A Real Time Vision System for Autonomous Systems: Characterization during a Middle Size Match

H. Silva, J.M. Almeida, L. Lima, A. Martins, and E.P. Silva

LSA - Autonomous System Laboratory
Institute of Engineering of Porto
Rua Dr Antonio Bernardino de Almeida 431 4200-072 Porto, Portugal
+351 22 834 0500 (ext. 1409)
{hsilva,jma,llima,amartins,eaps}@lsa.isep.ipp.pt
http://www.lsa.isep.ipp.pt

Abstract. This paper propose a real-time vision framework for mobile robotics and describes the current implementation. The 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 autonomous systems applications. A characterization of the implemented system in the Robocup scenario, during competition matches, is presented.

1 Introduction

Artificial vision systems are primordial elements in robotics navigation, localization and perception. This is due, to the their great sensing capabilities and low cost. Furthermore embedded solutions with hardware parallel implementation are starting to surface allowing to solve one of the key problems in artificial vision, the high computational resources required.

Considering robotic vision to be a real-time problem, there are certain amount of functionalities, that the vision system must possess in order to face the highly dynamic environmental changes and be able to track external moving objects. Namely, all robotic vision frameworks have to deal with real time aspects and restrictions imposed on the robot modules. System designers must balance between using time and computational consuming methods, against more simple methods that allow low energy consumption and still have a good degree of robustness. Finally all environmental properties must be perceived, in order to control the vision system and allow the autonomous system to work.

As a consequence several vision software tend to be application oriented. In order to prevent this, a conceptual architecture is required. Most state-of-art frameworks for real time applications are mainly concern with the image processing modules[1][2]. There are other applications like active vision systems mounted in pan-tilt heads [3][4] or real-time human tracking methods for autonomous mobile robots[5]. Therefore, an overall real-time framework was developed to cope not only with the need to solve the image processing modules problems, but also through the use of a paralleled hardware

implementation[6] to be able to handle problems regarding image acquisition, power consumption and latency between frames.

One of the advantages of an modular framework, is that the different modules can be replaced or adapted to different applications and to different environmental conditions (indoor or outdoor, structured or not).

The modules are integrated in a pipeline structure architecture having two streams of data. One, the main data which concerns only the main modules, and the control data which connects the main modules with the auxiliary ones and provides the system with other type of self-adjusting and logging capabilities.

The framework here described was developed using the Robocup Middle Size League team ISePorto has a benchmark scenario, and has already been integrated in multiples autonomous vehicles [7][8] in different applications scenarios operating both indoors and outdoors.
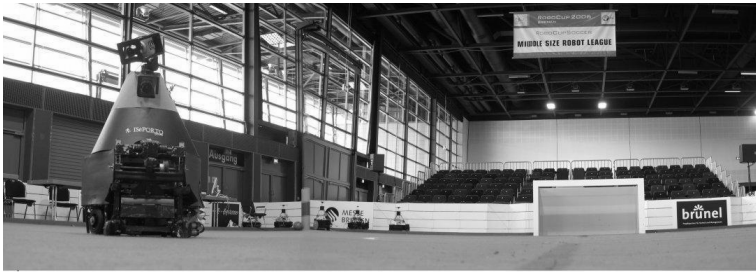

Fig. 1. ISePorto Robot

This paper is structured in the following way. In the ensuing section, we present the overall architecture of the system followed by a shortened explanation of the architecture modules, there inputs and outputs. In section 3 the hardware embedded solution is presented and its modules detailed, followed by results of temporal analysis and quality of measures obtained. These results where obtained using the framework in a Robocup scenario, some during a real Robocup match. Finally in section 5 a conclusion regarding the overall system functionalities and future research will be presented.

2   Vision System Architecture

The proposed architecture, see Figure 2, follows an image processing pipeline approach. Where some of the layers can be hardware or software implemented. The pipeline starts with the acquisition layer (three left blocks) that takes care of all hardware related and image acquisition configurations, camera settings and so on.

The system is programmed to acquire frames from different types of devices (embedded, USB). It detects which type of camera is plugged in, and automatically starts to acquire frames using that device. This layer also contains the color interpolation module, whose function is to assigned color information to image pixels. The system can also integrate, logarithmic scales, IR and monochromatic image types.
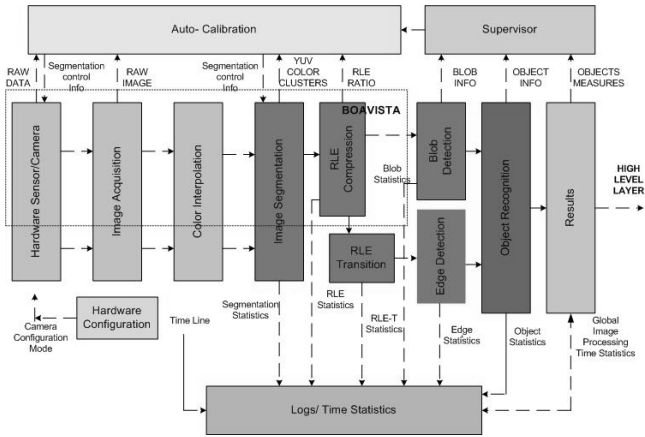
Fig. 2. Vision System Architecture

The image processing layer is responsible for the refinement and abstraction of the image data. After all image pixels have been given texture info, they will be segmented using the selected method. Currently, they are segmented into previously defined color clusters following a method proposed by[9]. Any other type of segmentation method, based on regions, histograms, clusters, neural networks, etc, can be used due to system modularity, by changing the segmentation module. We can for example use an hybrid approach like[10], that combines image segmentation with extraction of edge topological information. Afterwards, the pixels with color information will be compressed using run length encoding (RLE).

This compression method is used due to is effectiveness in conserving the information and reducing data size. Afterwards a run is conducted to look for previously sanctioned color transitions. When one of this transitions occurs, a color transition is created and stored. It will have similar information as a normal RLE: image position, color and number of pixels, but also deals with color transactions uncertainty. These may occur due to interpolation issues, occlusion or illumination problem.

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. In our system is about 1/8 of the total computational cost of the RLE module and only stores 1/15 of the RLE data. After all types of RLE have been processed, the connected RLE will be grouped into similar color regions (BLOBS). 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. Besides the data processing modules, the system also has some auto-calibrations tools, that are used to help the vision system dealing with environmental changes. In our system a white balance calibration is done to allow perception of the illumination

changes, some of the segmentation and color interpolation parameters are sent to a calibration module, that detects color clusters shifting. Statistical analysis on mean value for color pixels provides information for controlling camera parameters.

The architecture high level modules are being continually improved. Currently high level stereo is under development, this module will not work at pixel level but will merge high level objects information provided by the acquisition devices, leading to a change in some of the higher architecture modules. This method is less time consuming than merging camera information at pixel level.

3   Hardware Vision Sensor

The hardware embedded sensors are an emergent solution in robotics and autonomous systems applications. This is due to their hardware reconfiguration capabilities, low cost implementation, low energy consumption and low hardware concentration.

The vision systems built with reconfigurable and embedded hardware have some advantages over a standard vision system namely: cost, size, energy, computational resources and latency. It also possess some disadvantages like: processing capabilities, development cost and fixed point arithmetic.

These devices are a solution when dealing with hardware costs issues, several solutions exists using CCD and CMOS technologies. This one has the advantage of allowing more sensors addiction and with the use of field programmable gate arrays (FPGA) achieve low level processing with a low energy cost, taking advantages of the inner parallel processing architecture of this devices.
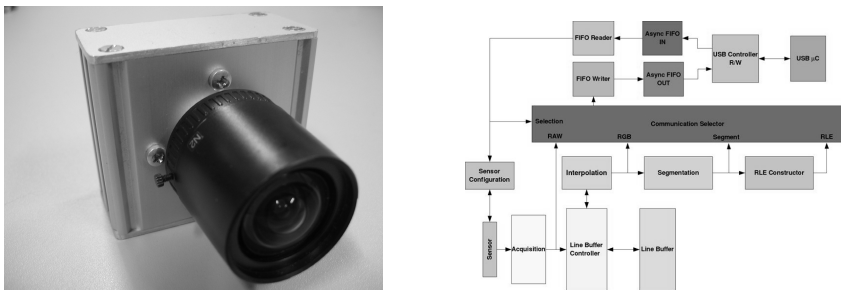

Fig. 3. Hardware Image Sensor Architecture

So taking this issues in consideration, the embedded vision sensor BOAVISTA was developed to free system resources from processing the most heaviest data. 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 namely in Robocup fields technology.

It is thus possible to 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 of 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.

In figure 4, 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 ones, for more information see[6].
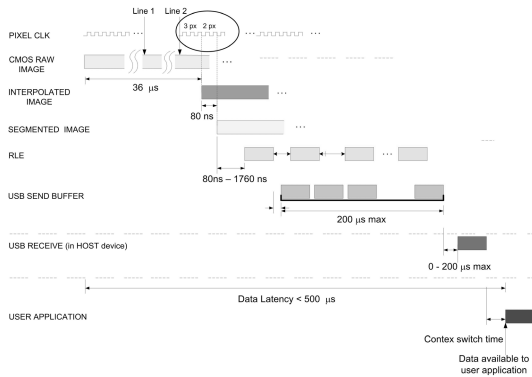

Fig. 4. Time diagram of the Embedded Hardware Vision Sensor

4   Results

The results presented in this paper where taken during a set of live Robocup Matches, and allows to establish and understand the type of data and resources that are needed to cope with a Robocup Vision System. This enables the use of this scenario has a benchmark scenario for other robotics applications.

The first experiment was to monitor the amount of data that each module of the vision system pipeline would generate during a match and how much time each processing modules consume. In order to do so we monitored the image processing modules during a Robocup Match. We can see the time analysis of each module by frame.

Table 1. Image Processing Modules Time Statistics (ms) during a Robocup Match

Image R.(320x240)	Seg+RLE	Color T	Edges	Blobs
Mean	6.045	0.345	0.425	0.708
St.deviation	2.032	0.375	0.696	0.979

These experiments were made during a live Robocup Match, with a robot with two cameras Head and Kick (2.8mm, F1.4,maxvision MVL2810M) both at 320x240 resolution, placed at a vertical height of 60 cm and 50 cm accordingly.

The most time consuming module is the segmentation/RLE module. Which is never under 5 ms, due to the minimum computational time required to perform segmentation and run length encoding operations, to an image with a 320x240 resolution in a PIII 1.2Ghz Tualatin.

The other modules, have residual time consuming performances and do not influence significantly the overall performance of the robot vision system.

The second experiment is intended to characterized the amount of data generated in each of the processing modules. This tests are important, in order to understand which parts of the system should or should not be integrated in embedded hardware and what advantages could come from that migration. This experiment shows how the information flows through the pipeline. We can see that the most heaviest data is processed in the early stages of the pipeline. Which enforce our choice of migrating these modules to an hardware embedded solution, thus achieving the same results at a fraction of the power consumption cost.

Table 2. Image Processing Modules Dimensions Statistics during a Robocup Match

Image R.(320x240)	RLE	Color T	Edges	Blobs
Mean	1988	468.3	96.2	140.2
St.deviation	758.9	196.7	41.2	78.2

Furthermore, this statistical results obtained during a live Robocup Match, can also be used to help the vision system to obtain self-adjusting capabilities. When the RLE statistical numbers increase dramatically to an unexpected number during a large quantity of frames, that would indicate that the segmentation parameters were off-balanced and that the robot should re-adjust his camera settings or color configurations.

In a Robocup scenario there are some statical objects that can be used by the robot has landmarks in order to achieve self-localization. One of them are the field goals. In the third experiment bearing measures to a field goal were taken during a match.

In figure 5 bearing measures observed are shown for a 5 minute period of match. We can see that there are only seven false positive occurrences during that period. These false positive are ignored by the vision system due to the fact, that all bearing measures have a reliability factor that in the case of the field goal, is the bearing variation of the vertical post of the goal. This variation (see figure 5) is much higher in the false positive cases, more than 2 degrees that the normal standard deviation of a goal post inferior to 1 degree.

Other landmarks present in a Robocup field are the corner posts and the field lines. In figure 6a we can see a histogram, with the different range distances detection to a corner post. There is a great deal occurrences in the 2 meters area probably due to the robot movement.

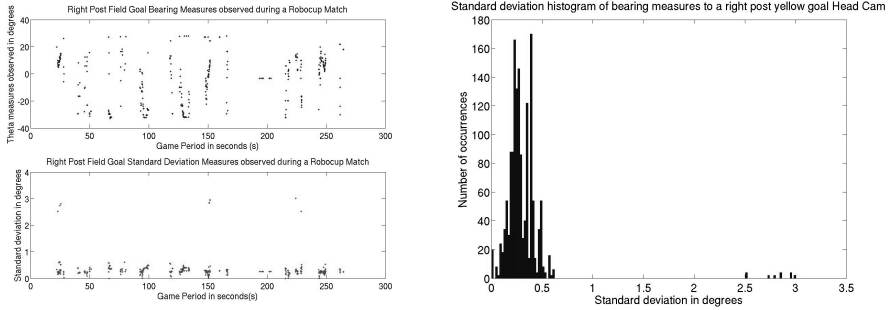In figure 6b a histogram showing field lines t-junction distance range of detections is displayed.

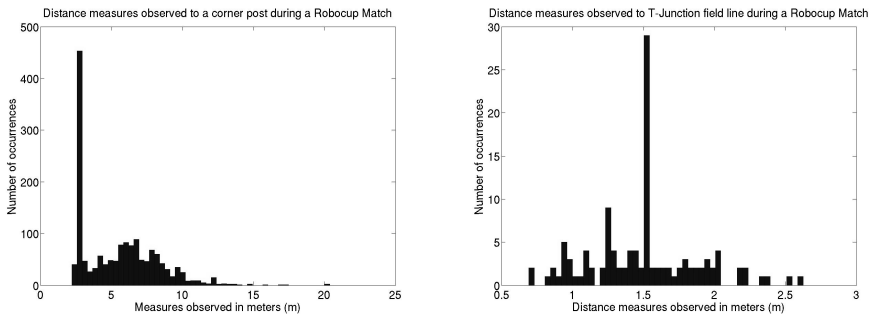**Fig. 5.** Bearing measures to a goal post and standard deviation

Fig. 6. Max. Distance Range to a corner post in a Robocup field and distance measures observed to a t-junction

5   Conclusion

In this work we presented a real-time vision system for mobile robotics and autonomous systems applications. The presented framework architecture allows latency reduction in sensor data reception. Very low power consumption solutions can be integrated. Our proposed organization allows a hardware and software transparent 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 60 fps. This vision sensor can use different image sensors with a higher frame-rate, resolutions and High Dynamic Range Image capabilities for used in outdoors applications.

The presented results, taken from a Robocup Match, allowed a overall evaluation of performance of the system, as well as, the characterization of the output data from each one of the vision processing modules.

Power consumption reduction was significant. It is now possible to segment and compress image for less than 1W. Information coherence is maintained through different levels of abstraction in the architecture with ploughable module integration. The

vision architecture provided clear advantages to mobile robot navigation and advanced image perception systems, having also been applied to fire detection with Unmanned Aerial Vehicles.

References

1. CMVision, `http://www.cs.cmu.edu/~jbruce/cmvision`
2. Hager, G., Toyama, K.: X-Vision: A Portable Substrate for Real-Time Vision Applications. In: Computer Vision and Image Understanding, January, vol. 69(1), p. 2337 (1998)
3. Hai, Z., Kui, Y., Jindong, J.: 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 (2003)
4. Peiig, J., Skrikaew, A., Wilkes, M., Kawamura, K., Peters, A.: An active vision system for Mobile Robots. In: Proceedings of IEEE Intelligent Conference on Robotics, Intelligent Systems and Image Processing 2000, Takamatsu, Japan (2000)
5. Doi, M., Nakakita, M., Aoki, Y., Hashimoto, S.: Real Time Vision System for autonomous mobile robotics. In: IEEE International Workshop on Robot and Human Interaction Communication (2001)
6. Lima, L., Almeida, J.M., Martins, A., Silva, E.P.: Development of a dedicated hardware vision system for mobile robot navigation. In: Robotica 2004 International Conference (2004)
7. Martins, A., Almeida, J.M., Silva, E.P., Pereira, F.L.: Vision-based Autonomous Surface Vehicle Doccking Manoeuvre. In: MCMC 2006 7th IFAC Conference on Manoeuvring and Control of Marine Craft, Lisbon, Portugal (September 2006)
8. Martins, A., Almeida, J.M., Silva, E.P., Santos, F., Bento., D.: Forest Fire Detection with a Small Fixed Wing Autonomous Aerial Vehicle. IAV (submitted, 2006)
9. Bruce, J., Balch, T., Veloso, M.: Fast and Inexpensive Color Image Segmentation for Interactive Robots. In: IEEE/RSJ International Conf. On Intelligent Robots and Systems, vol. 3, pp. 2061–2066 (2000)
10. Pavlidis, T., Liow, L.: Integrating Region Growing and Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(3) (March 1990)