

SISTEMA DE LOCALIZAÇÃO DE UM ROBÔ MÓVEL BASEADO EM FILTRO DE KALMAN ESTENDIDO PARA SLAM COM KINECT EM AMBIENTES INTERNOS

DANILO GIACOMIN SCHNEIDER*, MARCELO RICARDO STEMMER*

**Departamento de Automação e Sistemas
Universidade Federal de Santa Catarina
Florianópolis, Santa Catarina, Brasil*

Emails: `daniilo_gsch@hotmail.com`, `marcelo.stemmer@ufsc.br`

Abstract— In order to be entirely autonomous a robot needs some sort of environment representation in which it is present as well as its localization in this environment. Finding itself accurately and building a map of the environment allows the robot to reach predetermined goals. Localization is one of the most important tasks for autonomous navigation of mobile robots, where robustness and precision are fundamental characteristics. This task is based on the map of the environment. In contrast, building this map depends on the precise location of the mobile robot. In unknown environments this is a conflicting but correlated process. This paper describes the implementation of an Extended Kalman Filter using ROS to perform the fusion of three different sensors and estimate the position and orientation of the robot more accurately and robustly and, consequently, resulting in a more accurate mapping of the environment.

Keywords— Extended Kalman Filter, EKF, Mobile Robot, Rgbdslam, SLAM, Kinect, ROS.

Resumo— Para que um robô seja inteiramente autônomo ele precisa de alguma forma de representação do ambiente no qual ele está presente, assim como a sua localização em relação a este ambiente. Localizar-se com precisão e construir um mapa do ambiente permite que o robô possa alcançar metas pré-determinadas. A localização é uma das tarefas mais importantes para a navegação autônoma de robôs móveis, onde robustez e precisão são características fundamentais. Essa tarefa baseia-se no mapa do seu ambiente. Em contrapartida, a construção do mapa do ambiente depende da localização precisa do robô móvel. Em ambientes desconhecidos este é um processo conflitante, mas correlacionado. Este artigo apresenta a implementação, através do ROS, de um Filtro de Kalman Estendido para realizar a fusão de três sensores diferentes e estimar a posição e orientação do robô de forma mais robusta, precisa e, conseqüentemente, obter um mapeamento também mais fiel do ambiente.

Palavras-chave— Filtro de Kalman Estendido, EKF, Robô Móvel, Rgbdslam, SLAM, Kinect, ROS.

1 Introdução

A robótica é um ramo da ciência multidisciplinar, no qual ocorre um desenvolvimento expressivo em diversas áreas de pesquisa, a maior diversidade e crescimento de suas aplicações sugerem, cada vez mais, a importância deste tema em aplicações de sistemas produtivos, de serviços e do lazer (Holtkamp and de Jong, 2006).

No caso de aplicações que envolvam robôs móveis autônomos, é um requisito fundamental que eles consigam se localizar e obter uma representação do ambiente no qual se encontram, de modo que seja possível a navegação e realização das suas atividades. A localização é uma das características mais importantes para a navegação autônoma de robôs móveis, assim, robustez e precisão são fatores fundamentais. Em geral, essa tarefa baseia-se no mapa do ambiente em questão. Em contrapartida, a construção do mapa do local depende da localização precisa do robô. Em ambientes desconhecidos este é um processo conflitante, mas correlacionado, conhecido como localização e mapeamento simultâneos (SLAM) (Lin et al., 2012).

Uma grande variedade de sensores vêm sendo utilizados para realizar a tarefa de localização (Bicchi et al., 2004), como o sistema de posicionamento global (GPS), *laser range finder* (LRF), unidade de medida inercial (IMU), entre outros

que, ocasionalmente, podem ser caros. Independentemente dos sensores escolhidos, sempre existirão restrições, erros e ruídos de medição. Através da utilização de câmeras e visão computacional, robôs móveis são capazes de obter informações mais detalhadas do ambiente, aumentando a versatilidade e segurança nas operações do robô, como detecção e localização de objetos ou obstáculos, úteis no desempenho de tarefas de manipulação ou navegação, por exemplo. Por outro lado, se faz necessário um maior poder computacional para lidar com tantos detalhes, o que pode não ser viável em aplicações mais simples.

Câmeras RGB-D, como o Microsoft Kinect, proporcionam informações de cores e profundidade por pixel em imagens sincronizadas e em tempo real. Usando um sensor de profundidade, essas câmeras evitam a complexidade da computação visual correspondente para estimação de profundidade em combinações de imagens estéreo. Assim, sensores deste tipo vêm ganhando destaque em pesquisas relacionadas a localização e mapeamento, principalmente em ambientes internos onde são respeitadas as condições de iluminação e distância dos objetos.

1.1 Estado da Arte

Nas últimas décadas, o algoritmo do Filtro de Kalman Estendido (EKF) para estimar o movimento da câmera em tempo real é uma das principais técnicas para localização e mapeamento simultâneos (SLAM) em automação robótica e visão computacional (Davison et al., 2007; Civera et al., 2008; Montiel et al., 2006; Huang et al., 2017; Henry et al., 2010; Hervier et al., 2012; Ling et al., 2013). O processo EKF-SLAM procede exatamente como o EKF padrão utilizado para localização, com a diferença de utilizar um vetor de estados estendido (y_t) que contém não somente a pose do robô (x_t) como também a posição de todas as características no mapa (m_i), como mostrado na equação (1).

$$y_t = [x_t, m_0, \dots, m_{n-1}]^T \quad (1)$$

A cada iteração, tanto o estado quanto a matriz de covariância conjunta são atualizados, o que significa que o custo computacional cresce de maneira quadrática de acordo com o número de características. Outro problema do EKF-SLAM está na linearização feita pelo Filtro de Kalman Estendido, que é refletido pelo uso da matriz Jacobiana nas atualizações de movimento e medição. Infelizmente, tanto o movimento quanto o modelo de medição são tipicamente não-lineares, portanto, sua linearização pode ocasionar inconsistência ou divergência na solução. Além disso, o EKF-SLAM é sensível a associações incorretas de características, que acontecem quando o robô associa incorretamente a característica m_i com a característica m_j . Este problema se torna ainda mais importante no fechamento do ciclo (*loop closure*), ou seja, quando o robô retorna para reobservar características após um longo percurso (Siegwart et al., 2011).

Em trabalhos relacionados com SLAM 3D, Huang et al. (2017) propõe um sistema de voo autônomo para odometria visual e mapeamento utilizando o processo EKF-SLAM com Kinect. Henry et al. (2010) propõem uma estrutura de mapeamento RGB-D que constrói mapas 3D densos de ambientes internos usando TORO, uma ferramenta de otimização desenvolvida para SLAM. Hervier et al. (2012) construiu um mapa 3D preciso baseado na covariância estimada do ICP (*Iterative Closest Point*) e na fusão de dados do Kinect e um giroscópio por um EKF. Ling et al. (2013) propõe a utilização do algoritmo EKF-SLAM de modo iterado, relinearizando a equação de medida por meio da iteração de um máximo aproximado a posteriori (MAP), que é estimado em torno no estado atualizado, ao invés de depender somente do estado previsto. Nas abordagens de Huang et al. (2017), Henry et al. (2010) e Hervier et al. (2012), o EKF lineariza a predição de medida e todas as transformações no estado de predição usando expansão em séries, substituindo as matrizes Jacobianas por transformações lineares

no filtro de Kalman. Entretanto, tais linearizações assumem que os erros de predição podem ser bem aproximados por uma função linear. Caso essa condição não possa ser satisfeita, os erros serão propagados e podem resultar diretamente na divergência das estimativas.

1.2 Objetivo

O objetivo deste artigo é desenvolver um sistema de localização baseado em um filtro de Kalman estendido que realiza a fusão de dados de odometria visual, odometria dos *encoders* e a pose do robô obtida através do escaneamento de um laser 2D para a obtenção de uma pose absoluta mais precisa e robusta que será utilizada para o mapeamento 3D representado por nuvem de pontos (*point cloud*). Este sistema foi avaliado em um cenário real, onde um robô móvel (Pioneer 3-DX) foi montado com os sensores em questão para coleta dos dados, que foram processados remotamente em um computador para localização do robô e mapeamento do ambiente simultaneamente em tempo real utilizando ROS (*Robot Operating System*).

Neste projeto, não foi utilizado um vetor de estados como o descrito na equação (1), pela grande quantidade de características capturadas pelo Kinect, o poder computacional necessário para adicionar a localização de cada característica ao vetor seria muito grande para aplicações em tempo real. Métodos para realizar EKF-SLAM e outros algoritmos de SLAM probabilísticos com mapas 3D vêm sendo desenvolvidos, tipicamente dividindo o mapa em sub-mapas, mas, por enquanto, tais métodos estão fora do escopo deste projeto. O intuito aqui é realizar a fusão dos sensores de localização e usar o resultado no mapeamento.

2 Metodologia

2.1 Robot Operating System

O ROS é uma estrutura flexível para escrever software robótico. É uma coleção de ferramentas, bibliotecas e convenções que visa simplificar a tarefa de criar um comportamento robótico complexo e robusto em uma ampla variedade de plataformas. ROS está sob uma licença *open-source* e já existem diversos pacotes desenvolvidos pela comunidade para diversas finalidades, e.g. localização, SLAM, Kinect e controle do robô, que foram utilizados para o desenvolvimento deste projeto, entre eles: *RosAria*, *freenect_launch*, *sicktoolbox_wrapper*, *laser_scan_matcher*, *robot_localization*, *rtabmap_ros*, *teleop_twist_keyboard* e suas respectivas dependências.

A Fig. 1 apresenta um diagrama simplificado do sistema proposto nas experiências deste projeto. Comandos de velocidade linear e velocidade

angular de guinada ($\dot{x}, \dot{\gamma}$) são enviados ao robô via teclado. O pacote *RosAria* se encarrega de enviar esses valores ao controlador interno dos motores do robô e converter as leituras dos *encoders* em valores de odometria. Os escaneamentos feitos pelo laser são enviados ao computador que compara as varreduras para obtenção de uma pose absoluta estimada do laser, usamos os valores de odometria dos *encoders* para fornecer uma suposição para a posição atual do sensor toda vez que uma nova mensagem de varredura chegar, aumentando a velocidade de conversão e precisão do processo de estimação. Quando não há nenhum palpite disponível, uma suposição razoável (e amplamente usada) é que o sensor não se moveu (modelo de velocidade zero). O pacote *freenect_launch*, que possui as bibliotecas e drivers para comunicação com o Kinect, envia as imagens RGB-D ao computador que republica essas imagens através de um retransmissor (*Relay*), que foi utilizado para aliviar o consumo de banda da rede pois vários nós usam as mesmas imagens para suas funcionalidades (Odometria visual, mapeamento e visualização).

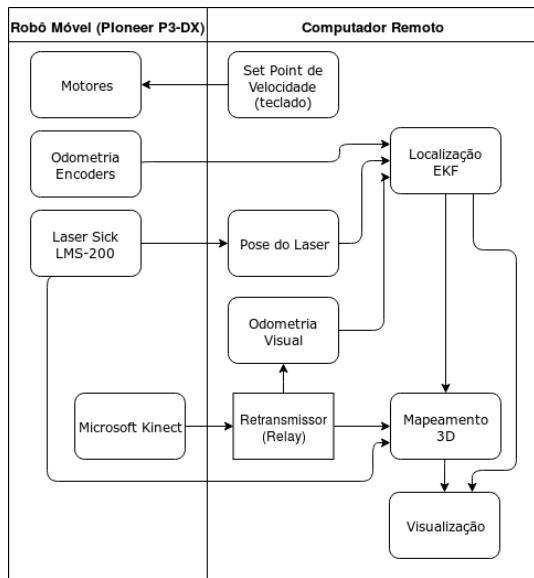


Figura 1: Diagrama simplificado do sistema de localização e mapeamento remotos usado no projeto

A fusão de sensores pelo algoritmo EKF foi aplicada pelo pacote *robot_localization* e é detalhada na próxima subseção deste artigo. A escolha desse método foi baseada em sua ampla aplicabilidade e por atender aos requisitos do sistema.

As etapas de odometria visual, mapeamento 3D e visualização são feitas pelo pacote *rtab-map-ros*, desenvolvido por Labbé and Michaud (2017), é uma abordagem SLAM RGB-D com um detector incremental de fechamento de ciclo baseado em aparências. Dentre as diversas possibilidades de configuração, a odometria visual e o dicionário visual foram configurados com o detector GFTT (*Good Features To Track*) (Shi

and Tomasi, 1993) para detectar as características na imagem e BRIEF (*Binary Robust Independent Elementary Features*) (Calonder et al., 2010) de 16 bytes como descritor binário dessas características. Este descritor binário pode ser usado sob a licença BSD. Embora menos descritivo do que os descritores tipo *float*, SIFT (Lowe, 1999) e SURF (Bay et al., 2006), para ambientes pequenos funciona bem. Em ambientes de grande escala, SURF ou SIFT são mais recomendados. Em testes realizados por Labbé and Michaud (2017), o modelo GFTT+BRIEF ofereceu maior confiabilidade entre todos os detectores de características binários disponíveis. Segundo eles, os detectores FAST ou ORB (Rublee et al., 2011) são mais rápidos que a GFTT, mas geram menos características boas para rastrear e os descritores FREAK (Alahi et al., 2012) e BRIEF fornecem resultados semelhantes. No mapeamento, as informações de varredura do laser também são utilizadas para estimação ICP 2D (*Iterative Closest Point*).

2.2 Algoritmo do Filtro de Kalman Estendido

A formulação do EKF é bem conhecida na literatura, o modelo utilizado por Moore and Stouch (2014) no desenvolvimento do pacote *robot_localization*, usado neste projeto, será detalhado aqui para mostrar as etapas importantes da implementação. O objetivo é estimar a pose 3D completa (6 DoF) e a velocidade de um robô móvel ao longo do tempo. Este processo pode ser descrito como um sistema dinâmico não-linear, descrito por

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}, \quad (2)$$

onde $\mathbf{x}_k$ é o vetor de estados do sistema (i.e., a posição e orientação 3D do robô) no tempo k , f é uma função não-linear de transição de estados e $\mathbf{w}_{k-1}$ é o ruído no processo, o qual é considerado ser distribuído normalmente (distribuição Gaussiana). No pacote *robot_localization* é possível configurar um vetor de estados de até 15 dimensões, composto pela posição 3D do veículo, orientação 3D, suas respectivas velocidades lineares e angulares e acelerações lineares. Valores angulares são expressados em ângulos de Euler: Rolagem (α), arfagem (β) e guinada (γ). Adicionalmente são recebidas medidas na forma

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k, \quad (3)$$

onde $\mathbf{z}_k$ é a medida no tempo k , h é o modelo não-linear do sensor que mapeia os estados no espaço de medição e $\mathbf{v}_k$ é o ruído de medida normalmente distribuído.

O primeiro passo do algoritmo é o da predição, mostrado nas equações (4) e (5), que projeta o estado atual estimado e suas covariâncias adiante no tempo:

$$\hat{\mathbf{x}}_k = f(\mathbf{x}_{k-1}). \quad (4)$$

$$\hat{\mathbf{P}}_k = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}. \quad (5)$$

Nessa aplicação, f é um modelo cinemático padrão derivado da mecânica Newtoniana. A covariância do erro estimada, $\hat{\mathbf{P}}$, é projetada por $\mathbf{F}$, a matriz Jacobiana de f e, em seguida, perturbada por $\mathbf{Q}$, a covariância do ruído do processo.

O próximo passo, de correção, é mostrado nas equações (6) a (8):

$$\mathbf{K} = \hat{\mathbf{P}}_k \mathbf{H}^T (\mathbf{H} \hat{\mathbf{P}}_k \mathbf{H}^T + \mathbf{R})^{-1}. \quad (6)$$

$$\mathbf{x}_k = \hat{\mathbf{x}}_k + \mathbf{K}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}_k). \quad (7)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}\mathbf{H})\hat{\mathbf{P}}_k(\mathbf{I} - \mathbf{K}\mathbf{H})^T + \mathbf{K}\mathbf{R}\mathbf{K}^T. \quad (8)$$

Primeiramente, o ganho de Kalman $\mathbf{K}$ é calculado a partir da matriz de observação $\mathbf{H}$, a covariância das medidas $\mathbf{R}$ e $\hat{\mathbf{P}}_k$. Em seguida, o ganho é utilizado para atualizar o vetor de estados e a matriz de covariâncias. A equação de atualização de covariâncias (8) é empregada para garantir a estabilidade do filtro, assegurando que $\mathbf{P}_k$ permaneça positiva semi-definida (Moore and Stouch, 2014).

Pela formulação padrão do EKF, $\mathbf{H}$ deve ser uma matriz Jacobiana da função do modelo de observação h . Para suportar uma ampla gama de sensores, assume-se que cada sensor produz medições das variáveis de estado que estão sendo estimadas. Assim, $\mathbf{H}$ é simplesmente uma matriz identidade. O filtro suporta atualizações parciais do vetor de estados, o que é necessário pois geralmente os sensores não disponibilizam todas as variáveis do vetor de estados. Na prática, isso é feito através de $\mathbf{H}$. Por exemplo, ao se medir apenas m variáveis, $\mathbf{H}$ torna-se uma matriz m por 15 de posto m , com valores de 1 e 0 nas colunas de variáveis medidas e não medidas, respectivamente.

Tabela 1: Tabela com as variáveis do espaço de estados que são medidas pelos sensores e as utilizadas para fusão pelo EKF.

Sensores	Variáveis disponíveis	Variáveis utilizadas
Odometria via <i>encoders</i>	$x, y, \gamma, \dot{x}, \dot{\gamma}$	$\dot{x}_{dif}, \dot{\gamma}_{dif}$
Odometria visual	$x, y, \gamma, \dot{x}, \dot{\gamma}$	$\dot{x}, \dot{\gamma}$
Pose via laser	x, y, γ	x, y, γ

É importante ressaltar que na teoria espera-se que os sensores reportem devidamente suas covariâncias em cada medição, mas na prática isso pode não ocorrer, como no caso deste projeto, em que os valores de odometria obtidos pelos *encoders* são reportados com covariâncias nulas, indicando uma medida sem erros que é impossível na prática. Para contornar esses casos, o algoritmo assume valores de covariância baixos (10^{-6}) e diferencia os

valores absolutos de posição e orientação (x, y, γ) para obter valores relativos de velocidade linear e angular $(\dot{x}_{dif}, \dot{\gamma}_{dif})$ que, então, são fundidos no filtro, deste modo, as covariâncias ganham uma pequena quantidade de erro a cada iteração.

3 Experimentos e Resultados

Para avaliar o desempenho do sistema desenvolvido, foram realizados experimentos de localização e mapeamento simultâneos remotamente, utilizando a plataforma mostrada na Fig. 2 e um computador remoto ligados em uma rede sem fio padrão, ambos com sistema operacional Ubuntu 14.04 LTS (*Trusty Tahr*) e ROS Indigo. Computador com processador Intel Core i5 2.6 GHz, 6 GB de memória RAM e placa de vídeo GEFORCE 2 GB.

Devido às restrições por parte da odometria via *encoders* (não são medidas z, α, β e suas respectivas velocidades e aceleração), o ambiente selecionado para a tarefa foi um ambiente plano, interno e com boa luminosidade, no qual foram desconsideradas as eventuais variações sutis no solo. Em razão disso, mapeamento, odometria visual e EKF foram restringidos ao modo 2D, zerando os valores de z, α, β e suas respectivas velocidades e aceleração, entretanto, o mapa gerado ainda é uma representação 3D do ambiente em nuvem de pontos. A posição inicial do robô foi considerada a origem do sistema de coordenadas.

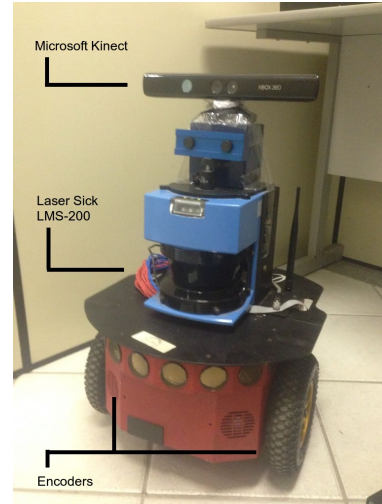


Figura 2: Foto do robô Pioneer 3-DX montado com os sensores usados no projeto.

Os dados dos sensores e a saída do filtro foram gravados em um arquivo *.bag* e então foram plotados gráficos via *Matlab* das variáveis de estado e covariâncias relevantes para análise. Os resultados podem ser observados nas Fig. (3) a (6).

Como pode ser observado na Fig. 3, os sensores reportam valores semelhantes e o EKF lineariza bem o sistema, sendo os valores estimados

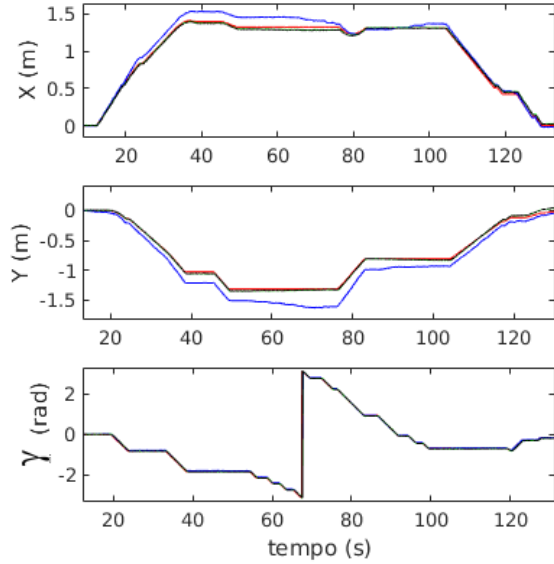


Figura 3: Gráfico da posição (x, y) e orientação (γ) absolutas reportados pelos sensores: Odometria *encoders* (vermelho), Odometria visual (azul), pose do laser (verde). E o resultado da fusão pelo EKF (preto).

bem próximos dos valores reportados pelo laser. Apesar do filtro não usar os valores absolutos reportados pelos *encoders* e odometria visual, esses valores foram plotados para efeito de comparação, assim como na Fig. 5.

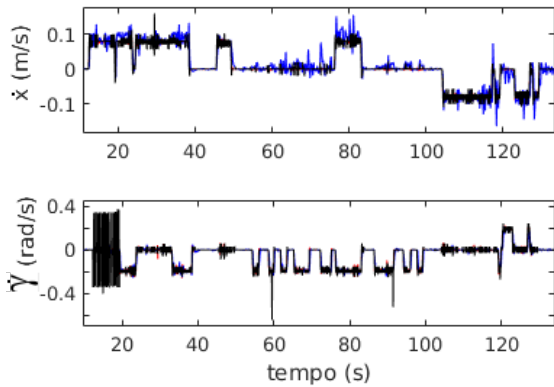


Figura 4: Gráfico das velocidade linear ($\dot{x}$) e angular ($\dot{\gamma}$) relativas reportados pelos sensores: Odometria via *encoders* (vermelho), Odometria visual (azul). E o resultado da fusão pelo EKF (preto).

Na Fig. 4, são mostrados os valores de velocidades usados pelo filtro. Apesar das oscilações, o filtro consegue convergir. Esses valores não são usados no mapeamento, onde somente a posição absoluta do robô é utilizada.

A Fig. 5 mostra a trajetória do robô vista de cima. Como pode ser observado, a trajetória obtida pelo EKF é bem semelhante a trajetória reportada pelo laser.

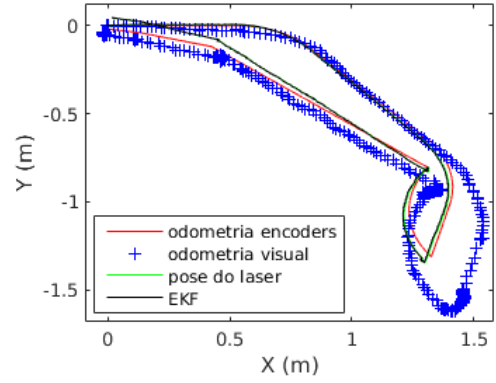


Figura 5: Gráfico da trajetória (x, y) do robô vista de cima. As legendas se encontram na figura.

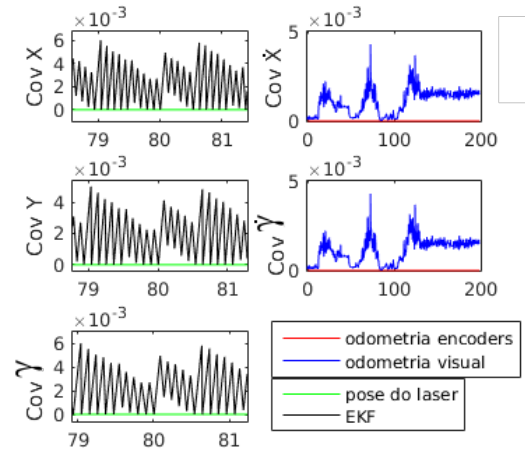


Figura 6: Gráfico das covariâncias reportadas pelos sensores e da fusão pelo EKF pelo tempo. As legendas se encontram na figura.

As covariâncias das variáveis relevantes ao sistema são mostradas na Fig. 6. Para melhor visualização, as covariâncias em x, y e γ foram ampliadas em torno de $t = 80s$. Caso o filtro não tivesse nenhuma fonte de medição absoluta (x, y ou γ), as covariâncias dessas variáveis iriam crescer sem limites, podendo levar à instabilidade. Neste projeto foram utilizadas as medições de posição absoluta obtidas pelo laser (como mostrado na Tabela 1), pelo fato deste ser o sensor com maior precisão entre os três. O comportamento das covariâncias no filtro é semelhante ao observado por Moore and Stouch (2014), que utilizaram unidades de medida inercial (IMU) e sistemas de posicionamento global (GPS) como sensores de localização em um ambiente externo.

Dois mapas completos do ambiente selecionado foram feitos, um deles utilizando o EKF para fusão de sensores e outro usando somente odometria visual. As grades de ocupação geradas através do pacote *rtabmap_ros* são exibidas na Fig. 7. Observa-se que ambos os métodos

proporcionaram um mapa bem representativo do ambiente, com uma boa detecção de fechamento de ciclo (*loop closure detection*). É possível notar um pequeno deslocamento angular em uma das salas do mapa obtido usando EKF devido à uma colisão acidental do robô com uma parede, indicando uma pequena desvantagem do sistema (*encoders* reportaram movimento quando o robô estava parado). Contudo, foi possível localizar o robô normalmente quando ele saiu desta sala. O mapa obtido exclusivamente pela odometria visual também apresenta pequenas imperfeições e nota-se que ele é um pouco maior. A planta real do ambiente não estava disponível no momento da pesquisa para comparações quantitativas.

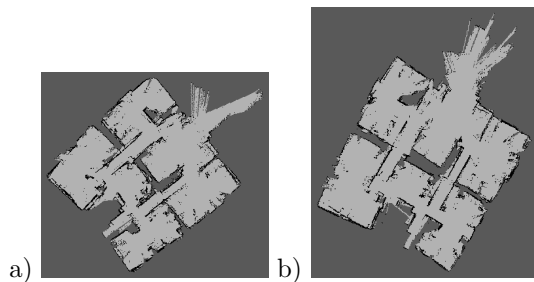


Figura 7: Grades de ocupação geradas pelo pacote *rtabmap_ros* nos casos: a) Localização EKF; b) Odometria visual.

Por fim, a Fig. 8 apresenta uma visão em perspectiva do mapa 3D em nuvem de pontos gerado pelo pacote *rtabmap_ros* com a localização feita pelo EKF.

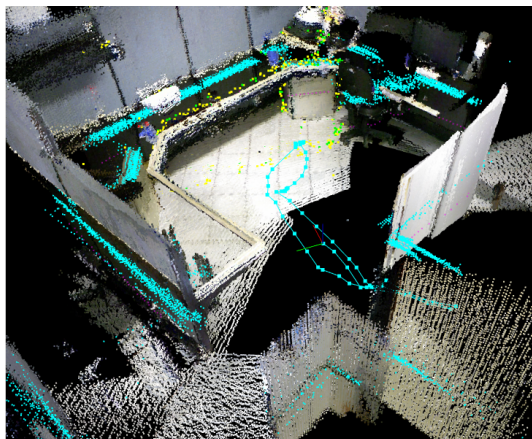


Figura 8: Visualização em perspectiva do mapa 3D de nuvem de pontos gerado pelo pacote *rtabmap_ros* (trajetória do robô em azul).

4 Conclusão e Trabalhos Futuros

Neste trabalho foi desenvolvido um sistema de localização e mapeamento 3D, usando um EKF para realizar a fusão de 3 fontes de sensoriamento distintas. Os resultados mostrados na seção anterior comprovam a eficiência do sistema, sendo a

linearização realizada pelo filtro adequada e a fusão dos sensores considerando suas covariância um sucesso. Em pequenos testes, também foi observado que quando a odometria visual "se perde", por exemplo em situações em que o robô se encontra de frente para uma parede lisa com poucas características, as covariâncias dessa medição têm um valor bem alto (10^4), fazendo com que o filtro ignore essas medições. A maior vantagem desse sistema é a maior robustez na localização, pois se ao menos um sensor estiver funcionando, o EKF ainda consegue estimar a localização do robô. Portanto, como os sensores são independentes, esse sistema funciona mesmo quando há falha de um ou dois sensores. A combinação de uma técnica probabilística de localização multissensorial com um método de mapeamento baseado em grafos se mostrou uma solução viável e robusta para o problema de SLAM.

Com relação à trabalhos futuros, algumas melhorias e aplicações desse sistema podem ser explorados:

- A localização do robô foi restringida no modo 2D e foi selecionado um ambiente plano, porém é possível a integração de outras variáveis de estado no filtro para uma localização 3D completa por meio de mais sensores, como IMU ou GPS. Assim, o sistema poderia ser usado também em UAVs e veículos subaquáticos em ambientes irregulares, por exemplo.
- O sistema desenvolvido também pode ser testado em um ambiente maior ou ambientes interligados, como um edifício inteiro.
- O mapa obtido pode ser utilizado no robô para localização, planejamento de trajetórias, detecção de obstáculos, etc. Assim, seria possível integrar este sistema em um sistema maior, possivelmente autônomo.
- Todos os sensores utilizados neste projeto apresentam restrições e erros. A obtenção de valores bem precisos do deslocamento do robô (*ground truth*), como os obtidos por Sturm et al. (2012) através de um conjunto de câmeras de alta resolução no ambiente, possibilitaria uma melhor avaliação do sistema e um melhor refinamento nas configurações do EKF (principalmente na matriz de covariâncias de ruídos do processo Q), para uma estimativa com erros menores.

Agradecimentos

Agradecemos à Universidade Federal de Santa Catarina (UFSC) e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelos recursos e conhecimentos proporcionados.

Referências

- Alahi, A., Ortiz, R. and Vandergheynst, P. (2012). Freak: Fast retina keypoint, *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*, Ieee, pp. 510–517.
- Bay, H., Tuytelaars, T. and Van Gool, L. (2006). Surf: Speeded up robust features, *European conference on computer vision*, Springer, pp. 404–417.
- Bicchi, A., Lorussi, F., Murrieri, P. and Scordio, V. (2004). On the problem of simultaneous localization, map building, and servoing of autonomous vehicles, *Advances in control of articulated and mobile robots*, Springer, pp. 223–242.
- Calonder, M., Lepetit, V., Strecha, C. and Fua, P. (2010). Brief: Binary robust independent elementary features, *European conference on computer vision*, Springer, pp. 778–792.
- Civera, J., Davison, A. J. and Montiel, J. M. (2008). Inverse depth parametrization for monocular slam, *IEEE transactions on robotics* **24**(5): 932–945.
- Davison, A. J., Reid, I. D., Molton, N. D. and Stasse, O. (2007). Monoslam: Real-time single camera slam, *IEEE transactions on pattern analysis and machine intelligence* **29**(6): 1052–1067.
- Henry, P., Krainin, M., Herbst, E., Ren, X. and Fox, D. (2010). Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments, *In the 12th International Symposium on Experimental Robotics (ISER, Cite-seer*.
- Hervier, T., Bonnabel, S. and Goulette, F. (2012). Accurate 3d maps from depth images and motion sensors via nonlinear kalman filtering, *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, IEEE, pp. 5291–5297.
- Holtkamp, M. J. and de Jong, S. (2006). Robot localisation using sift and active monocular vision.
- Huang, A. S., Bachrach, A., Henry, P., Krainin, M., Maturana, D., Fox, D. and Roy, N. (2017). Visual odometry and mapping for autonomous flight using an rgb-d camera, *Robotics Research*, Springer, pp. 235–252.
- Labbé, M. and Michaud, F. (2017). Long-term online multi-session graph-based splam with memory management, *Autonomous Robots* pp. 1–18.
- Lin, R., Wang, Z., Sun, R. and Sun, L. (2012). Vision-based mobile robot localization and mapping using the plot features, *Mechatronics and Automation (ICMA), 2012 International Conference on*, IEEE, pp. 1921–1927.
- Ling, L., Cheng, E. and Burnett, I. S. (2013). An iterated extended kalman filter for 3d mapping via kinect camera, *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, IEEE, pp. 1773–1777.
- Lowe, D. G. (1999). Object recognition from local scale-invariant features, *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, Vol. 2, Ieee, pp. 1150–1157.
- Montiel, J. M., Civera, J. and Davison, A. J. (2006). Unified inverse depth parametrization for monocular slam, *Robotics: Science and Systems*.
- Moore, T. and Stouch, D. (2014). A generalized extended kalman filter implementation for the robot operating system, *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*, Springer.
- Rublee, E., Rabaud, V., Konolige, K. and Bradski, G. (2011). Orb: An efficient alternative to sift or surf, *Computer Vision (ICCV), 2011 IEEE international conference on*, IEEE, pp. 2564–2571.
- Shi, J. and Tomasi, C. (1993). Good features to track, *Technical report*, Cornell University.
- Siegwart, R., Nourbakhsh, I. R. and Scaramuzza, D. (2011). *Introduction to autonomous mobile robots*, MIT press.
- Sturm, J., Engelhard, N., Endres, F., Burgard, W. and Cremers, D. (2012). A benchmark for the evaluation of rgb-d slam systems, *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, IEEE, pp. 573–580.