

Marciano Prates Salbego

# **Sistema de Calibragem de sensores laser do tipo LIDAR**

**Brasil**

**14 de Agosto de 2014**

Marciano Prates Salbego

## **Sistema de Calibragem de sensores laser do tipo LIDAR**

Trabalho de Conclusão de curso com o objetivo de obtenção do título de Bacharel em Engenharia Elétrica.

Universidade Federal do Pampa

Curso de Engenharia Elétrica

Orientador: Marcio Stefanello

Brasil

14 de Agosto de 2014

---

Marciano Prates Salbego

Sistema de Calibragem de sensores laser do tipo LIDAR/ Marciano Prates  
Salbego. – Brasil, 14 de Agosto de 2014-  
55 p. : il. (algumas color.) ; 30 cm.

Orientador: Marcio Stefanello

Trabalho de conclusão de curso (TCC) – Universidade Federal do Pampa  
Curso de Engenharia Elétrica  
, 14 de Agosto de 2014.

1. Palavra-chave1. 2. Palavra-chave2. I. Orientador. II. Universidade Federal  
do Pampa. III. Curso de Engenharia Elétrica. IV. Título

CDU 02:141:005.7

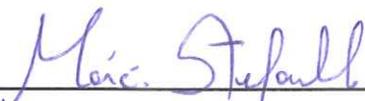
---

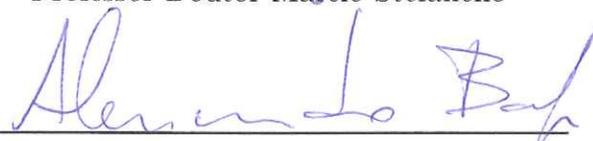
Marciano Prates Salbego

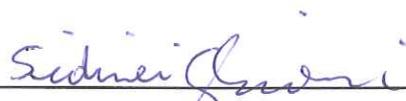
## Sistema de Calibragem de sensores laser do tipo LIDAR

Trabalho de Conclusão de curso com o objetivo de obtenção do título de Bacharel em Engenharia Elétrica.

Trabalho aprovado em 14 de Agosto de 2014.

  
\_\_\_\_\_  
Professor Doutor Marcio Stefanello

  
\_\_\_\_\_  
Professor Mestre Alessandro Bof de Oliveira

  
\_\_\_\_\_  
Professor Doutor Sidinei Ghissoni

Brasil

14 de Agosto de 2014

# Agradecimentos

Agradeço inicialmente à minha família, minha mãe Ivone, meu irmão Marcos e o meu pai Dejacir pelo carinho, compreensão e suporte durante toda a minha vida e em especial durante a graduação.

Agradeço em especial à minha namorada Marcela Perpétuo, pelo carinho, paciência e companheirismo durante o período em que estivemos longe para que eu pudesse realizar este trabalho.

Agradeço ao pessoal do Laboratório: Sérgio, Conrado, Marcelo, Roger, ao professor Márcio Stefanello, pelo suporte e dedicação na orientação deste trabalho, aos professores da Engenharia Elétrica, os funcionários da Direção e Administração, e a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

*"Damn the torpedoes, full speed ahead! "*  
*(David Farragut)*

# Resumo

Este trabalho apresenta a descrição do desenvolvimento de um sistema de calibragem para sensores laser do tipo LIDAR. A calibragem dos sensores utilizados em sistemas de localização e navegação de robôs é uma importante etapa do processamento dos dados lidos, visto que é este um dos sistemas responsáveis pela confiabilidade das leituras obtidas. Ao longo do trabalho apresenta-se as técnicas empregadas para a simulação do ambiente e do comportamento do sensor, bem como os algoritmos desenvolvidos para a identificação e calibragem dos sensores durante a sua operação. O sistema realizado é capaz de distinguir perturbações na posição e orientação angular do sensor de perturbações devido à incerteza presente na medida. A atuação do sistema de calibragem é realizada em termos da atualização do conhecimento da posição do sensor após a perturbação.

**Palavras-chaves:** LIDAR; Calibragem de Sensores; Algoritmos de Registro de Superfície

# Abstract

This work presents the development of a LIDAR laser sensor calibration system. Calibration of LIDAR sensors is an important aspect of the data processing stage in robotic systems of localization and navigation because it asserts the reliability of each measurement regarding the sensor referential. In the development of the text, it is presented the employed techniques for map creation and sensor virtualization. Environment and sensor simulation techniques are explored throughout the work, as well as the developed calibration algorithms. The system designed is capable of distinguish perturbations of sensor position from uncertainties of the measurement. The actuation process of the system is given in terms of the update of the knowledge of the sensor position after a perturbation.

**Keywords:** LIDAR; Sensor Calibration; Scan Matching Algorithms

# Lista de ilustrações

Figura 1 – Esquemático do funcionamento do sensor visto na primeira imagem em perspectiva, na segunda em corte de planta e na terceira apresentando os dados medidos. . . . .	12
Figura 2 – Foto do modelo do sensor laser LIDAR <i>Hokuyo URG 04LX</i> . . . . .	13
Figura 3 – Esquema de representação da instalação das duas câmeras estéreo fixas no ambiente. Retirado de (SAPUTRA et al., 2012) . . . . .	15
Figura 4 – Representação do mapa de um dos ambientes do ITER com os sensores LIDAR instalados. Adaptado de (FERREIRA; VALE; VENTURA, 2013) . . . . .	16
Figura 5 – Problema de localização sujeito à incerteza na posição do sensor. . . . .	17
Figura 6 – Sensor Lidar instalado no ambiente. O polígono em vermelho representa a sua área de cobertura e os pontos discretos as medições. . . . .	18
Figura 7 – Perturbação aplicada ao sensor apresentado na Figura 6. A posição do sensor foi alterada para a direita, resultando em um novo conjunto de medidas do ambiente. . . . .	19
Figura 8 – Esquema representando um sistema robótico móvel genérico. . . . .	20
Figura 9 – Tela do software <i>URG-Viewer</i> com leitura do sensor Laser. O diagrama polar possui variação angular de $240^\circ$ . . . . .	22
Figura 10 – Representação em planta baixa de um ambiente hipotético e a instalação do sensor LIDAR em duas posições distintas . . . . .	23
Figura 11 – Leitura do sensor instalado na Posição #1. . . . .	24
Figura 12 – Leitura do sensor instalado na Posição #2. . . . .	24
Figura 13 – Problema geométrico de um ponto orientado por um vetor $u_1$ em uma reta. . . . .	25
Figura 14 – Representação gráfica do procedimento de geração de leituras sintéticas do sensor. . . . .	26
Figura 15 – Particionamento do vetor de leituras geradas em dois vetores, um à esquerda da orientação central e outra a direita. . . . .	27
Figura 16 – Medidas presentes em um mesmo segmento de reta. . . . .	28
Figura 17 – Versão acelerada do algoritmo de geração de leituras do sensor. . . . .	29
Figura 18 – Processo de sobreposição de duas imagens utilizando descritores invariantes e uma transformação não-rígida. Retirado de (ZITOVÁ; FLUSSER, 2003). . . . .	32
Figura 19 – Nuvens de pontos $\mathbf{p}_i$ (azul) e $\mathbf{p}'_i$ (verde). . . . .	33
Figura 20 – Métrica de associação entre as duas nuvens de pontos $\mathbf{p}_i$ e $\mathbf{p}'_i$ utilizando a abordagem <i>point-to-line</i> . O ponto em verde indica o ponto $\Pi_i\{S_{ref}, \mathbf{p}'_i\}$ . Modificado de (CENSI, 2008). . . . .	37

Figura 21 – Sensor laser alocado em um ambiente. Em vermelho os pontos medidos	41
Figura 22 – Deslocamento na posição e orientação angular do sensor. . . . .	42
Figura 23 – Nuvens de pontos do sensor nas duas posições. . . . .	42
Figura 24 – Sobreposição do campo de visão do sensor na situação de pré e pós perturbação. . . . .	43
Figura 25 – Diagrama de blocos do sistema desenvolvido. . . . .	44
Figura 26 – Interface da aplicação desenvolvida. . . . .	45
Figura 27 – Estratégia para a simulação de perturbações no sensor. . . . .	46
Figura 28 – Não sobreposição do campo de visão do sensor na posição inicial e após a perturbação. . . . .	47
Figura 29 – Distribuição do número de iterações para o cenário 1. . . . .	49
Figura 30 – Distribuição do número de iterações para o cenário 2. . . . .	49
Figura 31 – Distribuição do número de iterações para o cenário 3. . . . .	49
Figura 32 – Distribuição acumulado do número de iterações para os três cenários. .	50

# Sumário

0.1	Representação do ambiente . . . . .	16
0.2	Calibragem de sensores . . . . .	17
0.3	Objetivos . . . . .	20
0.4	Estrutura do Trabalho . . . . .	20
1	<b>MODELAGEM DO SENSOR . . . . .</b>	<b>21</b>
1.1	Leituras típicas dos sensores LIDAR . . . . .	22
1.2	Simulação de leituras do sensor . . . . .	23
1.3	Estimativa da variância do ruído . . . . .	30
2	<b>ALGORITMOS DE REGISTRO DE SUPERFÍCIE . . . . .</b>	<b>31</b>
2.1	Algoritmo <i>Iterative closest Point (ICP)</i> . . . . .	32
2.2	Formulação do problema de minimização . . . . .	34
2.2.1	Algoritmo de Arun . . . . .	34
2.2.2	Derivação do algoritmo Arun . . . . .	35
2.3	<b>ICP point-to-line . . . . .</b>	<b>36</b>
2.3.1	Derivação do algoritmo pIIICP . . . . .	38
2.3.2	Definição do algoritmo . . . . .	40
3	<b>SISTEMA DE CALIBRAGEM DE SENSORES . . . . .</b>	<b>41</b>
3.1	Sistema Computacional . . . . .	44
3.2	Noções sobre Convergência . . . . .	46
3.3	Estudo de Caso . . . . .	47
	Conclusão . . . . .	51
	Referências . . . . .	54

# Introdução

Este trabalho apresenta a realização de um sistema para a calibragem de sensores laser do tipo LIDAR (acrônimo para *Light Detection and Range*). Sensores deste tipo estão presente em inúmeras aplicações em Robótica, em especial aquelas em que necessita-se somente de informação de uma fatia do ambiente em que o sensor está inserido.

Sensores Lasers são dispositivos ópticos de medida que calculam a distância de um objeto a partir do tempo de reflexão de um feixe de luz emitido de maneira direcional. Usualmente contam com um motor de passo, que faz com que o mecanismo emissor de luz possa variar sua referência angular. Esta variação angular permite a obtenção de um conjunto de medidas em um intervalo angular especificado.

Estes dispositivos são essenciais em aplicações que demandem medidas precisas e com um grande alcance. Quando comparado com sensores do tipo sonar, os sensores laser são muito mais robustos quanto à variabilidade das propriedades da superfície e com respeito ao seu alcance. O seu funcionamento pode ser explicado pela Figura 1 (a). O feixe laser é emitido a partir de uma unidade montada em um sistema rotacional. Este sistema rotacional é composto de um motor de passo que gira em passos angulares fixos ao longo de um intervalo angular (modelos comerciais apresentam coberturas angulares usuais entre  $240^\circ$  e  $360^\circ$ ). A informação de reflexão do feixe emitido é obtida por um fotossensor e processada internamente para verificação do tempo necessário para a luz atingir o objeto e retornar até o sensor. A Figura 1 b) mostra o sensor instalado em um ambiente composto por quatro paredes e um objeto circular localizado na posição inferior direita. O primeiro feixe laser é emitido pelo sensor, resultando na primeira medida, representada aqui como o primeiro ponto azul da Figura 1 (c). Após a primeira medida o sistema rotórico move a unidade óptica por um incremento angular  $\delta\theta$ . Uma nova medida é realizada na nova posição, representada pelo segundo ponto na Figura 1 (c). O processo continua até o sensor cobrir toda a região do seu alcance.

A figura 2 apresenta um modelo comercial de sensor LIDAR (*Hokuyo URG 04LX*). A unidade de emissão de luz fica contida na cápsula superior, bem como o seu dispositivo de rotação. As especificações deste modelo são: Frequência máxima de amostragem de  $10Hz/scan$ , cobertura de área de  $240^\circ$  com passo de  $0.36^\circ$  e range de detecção entre  $20mm$  até  $5600mm$ .

No contexto de aplicação destes sensores, pode-se destacar aplicações em duas classes importantes: O mapeamento e localização simultâneos de robôs móveis (do inglês *SLAM - Simultaneous Localization and Mapping*) e a localização de veículos autônomos em ambientes previamente conhecidos.

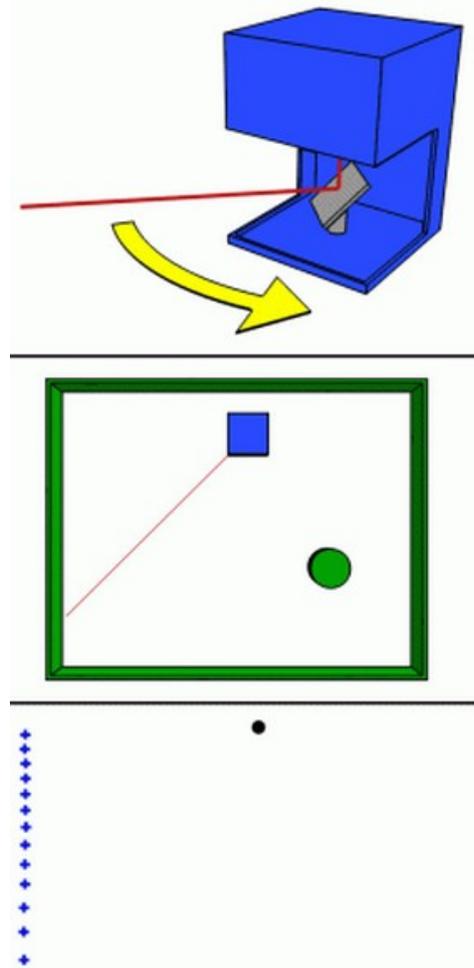


Fig. 1 – Esquemático do funcionamento do sensor visto na primeira imagem em perspectiva, na segunda em corte de planta e na terceira apresentando os dados medidos.

Com respeito à primeira classe de aplicação mencionada, inúmeros trabalhos abordam a utilização dos sensores para a estruturação de um ambiente desconhecido. Dentre estes trabalhos, é valoroso destacar o trabalho de (WANG et al., 2013), onde um sistema de mapeamento foi realizado a partir de um arranjo de sensores LIDAR com cobertura espacial de 360°. Após a obtenção dos dados do sensor, um processo de extração de propriedades (neste caso, objetos colunares como árvores postes, etc) foi realizado a fim de simplificar o conjunto de dados obtidos. Em (MOGHADAM; WIJESOMA; FENG, 2008), as leituras do sensor laser são fundidas com as informações de uma câmera estéreo montada no veículo, permitindo a navegação e o mapeamento deste em ambientes muito mais complexos.

Considerando a segunda classe de aplicações descrita acima, pode-se destacar a utilização dos sensores como fonte de informação sobre a posição de veículos autônomos em ambientes internos. Neste tipo de ambiente restrito, usualmente têm-se o conhecimento do mapa de maneira bem definida (e.g. A localização de um robô dentro de um prédio). Além disso, é razoável definir que o veículo em questão irá operar sempre dentro dos limites do



Fig. 2 – Foto do modelo do sensor laser LIDAR *Hokuyo URG 04LX*.

prédio. Desta forma, cria-se um cenário de atuação contido, tornando possível o uso de algumas ferramentas, como é o caso da utilização dos sensores de detecção e medida fixos no ambiente e não embarcados no próprio veículo.

Em ambientes internos, a utilização dos sensores LIDAR pode ser feita em três arranjos de configurações:

- Instalação no veículo: Neste tipo de instalação faz-se uso do sensor montado diretamente no veículo, onde a informação de distância é referenciada ao frame móvel do próprio robô (i.e. a origem do eixo de coordenadas coincide com a posição do robô para qualquer instante de tempo  $t$  durante a navegação). Fazendo uso das leituras do sensor e de posse de uma estimativa da posição do robô e do conhecimento do mapa, é possível localiza-lo continuamente ao longo de sua trajetória. As vantagens desta configuração são sua simplicidade e relativa facilidade de implementação, visto que trata-se de um sensor único e não de uma rede de sensores.
- Instalação no ambiente: A instalação no ambiente refere-se à utilização de sensores estáticos em pontos específicos do ambiente considerado. Os sensores são posicionados com relação ao referencial global de localização (a origem do plano de coordenadas de cada sensor é estática). Neste caso, é necessário a garantia de que os sensores utilizados cubram todo o espaço em que o robô vai se deslocar de maneira redundante. Desta forma, faz-se necessário a utilização de uma rede de sensores com sensores espalhados em todo o ambiente.

Para uma rede de sensores, o problema de localização de um robô difere-se ligeiramente. Enquanto que no caso do sensor instalado no veículo a localização deste é dada a partir do veículo com relação ao mapa, no caso da montagem dos sensores no

ambiente a sua localização é dada a partir do veículo em relação à posição relativa da rede de sensores.

As vantagens desse tipo de arranjo são que o veículo não precisa ter qualquer tipo de sensor no seu corpo e que vários veículos podem operar em um mesmo ambiente utilizando a mesma rede de sensores. As desvantagens são o aumento do custo justificado pela construção da rede de sensores e a necessidade de garantia de cobertura redundante de área. Como implicação direta desta necessidade, destaca-se a dependência de que todos os sensores funcionem corretamente (diminuição na confiabilidade do sistema).

- Instalação conjunta no veículo e ambiente: Para este arranjo, o robô faz uso tanto da informação dos sensores estáticos quanto dos sensores montados na sua própria estrutura. Esta última opção perfaz, sob o ponto de vista de confiabilidade, a forma mais segura de localização, visto que o sistema de localização pode utilizar os dois tipos de informação. Ainda, a utilização de sensores instalados no veículo torna viável a concepção de uma rede de sensores que não cobre todo o ambiente, reduzindo os custos de implementação.

Ainda que a configuração mais usual para instalação do sensor seja a primeira, existem aplicações específicas em que se faz necessário a instalação dos sensores no ambiente. Um exemplo bastante simples é a utilização do sensor de profundidade *Kinect* para a identificação do movimento de uma pessoa. Dentro desta suite de aplicações, referencie-se, por exemplo, ao trabalho de (SAPUTRA et al., 2012), no qual foi realizado um sistema de identificação e rastreamento de formas humanas em uma sala. Para esta montagem, duas câmeras estéreo foram instaladas na sala, sendo suas medidas obtidas e processadas em tempo real.

A Figura 3 apresenta a montagem dos dois sensores *kinect* no ambiente. Em azul é representado as respectivas áreas de cobertura de cada sensor.

Ainda no contexto de aplicações com os sensores montados no ambiente, um importante trabalho a ser referenciado é o de (FERREIRA; VALE; VENTURA, 2013). Neste, é apresentada uma estratégia de localização de um veículo com cinemática rômica para a planta do ITER (*International Experimental Thermonuclear Reactor*). O ITER é um projeto de colaboração científica internacional de caráter experimental que visa construir um reator de fusão nuclear do tipo *tokamak*, com o intuito de demonstrar a segurança e eficiência da utilização deste tipo de tecnologia na produção comercial de energia elétrica.

O controle e manipulação dos veículos que acessam o reator deve ser autônomo e o veículo deve fazer uso de sensores eficientes para localizar-se. Além disso, há uma restrição quanto à utilização de sensores montados diretamente nos veículos devido à iteração com

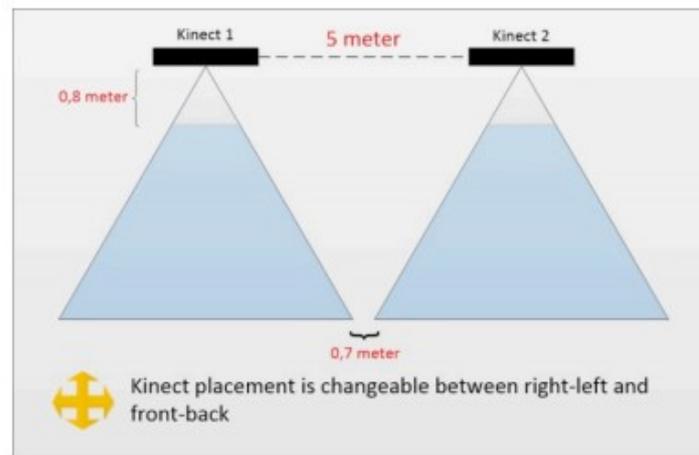


Fig. 3 – Esquema de representação da instalação das duas câmeras estéreo fixas no ambiente. Retirado de (SAPUTRA et al., 2012)

possíveis fontes de radioatividade, fator este que pode causar uma redução na vida útil do equipamento e um aumento da incerteza nas medições.

Frente ao desafio imposto por estas restrições, é proposto em (FERREIRA; VALE; VENTURA, 2013) um sistema de localização utilizando uma rede de sensores laser do tipo LIDAR. Nesta rede, a distribuição dos sensores é realizada a partir de um procedimento de otimização que visa minimizar o número de sensores enquanto maximiza a área coberta e a redundância de cobertura. Dois algoritmos de localização foram testados nesta configuração (*Extended Kalman Filter* e *Particle Filter*). Para ambos o resultado da localização foi eficiente.

A Figura 4 mostra a instalação dos sensores laser em um dos ambientes da planta do ITER. São apresentados quatro sensores montados na parede, bem como os seus polígonos de cobertura, apresentados pela cor amarelo, verde, vermelho e roxo (partindo do sensor inferior em sentido horário, respectivamente). O veículo em consideração é representado pelo retângulo azul.

O escopo do presente trabalho está contido na concepção de um sistema de calibragem e atuação de sensores laser do tipo LIDAR considerando cenários em que estes encontrem-se montados de maneira fixa no ambiente. Para isso, técnicas de computação gráfica serão empregadas na utilização de algoritmos de registro de superfície para a realização de comparações entre a posição real do sensor e a posição esperada pelo sistema de localização.

Nas próximas seções serão introduzidos alguns conceitos pertinentes para o desenvolvimento deste trabalho. Inicialmente será apresentada uma discussão sobre a representação computacional do ambiente, seguido da definição do problema de calibragem, da definição dos objetivos e finalmente uma seção acerca da estrutura do documento.

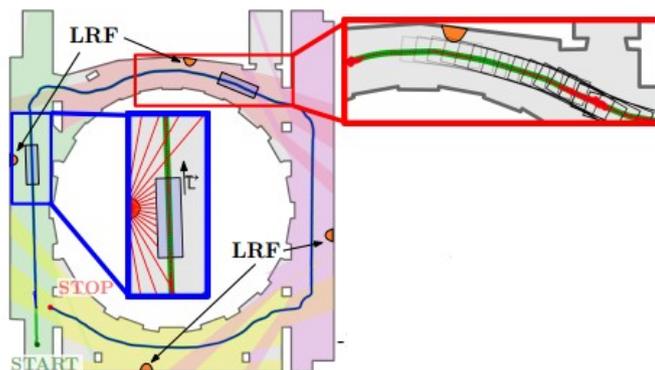


Fig. 4 – Representação do mapa de um dos ambientes do ITER com os sensores LIDAR instalados. Adaptado de (FERREIRA; VALE; VENTURA, 2013)

## 0.1 Representação do ambiente

No contexto da representação do ambiente, pode-se fazer uso da representação em planta baixa do mesmo. Para a descrição eficiente da planta baixa, pode-se representar as paredes como linhas contínuas, dando origem, assim, a um polígono representativo da forma e organização do ambiente. A representação mais simples de linhas em um computador é feita utilizando uma lista de pares  $(P_i \rightarrow P_j)$ , onde  $P_i$  e  $P_j$  representam pontos arbitrários no espaço bidimensional. Cada elemento desta lista representa um seguimento de linha com início em  $P_i$  e término em  $P_j$ . Dados dois pontos, a equação paramétrica da reta (em função de um parâmetro  $x \in [0, 1]$ ) pode ser descrita pela Equação 1.

$$y(x) = (1 - x)P_i + xP_j \quad (1)$$

Embora a representação apresentada anteriormente seja conveniente, deve-se considerar suas limitações. O polígono de interesse a ser descrito deve possuir vértices angulosos, visto que tal representação é incapaz de prover curvatura para o seguimento parametrizado por  $(P_i \rightarrow P_j)$ . Para ultrapassar esta dificuldade, a solução seria considerar a parametrização de um seguimento como uma curva de Bézier quadrática ou de ordens superiores. Considerando a construção de uma curva de Bézier quadrática, a parametrização de cada seguimento é representada por uma tripla de pontos  $(P_i, P_j, P_k)$  e a equação paramétrica do seguimento (em função de um parâmetro  $x \in [0, 1]$ ) é descrita pela Equação 2.

$$y(x) = (1 - x)^2P_0 + 2(1 - x)xP_1 + x^2P_2 \quad (2)$$

Devido à simplicidade dos ambientes tratados neste trabalho, optou-se pela representação dada pela Equação 1. Usualmente serão utilizadas descrições de plantas de ambientes simples, compostos essencialmente de ângulos retos.

## 0.2 Calibragem de sensores

Para motivar e introduzir o conceito de calibragem de sensores, iremos, inicialmente, considerar um problema simples de localização. Considere, para tal, a Figura 5. O robô apresentado possui rodas e pode mover-se em ambas as direções, esquerda e direita.

O sistema de localização do robô utiliza a informação de distância  $d_1$  medida para localizar-se em relação à posição do sensor  $S_i$ . Desta maneira, a posição  $P_i$  pode ser descrita simplesmente por  $P_i = S_i + d_1$ . O leitor deve notar que este tipo de localização é absoluta e leva em consideração a origem do eixo de medidas. Em um contexto relativo, a posição do robô seria dada simplesmente pela distância  $d_1$ .

Considere agora o segundo quadro da Figura 5. Assumindo que uma perturbação de natureza aleatória tenha alterado a posição do sensor, estamos interessados em quantificar os efeitos desta perturbação na estimativa de posição do robô. Devido à natureza aleatória da perturbação, suponha que o sistema de localização tem acesso somente à última posição conhecida do sensor  $S_i$ . Considerando este cenário, tem-se que a estimativa de posição  $P_i$  do robô é dada por  $P_i = S_i + d_2$ . Fica claro, portanto, que a estimativa do sistema de localização possui um erro associado, dado por  $\delta P_i = \Delta S$ .

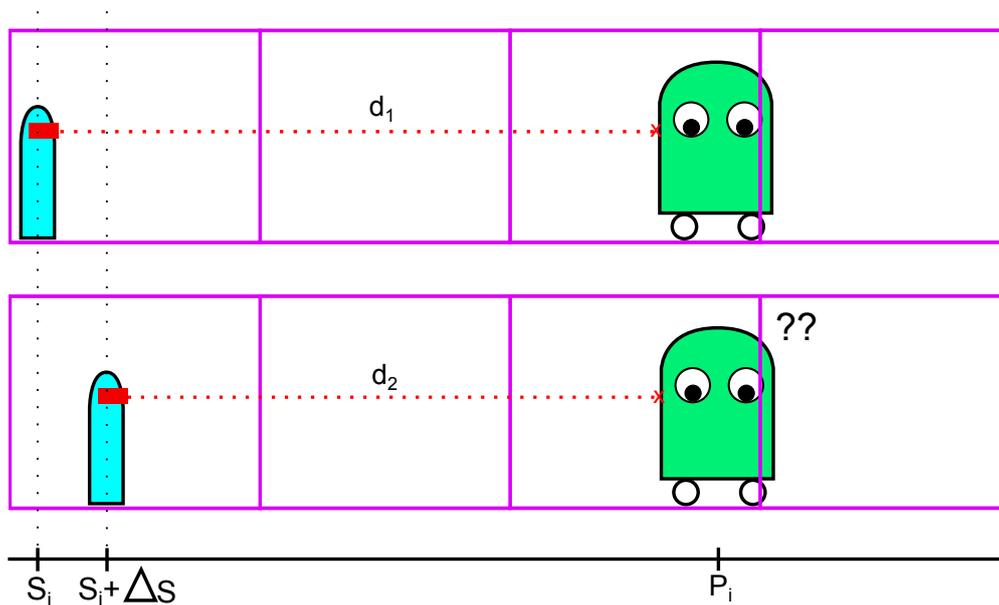


Fig. 5 – Problema de localização sujeito à incerteza na posição do sensor.

O exemplo acima ilustra, embora de maneira simples, a necessidade do controle da posição dos sensores por parte do sistema de localização. A correta estimativa da posição do robô depende do conhecimento preciso da posição e orientação dos sensores durante todo o processo de navegação. Desta maneira surge a necessidade do desenvolvimento de um sistema de monitoramento com vistas a verificar o estado do sensor e, por meio de alguma estratégia, continuamente atualizar o estado do sensor no sistema de localização.

Podemos, agora, definir precisamente o significado do termo calibração no contexto deste trabalho. Calibração refere-se ao processo de atualização do conhecimento do estado de um ou mais sensores continuamente ao longo da operação do sistema de localização. Por estado entende-se as suas variáveis de posição e orientação (posição com relação ao plano bidimensional e orientação com relação ao ângulo  $0^\circ$  definido *a priori* com base em alguma referência).

Com base na definição completa de calibragem, a extensão do problema para o caso do sensor laser LIDAR segue naturalmente. Consideremos, portanto, a ilustração apresentada na Figura 6. Na Figura, um sensor laser é mostrado, bem como a sua área de cobertura e os pontos medidos (asteriscos vermelhos). Para a orientação apresentada, o sensor cobre a região entre as duas colunas.

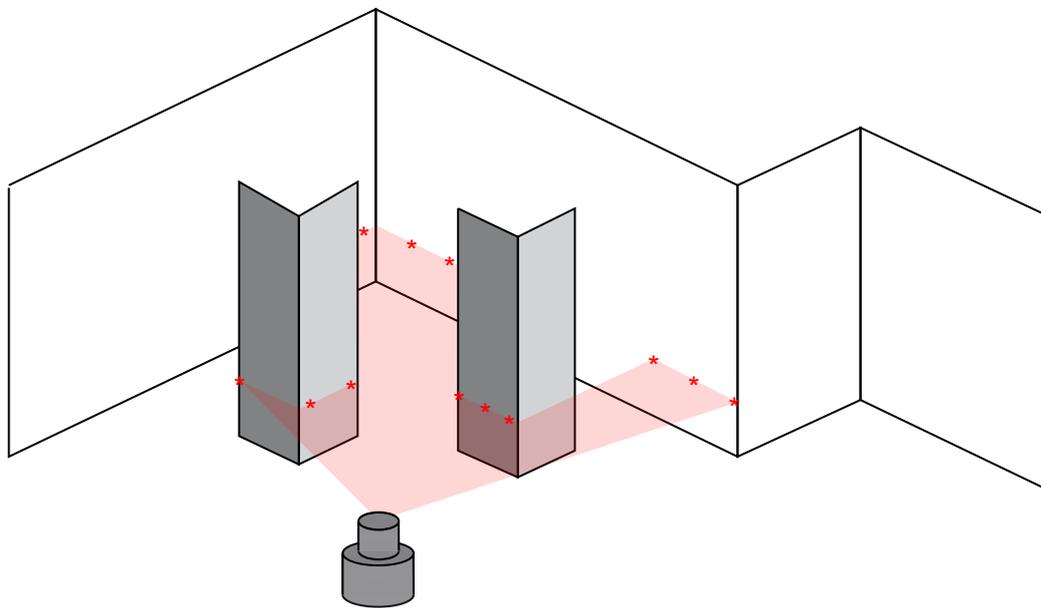


Fig. 6 – Sensor Lidar instalado no ambiente. O polígono em vermelho representa a sua área de cobertura e os pontos discretos as medições.

De maneira semelhante ao problema apresentado no início desta seção, assume-se a ação de uma perturbação de natureza aleatória atuante sobre a posição do sensor. Para este exemplo será considerado somente uma perturbação na sua posição. O parâmetro de orientação permanece o mesmo. A Figura 7 demonstra tal situação. A nova posição do sensor altera também o seu polígono de cobertura (representado em azul) e os pontos lidos pelo sensor.

O problema de calibragem para este caso consiste na atualização da posição do sensor para a nova posição após a realização da perturbação. Devemos atentar ao fato de que, diferentemente do primeiro problema apresentado nesta seção, o sensor laser está sujeito à mais um tipo de perturbação, a perturbação na sua orientação angular. Ainda, o

sistema de calibragem deve ser capaz de identificar qualquer tipo de alteração nas medições recebidas e diferenciar perturbações na posição do sensor de perturbações causadas pelo ruído inerente ao sensor.

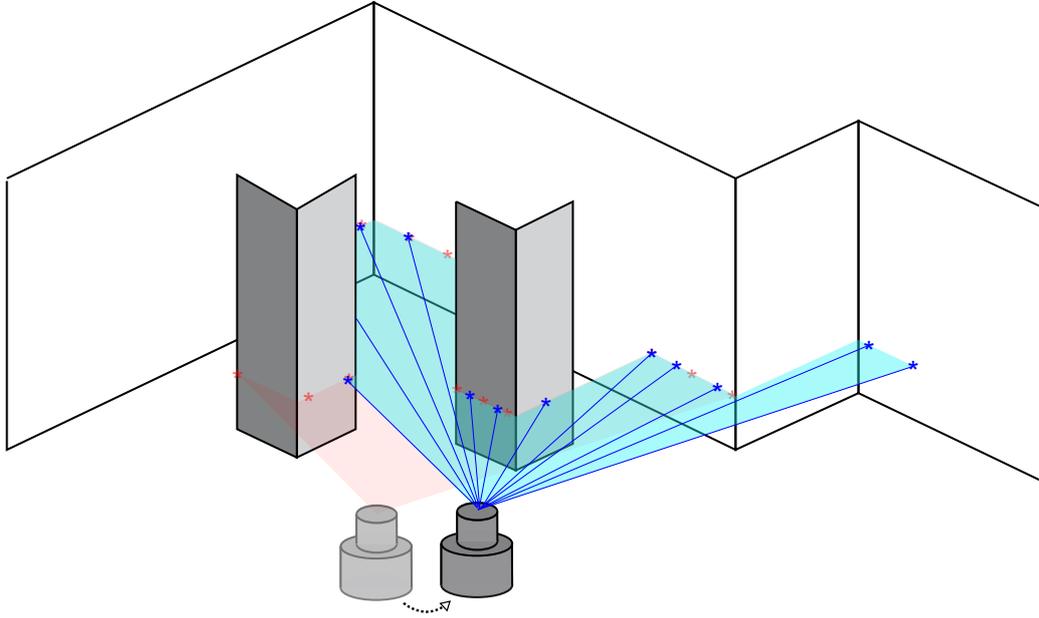


Fig. 7 – Perturbação aplicada ao sensor apresentado na Figura 6. A posição do sensor foi alterada para a direita, resultando em um novo conjunto de medidas do ambiente.

De maneira sintética, pode-se definir o sistema de calibragem dentro do contexto genérico de um sistema robótico. O esquema apresentado na Figura 8 mostra essa estruturação. O sistema robótico faz uso de informações obtidas do sensor para realizar a sua localização no espaço. O módulo de navegação controla os atuadores, com base na informação de localização e na lei de controle. O Sistema de calibragem, por sua vez, garante que as informações obtidas pelos sensores sejam confiáveis, isto é, que o referencial em que o sistema robótico acredita que os sensores estejam posicionados coincida com os estados dos sensores instalados no ambiente. Em caso de divergência entre a leitura obtida pelos sensores e aquela esperada, o sistema de calibragem estima a nova posição e informa o sistema de localização as novas posições/orientações dos sensores.

Deve-se observar que a ação do sistema de calibragem deve ser contínua e, preferencialmente, com uma frequência de atuação igual à frequência de amostragem do sensor. Como será visto posteriormente, limitações no tempo computacional dos algoritmos de registro de superfície não permitem uma operação na mesma frequência que os sensores LIDAR comerciais. Assim, o sistema passa a considerar somente parte das leituras obtidas para verificar a calibragem dos sensores.

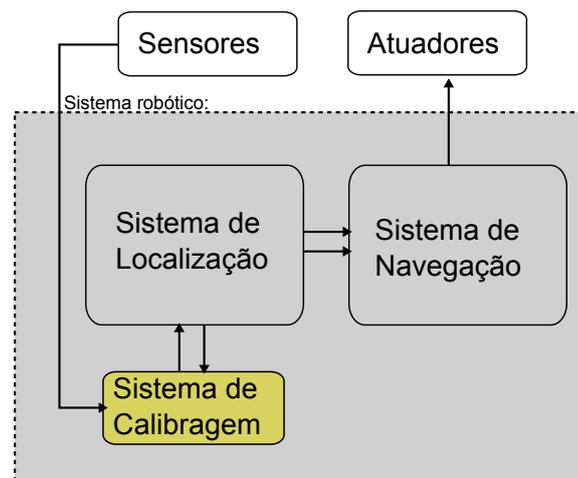


Fig. 8 – Esquema representando um sistema robótico móvel genérico.

### 0.3 Objetivos

Este trabalho tem como objetivo o desenvolvimento teórico e prático de um sistema de calibragem para sensores laser do tipo LIDAR. Para tanto, objetiva-se apresentar técnicas de simulação dos sensores laser, bem como um desenvolvimento de técnicas de algoritmos de registro de superfície, que serão utilizados no processo de calibragem.

### 0.4 Estrutura do Trabalho

Este trabalho está dividido nos seguintes capítulos:

- Introdução;
- Modelagem e simulação do Sensor;
- Algoritmos de registro de superfície;
- Sistema de Calibragem de Sensores;
- Conclusão;

# 1 Modelagem do sensor

Como visto na Introdução, os sensores laser LIDAR são um importante componente no contexto de um sistema robótico de localização, especialmente pela sua precisão, pela sua amplitude de cobertura angular e sua distância de cobertura. São usualmente empregados em aplicações que demandam conhecimento de uma fatia do ambiente, isto é, quando o comportamento do objeto a ser rastreado é invariante com a mudança na cota do ambiente (e.g. um veículo autônomo deslocando-se de maneira rígida em um prédio). Ainda, com respeito à precisão do sensor, pode-se ressaltar o fato de que o tempo de reflexão de um feixe laser varia muito pouco com o tipo de superfície ao qual é refletido. Esta diferença se torna óbvia quando considera-se outros tipos de sensores de distância, como o sonar. No caso específico do sonar, há uma sensibilidade muito maior quanto ao meio de propagação da onda sonora, o que o torna menos eficaz em algumas aplicações. Ainda, a não direcionalidade da onda sonora faz com que haja uma maior incerteza na medida, sobretudo para distâncias de cobertura longas.

A modelagem do sensor laser LIDAR é realizada em função de seus parâmetros básicos, nomeadamente a amplitude de cobertura angular, a resolução de cada leitura, a distância máxima de cobertura e o ruído associado à cada leitura. A simulação deste tipo de sensor em um ambiente computacional faz uso de todas estas propriedades e ainda considera um ambiente no qual o sensor está inserido. Desta forma, o desenvolvimento de um sistema computacional genérico capaz de receber os parâmetros de modelagem do sensor será providencial para a validação do sistema de calibragem. Ainda, a simulação das leituras do sensor deve considerar sua frequência de atuação. Valores comerciais para a frequência de leitura estão na faixa de 10 a 20Hz.

A correta simulação do sensor envolve o desenvolvimento de um sistema que trate o comportamento do sensor e inclua as informações do ambiente em que este está inserido. Desta forma, a obtenção de leituras consiste de um problema intrinsecamente geométrico, em que o sensor é descrito por um vetor  $\mathbf{P}_s = (p_x^s, p_y^s, \theta_s)$ , onde o par  $(p_x^s, p_y^s)$  representa, respectivamente, a posição no eixo de coordenadas  $x$  e no eixo de coordenadas  $y$  do plano cartesiano  $\mathbb{R}^2$  e o ângulo  $\theta_s$  representa a orientação do sensor. O vetor  $\mathbf{P}_s$  será identificado como a pose do sensor ao longo do texto. O problema geométrico em questão consiste na obtenção da distância para um incremento de ângulo com relação ao ângulo  $\theta_s$ , como exemplo  $\theta_s + \delta\theta$ , do ponto  $(p_x^s, p_y^s)$  até a linha mais próxima representando uma porção do ambiente. Vale lembrar que o ambiente em que o sensor encontra-se inserido é descrito por uma estrutura computacional de poli-linha, formada por uma lista de pares  $(P_i, P_j)$ , onde  $P_i$  e  $P_j$  são pontos em  $\mathbb{R}^2$ .

A leitura do sensor é dada por um vetor  $\mathbf{y}_t = (d_i, \theta_i)$ , isto é, pela distância  $d_i$  de reflexão para o ângulo  $\theta_i$ . O ângulo  $\theta_i$  é medido com relação à orientação do sensor  $\theta_s$  no frame de coordenadas do sensor. A adição do ruído nas leituras é realizada de duas maneiras: adição da incerteza utilizando o valor especificado pelo datasheet do fabricante, usualmente descrito em termos de potência em  $dB$ , ou pela estimativa em tempo de execução da variância do ruído de cada medida obtida. Desta forma, a expressão genérica para uma leitura do sensor é dada por  $\mathbf{y}_t = (d_i + n_d, \theta_i)$ , com  $n_d$  representando o ruído de natureza aleatória que está presente nas medidas.

Neste capítulo será apresentado o desenvolvimento feito em direção da correta simulação das medidas do sensor. Para tanto, será introduzido os aspectos de leituras típicas, a modelagem dos parâmetros e a inclusão da incerteza na medida.

## 1.1 Leituras típicas dos sensores LIDAR

As leituras típicas do sensor LIDAR são expressas normalmente em gráficos polares com o ângulo  $\theta$  variando de acordo com o domínio de cobertura angular do sensor. Desta forma obtêm-se leituras contínuas do ambiente. A Figura 9 apresenta a tela do software de aquisição de dados *URG-Viewer*. Nela podemos observar uma leitura do sensor laser LIDAR *URG-04Lx*. O sensor possui ângulo de cobertura de  $240^\circ$  e gera uma leitura completa a cada  $25ms$ .

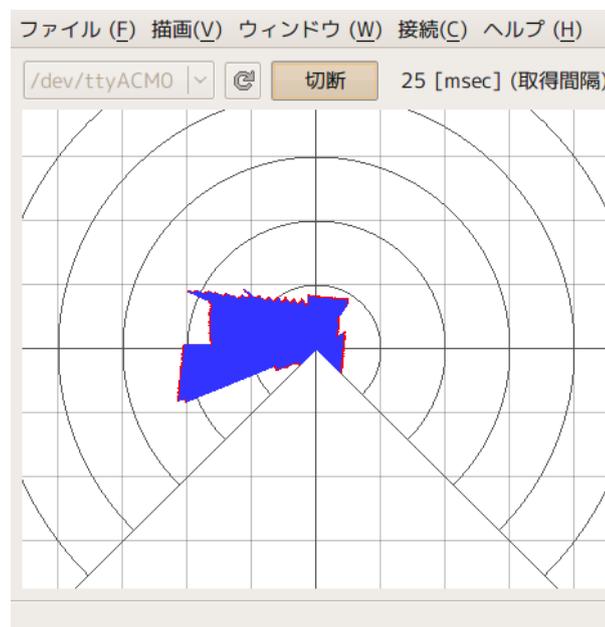


Fig. 9 – Tela do software *URG-Viewer* com leitura do sensor Laser. O diagrama polar possui variação angular de  $240^\circ$ .

Ainda considerando as leituras típicas do sensor, apresentaremos a seguir, um exemplo prático de instalação do sensor em um ambiente. Para tanto, considere inicialmente

a estrutura apresentada na Figura 10. Nela podemos observar a planta baixa de um ambiente hipotético e o posicionamento do sensor em duas configurações: Na primeira configuração (Posição #1) observa-se a orientação do sensor em  $90^\circ$  com relação ao eixo horizontal, enquanto que na segunda configuração (Posição #2) observa-se uma orientação paralela ao eixo horizontal. As Figuras 11 e 12 apresentam as leituras do sensor nas duas

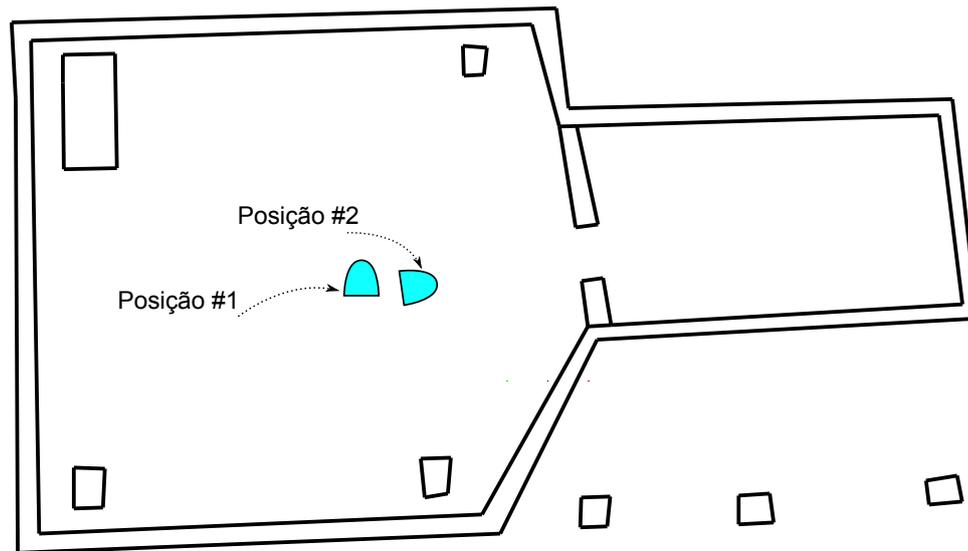


Fig. 10 – Representação em planta baixa de um ambiente hipotético e a instalação do sensor LIDAR em duas posições distintas

posições, respectivamente. Adotou-se para esta simulação uma variação angular de  $240^\circ$  e ruído aditivo gaussiano associado à medida de distância. Ainda, podemos observar que a mudança na orientação angular do sensor fez com que houvesse uma rotação nas estruturas mapeadas de maneira equivalente. Para ilustrar esse efeito, considere a porção do ambiente obtida na leitura que representa a sala do lado direito da planta. Na primeira leitura observa-se que esta estrutura encontra-se próxima ao ângulo de  $0^\circ$ . Na segunda leitura esta estrutura sofreu rotação equivalente à rotação realizada no sensor, localizando-se agora em aproximadamente  $90^\circ$ .

## 1.2 Simulação de leituras do sensor

Considerando o sistema de calibragem a ser desenvolvido neste trabalho, observa-se a necessidade do desenvolvimento de um sistema capaz de realizar simulações de leituras do sensor, valendo-se do conhecimento do mapa e da posição e orientação angular do sensor. Esta seção apresenta o desenvolvimento da modelagem matemática do problema geométrico de geração de leituras simuladas do sensor laser LIDAR.

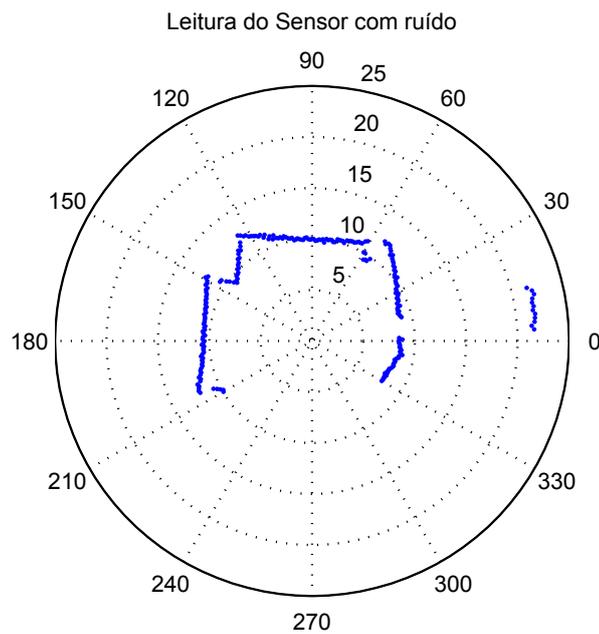


Fig. 11 – Leitura do sensor instalado na Posição #1.

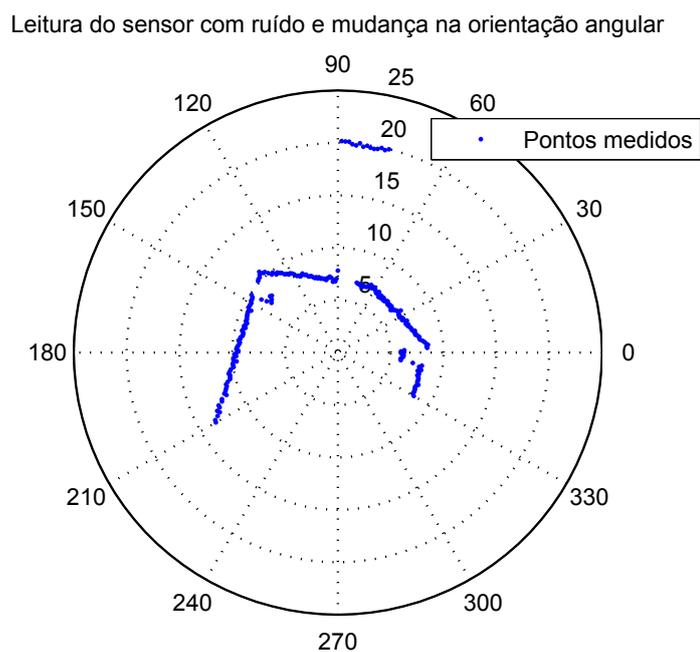


Fig. 12 – Leitura do sensor instalado na Posição #2.

O problema de geração de leituras sintéticas do sensor laser pode ser entendido como um problema puramente geométrico, sendo definido em termos da estrutura geométrica do ambiente e da posição e orientação angular do sensor. Para apresentar o problema, considere inicialmente o caso apresentado na Figura 13. Observa-se na Figura um ponto  $p_1$  arbitrário no espaço e dois segmentos de retas definido pelos pontos  $P_{11} - P_{12}$  e  $P_{21} - P_{22}$ . Os pontos  $q_1$  e  $q_2$  representam, respectivamente, as projeções euclidianas dos segmentos de

linha com início em  $p_1$  e orientação de  $u_1$  e  $u_2$  nos dois segmentos de reta com tracejado sólido.

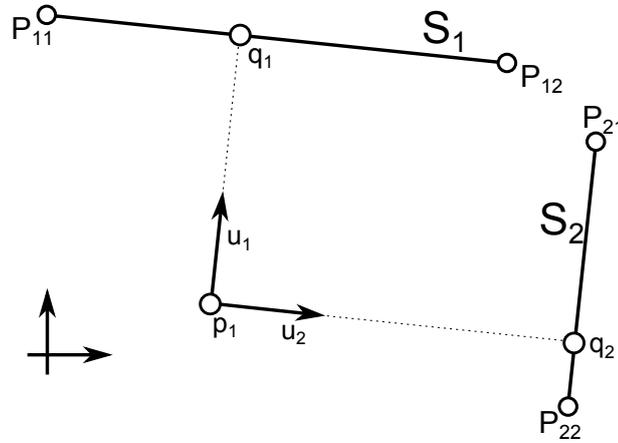


Fig. 13 – Problema geométrico de um ponto orientado por um vetor  $u_1$  em uma reta.

O cálculo da projeção euclidiana de pontos ou retas em funções contínuas é um problema clássico de otimização. Para o caso particular em que a função considerada é afim obtém-se solução fechada através da formulação utilizando mínimos quadrados. Desta forma, considere o segmento  $S_1$  em sua forma parametrizável, dado pela Equação 1.2.

$$S_1(t) = \{\hat{u}t + P_{11} : t \in \mathbb{R}^+\} \quad (1.1)$$

$$\hat{u} = \frac{(P_{12} - P_{11})}{\|P_{12} - P_{11}\|_2}$$

Considere agora um segmento de reta com início em  $p_1 \in \mathbb{R}^2$  e orientado ao longo do vetor unitário  $u_1$  com parâmetro  $t$  irrestrito. O ponto  $q_1$  de menor distância entre os dois segmentos de reta é chamado de projeção do segmento orientado por  $u_1$  no segmento orientado por  $\hat{u}$ .

$$\underset{t}{\text{minimize}} \quad \|\hat{u}t + P_{11} - (u_1t + p_1)\|_2^2 \quad (1.2)$$

Para encontrar a solução para o problema da Equação 1.2 devemos, inicialmente, expandir a função objetivo. Desta forma obtêm-se:

$$\begin{aligned} \|\hat{u}t + P_{11} - (u_1t + p_1)\|_2^2 &= [(\hat{u} - u_1)t + (P_{11} - p_1)]^T [(\hat{u} - u_1)t + (P_{11} - p_1)] \\ &= t^2 - 2t(\hat{u} - u_1)^T(P_{11} - p_1) + \|P_{11} - p_1\|_2^2 \end{aligned} \quad (1.3)$$

Derivando a Equação 1.3 com respeito à variável de otimização obtêm-se a expressão para o ponto ótimo dado pela Equação 1.4.

$$t^* = (\hat{u} - u_1)^T(P_{11} - p_1) \quad (1.4)$$

O ponto  $q_1$  pode ser expresso em termos do parâmetro  $t^*$ . Desta forma, utilizando o resultado da Equação 1.4 na Equação 1.2 obtêm-se a expressão fechada para o ponto  $q_1$  como dado pela Equação 1.5.

$$q_1 = \hat{u}(\hat{u} - u_1)^T(P_{11} - p_1) + P_{11} \quad (1.5)$$

Por brevidade de notação, passaremos a utilizar no decorrer do texto o símbolo  $\Pi\{a, b\}$  para representar o projetor euclidiano da estrutura geométrica  $a$  em  $b$ . Admite-se que  $a$  e  $b$  possam ser pontos ou segmentos de retas no espaço  $\mathbb{R}^2$ . Desta forma o ponto  $q_1$  da Figura 13 pode ser expresso simplesmente por  $q_1 = \Pi\{S_1, u_1 t + p_1\}$ .

Após esta breve introdução, podemos agora utilizar os conceitos de projeção euclidiana para a geração de leituras sintéticas dos sensores laser. Para tanto, considere que o mapa do ambiente é descrito por uma estrutura de dados  $X_i = [P_{i1} P_{i2}]$ , em que  $P_{i1}$  e  $P_{i2}$  representam os pontos iniciais e finais, respectivamente, do  $i$ -ésimo segmento de reta na estrutura  $X_i$ . Considere ainda que a posição e orientação angular do sensor são conhecidas  $\mathbf{P}_s = (p_x^s, p_y^s, \theta_s)$ , bem como a resolução angular  $\delta\theta$  e a amplitude angular da leitura  $\theta_{max}$ . A Figura 14 apresenta uma configuração desta natureza.

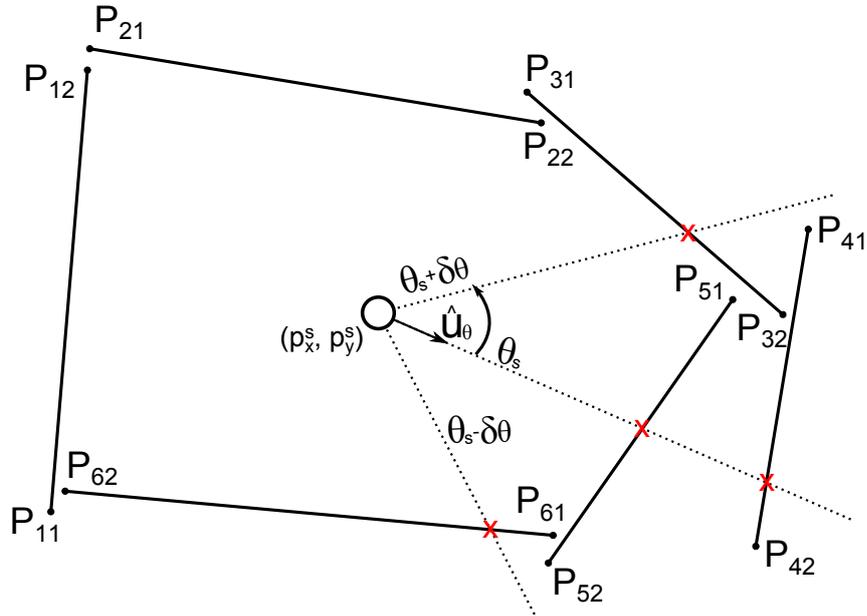


Fig. 14 – Representação gráfica do procedimento de geração de leituras sintéticas do sensor.

Estamos interessados em gerar leituras como aquelas apresentadas nas Figuras 11 e 12. O Algoritmo 1 apresenta uma solução simples para o problema. A saída esperada do algoritmo é uma nuvem de pontos  $y_t = (d_i, \theta_i)$  com  $n$  pares ordenados de valores de distância  $d_i$  e orientação angular  $\theta_i$ . Observe que ainda não foi considerado a inclusão do

ruído nas medidas. Esta inclusão será realizada na próxima seção deste capítulo.

**Data:** Mapa  $\mathbf{X}_i$ ; Pose  $(p_x^s, p_y^s, \theta^s)$  do sensor

**Result:** Nuvem de pontos  $\mathbf{y}_t$ ;

Particione  $y_t$  em  $y_t^{esq}$  e  $y_t^{dir}$ ;

Inicialize  $i = 1$ ;

$m = size(y_t^{dir})$ ;

**for**  $\theta = \theta_s : \delta\theta : \theta_{max}/2$  **do**

$$\left| \begin{array}{l} y_t^{esq}(i) = (\theta, \min \Pi\{S_{X_k}, \hat{u}_\theta t + (p_x^s, p_y^s)\}), \quad k = 1, 2, \dots, m; \\ y_t^{dir}(i) = (\theta - m\delta\theta, \min \Pi\{S_{X_k}, \hat{u}_{(\theta - m\delta\theta)} t + (p_x^s, p_y^s)\}), \quad k = 1, 2, \dots, m; \\ i = i + 1; \end{array} \right.$$

**end**

**Algoritmo 1:** Algoritmo para gerar leituras do sensor  $\mathbf{y}_t$ .

Onde  $S_{X_k}$  representa o  $k$ -ésimo segmento de reta formado pelo par de pontos  $P_{k1}$  e  $P_{k2}$  pertencentes à estrutura  $X_k$  e  $\hat{u}_\theta$  representa o vetor unitário com origem em  $(p_x^s, p_y^s)$  e direção dada pelo ângulo  $\theta$  e o operador  $\min(\cdot)$  retorna o ponto projetado com menor distância de  $(p_x^s, p_y^s)$ . Considerando o Algoritmo 1, observa-se que a tarefa que demora mais tempo é o cálculo do ponto com menor distância de  $(p_x^s, p_y^s)$  na direção de  $\hat{u}_\theta$  em relação a todos os segmentos de reta contidos em  $X_k$ . Algumas melhorias podem ser feitas nesse sentido, como será visto posteriormente. A Figura 15 ilustra o processo de particionamento do vetor de leituras. As duas primeiras leituras são obtidas para um ângulo  $\theta$  de  $0^\circ$  e  $90^\circ$ , respectivamente. O processo continua iterativamente até o número total de leituras ser atingido.

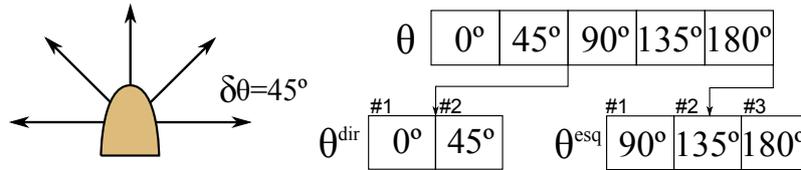


Fig. 15 – Particionamento do vetor de leituras geradas em dois vetores, um à esquerda da orientação central e outra a direita.

Algumas melhorias podem ser feitas no Algoritmo 1. Considerando o passo de cálculo de  $\min \Pi\{S_{X_k}, \hat{u}_\theta t + (p_x^s, p_y^s)\}$  é possível acelerar a convergência do método de minimização. Inicialmente, observa-se que há uma dependência linear do tempo de execução do processo com o número de segmentos que descrevem o ambiente. Considerando o pior caso, a menor distância será encontrada quando for calculada a projeção no último seguimento de  $X_k$ . No entanto, pode-se tomar vantagem da estrutura do mapa e das dimensões dos segmentos de reta em relação ao passo angular do sensor  $\delta\theta$ . Para ilustrar o procedimento, consideremos inicialmente uma restrição para a parametrização do mapa: os segmentos que representam um mesmo ambiente são inseridos na estrutura  $X_k$  de maneira sequencial, de modo que duas paredes adjacentes são parametrizadas com índices  $k$  e  $k + 1$ .

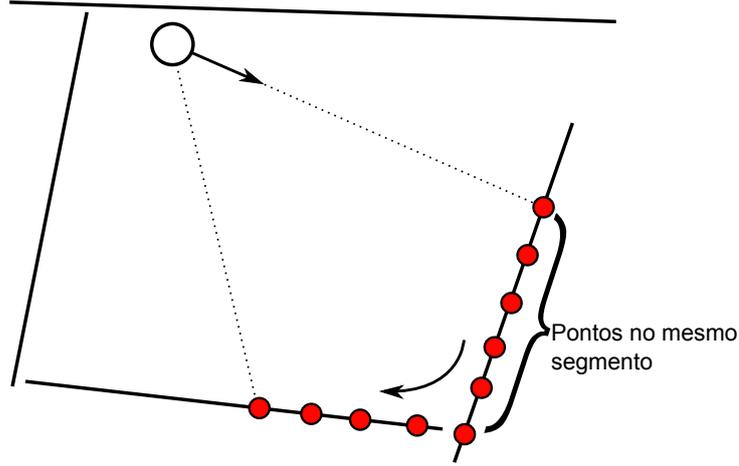


Fig. 16 – Medidas presentes em um mesmo segmento de reta.

Como pode ser observado pela Figura 16, existe uma grande probabilidade de que, para passos angulares adjacentes, o segmento com menor distância de projeção seja o mesmo do passo angular anterior. Desta maneira, pode-se explorar este aspecto para acelerar a busca pelo segmento com menor distância em relação à posição do sensor e a orientação angular utilizada. Para tanto, considere inicialmente o Algoritmo 2 que realiza o cálculo de  $\min \Pi\{S_{X_k}, \hat{u}_{\delta\theta}t + (p_x^s, p_y^s)\}$  de maneira direta.

**Data:** Posição do sensor  $(p_x^s, p_y^s)$ , Vetor de direção  $\hat{u}_{\delta\theta}$  e Mapa  $X_k$ ;

**Result:** Ponto  $q$  com mínima distância ao sensor na direção de  $\hat{u}_{\delta\theta}$ ;

$m = \text{size}(X_k)$ ;

$dist = \|(p_x^s, p_y^s) - \Pi\{S_{X_1}, \hat{u}_{\delta\theta}t + (p_x^s, p_y^s)\}\|_2$  ;

**for**  $i = 1 : 1 : m$  **do**

**if**  $\|(p_x^s, p_y^s) - \Pi\{S_{X_i}, \hat{u}_{\delta\theta}t + (p_x^s, p_y^s)\}\|_2 \leq dist$  **then**

$q = \Pi\{S_{X_i}, \hat{u}_{\delta\theta}t + (p_x^s, p_y^s)\}$ ;

$dist = \|(p_x^s, p_y^s) - \Pi\{S_{X_i}, \hat{u}_{\delta\theta}t + (p_x^s, p_y^s)\}\|_2$  ;

**end**

**end**

**Algoritmo 2:** Implementação direta do Algoritmo para cálculo de  $\min \Pi\{S_{X_k}, \hat{u}_{\delta\theta}t + (p_x^s, p_y^s)\}$ .

Observa-se que para cada ponto do vetor  $y_t$  gerado é necessário o cálculo da projeção do segmento de reta com início em  $(p_x^s, p_y^s)$  e orientação dada por  $\hat{u}_{\delta\theta}$  com relação a todos os segmentos de linhas do mapa  $X_k$  (Algoritmo 2). No entanto, observa-se na Figura 16 que a distância entre cada passo angular  $\delta\theta$  é muito menor que o comprimento do segmento de reta que representa a parede do ambiente. Este fato pode ser verificado em ambientes práticos na maioria dos casos.

Assim, o Algoritmo 1 pode ser acelerado se considerarmos que os próximos pontos da leitura simulada pertencem ao mesmo segmento de reta. Para modelar esta estratégia, é

necessário verificar se, para dois pontos de  $y_t$  separados por um número inteiro de pontos  $\epsilon$ , digamos  $y_t(i)$  e  $y_t(i + \epsilon)$ , os respectivos pontos projetados com mínima distância calculados a partir do Algoritmo 2 pertencem ao mesmo segmento de reta  $l$  do mapa  $X_k$ . Em caso afirmativo, assume-se que todos os pontos intermediários entre  $y_t(i)$  e  $y_t(i + \epsilon)$  possuem distância mínima de projeção com o segmento de reta  $l$  do mapa  $X_k$ , não sendo necessária a comparação de cada ponto com todos os segmentos de reta do mapa. Em caso negativo, isto é, se o segmento de reta do mapa  $X_k$  com mínima distância em relação à  $y_t(i + \epsilon)$  for diferente do segmento de reta com mínima distância de  $y_t(i)$ , então deve-se realizar a busca completa nos pontos anteriores a  $y_t(i + \epsilon)$  para verificar quais pontos possuem mínima distância com relação ao mesmo segmento do mapa que  $y_t(i + \epsilon)$ .

Para ilustrar o procedimento, considere a Figura 17. Observa-se a leitura simulada do sensor instalado em um ambiente com duas paredes (representadas pelos segmentos  $S_1$  e  $S_2$ ). Os pontos azuis representam aqueles que foram calculados a partir da busca completa em todos os segmentos de retas do mapa, enquanto que os pontos vermelhos são aqueles em que houve a consideração de que o segmento com mínima distância era o mesmo que o segmento do seu vizinho adjacente. Para o cálculo do primeiro ponto  $y_t(1)$  é necessário calcular a projeção para os dois segmentos de reta, sendo o segmento  $S_1$  com mínima distância. O algoritmo salta 3 pontos e realiza uma busca completa para a obtenção da medida  $y_t(5)$ . Como o seguimento com mínima distância em relação à  $y_t(5)$  continua sendo  $S_1$ , assume-se o mesmo para as leituras de posição 2 e 3 e 4. Observe que para estes últimos pontos não é necessário verificar qual segmento resulta na projeção mínima. Considerando agora o ponto  $y_t(6)$ , observa-se que o salto de 3 posições resulta em um ponto ( $y_t(10)$ ) cujo segmento com distância projetada mínima difere de  $S_1$ . Desta forma, é necessário fazer uma busca completa para as leituras anteriores de  $y_t(10)$ . O retorno acaba quando encontra-se o ponto cuja distância projetada mínima está associada com o segmento  $S_1$  ( $y_t(7)$  neste caso).

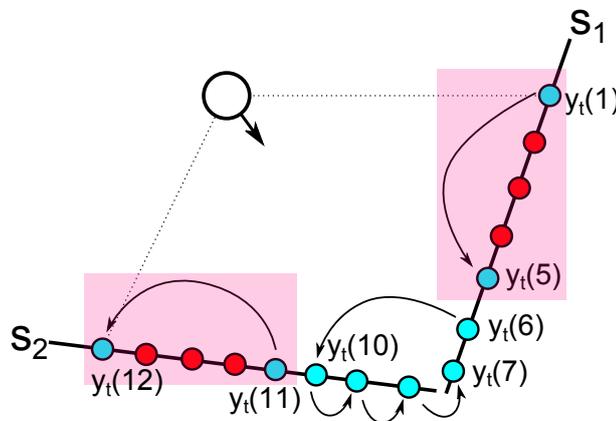


Fig. 17 – Versão acelerada do algoritmo de geração de leituras do sensor.

Observa-se que a escolha do parâmetro do número de leituras que são puladas

impacta diretamente no número de buscas completas realizadas. Este parâmetro não deve exceder o tamanho da menor estrutura representada no ambiente e deve ser suficientemente grande para evitar buscas completas desnecessárias em segmentos longos.

### 1.3 Estimativa da variância do ruído

O ruído presente na medida do sensor é um importante parâmetro na modelagem e simulação do seu comportamento. Desta maneira, se faz necessário o estudo da característica de ruído presente em um sensor real. Para realizar esta estimativa, adotou-se uma metodologia baseada na aplicação do filtro de medianas. Esta metodologia, comumente utilizada em processamento de imagens, consiste na aplicação de um vetor de convolução sobre o sinal em consideração, atribuindo a cada amostra o valor da mediana das amostras vizinhas determinadas pelo vetor de convolução. Desta maneira, o sinal resultante da aplicação do filtro de medianas pode ser subtraído do sinal original, resultando no resíduo entre o sinal filtrado e o sinal original. Para a quantificação das características de ruído, extrai-se os parâmetros estatísticos do resíduo (média, mediana e variância) assumindo que trata-se de ruído gaussiano branco.

## 2 Algoritmos de registro de superfície

Como definido em (ZITOVÁ; FLUSSER, 2003), o problema de registro de imagens consiste na sobreposição de duas ou mais imagens obtidas em tempos distintos, com pontos de vista distintos e/ou com diferentes sensores. Esta definição, embora genérica, define precisamente o problema. A sobreposição das imagens é realizada por meio de transformações aplicadas à uma das imagens, de maneira que esta, após a transformação, satisfaça uma métrica de erro definida entre a imagem referência e a imagem sentida após a transformação aplicada. No contexto das transformações possíveis, podemos classifica-las entre rígidas e não-rígidas. As transformações rígidas são de característica mais simples e, usualmente, consistem de um conjunto de transformações de corpo rígido (Rotação seguida de translação). As transformadas não-rígidas, por sua vez, consistem de transformações que mapeiam não linearmente as equivalências entre propriedades únicas de cada imagem.

A Figura 18 apresenta os passos de execução de um algoritmo de registro de imagens típico. Neste caso em específico, observa-se inicialmente a obtenção de descritores invariantes (pontos definidos pela interceptação das retas do prédio) para ambas as imagens. Da obtenção dos descritores segue diretamente o casamento entre as duas imagens, com o objetivo de minimizar uma métrica de erro associada aos descritores. O resultado final pode ser observado na terceira imagem da sequência.

Devido à incrível quantidade de problemas envolvendo registro de imagens, observa-se a existência de inúmeros métodos de casamento, com os mais variados condicionamentos de problemas e dimensões. Trabalhos neste contexto incluem o reconhecimento facial a partir de um banco de dados de imagens (AMOR; ARDABILIAN; CHEN, 2006), reconstrução tridimensional de modelos (VEN; RAMOS; TIPALDI, 2010) e localização e navegação de robôs autônomos móveis (MAGREE; JOHNSON, 2014).

Usualmente os algoritmos de registro de imagens fazem uso de um par de imagens a cada iteração. A primeira imagem, discriminada como modelo, representa a imagem que queremos utilizar como base para o casamento. A segunda, denominada imagem sentida, representa a imagem que será conformada ao modelo. No caso específico do problema de calibragem de sensores laser do tipo LIDAR, as imagens utilizadas consistem de vetores de pares ordenados  $(\theta, d)$ , com  $\theta$  sendo o ângulo de leitura do sensor e  $d$  a distância relativa ao obstáculo mais próximo. A imagem sentida é obtida a partir de leituras periódicas dos dados do sensor laser LIDAR. A imagem modelo, por sua vez, é obtida a partir de leituras sintéticas produzidas a partir do conhecimento da posição do sensor e do mapa do ambiente em que este está instalado. Neste contexto, deseja-se encontrar a transformação de corpo rígido que sobrepõe de maneira ótima as duas imagens.

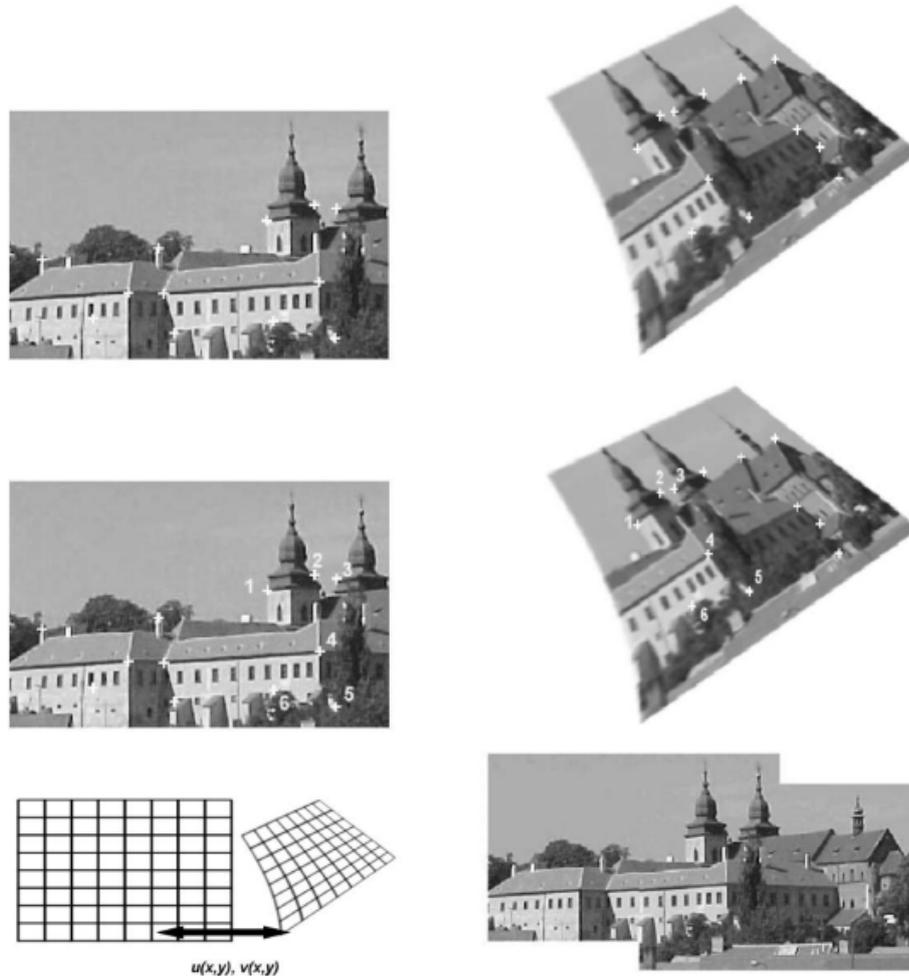


Fig. 18 – Processo de sobreposição de duas imagens utilizando descritores invariantes e uma transformação não-rígida. Retirado de (ZITOVÁ; FLUSSER, 2003).

Nas próximas seções será apresentada uma introdução ao método de registro de superfícies utilizado neste trabalho, nomeadamente o método ICP (*Iterative Closest Point*), bem como a formulação de dois algoritmos para a resolução do problema.

## 2.1 Algoritmo *Iterative closest Point* (ICP)

O método ICP é um algoritmo de registro de superfície muito popular. O seu objetivo consiste na minimização iterativamente da diferença entre duas nuvens de pontos por meio da aplicação sucessiva de Transformações de Corpo Rígido (*TCR*) que relacionam a nuvem modelo com a nuvem sentida.

A proposição original do algoritmo foi realizada em (CHEN; MEDIONI, 1991), sendo este extensivamente melhorado desde sua proposição inicial. O algoritmo pode ser definido genericamente pelos passos apresentados abaixo (variantes específicas ao contexto foram introduzidas na literatura, sendo estes os passos apresentados abaixo fundamentais

para a execução do algoritmo). Considere inicialmente a existência de duas nuvens de pontos  $\mathbf{p}_i$  e  $\mathbf{p}'_i$ . A Figura 19 apresenta um exemplo de duas nuvens, sendo a primeira, em azul, representando a nuvem de pontos modelo e a segunda, em verde, a nuvem de pontos sentida (resultado de uma rotação de  $45^\circ$  e adição de ruído com relação à nuvem modelo).

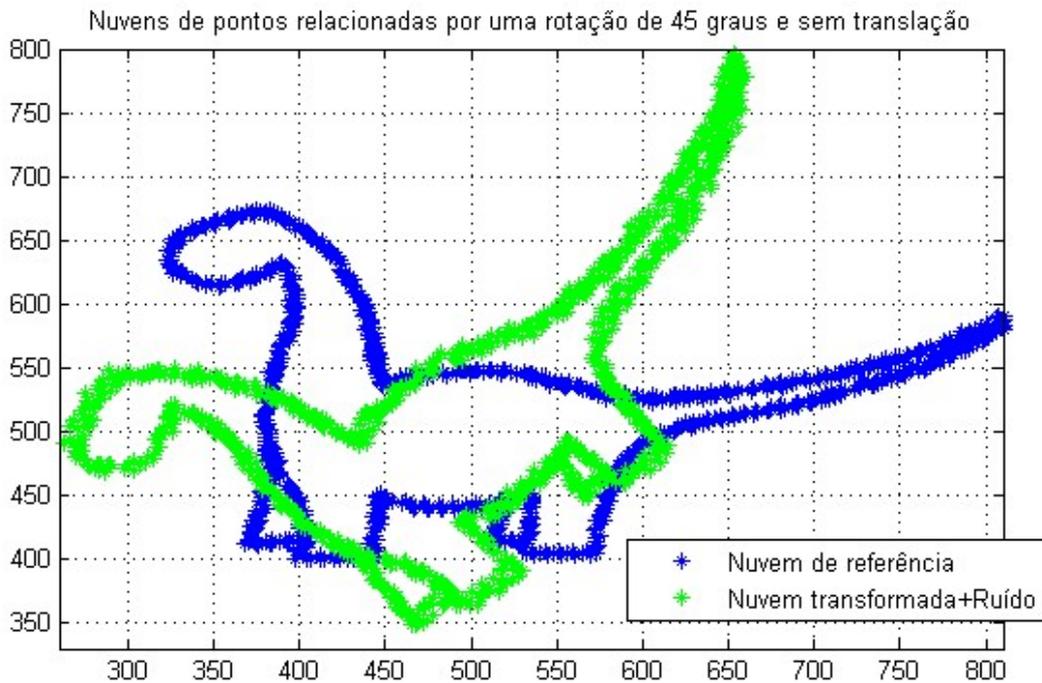


Fig. 19 – Nuvens de pontos  $\mathbf{p}_i$  (azul) e  $\mathbf{p}'_i$  (verde).

Passos executados iterativamente considerando a estrutura genérica de um algoritmo *ICP* :

- Casamento: Nesta etapa é realizada a associação de cada ponto na imagem sentida com um ponto correspondente na imagem modelo utilizando alguma métrica de minimização, usualmente relacionada à distância entre os pontos. Diversas métricas podem ser utilizadas, como por exemplo a mínima distância ortogonal entre os pontos, a mínima distância com respeito a projeção de um seguimento de reta na imagem sentida, etc.
- Minimização: A métrica de erro é minimizada. Para o caso de transformações de corpo rígido o problema é formulado em termos da matriz de rotação  $\mathbf{R}$  e do vetor de translação  $\mathbf{T}$ , que minimiza o erro associado à transformação da imagem sentida com relação ao modelo. Considerando a versão tridimensional do algoritmo ICP, a matriz  $\mathbf{R} \in \mathbb{R}^{3 \times 3}$  e o vetor  $\mathbf{T} \in \mathbb{R}^{1 \times 3}$ .
- Transformação: Neste passo a imagem sentida é transformada utilizando a T.C.R. definida pelo par  $(\mathbf{R}, \mathbf{T})$ . O algoritmo é atualizado sendo a imagem sentida atualizada.

Os passos definidos acima são repetidos iterativamente com a atualização a cada iteração da imagem sentida em função da transformação  $(\mathbf{R}, \mathbf{T})$ . O algoritmo converge quando a imagem sentida sucessivamente transformada sobrepõe a imagem modelo com base em um limiar de erro. A execução iterativa do termina quando o algoritmo convergir ou o número de execuções for superior à um limite de iteração  $\rho$ .

Nas próximas seções serão apresentados em detalhe cada etapa do algoritmo, apresentando possíveis variantes que possam tornar a execução do algoritmo mais eficiente.

## 2.2 Formulação do problema de minimização

O segundo passo do algoritmo ICP representa a sua principal dificuldade. A formulação do problema de minimização deve ter como variáveis a matriz de rotação  $\mathbf{R}$  e o vetor de translação  $\mathbf{T}$ . Neste contexto, a minimização da diferença entre as duas nuvens no sentido de mínimos quadrados pode ser expressa diretamente em função do par  $(\mathbf{R}, \mathbf{T})$ . Considere, portanto, o problema de minimização irrestrito dado pela Equação 2.1.

$$J(\mathbf{R}, \mathbf{T}) = \arg \min_{\mathbf{R}, \mathbf{T}} \sum_{i=0}^N \|\mathbf{R}\mathbf{p}'_i + \mathbf{T} - \mathbf{p}_i\|^2 \quad (2.1)$$

Estamos interessados em minimizar a diferença quadrática entre a nuvem de pontos  $\mathbf{p}_i$  e a nuvem de pontos  $\mathbf{p}'_i$  após a aplicação da rotação  $\mathbf{R}$  e da translação  $\mathbf{T}$ . O problema consiste, portanto, na busca do par  $(\mathbf{R}, \mathbf{T})$  que minimiza a diferença entre as duas nuvens. Ao longo dos anos, inúmeras soluções para este problema foram propostas, dentre as quais podemos destacar o trabalho de (ARUN; HUANG; BLOSTEIN, 1987) e (CENSI, 2008). Nas subseções abaixo serão apresentados os dois algoritmos.

### 2.2.1 Algoritmo de Arun

O algoritmo de Arun (ARUN; HUANG; BLOSTEIN, 1987) fornece uma solução fechada para a Transformação de Corpo Rígido ótima entre duas nuvens de pontos arbitrárias. O procedimento basea-se no desacoplamento da busca pela matriz de rotação  $\mathbf{R}$  ótima com respeito ao vetor de translação  $\mathbf{T}$ . O algoritmo é apresentado abaixo, conforme a sua proposição original.

Considere, para tanto, o problema de minimização irrestrito dado pela Equação

2.1. O par transformada  $(\mathbf{R}, \mathbf{T})$  ótimo pode ser computado a partir do Algoritmo 3.

**Data:** Nuvens de pontos  $\mathbf{p}_i$  e  $\mathbf{p}'_i$

**Result:** matriz de rotação  $\mathbf{R}$  e vetor de translação  $\mathbf{T}$

Dadas as duas nuvens de pontos  $\mathbf{p}_i$  e  $\mathbf{p}'_i$ ;

Calcular  $\mathbf{p}$ ,  $\mathbf{p}'$  e  $\mathbf{q}_i$  e  $\mathbf{q}'_i$  como apresentado na seção 1.2.2;

Calcular a matriz  $\mathbf{H} = \sum_{i=1}^N \mathbf{q}_i \mathbf{q}'_i{}^T \in R^3$ ;

Encontrar a decomposição *SVD* de  $\mathbf{H}$ :  $\mathbf{H} = \mathbf{U} \mathbf{\Lambda} \mathbf{V}^T$ ;

Calcular  $\det(\mathbf{V} \mathbf{U}^T)$  ;

**if**  $\det(\mathbf{V} \mathbf{U}^T) == 1$  **then**

$\mathbf{R} = \mathbf{V} \mathbf{U}^T$ ;

$\mathbf{T} = \mathbf{R} \mathbf{p}'_i - \mathbf{p}_i$  ;

**else**

**Algoritmo 3:** Algoritmo de Arun para o cálculo da matriz de rotação  $\mathbf{R}$  e o vetor de translação  $\mathbf{T}$ .

Considerando a complexidade computacional do algoritmo pode-se analisar alguns aspectos. Observa-se que para o comupto da matriz  $\mathbf{H}$  é necessário o cálculo dos centróides das duas nuvens  $\mathbf{p}_i$  e  $\mathbf{p}'_i$ . Além disso, a matriz  $\mathbf{H}$  trata-se da soma sucessiva do produto matricial entre os vetores  $\mathbf{q}_i \in \mathbb{R}^{1 \times 3}$  e  $\mathbf{q}'_i{}^T \in \mathbb{R}^{3 \times 1}$ . Após o cálculo da matriz  $\mathbf{H}$ , segue a decomposição *SVD* da mesma, bem como o cálculo do determinante do termo  $\mathbf{V} \mathbf{U}^T$ . Desta forma, observa-se que o algoritmo comporta-se linearmente em função do número de pontos  $N$ . Considerando os aspecto temporal de sua execução, pode-se atribuir ao cálculo da matriz  $\mathbf{H}$  a tarefa mais demorada no tempo de execução do algoritmo.

## 2.2.2 Derivação do algoritmo Arun

Considerando a Equação 2.1, tomamos inicialmente a derivada parcial com respeito à variável  $\mathbf{T}$ .

$$\frac{\partial J(\mathbf{R}, \mathbf{T})}{\partial \mathbf{T}} = 2\mathbf{T} \sum_{i=1}^N 1 + 2\mathbf{R} \sum_{i=1}^N \mathbf{p}_i - 2 \sum_{i=1}^N \mathbf{p}'_i \quad (2.2)$$

Igualando o resultado à zero obtêm-se a translação ótima, que neste caso é dada pela Equação 2.3.

$$\mathbf{T} = \frac{\sum \mathbf{p}'_i}{N} + \mathbf{R} \frac{\sum \mathbf{p}_i}{N} \quad (2.3)$$

É de interesse agora verificar a existência de uma forma fechada para a solução da busca da matriz  $\mathbf{R}$  ótima. Para tal, Define-se as variáveis  $p$  e  $p'$ , aqui denominadas de centróides de  $\mathbf{p}_i$  e  $\mathbf{p}'_i$ , respectivamente, como sendo:

$$p = \frac{\sum \mathbf{p}_i}{N}, \quad p' = \frac{\sum \mathbf{p}'_i}{N}$$

Utilizando da definição dos centróides, a Equação 2.1 pode ser reescrita como sendo:

$$J(\mathbf{R}, \mathbf{T}) = \sum_{i=1}^N \|(\mathbf{R}\mathbf{p}_i + \mathbf{p}' - \mathbf{R}\mathbf{p} - \mathbf{p}'_i)\|^2 \quad (2.4)$$

$$= \sum_{i=1}^N \|(\mathbf{R}(\mathbf{p}_i - \mathbf{p}) - (\mathbf{p}'_i - \mathbf{p}'))\|^2 \quad (2.5)$$

Introduzindo as novas variáveis  $\mathbf{q}_i = \mathbf{p}_i - \mathbf{p}$  e  $\mathbf{q}'_i = \mathbf{p}'_i - \mathbf{p}'$  é possível reformular o problema de minimização somente em função da variável  $\mathbf{R}$ , como descrito na Equação 2.6.

$$J = \arg \min_{\mathbf{R}} \sum_{i=1}^N \|\mathbf{R}\mathbf{q}_i - \mathbf{q}'_i\|^2 \quad (2.6)$$

Expandindo o termo quadrático obtêm-se:

$$\begin{aligned} J(\mathbf{R}) &= \arg \min_{\mathbf{R}} -2 \sum_{i=1}^N \mathbf{q}'_i{}^T \mathbf{R} \mathbf{q}_i \\ &= \arg \max_{\mathbf{R}} \sum_{i=1}^N \mathbf{q}'_i{}^T \mathbf{R} \mathbf{q}_i \\ \rightarrow J(\mathbf{R}) &= \arg \max_{\mathbf{R}} \text{Tr}(\mathbf{Q}'_i \mathbf{R} \mathbf{Q}_i) \end{aligned} \quad (2.7)$$

Considerando a propriedade do operador traço  $\text{Tr}(\mathbf{AB}) = \text{Tr}(\mathbf{BA})$ , podemos reescrever a Equação 2.7 como sendo:

$$J(\mathbf{R}) = \arg \max_{\mathbf{R}} \text{Tr}(\mathbf{R} \mathbf{Q}_i \mathbf{Q}'_i{}^T) \quad (2.8)$$

Considere agora a forma matricial  $\mathbf{Q}_i \mathbf{Q}'_i{}^T$ . Segue da observação que este termo trata-se da decomposição SVD de alguma matriz  $\mathbf{P}$ . Dessa maneira, é direta a conclusão de que ambos  $\mathbf{Q}_i$  e  $\mathbf{Q}'_i$  são matrizes ortogonais. Pode ser demonstrado, fazendo uso da propriedade de ortogonalidade de cada matriz, que o Traço é maximizado quando o produto  $\mathbf{Q}'_i \mathbf{R} \mathbf{Q}_i$  for igual à matriz identidade. Com esta condição, podemos escrever a condição de optimalidade como sendo:

$$\mathbf{R} = \mathbf{Q}'_i{}^T \mathbf{Q}_i \quad (2.9)$$

O computo do determinante do termo  $\det(\mathbf{V}\mathbf{U}^T)$  no Algoritmo 3 garante que o termo  $\mathbf{V}\mathbf{U}^T$  trata-se, de fato, de uma matriz ortogonal do tipo rotação.

## 2.3 ICP point-to-line

O trabalho recente de (CENSI, 2008) propõe uma métrica alternativa para a etapa de casamento dos pontos das duas nuvens  $\mathbf{p}_i$  e  $\mathbf{p}'_i$ . A métrica de associação em questão

considera uma superfície de referência  $S_{ref}$  na nuvem de pontos do modelo. Esta estrutura trata-se de um conjunto de segmentos de reta formados pela união de pontos adjacentes. Desta forma, cada ponto presente na nuvem sentida ( $\mathbf{p}'_i$ ) associa-se com um segmento de reta dado pelos dois pontos mais próximos de  $\mathbf{p}'_i$  em  $\mathbf{p}_i$ . O procedimento de associação é ilustrado na Figura 20. Os índices  $j_1^i$  e  $j_2^i$  estão associados com os dois pontos mais próximos de  $\mathbf{p}'_i$  em  $\mathbf{p}_i$ .

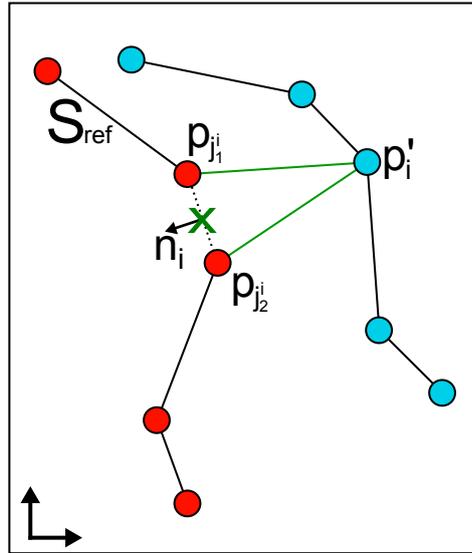


Fig. 20 – Métrica de associação entre as duas nuvens de pontos  $\mathbf{p}_i$  e  $\mathbf{p}'_i$  utilizando a abordagem *point-to-line*. O ponto em verde indica o ponto  $\Pi_i\{S_{ref}, \mathbf{p}'_i\}$ . Modificado de (CENSI, 2008).

Como consequência desta abordagem de associação é possível reescrever a função custo do problema de minimização dado pela Equação 2.1, resultando em um problema de minimização com restrições. Como apresentado em (CENSI, 2008), a resolução de uma iteração do algoritmo equivale à uma iteração *Gauss-Newton*. Desta forma, enquanto que a implementação tradicional do ICP (utilizando o algoritmo de (ARUN; HUANG; BLOSTEIN, 1987)) apresenta convergência linear, o algoritmo pIICP apresenta convergência quadrática. Ainda, se considerarmos certas condições, é possível fornecer garantias matemáticas de convergência em um número finito de passos. Dentre as condições necessárias para tal conclusão destaca-se a necessidade da superfície de referência possuir derivadas limitadas e de haver um bom chute inicial para a TCR. Estes requisitos não são usualmente satisfeitos em aplicações práticas. Para o caso específico das leituras do sensor LIDAR, podemos observar que a superfície de referência é finitamente descontínua e que, portanto, não possui derivadas limitadas. Ainda, a existência de um bom chute inicial não pode ser considerada se levarmos em consideração a imprevisibilidade das ações de um sistema robótico.

O desenvolvimento da metodologia para o algoritmo pIICP será apresentado na

seções subsequentes. Optou-se pela apresentação da derivação do algoritmo antes da apresentação do algoritmo em si. O leitor interessado na implementação prática do algoritmo pode ignorar a seção de derivação, sem perda de generalidade.

### 2.3.1 Derivação do algoritmo plICP

Inicialmente, considere a forma genérica do quadrado da norma  $\|\mathbf{a}\|_{\mathbf{C}}^2$  definido por  $\|\mathbf{a}\|_{\mathbf{C}}^2 = \mathbf{a}^T \mathbf{C} \mathbf{a}$ . Para que o operador seja uma norma, é necessário que a matriz  $\mathbf{C}$  seja ortonormal.

Se utilizarmos a nova norma para escrever o problema de minimização, a Equação 2.1 pode ser reescrita como:

$$J(\mathbf{R}, \mathbf{T}) = \min \arg_{\mathbf{R}, \mathbf{T}} \sum_{i=0}^N \|(\mathbf{R}\mathbf{p}'_i + \mathbf{T}) - \mathbf{\Pi}_i\|_{\mathbf{C}_i}^2 \quad (2.10)$$

Aqui vale observar uma ligeira mudança com relação à Equação 2.1. Ao invés de utilizarmos os próprios pontos da nuvem  $p_i$ , estamos utilizando o projetor euclidiano  $\mathbf{\Pi}_i$ , isto é, a projeção do ponto  $\mathbf{p}'_i$  na superfície de referência  $S_{ref}$  formada pela poli-linha que conecta todos os pontos adjacentes de  $\mathbf{p}_i$  (observe a Figura 20 para uma ilustração da estrutura).

Para encontrar a solução fechada na forma de um problema de minimização com restrições, devemos inicialmente expandir o espaço de busca. Considerando o par  $(R, T)$ , observa-se que as variáveis de otimização são  $(\theta, T_x, T_y)$ , representando, respectivamente, o ângulo de rotação  $\theta$  e as componentes de translação no plano bidimensional  $T_x$  e  $T_y$ . Se considerarmos a extensão para o vetor de variáveis  $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4] \triangleq [\cos \theta \ \sin \theta \ T_x \ T_y]$  e incluirmos a restrição de que  $x_1^2 + x_2^2 = 1$ , podemos reescrever o problema no domínio de busca  $\mathbb{R}^4$ .

Para tanto, decompõe-se o ponto  $\mathbf{p}'_i$  em suas componentes euclidianas  $\mathbf{p}'_i = (p'_{i0}, p'_{i1})$ . Ainda, introduzindo a seguinte matriz  $\mathbf{M}_i$  definida pela Equação 2.11.

$$\mathbf{M}_i = \begin{bmatrix} 1 & 0 & p'_{i0} & -p'_{i1} \\ 0 & 1 & p'_{i1} & p'_{i0} \end{bmatrix} \quad (2.11)$$

Utilizando a definição do quadrado da norma  $\|\mathbf{a}\|_{\mathbf{C}}^2$  podemos expandir a Equação 2.10 em função da matriz  $\mathbf{M}$  e do vetor extendido quadridimensional  $\mathbf{x}$  como segue.

$$\begin{aligned} & \sum_{i=0}^N (\mathbf{M}_i \mathbf{x} - \mathbf{\Pi}_i)^T \mathbf{C}_i (\mathbf{M}_i \mathbf{x} - \mathbf{\Pi}_i) = \\ & \sum_{i=0}^N (\mathbf{x}^T \mathbf{M}_i^T \mathbf{C}_i \mathbf{M}_i \mathbf{x} + \mathbf{\Pi}_i^T \mathbf{C}_i \mathbf{\Pi}_i - 2\mathbf{\Pi}_i^T \mathbf{C}_i \mathbf{M}_i \mathbf{x}) \end{aligned} \quad (2.12)$$

Como estamos interessados em encontrar o valor mínimo da variável de otimização  $\mathbf{x}$ , podemos desconsiderar os termos que independem de  $\mathbf{x}$  na Equação 2.12. Assim,

reescrevendo a Equação para exprimir a variável de interesse obtêm-se a Equação 2.13.

$$\mathbf{x}^T \left( \sum_{i=0}^N \mathbf{M}_i^T \mathbf{C}_i \mathbf{M}_i \right) \mathbf{x} + \left( \sum_{i=0}^N -2 \mathbf{\Pi}_i^T \mathbf{C}_i \mathbf{M}_i \right) \mathbf{x}. \quad (2.13)$$

Definindo agora, as matrizes  $\mathbf{M}$ ,  $\mathbf{g}$  como sendo:

$$\begin{aligned} \mathbf{M} &= \mathbf{M}_i^T \mathbf{C}_i \mathbf{M}_i \\ \mathbf{g} &= \mathbf{\Pi}_i^T \mathbf{C}_i \mathbf{M}_i \end{aligned}$$

Pode-se reescrever a Função objetivo de maneira compacta como dado pela Equação 2.14.

$$\mathbf{x}^T \mathbf{M} \mathbf{x} + \mathbf{g}^T \mathbf{x} \quad (2.14)$$

Desta forma, resta agora somente enunciar o problema de minimização reescrito para o espaço  $\mathbb{R}^4$ . Para isso, devemos considerar a restrição dada no começo da derivação que estabelece que  $x_1^2 + x_2^2 = 1$ . Matricialmente, esta restrição trata-se de uma forma quadrática envolvendo o vetor  $x^* = [x_1 \ x_2]$  em que a matriz é a identidade  $\mathbf{I}_{2 \times 2}$ . Assim, introduzindo a matriz  $\mathbf{W}$ , dada pela Equação 2.15, pode-se enunciar o problema de minimização completo conforme a Equação 2.16.

$$\mathbf{W} = \begin{bmatrix} \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \mathbf{I}_{2 \times 2} \end{bmatrix} \quad (2.15)$$

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{x}^T \mathbf{M} \mathbf{x} + \mathbf{g}^T \mathbf{x} \\ \text{sujeito a} \quad & \mathbf{x}^T \mathbf{W} \mathbf{x} = 1 \end{aligned} \quad (2.16)$$

O problema de minimização dado pela Equação 2.10 é um problema clássico que envolve a minimização de uma forma quadrática sob restrições quadráticas de igualdade. A solução mais simples desse problema é a inclusão das restrições na função objetivo por intermédio de multiplicadores de lagrange. Desta forma, escrevendo a função lagrangiana, como descrito em (CENSI, 2008), obtêm-se:

$$\mathcal{L}(x, \lambda) = \mathbf{x}^T \mathbf{M} \mathbf{x} + \mathbf{g}^T \mathbf{x} + \lambda (\mathbf{x}^T \mathbf{W} \mathbf{x} - 1) \quad (2.17)$$

Nesses termos, a condição de optimalidade da solução define que  $\partial \mathcal{L} / \partial \mathbf{x} = 0$ . Derivando a Equação 2.17 com respeito à variável de otimização  $\mathbf{x}$  obtem-se:

$$\mathbf{x} = -(2\mathbf{M} + 2\lambda\mathbf{W})^{-T} \mathbf{g} \quad (2.18)$$

Resta agora a resolução dos multiplicadores de lagrange para a obtenção da solução em termos de  $\mathbf{x}$ , para isso, deve-se conectar a Equação 2.18 na restrição da Equação 2.16. O resultado desse procedimento é um polinômio de grau 4 na variável  $\lambda$ . Por motivos de

brevidade o desenvolvimento desta etapa foi omitido no texto. Para um desenvolvimento completo da solução do problema de minimização consulte (CENSI, 2008). O resultado que estamos interessados para os multiplicadores de lagrange é apresentado na longa Equação 2.19.

$$\lambda^2 4\mathbf{g}^T \begin{bmatrix} \mathbf{A}^{-1}\mathbf{B}\mathbf{B}^T\mathbf{A}^{-T} & -\mathbf{A}^{-1}\mathbf{B} \\ \text{(symm)} & \mathbf{I} \end{bmatrix} \mathbf{g} + \lambda 4\mathbf{g}^T \begin{bmatrix} \mathbf{A}^{-1}\mathbf{B}\mathbf{S}^A\mathbf{B}^T\mathbf{A}^{-T} & -\mathbf{A}^{-1}\mathbf{B}\mathbf{S}^A \\ \text{(symm)} & \mathbf{S}^A \end{bmatrix} \mathbf{g} + \mathbf{g}^T \begin{bmatrix} \mathbf{A}^{-1}\mathbf{B}\mathbf{S}^A\mathbf{B}^T\mathbf{A}^{-T} & -\mathbf{A}^{-1}\mathbf{B}\mathbf{S}^A\mathbf{T}\mathbf{S}^A \\ \text{(symm)} & \mathbf{S}^A\mathbf{T}\mathbf{S}^A \end{bmatrix} \mathbf{g} = [\det(\mathbf{D} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B} + 2\lambda\mathbf{I})]^2 \quad (2.19)$$

Em que:

$$\mathbf{S}^A = \det(\mathbf{D} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})(\mathbf{D} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1} \quad (2.20)$$

Resolvendo a Equação 2.19 para a variável  $\lambda$  obtem-se quatro raízes que, por sua vez, são utilizadas para encontrar o valor ótimo da variável de otimização  $\mathbf{x}$ .

### 2.3.2 Definição do algoritmo

Utilizando o resultado do problema de minimização derivado na seção anterior, pode-se descrever o algoritmo completo pIICP, como apresentado abaixo (retirado de (CENSI, 2008)).

**Data:** Nuvens de pontos  $\mathbf{p}_i$  e  $\mathbf{p}'_i$

**Result:** matriz de rotação  $\mathbf{R}$  e vetor de translação  $\mathbf{T}$

Realizar a transformação de corpo rígido de acordo com a transformação de corpo rígido estimada atual  $(\mathbf{R}_k, \mathbf{T}_k)$ ,  $\mathbf{p}_i^w = \mathbf{R}_k\mathbf{p}_i + \mathbf{T}_k$  ;

Para cada ponto em  $\mathbf{p}_i^w$  encontrar os dois pontos mais próximos na nuvem  $\mathbf{p}'_i$ , identificando seus índices como  $\mathbf{j}_1^i$  e  $\mathbf{j}_2^i$  ;

Construir a estrutura  $\mathbf{C}_k$  composta de triplas  $(i, j_1^i, j_2^i)$  ;

Minimizar a função dada pela Equação

$J([\mathbf{R}_{k+1}, \mathbf{T}_{k+1}], \mathbf{C}_k) = \sum_i (n_i^T [\mathbf{R}_k\mathbf{p}'_i + \mathbf{T}_k - \mathbf{p}'_{j_1^i}])^2$  utilizando a metodologia descrita na seção de Derivação do Algoritmo;

**Algoritmo 4:** Algoritmo pIICP para o cálculo da matriz de rotação  $R$  e o vetor de translação  $T$  entre duas nuvens.

A estrutura  $\mathbf{C}_k$  estabelece, para um ponto de índice  $i$  na nuvem  $\mathbf{p}'_i$ , os dois pontos mais próximos, de índices  $j_1^i$  e  $j_2^i$ , na nuvem  $\mathbf{p}_i^w$ . A partir da definição do algoritmo fica claro a necessidade de um chute inicial para a transformação.

### 3 Sistema de calibragem de Sensores

Fazendo uso do conhecimento desenvolvido até agora, nomeadamente sobre a modelagem do sensor LIDAR e os algoritmos de registro de superfície, pode-se agora apresentar o desenvolvimento em direção à concepção de um sistema capaz de monitorar as leituras em tempo real do sensor Laser a fim de realizar a calibragem do mesmo. Para introduzir a solução implementada para a calibragem, considere inicialmente o exemplo apresentado pelas Figuras 21 e 22. A Figura 21 apresenta um ambiente descrito em corte de vista superior e o esquema de um sensor instalado no ambiente. Os pontos vermelhos representam as leituras do sensor instalado. A Figura 22 representa o mesmo ambiente e o sensor, após este sofrer uma perturbação no ângulo de orientação e na posição. Note que como consequência da perturbação, o polígono de cobertura do sensor difere do polígono de cobertura inicial.

Estamos interessados em quantificar a Transformação de Corpo Rígido que relaciona a nova posição do Sensor com relação à posição inicial deste. Como estabelecido no capítulo de introdução deste trabalho, o sistema de Localização e Navegação possui conhecimento apenas da posição pré-perturbação do sensor e é com base nesta posição que é realizada a localização do robô. Desta forma, a saída esperada do sistema de calibragem para esta situação é a nova posição do sensor após a perturbação, com vistas a mitigar o erro de localização introduzido pela perturbação do sensor.

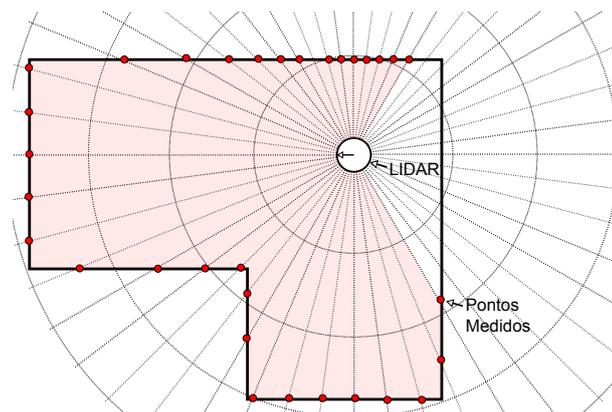


Fig. 21 – Sensor laser alocado em um ambiente. Em vermelho os pontos medidos

Considere agora as leituras dos dois sensores, na situação inicial e pós-perturbação, como apresentado na Figura 23. Deve-se observar que estas leituras estão apresentadas no frame de referência do sensor em sua posição inicial. Desta maneira, fica evidente a relação entre as duas nuvens de pontos, sendo estas relacionadas por uma transformação de corpo rígido. Pode-se utilizar a transformação de corpo Rígido ( $\mathbf{R}$ ,  $\mathbf{T}$ ) que relaciona as duas nuvens para estimar o movimento realizado pelo sensor. Para verificar a relação entre

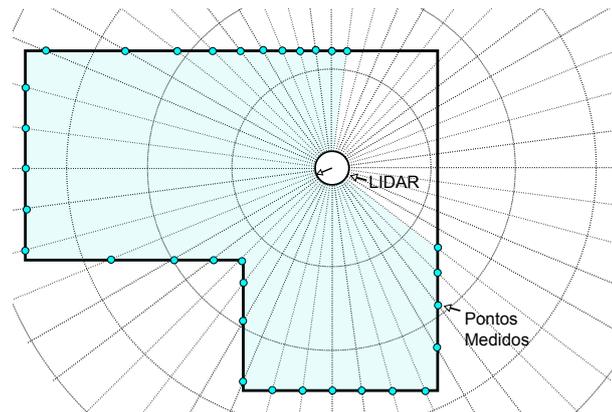


Fig. 22 – Deslocamento na posição e orientação angular do sensor.

a rotação das nuvens de pontos e a mudança de posição do sensor, considere a situação apresentada na Figura 24. Na Figura 24 observa-se que o movimento relativo do sensor faz com que haja um movimento respectivo na perspectiva do sensor, alterando a orientação das leituras de maneira equivalente. Assim, a quantificação do movimento relativo das medidas do sensor na posição pós-perturbação com relação à posição inicial, fornece uma estimativa adequada para a quantificação do movimento do sensor. Fazendo uso desta definição, pode-se empregar os algoritmos de simulação do sensor e de registro de superfície (desenvolvidos nos Capítulos 1 e 2, respectivamente) para atualizar o conhecimento da posição do sensor após a identificação de uma perturbação.

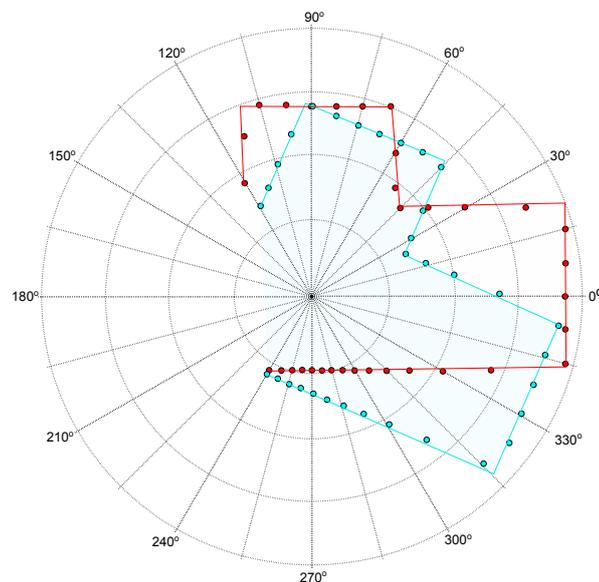


Fig. 23 – Nuvens de pontos do sensor nas duas posições.

O Sistema computacional desenvolvido pode ser sintetizado pelo Algoritmo 5. Este algoritmo consiste na obtenção da nova posição do sensor após uma perturbação na orientação e na posição. Observa-se que o algoritmo faz uso da leitura do sensor  $\mathbf{y}(t)$ , da

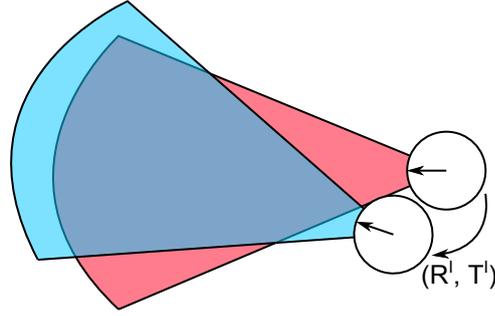


Fig. 24 – Sobreposição do campo de visão do sensor na situação de pré e pós perturbação.

última posição conhecida  $\mathbf{P}_s^k$ , referente à posição do sensor anterior a perturbação e do mapa  $\mathbf{X}_i$ , do qual é extraída a leitura simulada do sensor  $\mathbf{y}'(\mathbf{t})$  utilizando a última posição conhecida do sensor. Caso a leitura simulada seja diferente da leitura obtida do sensor, ou seja, o resíduo entre as duas nuvens seja superior a um nível de threshold pré-definido, o algoritmo estima a Transformação de Corpo Rígido que relaciona as duas nuvens e atualiza o conhecimento da posição do sensor para a nova pose  $\mathbf{P}_s^{k+1}$ . A atualização do conhecimento da pose do sensor é realizada iterativamente até que o resíduo entre as duas nuvens seja menor que o nível de threshold pré-definido.

**Data:** Leitura do sensor  $\mathbf{y}(\mathbf{t})$ , última posição conhecida do sensor  $\mathbf{P}_s^k = (p_x^s, p_y^s, \theta_s)$  e mapa do ambiente  $X_k$

**Result:** Nova posição conhecida  $\mathbf{P}_s^{k+1}$

Atribuir  $\mathbf{P}_s^{k+1} = \mathbf{P}_s^k$  ;

Gerar uma leitura sintética  $\mathbf{y}'(\mathbf{t})$  relativa a posição  $\mathbf{P}_s^k$  fazendo uso do Algoritmo 1 ;

Calcular o resíduo  $\Delta y = \|\mathbf{y}'(\mathbf{t}) - \mathbf{y}(\mathbf{t})\|_2^2$  ;

**while**  $\Delta y \geq threshold$  **do**

    Encontrar a transformada  $(\mathbf{R}, \mathbf{T})$  que relaciona as duas nuvens  $(\mathbf{y}(\mathbf{t})$  e  $\mathbf{y}'(\mathbf{t}))$ , usando o Algoritmo 4;

    Aplicar a transformada  $(\mathbf{R}, \mathbf{T})$  na pose  $\mathbf{P}_s^{k+1}$  ;

    Atualizar  $\mathbf{y}'(\mathbf{t})$  com uma leitura sintética na posição  $\mathbf{P}_s^{k+1}$  ;

    Calcular o resíduo  $\Delta y = \|\mathbf{y}'(\mathbf{t}) - \mathbf{y}(\mathbf{t})\|_2^2$  ;

**end**

**Algoritmo 5:** Algoritmo para a calibragem de um sensor laser LIDAR.

A definição do parâmetro de threshold é realizada a partir da experimentação do sistema de Calibragem e deve levar em consideração os níveis de ruído presentes na medida do sensor. Para definição deste nível de threshold considere a obtenção de uma medida do sensor laser em uma posição perfeitamente conhecida. O resíduo entre a medida obtida e uma leitura sintética gerada computacionalmente a partir do mapa e da posição conhecida resulta em uma estimativa da magnitude deste threshold, considerando que nesta situação o Algoritmo não deve atualizar a posição do sensor (ou seja, o sensor não sofreu qualquer

perturbação e a fonte do resíduo é somente devido à incerteza da medida do sensor). A estrutura geral do sistema implementado esta apresentada na Figura 25. Observa-se que o Sistema de calibragem faz uso do Algoritmo de Simulação de sensores (Algoritmo 1) e do Algoritmo pIIcp (Algoritmo 5).

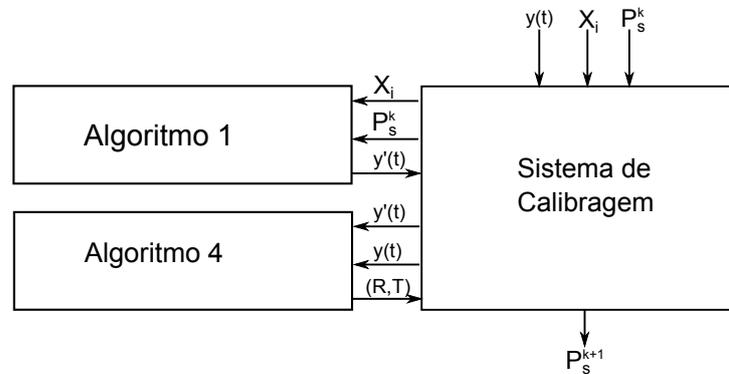


Fig. 25 – Diagrama de blocos do sistema desenvolvido.

Deve-se observar que o Sistema apresentado na Figura 25 deve ser alimentado continuamente ao longo da operação do sistema com as medidas do sensor laser, bem como com a posição em que acredita-se que o sensor esteja instalado. O sistema de Calibragem é responsável por aferir a confiabilidade das medidas em relação à posição declarada e, em caso de divergência entre a leitura esperada e a leitura obtida pelo sensor, atualizar o conhecimento da nova posição do sensor.

### 3.1 Sistema Computacional

Fazendo uso do Algoritmo 5 realizou-se a síntese de um Sistema Computacional capaz de realizar a interface entre a criação de mapas e a calibragem de sensores a partir da simulação do mesmo. O Sistema desenvolvido possui as seguintes características:

- Criação de mapas em tempo real;
- Possibilidade de Importação de mapas previamente salvos;
- Simulação de leituras do sensor laser LIDAR para diversos parâmetros do sensor;
- Simulação de perturbações na posição e na orientação angular do sensor;
- Calibragem do Sensor LIDAR;

A interface do Sistema desenvolvido esta apresentada na Figura 26. O Sensor é representado em azul e os pontos medidos estão ao final de cada segmento representando um feixe do sensor. O mapa é apresentado através dos segmentos pretos. O botão "Construir

mapa" permite ao usuário a descrição do mapa diretamente na tela utilizando o cursor. O botão "Carregar mapa" realiza a importação de um mapa previamente salvo descrito por uma lista de segmentos de retas definidos por um ponto inicial  $P_i$  e um ponto final  $Q_i$ . A posição do sensor, bem como a sua orientação angular podem ser alteradas arrastando o sensor para a posição desejada.

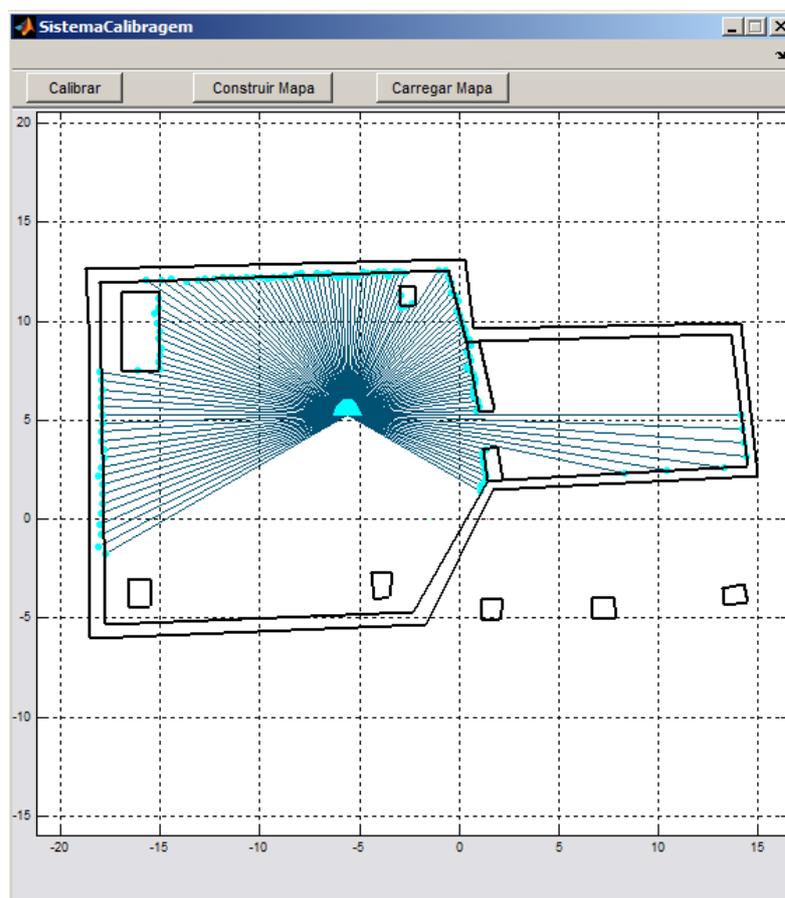


Fig. 26 – Interface da aplicação desenvolvida.

Para simular o comportamento de perturbação utilizou-se de dois sensores virtuais. A Figura 28 apresenta esta situação. O Sensor Azul representa o sensor em sua posição real após a perturbação, enquanto que o sensor vermelho representa a última posição conhecida do sensor. Desta forma, o sistema de calibragem tem acesso à informação de posição e orientação angular do sensor vermelho e às leituras do sensor Azul, sendo desconhecido à posição e orientação angular do sensor Azul (parâmetro que o sistema de calibragem deve estimar e atualizar). Neste cenário, a posição conhecida do sensor, identificada pelo sensor vermelho, deve ser continuamente atualizada até a posição e orientação do sensor coincidir com a posição real do mesmo.

Observa-se a partir da Figura 28 que as leituras do sensor real apresentam incerteza na medida, calculada utilizando o método de estimativa de variância de ruído apresentado no Capítulo 1. Em contrapartida, as leituras geradas a partir da posição conhecida (sensor

Vermelho) não possuem ruído, fato que pode ser explicado pela natureza deste Sensor. O Sensor Vermelho é uma virtualização do sensor em sua última pose conhecida, representando as medidas que o sensor obteria caso este estivesse instalado na posição que o sistema de Localização e Navegação acredita que o sensor esteja localizado (cenário pré-perturbação). Desta forma, podemos gerar um conjunto de medidas sintéticas sem ruído para o sensor Vermelho, visto que são estes dados que são comparados com as leituras do sensor pelo Algoritmo 5 para verificar a calibragem do sensor. A possível Inclusão do ruído nas leituras do Sensor Vermelho representaria uma fonte maior de incerteza, prejudicando o desempenho do Algoritmo pIICP.

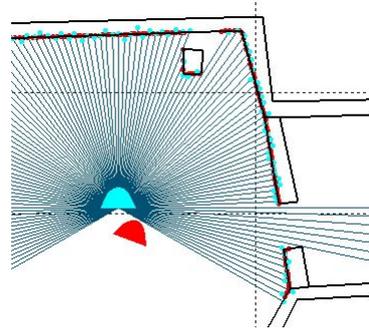


Fig. 27 – Estratégia para a simulação de perturbações no sensor.

## 3.2 Noções sobre Convergência

Considerando o Algoritmo 5, podemos observar algumas propriedades quanto à convergência do método. O polígono de cobertura dos sensores laser LIDAR comerciais apresenta cobertura limitada, variando entre 200 e 240 graus. Esta limitação impõe um problema para a calibragem do sensor utilizando a metodologia deste trabalho. A limitação encontra-se no fato de que o Algoritmo responsável por calcular a transformação roto-translacional entre as duas nuvens de pontos (Algoritmo 4) assume que os dois conjuntos de dados estão relacionados por uma transformação de corpo rígido, fato que não pode ser verificado se a perturbação considerada resultar em um polígono de cobertura do sensor completamente diferente em relação ao polígono de cobertura inicial. Nesta cenário, o polígono de cobertura do sensor após a perturbação representa uma porção completamente distinta do ambiente, resultando na não convergência do Algoritmo. A Figura apresentada abaixo representa este efeito.

Observa-se na Figura 28 que a maior parte do polígono de cobertura do sensor após a perturbação não coincide com o polígono de cobertura na situação anterior à perturbação. Como resultado desta disparidade, observa-se que as duas nuvens de pontos, medidas do sensor laser e medidas sintéticas obtidas a partir do conhecimento da posição pré-perturbação do sensor e do mapa, não representam a mesma porção do ambiente e,

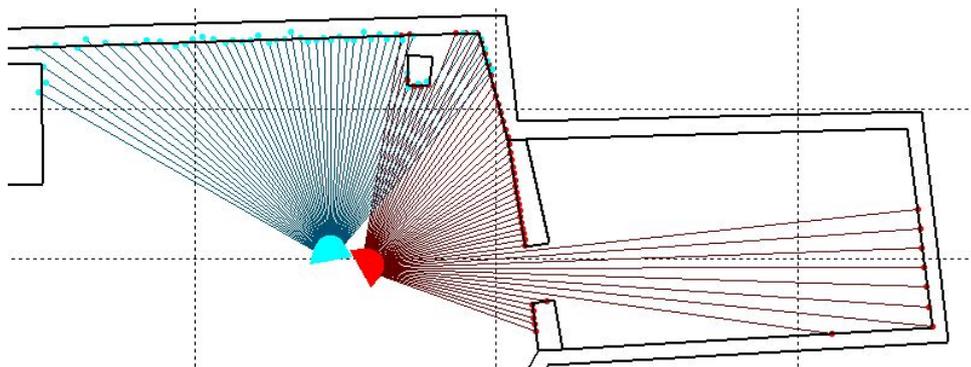


Fig. 28 – Não sobreposição do campo de visão do sensor na posição inicial e após a perturbação.

portanto, não podem ser relacionadas através de uma transformação de corpo rígido. O caso apresentado acima é um limitador do desempenho do método de calibragem. Em situações normais, em que o sistema de calibragem atua continuamente atualizando a posição do sensor, deve-se observar que as perturbações são continuamente rastreadas e a não convergência do Algoritmo esta condicionada somente a situações em que a perturbação é mais rápida que a atuação do sistema de calibragem.

Ainda, devemos observar que o número de iterações necessárias para a convergência do Algoritmo 5 aumenta ao passo que a diferença entre os polígonos de cobertura da posição inicial e da posição pós-perturbação aumenta. No entanto, em caso de convergência, cada iteração do Algoritmo aumenta a área comum dos polígonos de cobertura, resultando no mesmo polígono de cobertura para o caso em que o sensor encontra-se calibrado. Fazendo uso desta definição de convergência podemos agora testar o desempenho do Sistema de Calibragem para um estudo de caso estatístico. A próxima seção apresenta este desenvolvimento.

### 3.3 Estudo de Caso

Para a validação do Sistema de Calibragem realizou-se um estudo de caso para a calibragem de um sensor Laser Lidar com parâmetros ajustados ao modelo (*URG\_047x*). A Tabela 1 sumariza os parâmetros adotados para o sensor.

Parâmetro	Valor
Campo de visão	240°
Resolução Angular	0.36°
Frequência	10Hz/Scan
Número de pontos	667
Incerteza	±5%

O Ambiente considerado está apresentado na Figura 2. Observe que o ambiente

contem estruturas pequenas, representando pilares e portas. A escolha deste ambiente é justificada pelo fato de que o ambiente contém estruturas grandes, representando baixas frequências no espectro da medida do sensor, e estruturas pequenas, representando componentes espectrais de alta frequência da medida. A presença de componentes espectrais de altas frequências é importante para avaliar o comportamento do estimador de ruído frente à tais estruturas. Desta forma, o processo de filtragem deve ser condicionado ao conhecimento do mapa, com vistas a não classificar como ruído detalhes importantes do mapa.

Como estratégia de validação adotou-se uma configuração de testes composta de três cenários. Os cenários de teste estão descritos abaixo.

- Cenário 1: 1000 perturbações aleatórias na posição do sensor com uma região de alocação circular das perturbações, sendo esta região limitada por um raio de 2,5 unidades de medida;
- Cenário 2: 1000 perturbações aleatórias na orientação angular do sensor, com ângulo de perturbação limitado em 90 graus;
- Cenário 3: 1000 perturbações aleatórias na posição e na orientação angular do sensor com as restrições dos cenários anteriores.

Além dos 3 cenários, fixou-se o número máximo de 10 iterações para o Algoritmo 5 e um threshold adequado à incerteza das medidas obtidas pelo Sensor. O número fixo de iterações deve ser definido devido à necessidade de resposta em um tempo finito do algoritmo. A condição ideal é que a cada leitura disponível do sensor possa ser comparada a fim de verificar a calibragem do sensor.

Os resultados de convergência para os três cenários estão apresentados nas Figuras 29, 30 e 31. Observa-se que para o caso de perturbação na posição, o sistema não atingiu o limite de 10 iterações em nenhuma execução e convergiu em mais de 50% das execuções em até 3 iterações. Esta característica deve-se ao fato de que o algoritmo 5, utilizado para o cálculo da roto-translação converge mais rapidamente para simples translações. Para o caso de perturbação de natureza rotacional, observa-se um aumento no número de iterações necessárias para a convergência e ainda algumas execuções que atingiram o limite de iterações. Para o cenário das perturbações roto-translacionais observa-se um aumento no número de execuções que atingiram o número máximo de iterações foi de 13%.

Com respeito às execuções que atingiram o número máximo de iterações podemos afirmar que estas estão convergindo e convergiram em mais algumas iterações ou divergiram. A Figura 32 apresenta o percentual acumulado de execuções para cada iteração. O gráfico confirma a observação de que as execuções do cenário 1 convergiram com um número menor de iterações.

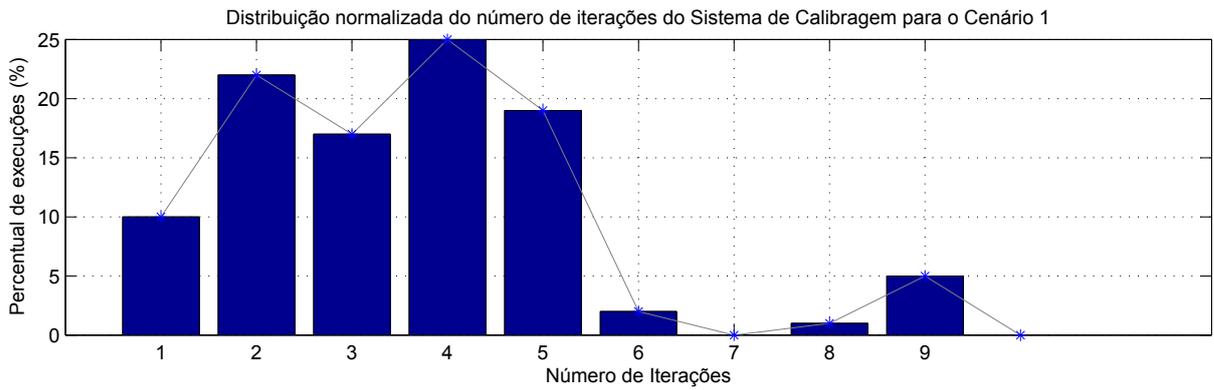


Fig. 29 – Distribuição do número de iterações para o cenário 1.

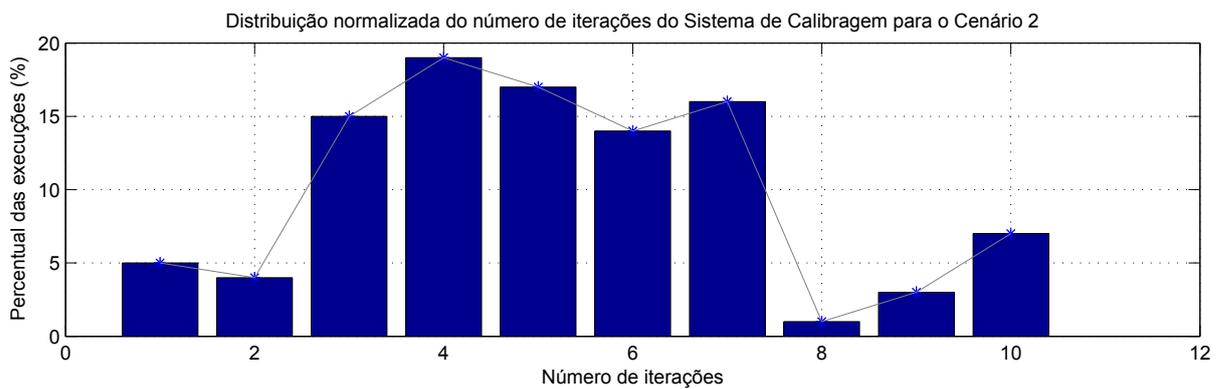


Fig. 30 – Distribuição do número de iterações para o cenário 2.

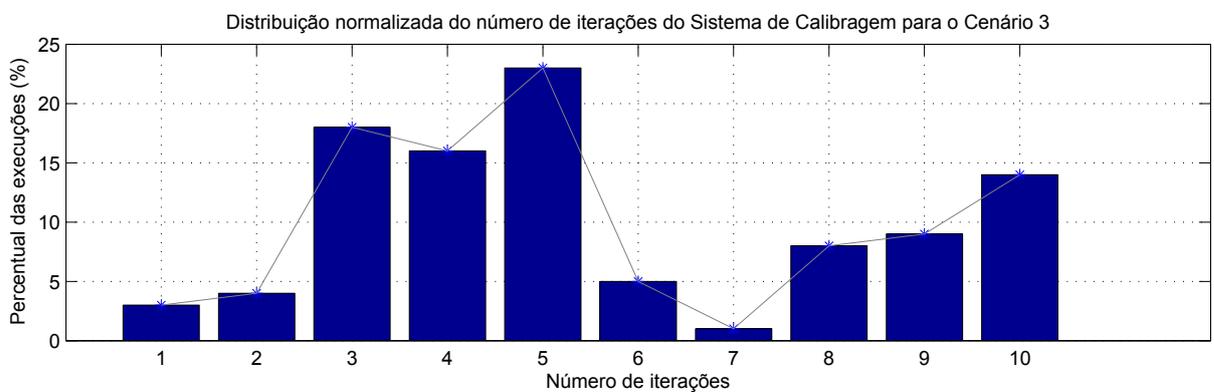


Fig. 31 – Distribuição do número de iterações para o cenário 3.

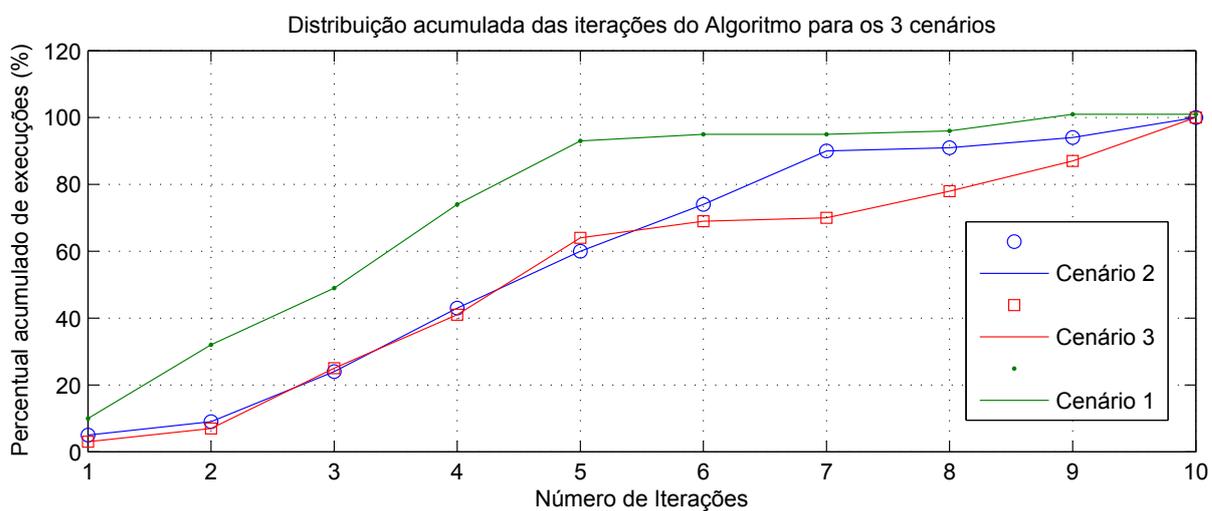


Fig. 32 – Distribuição acumulado do número de iterações para os três cenários.

# Conclusão

O desenvolvimento do sistema de calibragem mostrou-se como um importante componente dentro de um sistema robótico, principalmente por garantir confiabilidade no posicionamento dos sensores utilizados para localização e navegação de um robô. O monitoramento das leituras do sensor permite identificar quando as leituras do sensor diferem das leituras esperadas pelo sistema, com base no conhecimento da posição do sensor e do mapa do ambiente. A utilização de uma ferramenta de calibragem faz com que o erro de localização em função da referência incorreta das posições dos sensores instalados no ambiente seja minimizada.

O sistema desenvolvido teve uma boa performance considerando o estudo de caso realizado. No cenário mais complexo, em que realizou-se a simulação de 1000 perturbações de natureza roto-translacional, o mesmo convergiu para 86% dos casos. Ainda, deve-se ressaltar que dentre as execuções que atingiram o limite máximo de iterações existem aquelas em que a posição conhecida está convergindo para a posição real do sensor, sendo necessário mais iterações. Além disso, a observação de que uma perturbação atuante na posição do sensor é continuamente rastreada pelo sistema de calibragem permite restringir a operação do sistema somente à variações pequenas na posição e na orientação angular do mesmo. Com isso, espera-se que o desempenho da calibragem em uma situação real seja superior ao desempenho obtido durante a simulação.

Os algoritmos explorados permitem a calibragem do sensor laser LIDAR fazendo uso de informações usualmente disponíveis. As medidas obtidas pelo sensor em dado instante, o conhecimento da posição inicial do sensor e a descrição do ambiente em que o sensor está instalado são informações de fácil aquisição e podem ser utilizadas diretamente pelos algoritmos desenvolvidos. Deve ser ressaltado que a necessidade do conhecimento e da descrição exata do ambiente no qual o sensor será instalado representa uma limitação do sistema, fazendo com que este não possa ser utilizado em ambientes externos, por exemplo. Ainda, o conhecimento da posição inicial requer que o utilizador ajuste este parâmetro com base na descrição em planta do ambiente. Esta tarefa deve ser realizada através de tentativa e erro, verificando a descrição de posição exata que resulta em um sistema calibrado, visto que o sistema é capaz de identificar quando a leitura obtida através do sensor coincide com a leitura esperada pela posição descrita.

A simulação do sensor, a fim de gerar leituras simuladas do sensor em um ambiente apresentou tempo de execução dependente do número de estruturas presentes no mapa que descreve o ambiente. A estratégia de aceleração do algoritmo de geração de leituras permite reduzir o número de buscas e de operações realizadas para gerar uma leitura completa

do sensor. Os algoritmos de registro de superfície estudados mostraram-se eficientes na estimativa da transformação de corpo rígido que relaciona a posição inicial do sensor e a posição pós-perturbação. O algoritmo pIIICP apresenta um menor número de iterações e é mais robusto em relação ao ruído na medida que a implementação direta do Algoritmo ICP.

A definição do parâmetro de limiar de comparação (Algoritmo 5) entre a leitura obtida pelo sensor e a leitura esperada gerada sinteticamente pelo sistema deve obedecer as características de ruído do sensor. Desta maneira, a implementação de um sistema adaptativo que leve em consideração as variações da incerteza das medidas pode trazer ganhos para o sistema, especialmente na distinção correta entre situações em que o sensor encontra-se calibrado e situações em que há a necessidade de calibragem.

Pode-se ressaltar ainda a possibilidade de utilização do sistema de calibragem como um sistema de Localização para veículos móveis. Para tanto considere um robô com um sensor laser LIDAR instalado em seu corpo movimentando-se em um ambiente interno perfeitamente conhecido e delimitado. Se considerarmos a posição inicial do veículo em analogia à posição inicial do sensor e o movimento realizado análogo à perturbação e ainda assumindo que a velocidade de execução do algoritmo de calibragem e da obtenção de leituras do sensor seja comparável a velocidade da dinâmica do veículo, é possível rastrear continuamente a posição deste ao longo de uma trajetória. A hipótese principal que deve ser observada é a de que a cada execução do algoritmo a perturbação na posição do veículo deve ser pequena. Como desenvolvimento futuro, pretende-se explorar a expansão do sistema em trs situações:

- Calibragem de uma rede de sensores: A utilização de uma rede de sensores permite que a redundância de cobertura espacial da rede de sensores seja explorada para acelerar o processo de calibragem. A calibragem pode ser realizada em relação a um sensor para outro, tornando o sistema mais rápido.
- Calibragem dos sensores sem o conhecimento do mapa: Para este cenário faz-se necessário o emprego de métodos de estimativa de estrutura a partir de leituras. Tipicamente, esta situação envolve um sistema de SLAM (Simultaneous Localization and Mapping) e um veículo dotado de sensores de leitura a fim de realizar o mapeamento das estruturas. Após a realização do mapeamento do ambiente, o sistema desenvolvido neste trabalho pode ser utilizado para a calibragem dos sensores.
- Calibragem dos sensores sem o conhecimento do mapa e sem um chute inicial para a posição inicial: Esta situação representa o cenário mais desafiador, especialmente se considerarmos um ambiente pobre em propriedades únicas e com regiões de singularidade. Uma estratégia interessante para o desenvolvimento nesta linha de trabalho é a utilização de algoritmos combinatórios de otimização heurística (Simu-

lated Annealing, algoritmos genéticos) para a busca da posição inicial dos sensores em relação ao ambiente mapeado.

## Referências

- AMOR, B.; ARDABILIAN, M.; CHEN, L. New experiments on icp-based 3d face recognition and authentication. In: *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*. [S.l.: s.n.], 2006. v. 3, p. 1195–1199. ISSN 1051-4651. Citado na página 31.
- ARUN, K.; HUANG, T. S.; BLOSTEIN, S. D. Least-squares fitting of two 3-d point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-9, n. 5, p. 698–700, Sept 1987. ISSN 0162-8828. Citado 2 vezes nas páginas 34 e 37.
- CENSI, A. An icp variant using a point-to-line metric. In: *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. [S.l.: s.n.], 2008. p. 19–25. ISSN 1050-4729. Citado 6 vezes nas páginas 8, 34, 36, 37, 39 e 40.
- CHEN, Y.; MEDIONI, G. Object modeling by registration of multiple range images. In: *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. [S.l.: s.n.], 1991. p. 2724–2729 vol.3. Citado na página 32.
- FERREIRA, J.; VALE, A.; VENTURA, R. Vehicle localization system using offboard range sensor network. In: *Intelligent Autonomous Vehicle, 8th Symposium on*. [S.l.: s.n.], 2013. p. 102–107. Citado 4 vezes nas páginas 8, 14, 15 e 16.
- MAGREE, D.; JOHNSON, E. N. Combined laser and vision-aided inertial navigation for an indoor unmanned aerial vehicle. In: *American Control Conference (ACC), 2014*. [S.l.: s.n.], 2014. p. 1900–1905. ISSN 0743-1619. Citado na página 31.
- MOGHADAM, P.; WIJESOMA, W.; FENG, D. J. Improving path planning and mapping based on stereo vision and lidar. In: *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*. [S.l.: s.n.], 2008. p. 384–389. Citado na página 12.
- SAPUTRA, M. et al. Indoor human tracking application using multiple depth-cameras. In: *Advanced Computer Science and Information Systems (ICACSIS), 2012 International Conference on*. [S.l.: s.n.], 2012. p. 307–312. Citado 3 vezes nas páginas 8, 14 e 15.
- VEN, J. van de; RAMOS, F.; TIPALDI, G. An integrated probabilistic model for scan-matching, moving object detection and motion estimation. In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. [S.l.: s.n.], 2010. p. 887–894. ISSN 1050-4729. Citado na página 31.
- WANG, D. et al. Lidar scan matching ekf-slam using the differential model of vehicle motion. In: *Intelligent Vehicles Symposium (IV), 2013 IEEE*. [S.l.: s.n.], 2013. p. 908–912. ISSN 1931-0587. Citado na página 12.
- ZITOVÁ, B.; FLUSSER, J. Image registration methods: a survey. *Image and Vision Computing*, v. 21, n. 11, p. 977 – 1000, 2003. ISSN 0262-8856. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0262885603001379>>. Citado 3 vezes nas páginas 8, 31 e 32.

□