

REDUÇÃO DE DIMENSIONALIDADE DE DADOS PARA DETECÇÃO DE FECHAMENTO DE *LOOP* ATRAVÉS DE MÉTODO ESPECTRAL: IMPLEMENTAÇÃO EXPERIMENTAL

LEANDRO A. S. MOREIRA*, CLAUDIA M. JUSTEL*, PAULO F. F. ROSA*

**Instituto Militar de Engenharia
Praça Gen. Tibúrcio, 80
Rio de Janeiro, RJ, Brasil*

Emails: leandromoreira75@gmail.com, cjustel@ime.eb.br, rpaulo@ime.eb.br

Abstract— This paper presents results of loop closure detection in mobile robotics through spectral description of images sets and data dimensionality reduction. Both spectral description as the representation in low dimension depend heavily on the concept of dominant eigenvector. Integration between Matlab and ROS interface was exploited to perform experiments, which were made both in computationally simulated environment as in real environment. Experimental results have shown that the method is capable of performing correct loop closure detection at a significantly lower computational cost compared to the use of *OpenCV* library for visual analysis.

Keywords— SLAM, loop closure, dimensionality reduction, Matlab/ROS integration, OpenCV

Resumo— Este artigo apresenta resultados de detecção de fechamento de *loop* em robótica móvel por meio de descrição espectral de conjuntos de imagens e redução de dimensionalidade de dados. Tanto a descrição espectral quanto a representação em baixa dimensão dependem fortemente do conceito de autovetor dominante. A integração entre Matlab e a interface ROS foi explorada para a realização dos experimentos, que foram feitos tanto em ambiente simulado computacionalmente quanto em ambiente real. Os resultados experimentais obtidos mostraram que o método é capaz de realizar detecções corretas de fechamento de *loop* a um custo computacional significativamente menor se comparado à utilização da biblioteca *OpenCV* para análise visual.

Palavras-chave— SLAM, fechamento de *loop*, redução de dimensionalidade, integração Matlab/ROS, OpenCV

1 Introdução

Nas técnicas de localização e mapeamento simultâneos (SLAM), sensores embarcados no robô extraem marcos de um ambiente não-estruturado, e algoritmos de associação de dados constroem modelos probabilísticos para a estimativa da pose do robô e também da localização dos marcos. Entretanto, incertezas em relação tanto à pose do robô quanto à localização dos marcos são inerentes às técnicas de SLAM, e tais incertezas aumentam até que o resultado da associação de dados assegure que o robô está revisitando uma área previamente mapeada [Thrun et al. (2005)]. Quando isso acontece, diz-se que foi resolvido um problema conhecido como fechamento de *loop* (ver Figura 1). Em anos recentes, muitas técnicas de localização e mapeamento simultâneos baseados em visão computacional têm sido desenvolvidas para tornar mais robusta a associação de dados mencionada anteriormente. Tais técnicas, chamadas genericamente de SLAM visual, utilizam predominantemente análises de imagens coletadas pelo robô durante o processo de navegação para o reconhecimento de marcos. A análise visual pode contribuir para a resolução do problema de fechamento de *loop*, na medida em que é possível associar conjuntos de marcos de imagens entre si, determinando o quanto tais imagens são similares. Isso pode ser alcançado, por exemplo, por meio de técnicas de extração de pontos característicos

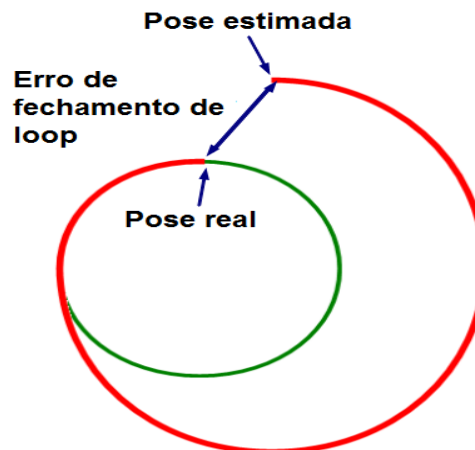


Figura 1: Representação esquemática do problema de fechamento de *loop*.

das imagens e formação de descritores locais, tais como SIFT e SURF.

Este artigo emprega o método de detecção de fechamento de *loop* baseado em descrição espectral e redução de dimensionalidade de dados desenvolvido por Moreira et al. (2015a) e Moreira et al. (2015b) e apresenta resultados de experimentos simulados computacionalmente assim como de experimentos utilizando-se um robô real. O restante desse artigo está organizado do seguinte modo: a Seção 2 descreve alguns trabalhos relacionados com a detecção de fechamento

de *loop*; a Seção 3 recapitula as ideias principais descritas pelos dois artigos mencionados anteriormente; a Seção 4 descreve os recursos computacionais utilizados para a realização dos experimentos e mostra os resultados obtidos; e a Seção 5 apresenta as conclusões.

2 Trabalhos relacionados

Para a detecção de fechamento de *loop*, Newman et al. (2006) apresentaram uma técnica de análise de similaridade que utiliza descritores SIFT [Lowe (2004)] e aproximação matricial de menor posto para filtrar informações visuais comuns. Cummins and Newman (2008) construíram vocabulários visuais a partir de descritores SURF [Bay and Gool (2006)] por meio do FAB-Map, um método probabilístico baseado em aparência. Hartmann et al. (2013) fizeram uma análise comparativa entre SIFT e SURF, entre outros descritores, para SLAM visual. No presente trabalho, um único vetor de informações associado a esses descritores locais é usado, sem a construção de qualquer vocabulário visual.

Câmeras *stereo* foram empregadas por Cadena et al. (2010) em um processo que considera tanto cenas próximas e distantes para melhorar o processo de reconhecimento de locais. Por outro lado, o processo descrito neste artigo pode utilizar câmeras tanto *stereo* quanto monoculares.

Fechamentos de *loop* bidirecionais em imagens panorâmicas foram identificados por Arroyo et al. (2014) por meio de correlação entre subpanoramas e a partir de descritores binários globais. Descritores binários também foram utilizados por Garcia-Fidalgo and Ortiz (2014) para computar um vocabulário visual de maneira *online* e evitar a fase de treinamento. Este trabalho também não depende de nenhuma etapa de treinamento, mas apenas fechamentos de *loop* unidirecionais são detectados.

O conceito de autovetor dominante como descritor visual acoplado a redução de dimensionalidade de dados por Mapeamentos por Difusão [Coifman and Lafon (2006)] foi empregado por Moreira et al. (2015a) e Moreira et al. (2015b) para identificar fechamentos de *loop* em conjuntos de imagens *indoor* e *outdoor*, através de uma abordagem espectral. Este artigo aplica as ideias desenvolvidas nesses dois últimos trabalhos à realização de experimentos simulados e reais utilizando-se a integração computacional entre Matlab e a interface ROS (*Robot Operating System*), e compara os tempos de processamento obtidos por este método com os tempos obtidos através da utilização da biblioteca *OpenCV*.

3 Fechamento de Loop por Método Espectral

Esta seção descreve sucintamente a abordagem espectral adotada por Moreira et al. (2015a) e Moreira et al. (2015b) para detecção de fechamento de *loop*. Basicamente, a metodologia empregada por aqueles autores divide-se em três partes:

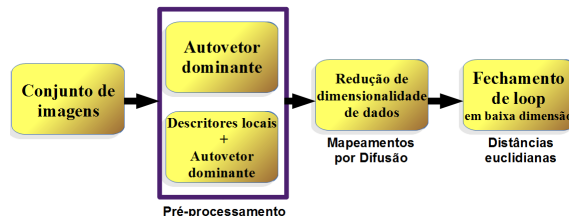


Figura 2: Estrutura geral do método de detecção de fechamento de *loop* proposto por Moreira et al. (2015a) e Moreira et al. (2015b).

3.1 Pré-processamento

A fase de pré-processamento consiste em determinar o autovetor dominante v_1 da decomposição espectral de uma matriz simétrica obtida por (i) $M_{n \times n} = I^T I$, onde $I_{m \times n}$ é a matriz de pixels correspondente a uma imagem; ou (ii) $M_{r \times r} = \mathbb{D}^T \mathbb{D}$, onde $\mathbb{D}_{k \times r}$ é a matriz de k descritores locais de r componentes obtida de uma imagem I . Neste último caso, foram analisados resultados provenientes dos descritores SIFT [Lowe (2004)] e SURF [Bay and Gool (2006)], com $r = 128$ e $r = 64$ respectivamente. O autovetor dominante v_1^i será empregado como único vetor representativo de uma dada imagem I_i .

3.2 Redução de dimensionalidade de dados

Coifman and Lafon (2006) desenvolveram o método espectral Mapeamentos por Difusão (*Diffusion Maps*), o qual permite realizar redução de dimensionalidade de dados por meio da construção de um sistema de coordenadas em baixa dimensão formado por autovalores e autovetores de uma matriz de transição de probabilidades. Nas aplicações de Moreira et al. (2015a) e Moreira et al. (2015b), o que se faz é reduzir a dimensão dos autovetores dominantes v_1^i obtidos na fase de pré-processamento. Assim, os elementos v_1^i de um conjunto finito de imagens X são relacionados entre si por meio do *kernel* gaussiano $k(v_1^i, v_1^j) = e^{-\|v_1^i - v_1^j\|^2 / \varepsilon}$. Essas avaliações são armazenadas em uma matriz K , que pode ser normalizada dando origem a uma matriz de transição $P = D^{-1}K$, onde D é a matriz diagonal tal que $D_{ii} = \sum_j k_{ij}$. O método de Mapeamentos por Difusão define por meio dos autovalores e autovetores de P um sistema de coordenadas no espaço

euclidiano de baixa dimensão $\mathbb{R}^s$ no qual os autovetores dominantes v_1^i são mapeados. Assim, é construída uma aplicação $\Psi_t : X \rightarrow \mathbb{R}^s$ que produz, para um determinado instante t , uma imersão de X em $\mathbb{R}^s$ com a norma euclidiana usual. Então, dado um elemento $v_1^i \in X$, sua representação no espaço $\mathbb{R}^s$ será dada por:

$$\Psi_t(x_i) = [\lambda_1^t \psi_1(v_1^i); \lambda_2^t \psi_2(v_1^i); \dots; \lambda_s^t \psi_s(v_1^i)]$$

onde $\psi_q(v_1^i)$ significa a i -ésima componente do q -ésimo autovetor de P .

3.3 Fechamento de loop em baixa dimensão

Após o processo de redução de dimensionalidade de dados, cada autovetor dominante v_1 é representado por um ponto em um espaço de baixa dimensão. Calculando-se a distância euclidiana ponto a ponto, são procurados indícios de fechamento de *loop* de acordo com o seguinte raciocínio: se duas imagens são semelhantes, então seus autovetores dominantes também são semelhantes, o que faz com que, em dimensão reduzida, sejam representados por pontos cujas coordenadas são próximas. A estrutura geral do método proposto é ilustrada pela Figura 2. Um exemplo de detecção de múltiplos fechamentos de *loop* usando-se o pré-processamento (i) da Seção 3.1 é apresentado na Seção 3.4.

3.4 Exemplo de aplicação do método

Considere a base de dados *New College* [Cummins and Newman (2008)], amplamente utilizada como *benchmark* por pesquisadores das áreas de reconhecimento e mapeamento visuais. Consiste de um conjunto de 992 pares de imagens (esquerda e direita) de 640×480 pixels capturadas por uma câmera montada sobre um suporte *pan-tilt*, instalado em um robô teleoperado que percorreu uma trajetória pré-definida no New College, na Universidade de Oxford. A Figura 3 mostra o mapa do local com o desenho do percurso realizado, onde o mapa de cores – azul para o início do percurso, vermelho para o fim – indica a sequência de captura das imagens. Este conjunto de dados sofreu o pré-processamento (i) da Seção 3.1, onde cada matriz de pixels I é o resultado da concatenação das imagens à esquerda e à direita de cada par.



Figura 3: Mapa referente à base de dados *New College* e o percurso realizado por um robô teleoperado que capturou as imagens.

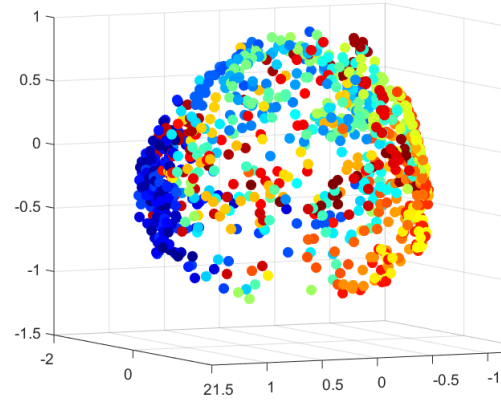


Figura 4: Representação em baixa dimensão dos autovetores dominantes das matrizes $M = I^T I$ correspondentes aos 992 pares de imagens do conjunto de dados New College.

Como cada imagem tem 640×480 pixels, a matriz correspondente a um par concatenado será representada por $I_{480 \times 1280}$, e o autovetor dominante v_1 de $M = I^T I$ é calculado. Assim, cada v_1^i usado como entrada para o método de Mapeamentos por Difusão tem dimensão 1280. A representação em baixa dimensão (dimensão 3) obtida após a aplicação do método de Mapeamentos por Difusão ao conjunto de dados New College é mostrada na Figura 4.

Calculando-se a distância euclidiana entre todos os pares de pontos em baixa dimensão, obtém-se uma representação matricial que revela múltiplos fechamentos de *loop* detectados no percurso considerado. Tais fechamentos são revelados pelas sequências de pontos paralelas à diagonal principal da matriz. Nesta representação matricial, preferiu-se mostrar, para cada ponto p , apenas os 20 pontos mais próximos a p . A Figura 5 mostra a correspondência entre os fechamentos de *loop* detectados e os fechamentos de *loop* presentes de fato no percurso considerado. Os tempos de execução para o processamento deste conjunto de dados são apresentados na Tabela 1, e foram obtidos usando-se um computador com processador Intel Core i7 de 2.4 GHz, memória RAM de 8GB e sistema operacional Windows 8 de 64 bits.

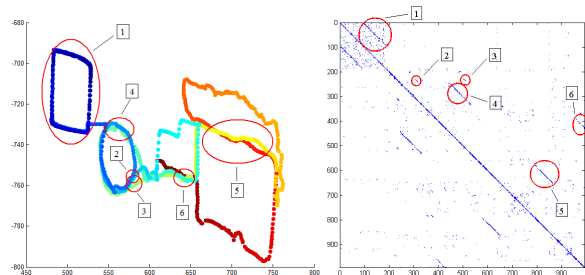


Figura 5: Fechamentos de *loop* corretamente detectados em relação ao percurso considerado.

Uso de aut. dom.	Tempos
Pré-processamento	91,6871 s
Médio por imagem	0,0924 s
Kernel Map. Dif.	0,0987 s
Coordenadas dim 3	0,2475 s

Tabela 1: Tempos de execução para o conjunto de dados New College, pré-processado usando-se apenas autovetores dominantes.

4 Organização experimental e resultados

Esta seção faz uma breve descrição dos recursos computacionais utilizados para a realização dos experimentos e apresenta resultados de detecção de fechamento de *loop* a partir de imagens capturadas tanto em ambiente simulado computacionalmente quanto em ambiente real *indoor*. ROS (*Robot Operating System*) é uma estrutura de código aberto e multiplataforma de desenvolvimento para robótica, que permite a abstração do *hardware* e consiste de ferramentas, bibliotecas e pacotes que simplificam e padronizam a criação de rotinas para robôs. O funcionamento do ROS baseia-se na construção de uma rede de nós especializados em controlar tarefas específicas, que trocam mensagens entre si de acordo com um modelo *publicador-assinante*. Corke (2015) mostra como estabelecer a comunicação entre uma rede ROS inicializada a partir da biblioteca *Robotics* do Matlab 2015.a – rodando em Windows – e outra rede ROS inicializada em uma máquina virtual (MV) Linux. Esta integração é particularmente interessante para o presente trabalho, dado que todos os passos descritos na Seção 3 se encontravam implementados em Matlab, enquanto os nós de controle do robô e de captura de imagens da câmera nele embarcada estavam em Linux. Ainda assim, foi necessário testar diferentes MVs, com diferentes distribuições ROS instaladas, de modo a se obter estabilidade e eficiência em tempo de resposta compatível com o *hardware* já descrito na Seção 3. Para os propósitos deste trabalho, os melhores resultados foram obtidos com uma MV Linux Ubuntu 64 bits, com a distribuição ROS Hydro, rodando via VMWare *player*. Outro recurso computacional utilizado foi o simulador Gazebo, uma ferramenta gratuita que permite facilmente criar ambientes complexos *indoor* e *outdoor* e emular de forma robusta o comportamento de vários tipos de robôs reais. Por fim, e apenas para efeito de comparação entre tempos de execução, utilizou-se o *OpenCV*, uma biblioteca gratuita, de código aberto, multiplataforma, com interfaces para C++, C, Python e Java, e que contém centenas de algoritmos de visão computacional. É possível utilizar *OpenCV* para extrair pontos característicos de uma imagem e formar um conjunto de descritores locais, tais como os descritores SIFT e SURF, que são



Figura 6: Amostra das 731 imagens capturadas em um ambiente criado no simulador Gazebo.

usados para comparar imagens por meio de processos de associação implementados em algoritmos de pesquisa por descritores vizinhos mais próximos. Em geral, esses algoritmos são otimizados por estruturas de busca do tipo *KD-tree*.

4.1 Conjunto de dados 1: imagens de um ambiente simulado em Gazebo

O primeiro conjunto de dados consiste de uma sequência de 731 imagens de 640×480 pixels capturadas de um ambiente criado no simulador Gazebo. O experimento foi realizado empregando-se um Turtlebot com uma câmera RGB-D embarcada. Neste primeiro experimento, o robô realizou uma trajetória retangular e as imagens foram capturadas ao longo de uma volta e meia de percurso, o que gerou uma sequência de imagens correspondente a um fechamento de *loop*. Algumas das imagens deste conjunto são mostradas na Figura 6. As 731 imagens deste conjunto de dados foram submetidas ao pré-processamento (ii) da Seção 3.1, com a utilização de descritores SIFT e SURF. Foram analisados resultados de detecção de fechamento de *loop* após a redução de dimensionalidade de dados (Seção 3.2) e cálculo de distâncias euclidianas em baixa dimensão (Seção 3.3). A Figura 7 mostra a representação em baixa dimensão para o conjunto de autovetores dominantes relacionados às matrizes de descritores SIFT [Figura 7(a)] e SURF [Figura 7(c)], respectivamente, assim como as correspondentes representações matriciais das distâncias euclidianas ponto a ponto em baixa dimensão [Figura 7(b) e Figura 7(d)]. Nas representações matriciais, é possível observar as sequências de fechamentos de *loop* detectadas e identificadas pelas diagonais secundárias.

4.2 Conjunto de dados 2: imagens de um ambiente real

O segundo conjunto de dados consiste de uma sequência de 565 imagens de 640×480 pixels que

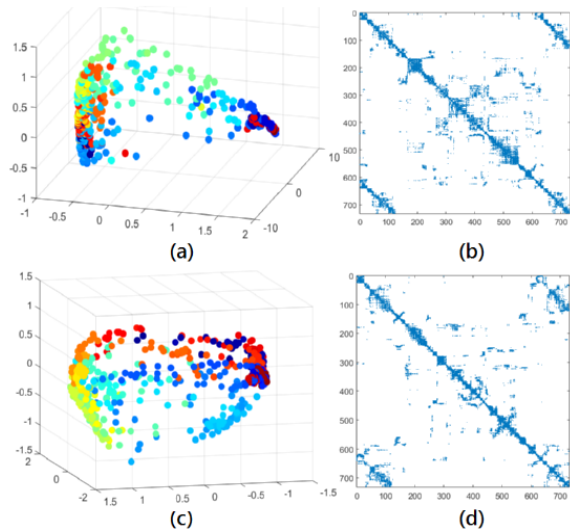


Figura 7: Mapeamentos em baixa dimensão [(a) e (c)] e matrizes de distâncias euclidianas [(b) e (d)] correspondentes ao conjunto de dados 1.

foram obtidas de um ambiente *indoor* por meio do robô Pioneer 3DX com uma câmera RGB-D embarcada [Figura 8]. Neste segundo experimento, o robô também realizou uma trajetória retangular. As imagens foram capturadas durante uma volta e meia do robô nessa trajetória, o que gerou uma sequência de imagens de um fechamento de *loop*. As imagens deste conjunto de dados também foram submetidas ao pré-processamento (ii) da Seção 3.1, com a utilização tanto de descritores SIFT quanto de descritores SURF. Novamente, o fechamento de *loop* foi detectado após a redução de dimensionalidade de dados (Seção 3.2) e cálculo de distâncias euclidianas em baixa dimensão (Seção 3.3). A Figura 8 mostra o robô utilizado no experimento e algumas das imagens capturadas. A Figura 9 mostra a representação em baixa dimensão para o conjunto de autovetores dominantes relacionados às matrizes de descritores SIFT [Figura 9(a)] e SURF [Figura 9(c)], respectivamente, assim como as correspondentes representações matriciais das distâncias euclidianas ponto a ponto em baixa dimensão [Figura 9(b) e Figura 9(d)].



Figura 8: Robô Pioneer 3DX com câmera RGB-D embarcada e amostra das 565 imagens capturadas em um ambiente *indoor*.

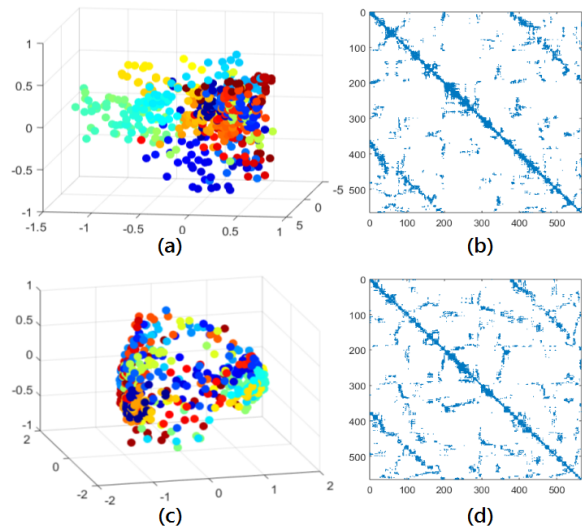


Figura 9: Mapeamentos em baixa dimensão [(a) e (c)] e matrizes de distâncias euclidianas [(b) e (d)] correspondentes ao conjunto de dados 2.

4.3 Análise dos resultados

O processo de associação de descritores locais para comparação de imagens realizado via *OpenCV* pode ser utilizado para detecção de fechamentos de *loop*, e isso é o que se faz em geral. Mas tal processo não tem um custo trivial, mesmo quando se utilizam estruturas otimizadas de busca, principalmente se é grande o número de descritores por imagem. Por outro lado, deve-se lembrar que a detecção de fechamento de *loop* por redução de

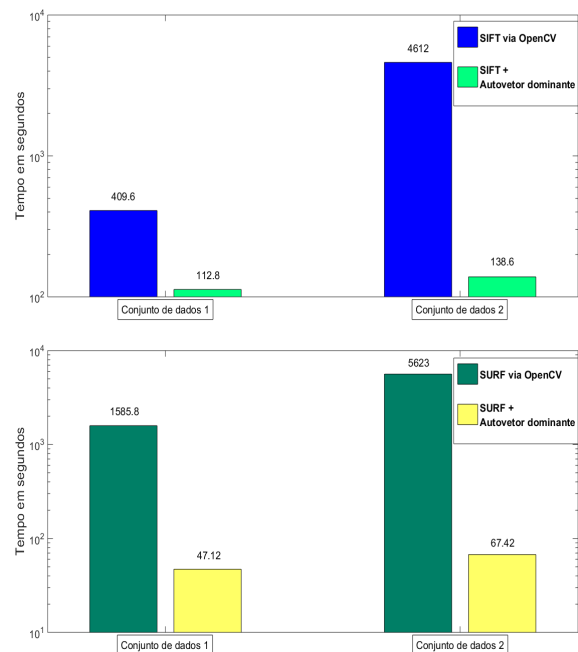


Figura 10: Tempos totais de processamento (em segundos) efetuados pelo *OpenCV* e pelo método espectral utilizado neste trabalho. Os eixos verticais estão representados em escala logarítmica.

dimensionalidade de dados baseia-se em um *kernel* de similaridade que n o depende do processo de associa  o entre m ltiplos descritores em cada imagem, por isso o custo computacional   significativamente menor. A Figura 10 mostra, para os mesmos conjuntos de imagens, a compara  o entre os tempos totais de processamento entre o *OpenCV* e o m todo espectral utilizado neste trabalho. O primeiro gr fico ilustra os tempos relacionados   utiliza  o dos descritores SIFT, enquanto os tempos relacionados   utiliza  o dos descritores SURF encontram-se representados no segundo gr fico.

5 Conclus es

Este artigo apresentou resultados experimentais de detec  o de fechamento de *loop* a partir de descri  o espectral de conjuntos de imagens e constru  o de representa  es em baixa dimens o. Os experimentos foram realizados em ambiente simulado computacionalmente, utilizando-se Gazebo, e tamb m em ambiente real *indoor* com um rob  Pi-oneer 3DX. Em ambos os casos, utilizou-se a integra  o entre as ferramentas computacionais Matlab e ROS – este rodando em m quina virtual.

Os tempos de execu  o mostram que a estrutura espectral utilizada – onde s o empregados autovetores dominantes associados a matrizes de descritores locais –   significativamente mais eficiente em compara  o com a associa  o de descritores realizada pelo *OpenCV*.

Um trabalho futuro interessante ser  adaptar a estrutura espectral do m todo utilizado para processamento *online*, de modo que a representa  o em coordenadas em baixa dimens o seja atualizada em tempo real, com baixo custo computacional,   medida que o rob  coletar informa  o do ambiente. Outro ponto a ser tratado   analisar a robustez do m todo em rela  o   detec  o de falsos positivos e falsos negativos. Para isso, as medidas de precis o e revoca  o dever o ser empregadas nessa an lise.

Agradecimentos

Os autores agradecem o suporte financeiro da Coordena  o de Aperfei oamento de Pessoal de N vel Superior (CAPES) – Edital 21-2011 Pr -Estrat gia, CAPES-DS, Financiadora de Estudos e Projetos (FINEP) Chamada P blica CT-AERO Vant 2009 e CNPq (edital Universal - 484767/2011-9).

Refer ncias

Arroyo, R., Alcantarilla, P. F., Bergasa, L. M., Torres, J. J. Y. and Gamez, S. (2014). Bidirectional loop closure detection on panoramas for visual navigation, *2014 IEEE*

Intelligent Vehicles Symposium Proceedings, pp. 1378–1383.

Bay, H. and Gool, L. V. (2006). Surf: Speeded up robust features, *Proceedings of 9th European Conference on Computer Vision*, pp. 404 – 417.

Cadena, C., G lvez-L pez, D., Ramos, F. and Tard s, J. D. (2010). Robust place recognition with stereo cameras, *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 5182 – 5189.

Coifman, R. R. and Lafon, S. (2006). Diffusion maps, *Applied and Computational Harmonic Analysis* **21**(1): 5 – 30.

Corke, P. (2015). Integrating ros and matlab, *IEEE Robotics and Automation Magazine* **22**(2): 18 – 20.

Cummins, M. and Newman, P. (2008). Fab-map: Probabilistic localization and mapping in the space of appearance, *The International Journal of Robotics Research* **27**(6): 647–665.

Garcia-Fidalgo, E. and Ortiz, A. (2014). On the use of binary feature descriptors for loop closure detection, *2014 IEEE Emerging Technology and Factory Automation*, pp. 1–8.

Hartmann, J., Klussendorff, J. and Maehle, E. (2013). A comparison of feature descriptors for visual slam, *IEEE European Conference on Mobile Robots (ECMR)*, pp. 56–61.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision* **60**(2): 91–110.

Moreira, L. A. S., Justel, C. M. and Rosa, P. F. F. (2015a). Detec  o de fechamento de loop via mapeamentos por difus o - uma aplica  o para rob s m veis, *Anais on-line do SBAI/DINCON 2015*, SBAI '15, pp. 1071–1076.

Moreira, L. A. S., Justel, C. M. and Rosa, P. F. F. (2015b). Visual similarity analysis in loop closure through data dimensionality reduction via diffusion maps, *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, SAC '15, pp. 289–294.

Newman, P. M., Cole, D. M. and Ho, K. L. (2006). Outdoor slam using visual appearance and laser ranging, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1180–1187.

Thrun, S., Burgard, W. and Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, The MIT Press.