REAL-TIME IMAGE ANALYSIS

***Dan – Laurențiu GRECU, PhD, Faculty of Informatics , University of „ Titu Maiorescu",Bucharest, Calea Văcăreşti, nr.187, Romania, Email:* danlaurentiugrecu@gmail.com**
Andrei-Laurențiu SAVU, student, Faculty of Informatics, University of „ Titu Maiorescu",Bucharest, Calea Văcăreşti, nr.187, Romania, Email:
andrei.laurentiu.savu@gmail.com.

Abstract – This article presents some results on analyzing various images with multiple purposes from traffic sign recognition to target recognition for various autonomous machines. Our projects aims to produce a software capable of real time object tracking while using a low cost platform, in this case the Raspberry Pi 2. The base of our project stands in the Markov chain, image processing and entropy algorithm. The program is a 6-step process: image capture, image filtering and color separation, the Markov chain algorithm for detecting possible polygons, target properties general physical properties analysis, the entropy algorithm, which analyses the possible zone of interest and reach a conclusion and a monitoring system, which prevents false positives. In the future we plan to develop a human and voice recognition extension.

Index terms: *Raspberry Pi 2, Linux, Debian, traffic sign recognition, C++11, OPENCV.*

I.INTRODUCTION

Once with the evolution of the industry, many new possibilities have been created, which can help a person in various tasks.

For example, cars which were a few decades ago a luxury have become a common possession for a person. As a response to this change the number of vehicles has increased exponentially and as a result the traffic risk has also raised. For example, there are lots of accidents on a daily basis just because some of the drivers do not pay enough attention to the road, not seeing some of the traffic signs and won't have enough time to react in case of imminent danger. There is a statistic made by World Health Organization that road traffic injuries caused an estimated 1.24 million deaths worldwide in the year 2010 which means that one person is killed every 25 seconds in traffic accidents. A possible response to the number of accidents may be a system that can inform the driver in a polite manner, about traffic signs which may not been seen or other possible dangers in real time. This idea has been implemented in the last 2-3 years by some premium and high-level automobile constructors, but their implementation is not cost effective while we want to provide a suitable solution for all drivers.

Another example is the risks that certain tasks can present to a person, in special on hazardous environments, such as a fire or a nuclear disaster area. The solution can be a series of specialized robots that can resolve certain tasks. Again, a major problem is the costs of such robots.

The purpose of resolving the mentioned problems we have analyzed the idea of developing a application which can work on a cost-effective platform.

For this challenge we have choose the following hardware: Raspberry Pi2 microcomputer for the image processing and 2 electrical engines connected to it in order to simulate a maneuvering device. For the software we have choose to implement a solution using the C++ programming language , 2011 revision which provides us with multithreading support and an image processing library OpenCV.

OpenCV is a C++ library that has a well meant purpose to help developers implement image relating tasks like "Facial Recognition System", "Gesture Recognition", "Object identification", "Motion Tracking", "Segmentation and Recognition", etc.

II.MAIN IDEA

The main idea of the project is to capture the image using a camera connected to the Raspberry Pi2 to transmit the image and process it in almost real time without using neural networks which are exceptional slow.

The data flux which has reached the memory buffer will be directly processed in order to reach our target on different levels. A first level of image processing for color isolation, a second one which uses a polygon detection algorithm (based on the Markov chains) to detect potential ROI (region of interest). On the third level our regions of interest are isolated from the original data flux and reanalyzed using a more complex algorithms for color filtering, special physical properties of the desired target. On the last level a two process entropy algorithm is used to check the existence of a desired target and returns a corresponding code from which a set of actions are triggered such as the slows down, maneuvers or certain device activation. The system can been see on the schematic bellow.

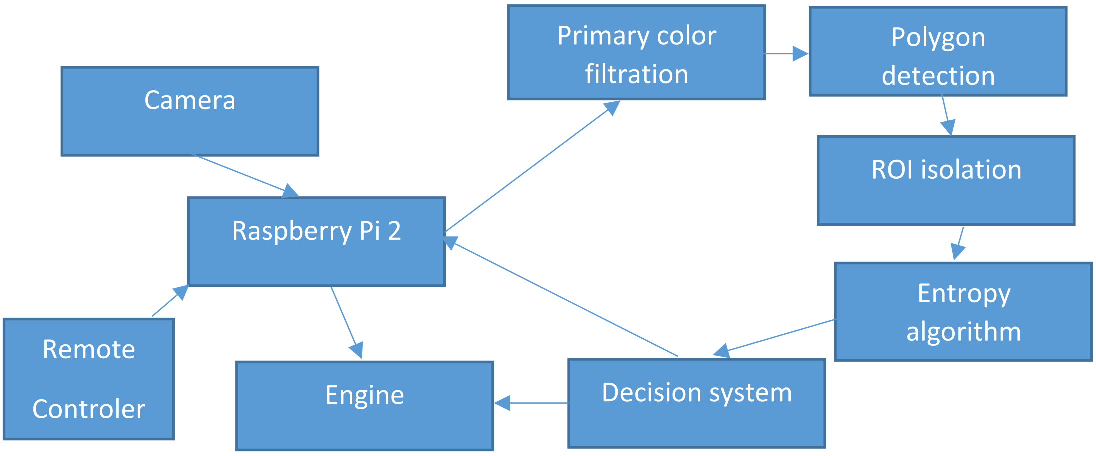

Figure 1 Basic schematic of the system

The system is based on the Raspberry Pi2 for its small size factor and capability to directly connect electronic motion devices, cameras and a Wi-Fi Adapter, which offers the capability to remote control, and see the results from a distance on terminals using Android or Windows operating systems.

As a general consensus from this moment this system will be referred as Real Time Traffic Assistance or RTIA.

II. HARDWARE DESCRIPTION

For RTIA the most important aspects are its low price, a high degree of reliability. In order to reach our target within acceptable boundaries we selected the following components as a minimum base for our solution deployment:

- Raspberry Pi 2 model B, a microcomputer which has a quad-core CPU clocked at 800MHz and 1 GB of DDR3 RAM has the highest efficency ratio for our porpouse and reach real time analysis of the road through our program. Also it posses multiple connectivity options such as 4 USB ports, 1 CSI camera interface , ethernet port, Wi-Fi support and 40 GPIO pins all which offers posible expansion capabilities.

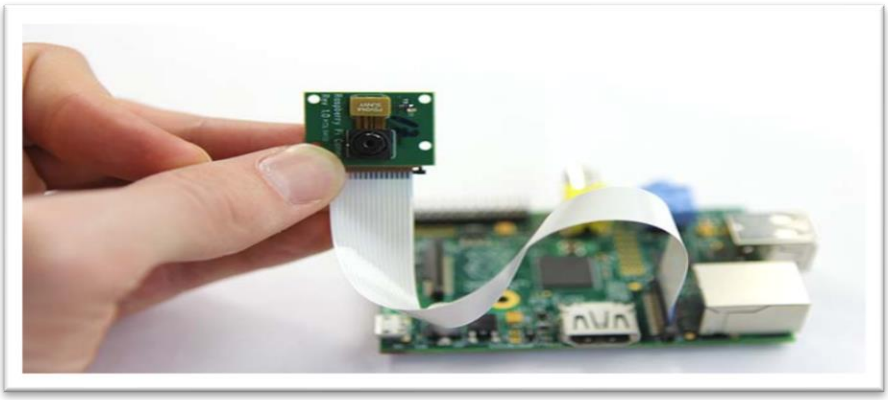

Figure 2. Raspberry Pi2 model B computer with its camera

- Raspberry Pi Camera Board is a small camera that can record high quality images at HD resolutions and offers acces to the data flux directly.

- 2 micro-motors which offers us the capability to simulate a mobile platform connected at the Raspberry Pi2 GPIO pins.

Combining these three elements, we can obtain o prototype capable of simulating at minimum level an automotive platform and has the capabilities to use our software to recognize signs at an acceptable level.

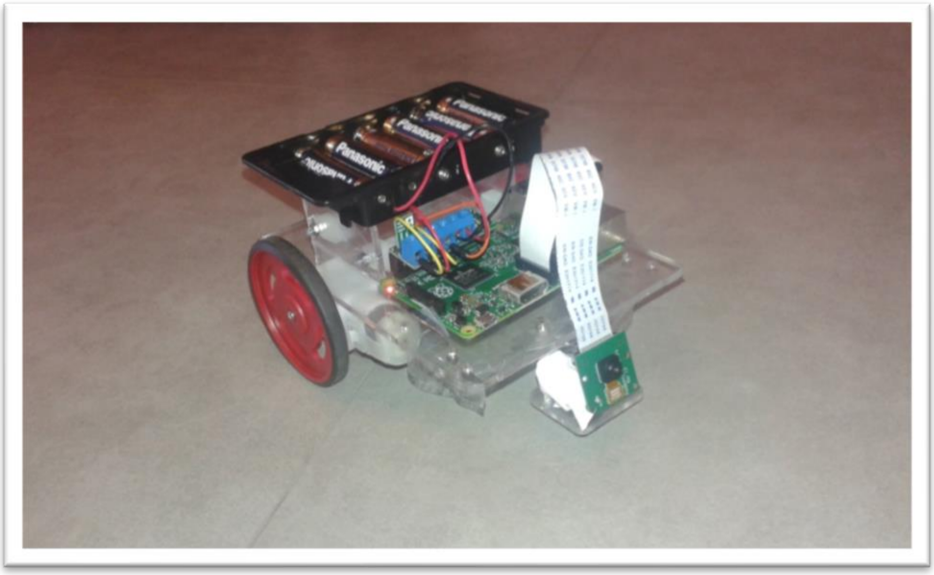

Figure 3 Robot simulating a care equipped with the RTIA system

III. SOFTWARE DESCRIPTION

Our software is developed in the C++ 11 programing language, which offers extended multithreading capabilities, useful for lowering the processing power requested from a single core, its capability of combined procedural, object oriented programming and the possibility of accessing data at bit level.

The compiler used for this project is Microsoft Visual Studio, which complies with the latest C++ standard and offers extended debugging, resource monitoring tools and a variety of project management solution (Object Designer and Viewer, Multi-platform management, SDL (Security Development Lifecycle) check). Those tools help us simplify our development project and even provide us with a solution to build ARM-based programs for devices such as Raspberry Pi2.

The main purpose of our program is for it to run on low-cost platforms such as the one chosen by us but not necessary. The application is capable of running in two modes, which are available depending on the platform that is used, and its processing capabilities. Those two modes of operation are Real Time Mode and a reduced sized mode Picture Frame Mode.

Real Time Mode

This is the complete version of the RTIA and can analyze data on real time using the following processes:

The first step in running this application is gathering the necessary data about the camera specification, which directly affect some of the running parameters of the whole process tree.

In the next level, one of the main reasons for choosing the new 2011 standard of the C++ is to reduce the stress on a single-core. On this step we are using multithreading for running two processes concurrently, capture the data flux for basic filtering respectively saving the image in the memory buffer and performing more advance filtering using all the remaining usable cores. This is necessary for two reasons: execution time and original data availability for future uses. The basic filtering consists of analyzing the data flux gathered in the HSV system to isolate the regions, which contains zones with the colors typically found on the target such as red, blue, yellow and generates a new image frames, which contains just the colors that interests us individually.

In the third step, the filtered image runs a polygon detection algorithm, which is based on the Markov chains. The image is analyzed by a 7×7 array from which possible contours are deducted and stored in a buffer and reused when new arrays reach the processing stage.
As a result of the algorithm we are generated a list of possible ROI, which even before the analysis is completed are used to launch the next process.
On this step we use the power of multithreading to analyze multiple ROI concurrently as those are generated by the polygon algorithm. On this step, execution speed varies greatly depending on the number of cores available on the system. The ROI are isolated on individual image objects and after the new image is filtered in a more advanced way. This advance filtering consists of image blur cancelation and deep color filtering and generation of the entropy table for the image. The entropy table and aspect ratio of the image is then compared with a constructed database and returns a value, which corresponds, to a certain action.

On the last level, a decision system that runs on the main thread is implemented that receives the values sent on the previous step. This information is centralized and counted on consecutives frames to see that the target is not a false positive. This failsafe has the role of preventing errors cases, which can have great consequences.

This mode can be used to create semi-automated or automated devices at a low price.

Picture Frame Mode

This is a scaled down version of the system. The primary difference between it and Real Time Mode is that, in this mode a number of images are gathered at a regular interval of time. Those images are then processed using the same algorithm based on the entropy but does not use the multithreading and using a RGB spectrum instead of the HSV which is less time consuming. Also in this mode the decision system does not have access to control mechanisms because this mode is less reliable. This mode is recommended for single core systems or less than 256MB of RAM.

This mode is useful for assisting a variety of operators in different objectives.

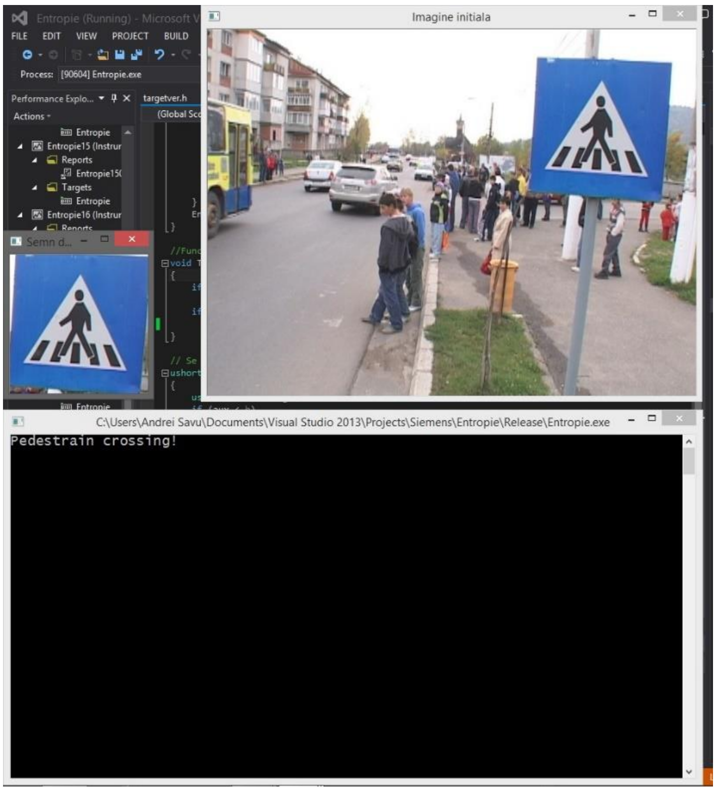

Figure 4 Demonstration of Picture Frame Mode

IV. CONCLUSION

To conclude this subject, we aimed to create a new add-on like software meant to help support a large number of tasks with a cost-effective implementation.

The complexity of the algorithms combined with the help of OpenCV library and the wonders of multithreading, we had developed a reliable and dynamic module that can interpret images in real time. Because this is still in the project stage, we may add new features like GPS-module for new functions.

References

1. *Anthony Williams,"C++ Concurrency in Action", Manning*
2. *C.Dascalescu, Programare orientată pe Obiecte (C++) - Note de curs, unpublished*
3. D. Knuth, "The Art of Computer Programming" Volumes 1-4, ISBN-13: 978-32175104
4. *https://opencv.org*
5. *https://github.com*
6. *R.Boriga, Programare procedural - Note de curs, unpublished*