô, porém descobriu-se que a API apresenta um problema que não foi nem será resolvido que impossibilita se adquirir esta coordenada (informação dada pelo suporte da empresa *Evolution Robotics*TM).
- A resposta ao comando “*move*” apresenta discrepâncias em relação às distâncias percorridas ou giradas em relação às distâncias pedidas. Isto significa que é necessário estar checando a posição do robô para se corrigir possíveis erros. Porém, isto não é possível de ser feito quando o robô gira porque não é possível se obter a coordenada de giro do robô como já discutido. Estes problemas não deveriam ser esperados já que o robô trabalha com motores de passo e *encoders*, mas infelizmente acontecem.
- O acesso ao feedback do robô é lento e complicado como já descrito anteriormente;
- A resposta aos comandos não é imediata fazendo com que a comunicação com o robô possa ser lenta em vários momentos;

Outra carência do sistema experimental está na implementação feita toda em *Matlab*. Apesar deste ser um programa bastante adequado para experimentos e prototipagem, trata-se de uma linguagem de programação interpretada, e portanto é pouco eficiente em termos computacionais.

Novas etapas para o presente projeto que podem ser desenvolvidas em trabalhos futuros são:

- Uma segunda etapa no projeto envolvendo o refinamento do modelo interno criado podendo trabalhar com técnicas de pré-processamento de imagem e *feature extraction* sobre as imagens obtidas durante a exploração;
- Uma terceira etapa abordaria a navegação do robô após o reconhecimento do ambiente ter sido feito. Isto seria feito apresentando ao robô imagens de objetos ou lugares para o qual este deverá navegar. Neste momento, podem ser aplicadas técnicas de

Redes Neurais e *Feature Extraction* entre outras já citadas para poder se fazer o reconhecimento das imagens apresentadas em seu banco de imagens panorâmicas já criado. Após a identificação do nó que possui a imagem apresentada, o robô deve se auto-localizar e percorrer os caminhos já aprendidos até o nó desejado;

- Adaptar o algoritmo proposto para ambientes não estáticos;
- Explorar características métricas do ambiente durante a exploração;
- Adequar o presente projeto para ambientes não planos;
- Estudar a interação da dinâmica do robô dentro do algoritmo de exploração e navegação proposto;