

COMPORTAMENTOS REATIVOS PARA SEGUIR PISTAS EM UM ROBÔ MÓVEL GUIADO POR VISÃO

REINALDO A. C. BIANCHI, ALEXANDRA SILVA SIMÕES, ANNA HELENA REALI COSTA

Laboratório de Técnicas Inteligentes, Departamento de Engenharia de Computação e Sistemas Digitais,
Escola Politécnica da Universidade de São Paulo
Av. Prof. Luciano Gualberto, Travessa. 3, 158. CEP: 05508-900 São Paulo, SP, BRASIL
E-mails: {rbianchi, assimoes, anna}@pcs.usp.br

Resumo — Este artigo descreve o projeto e a implementação de um sistema de controle para um robô móvel autônomo dotado de percepção, capaz de seguir um caminho desenhado nas superfícies sobre as quais o robô se move e evitar colisões. Este sistema foi implementado seguindo um método guiado por experimentação, utilizando um mini-robô Khepera. Dois comportamentos reativos foram implementados — seguir uma linha (pista), baseado em informações visuais de desvio de obstáculos, baseado em formação de sensores infravermelho previamente definidos em hardware no robô. A arquitetura usada é composta por quatro módulos básicos: módulo de aquisição de imagens; módulo de detecção de cores baseado em limiares; módulo de identificação da posição, direção e trajetória do robô; e módulo de controle do mini-robô Khepera. Pode-se concluir que o sistema é eficiente, sendo capaz de seguir qualquer caminho desenhado por um usuário.

Abstract — This paper describes the design and the implementation of a control system for an autonomous mobile robot capable of following a path drawn in the surface on which the robot moves and to avoid collisions. This system was implemented using the mini-robot Khepera, employing an experimentation-guided method. In this paper, first the reactive behaviors implemented are represented — path following based on visual information and obstacles avoidance based on infrared sensors information, this last one previously defined in the robot. Next is described the architecture of the system, composed by four basic modules: image acquisition module; color detection module based on thresholds; robot's position, direction and trajectory identification module; and control module of the mini-robot Khepera. It can be concluded that the system is efficient, being capable to follow any path drawn by a user.

Keywords — AI Robotics; Autonomous Mobile Robots; Reactive Robots; Computer Vision.

1 Introdução

Como resultado da tendência atual das pesquisas na área de Inteligência Artificial, diversas contribuições integrando percepção, raciocínio e ação têm sido propostas para a solução de problemas variados, porém muitas vezes não cotidianos (Bianchi e Reali Costa, 2000; Reali Costa et al., 1999).

Um dos campos de aplicação onde está integrada tem sido a área da robótica, onde dispositivos de hardware especializados (sensores e atuadores), interpretação e fusão sensorial, redes neurais, computação evolutiva, visão computacional, planejamento inteligente, sistemas multi-agentes e teorias de controle são exemplos de campos que devem ser integrados.

Neste contexto, mini-robôs têm sido usados em pesquisas na última década por serem mais baratos, flexíveis e versáteis — além de ocuparem menos espaço — que os robôs convencionais. Dentre os mini-robôs, a plataforma Khepera (K-Team, 1999) foi uma das primeiras a surgir (em 1991) e é hoje usada mais a ceitas, tendo centenas de robôs espalhados pelo mundo e até congregados em workshops internacionais específicos (Löfler, 1999).

O objetivo deste trabalho é a descrição do projeto e a implementação de um sistema de controle para um mini-robô móvel autônomo dotado de percepção, capaz de rastrear pistas e se movimentar sobre um caminho não conhecido previamente. Este sistema foi implementado seguindo um método guiado por experimentação (Ribeiro et al., 2001), utilizando um robô Khepera.

Para atingir este objetivo foram implementados dois comportamentos reativos — seguir uma linha e desviar de obstáculos — com definição prévia de uma hierarquia de prioridades e uma abordagem competitiva entre eles. O sistema implementado possui os seguintes módulos básicos: módulo de aquisição de imagens; módulo de detecção de cores baseado em limiares (*thresholds*); módulo de identificação da posição, direção e trajetória do robô; e módulo de controle do robô Khepera. Estes módulos serão descritos neste trabalho.

O restante do trabalho está organizado da seguinte maneira: a Seção 2 descreve a tarefa a ser executada e os comportamentos implementados. A Seção 3 apresenta a plataforma Khepera e a arquitetura do sistema. A Seção 4 descreve o sistema visual do robô enquanto a Seção 5 descreve o sistema de controle do mesmo. Finalmente, a Seção 6 apresenta os resultados obtidos e a Seção 7, as conclusões deste trabalho.

2 Definição das tarefas realizadas

Segundo Ribeiro, Reali-Costa e Romero (2001), são três os métodos mais comuns para projetar robôs móveis inteligentes: os baseados em Etologia, que recorrem a modelos biológicos que, traduzidos e simplificados, são usados para construir entidades artificiais; os baseados em Atividade Simulada, que definem cada situação que o robô pode vir a encontrar e os comportamentos correspondentes, e os Guiados por Experimentação, onde um conjunto mínimo de comportamentos reativos são previamente definidos, de acordo com o domínio da(s) tarefa(s) que o

robô realizará, e, por meio de experimentos, os comportamentos são modificados ou novos comportamentos são adicionados de forma que a(s) tarefa(s) seja(m) executadas com maior grau de sucesso (Ribeiro et al., 2001) – este último método foi adotado neste trabalho.

Seguindo o paradigma reativo para robótica mais veloz e inteligente, foram definidos inicialmente dois comportamentos: desviar de obstáculos e seguir uma linha (pista).

A tarefa de desviar de obstáculos utiliza um programa em que o robô, quando encontra um obstáculo, muda de direção e segue uma trajetória semelhante à de um veículo de Braitenberg (Braitenberg, 1984). Este comportamento utiliza informações sensoriais fornecidas por oito sensores infravermelhos distribuídos no perímetro da base circular do robô.

O segundo comportamento exige que o robô se mova seguindo uma linha desenhada na superfície sobre a qual o robô se encontra. Este caminho não é pré-definido, podendo ser desenhado qualquer percurso para o robô executar. Este comportamento utiliza um sistema de visão global para encontrar a posição do robô e o caminho a seguir, e foi implementado em software em um microcomputador.

Na arquitetura implementada, de acordo com as informações sensoriais captadas, cada comportamento “indica” uma ação a ser efetuada pelo robô. Usou-se o mecanismo competitivo de coordenação dos comportamentos onde, a cada instante, apenas uma única ação pode ser escolhida. Os conflitos entre os comportamentos são resolvidos com base em uma hierarquia de prioridades previamente definida.

Pode-se mapear os estímulos sensoriais em ações de maneira discreta ou contínua (Ribeiro et al., 2001). A codificação discreta são aquelas que definem uma ação a realizar a partir de um certo estímulo, por exemplo, regras do tipo “se-então” usadas em sistemas de produção. Já as codificações contínuas definem funções que permitem a realização do mesmo mapeamento de maneira contínua, como por exemplo a técnica dos campos de potenciais. Neste trabalho foi realizado um mapeamento discreto para codificar os comportamentos.

3 A plataforma Khepera

Uma das plataformas robóticas que tem obtido maior aceitação junto à comunidade científica mundial é a família de robôs Khepera. Esta aceitação deve-se principalmente ao fato de seu tamanho reduzido quando em comparação com outras plataformas de sua fácil manipulação, instalação e configuração.

A principal característica do Khepera (Figura 1) é o seu tamanho reduzido (altura de 4 cm), com motores, sensores, baterias e processador incluídos.

Apesar de pequeno, o Khepera possui uma grande capacidade computacional embarcada, baseada em um sistema composto por um microcontrolador de 32

bits (MC68331 @ 16 MHz), memória, bus de extensão paralela, portas seriadas, portas de entrada analógicas, além de uma pequena rede de comunicação local que permite a comunicação com equipamentos adicionais.

ó-
m-
i-

o-
m-

i-

r-
i-
o-
n-

a-

e-
i-



Figura 1. O Robô móvel Khepera utilizado sobre um monitor (largura da etiqueta branca: 10 cm).

O mini-robô possui oito sensores de proximidade, que estão posicionados em volta do robô como mostrado na Figura 2. Estes sensores são compostos por um emissor e um receptor de luz infravermelha e permitem medir a proximidade de obstáculos através da luz que é refletida pelos objetos próximos. Esta medida, pelas características dos sensores, depende principalmente de dois fatores: a reflexividade do obstáculo e a iluminação ambiente. A faixa de operação média se encontra numa distância de 50 milímetros.

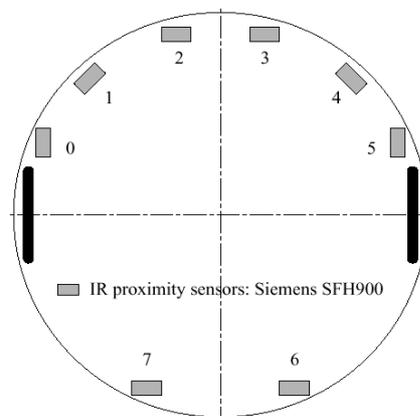


Figura 2. Posição dos sensores do Robô Khepera (K-Team, 1999)

Como atuadores o Khepera possui dois motores DC acoplados às rodas através de uma redução de 25:1. Um encoder localizado no eixo do motor permite medidas de odometria, gerando 600 pulsos por revolução da roda. Os motores são controlados diretamente pelo processador utilizando o método PWM. As ações de controle do motor de aquisição de dados dos sensores são descritas na Seção 5.

O Khepera é controlável através de softwares como o LabView e Matlab ou diretamente através de palavras de controle enviadas através de uma porta de comunicação serial, o que pode ser implementado

n-
ó-

a-

a-
e-

o-
a-

a-

í-

n-

apartir de qualquer computador sem necessidade de aquisição de software adicional.

A arquitetura do sistema implementado é composta pelos seguintes módulos básicos:

- Módulo de aquisição e visualização de imagens;
 - Módulo de detecção de cores utilizando limites;
 - Módulo de identificação de posição e direção de trajetória do robô;
 - Módulo de controle do robô Khepera;
- Os três primeiros módulos, que processam a formação visual, são descritos a seguir.

4 Processamento da informação visual

A visão computacional tem por objetivo a extração de parâmetros que permitam associar regiões da imagem com objetos da cena (Ballard e Brown, 1982). Assim, busca-se particionar a imagem digital em regiões disjuntas utilizando-se como parâmetros atributos da imagem.

Um dos atributos importantes em uma imagem é o acordo de seus pixels. Usou-se para a tarefa de segmentação de imagens, a técnica de imposição de limites para definir faixas de interesse do atributo cor na imagem.

O processamento visual do robô pode ser separado em três etapas distintas: *i*) aquisição e visualização de imagens; *ii*) identificação de cores e *iii*) identificação dos objetos de interesse. A seguir, cada uma desses módulos é detalhada.

4.1 Aquisição e visualização de imagens

Para realizar a tarefa de aquisição de imagens foi instalada uma câmera de vídeo colorida sobre a área de movimentação do Khepera (visão global). Uma placa de aquisição de imagens genérica contendo o C.I. BT-848 (que é suportada pelo sistema Linux via V4L - *video for linux device drivers*) foi utilizada para promover a interface entre a câmera e o computador.

Objetivando a visualização das imagens sem tempo real, desenvolveu-se uma interface visual em Xlib (Nyé, 1990). Tal interface possibilitou ainda a seleção dos níveis RGB de pixels correspondentes aos objetos de interesse na imagem; na fase de calibração, os valores dos limites das cores de interesse são estabelecidos.

4.2 Detecção de cores baseada em limites

Embora aparentemente imediato, o conceito que cerca a palavra cores contém muitas noções de ordem física e psicológica, o que torna a cor um conceito bastante difícil de modelar. No entanto, a identificação de cores em uma imagem pode ser abordada como um problema de separação de grupos de cores em um espaço n-dimensional, dado pela transformação da imagem de interesse para um espaço apropriado de representação de cores. Considere a Figura 3 como exemplo. Trata-se de uma vista superior de um robô

Khepera utilizando formas geométricas coloridas (rosa, preto e amarelo) para facilitar sua localização pelo sistema de visão. A transformação desta imagem para o espaço de cores RGB pode ser vista na Figura 4.

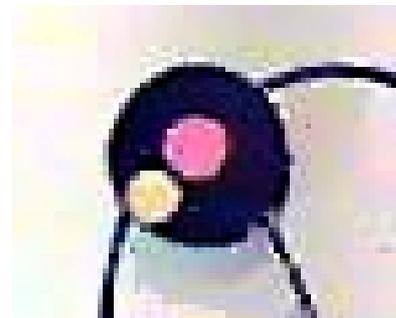


Figura 3. Vista superior de um robô Khepera.

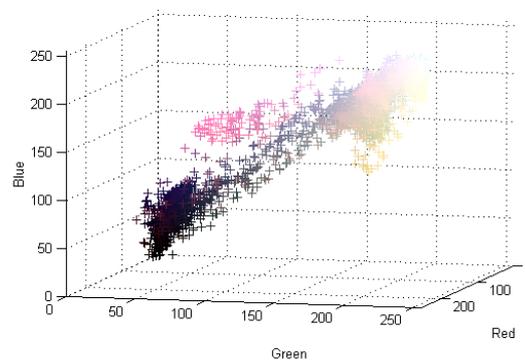


Figura 4. Figura 3 no espaço de cores RGB.

A abordagem conhecida como técnica de imposição de limites é uma das abordagens para a partição do espaço de cores. Nesta abordagem, o conteúdo de cada pixel é considerado, ignorando-se totalmente sua posição espacial e conectividade. A abordagem de imposição de limites visa agrupar os sólidos formados por cada região no espaço de cores utilizando paralelepípedos (considerando-se um sistema matricial de representação). Nesse sentido, é necessário e indispensável a um sistema baseado na técnica de fornecimento (direto ou indireto) dos limites do paralelepípedo que delimita cada cor. Para o trabalho proposto, exemplos de cada cor foram fornecidos através de amostras do ambiente (do próprio robô) no ambiente gráfico. De posse de exemplos das várias cores que integram o ambiente, os limites mais adequados para a classificação foram determinados pelo sistema. Umajuste mais refinado pode ser realizado com interação com o usuário. A Figura 5 ilustra possíveis valores de limites para a separação das cores rosa e amarelo.

Uma vez identificadas as cores de interesse, inicia-se o processo de identificação dos objetos de interesse na imagem (robô e pista a seguir).

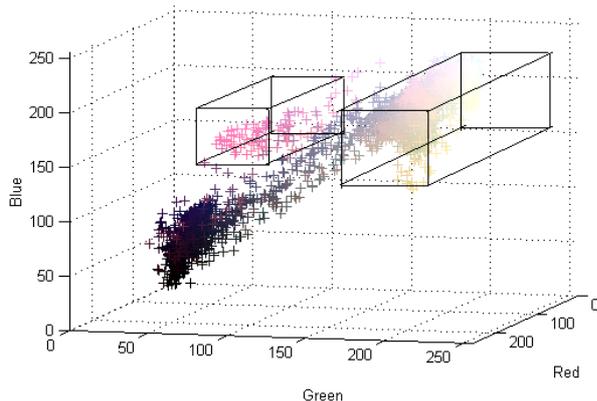


Figura 5. Exemplos de limites para as cores rosa e amarelo.

4.3 A identificação da posição e direção do robô e caminho a seguir.

Visando identificar os componentes da cena através de um processo simples de visão computacional, os objetos importantes foram marcados com diferentes cores. Sobre o robô da família Khepera foi inserido um rótulo contendo dois círculos de diferentes cores: um no centro geométrico do robô (circular) e outro indicando o sentido da "frente" do robô. Assim sendo, o processo de determinação da direção de movimentação do robô ficou restrito ao processo de determinação do centro das duas áreas circulares de cores pré-definidas. Estas áreas foram determinadas, após identificação de cores dos pixels, por um processo de aglutinação de pixels vizinhos conhecido por *crescimento da região* (Gonzales, 1997). Uma vez determinados os centros dos círculos, é imediata a determinação do vetor que passa por seus centros (vetor de direção do robô).

Para a determinação da direção do caminho a ser seguido, observou-se a interseção de um quadrado centrado no centro geométrico do robô e a corda tangente à demarcação do caminho (azul). O quadrado intercepta o caminho em duas posições distintas: uma por onde o robô já passou (mais distante da frente do robô) e uma mais próxima, à frente deste. O vetor da direção desejada é o vetor que une o centro geométrico do robô ao ponto determinado pela interseção desejada, foi possível determinar o deslocamento desejado para o robô. A Figura 6 ilustra o funcionamento do sistema.

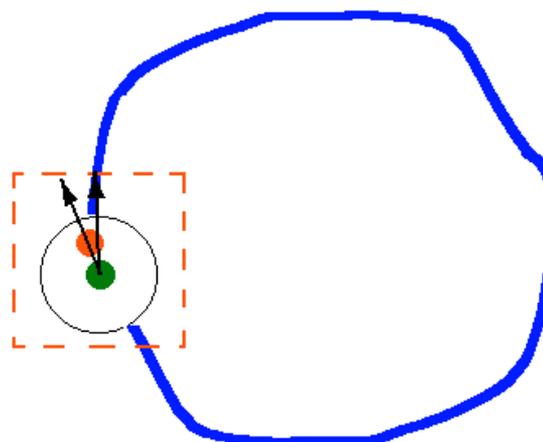


Figura 6. Representação da perseguição do caminho pelo robô.

5 O Controle do robô

O controle da movimentação do Khepera é a aquisição dos dados dos sensores embarcados e realizado através de uma porta de comunicação serial, utilizando um protocolo pré-definido. Este protocolo permite o controle de todas as funções do robô, sendo constituído por comandos e respostas, ambos em código ASCII.

Entre os diversos comandos, que incluem o controle de velocidade, aceleração, posição, etc., os mais importantes para estes trabalhos são:

- Define posição (formato: C, pos_left, pos_right): indica a posição absoluta a ser alcançada pelas rodas.
- Lê Posição (H): Lê o contador de 32 bits de cada motor. A unidade de retorno é a que vale a um pulso do encoder, que corresponde a 0,08mm.
- Define velocidade ($D, speed_motor_left, speed_motor_right$): define a velocidade dos motores, usando como unidade o pulso/10 ms., que corresponde a 8mm por segundo. A velocidade máxima é de 127/10ms, que corresponde a 1m/s.
- Lê Velocidade (E): lê a velocidade instantânea de cada motor (retorna pulsos/10ms.).
- Lê sensores de proximidade (N): retorna valores de 10 bits dos oito sensores de proximidade (iniciando no sensor frontal mais à esquerda e continuando no sentido horário até os sensores traseiros).

Para enviar o comando é usada uma porta de comunicação serial padrão RS-232, com velocidade de 9600 Baud, 8 bits de dados, 1 start bit, 2 stop bits, sempre em modo de transmissão.

6 Implementação e Resultados

O sistema aqui apresentado foi implementado utilizando um computador padrão IBM PC Pentium II 266 MHz equipado com o sistema operacional Linux Red Hat 5.2. A programação de todos os sistemas de

são computacional quanto do controle do robô foi realizada em Linguagem C/C++ e os arquivos executáveis foram gerados utilizando o compilador GNU C++.

A Figura 7 apresenta a execução do sistema, como o robô seguindo um caminho desenhado.

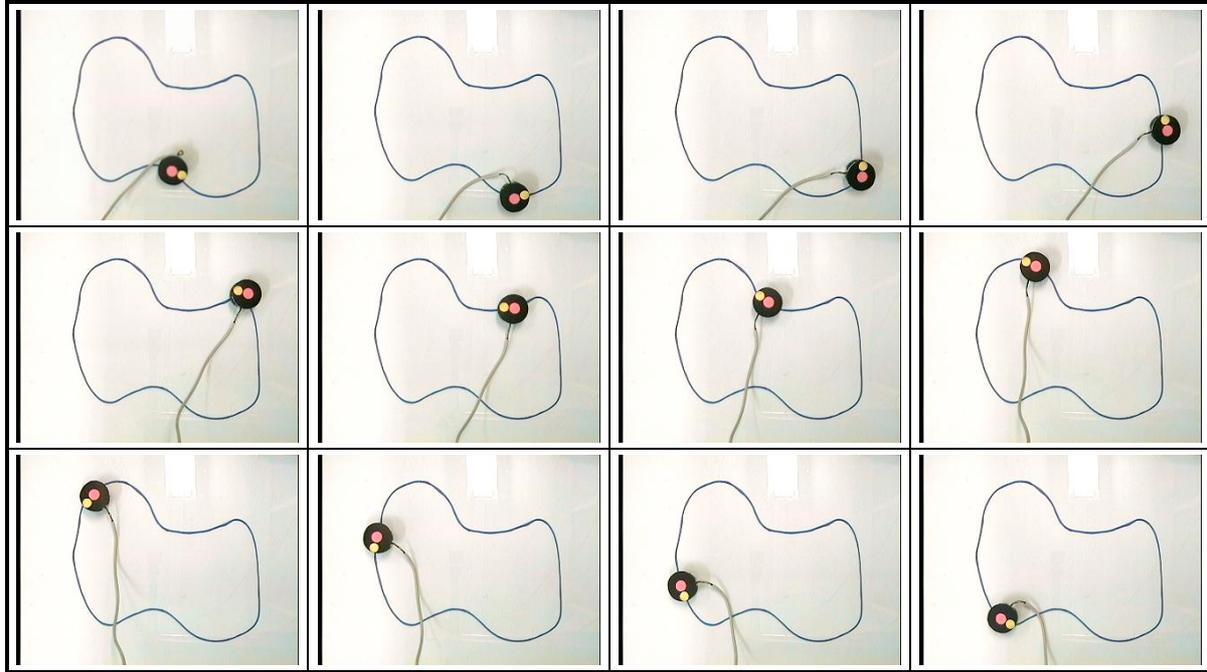


Figura 7. Robô Khepera seguindo um caminho.

7 Conclusão

O trabalho desenvolvido teve como resultado um sistema robótico bastante eficaz capaz de seguir qualquer trajetória desenhada por um usuário em um papel (ou quadro branco).

A plataforma Khepera é bastante estável e recomendável para o desenvolvimento de pesquisas na área robótica, dada sua característica já destacadas.

Com relação ao sistema de visão, embora a abordagem utilizada tenha mostrado bastante satisfatória para o trabalho em questão, ela não é recomendada para o menos dois casos: *i*) quando o ambiente de trabalho não mostrar “bem comportado” (visto que nem sempre os sólidos de cores podem ser separados por paralelepípedos) e *ii*) quando não existir homogeneidade na distribuição da iluminação do ambiente (visto que os limites de cores podem apresentar variações no espaço ou no tempo).

Algumas soluções estão sendo propostas no sentido de conferir maior robustez aos sistemas de visão baseados na classificação de cores, tais como a adoção de limites adaptativos (Bianchi Reali-Costa, 2000) - onde as bordas separadoras no espaço de cores não serão restringidas para os paralelepípedos estáticos - ou a classificação de cores utilizando redes neurais artificiais (Simões e Reali-Costa, 2000), onde as bordas se

param das regiões de interesse no espaço de cores podem assumir formas irregulares.

Nesse sentido, trabalhos futuros incluem a implementação de diferentes métodos de classificação de cores e sua comparação com o método imposto de limites, implementado no presente trabalho. Com a implementação de tais metodologias de classificação de cores, espera-se obter um sistema mais robusto e de fato aplicável à automação baseada em visão.

Finalmente, diversos vídeos demonstrativos deste trabalho podem ser encontrados na página do projeto: www.lti.pcs.usp.br/robotics/khepera.

Agradecimentos

Alexandra da Silva Simões recebeu bolsa de mestrado CAPES, através da Escola Politécnica da Universidade de São Paulo, de 1998 a 2000.

Este projeto foi parcialmente financiado pelo FINEP, projeto RECOPE-MANET, Proc. n. 77.97.0937.00, e pelo NSF/CNPQ ProTeM-CC projeto MAPPEL, Proc. n. 01.680033/99-8.

Os pesquisadores do LTI gostariam de agradecer à SOFTECHNOLtda. (www.softtechno.com.br) que gentilmente emprestou à Escola Politécnica o robô Khepera utilizado.

Referências Bibliográficas

- Ballard, D. e Brown, C. (1982). *Computer Vision*. Englewood Cliffs, Prentice-Hall.
- Bianchi, R. A. C.; Reali-Costa, A. H. (2000) OSistema de Visão Computacional do time FUTEPOLI de Futebol de Robôs. In: *Congresso Brasileiro de Automática*, 13^o, Florianópolis, 2000. **Anais**. Florianópolis, UFSC/Sociedade Brasileira de Automática, 2000. P. 2156-2162.
- Braitenberg, V. (1984) *Vehicles: Experiments in Synthetic Psychology*. Cambridge, MIT Press.
- Foley, J. D.; Van Dam, A.; Feiner, S. K.; Hughes, J. F. (1990) *Computer Graphics: Principles and Practice*. Addison-Wesley, 2nd edition, Cornell University.
- Gonzales, R. C.; Woods, R., E. (1992) *Digital Image Processing*. 2nd edition. Massachusetts, Addison-Wesley.
- K-Team. (1999) *Khepera User Manual*. Version 5.02. Lausanne. (disponível em www.k-team.com/robots/khepera/index.html)
- Löffler, A.; Mondada, F.; Rückert, U. (eds.) (1999) *Experiments with the Mini-Robot Khepera: Proceedings of the 1st International Khepera Workshop*. HNI-Verlagsschriftenreihe, Band 64, Paderborn.
- Nye, A. (1990) *Xlib Programming Manual*. O'Reilly and associates, Inc.
- Reali-Costa, A. H.; Pegoraro, R.; Stolfi, G.; Sichman, J.; Pait, F. M.; Ferasoli, H. (1999) Guaraná robot-soccer team: some architectural issues. In *Anais do 4^o Simpósio Brasileiro de Automação Inteligente*, São Paulo.
- Ribeiro, C. H. C.; Reali-Costa, A. H.; Romero, R. A. F. (2001) *Robôs Móveis Inteligentes: Princípios e Técnicas*. Jornada de Atualização em Inteligência Artificial do Congresso da Sociedade Brasileira de Computação.
- Simões, A. S.; Reali-Costa, A. H. (2000) Segmentação de imagens por classificação de cores: uma abordagem neural para a representação RGB. In: *Workshop de Computação WORKCOMP'2000*, Carlos H. C. Ribeiro e Milton T. S. Sakude (eds.), Instituto Tecnológico de Aeronáutica-ITA, São José dos Campos, SP, Outubro de 2000, Pg. 25-31.