



DESENVOLVIMENTO DE UM ROBÔ MÓVEL COM VISÃO PARA PROPÓSITOS EDUCACIONAIS

Eduardo B. Viécili – viecili@ieee.org

Marlon F. Alcantra – marlonmfa@gmail.com

André Romão – andre.gabiruh@gmail.com

Marcelo S. Hounsell – marcelo@joinville.udesc.br

Maurício A. Pillon – mpillon@joinville.udesc.br

Milton R. Heinen – miltonh@joinville.udesc.br

Universidade do Estado de Santa Catarina (UDESC), Dep. de Ciência da Computação

Rua Paulo Malschitzki, s/numero - Campus Universitário Prof. Avelino Marcante

89219-710 – Joinville - SC

Resumo: Desenvolver robôs móveis é um desafio estimulante para os estudantes. É possível encontrar várias soluções e kits de robótica móvel, mas não é muito comum encontrar o recurso da visão computacional nestes kits. A visão computacional pode substituir vários sensores de uma vez (e.g.: proximidade, posição e orientação) e com vantagens, pois permite identificar distâncias e ângulos em relação aos objetos, além de ser capaz de identificar o que é o objeto e suas dimensões. Neste artigo é apresentado o desenvolvimento de uma plataforma para o ensino de engenharia e robótica. Esta plataforma, nomeada Kbo, é composta por um robô modular simples de arquitetura aberta, com o acionamento das rodas em malha aberta, no qual se faz uso da visão computacional para permitir a exploração de um ambiente desconhecido. Para este robô foram desenvolvidas estratégias de movimentação na execução de tarefas específicas, as quais foram satisfatoriamente concluídas devido a flexibilidade no desenvolvimento que advém do uso da biblioteca QRPO (Quick Response Barcode with Position and Orientation). Esta biblioteca foi implementada sobre a estrutura do robô, juntamente com um software de apoio, o que permite ao estudante experimentar várias estratégias no uso da visão computacional. A plataforma robótica desenvolvida e o QRPO são ferramentas para o estudo dos robôs móveis, pois permitem que os estudantes possam praticar diversos conceitos de engenharia ligados a visão computacional e robótica.

Palavras-chave: Ensino, Robótica móvel, Visão computacional, QRPO, QR code.

Realização:

 **ABENGE**

Organização:



**O ENGENHEIRO
PROFESSOR E O
DESAFIO DE EDUCAR**



1. INTRODUÇÃO

Para que um aluno possa efetivamente exercitar sua análise e tomada de decisão envolvendo situações com critérios de custo, facilidade, flexibilidade, qualidade, dentre outros, se fazem necessárias no processo de ensino-aprendizagem experiências no domínio do concreto. Estas experiências concretas representam formas de poder confirmar e verificar uma abordagem, de ter uma introspecção através do conhecimento da forma física, e a possibilidade de testar materiais e processos na origem do projeto (VIECILI, 2011). Portanto, estas experiências são enriquecidas quando se constroem protótipos, que oferecem um potencial de surpreender e gerar resultados, pois trata-se de possibilitar os estudos práticos.

Existem vários kits para montagem de robôs móveis, modulares e fáceis de utilizar. Entretanto, um kit que disponibilize uma câmera de vídeo como dispositivo sensor não é comum. Isto vai exigir um módulo software de Visão Computacional (VC) que possa processar a imagem da cena e ser usado pelo algoritmo de decisão de movimentação (DAVIES, 2005).

No presente caso, os estudos estão relacionados a VC, métodos de controle e atuação, arquiteturas de robôs móveis, programação de baixo nível (microcontroladores) e de alto nível (estratégias para solução de tarefas), dentre outros. Portanto, o presente texto apresenta um robô móvel com comunicação sem fio, com uma câmera como sensor, um módulo de software de VC baseado em código de barras e uma interface para exploração e desenvolvimento. O foco deste trabalho está não somente na construção do robô mas também no aprendizado de uso da VC como recursos de planejamento de trajetória para robôs móveis. É importante ressaltar que o presente trabalho é uma ferramenta muito importante no ensino de engenharia, pois permite que o aluno vivencie na prática conceitos de diversas áreas, como por exemplo engenharia elétrica, engenharia mecânica e ciência da computação.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta alguns robôs móveis existentes que têm propósitos didáticos e suas limitações; a Seção 3 apresenta o robô Kbo em todos os seus detalhes de projeto e construtivos; a Seção 4 apresenta os recursos do robô implementado, tanto em hardware quanto em software; a Seção 5 apresenta como o Kbo pôde ser usado para desenvolver a solução para um problema existente; a Seção 6 apresenta as conclusões deste texto.

2. ROBÔS MÓVEIS ASSEMELHADOS

Existem diversas arquiteturas voltadas para a robótica móvel que servem como plataformas de estudo para VC, desenvolvimento da criatividade e/ou aprendizagem de programação. Para comparação apresentam-se alguns robôs existentes e suas características, dentre as quais cabe ressaltar que para fins didáticos de estudantes de engenharia e computação, visando oportunidades de aprendizado, flexibilidade, criatividade e custo, é recomendável desenvolver sua própria plataforma.

LEGO MindStorms (LEGOGROUPS, 2009) é um brinquedo educacional comercial que permite criar suas próprias invenções em robótica móvel usando as peças LEGO, que incluem sensores e atuadores (BAGNALL, 2007). A arquitetura de hardware é proprietária, mas a de software tem versões abertas disponíveis na internet.

Khepera (K-TEAM, 2012) é um robô móvel de pequenas dimensões. Sua arquitetura padrão possui oito sensores de proximidade em volta do robô. Para desenvolvimento e testes o Khepera é controlável através de softwares como o LabView e o MatLab, ou diretamente através de uma porta de comunicação serial que pode ser facilmente implementada



(BIANCHI et al., 2001). O Khepera é um robô comercial, vendido em módulos, com arquitetura de software aberta.

O trabalho de (PROBST et al, 2010) fez o desenvolvimento de uma plataforma de hardware para um robô móvel autônomo com tecnologia aberta. O trabalho leva em consideração os aspectos eletroeletrônicos, mecânicos e de software, e dá ênfase aos softwares relativos à navegação, locomoção, sistemas de VC e interface homem-máquina. O projeto denominado OpenVisionBot (OVB) tem por objetivo distribuir a tecnologia para pesquisa em ambientes acadêmicos ou de negócio. Entretanto o OVB ainda está em fase de desenvolvimento, e teria que ser totalmente reconstruído (peças com formatos e dimensões específicas, não padronizadas) pois não existem módulos comerciais a venda.

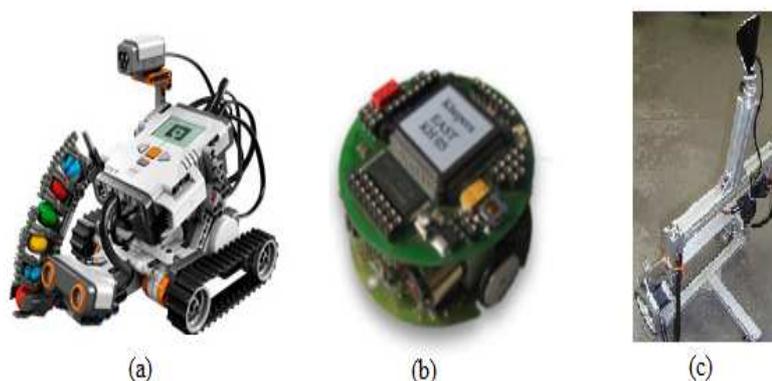


Figura 1. Robôs móveis assemelhados: a) Lego MindStorms; b) Khepera; c) OVB.

A Tabela 1 exibe as características das arquiteturas analisadas. O custo foi informado como o valor médio aproximado encontrado entre os revendedores no mês de março de 2012, onde é possível observar que o custo das tecnologias Khepera e Lego MindStorms supera o valor de R\$ 1900,00 e possuem elementos que não são necessários para os testes pretendidos com a aplicação, como o grande número de sensores do LEGO e Khepera, e possibilidade de diversos atuadores do kit LEGO MindStorms.

Tabela 1. Comparativo entre plataformas. Fonte: (ALCANTARA, 2012)

	Legó	Khepera	OVB	Kbo
Custo	R\$ 1950	R\$ 2300	Não consta	R\$ 600,00
Estrutura	Flexível	Fixa	Fixa	Flexível
Comunicação	Bluetooth ou USB	módulo RF	USB	Wi-fi, Zigbee
Sensores Quantidade	5	8	2	1
Sensores tipo	Diversos (kit)	Infravermelho	Toque e câmera	Câmera IP
Bateria/Fonte	3.7V	7.4V	9V e 24V	12V



O trabalho de (PROBST et al., 2010) apesar de não definir um valor para a arquitetura construída, fornece informação suficiente para replicar a montagem mas, a falta de módulos facilmente extensíveis ou acopláveis, limita sua utilização.

Devido ao custo, incompatibilidade de tamanhos, dificuldade de montagem com peças convencionais, decidiu-se montar um robô móvel baseado na arquitetura Arduino. Esta solução é composta por um hardware aberto (microprocessador aberto, programável em C e C++), onde se encontram vários fornecedores tanto de peças, quanto de módulos construtivos o que permite estender e alterar a estrutura do robô, conforme a necessidade. Ainda, um dos principais fatores, o custo da montagem resulta num valor abaixo das demais soluções mencionadas acima, conforme pode ser observado na Tabela 1. Além disso, a montagem do robô é muito importante do ponto de vista educacional, pois propicia que o estudante tenha uma interação prática construtiva.

3. DESENVOLVIMENTO

O desenvolvimento foi feito em blocos modulares (Figura 2), permitindo uma flexibilidade em alterações e melhoramentos. Assim cinco estruturas independentes foram definidas: hardware do robô, comunicação, visão computacional (VC), estratégias e interface. Isto permite que cada um dos blocos possa ser estudado e alterado em separado. Por exemplo:

- No hardware do robô, realiza-se pesquisas envolvendo projetos mecânicos, engenharia de materiais, estudos de cinemática, controle e atuação de motores, fontes de energia, eletrônica de potência, e demais áreas;
- Na comunicação, sistemas de envio e recebimento de informações, técnicas de rádio frequência, protocolos de comunicação, e relacionados;
- Na VC, se faz uso do processamento morfológico, reconhecimento de padrões, ambientes com realidade aumentada, e leituras de informações;
- No bloco de estratégias, busca-se técnicas de inteligência, rotinas, lógica, mapeamento e ambientação;
- Por fim na interface temos a engenharia de software, projeto de algoritmos e técnicas de programação, gerenciamento, etc.

Visto que as características envolvidas na finalidade do projeto para a estrutura modular exigem que se tenha um robô autônomo com VC, optou-se por realizar o processamento externo ao robô dada a demanda computacional, principalmente no processamento das imagens da câmera. Isto significa que essa separação do hardware e controle exigiu o desenvolvimento de uma comunicação, tanto para informar a imagem da câmera quanto para controlar a movimentação do robô, onde o interpretador de imagem recebe e faz o processamento gráfico, gerando as informações então aproveitadas pelo bloco de estratégias para tomar decisões, as quais por fim são enviadas ao robô.

O robô foi denominado “Kbo” por possuir só um olho, da mesma forma como os seres mitológicos gregos ciclopes. Mas, por ser uma entidade em desenvolvimento (como uma criança) encontrou-se o nome Cable, não na mitologia grega e sim no universo Marvel¹, que é o nome do filho de Ciclope. Tendo definido como Cable o nome, realizou-se apenas uma abreviação para as letras Kbo.

1 <http://marvel.com/>



3.1 Hardware do robô

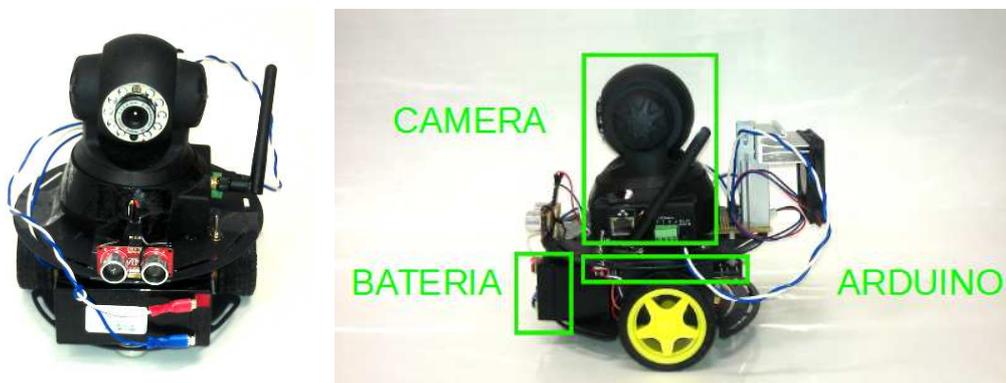


Figura 2. Robô móvel Kbo

O Kbo possui duas rodas controladas por motores independentes, um terceiro ponto de apoio no chão para equilíbrio do robô, um sensor ultra-sônico (como segurança em caso de falha acidental), uma câmera IP, um controlador baseado em Arduino², um módulo *Xbee*, um *shield Xbee* e uma bateria de 12 volts, que alimenta o controlador, os motores das rodas e a câmera IP. Além do robô, há um computador remoto, conectado à um Arduino com um módulo *Xbee* e um *shield Xbee*.

Uma das limitações do robô é que o controle de acionamento das rodas em malha aberta compromete a capacidade de se prever a posição do Kbo após uma movimentação. Esta limitação pode ser melhorada com o uso de motores de passo, mas ela não compromete a aplicação em si. Na verdade ela acaba por aumentar a importância do sistema de VC e as funcionalidades do QRPO, pois os algoritmos de controle precisam compensar as limitações do robô para que as tarefas sejam realizadas satisfatoriamente, como será visto mais adiante.

O controlador foi desenvolvido para projetos de robótica e possui um microcontrolador, que comporta o armazenamento de um programa que pode receber informações dos sensores conectados à ele, e do computador. Com base nessas informações, o controlador é responsável por acionar duas saídas digitais, ligadas aos motores das rodas do robô, tornando possível que o robô movimente-se para frente, para trás, para esquerda ou para direita.

A utilização da câmera IP minimiza a necessidade do uso de diversos sensores para evitar o mal reconhecimento do ambiente. Sensores *laser*, por exemplo, não reconhecem vidros à sua frente, e sonares não reconhecem objetos com ângulo agudo à sua frente (SIEGWART; NOURBAKSH, 2004). Além disso a câmera pode se movimentar, o que ajuda o robô a explorar a cena de forma ativa, embora esse recurso não tenha sido utilizado nos experimentos descritos neste artigo (Seção 4). Assim a câmera envia as imagens obtidas do ambiente para o computador remoto, ficando a cargo deste fazer uso de algoritmos para a detecção de objetos nas imagens, detecção de marcadores padronizados, interpretação de *QR Code*³, além de permitir o simples monitoramento e exploração do ambiente.

2 Arduino é uma plataforma de hardware de código aberto.

3 QR Code (*Quick Response Code*) é um código de barras bidimensional, suas especificações estão detalhadas na norma ISO/IEC 18004:2006.



A adição do sensor ultrassônico à frente do robô não afeta o processamento dos algoritmos de controle. Porém, caso esses algoritmos venham a apresentar falhas e enviem informações que façam o robô colidir frontalmente, o sensor ultrassônico fará leituras da distância do robô ao objeto mais próximo. Caso essa distância seja inferior à quinze centímetros, o robô irá recuar automaticamente. Vale ressaltar que com esta estrutura simples de acionamento das rodas, o Kbo é um robô impreciso quanto a sua movimentação e portanto precisa da VC para a condução de suas ações, pois ele não sabe onde está em momento algum se for considerar apenas as instruções de movimentação (ele não possui informações de odometria confiáveis).

3.2 QRPO: SUPORTE DE VISÃO COMPUTACIONAL

Para agregar a Visão Computacional (VC) de reconhecimento de objetos ao robô fez-se uso do QRPO (*Quick Response Barcode with Position and Orientation*), que é uma biblioteca que permite ler um QR Code e ainda identificar posição e orientação. Tem como funcionalidades integradas (ALCANTARA et al., 2011): a capacidade de armazenamento de dados dos códigos de barras no padrão QR Code e a possibilidade de verificar a posição e orientação da câmera em relação ao marcador, podendo obter três parâmetros de saída: a posição relativa; a orientação; e os dados contidos no código de barras.

Entende-se como possíveis aplicações para o QRPO qualquer aplicação que necessite de um meio de adquirir posição/orientação e dados e que dependa destas informações, juntas ou separadamente, para um determinado propósito. O QRPO foi desenvolvido para receber como parâmetros as especificações da câmera IP e os *frames* obtidos por ela, e retornar como saída: a existência ou não de um QRPO na cena; a posição e orientação caso exista o marcador; e dados, caso seja possível a decodificação. Dessa forma, fica a cargo do programa a ser desenvolvido pelo estudante fazer uso destas informações.

O QRPO também pode ser utilizado para a criação de mapas em tempo real do ambiente ao qual está inserido. Para isso se faz necessário que cada obstáculo do ambiente contenha um QRPO de forma a informar sua posição e dimensões ao robô. É importante ressaltar que o QRPO consegue identificar posição e orientação (PO) a uma distância muito maior que aquela requerida para identificar os dados (do QR Code). Isto significa que ele primeiro identifica onde está um marcador mas precisa chegar perto para saber o que é.

3.3 Comunicação

Segundo (LABIOD et al, 2007), a área de alcance das classes Bluetooth⁴ está diretamente ligada ao gasto de energia dos dispositivos que utilizam esta tecnologia. Isto reduz consideravelmente o tempo de autonomia da bateria de robôs móveis. Desta forma, a utilização da tecnologia Zigbee mostra-se mais eficiente para o caso, pois há baixo consumo de energia (ALLIANCE, 2010) e não há necessidade de troca de grandes taxas de dados entre os controladores Arduino. Portanto a comunicação entre o computador e o robô é feita através de módulos *Zigbee*, que permitem que o Arduino receba informações.

A distribuição do sistema parte do princípio de que a plataforma escolhida do robô móvel não suporta o processamento de imagem necessário para a identificação da área do código, bem como a leitura da informação. Este esquema está ilustrado na Figura 3, o computador, conectado a mesma rede da câmera IP, recupera continuamente, através de servidor web

4 Bluetooth é uma especificação para comunicação sem fio, mais informações em www.bluetooth.com



localizado na câmera, as imagens captadas pela câmera IP. Uma ferramenta baseada na biblioteca QRPO analisa as imagens recebidas e, caso identifique uma informação em um QR Code válido na cena, envia um comando de movimentação a base do robô, via *zigbee*. Os QRPO espalhados pelo ambiente possuem informações de posição e orientação que permite ao computador definir o comando de movimentação adequado ao robô a partir do ponto atual.



Figura 3: Esquema de Comunicação (ALCANTARA, 2012)

O tratamento das imagens e tomadas de decisão é processado pelo computador, que apenas informa ao programa embarcado do robô os comandos básicos de movimentação. Após receber os comandos, este programa embarcado passa a executar a operação através dos motores das rodas. Com o movimento das rodas, as imagens captadas pela câmera IP serão modificadas, tendo em vista que a câmera encontra-se sobre o robô, dessa forma realimentando o sistema e caracterizando um sistema de controle de malha fechada, onde a VC faz o fechamento da malha.

3.4 Estratégia de Programação

A ferramenta de controle do robô foi modelada orientada a objetos, usufruindo-se, portanto, da possibilidade de herança de classes, o que permite o uso desta mesma ferramenta para outros robôs. O usuário pode reescrever, isoladamente, as estratégias de movimentação, tratamento da imagem, sem precisar alterar a interface gráfica ou o módulo de controle. Com isso, usuários podem usar o KBO para resolver diversos problemas modificando somente o módulo que o interessa. O estudo de caso deste trata o problema Rato-Queijo.

3.5 Interface gráfica

O software foi desenvolvido em C++ para Linux, e tem por função realizar o gerenciamento da plataforma do Kbo e de seus módulos. Como objetivo dependendo da aplicação: por exemplo, a interface permite configurar o modo de operação do robô (autônomo ou não); se o robô estiver em um local não visível pelo sistema; também a interface permite ao usuário do sistema saber o que o robô está vendo; a interface ainda permite que resultados obtidos pelo robô, parciais ou finais, sejam observados e salvos.

A Figura 4 ilustra esta interface gráfica, na qual pode se visualizar a imagem interpretada (círculo contendo o nº 1 na Figura 4), o mapeamento mais tempo decorrido (ver círculo nº 2), as informações extraídas de posição, orientação e dados (ver círculo nº 3), operação remota (ver círculo nº 4) e descrição das ações tomadas (ver círculo nº 5).



Figura 4. Interface gráfica da interface feita em C++.

4. ESTUDO DE CASO: TAREFA RATO-QUEIJO

A aplicação Rato-Queijo se caracteriza pela busca do Rato, no caso o Kbo, a um queijo, no caso um objeto com QRPO específico no ambiente de estudo. O Rato movimentar-se na cena segundo informações que ele vem recebendo ao longo do caminho, QRPO de dicas espalhados pela cena. Em alguns caso o Rato pode traçar um caminho direto ao queijo, ou seja, sem obstáculos e utiliza-se das dicas (QRPO) somente para corrigir seus erros de rotas, e em outros, ele deve atingir pontos intermediários e contornar obstáculos.

Neste estudo de caso, o estudante deve implementar uma aplicação com estratégia de movimentação buscando encontrar o menor caminho ou o mais rápido. Com isso, espera-se que o estudante desenvolva aptidões da robótica móvel vinculadas ao posicionamento do robô, estratégia de movimentação e tratamento de obstáculos em uma plataforma acessível e totalmente controlável. Na inicialização do robô, ele recebe as coordenadas do queijo no plano, porém ele não possui a informação da sua posição ou dos demais objetos neste plano.

Para resolver o problema da movimentação, o aluno deverá resolver o problema trigonométrico de posicionamento do robô no plano com base na posição obtida pela VC. Ele deve ainda definir formas de transpor os obstáculos, tomando decisões envolvendo o gerenciamento de riscos, otimização do caminho e do tempo gasto. O robô utiliza-se da orientação visual através de marcadores de VC, portanto, caso ele os perca ao movimentar-se, ele também perde sua orientação no plano. Outras formas complementares de posicionamento podem ser acrescentadas a este sistema, por exemplo, o uso da triangulação das antenas *zigbee* para indicação da localização do KBO no plano (WRUCK, 2011).

Para a tarefa do *Rato-Queijo* foram dispostos obstáculos prismáticos (retangulares) no ambiente, que possuem QRPOs agregados a cada uma das quatro faces perpendiculares ao plano. O QRPO contém a informação da posição do obstáculo no plano, e para qual ponto cardeal está voltada a face que está sendo visualizada. Também é informado no próprio QRPO as dimensões (largura e comprimento) da caixa para que o *rato* possa utilizar a informação para desviar do obstáculo caso seja necessário.

A disposição das informações segue a Figura 5, e toda a formatação, como por exemplo abertura de colchetes e vírgulas, devem ser respeitados. O item 'P' da Figura 5 será



substituído devidamente pela inicial de um dos pontos cardeais em inglês ('N' - North, 'S' - South, 'W' - West, 'E' - East).



Figura 5. Padrão da informação interna do QRPO na aplicação (ALCANTARA, 2012)

O queijo, por ser o objeto alvo, possui uma informação diferenciada no QRPO. Ele não guarda as suas coordenadas, já que estas são dados de entrada da aplicação, nem suas dimensões, haja visto que para encontrá-lo não é necessário saber suas dimensões. Logo, o objeto contém apenas a informação de que ele mesmo é o “queijo”, que ao ser lida faz o robô posicionar-se em frente a ele e então o programa encerra.

A Figura 6 mostra uma imagem da configuração de uma aplicação rato-queijo contendo dois objetos que representam podem virar obstáculos, dependendo da posição arbitrária em que é deixado o robô.

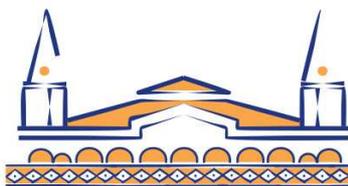


Figura 6. Configuração da aplicação Rato-Queijo.

Na estratégia de movimentação simples, para encontrar objetos específicos (*Rato-Queijo*), a tomada de decisão é realizada por etapas. Primeiro, o robô busca no ambiente uma marcação visual qualquer, faz a aproximação e posicionamento em relação a este objeto. Em seguida, processa a informação contida no QR Code, aplicando a movimentação definida pelo computador para atingir o próximo QR Code. Segundo a informação, o robô pode traçar uma rota em direção ao queijo, pode contornar um obstáculo ou ainda seguir em direção a outra dica, novo QR Code. Este procedimento se repete até que o robô encontre o queijo.

Usando a abordagem proposta, o robô cumpriu com sucesso a aplicação do rato-queijo em todos experimentos realizados. Ou seja, o robô conseguiu através do processamento de VC, se localizar, localizar o queijo e se movimentar em direção ao mesmo contornando os obstáculos encontrados⁵. Mesmo sem ter informações precisas de odometria e baseado somente nas informações transmitidas pelos QR Codes.

5 Mais detalhes sobre o robô nesta tarefa específica podem ser vistos em video no link <http://youtu.be/rt0oFDSj2mU>



CONCLUSÃO

O desenvolvimento de robôs móveis de pequenas dimensões tem sido usado como fator motivacional com boa repercussão educacional, pois estes envolvem vários aspectos do processo de engenharia e programação dentro do domínio de concreto. Aplicações com robôs móveis envolvem estudantes de engenharias e computação facilmente, além de possuir promissoras aplicações na indústria. Concomitantemente, vê-se a evolução da Visão Computacional (VC) como uma alternativa de sensoriamento que traz uma capacidade extra, flexível e variada para o contexto dos robôs móveis.

O robô apresentado neste trabalho, denominado Kbo (uma referência ao filho de *Cyclops*, por conter um “olho” somente) é um robô de baixo custo, construído sobre plataformas modulares e hardware aberto, que carrega uma câmera IP. As imagens capturadas pela câmera são tratadas por um módulo de software, denominado QRPO, que executa a VC, identificando posição, orientação e dados oriundos de marcadores (imagens padronizadas impressas em papel) dispostos no ambiente.

A solução para o problema Rato-Queijo pode variar em função da estratégia implementada. As decisões tomadas são baseadas na geometria e na posição do robô no plano que ele encontra-se no início da tarefa. Estas decisões afetam diretamente o tempo final de execução da tarefa. Vale lembrar que o tempo de processamento da imagem para a VC e o tempo de processamento da estratégia não são críticos, pois são processados em um computador que tem grande capacidade de processamento. Assim, o que realmente faz diferença é a qualidade da estratégia em função das limitações de movimentação do Kbo. Desta forma, para os estudantes é requerido que se concentrem no problema concreto e não nas dificuldades de processamento.

Sobre este arranjo foi composta uma tarefa, denominada *Rato-Queijo*, para a qual os estudantes tem que desenvolver uma estratégia de solução através de um algoritmo de controle. Devido ao baixo custo do robô e a disponibilização gratuita do QRPO, esta configuração pode ser facilmente replicada, o que vem a permitir estudantes a usar de maneira prática e efetiva VC em robôs móveis autônomos.

REFERÊNCIAS

ALCANTARA, M. F.; HOUNSELL, M. S.; SILVA, A. G. (2011) Alternative position, orientation and data recognition algorithms for augmented reality markers. **Anais: International Association for Development of Information Society –Conference on Applied Computing. IADIS-AC.** Rio de Janeiro. 557-561.

ALCANTARA, M. F. **Códigos de Barra Enriquecidos: Desenvolvimento e Aplicações.** 2012. Nf. Dissertação (Mestrado Profissional em Engenharia Elétrica – Área: Processamento de Imagens e Automação) – Universidade do Estado de Santa Catarina, Programa de Pós-Graduação em Engenharia Elétrica, Joinville, 2012.

ALLIANCE, Z. **ZigBee Alliance: ZigBee Intro.** 2010. Disponível em: <[http://www-zigbee.org/About/ZigBeeIntro.aspx](http://www.zigbee.org/About/ZigBeeIntro.aspx)>.



BAGNALL, B. **Maximum LEGO NXT: Building Robots with Java Brains**. [S.l.]: Variant Press, 2007. 524 p.

BIANCHI, R. A.; SIMÕES, A. S.; COSTA, A. H. R. Comportamentos reativos para seguir pistas em um robô móvel guiado por visão. **Anais: Simpósio Brasileiro de Automação Inteligente**. [S.l.: s.n.], 2001.

DAVIES, E. R. **Machine Vision Theory, Algorithms, Practicalities**. 3. ed. San Francisco, California: Elsevier, 2005.

LABIOD, H.; AFIFI, H.; De SANTIS, C. **Wi-Fi, Bluetooth, Zigbee and WiMAX**. [S.l.]: Springer Verlag, 2007.

LEGOGROUPS. Lego.com **MINDSTORMS NXT Home**. 2009. Disponível em: <<http://mindstorms.lego.com>>. Acesso em: 23/02/12.

K-TEAM. **The Khepera Robot**. 2012. Disponível em: <<http://www.k-team.com/>>. Acesso em: 23/02/12.

PROBST, G. M.; MELO, D. F. F.; ROLOFF, M. L. Openvisionbot - pesquisa e desenvolvimento de uma plataforma robótica móvel aberta. **Anais: 9th IEEE/IAS International Conference on Industry Applications (INDUSCON)**. São Paulo, São Paulo: 2010. p. 1–6.

SIEGWART, R.; NOURBAKSHI, I. R. **Introduction to: Autonomous Mobile Robots**. [S.l.]: The MIT Press, 2004.

VIECILI, E. B. **Projeto e Desenvolvimento de um Bípede**. Monografia de Graduação em Engenharia Elétrica. Universidade Regional do Noroeste do Estado do Rio Grande do Sul (UNIJUÍ). 2011.

WRUCK, R. **Especificação e Desenvolvimento de uma Plataforma para Orientação de um Robô Móvel Autônomo, Baseada em Sensores e Redes sem Fio**. Monografia de Graduação. Bacharelado em Ciência da Computação. Universidade do Estado de Santa Catarina (UDESC). 2011.



DEVELOPMENT OF A VISION-BASED MOBILE ROBOT FOR EDUCATIONAL PURPOSES

Abstract: *The development of mobile robots is an exciting challenge to students. It is quite easy to find solutions and kits for mobile robots, but it is not so easy to find them with computer vision support. The use of computer vision can replace several sensors at once (e.g.: proximity, position, orientation) with benefits because it can detect and measure distance and angle in relation to the objects, and are capable of identifying what is the object and its dimensions. In this paper the development of a home-made mobile robot platform for teaching engineering and robotics is presented. The robot, named Kbo, is arranged in a simple modular fashion with an open architecture with an open-loop controller for the wheels but uses computer vision to allow a robot to explore an unknown environment. Movement strategies for handling various tasks have been developed which were completed in a satisfactory manner thanks to the flexibility that comes from using QRPO (Quick Response Barcode with Position and Orientation) library. QRPO was included in the robot structure, along with a software that supports it, to allow a student to try out several solution strategies that uses computer vision. Kbo and QRPO altogether help teaching and learning of mobile robots and computer vision once they allow students to practice many engineering and computational features that are required to more vision-based autonomous applications.*

Key-words: *Education, Mobile robot, Computer vision, QRPO, QR code.*