H. M. Silva, J. M. Almeida, L. Lima,
A. Martins, E. P. Silva, A. Patacho
LSA - Autonomous Systems Laboratory
ISEP/IPP- Instituto Superior de Engenharia do Porto - Potugal
{hsilva,jma,llima,amartins,eaps,anp}@lsa.isep.ipp.pt

REAL-TIME VISION SYSTEM FOR MOBILE ROBOTICS

ABSTRACT

This paper describes a real-time vision architecture for mobile robotics. It is integrated in the research program on mobile robotics pursued at the Autonomous Systems Lab ISEP-IPP. The implemented architecture is characterized by: low computational cost, low latency, low power, highly modularity, configurability, adaptability and scalability. A new method using run length encoding (RLE) colour transition allows real-time edge determination at low computational cost. A pipeline structure further reduces latency and allows a paralleled hardware implementation. A dedicated hardware vision sensor was developed in order to take advantage of the proposed architecture. The real-time characteristics and hardware partial implementation, coupled with low energy consumption address typical of autonomous systems applications.

Keywords: Autonomous mobile robots, image processing, real-time system, pipelined processing, vision.

1. INTRODUCTION

Artificial vision systems are key elements in robotics navigation and localization systems. This is due, to their great sensing capabilities and low cost. The following paper describes a intelligent real-time vision system architecture with a mobile robotics scenario in mind.

The presented implemented architecture has low computational cost, low latency, low power, highly modularity, configurability, adaptability, and scalability. The architecture allows pipeline processing, with modules hardware (Lima, 2004) or software implemented.

Although with the Robocup scenario has a benchmark scenario and the ISEP/IPP MSL ISePorto team (Almeida, 2003) as a initial application, the system is designed to be used in all kinds of autonomous systems, land, air or sea.

A dedicated hardware vision system sensor was developed implementing part of the overall vision architecture. There has been a continuous trend to the development of vision systems for mobile robots, since the use of active vision systems mounted in pan & tilt heads (Hai, 2003, Peig, 2000) to the use of real-time human tracking methods for autonomous mobile robots (Doi, 2001). The vision systems are having to deal with a more and more complex and dynamic environments, which will lead to the expansion and refinement of the image processing algorithms and their hardware embedded implementation.

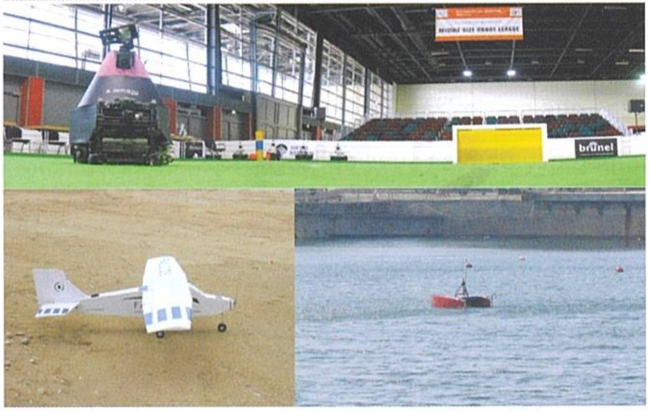

Figure 1 · ISEP/IPP Autonomous Vehicles in which the real time vision system has been integrated: on top the MSL Team of Robocup ISePorto, and on the left bottom the UAV FALCOS and on the right bottom the autonomous surface vehicle ROAZ.

Furthermore a new and innovative solution to edge identification based on the notion of run length encoding (RLE) transition will be presented.

2. VISION SYSTEM ARCHITECTURE

The architecture is developed in a way that the data flow works as in a pipeline, where some layers can be hardware or software implemented.

The system was also developed in a way that as data is processed, there is a decrease on the data size and a increase in the information abstraction thus most heaviest processing is done in the bottom layers.

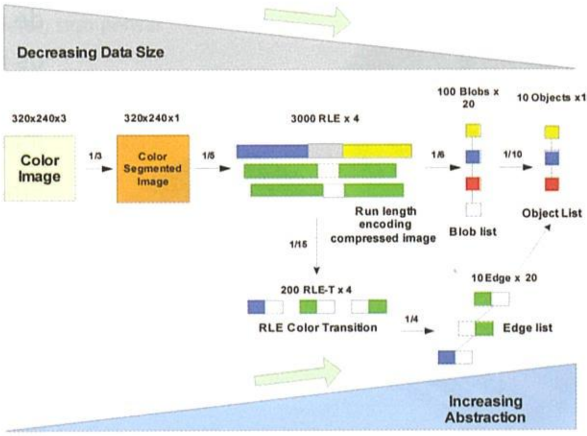

Figure 3 · Data size in the system pipeline.

Furthermore any other type of segmentation method is still valid and due to system modularity, and can be implemented only changing the segmentation module. After that, the pixels with colour information will be compressed using RLE. This compression method is used due to is effectiveness in conserving

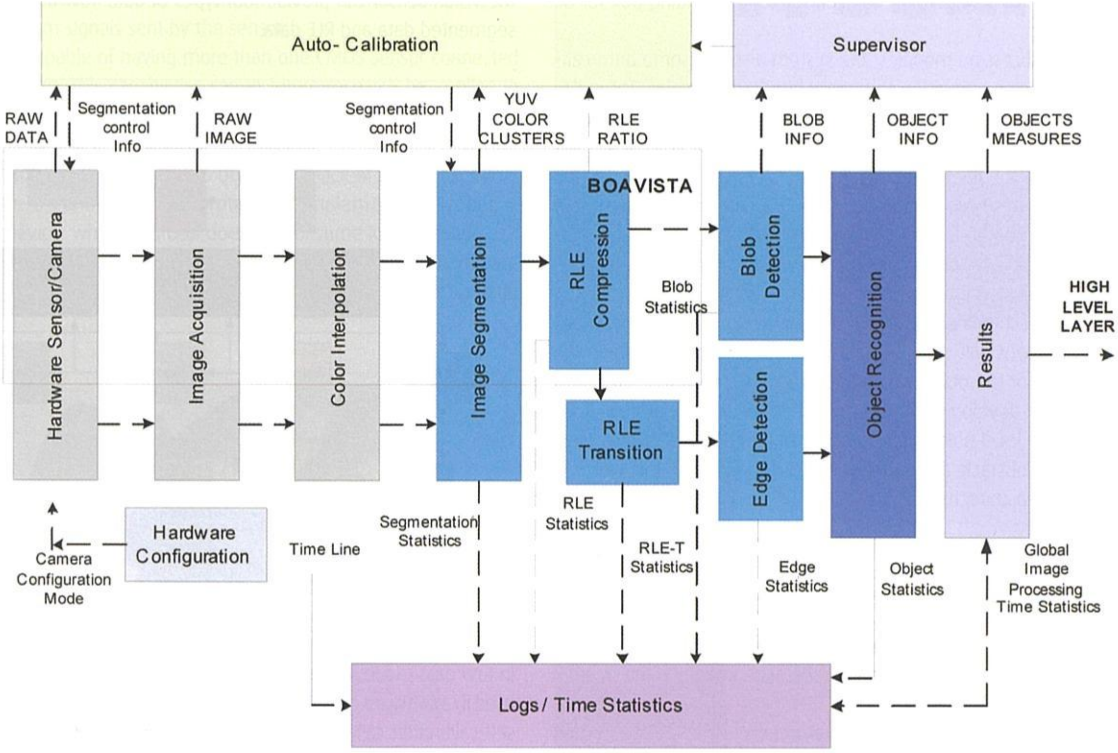

Figure 2 · Real Time pipelined vision architecture for mobile robotics.

the information and reducing data size.

One of the most positive and innovating aspects of our approach is that it allows to extract structured edge information. This starts in the RLE transition colour module. After all data has been compressed into RLE format, a run is executed to search previously sanctioned colour transitions. When one of this transitions occurs, a transition RLE is created and stored. It will have the same information as a normal RLE: image position, colour and number of pixels. The method also deals with color transactions uncertainty.

Which may occur due to interpolation issues, occlusion or illumination problems. (See Fig 4).

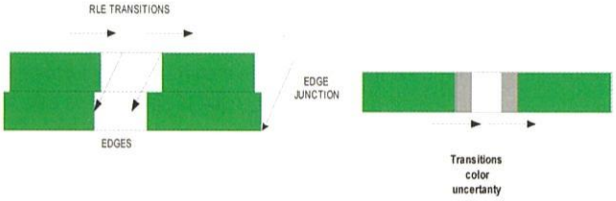

Figure 4 · Example of a RLE colour transition with edge formation.

The difference pixels between the end of the first RLE and of the second RLE is stored, so that later on computational weights in the object detection modules can be applied. The shorter the difference between pixels more accurate the bearing measures will be.

Last but not least, one of the key points of the method is that the number of RLE transitions are not directly attached to the number of image pixels, but are attached to the number of image RLE. So the computational cost of this method compared to other methods of edge finding in the same scenario (Hundelhansen, 2003) is residual. In our system is about 1/8 of the computational cost of the RLE module and only stores 1/15 of the RLE data.

Figure 5 · Example of robot vision blob algorithm and edge finding algorithm.

After all types of RLE have been processed, the RLE will be grouped into similar color regions (BLOBS), and the transition RLE will be grouped into an edge list.

Once all the RLE are grouped into BLOBS further processing is done in the top architecture layer, the high level data layer (Object Recognition block).

The high level data layer is constituted by the modules that detect features for the robot localization and navigation sub-systems, thus closely related with the application. Lower level layers are relatively application independent and can be used in multiple autonomous scenarios. Image information at this stage already contains edge and blob identification, allowing particular object search.

In (Almeida, 2003) the ball, robot markers, dark, blue and yellow objects are detected using a blob-based algorithm in conjunction with edge vectors.

This provides information regarding ball centroid and bounding box for all possible clusters.

Besides the data processing modules, the system also has some auto-calibrations tools like color calibration through white balance color. The auto-calibration tools are used to help the vision system deal with environment changes. Some of the segmentation and color interpolation parameters are sent to a calibration module, detecting color clusters shifting. In case that happens, information is sent to the acquisition module to change camera settings, a example can be found in (Browning, 2005).

Furthermore the high level data is being continually improved. At this moment new features involving the use of graph vectors as a way of achieving high level object descriptions (Wilson, 2005) are being implemented.

The connectivity graphs will allow structured image search, diminishing the computational cost of the object recognition modules. High level stereo is also currently under development, this module will not work at pixel level but will merge high level objects information. One of its most relevant applications would be obstacle avoidance in a Robocup scenario, this will lead to a new improved architecture.

3. HARDWARE VISION SENSOR

The hardware vision sensor BOAVISTA was developed in the ISEP/IPP LSA to free system resources from processing the most heaviest data processing. In order to do so, a FPGA platform is used to process the image on the fly from the CMOS sensor.

This sensor allows the implementation of lower architecture levels at a fraction of power required in standard CPUs by taking advantage of the inherent parallel nature of image information and architecture pipeline structure.

The substantial power reduction constitutes a fundamental advantage to the use in autonomous systems. It is thus possible implement advanced sensing capabilities in low power systems and widen the range and scope of applications.

This image processing layer within the FPGA is divided into different modules, allowing access by the overall system to different kinds of data. As a result, the vision sensor can provide four types of data: raw data, RGB mode data, segmented data and RLE data.

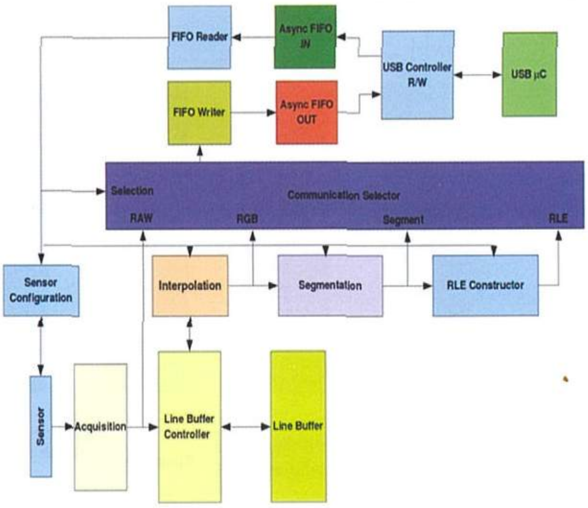

Figure 6 · Hardware Vision Sensor Modules

In raw data mode, the sensor sends only raw image in Bayer pattern. This is used in a software complementary statistical module that provides the system self-calibration capabilities. In RGB mode, data is sent to allow image gain control. In segmented mode, the data is sent to search possible color clusters. In RLE mode, the image is sent in a compressed lossless form.

In figure 7 an information processing pipeline time diagram description is presented. A maximum latency of 500 µs is achieved from the initial pixel acquisition in the CMOS sensor to the processed data reception at the user level application. This maximum latency includes processing and communication delays (USB bus).

The interpolation, segmentation and RLE modules are similar to the software

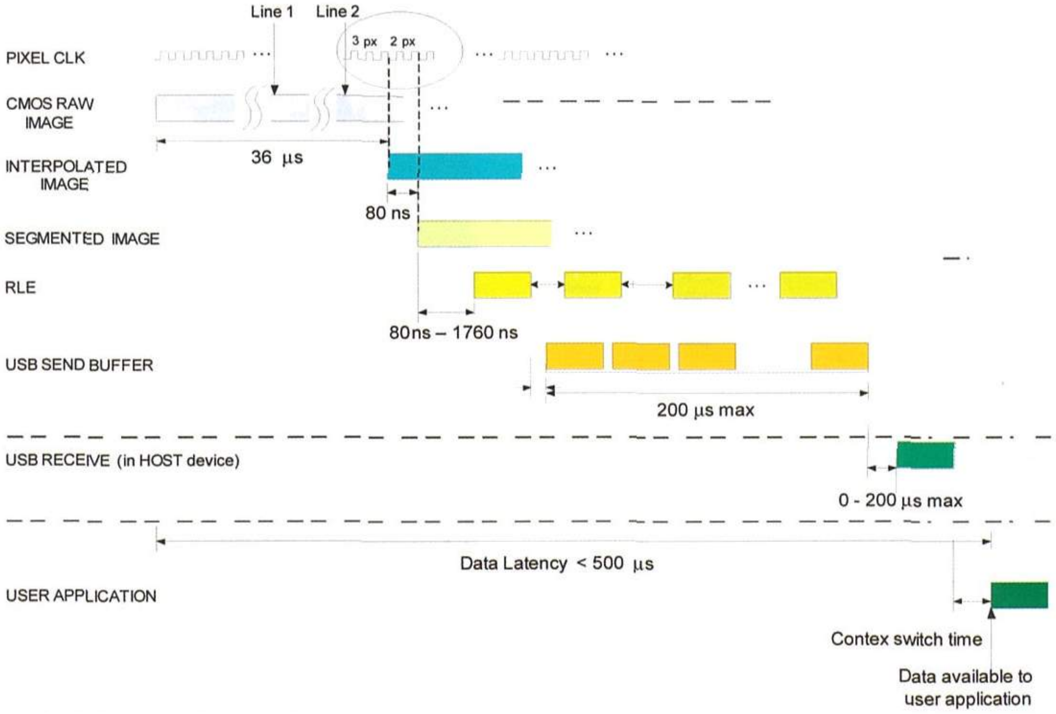

Figure 7 · Temporal Diagram detailing the time line in the Hardware Vision Sensor.

ones. The signal acquisition module contains an extern clock for sampling the data and synchronism signals sent by the sensor.

Moreover its also capable of having more than one CMOS sensor connected to the same platform. This capabilities can in future research be applied in stereo vision.

The communication link between the hardware sensor and the software application is made by USB connection. Due to the lack of resources within the FPGA device a communication protocol is implemented through a microcontroller device, whose protocol does not consume lots of resources.

To compare the use of the CMOS and FPGA tecnologies used in the hardware vision sensor, against other tecnologies applied in the mobile robotics scenario see (Lima, 2004).

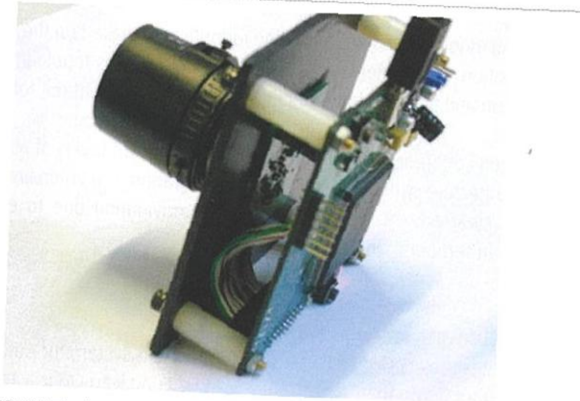

Figure 8 · BOAVISTA Hardware vision sensor image

4. OBJECT DETECTION FEATURES

The object recognition module is where the high level features recognised and interpreted by the other robot sub-systems are characterised. In Robocup scenario these objects are the landmarks and the world targets. These landmarks and world targets are detected using a deterministic object descriptor.

These object descriptors uses a specific language for each of the objects. Taking the goals as an example. A combination of edge and blob data is used to access the possible veracity of the candidates, moreover statistics are used to decrease the level of false detections. When this procedures are satisfied bearing and distance measures to the object are taken. The mixed information between edge and blob information, allows good reliability in the quality of the measures observed. Goal post bearing distribuition measurements with the robot stopped diagram as in figure 9 scenario is presented in figure 10. A normal probability distribution of the same measures can also be observed in figure 11.

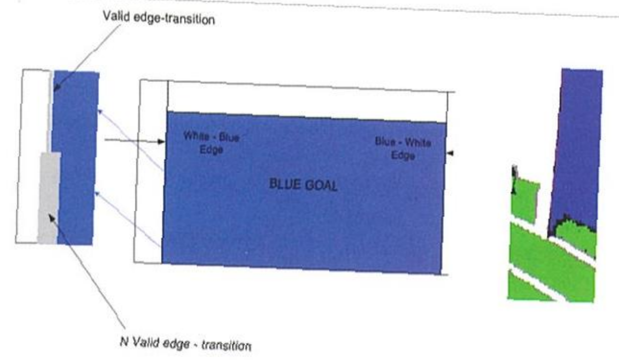

Figure 9 · Goal post edge-blob validation descriptor an real robot goal validation image.

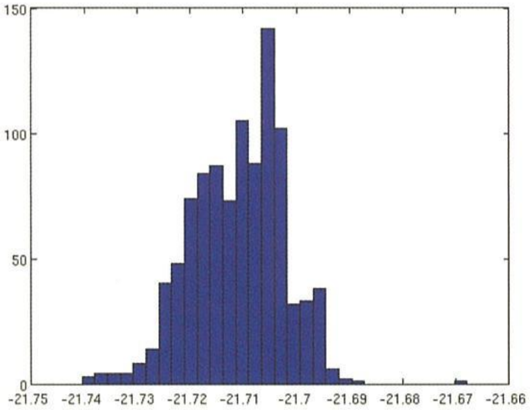

Figure 10 · Bearing measures between a robot and a goal post in degrees.

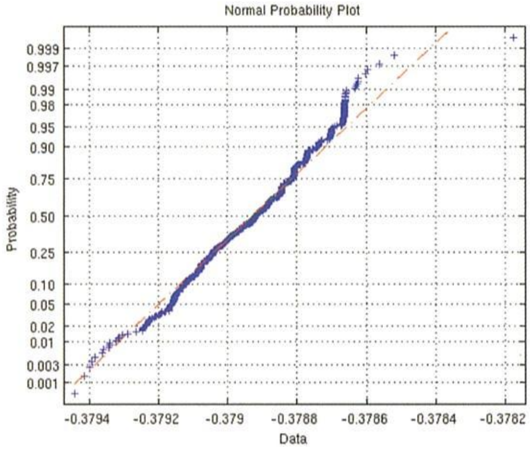

Figure 11 · Normal probability distribuition function

5. RESULTS

In Table 1. is shown the amount of average computational resources required to process a single frame. With the hardware vision system acquiring at higher resolution (640x480), the global image processing is still minor than with the software solution (320x240).

Table 1 · Comparison of the computacional cost of the different system modules between a hardware sensor plus software implementation or only software solution.

	Image Segmentation	RLE	RLE Transition	Blobs	Edges	Objects	G. Image Processing
Hardware vision system (640x480)	—	—	0,75	3,5	3,0	1,2	8,45 ms
Software (320x240)	3,0	2,5	0,3	1,4	1,2	1,0	9,40 ms

One of the biggest advantages of the hardware implementation is the reduced latency when compared to the software implementation (see table 2). This is due to the software implementation, use of image compression for dispatching image data to the USB connection.

Table 2 · Latency comparison between the hardware and software implementations.

	LATENCY
Hardware Vision System	< 0,5 ms
Software (standard USB cam)	> 50 ms

6. CONCLUSION

In this work we presented a real-time vision system developed in Autonomous System Laboratory ISEP/IPP for mobile robotics applications. The presented architecture allows latency reduction in sensor data reception. Very low power consumption solutions can be integrated.

System organization allows hardware and software transparent solutions implementation. A dedicated hardware vision sensor was developed to implement the more time consuming processing steps, taking advantage of image information parallelism. A high performance programmable logic device (FPGA) was used to process data from a CMOS sensor capable of VGA resolutions at 60fps. This sensor can use different image sensors and currently one with 3 Mpixels is under development.

Power consumption reduction was significant, being possible now to segment and compress image for less than 1W.

A new and innovative solution to edge identification based on the notion of RLE transition was implemented. This solution improves topological image information and allows edge detection in real-time and without total image processing.

Information coherence is maintained through different levels of abstraction in the architecture with pluggable module integration. The vision architecture provided clear advantages to mobile robot navigation due to enhanced precision in temporal image characterization.

Acknowledgments

The BoaVista project research – "A Dedicated Vision System for Autonomous Mobile Robot Navigation" was sponsored by Fundação Ciência Tecnologia (FCT) referência *POSI/ROBO/43914/2002*.

REFERENCES

Almeida, J., Martins, A., Silva, E., Baptista, J., Patacho, A., Lima, L., Cerqueira, V., Almeida, C., Picas, R., 2003: "ISePorto Robotic Soccer Team for Robocup 2003", In: *RoboCup 2003 Int. Symposium*, Padua, Italy.

Bruce, J., Balch, T., Veloso, M., 2000: Fast and Inexpensive Color Image Segmentation for Interactive Robots. *IEEE/RSJ International Conf. On Intelligent Robots and Systems*, 3(2000) 2061-2066.

Browning, B., Veloso, M., 2005: Real Time Adaptive Color based Robot Vision. In: *Proceedings of IROS2005.*

Doi, M., Nakakita, M., Aoki, Y., Hashimoto, S., 2001: Real Time Vision System for autonomous mobile robotics. In: *IEEE International Workshop on Robot and Human Interaction Communication* 2001.

Hundelhansen, F., Rojas, R., 2003: Tracking regions and edges by shirinking and growing. In: *Computer Vision Winter Workshop* (CVWW03).

Heath, M., Sarkar, S., Sanocki, T., Bowyer, K., 1996: Comparison of edge detectors: In *IEEE Computer Vision and Pattern Recognition* 1996.

Hai, Z., Kui, Y., Jindong, L., 2003: A Fast and Robust Vision System for Autonomous Mobile Robots. In: *Proceedings of IEEE Intelligent Conference on Robotics, Intelligent Systems and Image Processing* 2003 China.

Lima, L., Almeida, J., Martins, A., Silva, E., 2004: Development of a dedicated hardware vision system for mobile robot navigation. Submmited *robotica 2004 international conference.*

Peiig, J., Skrikaew, A., Wilkes, M., Kawamura, K., Peters, A., 2000: An active vision system for Mobile Robots In: *IEEE* (2000).

Rowe, A., Rosenberg, C., Nourbakhsh, I., 2002: A Low Cost Embedded Color Vision System. In: *Proceedings of IROS2002.*

Wilson, R., Hancock, R., Luo, B., 2005: Pattern Vectors from Algebraic Graph Theory. *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol 27, nº 7, July 2005.