

UM ARCABOUÇO PARA LOCALIZAÇÃO DE ENXAMES DE ROBÔS

RENATO F. GARCIA*, PEDRO M. SHIROMA*, LUIZ CHAIMOWICZ*, MARIO F. M. CAMPOS*

**VeRLab – Laboratório de Visão e Robótica
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, MG, Brasil*

Emails: {renatofg, pshiroma, chaimo, mario}@dcc.ufmg.br

Abstract— This paper presents a framework for robot swarm localization. Robots are tagged with geometrical markers and a group of overhead cameras is used to localize and uniquely identify the robots. The use of geometrical markers makes the system scalable to large numbers of robots while the association of multiple cameras allows the coverage of a larger work area. A modular and distributed software system that may run in different computers is responsible for gathering information from multiple cameras and localize the robots. Real experiments show the accuracy, robustness and scalability of the framework.

Keywords— Robot Swarms, Localization.

Resumo— Esse artigo apresenta um arcabouço para a localização de enxames de robôs. Os robôs carregam marcos visuais geométricos que são identificados e localizados por um conjunto de câmeras colocadas sobre a área de trabalho. O uso de marcos geométricos torna o sistema escalável para um grande número de robôs, enquanto a associação de múltiplas câmeras permite uma maior área de cobertura. Foi desenvolvido um sistema de software distribuído capaz de localizar os robôs a partir de informações obtidas por câmeras conectadas em múltiplos computadores. Experimentos foram realizados de forma a demonstrar a exatidão, robustez e escalabilidade do arcabouço desenvolvido.

Keywords— Enxames de Robôs, Localização.

1 Introdução

De forma geral, a robótica cooperativa consiste na utilização de grupos de robôs trabalhando de maneira coordenada na execução de diversas tarefas, trazendo ganhos de desempenho, uma maior tolerância a falhas e possibilitando a execução de tarefas que não poderiam ser realizadas por um único robô. Dentro da robótica cooperativa, uma área nova e que tem atraído bastante interesse recentemente é a utilização de grupos compostos por um grande número de robôs mais simples, que individualmente não possuem muita capacidade, mas em conjunto podem realizar diversos tipos de tarefas. Esses grupos são genericamente chamados de “enxames” de robôs (*swarms*). Um dos principais desafios nessa área é o desenvolvimento de mecanismos que sejam escaláveis de acordo com o número de indivíduos de forma que grupos com 10, 50, 100 ou 1000 robôs possam ser controlados eficientemente.

Grande parte dos trabalhos nessa área tem focado no planejamento e controle de trajetórias para enxames robôs, por exemplo (Kloetzer and Belta, 2006; Kamphuis and Overmars, 2004; Chaimowicz et al., 2005). Na maior parte desses trabalhos são realizadas simulações que assumem que os robôs possuem uma estimativa precisa de sua postura. Infelizmente, na prática, isso é uma tarefa difícil devido ao conjunto limitado de sensores que normalmente equipam esse tipo de robô. Portanto, para realização de experimentos com enxames, é necessário desenvolver mecanismos es-

caláveis que permitam localizar e identificar unicamente uma grande quantidade de robôs.

O objetivo deste trabalho é desenvolver um arcabouço que permita a localização e identificação de grandes grupos de robôs. O arcabouço é composto por um conjunto de câmeras fixado sobre a área de trabalho e um sistema de software capaz de localizar e identificar os robôs, que nesse caso carregam algum tipo de marco visual.

Como será discutido na próxima seção, apesar de sistemas similares já terem sido desenvolvidos, poucos são adequados para se trabalhar com grandes grupos de robôs. Nesse sentido, esse artigo traz algumas contribuições importantes: o uso de marcos visuais geométricos permite a determinação da postura e identificação única de uma grande quantidade de robôs. Por sua vez, o sistema de software desenvolvido é modular e distribuído, o que permite o uso de múltiplas câmeras em diferentes computadores de forma completamente transparente. Essas características tornam o sistema escalável, o que é de fundamental importância quando se trabalha com enxames de robôs.

Este artigo está organizado da seguinte forma: a próxima seção discute alguns trabalhos relacionados na área. A Seção 3 apresenta o arcabouço de localização implementado detalhando o sistema de câmeras, os marcos visuais utilizados e o software desenvolvido. A Seção 4 apresenta os experimentos realizados de forma a medir o desempenho do sistema e a Seção 5 traz as conclusões e perspectivas de trabalhos futuros.

2 Trabalhos Relacionados

Localizar precisamente um robô representa um dos problemas centrais em robótica móvel. Uma das primeiras técnicas concebidas foi a odometria e, ainda hoje, apesar de seus problemas, é possível encontrar trabalhos onde ela é utilizada como fonte principal de dados (Mei et al., 2006). A integração do GPS com odometria (Goel et al., 1999; Aono et al., 1998) consistiu num avanço importante ao impor limites no erro absoluto, permitindo uma localização robusta por longos períodos de tempo. Porém tais técnicas estão restritas a ambientes externos. Em ambientes internos, caso o mapa seja conhecido *a priori* pode-se utilizar métodos de localização probabilísticos (Dellaert et al., 1999) para estimar a posição do robô. Caso contrário algoritmos de mapeamento e localização simultâneos (SLAM) (Thrun et al., 1998) são frequentemente usados.

Entretanto, as abordagens anteriores requerem robôs com alta capacidade sensorial e computacional, conflitando diretamente com a filosofia de enxames de robôs. Uma segunda abordagem consiste no desenvolvimento de um sistema externo de localização. Em tais sistemas um marco artificial é fixado em cada objeto de interesse e uma câmera é geralmente posicionada em uma parte superior da cena. O marco deve ser projetado de forma a permitir a detecção, identificação e localização do objeto cuja trajetória será rastreada. Nestas abordagens o erro independe do estado interno do robô.

Marcos baseados em padrões coloridos são geralmente utilizados por apresentarem uma implementação simples, eficiente e relativamente robusta. Um exemplo onde eles são bastante utilizados é nas competições do futebol de robôs (Bruce and Veloso, 2003; Cerqueira et al., 2005; Bianchi and Reali-Costa, 2000). Entretanto, a medida que o número de marcos diferentes cresce, a separabilidade entre as cores diminui, degradando a robustez do sistema. Outra deficiência apresenta-se quando há mudança de iluminação alterando as cores dos objetos percebidas pela câmera.

Por outro lado, marcos capazes de emitir energia podem ser usados para melhorar o processo de detecção. Por exemplo, (Hada and Takase, 2001) desenvolveu o iGPS, um sistema baseado em LED's infravermelhos capaz de localizar diversos marcos com alta precisão usando equipamentos de baixo custo. Porém ele é incapaz de associar identidades distintas e o espaço físico requerido para a sua colocação no robô torna-o proibitivo para o uso em enxames.

Mais recentemente padrões geométricos foram utilizados para identificar e localizar robôs (Hosoi and Sugimoto, 2006). Neste caso o marco é composto de uma camada externa, responsável pela detecção e localização, e uma camada interna, re-

sponsável pela identificação. O uso de algoritmos de casamento de padrões permite a representação de um número maior de objetos.

Neste trabalho é apresentado um sistema de localização externo adequado para um grande grupo de robôs de pequeno porte. O sistema é modular e flexível, permitindo a utilização de diferentes câmeras e tipos de marcos visuais artificiais. Como mencionado, o principal diferencial desse trabalho é a preocupação em se desenvolver um sistema escalável, que possa ser utilizado com enxames de robôs.

3 Arcabouço para Localização

O arcabouço proposto é composto por um conjunto externo de câmeras fixadas sobre a área de trabalho. Os robôs a serem localizados carregam marcos que possibilitam a sua identificação e localização. A Figura 1 mostra um diagrama do sistema desenvolvido.

A implementação foi estruturada de forma que os módulos de processamento de cada câmera ficassem separados do componente responsável pela fusão das informações obtidas por cada um dos módulos. Cada câmera, ou um conjunto delas, pode trabalhar em máquinas diferentes interligadas em rede, permitindo a distribuição da execução entre vários computadores. Isso traz vantagens tanto em termos de desempenho e escalabilidade, uma vez que o processamento é dividido por várias máquinas, quanto em termos de cobertura, pois distribuindo fisicamente as câmeras e os computadores é possível montar a infra-estrutura sem que haja a necessidade de cabos muito longos ou o uso de hubs e repetidores.

Para permitir isso, a implementação do arcabouço foi dividida em dois programas diferentes. O primeiro é um programa cliente que controla as câmeras e cuida da aquisição de informações sobre a localização dos alvos que se encontram em seu campo de visão. Este programa pode ser instanciado em diferentes computadores, para cada câmera conectada. O segundo é um programa servidor, cuja função é centralizar as informações obtidas em todas as instâncias dos clientes, e fornecer como saída a localização de todos os alvos encontrados.

3.1 Câmeras

Cada programa cliente gerencia uma ou mais câmeras. Essas podem ser câmeras de vídeo físicas, que farão a captura e digitalização de imagens do ambiente externo; ou uma câmera virtual, que fornecerá ao programa imagens lidas do disco.

A câmera física pode ser de qualquer marca ou modelo disponível, desde que seja possível criar uma classe que encapsule a sua API e que seja capaz de fornecer como retorno uma imagem digital

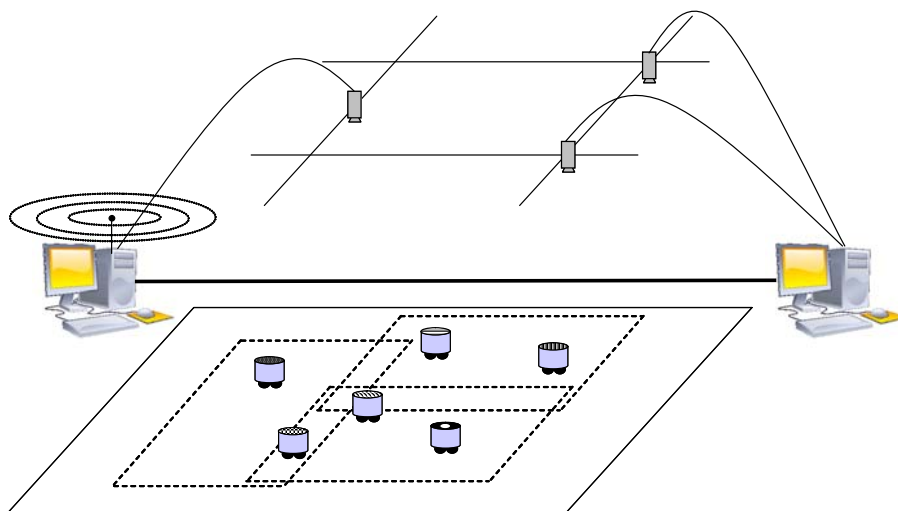


Figura 1: Diagrama do arcabouço: o uso de múltiplas câmeras permite a obtenção de uma maior cobertura. As câmeras podem ficar distribuídas por vários computadores ligados em rede e um computador é responsável por centralizar as informações enviar aos robôs.

em formato apropriado. Cada câmera física deve ser calibrada para se determinar os seus parâmetros intrínsecos (foco, distorção radial, etc), e a sua matriz de transformação, nesse caso uma homografia. Esses parâmetros correlacionam a posição dos robôs nas imagens, dada em pixels, com a sua posição no mundo real. A homografia é computada considerando-se que os objetos a serem localizados estão todos no mesmo plano. Isso facilita as transformações de coordenadas no plano da imagem para as coordenadas do mundo. Nessa aplicação específica, isso não é um problema uma vez que os swarms são geralmente compostos por robôs idênticos que irão se movimentar no plano do chão.

O uso de uma câmera virtual é útil na realização de experimentos mais controlados (por exemplo testar diferentes algoritmos com a mesma seqüência de imagens), ou no ajuste dos parâmetros das câmeras físicas para uma configuração adequada ao ambiente. Nesse caso, é feito o registro das imagens a serem analisadas usando-se a câmera física. Dessa forma haverá uma maior facilidade e certeza de reprodutibilidade da seqüência de movimentos realizados quando elas forem lidas do disco pela câmera virtual.

3.2 Marcos Visuais

O sistema de localização implementado funciona por meio de reconhecimento de marcos pré-determinados. Cada robô deve carregar um ou mais marcos de forma que eles sejam identificados unicamente pelas câmeras.

No arcabouço desenvolvido podem ser utilizados diferentes tipos de marcos visuais. Um modelo simples de marco bastante utilizado em diversas aplicações se baseia em cores. Nesse caso, cada

robô leva consigo dois marcos circulares com cores diferentes. Um desses círculos possui uma cor que é comum a todos os outros robôs, já a segunda cor identifica unicamente cada um deles. A composição dos dois marcos permite não só determinar a posição do robô mas também a sua orientação.

O processo de localização por cor é feito primeiramente filtrando-se a imagem obtida pela câmera. Com isso, são segmentados os pixels cujos valores estejam dentro de algum dos intervalos que definem cada uma das possíveis cores (dados por uma tabela de cores no espaço YUV). Cada aglomerado de pixels filtrados com área suficientemente grande para corresponder a um alvo é chamado de *blob*, e são eles somente que permanecerão na imagem após a filtragem. Usando a biblioteca OpenCV (Intel, 2007) é determinado o centro de cada blob. Essa posição na imagem é recalculada para corrigir a distorção radial, e multiplicada por uma matriz de homografia, então a posição real do blob é encontrada. A etapa final, de identificação dos pares de blobs correspondentes a um robô e a determinação de sua pose, é feita no programa servidor, uma vez que os blobs podem ter sido observados em diferentes câmeras, caso o robô esteja muito próximo das bordas dos campos de visão.

Como mencionado na Seção 2, o principal problema dos marcos visuais baseados em cores é a dificuldade de se identificar unicamente os robôs quando o número de robôs aumenta. Normalmente, devido à complexidade inerente ao processo de determinação da tabela de cores, esses sistemas não conseguem diferenciar de forma exata mais do que 8 cores diferentes. Portanto, como o arcabouço visa a localização de grandes grupos de robôs, outra alternativa deve ser utilizada. Nesse caso, decidiu-se por utilizar marcos geométricos



Figura 2: Exemplos de marcos visuais da biblioteca ARToolKitPlus.

fornecidos pela biblioteca ARToolKitPlus (Wagner and Schmalstieg, 2007). As marcas identificadas por ela são formadas por uma matriz quadrada 4x4, onde cada item pode ser branco ou preto, mais uma borda da mesma largura que um item, sempre preta (Figura 2). Os padrões de preenchimento dessa matriz são escolhidos para que não apresentem simetria entre si ou simetria própria quando o robô girar, e assim pode-se conhecer não somente a sua posição mas também a sua orientação a cada instante. Diferentemente da localização por cores, somente com uma marca já é possível conhecer a identidade e a pose do robô. Além disso, a biblioteca ARToolKitPlus já faz a segmentação dos marcos, bastando ao programa cliente aplicar a matriz de homografia e transmitir os dados ao servidor.

3.3 Software

O sistema de software foi desenvolvido de maneira completamente modular. Dessa forma, diferentes tipos de câmeras e métodos de identificação podem ser utilizados de forma transparente. Além disso, como mencionado, o arcabouço permite a distribuição das câmeras por múltiplos computadores através de um mecanismo cliente-servidor. Esse mecanismo também é independente de arquitetura, podendo rodar em diferentes tipos de máquinas e sistemas operacionais.

O funcionamento segue o seguinte protocolo: ao ser executado, o servidor ficará esperando conexões de clientes em uma porta pré-determinada. Ao receber uma solicitação de conexão um cliente, o servidor responde com o número de uma porta disponível. Cada cliente por sua vez possui uma *thread* principal que solicita a conexão e, ao receber a resposta do servidor, se conecta na porta livre. Nesse intermédio, o servidor cria uma *thread* que escutará a porta livre cujo número foi enviado como resposta, e após estabelecida uma conexão cliente-servidor, cuidará dos dados recebidos do cliente. Toda comunicação entre os programas é feita utilizando o protocolo UDP.

Conexão estabelecida, as várias instâncias dos programas clientes irão enviar continuamente as informações recolhidas do ambiente, e estas serão agrupadas em um container existente no servidor. De acordo com o tipo de dado presente, obtidos

de marcos geométricos ou blobs, uma função percorrerá esse container periodicamente, e determinará a pose do conjunto dos robôs identificados. Um ponto importante é que, para haver cobertura total da área de trabalho, é necessário que haja sobreposição das áreas de cobertura de cada câmera. Caso um marco seja identificado em mais de uma câmera, o servidor se encarrega em calcular a correta posição desse marco. Especificamente, o arcabouço implementado utiliza a média das posições obtidas.

Por fim, a informação da localização de cada robô poderá ser fornecida a um terceiro programa, que irá utilizar esses dados ou enviá-los diretamente aos robôs.

4 Experimentos

Foram realizados diversos experimentos para avaliar a exatidão, escalabilidade e robustez do arcabouço de localização desenvolvido.

Nos experimentos realizados, foram utilizadas três câmeras modelo *Dragonfly* montadas com lentes de 4mm. Elas foram fixadas no teto a uma altura aproximada de 2,7m e orientadas perpendicularmente ao plano do chão. Todas as câmeras foram configuradas para adquirir imagens a uma taxa de 30 quadros por segundo e trabalhar com uma resolução de 640x480 pixels. Os parâmetros de tempo de exposição, abertura, ganho e distância focal de cada câmera foram configurados manualmente objetivando aumentar o contraste entre as regiões branca e preta dos marcos. A calibração intrínseca foi realizada usando uma toolbox do Matlab. Cinco pontos de coordenadas conhecidas relativas a um referencial global foram escolhidos e, após a correção radial da imagem, a matriz de homografia foi calculada.

De forma a determinar a exatidão do sistema, um marco visual da biblioteca ARToolKitPlus de 10cm de largura foi fixado na extremidade de uma haste rígida com 50cm de comprimento. A outra extremidade possui um orifício que foi fixado em seqüência em cinco pontos diferentes do ambiente. Rotacionando essa haste, foi possível obter trajetórias circulares para os marcos. A Figura 3 mostra as trajetórias efetuadas pelos marcos e também a área de cobertura de cada câmera.

Pode-se observar que os marcos foram localizados corretamente em relação aos círculos bases (*ground-truth*), mesmo nas áreas de sobreposição das câmeras. Considerando-se que em cada rotação o marco descreveu um círculo de 50cm, o erro médio obtido foi de -0,18cm, o desvio padrão 0,57cm, e o erro máximo foi de 2,29cm, o que demonstra um alto grau de exatidão do sistema. O fato dos pontos não se encontrarem espaçados regularmente no gráfico da Figura 3 é causado por imperfeições no mecanismo manual de rotação utilizado.

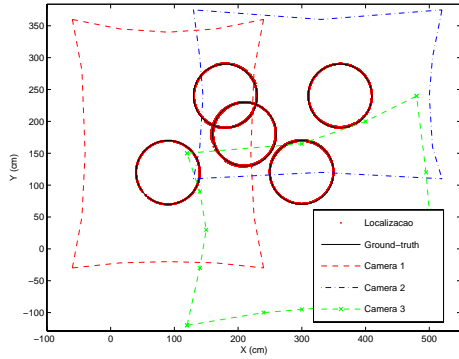


Figura 3: Trajetórias circulares realizadas por um marco e a área de cobertura de cada câmera.

De forma a observar o erro individual de cada câmera, o raio do círculo foi alterado para 80cm, de forma que cada câmera observasse completamente uma revolução. A nova configuração foi repetida para cada câmera isoladamente e os erros médios calculados. A tabela 1 mostra o erro médio e desvio padrão determinado para cada câmera. Nota-se uma pequena diferença entre as câmeras, o que provavelmente se deve a imperfeições no processo de calibração.

	Erro médio (cm)	Desv. padrão (cm)
Câmera 1	-0,5157	0,2544
Câmera 2	0,0353	0,3641
Câmera 3	-1,0701	0,5888

Tabela 1: Erro médio observado por cada câmera ao medir a trajetória de um círculo de raio 80cm.

Para analisar a escalabilidade do sistema foram feitos dois experimentos. No primeiro investigou-se a confiabilidade do sistema na localização simultânea de múltiplos marcos em movimento. Para isso, sete marcos diferentes foram colocados juntos no topo de um robô *Pioneer 3AT*, que descreveu uma trajetória sob as câmeras. O resultado obtido pode ser observado na figura 4. Os erros de localização corresponderam a 0,2% das operações efetuadas.

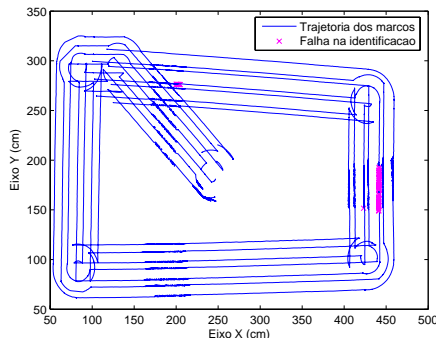


Figura 4: Trajetória de sete marcos simultâneos.

Em seguida verificou-se a escalabilidade do sistema em relação ao número de câmeras e ao número de marcos. Para isso, três câmeras foram conectadas em um Pentium Core Duo 2.8 Ghz com 1Gb de memória RAM, e quarenta marcos diferentes foram divididos entre os campos de visão das três câmeras na proporção 13-13-14. O processo consistiu em localizar simultaneamente os marcos em trezentos frames, sendo que após este intervalo um marco era retirado do conjunto. Esse procedimento foi repetido até que não houvesse mais marcos, e o resultado obtido pode ser visto na figura 5.

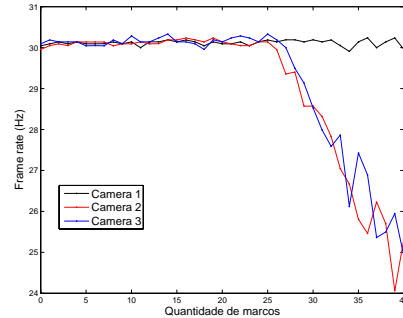


Figura 5: Teste de escalabilidade do sistema.

Observe que, enquanto a curva para a câmera 1 permanece constante, as outras duas câmeras possuem um decaimento equitativo a medida que o número de marcos aumenta. Este comportamento decorre do uso de um único computador rodando múltiplas *threads* nessa implementação. Apesar disso, o sistema apresentou desempenho adequado, sendo capaz de processar até 25 marcos com frequência máxima (30Hz), decaindo para 25Hz quando 40 marcos estavam presentes na cena. É importante ressaltar que nesse experimento as três câmeras estavam conectadas ao mesmo computador. Apesar de ainda não terem sido realizados experimentos utilizando-se múltiplas câmeras em múltiplos computadores, espera-se que o sistema seja capaz de manter a frequência de 30Hz para um número bem maior de marcos. Em testes realizados com uma só câmera o decaimento na taxa de quadros se deu a partir de 25 marcos, o que permite inferir que o sistema seria capaz de detectar aproximadamente 75 marcos a taxa de 30Hz com três computadores.

Por fim, durante os testes de escalabilidade os marcos foram detectados corretamente em 99.91% dos casos demonstrando o alto grau de robustez do sistema.

5 Conclusões

Esse artigo apresentou um arcabouço de localização para grandes grupos de robôs. Os robôs carregam marcos visuais geométricos que são identificados e localizados por um conjunto de câmeras

colocadas sobre a área de trabalho. Os experimentos demonstraram um alto nível de exatidão, com erros médios de menos de 1cm na localização dos marcos. Além disso, o uso de marcos visuais geométricos e um sistema de processamento distribuído permitiu o desenvolvimento de um sistema escalável, o que é de fundamental importância quando se trabalha com enxames de robôs.

Esse arcabouço será utilizado em diversos projetos do Laboratório de Robótica e Visão Computacional da UFMG (VeRLab). No momento, o VeRLab está adquirindo um conjunto de robôs *e-puck* (www.e-puck.org) para a realização de experimentos com grupos de robôs. A idéia é implementar em robôs reais diversos algoritmos de coordenação que foram testados principalmente em simulação (Chaimowicz et al., 2005; Pimenta et al., 2006). Além disso, ele poderá ser utilizado em outros experimentos em robótica móvel que necessitem de um mecanismo de *ground-truth*, em face da exatidão dos resultados obtidos.

Agradecimentos

Os autores gostariam de agradecer o apoio do CNPq e Fapemig na realização deste trabalho.

Referências

- Aono, T., Fujii, K., Hatsuoto, S. and Karniya, T. (1998). Positioning of vehicle on undulating ground using gps and dead reckoning, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Leuven, Belgium, pp. 3443–3448.
- Bianchi, R. A. and Reali-Costa, A. H. (2000). O sistema de visão computacional do time futeboli de futebol de robôs, *Congresso Brasileiro de Automática*, pp. 2156–2161.
- Bruce, J. and Veloso, M. (2003). Fast and accurate vision-based pattern detection and identification, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taiwan.
- Cerqueira, A. C. T., de A. Lins, F. C., Pereira, J. P. P., Alsina, P. J. and Medeiros, A. A. D. (2005). O time POTI de futebol de robôs da UFRN, *Simpósio Brasileiro de Automação Inteligente*.
- Chaimowicz, L., Michael, N. and Kumar, V. (2005). Controlling swarms of robots using interpolated implicit functions, *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 2498–2503.
- Dellaert, F., Fox, D., Burgard, W. and Thrun, S. (1999). Monte carlo localization for mobile robots, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.
- Goel, P., Roumeliotis, S. I. and Sukhatme, G. S. (1999). Robust localization using relative and absolute position estimates, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 1134–1140.
- Hada, Y. and Takase, K. (2001). Multiple mobile robot navigation using the indoor global positioning system (igps), *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, Maui, Hawaii, USA, pp. 1005–1010.
- Hosoi, K. and Sugimoto, M. (2006). Shepherd: An interface for overcoming reference frame transformations in robot control, *Intern. Conf. on Robotics and Biomimetics*, Kunming, China.
- Intel (2007). OpenCV - Open Source Computer Vision Library, <http://www.intel.com/technology/computing/opencv/>. Acessado em 30/04/07.
- Kamphuis, A. and Overmars, M. (2004). Motion planning for coherent groups of entities, *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, New Orleans, Louisiana, pp. 3815–3821.
- Kloetzer, M. and Belta, C. (2006). Hierarchical abstractions for robotic swarms, *Proceedings of the 2006 International Conference on Robotics and Automation (ICRA)*, Orlando, FL, pp. 952–957.
- Mei, C., Benhimane, S., Malis, E. and Rives, P. (2006). Homography-based tracking for central catadioptric cameras, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*.
- Pimenta, L. C. A., Mendes, M. L., Mesquita, R. C. and Pereira, G. A. S. (2006). Fluids, particles, and multiple robots in electrostatic fields, *Proceedings of the IEEE Conference on Electromagnetic Field Computation (CEFC'06)*, Miami, FL, p. 309.
- Thrun, S., Burgard, W. and Fox, D. (1998). A probabilistic approach to concurrent mapping and localization for mobile robots, *Machine Learning and Autonomous Robots* **31**(5): 1–25.
- Wagner, D. and Schmalstieg, D. (2007). Ar-toolkitplus for pose tracking on mobile devices, *Computer Vision Winter Workshop*, St. Lambrecht, Austria.