

BoaVista – Sensor Dedicado de Visão Artificial Baseado em Hardware (Re)configurável

Luís V. Lima, José M. Almeida, Alfredo Martins, Eduardo Silva
LSA-ISEP, Rua Dr. António Bernardino de Almeida 431, 4200-072 Porto Portugal
{llima,jma,amartins,eaps}@isep.ipp.pt

Resumo

Este artigo aborda o projecto de um sistema de visão dedicado para robótica móvel autónoma, que beneficia das capacidades de execução paralela do hardware reconfigurável, processando em “pipeline” as imagens provenientes de um sensor de imagem CMOS de alto desempenho em simultâneo com a aquisição das mesmas. Apresentamos um sistema com a capacidade de adquirir e processar imagens com resoluções de 640x480 a uma taxa de 60 fps, baixo custo e capaz de disponibilizar para o sistema central apenas a informação pretendida extraída da imagem. Este ponto, permite libertar os recursos computacionais do robot traduzindo-se em reduções de consumo significativas e consequente aumento da autonomia energética do mesmo.

1. Introdução

O objectivo deste artigo, é descrever uma das múltiplas utilizações que podem ser dadas a circuitos de lógica programável. Na base deste trabalho está um projecto em desenvolvimento no Laboratório de Sistemas Autónomos (LSA) do ISEP-IPP e que se enquadra no esforço de desenvolver sistemas sensoriais inovadores e/ou dedicados que permitam um aumento das capacidades de percepção dos veículos autónomos projectados e desenvolvidos no LSA. Este desenvolvimento insere-se no âmbito do projecto BoaVista – “A Dedicated Vision System for Autonomous Mobile Robot Navigation” com financiamento da FCT - POSI/ROBO/43914/2002.

O projecto[1] visa o desenvolvimento de um sistema de visão (re)configurável com o propósito de diminuir o atraso de processamento de imagem e fornecer como resultado final informação com diferentes níveis de abstracção. É também considerado o facto de que o produto final deverá ter simultaneamente um custo reduzido e elevado desempenho. Como motivação para este trabalho está em primeira análise a limitação temporal de processamento de imagem nos *robots* da equipa de futebol robótico do ISEP-IPP, IsePorto.[2]

Adicionalmente, a evolução tecnológica ocorrida nos últimos tempos nos sensores de visão CMOS, relativamente às características de aquisição e de configuração, à relação qualidade/preço bem como no nível de integração, confere uma utilização

acessível e viável para o desenvolvimento de projectos a este nível, uma vez que todo o tratamento de sinal é feito internamente, eliminando desta forma fontes de ruído no processo de digitalização da imagem adquirida, fornecendo uma interface totalmente digital permitindo uma fácil integração com dispositivos de lógica programável.

Neste trabalho será feita uma breve análise do componente configurável utilizado, que é uma FPGA da Xilinx, a SPARTANII XC2S200E e do sensor de imagem LM9647 da National. Em [1] pode ser encontrada uma descrição de outras abordagens a sensores dedicados para visão artificial baseados em uControladores[3], DSP[4] e hardware reconfigurável [5].

2. Análise de requisitos

Um sistema de visão artificial tem inúmeras aplicações, desde simples supervisão de uma linha de montagem, passando pelo divertimento e lazer, mas também para trabalhos de exploração em ambientes inacessíveis ao ser humano, como seja a inspecção de condutas ou exploração de outros planetas. As características do sistema de visão variam com os requisitos das aplicações e metodologias utilizadas, sendo vantajoso a utilização de um sistema que possa ser modificado por forma a satisfazer os diferentes cenários.

A linha estruturante de orientação para o desenvolvimento do sistema BoaVista é a sua utilização nos cenários semi-estruturadas por cor. A principal aplicação deste projecto será na robótica móvel de onde se destaca o futebol robótico nas ligas de tamanho médio e pequeno, mas também em aplicações de micro-robótica e noutras competições de robótica como é o caso da prova de condução autónoma do Festival Nacional de Robótica.

Estes cenários aplicativos impõem alguns requisitos funcionais e também estruturais, tais como: a robustez física; o baixo consumo; identificar, classificar e processar informação de cor, fronteiras e identificação de objectos; o suporte de múltiplas câmaras ou sensores; permitir um *frame-rate* adequado à aplicação e com latência determinista e reduzida; a portabilidade de software suportado por várias arquitecturas de processadores e dispositivos de aquisição de imagem; a escalabilidade e modularidade permitindo diferentes

aplicações; a adaptabilidade às variações de ambiente (luz, cor, deslocação etc.); o fornecimento de estimativas e medidas sobre objectos no cenário bem como estatísticas de qualidade da medida; o baixo custo do sistema, do desenvolvimento do mesmo, bem como da manutenção do projecto; deve ser compacto e de pequenas dimensões, permitindo assim a sua utilização em múltiplos sistemas autónomos.

3. Projecto do Sensor de Visão

A possibilidade de reconfiguração de um sistema sem alterações a nível de estrutura de componentes de *hardware* proporciona substanciais vantagens. Entre estas destaca-se o incremento na rapidez e qualidade do projecto e do desenvolvimento, bem como a nível do utilizador final com inerentes possibilidades de adaptação específica para diferentes cenários de aplicação aproveitando eficientemente o hardware disponível.

Utilizando dispositivos de lógica programável como as FPGA's, temos a possibilidade de "programar" o hardware fazendo com que este perca o cariz de elemento de baixo nível e possa ser visto como "mais uma função" que terá de ser escrita em determinado programa. É de crucial importância neste tipo de dispositivos temos a possibilidade de implementação de diferentes algoritmos de processamento por hardware e consequentemente de forma paralela. A paralelização de processamento proporciona relevantes incrementos na rapidez de execução, redução de latência e de consumo energético.

Do ponto de vista do projecto BoaVista, todas estas possibilidades são de ter em conta uma vez que se enquadram plenamente no objectivo de construir um sensor de visão de alto desempenho e com grande possibilidade de configuração. Por estes motivos está a ser explorada a tecnologia FPGA para processamento de imagem, contudo existem alguns requisitos que não dependem apenas do elemento reconfigurável mas também do sensor de imagem. Neste último está a ser utilizado um LM9647 da National que encerra também características muito interessantes para o projecto, tais como:

- Resolução de 640x480 a 60 fps;
- Permitir a definição de janelas e sub amostragens da imagem, podendo nestes casos disponibilizar taxas de aquisição mais elevadas;
- Disponibilizar a informação em formato digital;
- Controlo do tempo de exposição.

A FPGA utilizada é uma SPARTANII da Xilinx. Este componente foi escolhido não só pelo preço competitivo que apresenta face ao de outros fabricantes, mas também pelo facto de esta conter, módulos de RAM suficientes, para construir os *buffers* necessários para o processamento da imagem.

Para ligação do sensor de visão dedicado com o sistema central foi usada uma interface USB usando CY7C68013 da Cypress Semiconductors, que integra num mesmo componente o "transceiver" e o controlador USB 2.0 "high speed", bem como um micro controlador da família 8051.

A integração de um microcontrolador permite o *download* do *configware* da FPGA pelo USB e a programação do dispositivo de lógica programável.

A estrutura do hardware descrito nesta secção, é apresentada na figura 1.

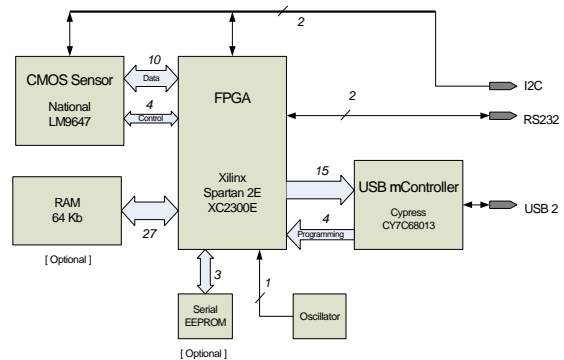


Fig. 1. Diagrama de Blocos do Hardware

4. O "Configware"

A figura 2 representa todos os blocos que constituem o projecto BoaVista.

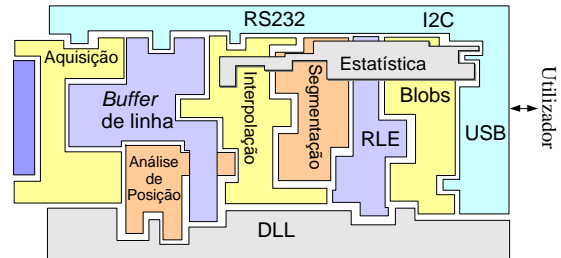


Fig. 2. Estrutura Geral

Todos os blocos de processamento de imagem desenhados na FPGA, com a excepção dos blocos de segmentação, trabalham como máquinas de estado integradas numa linha de processamento. Cada uma delas tem a seu cargo uma tarefa a realizar. Desta forma sempre que uma tarefa está concluída, o resultado é passado ao bloco seguinte que executará outra operação, até que por fim o resultado do processamento é disponibilizado para o utilizador.

A aquisição da imagem do sensor LM9647 é efectuada ponto a ponto, sendo a cadência de pontos idêntica à taxa do relógio, que pode tomar um valor máximo de 27MHz. No projecto, e uma vez que o oscilador utilizado no sistema é de 50MHz, está a ser usada uma frequência de 25MHz obtida pela simples divisão por 2 do *clock* do sistema.

A aquisição dos dados provenientes do sensor, é realizada por um bloco que faz a amostragem dos sinais de *pixel clock*, sincronismo horizontal e

vertical. A frequência de amostragem destes sinais é de 200MHz. Este valor é obtido a partir do cristal de 50MHz, usando 2 DLL's em cascata, das 4 disponibilizadas na FPGA. A frequência assim obtida, vai ser utilizada pela maior parte dos blocos desenhados, e servirá como relógio para o seu funcionamento. Desta forma podemos no mesmo bloco de processamento durante o intervalo de tempo entre dois *pixel* (cerca de 40ns) fazer com que em cada bloco possam existir 8 estados para a realização da tarefa que tem a seu cargo.

A imagem é obtida no sensor segundo o padrão *bayer*. Para obter as componentes de cor que faltam a cada *pixel*, é realizada uma operação de interpolação destas componentes de forma a ser possível apresentar cada ponto um formato RGB.

A operação de interpolação é dependente da localização do ponto que está a ser processado se se encontra numa linha/coluna par ou ímpar. Por isso, existe um bloco que determina a posição do ponto de forma a aplicar a correspondente operação de interpolação conforme apresentado na figura 3. A posição é obtida usando dois *bit's* que são complementados individualmente sempre que é detectado o início de uma nova linha ou um novo *pixel*. Desta forma a combinação dos dois *bit's* indica qual dos casos deverá se utilizado para a interpolação.

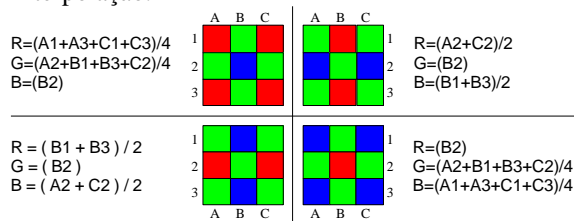


Fig. 3. Interpolação

Para a construção desta matriz, que designamos por Matriz de Interpolação, é necessário ter em memória as duas últimas linhas lidas bem como os dois últimos *pixel*. Para isso foram usados 3 *RAMBlocks* de 256 posições com 16*bit's* cada posição, construindo assim um buffer de linhas com 1536*Bytes* dos quais estão a ser utilizados 1280*Bytes*, correspondentes a duas linhas de 640 pontos. Este buffer permite guardar em cada posição dois pontos da imagem de linhas consecutivas e com a mesma posição na linha, sendo possível num único acesso á memória ler ou escrever dois pontos de cada vez, tornando o processo de actualização da Matriz de Interpolação e do *buffer*, mais rápido. Para proceder à actualização desta matriz seria necessário, ainda assim, realizar 4 acessos ao *buffer*, o que permitiria ler 8 dos 9 *pixel* que constituem a matriz e acrescentar o novo *pixel* adquirido. Para que isto não aconteça, a matriz é mantida em registos de memória RAM distribuída, consistindo a sua actualização, numa rotação à esquerda das colunas da matriz, na escrita na coluna da direita dos valores lidos do buffer (um único acesso) e por fim,

acrescentar o *pixel* que acaba de ser lido do sensor. Desta forma temos a matriz actualizada recorrendo ao buffer apenas uma vez. Este processo é ilustrado na figura 4.

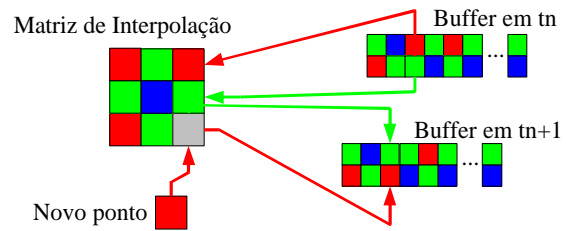


Fig. 4. Actualização da Matriz e do *buffer* de linha

Terminada a interpolação, o resultado deste bloco é passado ao bloco de Segmentação cuja função é determinar se o um ponto tem ou não alguma validade para a aplicação em causa, isto é, no caso concreto de um cenário de jogo de futebol robótico, um ponto de cor violeta, num cenário onde seguramente esta não deveria existir, é considerado como não válido, ou não segmentado, tomando por isso a designação de segmento não classificado. Esta classificação é realizada de acordo com limites atribuídos a cada uma das componentes de cor criando desta forma pequenos cubos dentro do cubo RGB, representando cada um deles um segmento a considerar da imagem. O bloco de segmentação é desenhado utilizando 15 blocos de comparação que funcionam paralelamente, tornando possível saber se o ponto pertence a algum dos segmentos comparando em simultâneo com todos os cubos. O resultado deste bloco é disponibilizado em 4 *bits* que codificam de 0 a 15 cada *pixel*, sendo 0 o código para um ponto não classificado.

A comparação é realizada recorrendo a lógica discreta, não dependendo estes blocos do sinal de *clock* do sistema mas sim de uma "flag" disponibilizada pelo bloco de interpolação que indica a existência de um novo ponto. Ainda dentro do bloco de segmentação, existe um selector de resultado, que elimina a múltipla segmentação. Esta pode ocorrer no caso de algum dos cubos definidos pelos limites estabelecidos tiver pontos comuns com outro. Este selector, actualmente dá maior prioridade ao segmento que tiver sido codificado com o menor dos valores possíveis.

O resultado da segmentação é passados ao bloco de construção de RLE (*Run Length Encoded*), a função deste é a de contar pontos classificados com o mesmo código de segmento e fornecer ao utilizador, a imagem segundo um formato onde é indicado o código do segmento que foi agrupado e quantas vezes este surgiu consecutivamente. O *reset* do contador é feito sempre que é detectado um código de segmento diferente do anterior ou quando termina a linha que está a ser processada obtendo uma compressão dos dados a enviar para o sistema central. Os RLE resultantes vão ser elementos base

de trabalho para o bloco seguinte que não está ainda implementado. Neste bloco serão realizados um conjunto de operações de teste de conectividade e caracterização de manchas de pontos da mesma cor (*Blobs*).

O ultimo módulo é o de comunicação USB, que actualmente é unidireccional e apenas permite o envio da informação relativa à imagem que está a ser processada, mas terá também na sua versão final a função de fazer chegar parâmetros para a configuração por exemplo dos limites da segmentação ou para a configuração do sensor.

Posteriormente será acrescentado um bloco que permitirá calcular alguns dados estatísticos como por exemplo a média de pontos classificados para cada um dos segmentos, permitindo no futuro estudar a possibilidade de auto calibração do sensor.



5. O Protótipo

Inserido na linha de desenvolvimento de visão para robótica móvel, em 2001 foi implementado um primeiro protótipo baseado no CMOS Omnivision OV7620 e uma FPGA de 10kgates efectuando apenas segmentação de cor não permitindo métodos de interpolação sofisticados. O protótipo actual da solução BoaVista designado por ERPROS que significa *Embedded Reconfigurable Processing System* é mostrado na Figura 5.

Fig. 5. Placa ERPROS com elemento reconfiguravel, ligação USB e sensor de imagem LM9647.

A implementação actual dos algoritmos descritos na secção anterior na plataforma de *hardware* ERPROS resultou na ocupação de recursos da FPGA apresentada na tabela 1.

| Descrição | Gasto/Total | % |
|------------------|-------------|----|
| Slices | 677/2352 | 28 |
| Slice Flip Flops | 338/4704 | 7 |
| 4 input LUTs | 1152/4704 | 24 |
| bonded IOBs | 37/146 | 25 |
| TBUFs | 48/2352 | 2 |
| BRAMs | 3/14 | 21 |
| GCLKs | 2/4 | 50 |

Tabela 1. Imagem do Protótipos - Placa ERPROS

Os consumos de todo o projecto, actualmente são da ordem do 1,5Watt isto somando ao valor estimado para a FPGA os consumos do controlador USB e de alguns circuitos indispensáveis para o funcionamento da placa como sejam reguladores de tensão, oscilador, circuitos de *reset*, entre outros. Este valor é significativamente reduzido, face ao consumo apresentado pelas actuais soluções de

sistemas de visão onde este atinge facilmente os 2Watt sem que seja realizado qualquer processamento da imagem, sendo para esse fim, necessário ter um processador que iria consumir algumas dezenas de Watt para a realização dessa tarefa.

6. Conclusões

Neste artigo foi feita uma breve descrição do trabalho já realizado no âmbito do projecto BoaVista – “A Dedicated Vision System for Autonomous Mobile Robot Navigation”, destacando o trabalho desenvolvido ao longo da linha da lógica programável. O processamento é efectuado por diversos blocos em paralelo, organizados numa arquitectura em “pipeline” originando um atraso mínimo na obtenção dos resultados do sistemas.

A capacidade de adquirir e processar imagens com resoluções de 640x480 a uma taxa de 60 fps com consumos energéticos muito reduzidos e um baixo custo, disponibilizando para o sistema central apenas a informação pretendida que é extraída da imagem, permitindo deste modo, quer a utilização em sistemas com escassos recursos computacionais onde antes seria impraticável a utilização de sensores de visão, quer libertar os recursos computacionais de sistemas onde o consumo de energia e de tempo de processamento é um factor importante, deixando assim o CPU central com mais tempo para processamento de mais alto nível.

Os desenvolvimentos futuros passam pela necessidade de testar todos os módulos em funcionamento totalmente integrados no protótipo e validação da interface USB do sensor dedicado com o sistema operativo GNU/Linux nos sistemas computacionais dos robots. Uma última vertente do trabalho futuro consiste na integração deste sistema nos *robots* da equipa de futebol robótico do LSA e em aplicações de micro robótica.

Referências

- [1] Luís Lima, José M. Almeida, A. Martins, António Patacho, Rui Picas, Eduardo Silva, BOAVISTA: Projecto de um Sistema de Visão de Baixo Custo em hardware dedicado e Reconfigurável para Robótica Móvel, In Proc. Robótica04, 2004, Porto, Portugal.
- [2] Almeida, J., A. Martins, E. P. Silva, J. Baptista, A. Patacho, L. Lima, V. Cerqueira, C. Almeida, R. Picas, “ISePorto Robotic Soccer Team for Robocup 2003”, In: RoboCup 2003 Int. Symposium, Padua, Italy.
- [3] Anthony Rowe, Charles Rosenberg, Illah Nourbakhsh, A Low Cost Embedded Color Vision System. In: Proceedings of IROS2002.
- [4] Sargent, R. and A. Wright. The Cognachrome Color Vision System, In <http://www.newtonlabs.com/cognachrome>
- [5] Johan André-Dinerf, Lars Asplund, et al “Chip Vision: A Vision System for Robots Based on Reconfigurable Hardware”, In: In: RoboCup 2003 Int.Symposium, Padua, Italy.