Auto-Localização Autônoma de Robôs Móveis por Visão Computacional Baseada em Pontos de Referência

Leandro Couto[1], Fernando Osório[1]

[1]Instituto de Ciências Matemáticas e de Computação - ICMC
Universidade de São Paulo - USP
Cx. Postal 668 - CEP: 13560-970 - São Carlos – SP – Brazil

landovers@gmail.com,fosorio@icmc.usp.br

Abstract. *This paper demonstrates a method of global localization for autonomous mobile robots based on the creation of a visual memory map, through detection and description of reference points of captured images associated to odometer data in a specific environment. This procedure allows localization to be achieved at a later stage through association of these memorized features with the scene being observed in real time. Experiments are conducted to show the effectiveness of the method for the localization of patrolling robots in indoor environments. The results are analyzed and navigation alternatives and possible future refinements are discussed.*

Resumo. *Este artigo demonstra um método de localização global para robôs móveis autônomos baseado na criação de um mapa de memória visual, através da detecção e descrição de pontos de referência de imagens capturadas de um ambiente específico, associados a informação odométrica. Este procedimento permite que a localização seja alcançada posteriormente através da associação destes pontos característicos memorizados com a cena observada em tempo real. Experimentos são realizados para mostrar a eficácia do método para localização de robôs patrulheiros em ambientes internos. Os resultados são analisados, bem como são discutidas alternativas para navegação e possibilidades de futuros aprimoramentos.*

1. Introdução

A pesquisa e o desenvolvimento de robôs de serviço têm crescido muito nos últimos anos, visando criar aplicações baseadas no uso de robôs móveis autônomos. Este trabalho tem como objetivo maior a pesquisa e desenvolvimento de robôs móveis para tarefas de monitoramento e segurança em ambientes internos (*indoor*). Os robôs móveis autônomos possuem usualmente módulos responsáveis por determinar sua localização (auto-localização) e realizar sua navegação (planejamento e execução de deslocamento buscando alcançar um determinado objetivo).

Um dos aspectos mais importantes que devem ser abordados na robótica móvel (Thrun et al. 2006) é o da localização global, frequentemente denotado através do problema do robô seqüestrado (Howie et al, 2005), no qual o robô é colocado em operação em um local arbitrário do ambiente, em uma pose arbitrária. Análises sobre esse tipo de caso são importantes, pois além de esta não ser uma situação incomum em aplicações de robótica móvel, suas soluções relacionam-se com a capacidade do sistema de se recuperar de possíveis falhas na manutenção da localização do robô. Assim, este é um problema amplamente estudado na robótica, para o qual existem diversas

abordagens, com variados tipos de sensores. A câmera de vídeo, sensor usado no presente trabalho, é uma escolha interessante por ser um sensor ubíquo, de fácil acesso e baixo custo em relação a outros sensores, e a informação visual trazida é rica. Além disso, a visão é o principal sentido usado pelo Homem para a localização, o que promove uma analogia imediata com a localização humana. De fato, com o avanço da tecnologia e do poder de processamento, técnicas de Visão Computacional e pesquisas na área vêm crescendo em popularidade e alcançando resultados muito significativos.

O presente trabalho demonstra uma implementação de um sistema de localização global utilizando como sensor apenas uma câmera. O trabalho mostra como um método de extração de assinaturas de pontos de interesse de imagens associado à informação odométrica pode ser usado para construir no robô uma memória do ambiente. Esta poderá servir de referência para que o robô recupere sua posição em uma situação futura, através da associação entre a assinatura observada e as cenas armazenadas em sua memória.

A proposta consiste da capacitação do robô para se localizar dentro de um percurso. O percurso é descrito de forma semelhante à VSRR (*View Sequenced Route Representation*) (Matsumoto & Inoue, 1996), representação visual de uma rota proposta por Matsumoto. No VSRR, um percurso de gravação é realizado pelo robô enquanto este é guiado por um ser humano. Durante este percurso guiado o robô realiza a captura de imagens do ambiente, onde as imagens capturadas estão associadas a informação odométrica (posição e orientação do robô). Posteriormente, quando o robô é ativado, sem saber sua localização, ele é capaz de associar a imagem atual observada com as cenas armazenadas durante o percurso de gravação, e assim é possível aproximar a pose atual do robô relativamente à pose medida na gravação. Note que a medida odométrica do percurso inicial carrega em si um erro odométrico cumulativo, mas isso não influencia a localização, pois esta é apenas relativa ao percurso original. O importante neste caso é que a informação visual esteja correta, e desta forma ela pode ser usada durante a localização e navegação pra mitigar o erro odométrico, permitindo ao robô retraçar o percurso memorizado. Isso simplifica significativamente o percurso de gravação, já que não há necessidade de corrigir possíveis erros de odometria durante a gravação.

A localização propriamente se dá pelo *matching* da assinatura (o conjunto de pontos de referência obtidos para uma determinada imagem) da cena observada pelo robô com as imagens em sua memória visual. Procura-se parear cada um dos pontos de referência da assinatura entre as imagens, sendo a imagem com maior número de *matches* (pares de pontos associados entre as imagens) considerada a melhor candidata para a posição do robô. Um *match* correto é definido como a correta associação entre os mesmos pontos de uma mesma cena em imagens diferentes. O método detector-descritor de pontos de interesse busca ser robusto a variações de iluminação, rotação, escala, ruído e perspectiva, mas falsos positivos (onde *features* são incorretamente pareadas) e falsos negativos (onde a mesma *feature* em duas imagens não é pareada) podem ocorrer, sendo a taxa de ambos dependente principalmente da seletividade do detector-descritor, ou seja, do grau de exigência na escolha dos pontos relevantes. Ainda assim, os experimentos mostraram, como esperado, que as imagens com maior número de *matches* frequentemente representam corretamente o resultado mais preciso (associação da cena atual com a cena mais próxima dentre as cenas da memória – localização global), e a obtenção da posição relativa entre a imagem atual e a recuperada pode ser obtida através de análise das posições relativas das assinaturas de

ambas nas respectivas imagens (localização local). No caso de empates, o número de *matches* de ambas as imagens vizinhas são avaliadas, e a imagem com os melhores vizinhos é escolhida. Esta heurística permite a eliminação de falsos positivos.

Os experimentos realizados utilizando a técnica proposta são apresentados a seguir e os resultados são então avaliados, seguindo-se uma discussão das características e particularidades desta abordagem.

2. Descrição da Abordagem Adotada

São apresentados agora o hardware e software utilizados neste trabalho, bem como a metodologia aplicada nos experimentos. Para o desenvolvimento da proposta foi usado um robô Pioneer 3-AT. O sistema implementado fez uso da biblioteca OpenCV (versão 2.3), da ferramenta Player (versão 3.0.2) e do método de extração de características OpenSURF (*build* de 27/05/2010).

2.1 A Plataforma Robótica

A plataforma robótica escolhida para os testes foi o Pioneer 3-AT. O robô móvel Pioneer 3-AT, desenvolvido pela empresa MobileRobots Inc., é versátil e robusto, é uma das plataformas mais reconhecidas e difundidas na pesquisa acadêmica na atualidade, e é suportado por diversas das principais plataformas e ferramentas de robótica.

2.2 O Detector e Descritor de Pontos de Referência

Para obtenção de pontos de referência robustos, optou-se por um detector-descritor de *features* baseado em espaço de escala. O método descritor de características escolhido foi o SURF (Bay et al, 2006), e a implementação do SURF usada foi o OpenSURF, esta mesmo tendo sido implementada com o OpenCV. A escolha do método não foi arbitrária, mas optou-se pela alternativa mais adequada às prioridades do projeto. Métodos alternativos de detecção de pontos de interesse como o FAST (Rosten & Drummond, 2006) e o Harris (Harris & Stephens, 1988), ambos baseados em detecção de cantos (conhecidos como *corner detectors*) são pouco estáveis a variações de escala, e foram descartados devido à dependência do projeto de detecção de assinaturas em diferentes escalas. O detector deve possuir boa repetibilidade, enquanto o descritor deve oferecer descrições distintas para os pontos de referência.

O método SURF (*Speeded Up Robust Features*) apresenta bom desempenho quando comparado com outros métodos de detecção e descrição de *features* que usam espaço de escala, e um aspecto onde o SURF com frequência supera outros métodos similares como o pioneiro SIFT (Lowe, 1999) é na velocidade de processamento, em termos, por exemplo, de *matches* por segundo na etapa de comparação de *feature*s (Bauer et al, 2007). Como o presente trabalho propõe avaliar quantidades relativamente grandes de imagens em tempo-real para localização do robô, este fator foi determinante na escolha do SURF. Outros trabalhos comparativos (Juan & Gwun, 2010) também demonstram que o SURF tem taxa de acerto semelhante comparativamente com o SIFT, tendo porém uma tendência maior de ser prejudicado por variações de rotação. Como o projeto concentra-se em navegação em ambientes internos espera-se que a navegação ocorra, via de regra, em terrenos planos, o que faria com que a rotação da perspectiva fosse mínima, e até mesmo negligenciada. De fato, foi realizada uma alteração no método SURF para que o passo que garante invariância a rotação fosse ignorado (uma

variante conhecida como Upright-SURF, ou U-SURF (Bay et al, 2006)), aumentando a eficiência do método para esta aplicação específica, em termos de velocidade e capacidade discriminativa, evitando alguns possíveis falsos positivos. A pesquisa na área de descritores é bastante ativa, portanto existem alternativas recentes, como o CenSurE (Agrawal et al, 2008) e algumas variantes, que alegam desempenho similar ao SURF mas ainda assim não se mostram claramente superiores, sendo o SURF uma opção mais estabelecida.

2.3 Percurso de Gravação

Havendo estabelecido o método de detecção de pontos de referência, foi executado o percurso de coleta de dados com o robô. O robô móvel foi equipado com o notebook, ao qual uma câmera foi ligada. O percurso foi conduzido por um ser humano, guiando o robô com um joystick. Durante todo o percurso, o robô realizou a captura de imagens do ambiente, na freqüência de um frame por segundo de execução. Esta freqüência resultou em uma densidade de imagens considerada suficiente para que o mapa de imagens tivesse uma definição grande o bastante para a aplicação (maior que no trabalho de Matsumoto). Simultaneamente à captura das imagens, a informação odométrica foi gravada em um registro de *log*, especificamente os parâmetros de posição (x,y) e a rotação (z), considerados em referência à coordenada onde o robô iniciou sua execução (onde todos os parâmetros são zero). Cada imagem foi capturada associada a uma informação odométrica, de forma que um mapa de imagens do ambiente foi obtido. Com essa taxa de captura de imagens e baseado na informação odométrica capturada calcula-se que a distância média entre frames capturados foi entre 4 cm e 6 cm para os diferentes percursos capturados, uma resolução considerada adequada para a aplicação.

Em seguida foi desenvolvido um software capaz de aplicar a extração de características a um conjunto de imagens, obtendo assinaturas de características específicas de cada uma delas, através do SURF. Este *software* recebeu como entrada o conjunto de imagens obtidas pelo robô nos percursos de teste. As assinaturas das imagens foram então obtidas e gravadas em uma base de referência de assinaturas. A assinatura utilizada foi de 64 bits, pois a assinatura de 128 bits não resulta em grandes melhoras na qualidade dos descritores e acrescenta ao tempo de processamento. Alguns testes *offline* foram realizados para verificar a qualidade da associação de pontos de referência entre imagens, tanto em ambientes ricos em detalhes visuais, com muitos objetos e elementos distintos, quanto em corredores vazios.

2.4 Localização Global

Os experimentos concentraram-se no teste do algoritmo de localização global. Para testar a técnica, o robô foi posto em funcionamento em um ponto próximo ao percurso sem conhecimento de sua localização. O conjunto de *frames* que constituiu de base de dados de referência do robô englobava imagens de um percurso de ida e volta em um corredor de cerca de 20 metros, assim como percursos de entrada e saída de uma sala pequena. Este processo de localização global corresponde à parcela mais custosa do programa em termos de processamento, já que a análise é feita sobre um subconjunto grande de imagens do robô. Neste passo, o robô recém-ativado procura associar a cena capturada pela câmera no presente com a cena que mais se aproxime desta em sua memória.

Na intenção de obter melhor desempenho, a busca pela imagem correspondente adequada não foi realizada aplicando-se às cegas o algoritmo a toda a base de dados. A maneira de codificação do programa consistiu da amostragem, em intervalos fixos, de diversos quadros da base de imagens completa. Os melhores quadros são aqueles com o maior número de *matches* em relação à imagem original, ou seja, o maior número de características associadas positivamente. Assim, os melhores quadros candidatos eram selecionados e outros quadros na vizinhança destes eram também analisados. Esta heurística foi escolhida após observação do padrão dos *matches* de imagens semelhantes (ou seja, obtidos a partir de poses próximas); com freqüência foi observado que a imagem mais adequada apresentava maior número de *matches*, mas pelas características intrínsecas de invariância a perspectiva e escala do método SURF, o número de *matches* nas imagens próximas à melhor candidata variava de forma suave e portanto também era alto. Isto permitiu que a busca fosse bastante otimizada, devendo-se apenas evitar que os intervalos de amostragem fossem muito grandes, a ponto de fazer o sistema tender a um máximo local. Assim é possível adaptar o tamanho do intervalo da base de imagens, na busca de um equilíbrio entre melhor desempenho e meticulosidade da busca, de acordo com as necessidades da aplicação. A suavidade na transição entre os números de *matches* de imagens próximas era esperada devido à similaridade observada entre imagens de apenas alguns metros de distância, especialmente em casos onde há pouca rotação do robô e a variável da assinatura que mais se altera é a escala. Esta observação é de interesse pois também permite a eliminação de falsos positivos se houver uma densidade de imagens suficiente (falsos positivos costumam ter muito menos *matches* observados em seus vizinhos), e isto é possível graças ao fato de as assinaturas detectadas pelo método SURF serem relativamente resistentes a variações de escala, mas não completamente imunes a elas.

3. Resultados

A acurácia da localização global excedeu as expectativas para quando o robô era colocado em uma posição suficientemente próxima ao percurso original, sendo que raramente a localização resultou em um falso positivo, mesmo em ambientes com cenas difíceis, com poucas características de destaque que pudessem oferecer bons pontos de referência. O percurso testado foi bastante variado, com percursos através de corredores com poucas características e também um percurso dentro de uma sala grande com muitos objetos (Figura 1), e testes foram feitos com o robô sendo iniciado em diferentes posições do percurso. Vale notar que em cenas absolutamente desprovidas de pontos de interesse perceptíveis, como quando a câmera estava muito próxima de paredes, a comparação das assinaturas retornava valores muito pequenos de *matches*. Neste caso, o robô provavelmente teria que buscar uma perspectiva mais adequada durante a navegação. A quantidade de *matches* nas cenas mais adequadas era significativamente maior que nas cenas incorretas (Figura 2).

No computador usado para o processamento, Intel Core 2 Duo de 2.20GHz com 2GB de memória RAM, para a associação entre as assinaturas da cena observada e cada cena da memória (independente do número de matches), incluindo o carregamento das imagens da memória e aplicação do algoritmo, o tempo gasto foi de, em média, 415 ms.

Para fins experimentais, o *matching* foi também testado entre dois conjuntos de imagens. Cada conjunto era composto de 260 imagens, dentro de um percurso de 17 metros em um dos casos mais difíceis, um corredor com poucas características distintas. Procurou-se assim avaliar a precisão dos *matches* obtidos, levando em conta a precisão

da estimação da distância frontal (eixo normal ao plano da câmera). Na base de teste usada, estavam mapeados o corredor e sala em memória, com média de distância entre imagens variando (de acordo com os dados de odometria) entre 4 cm e 6 cm.

Com essas configurações, sendo 260 imagens avaliadas contra outras 260 outras imagens, os resultados ofereceram acertos de localização com erros abaixo de 0,5 metros de erro em 149 casos, com 180 resultados dentro da margem de 1 metro e 218 casos dentro da margem de 2 metros.

Figura 1. Quadros capturados em diferentes pontos do percurso utilizado nos testes, mostrando os diferentes perfis de cenas capturadas durante o percurso.

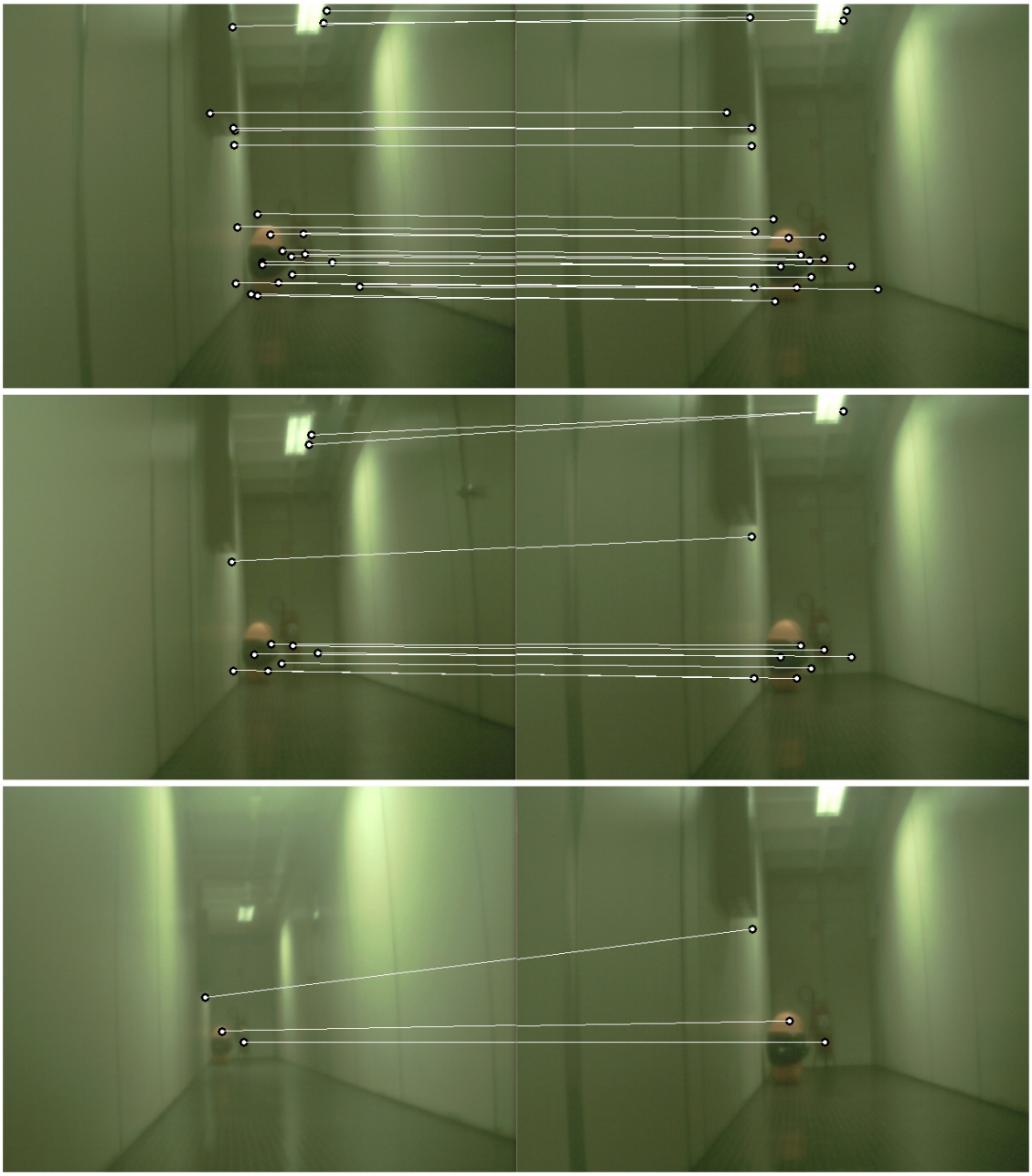

Figura 2. A figura mostra 3 resultados do algoritmo de associação usando o SURF, em um mesmo corredor. Pode-se ver a mudança gradual do número de matches em relação à mudança de ponto de vista. No primeiro par de imagens, a cena está sendo observada a partir de uma distância semelhante, resultando em um alto número de *matches* detectados. O segundo par de imagens demonstra um caso intermediário, e o terceiro par de imagens mostra como o algoritmo obtém poucas *matches* para pontos de observação díspares em termos de distância, mesmo que a cena seja essencialmente a mesma.

A aplicação do SURF para fazer a comparação das assinaturas de descritores entre uma cena específica e um conjunto completo de imagens mostra como é possível visualizar a área onde as cenas são mais semelhantes, e portanto a área que possivelmente melhor representa a atual posição do robô móvel (Figura 3).

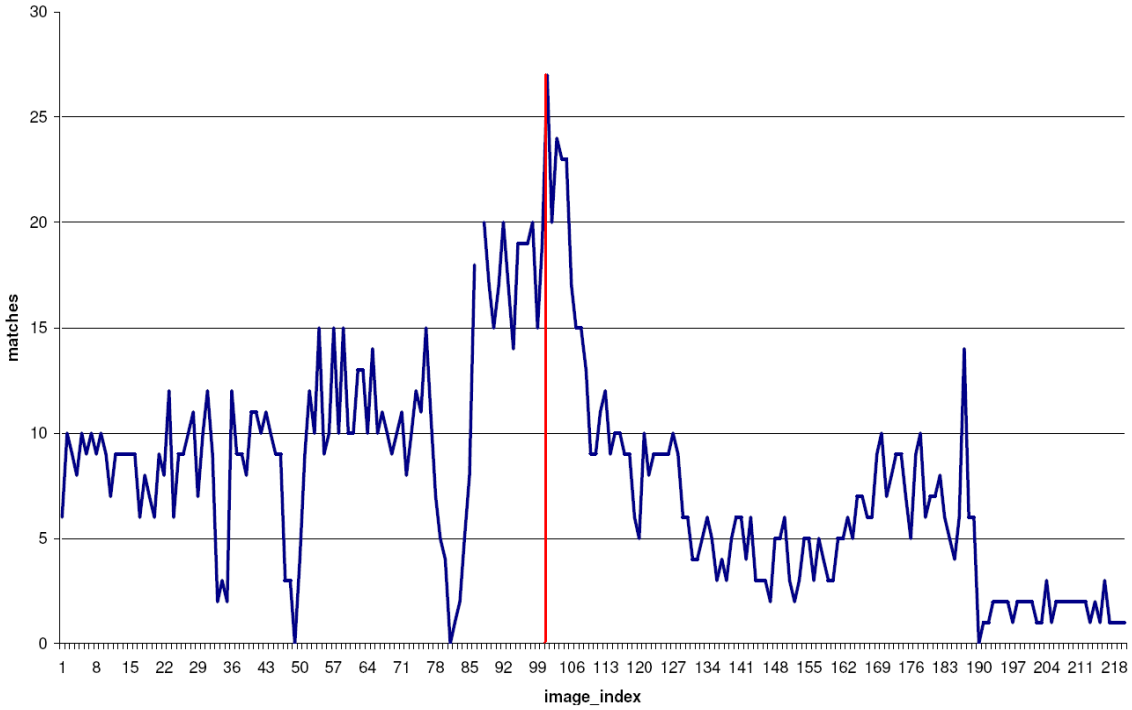

Figura 3. O gráfico mostra o número de *matches* ao aplicar o SURF entre cada imagem de um percurso representando um corredor (o mesmo da Figura 1) e uma imagem de uma cena arbitrária obtida posteriormente no mesmo percurso. A figura mostra como o número de *matches* cresce na vizinhança da cena mais adequada (marcada em *image_index* = 100).

A assinatura SURF não leva em consideração a posição relativa de uma assinatura na imagem, portanto *matches* podem ocorrer em cenas obtidas de perspectivas levemente diferentes. Este tipo de propriedade do SURF permite que o mesmo seja aplicado junto a tarefas de navegação, buscando corrigir a rota do robô (comportamento reativo local), a partir das informações dos *matches* entre imagens levemente diferentes. Uma vez obtida a melhor imagem candidata é possível realizar ajustes para corrigir diferenças de rotação ou *panning* (deslocamento horizontal de perspectiva) de forma simples, em tempo de execução durante a navegação. A medição do erro de rotação (variável *z*) é alcançada tomando-se a posição (em *pixels*) de cada ponto de referência obtido pelo SURF e comparando-a à posição de seu par na imagem correspondente do percurso de coleta, e a diferença em *pixels* da posição de ambas (*dz*) é calculada. Este procedimento é aplicado em todos os pares de pontos de referência e a diferença média em pixels (*dz_total*) entre todos os pontos da assinatura da imagem é calculada. Este valor oferece uma boa avaliação da magnitude relativa do desvio de rotação da perspectiva, e denota uma possível rotação corretiva que pode ser aplicada pelo robô durante o controle de navegação.

Possíveis soluções para a navegação serão discutidas na Seção 4. É evidente que diferenças de perspectiva, a distância dos pares de *features* relativamente à câmera e falsos positivos causam divergências nos valores de *dz* dos diversos pares, o que deve influenciar o cálculo de *dz_total*, mas os resultados observados indicam que na prática esta abordagem simples é confiável. A aplicação prática deste aprimoramento na navegação, porém, está fora do escopo deste trabalho.

4. Conclusões e Discussão

Este trabalho demonstrou um sistema de localização global, baseado no uso de uma única câmera monocular, utilizando apenas dados obtidos a partir de um detector e descritor de pontos de referência. A câmera é um sensor de baixo custo e o sistema funciona adequadamente em ambientes internos, mesmo em áreas com poucos pontos de interesse. Tendo sido desenvolvido especificamente para áreas previamente mapeadas e em percursos predeterminados, o sistema é voltado para localização em tarefas como patrulha de ambientes internos por robôs táticos. Resultados mostraram que a estimativa de localização global do robô pode ser adquirida em tempos relativamente baixos. É evidente que métodos probabilísticos como filtros Bayesianos (Thrun et al. 2006) podem ser aplicados durante a navegação para aumentar a certeza na localização do robô.

Devido à natureza da câmera monocular, seria necessária uma análise aprofundada do deslocamento dos pontos de referência durante o movimento do robô para uma estimativa numérica do desvio de perspectiva, caso em que as distâncias poderiam ser estimadas por técnicas de fluxo ótico (Illeperuma & Sonnadara, 2011; Wang & Zhao, 2011) e odometria visual (Nister et al, 2004), mas ambas estas técnicas exigem que o robô se locomova no ambiente. Em contraste, para realizar o percurso em memória, a técnica requer apenas que o robô conheça a magnitude relativa e o sinal do desvio de perspectiva (*dz_total*) para ajustar a rotação durante a navegação.

Nas mesmas condições dos testes de localização, testes preliminares já demonstraram que um controlador de rotação PD (Proporcional-Derivativo) alimentado por *dz_total* pode ser suficiente para que o robô mantenha sua localização durante a navegação. Uma avaliação detalhada da localização e navegação atuando em conjunto está sendo realizada para trabalhos futuros. Pretende-se avaliar em mais detalhes variações de parâmetros (intervalos de busca de imagens, densidade de imagens, métodos e parâmetros dos métodos descritores, parâmetros da navegação) que possam oferecer um equilíbrio entre taxa de acerto e velocidade de execução. Pretende-se também utilizar técnicas de fluxo ótico para estimar medidas de distância entre a câmera no robô e as *features*, obtendo-se informações tridimensionais sobre o ambiente. Entre os objetivos de médio e longo prazo está a implementação de SLAM (*Simultaneous Localization and Mapping*) monocular, como o MonoSLAM (Davison et al, 2007), baseado em métodos detectores e descritores de pontos de referência baseados em espaço de escala. Isto permitiria que o robô não fosse restrito a uma rota especificada no percurso de gravação, criando o mapa de memória visual simultaneamente à navegação.

Referências

Agrawal, M. Konolige, K. Blas, M, R. CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching. In *ECCV*, 2008, Part IV. Pages 102 – 115.

Bauer, J. Sunderhauf, N. Protzel, P. Comparing several implementations of two recently published feature detectors. In *Proceedings of the International Conference on Intelligent and Autonomous Systems*, IAV, 2007.

Bay, Herbert et al. SURF: Speeded Up Robust Features. In ECCV, 2006. Pages 404 – 417.

Davison, A. Reid, I. Molton, N. Stasse, O. MonoSLAM: Real-Time single camera SLAM. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007. Pages 1052 – 1067.

Harris, C. Stephens, M. A Combined Corner and Edge Detector. In *Proceedings of the Fourth Alvey Vision Conference*, 1998. Pages 147 – 151.

Howie, Choset et al, 2005, Principles of Robot Motion: theory, algorithms and implementation, The MIT Press. p.302.

Illeperuma, G. D. Sonnadara, U. J. An Autonomous Robot Navigation System Based on Optical Flow. In *Proceedings of 6th International Conference on Industrial and Information Systems*, ICIIS 2011. Pages 489-492.

Juan, L. Gwun, O. A Comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)*, Volume 3, Issue 4, 2010. Pages 143 – 152.

Lowe, D. Object Recognition from Local Scale-Invariant Features. In *International Conference on Computer Vision*, 1999. Pages 1150-1157.

Matsumoto, Y. Inaba, M. Inoue, H. Visual navigation using view-sequenced route representation. In *Proceedings of the IEEE International Conference on Robotics and Automation* (ICRA), 1996. Pages 83 – 88.

Nister, D. Naroditsky, O. Bergen, J. Visual Odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and pattern Recognition*, 2004. Pages 652 - 659.

Rosten, E. Drummond, T. Machine learning for high-speed corner detection. European Conference on Computer Vision, 2006. Pages 430 – 443.

Thrun, Sebastian; Wolfram Burgard; Dieter Fox. Probabilistic robotics. Cambridge, Mass. : MIT Press, 2006.

Wang, Z. Zhao, J. Optical Flow Based Plane Detection for Mobile Robot Navigation. *In Proceedings of the 8th World Congress on Intelligent Control and Automation*, 2011. Pages 1156-1160.