

Correção de Odometria de Robôs Móveis Usando Visão Computacional

Fernando Paolieri, Carlos Tobar, Eliane Guimarães, Eleri Cardozo, Clésio Tozzi

Faculdade de Engenharia de Computação – Pontifícia Universidade Católica de
Campinas (PUC-Campinas)

Caixa Postal 317 – 13086-900 – Campinas – SP – Brasil

fernandopn@me.com

Abstract. *In this article a project for correcting cumulative odometry errors is presented. The idea is to improve the navigation of mobile robots employing autonomous, teleoperated, or assisted navigation. The resulted system is based in a camera, used as a sensor, and employs complex techniques of image segmentation, description of regions, and mathematic morphology in order to perform pattern matching on synthetic marks placed on the environment. Considering the objective of correcting about 50% of errors, the obtained results show a 70% odometry error correction in closed trajectories, and a 95% orientation error correction.*

Resumo. *Neste artigo, um trabalho para correção de erros de odometria acumulados é apresentado. A idéia é melhorar a navegação de robôs móveis que empregam navegação autônoma, teleoperada ou assistida. O sistema resultante é baseado em uma câmera, usada como sensor, e emprega técnicas complexas de segmentação de imagens, descrição de regiões e morfologia matemática para realizar a tarefa de reconhecimento de padrões em marcas sintéticas espalhadas pelo ambiente. Considerando o objetivo de corrigir erros em 50%, os resultados obtidos mostram 70% de correção de erros de odometria em trajetórias fechadas e 90% de correção de erros de direção.*

1. Introdução

No projeto REAL (*Remotely Accessible Laboratory*) [Cardozo et al. 2009; Moraes et al. 2009], da DRVC (Divisão de Robótica e Visão Computacional) do Centro de Tecnologia da Informação Renato Archer, desenvolve-se infra-estrutura para facilitar a implementação de laboratórios de acesso remoto (*WebLabs*).

Em experimentos com robôs móveis no REAL, é feito o uso freqüente da odometria, que é a estimativa de posição e orientação. Tenta-se obtê-la com um método simples e de baixo custo. Com *encoders* nas rodas, para medir o ângulo referente ao movimento da roda, e com o tamanho da roda do robô, pode-se calcular a distância percorrida, usando-se um integrador, que é um circuito elétrico responsável pelo cálculo da integral de um sinal ou, no caso discreto, a soma de sinais recebidos. Assim, erros pequenos, devidos a situações como piso irregular ou escorregamento da roda, remetem a erros acumulativos, até o ponto que os dados de odometria deixam de ser confiáveis.

Durante o andamento de testes com o robô Pioneer P3DX (Figura 1), foi detectado que, se for solicitado para o robô andar numa trajetória fechada de formato quadrado, após a primeira volta, ele não retornará à mesma configuração de origem, pois existe um erro de posição (X , Y) e um erro de direção (θ). Desta forma, uma estratégia de correção de erros de odometria se faz necessária, dentro de limites aceitáveis.



Figura 1. Robô Pioneer P3DX

O objetivo do projeto descrito neste artigo é o desenvolvimento de um sistema capaz de reduzir erros de odometria (X , Y e θ) em, pelo menos, 50%, em uma trajetória quadrada, usando técnicas de visão computacional.

Como o robô P3DX se encontra em ambiente controlado, este pode ser modificado a fim de se alcançar o objetivo. A proposta de solução consistiu em espalhar marcas em posições conhecidas no ambiente, que pudessem ser detectadas através de uma câmera de bordo do robô e, a partir de um processo de visão computacional, determinar a posição do robô em relação a um sistema de referência global.

Escolheu-se apresentar, a seguir, os seguintes resultados do trabalho: na seção seguinte está a arquitetura do sistema proposto; na terceira seção está uma sequência exemplo do funcionamento da correção projetada; e na última seção encontram-se dados da avaliação realizada e as considerações finais.

2. Arquitetura do Sistema Proposto

O sistema para correção de odometria baseada em visão foi organizado em módulos visando um fraco acoplamento aos periféricos que utiliza, como a câmera e os atuadores do robô. A modelagem é representada no digrama de arquitetura ilustrado na Figura 2.

O módulo de aquisição de imagens é um módulo de fronteira, que se comunica com uma câmera Axis, por meio de um protocolo proprietário fornecido pelo fabricante.

No sistema é utilizado o método de adquirir imagens da câmera, no formato do NumPy, que é uma ferramenta para manipulação de matrizes em Python. O formato da matriz é dado por d , um número de canais de cores, l , um número de linhas, e c , um número de colunas. l e c definem a resolução, informada no momento da aquisição.

O sistema utiliza uma base de dados SQLite e o Django, que é um servidor de aplicação desenvolvido em Python, segundo o modelo MVC. O módulo API robô Pioneer P3DX se comunica com o robô utilizando o protocolo definido pelo projeto REAL e foi implementado para ser genérico.

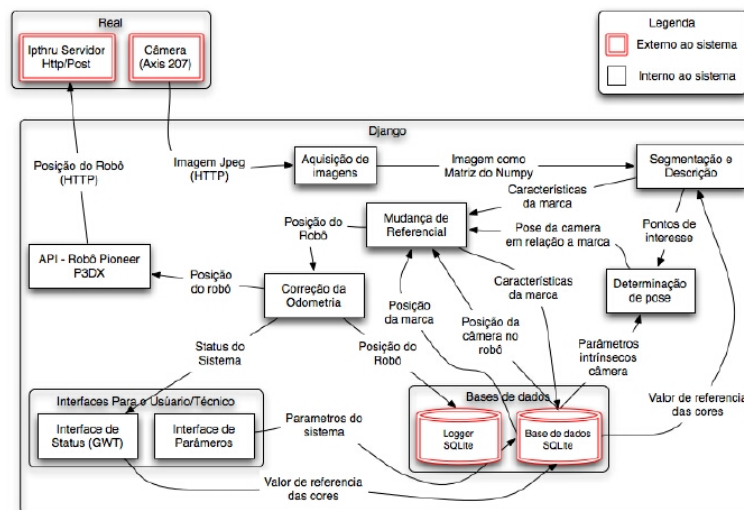


Figura 2. Diagrama de arquitetura

A funcionalidade do restante dos módulos é apresentada a seguir, através de um exemplo de como ocorre o processamento de uma imagem capturada pela câmera.

3. Funcionamento da Correção de Odometria Baseada em Visão

A solução proposta envolve atividades para processamento de imagens, dentre elas: segmentação, determinação de pose e a mudança de referencial. Além disso, os módulos de correção de odometria e as interfaces de parâmetros e de acompanhamento.

3.1. Segmentação

A primeira atividade é o rastreamento de uma marca, ou seja, a segmentação de uma imagem, que consiste em realçar pontos de interesse nela. No caso, é aplicado um limiar de corte, um filtro morfológico de abertura, a fim de retirar ruídos da imagem [Dougherty; Lotufo 2003], e um algoritmo de representação de regiões [Gonzalez; Woods 2008]. Após se obter a representação da marca, é determinada qual é essa marca e é extraída de uma base de dados a sua posição.

Usando a imagem ilustrada na Figura 3 como entrada para o módulo de segmentação e descrição, é necessário traçar um objetivo sobre o que deve ser segmentado. No caso são círculos e seus arcos. Como a informação de cor não é importante, é interessante transformar as cores em escalas de cinza (Figura 4).

A transformação de uma imagem colorida para uma imagem com níveis de cinza se dá pelo cálculo da intensidade (componente do modelo HSI) de cada *pixel*, que é calculada pela média das três componentes de cor. Portanto, tendo um *pixel* de intensidade (r, g, b) , o seu valor em nível de cinza i é dado pela média aritmética de r , g e b , que representam os níveis de vermelho, verde e azul. O modelo HSI representa as cores pela sua matiz (*Hue*), sua saturação (*Saturation*) e sua intensidade (*Intensity*).

Com uma imagem em preto e branco, é possível perceber que o contraste entre o círculo e o papel em que foi impresso é grande. Por isso, é simples conseguir realçar os círculos na imagem, aplicando um limiar, que consiste em separar seus *pixels* em dois grupos: valores abaixo de uma constante c e valores acima desta. O resultado é uma

imagem binária [Gonzalez; Woods 2008].



Figura 3. Imagem adquirida

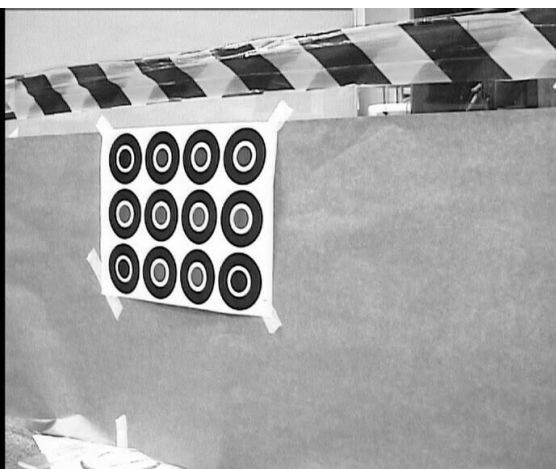


Figura 4. Figura 3 em escala de cinza

Como a limiarização é aplicada *pixel a pixel* e cada valor é mapeado em outro valor, pode-se dizer que esta é uma transformação de mapa de cores. No caso, o limiar aplicado foi $c = 150$ e foi gerada uma imagem, como ilustrado na Figura 5 que está com muito ruído. Para reduzir esse ruído, foi usada uma filtragem de abertura morfológica com círculo de raio 3 como elemento estruturante, cujo resultado aparece na Figura 6.

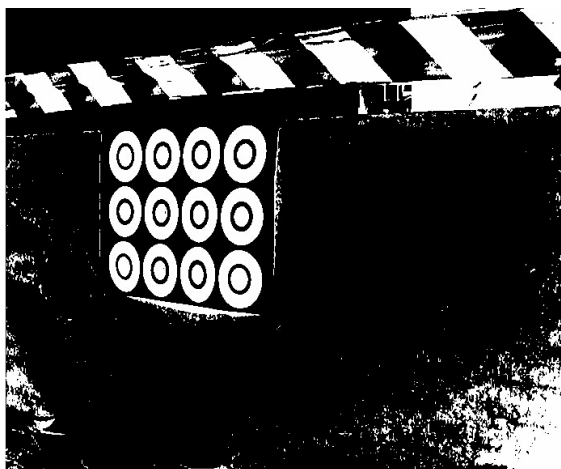


Figura 5. Figura 4 limiarizada

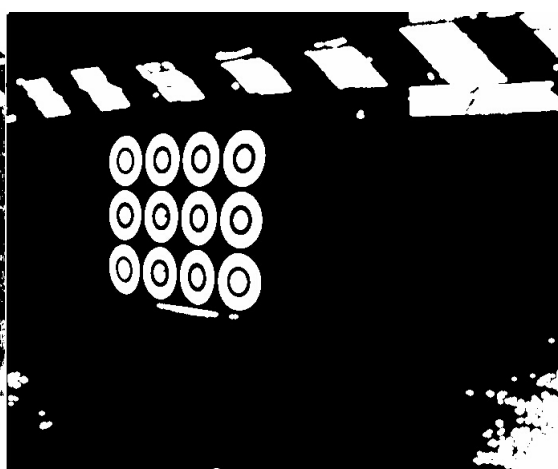


Figura 6. Figura 5 filtrada

A abertura morfológica consiste na aplicação de dois filtros, um deles é a erosão e outro a dilatação [Dougherty; Lotufo 2003]. Durante uma erosão, o elemento estruturante é colocado sobre a imagem em todas as posições possíveis. Se todos os *pixels* do elemento estruturante coincidirem com *pixels* da imagem, a posição do centro do elemento estruturante será marcada na imagem resultante. A dilatação realiza a atividade inversa da erosão, ou seja, em torno de cada pixel da imagem tenta-se identificar todos os pixels do elemento estruturante.

Após a filtragem, existem várias regiões segmentadas e nem todas pertencem a uma marca. Para distinguir quais pertencem e quais não pertencem, procura-se encontrar todos os centróides na imagem, considerando branco o objeto e preto o fundo. Com isto, conseguem-se as coordenadas de todos os centróides, ilustrado na Figura 7.

O alvo escolhido para compor uma marca é um círculo com anel ao redor. Esta escolha se deu pelo fato de que ele compõe duas regiões distintas com seus centróides próximos. Eles só não coincidem por problemas de discretização de *pixels*. Para filtrar o resultado obtido na Figura 7, foi feito o cálculo da distância euclidiana entre os pontos, desde que, havendo dois pontos com distância euclidiana menor que 5 *pixels*, os dois pontos se mantêm. Como resultado, o ruído foi filtrado, como ilustra a Figura 8.

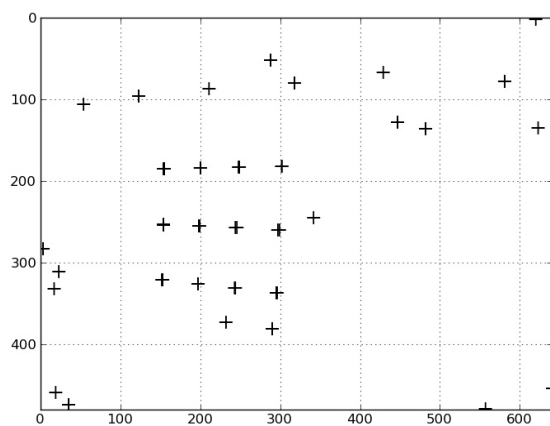


Figura 7. Centróides da Figura 6

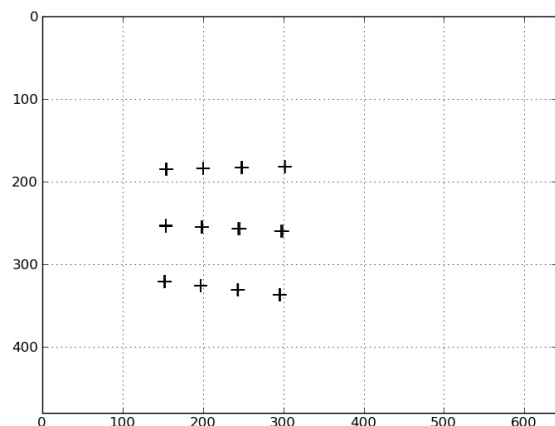


Figura 8. Centróides filtrados

Uma marca é composta por doze círculos. Cada círculo com uma circunferência colorida ao seu centro. As cores caracterizam a marca e fornecem a sua assinatura, que é o conjunto das cores da esquerda para a direita de cima para baixo. A distância entre os círculos de é de 68mm. Como existem dois conjuntos de coordenadas de centróides para cada marca (as duas regiões circulares) e sabe-se que a marca possui três linhas e quatro colunas, é feita uma regressão linear para encontrar sete retas, onde a interseção delas leva a um valor confiável do centro de cada círculo, como mostra a Figura 9.

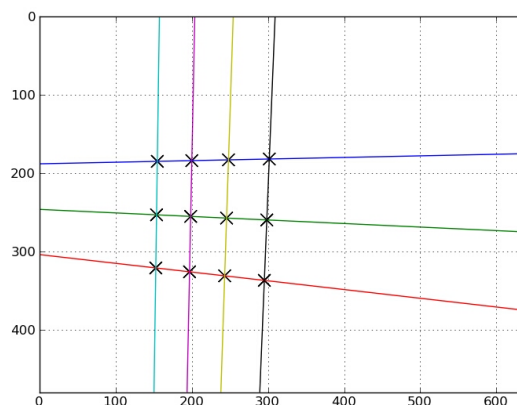


Figura 9. Retas calculadas a partir dos centróides

As coordenadas do centro de cada círculo são a saída do módulo de segmentação e descrição, bem como a assinatura da marca.

O reconhecimento de cores é feito com o modelo RGB. Para isso, é necessário ter uma referência da cor, no cubo RGB, para que seja possível calcular a distância euclidiana entre a cor aferida e a cor de referência. O resultado é a similaridade entre duas cores, ou seja, quanto menor a distância euclidiana, mais semelhantes são as cores.

3.2. Determinação de Pose

A segunda atividade é a determinação de pose. Ela consiste de um método onde, a partir de pontos conhecidos do espaço do objeto, utilizando o conhecimento do formato da marca, e seus equivalentes no espaço imagem, detectados na segmentação, é possível encontrar o modelo de câmera, que inclui os dados relativos à pose da câmera [Shapiro; Stockman 2008]. A técnica escolhida é baseada no difundido método de calibração de Tsai [1987], para o qual utilizou-se um algoritmo baseado na resolução de regressões lineares, que converte os pontos dados como entrada, vindos do módulo de segmentação e descrição, para um sistema de referência. Também converte o sistema de referência de imagem para o sistema de referência de câmera.

Como a marca e a distância entre os círculos são conhecidos, são determinadas as coordenadas dos pontos no sistema de referência global. Assim, é possível calcular os vetores de rotação e a posição do sistema de referência global e projetar seus eixos como ilustra a Figura 10.

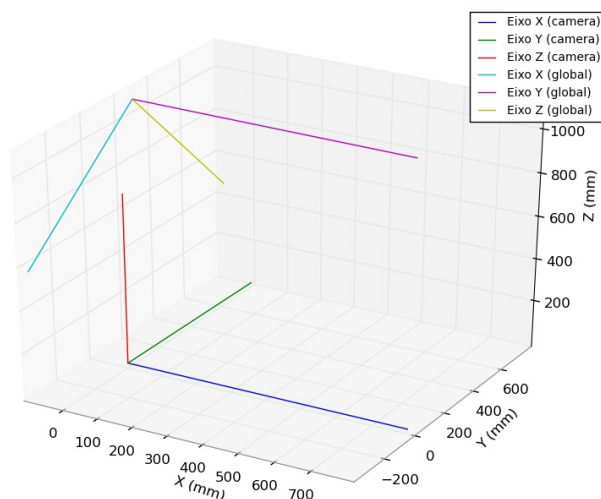


Figura 10. Sistema de referência global e de câmera

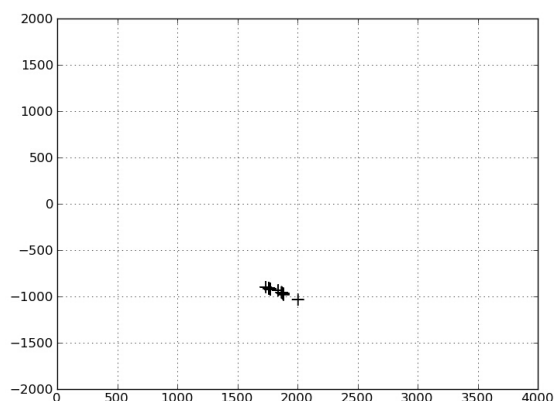


Figura 11. Coordenadas definidas de imagens tiradas da mesma posição

Em um teste utilizando uma seqüência de fotos tiradas do mesmo local, olhando para a mesma marca, foi possível gerar um gráfico que ilustra a variação das respostas

obtidas pelo sistema. Este gráfico é ilustrado na Figura 11. A variação é dada pelo problema de discretização dos *pixels*, causando uma mudança no centróide.

O algoritmo implementado tem duas limitações: o ponto mais baixo e mais à esquerda é utilizado como origem do sistema de referência absoluto e não pode estar próximo ao eixo óptico (eixo Z do sistema de referência da câmera); e a marca não pode estar paralela ao plano imagem. Nestes dois casos, o módulo retorna um aviso ao usuário. Este problema se dá por inconsistências numéricas previstas no artigo de Tsai.

A saída do módulo de determinação de pose é a pose do robô, ou seja, sua posição no espaço em relação à marcação e sua direção nos três eixos.

3.3. Mudança de Referencial

As marcas são cadastradas em relação a um sistema de referência chamado de sistema de referência global, que é determinado no momento em que as marcas são cadastradas.

O módulo de determinação de pose retorna a posição em relação à marca, mas não em relação ao sistema de referência global. Para determinar a posição do robô são necessárias:

- Uma transformação para que posição do robô em relação à marca seja determinada a partir da posição da câmera em relação à marca, pois a posição da câmera em relação ao centro do robô é conhecida.
- Uma transformação para que a posição do robô seja determinada a partir da posição da marca em relação ao sistema de referência global, pois a posição e orientação da marca são conhecidas.

3.4. Correção de Odometria

O módulo de correção de odometria tem três funções simples:

- Gerar uma entrada em um *log* com o status do sistema e com a posição do robô.
- Fazer uma chamada na API de controle do robô para informar a nova posição.
- Informar à interface do usuário o status do sistema.

Como o robô nem sempre estará parado durante o processo de correção de odometria, foi adotada a estratégia de adquirir a posição inicial e final do robô a cada ciclo de execução, para se ter uma estimativa da movimentação do robô durante o processamento de imagens. Assim, o resultado é somado com a estimativa de movimento determinado a partir da diferença entre as posições inicial e final adquiridas.

3.5. Interfaces de Configuração

A interface que trata as marcas (Figura 12), possibilita a inclusão, exclusão e edição de marcas, cujo cadastro é feito fornecendo-se as suas assinaturas e posições.

A interface de parâmetros (ilustrada na Figura 13) possibilita ao usuário a alteração de parâmetros do sistema, como endereço lógico do robô, endereço lógico da câmera, posição do robô, centro óptico e tamanho do *pixel*.

Signature	PositionX	PositionY	PositionZ	PositionTh
BRRGGGGBRR	2130.0	0.0	2930.0	270.0
DBBGGGGBRR	0.0	0.0	0.0	0.0
DCBBGGGBRR	-2090.0	0.0	1400.0	90.0
DCRBDGRBGRB	-950.0	0.0	4270.0	180.0
DRRRGGGGBBD	730.0	0.0	0.0	0.0

Figura 12. Cadastro de marcas

Name	Value
Ox	240.82206839010087
Oy	391.83946554308886
CamURL	http://192.168.0.101:8080/
sizeYMark	68
sizeXMark	68
PsizeX	4.624277456647e-03
PsizeY	4.615384615385e-03
CamPosX	0
CamPosY	200
CamPosZ	-100
RobotURL	http://10.1.22.100:4950/

Figura 13. Alteração de parâmetros

3.6. Interface de Acompanhamento

A interface de acompanhamento (ilustrada na Figura 14) tem como funcionalidade básica iniciar e parar o serviço, além de receber informações sobre o andamento do sistema. Há a possibilidade de visualizar a imagem gerada pelo módulo de segmentação e descrição, a fim de verificar se o sistema está conseguindo encontrar a marca.

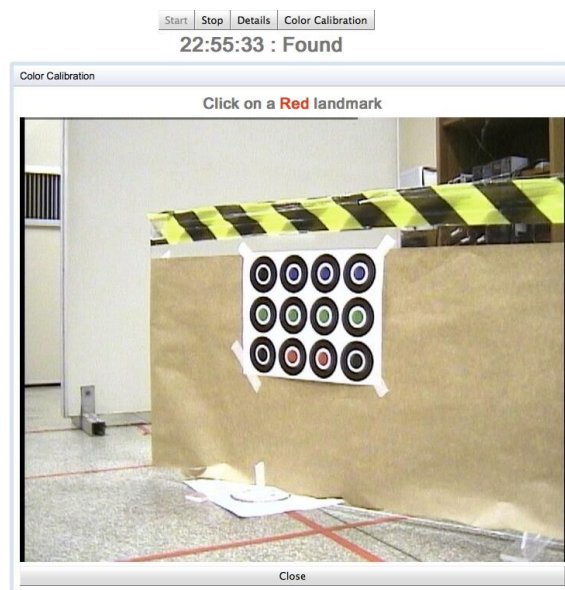


Figura 14. Interface de acompanhamento

4. Conclusão

O sistema de correção de odometria de robôs móveis descrito neste artigo objetiva corrigir erros em 50% nas três coordenadas, em X, Y e θ . Para isso, faz uso de técnicas de visão computacional para determinar a posição do robô a partir do rastreamento de marcas sintéticas, espalhadas em um ambiente em posições previamente conhecidas.

Para avaliar o sistema, foi realizado um experimento de dez ciclos sem o sistema e dez ciclos com o sistema. Um ciclo consistiu de três voltas em uma trajetória quadrada de lado 1,8 metros, em um ambiente com quatro marcas como ilustra a Figura 15. Após cada ciclo foram determinadas a posição e a direção finais do robô.

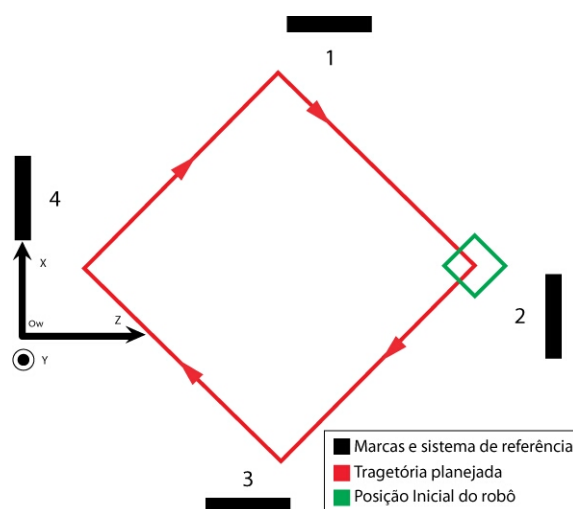


Figura 15. Posição das marcas e a trajetória programada

Após cada ciclo, foi medido o erro da posição do robô com a posição marcada no chão onde ele deveria estar. Os resultados obtidos estão nas Tabelas 1 e 2. As medições nestas tabelas foram feitas com o auxílio de uma fita métrica e de um transferidor.

Tabela 1. Resultados obtidos sem o sistema

Nº	Erro em X (cm)	Erro em Z (cm)	Erro em θ (graus)
1	56,0	35,0	30,0
2	60,0	44,0	30,0
3	63,0	40,0	31,0
4	62,0	35,0	40,0
5	32,0	39,0	10,0
6	57,0	37,0	31,0
7	60,0	45,0	32,0
8	65,0	40,0	33,0
9	64,0	34,0	38,0
10	33,0	40,0	9,0
Erro médio	55,2	38,9	28,4
Desvio Padrão	12,30	3,72	10,51

É possível notar uma diferença significativa entre o erro médio sem o sistema e com o sistema, realizando o cálculo da porcentagem do fator de correção dado pela fórmula $p = 1 - (Mc / MS)$, onde Mc é o erro médio com o sistema e Ms é o erro médio sem o sistema e p é o fator de correção do sistema. O fator de correção em cada um dos eixos foi de 54% para X, 61% para Z e 95% para θ .

Os resultados são bem expressivos e confirmam que o objetivo foi alcançado com sucesso nos três parâmetros que compõem a pose do robô.

Entre as dificuldades enfrentadas, encontra-se a limitação na resolução da câmera. A versão final do sistema de correção de odometria trabalha com imagens de 640x480 e, por isso, uma marca só é identificável a dois metros de distância. Para qualquer distância maior, o sistema ignora a marca, interpretando-a como um ruído.

Tabela 2. Resultados obtidos com o sistema

Nº	Erro em X (cm)	Erro em Z (cm)	Erro em θ (graus)
1	30,0	12,0	0,0
2	38,0	15,0	0,0
3	28,0	13,0	0,0
4	7,0	0,0	0,0
5	26,0	36,0	5,0
6	30,0	11,0	1,0
7	36,0	16,0	2,0
8	26,0	14,0	0,0
9	5,0	0,0	0,0
10	27,0	36,0	6,0
Erro médio	25,3	15,3	1,4
Desvio Padrão	10,94	12,30	2,27

Outra dificuldade advinda de se trabalhar com a odometria de um robô é a sua mobilidade. Pois, do momento que uma imagem foi adquirida até o momento em que esta é processada, houve movimentação do robô que tem que ser considerada.

Entre diversas melhorias que podem ser implementadas, salienta-se trabalhar com uma integração de sensores, que combinem a odometria implementada com dados de um sonar ou de outros sensores.

Referências

- CARDOZO, E. et al. (2009) Realabs-bot: a weblab in mobile robotics over high speed networks. "IFAR Workshop on Networked Robotics (NetRob 2009)", Colorado.
- DOUGHERTY, E. R.; LOTUFO, R. A. (2003) "Hands-on Morphological Image Processing". Washington: Spie Press.
- GONZALEZ, R. C.; WOODS, R. E. (2008) "Digital image processing". 3. ed. New Jersey: Prentice Hall.
- MORAES, D. et al. (2009) A network architecture for large mobile robots environments. "Second International Conference on Robot Communication and Coordination (ROBOCOMM 2009)", Odense.
- SHAPIRO, L. G.; STOCKMAN, G. C. (2008) "Computer Vision". [S.l.]: Prentice Hall.
- TSAI, R. (1987) A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. "IEEE Journal of Robotics and Automation", v. 3, n. 4, p. 323-344.