

FLUXO ÓPTICO APLICADO EM NAVEGAÇÃO ROBÓTICA

VINÍCIUS L. SOUZA*, ADEMAR FERREIRA†

**Laboratório de Automação e Controle, Departamento de Engenharia de Sistemas,
Escola Politécnica da Universidade de São Paulo
Av. Prof. Luciano Gualberto, Travessa. 3, 158. CEP: 05508-900 São Paulo, SP, BRASIL*

Emails: luluski@terra.com.br, ademar@lac.usp.br

Abstract— This work presents a technique for use of digital images for application in robotic navigation. Analyzing image sequences obtained in real time, a method (Lucas and Kanade) for obtaining optical flow is implemented, aiming the detection of obstacles in an environment to provide control information to a robot, seeking an autonomous navigation without collisions. The chosen method is implemented in a new architecture, utilizing the operating system multi-task resources, generating high performance results to obtain and apply optical flow diagrams in navigation. As a demonstration, a practical implementation of the algorithm is performed for navigation in an experimental robot, including corridor centering and obstacle avoidance, with promising results.

Keywords— Optical flow, Robotic Navigation, Computer Vision.

Resumo— Neste trabalho apresenta-se uma técnica de utilização de imagens digitais para aplicação em navegação robótica. Analisando-se seqüências de imagens obtidas em tempo real, implementa-se um método de obtenção de fluxo óptico, o de Lucas e Kanade, com o objetivo de detectar obstáculos em um ambiente e conseqüentemente prover informações de controle a um robô, visando uma navegação autônoma sem colisões. O método escolhido é implementado em uma nova arquitetura, utilizando-se de recursos multitarefa do sistema operacional, gerando resultados de alto desempenho para obtenção e aplicação dos diagramas de fluxo óptico na navegação. Como demonstração, realiza-se uma implementação de navegação prática em um robô experimental, incluindo centralização em corredores e desvio de obstáculos, com resultados promissores.

Keywords— Fluxo Óptico, Navegação Robótica, Visão Computacional.

1 Introdução

A visão computacional tem como objetivo a criação de um modelo do mundo real a partir de imagens (Horn, 1986). O resultado do processamento das imagens geralmente é utilizado por um computador para tarefas específicas, em diversas aplicações. Na robótica móvel, os dados provenientes de um sistema de visão podem ser utilizados para o reconhecimento de objetos, mapeamento de ambientes, navegação, dentre outras possibilidades. Um exemplo de tarefa de navegação consiste em o robô partir de uma posição inicial e atingir uma posição final, desviando de obstáculos durante a trajetória.

A visão computacional ainda apresenta muitas deficiências se comparada com a visão biológica, o mais importante sentido do ponto de vista informacional para os animais em geral. No entanto, para algumas tarefas específicas, já é possível seu emprego como um dispositivo sensor eficaz. Uma das técnicas de visão utilizadas na área da navegação robótica é a obtenção do fluxo óptico (Baratoff et al., 2000; Duchon et al., 1998; Young et al., 1992). Trata-se da extração de características de movimento a partir de uma seqüência de imagens, resultando em campos vetoriais bidimensionais. Existem diversos métodos de obtenção de fluxo óptico (Barron et al., 1994; Galvin et al., 1998), cada qual com suas características específicas, e alguns produzindo resultados que podem ser utilizados para uma navegação

robótica eficiente em circunstâncias apropriadas.

A escolha de um método de obtenção de fluxo óptico (FO) propício para aplicações em navegação robótica ainda é uma tarefa difícil. Outrossim, sua implementação pode apresentar diversas abordagens, com variações em termos de desempenho, precisão, dentre outras. Existem também diversas maneiras de utilização destes resultados em navegação robótica. Neste cenário, a motivação do trabalho reside na oportunidade de escolha e implementação de um método de obtenção do FO, através de uma nova abordagem, visando o aproveitamento dos recursos computacionais atuais, tendo em vista uma aplicação em navegação robótica. Este trabalho traz ainda uma breve introdução sobre o FO, justificando a escolha do método diferencial de Lucas e Kanade (Lucas and Kanade, 1981; Lucas, 1984) para a finalidade de navegação robótica.

Este artigo está organizado da seguinte maneira: na Seção 2, é apresentada a definição de FO e a justificativa para a escolha do método implementado. Na Seção 3, é apresentado o modelamento matemático do método escolhido de obtenção de FO e um detalhamento de sua implementação neste trabalho. Já na Seção 4, a aplicação na navegação robótica é exposta e seus resultados são apresentados em duas tarefas de navegação: centralização em corredores e desvio de obstáculos. Por fim, na Seção 5 é apresentada a conclusão do trabalho, com sugestões para futuras implementações.

2 Fluxo Óptico

O FO é um campo vetorial bidimensional induzido pelo movimento relativo entre um observador (câmera) e uma cena observada (ambiente), extraído a partir de uma seqüência de imagens. Trata-se da extração de características dos ambientes por intermédio da análise de variações de intensidade em uma seqüência de imagens, resultando um diagrama composto por vetores bidimensionais de velocidade, correspondentes aos movimentos no espaço tridimensional. Em uma imagem, cada pixel corresponde a um valor de intensidade obtido pela projeção de objetos no espaço sobre o plano da imagem. Com a movimentação dos objetos ou do observador, as projeções correspondentes também alteram suas posições. O FO é obtido através da determinação do campo vetorial de velocidades que mostra a direção e a amplitude das mudanças de intensidade de uma imagem para outra. Isto permite que o movimento relativo seja estimado a partir das mudanças nas imagens.

2.1 Escolha do Método

Os métodos para obtenção do FO podem ser categorizados como: diferenciais, baseados em correlação, em energia e em fase. Cada método possui suas características específicas, podendo ser comparados entre si de acordo com critérios de robustez, precisão, densidade e velocidade de execução.

Algumas análises experimentais comparativas entre as diversas variações destes métodos foram apresentadas anteriormente (Barron et al., 1994; Galvin et al., 1998). Nestas, o algoritmo de Lucas e Kanade (Lucas and Kanade, 1981; Lucas, 1984) apresentou o melhor resultado quantitativo e qualitativo em termos de confiabilidade e consistência entre os diversos testes realizados. Poucos autores apresentam o tempo de execução de seus algoritmos, porém uma nova análise, com um maior enfoque em navegação robótica, foi apresentada por (McCarthy, 2005). Nesta última é confirmado que os métodos diferenciais, pela característica de resolução de um sistema linear como etapa principal de seu algoritmo, apresentam um custo computacional menor se comparado com os demais métodos. Com este histórico apresentado, o *método diferencial local de Lucas e Kanade* será descrito na próxima seção e escolhido para a implementação neste trabalho.

3 Método de Lucas e Kanade

3.1 Descrição do Algoritmo

O método de Lucas e Kanade foi originalmente publicado em 1981, tendo como objetivo apresentar uma nova alternativa para o registro de

imagens, sem um custo computacional alto. O propósito principal era o de localizar a melhor correspondência entre imagens através do cálculo do gradiente espacial de intensidade das mesmas. O diagrama de FO determinado pelo método era o resultado da melhor correspondência entre as imagens analisadas.

Adotando $I(x, y, t)$ como a função intensidade do pixel em (x, y) no instante t , podemos afirmar que, para pequenos deslocamentos, temos a seguinte igualdade, expandida em uma série de Taylor de primeira ordem:

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + I_x \delta x + I_y \delta y + I_t \delta t, \quad (1)$$

onde I_x , I_y e I_t são as derivadas parciais de I em relação a x , y e t respectivamente. Considerando que o brilho varia suavemente com x , y e t , entre duas imagens consecutivas, podemos fazer a hipótese de que a intensidade entre as imagens seja constante, obtendo-se então a *equação de restrição do fluxo óptico*, uma vez que ela expressa uma restrição das componentes u e v do FO:

$$\nabla I \cdot \mathbf{v} + I_t = 0, \quad (2)$$

onde $\mathbf{v} = (u, v)^T$ é o vetor de velocidade em ambas as direções. Entretanto, apenas a Equação 2 não é suficiente para determinar as duas componentes u e v , dificuldade esta conhecida como *problema da abertura*. Para lidar com esta questão, o método assume que o vetor desconhecido de FO é constante dentro de uma vizinhança finita η do pixel. Entretanto, pode-se determinar apenas a componente normal do FO, isto é, a componente na direção do gradiente de intensidade.

Partindo-se da Equação 2, os gradientes obtidos podem ser suavizados através de uma função janela que atribui um maior peso às componentes centrais da vizinhança. Neste caso, é possível determinar o fluxo em cada local, através da minimização da função quadrática ponderada \mathbf{E} (Barron et al., 1994):

$$\mathbf{E} = \sum_{\eta} W^2 \cdot [\nabla I \cdot \mathbf{v} + I_t]^2, \quad (3)$$

onde W^2 representa a função janela de suavização. O par (u, v) que minimiza a Equação 3 satisfaz $\delta_u \mathbf{E} = 0$ e $\delta_v \mathbf{E} = 0$, gerando um sistema linear, que deve ser resolvido para obtenção dos vetores de FO.

3.2 Implementação do Algoritmo

A arquitetura adotada para implementação do algoritmo sugere uma nova forma de distribuição de tarefas para obtenção de FO (Souza, 2007).

Visando otimizar a execução do algoritmo em sincronismo com a captura de imagens de um dispositivo externo e o sistema de controle do robô, foi elaborada uma arquitetura de programação multitarefa, utilizando-se dos recursos oferecidos pelos sistemas operacionais atuais. Desta maneira, o algoritmo foi dividido em três tarefas: a aquisição de imagens, o cálculo do FO e o sincronismo com o sistema de controle do robô. Assim, as tarefas não esperam o início, execução e término das demais para interação das etapas. Elas executam, paralelamente, apenas sincronizando os resultados com as demais para atingir o objetivo final. As atribuições de cada tarefa desta aplicação são expostas a seguir, e detalhadas em (Souza, 2007).

Na primeira tarefa, cada imagem capturada é armazenada em uma memória temporária, sinalizando cada final de ciclo para a tarefa de execução do algoritmo. Esta tarefa absorve todo o tempo necessário para comunicação com o dispositivo de captura e armazenamento das imagens.

Na tarefa de execução do algoritmo, todas as etapas para a obtenção dos diagramas de FO são implementadas em sincronismo com a sinalização de novas imagens. São elas: a aplicação de uma pré-filtragem para suavização das imagens, a estimação das derivadas I_x , I_y e I_t , a suavização dos gradientes para eliminação de ruídos, e, finalmente, a construção e resolução do sistema linear gerado a partir da Equação 3, resultando nos vetores de FO para cada ponto da imagem.

A obtenção de um diagrama de FO completo é sinalizada então para a tarefa de controle, que é encarregada de extrair as informações necessárias dos diagramas obtidos e determinar o melhor comportamento para controle de navegação do robô móvel, conforme será exposto na próxima seção.

A tarefa de captura das imagens opera em uma velocidade de até 15 Hz. As imagens obtidas são armazenadas em um buffer simples, e portanto, a execução do algoritmo só utiliza a imagem mais recente disponível. Esta tarefa, por sua vez, gera diagramas de FO em uma taxa de até 9,38 Hz. Já o sincronismo com o controle consegue gerar informações de velocidade das rodas a uma taxa de cerca de 20 Hz, enviando-as, através da porta serial, a cada nova iteração. Entretanto, a *latência* do sistema de uso do FO para controle é limitada pelo tempo de execução do algoritmo, ou seja, 106 ms (9,38 Hz). Desta maneira, com a velocidade do robô a 7,5 cm/s, temos uma nova informação de controle a cada 0,8 cm de trajetória aproximadamente. Isto indica que a velocidade do robô poderia ser maior, e ainda assim a taxa de informações de controle seria adequada.

Esta velocidade foi obtida utilizando-se de um microcomputador com processador Pentium 1.6 GHz. Estes resultados mostram uma melhoria de desempenho em relação aos resultados obtidos por (McCarthy, 2005), que, para este algoritmo,

obteve uma taxa de cerca de 3,80 diagramas por segundo, utilizando um processador Pentium 866 MHz.

4 Aplicação em Navegação e Resultados

A plataforma utilizada para os testes propostos por este trabalho é um robô experimental, desenvolvido no Laboratório de Automação e Controle do Departamento de Engenharia de Telecomunicações e Controle da Escola Politécnica da USP (Augusto, 2007; Augusto and Ferreira, 2007). O aplicativo de obtenção de FO deste trabalho é executado no computador embarcado no robô, com o objetivo de avaliar o seu potencial para auxiliar a navegação robótica. Utiliza-se uma câmera USB acoplada, cujo sensor é do tipo CCD, com ângulo de abertura de aproximadamente 40°, configurada para obter imagens de 160x120 pixels. A interface entre o algoritmo e o controle do robô é feita através da troca de mensagens entre os aplicativos via comunicação serial.

4.1 Estratégia de Controle

Hoje, na área de navegação robótica, sugere-se a aplicação do FO em níveis hierárquicos baixos, atuando diretamente nos laços de controle dos robôs (McCarthy, 2005). Algumas implementações inspiram-se ainda nos sistemas biológicos, como apresentado por (Baratoff et al., 2000; Snirivasan and Zhang, 2000) para execução de tarefas de controle.

A aplicação de navegação realizada por este trabalho revela algumas dificuldades que devem ser consideradas na escolha da estratégia de controle. Uma delas é o efeito de rotação da câmera quando o robô desvia de uma trajetória retilínea. Esta rotação gera uma acentuação dos vetores de FO em toda a imagem, podendo levar a resultados indesejados de controle. Ainda, algoritmos complexos de controle requerem a utilização de um hardware também complexo (Coombs et al., 1998), como controle ativo do eixo de visada ("gaze") da câmera.

Tendo em vista dificuldades do tipo acima e os propósitos deste trabalho, decidiu-se implementar uma técnica de controle relativamente simples, dada pela *estratégia de balanço* (Duchon et al., 1998). Por sua simplicidade, permite realçar os efeitos do uso do FO no controle da navegação do robô.

Para construir a estratégia de balanço, o campo de FO resultante é dividido verticalmente em duas regiões. Em seguida, é implementado um somador, que faz a totalização dos módulos dos vetores de FO em cada região da imagem. Desta maneira, é possível detectar a região da imagem com maior incidência de obstáculos, proporcionalmente à densidade de vetores encontrada.

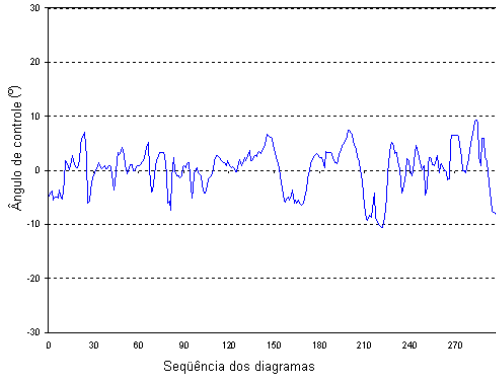


Figura 1: Gráfico do ângulo de controle calculado através da estratégia de balanço durante a trajetória no corredor sem obstáculos.

A Equação 4 representa a lei de controle definida por (Duchon et al., 1998) para a estratégia de balanço:

$$\Delta(F_E - F_D) = k \cdot \frac{\sum \|\vec{w}_E\| - \sum \|\vec{w}_D\|}{\sum \|\vec{w}_E\| + \sum \|\vec{w}_D\|}, \quad (4)$$

onde $\Delta(F_E - F_D)$ é a diferença de esforços de controle nas duas regiões do FO, k é uma constante e $\sum \|\vec{w}\|$ é a somatória das amplitudes de FO nas regiões da esquerda E e da direita D . A diferença de esforços resultante desta equação pode ser interpretada como um ângulo de controle ϕ , que indica qual a melhor direção de trajetória para o robô em cada instante. O valor deste ângulo é aplicado diretamente no cálculo das velocidades das rodas, o que, neste trabalho, propomos que seja realizado pelas equações:

$$V_E = V_{REF} \cdot (1 + \phi), \quad (5)$$

$$V_D = V_{REF} \cdot (1 - \phi), \quad (6)$$

onde V_E e V_D correspondem às velocidades da roda esquerda e direita respectivamente (o robô possui guiagem diferencial) e V_{REF} corresponde à velocidade de referência adotada para a navegação.

Para propiciar maior estabilidade ao robô e minimizar a influência de ruídos durante a navegação, o ângulo ϕ é comparado com um limiar de controle. Caso seu valor seja menor que o limiar, ambas as velocidades ficam iguais a V_{REF} . Este limiar pode ser definido na interface de controle durante a execução das experiências, de acordo com o ambiente de teste utilizado.

4.2 Resultados da Navegação

Nesta seção, são expostos os resultados de um experimento de centralização em corredor e outro



Figura 2: Seqüência de imagens conforme robô se aproxima e desvia do obstáculo.

de desvio de obstáculos realizados neste trabalho. Para estas experiências, foram adotadas as seguintes especificações:

- A velocidade de referência das rodas V_{REF} foi ajustada para 7,5cm/s;
- A constante k da equação de cálculo do ângulo de controle ϕ foi ajustada para 0.25;
- O limiar de controle do ângulo ϕ foi ajustado para 3°;

No primeiro experimento o robô foi posicionado no início do corredor, sem obstáculos que o obrigassem a realizar um desvio no percurso. Devido a características do ambiente (presença de janelas e portas em uma das paredes) foram detectados vetores de FO diferentes de zero durante toda a trajetória. Entretanto, como a estratégia de controle tende a balancear a somatória das duas laterais dos diagramas, o comportamento do robô, apesar de um pouco oscilatório, foi o de manter-se na posição central. Isto pode ser comprovado no gráfico da Figura 1, onde se observa que a média temporal do ângulo de controle é próxima a zero. O resultado é uma navegação sem colisões durante a trajetória e com a centralização no corredor sendo obtida com sucesso.

No segundo experimento, um objeto de grande porte foi posicionado na trajetória do robô na lateral direita do corredor, obrigando-o a desviar para a esquerda para evitar uma colisão. Na Figura 2, temos uma visão externa desta navegação, com o desvio do obstáculo ocorrendo conforme esperado. Já na Figura 3, observam-se algumas imagens de teste adquiridas pelo robô durante a trajetória, nos momentos em que ele se aproxima e desvia do obstáculo. Os diagramas de FO determinados durante a trajetória, correspondentes aos instantes das imagens de teste apresentadas (Figura 3), podem ser vistos na Figura 4.

Na Figura 4(a) já é possível identificar o contorno do obstáculo a ser desviado. Conforme o robô avança em sua trajetória, os vetores de FO referentes ao obstáculo aumentam de intensidade,

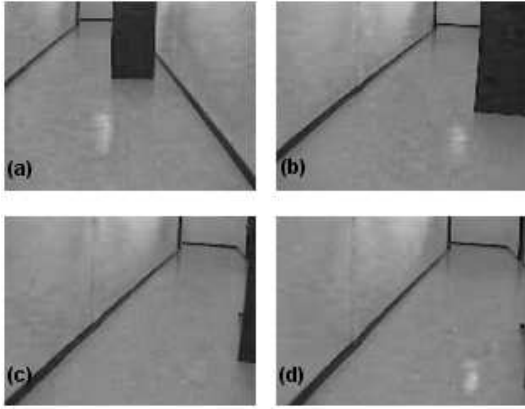


Figura 3: *Imagens de teste do corredor com o obstáculo à direita obtidas durante a trajetória: (a) Início da trajetória; (b) Início do desvio do obstáculo; (c) Término do desvio do obstáculo; (d) Sequência da trajetória após desvio.*

de acordo com Figura 4(b), forçando assim um desvio para o lado esquerdo. Enquanto o robô está desviando do obstáculo, devido à sua rotação, os vetores de FO são amplificados, como pode ser visto na Figura 4(c), porém, quando ambos os lados do diagrama são balanceados, o robô retoma a trajetória retilínea, passando o obstáculo sem colisão, conforme Figura 4(d).

No gráfico da Figura 5 pode-se observar a trajetória do robô neste percurso, com o obstáculo sendo representado pela marca 'X'. Valores positivos representam um movimento para a esquerda, enquanto que valores negativos representam um movimento para a direita, equivalente ao robô movimentando-se na direção do eixo. Nota-se, ainda, que o robô permanece em uma rota de desvio até que o obstáculo seja transposto, só então permitindo que se retome uma trajetória retilínea no restante da navegação.

4.3 Discussão dos Resultados

Para que estes resultados de navegação fossem obtidos, diversos experimentos preliminares de coleta de dados foram executados. O que se observa na literatura é que muitas vezes uma intervenção no ambiente é feita para que se obtenham resultados satisfatórios utilizando apenas o FO como dispositivo sensor para navegação. Esta modificação consiste principalmente em adicionar texturas a paredes, de modo a prover um maior nível de informação para o cálculo do FO. Um exemplo disso pode ser visto em (McCarthy, 2005), onde as paredes do ambiente são texturizadas para se obter uma centralização de corredor eficiente.

Nos experimentos realizados não se interferiu no ambiente. O corredor utilizado para os testes apresenta características notáveis para navegação, como os rodapés pretos e as paredes brancas. Di-

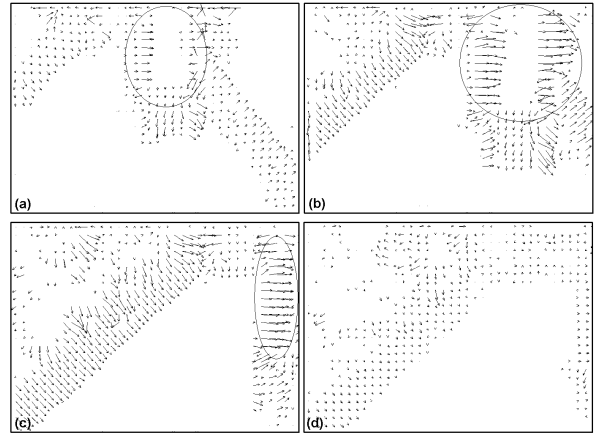


Figura 4: *Diagramas de FO resultantes das imagens de teste do corredor com o obstáculo à direita*

ficultades foram encontradas, entretanto, ao tentar diferenciar os obstáculos de características do ambiente, como portas e janelas. Para superar estas dificuldades, a câmera do robô foi posicionada em um ângulo de -45 graus em relação à linha de visão normal. Deste modo, conseguiu-se dar maior influência aos obstáculos e características mais próximos ao robô, com resultados satisfatórios. Tal recurso possibilitou o uso de FO para navegação em ambientes não preparados, mediante estratégias simples de controle. Este tipo de alteração foi por nós, posteriormente, também encontrado na literatura (Coombs et al., 1998).

5 Conclusões

Neste trabalho, foi apresentada uma implementação do algoritmo de obtenção de FO de Lucas e Kanade (Lucas and Kanade, 1981; Lucas, 1984) para aplicação em navegação robótica. Com a utilização dos recursos multitarefa presentes nos sistemas operacionais atuais, as tarefas necessárias para obtenção de imagens, cálculo do FO e sincronismo com o sistema de controle de um robô móvel foram divididas em diferentes fios de processamento (threads), otimizando a execução das mesmas e obtendo um melhor desempenho do que uma implementação sequencial. Uma outra vantagem obtida com a nova arquitetura utilizada é a possibilidade de implementação em um hardware com múltiplos processadores, onde os diversos fios de processamento são divididos automaticamente pelo sistema operacional entre os processadores, possibilitando maior velocidade de execução do algoritmo.

Os experimentos de navegação executados neste trabalho mostraram que a implementação do algoritmo de FO realizada, em conjunto com a estratégia de balanço, possibilita a geração de informações de controle em intervalos de tempo suficientemente pequenos, permitindo sua aplicação

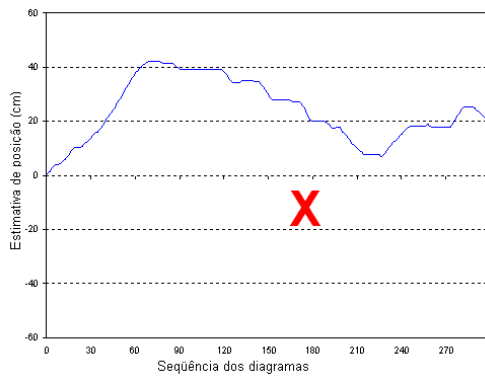


Figura 5: Gráfico da estimativa de posição acumulada no corredor com um obstáculo à direita determinada a partir das velocidades das rodas durante a trajetória.

em tarefas simples de navegação, como centralização em corredores e desvio de obstáculos.

Entretanto, os resultados mostraram também que a utilização do FO como fonte única para navegação pode não ser satisfatória em todos os ambientes, demandando estratégias de controle mais complexas para uma possível solução adequada.

5.1 Trabalhos Futuros

A flexibilidade obtida, a partir da arquitetura multitarefa, possibilita que sejam feitas análises mais apuradas entre a distribuição de tarefas de modo a otimizar ainda mais a execução do algoritmo, visando melhora de desempenho através do rebalanceamento de etapas mais lentas.

Estes resultados abrem um novo caminho para implementações de FO em sistemas que necessitem de uma resposta rápida, porém com informações confiáveis de controle. Certamente, a fusão deste algoritmo de FO com sinais de outros sensores, ou até mesmo com outras informações extraídas das imagens, possibilitará uma navegação mais autônoma, auxiliando tanto no controle como no planejamento dos robôs. Além de sua utilização direta em uma malha de controle, o FO pode também ser considerado como um sensoriamento importante em uma arquitetura robótica, provendo informações úteis para auxiliar uma unidade de planejamento a alcançar objetivos maiores.

Referências

- Augusto, S. R. (2007). *Uma Plataforma Robótica para Estudos de Autonomia*, Tese de Doutorado - USP.
- Augusto, S. R. and Ferreira, A. (2007). *A Robotic Platform for Autonomy Studies*, Aceita para

apresentação oral no ICINCO'07 - International Conference on Informatics in Control, Automation and Robotics, Angers, França.

- Baratoff, G., Toepfer, C. and Neumann, H. (2000). *Combined Space-variant Maps for Optical Flow Based Navigation*, Biological Cybernetics, 83, pp. 199-209.
- Barron, J. L., Fleet, D. J. and Beauchemin, S. S. (1994). *Performance of Optical Flow Techniques*, International Journal of Computer Vision, 12:1, pp. 43-77.
- Coombs, D., Herman, M., Hong, T. and Nashman, M. (1998). *Real-Time Obstacle Avoidance Using Central Flow Divergence and Peripheral Flow*, IEEE Transactions on Robotics and Automation, 14(1), pp. 49-59.
- Duchon, A. P., Warren, W. H. and Kaelbling, L. P. (1998). *Ecological Robotics*, Adaptive Behavior, vol. 6, number 3/4.
- Galvin, B., McCane, B., Novins, K., Mason, D. and Mills, S. (1998). *Recovering Motion Fields: An Evaluation of Eight Optical Flow Algorithms*, Proceedings of the Ninth British Machine Vision Conference.
- Horn, B. K. P. (1986). *Robot Vision*, McGraw Hill - The MIT Press.
- Lucas, B. D. (1984). *Generalized Image Matching by the Method of Differences*, Dissertação de PhD - Carnegie-Mellon University.
- Lucas, B. D. and Kanade, T. (1981). *An Iterative Image Registration Technique with an Application to Stereo Vision*, Proceedings of DARPA Image Understanding Workshop, pp. 121-130.
- McCarthy, C. D. (2005). *Performance of Optical Flow Techniques for Mobile Robot Navigation*, Dissertação de Mestrado - Melbourne University.
- Snirivasan, M. V. and Zhang, S. (2000). *Visual Navigation in Flying Insects*, International Review of Neurobiology, 44, pp. 67-92.
- Souza, V. L. (2007). *Fluxo Óptico Aplicado em Navegação Robótica*, Dissertação de Mestrado - USP.
- Young, G., Hong, T., Herman, M. and Yang, J. C. S. (1992). *New Visual Invariant for Obstacle detection Using Optical Flow Induced from General Motion*, Proceedings of IEEE Workshop Appl. Comput. Vision, pp. 100-109.