EYEVISION: An Innovative Framework for the Development of Artificial Vision Systems

Maura Pasquotti[1], **Marco Del Pin** [2]
[1] Area Science Park, Trieste, Italy
[2] EIDON KAIRES S.r.l., Trieste, Italy

Abstract - *This paper contains the description of EYEVision, a framework developed by EIDON for computer vision based systems. This software is designed to be multiplatform; its modular structure allows to use processing modules from different manufacturers and its intrinsic structure reduces the learning curve for newly introduced development team members.*

Keywords: Framework, Artificial Vision Software.

1 Introduction

EIDON, which is a privately held research company, has been working for the past 30 years in the field of high performances artificial vision based systems, which comprises Real Time elaboration of high dimension images, systems with cluster of cameras and tailored elaboration to very precise cases of study that cannot be traced back to standard solutions. All the research projects developed required an initial phase for problem formulation and a following one for the development of the solution, which comprises both the theoretical solution and the software implementation.

This approach allows to obtain a solution really tailored to the specific problem but requires a lot of human resources to be applied. In particular the development of a specific new software solution each time, means that a brand new code needs to be developed for every single project, with a lot of time wasted and a requirement of knowledge of the whole project by the software developer to effectively realize the software.

The objective is then to develop a configurable artificial vision software that allows the researchers to generate a complete system without starting from scratch, but re-using the knowledge acquired from the previous projects and adding only the specific algorithms required to solve the new problem.

At the state of the art, in both industrial and academic fields, there are some software tools to help developers to reuse code in order to create a new implementation. Some of these solution are well established and cover a wide range of standard applications (for example smart cameras, see [8] for further details) while others are specifically developed for a narrow field of interest (for example [9] a vision platform for mobile robot applications) and are very specialized.

However none of them presents both the characteristics of flexibility and high performances which are required to develop complex computer vision systems that operate in completely different environments, as Eidon currently does.

To overcome these limitations a brand new platform has been developed in the previous years and EYEVision is the results of this research line.

As analogy in the building industry, old development scheme can be considered as a house built brick by brick while the EYEVision way can be represented as a luxury prefabricated house, a quicker and cheaper version of the previous construction method.

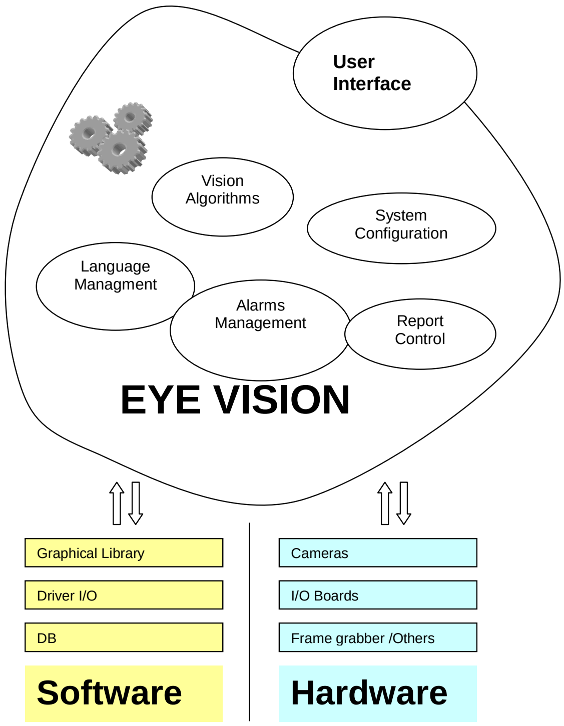

Figure 1. EYEVision Structure

2 Architecture

The EYEVision software has been designed with re-using patterns. The whole software can be divided in three different parts:

- The Acquisition Unit;
- The Elaboration Unit;
- The Display Unit.

Each part is independent from the others and can be developed separately.

The innovative idea implemented in the software is that each module communicates with the rest through an abstract interface layer, which is independent form the actual implementation, and is a plug-in style software

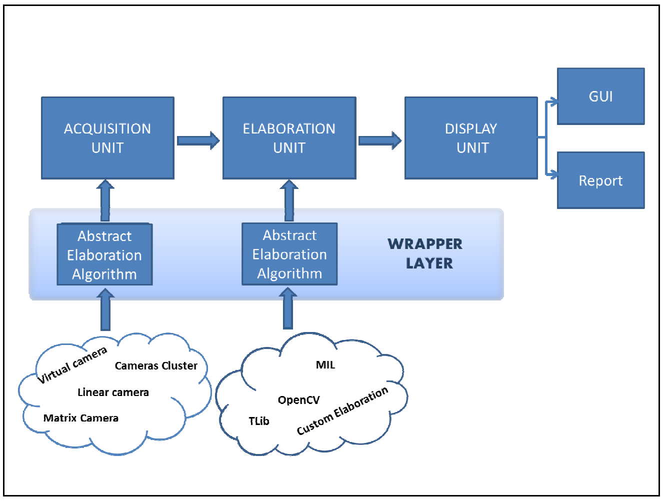

Figure 2. EYEVision Architecture.

.

3 Acquisition Unit

The idea that lays under the structure of the acquisition module is the presence in the software of a "generic camera" object, that interacts with the rest of the software..

This generic object is an abstract interface, under which there is the real device used. This scheme guarantee that changing the input device means that a new wrapper needs to be implemented, while the rest of the software remains untouched.

In this way the input camera can be easily changed and for example a virtual camera, that reads the data stored in the HD instead of the frames grabbed, can be realized.

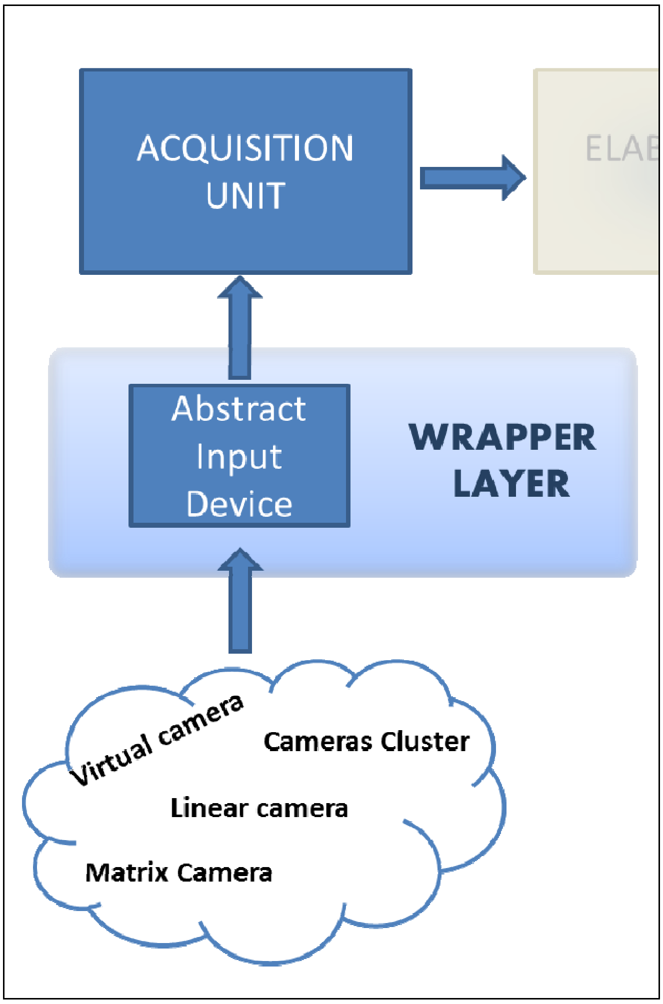

Figure 3. Acquisition Unit.

4 Elaboration Unit

Also the elaboration part of the software shows a plug-in structure. In this part there are a catalog of well-established image elaboration algorithms that can be enhanced with ad-hoc elaboration.

To add a new elaboration to the system the programmer need to follow some rules, but doesn't need to write all the code from scratch.

The presence of the wrapper layer in the software allows the developer to choose the appropriate graphical library that mostly fits their elaboration requirement without worrying about rewrite all the standard elaborations. In fact all the elaborations are one of this:

- Custom developed processing that requires innovative elaboration;
- Image processing that relies on standard algorithms.

In the latter case the standard algorithms are not directly called by the function, but are called through a library wrapper; changing the library, the elaborations remain the same and the only part that needs to be rewritten is the wrapper interface for the new library added. In particular the graphic library could be chosen between several options, that includes the OpenCV (Open Source Computer Vision library [3]), the TLIB ([4]), the MIL (Matrox Imaging Library [2]).

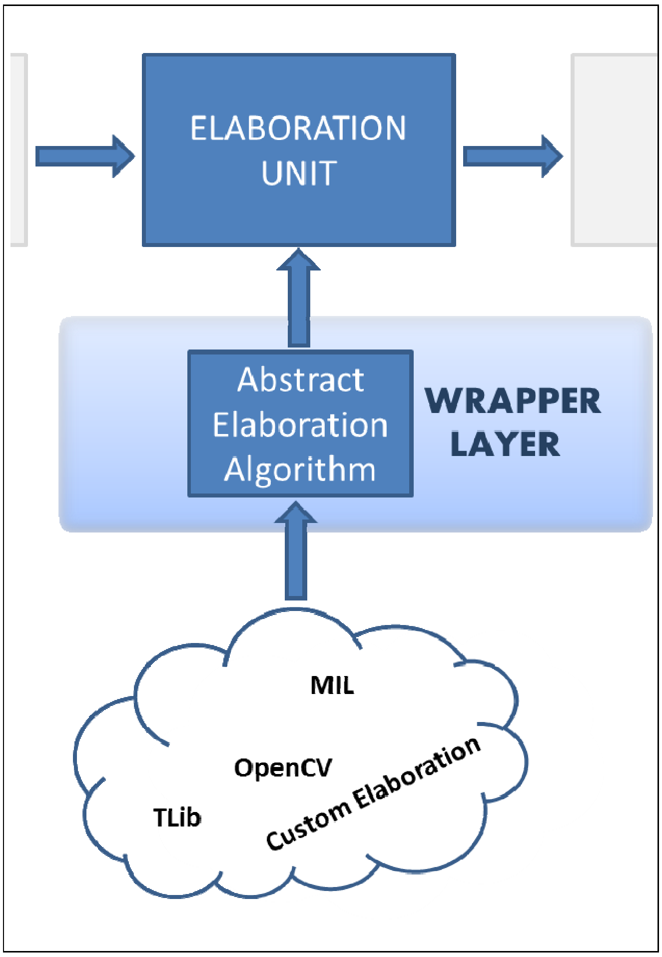

Figure 4. Elaboration Unit.

5 Display Unit

The Display Unit includes all the visual interaction between the user and the software.

It is composed by a Graphical User Interface (GUI) which shows on the monitor the results of the elaborations and the system status for monitoring purposes. The GUI presents some panels to calibrate the system before its normal functioning and in case also to define a new set of rules to create a different elaboration.

As well as the other parts of EYEVision, also the GUI presents a modular structure and is easily configurable; there is the possibility of adding various panels, starting from simple standard winform to more advanced vector graphic representation. There is also the possibility of adding data through the network.

In addition there is also a report module for statistics purposes, which is capable to store in a database all the results of tests carried out, and to retrieve them in order to see the analysis and their trends, to print reports that display the results obtained from the system analyzed in terms of pieces per lot, per shift, etc..

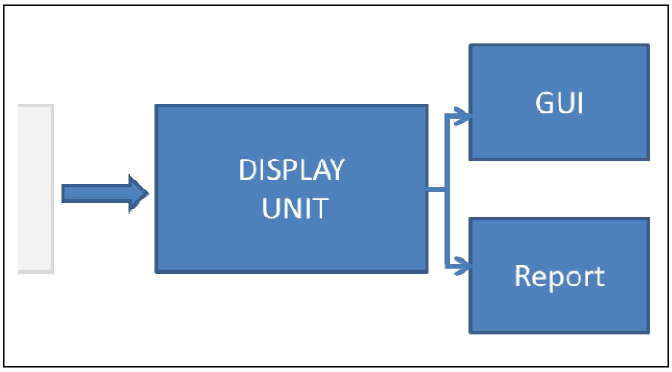

Figure 5. Display Unit.

6 Cases of Study

There are several examples of systems that can be easily generated with the EYEVision framework; figure 2 shows some possibilities.

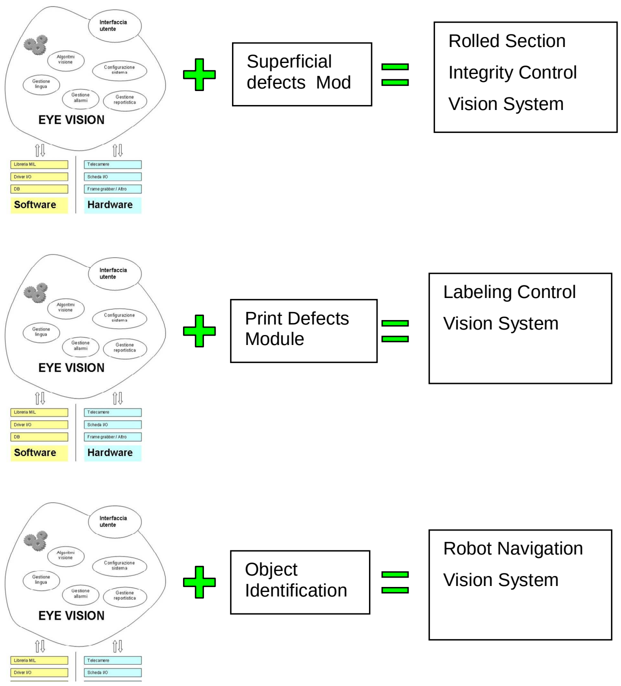

Figure 6. Example of system implementation with EYE Vision

In this paper we focus our attention to the specific problem of acquisition, classification and registration of natural stone slabs.

The physical system designed to solve the problem is essentially composed by a linear color camera and a conveyer belt for slab handling, as shown in figure 7.

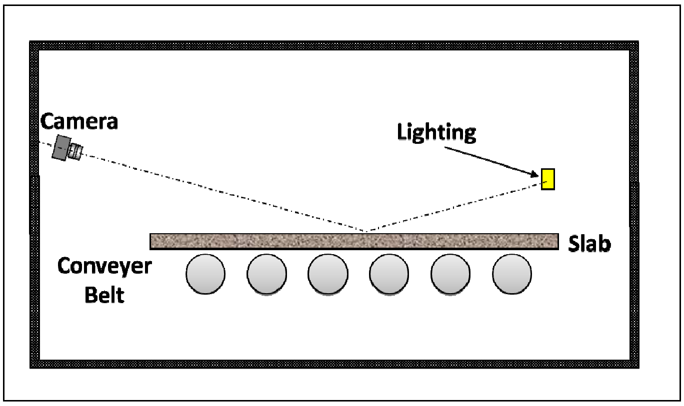

Figure 7. Slab Acquisition System.

Another requirement of the problem is the capability of the final system to supervise several acquisition units physically located in different places.

The analysis developed need to solve the following problems:

- Slab profile detection;
- Maximum usable slab surface;
- Slab pattern homogeneity recognition;
- Slab color homogeneity recognition;
- Slab quality calculation.

Some of the analysis required for this specific problem case can be carried out by standard algorithms, while others require ad-hoc solution.

The software development consists in the coding of the following:

- A wrapper for the linear color camera;
- A custom algorithm for the calculation of the maximum rectangle inscribed in the slab, in order to determine the biggest regular piece that can be cut off the slab;
- A custom algorithm for inhomogeneity detection for color and surface patterns;
- A module for multiple systems aggregation and monitoring.

All the rest of the code required to successfully solve the problem (acquisition management, interaction between acquisition devices and elaboration, graphical results presentation, etc.) is already a part of EYEVision software, so doesn't need to be implemented again (in figure 8 there is an example of a standard GUI panel for the slab detection project).

All the custom modules development can be carried out by a single person or split among a development team; in both cases each person doesn't need a deeply knowledge of the details of the whole system, but needs only to know the communication interfaces and the common rules for module implementation. Moreover, since EYEVision platform is realized in .NET development environment, a new module can be virtually written using any of the several programming languages supported.

This means that a new team member doesn't require a long and intense training period before effectively giving their contribution to the development process.

The time saved using this approach instead of developing the solution from scratch is considerable.

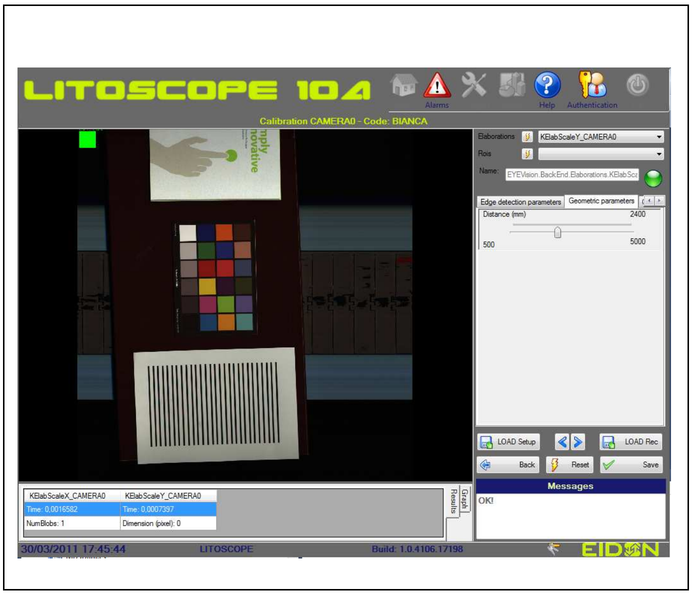

Figure 8. Graphical User Interface.

7 Conclusions

Current resource and time-to-market constraints on the software development process push developers to create a final product as quickly as possible. To accomplish this task developers need to base their development on successful past experiences [1]. Moreover, they should be able to reuse these experiences in well-structured computer-aided processes. In this context, the EYEVision framework appears to be a promising approach.

In this paper we have described its structure and pointed out the advantages of this framework, which includes:

- Reusability. The component is not recoded when the components are used in other domains or contexts, because the component implementation can be adapted to new business rules by changing the non-intrinsic dependencies. Then these components can be coupled with others components
- Adaptability. Programmers are offered the possibility of modifying the component descriptor by altering the final component functionality.
- Scalability. The system can be easily scalable because we obtain new component implementations and new component specifications. Then these new components with their dependencies can be used to compose new systems.
- Compressibility. Developing a new system is based on following a set of structured phases (Design and Specification, Implementation, Package, Assembly and Deployment).
- Reliability. The reuse of components already tested and approved increases the safety of the program

Acknowledgment

The authors would like to thank Mr Santoro and all the EIDON team for their support, and the AREA Science Park, Trieste, Italy.

References

[1] G Bertrand, M., Object-Oriented Software Construction, New York, NY: Prentice Hall, 1988.

[2] http://www.matrox.com/imaging/en/products/software/

[3] http://opencv.willowgarage.com/wiki/

[4] Sebastien Grange, Terrence W. Fong, and Charles Baur, "TLIB: A real-time computer vision library for HCI applications," Digital Image Computing - Techniques and Applications Conference, December, 2003.

[5] Dijkstra, Edsger W. A Discipline of Programming. Englandwood Cliff, NJ: Prentice-Hall, 1976.

[6] Fayad, M. E., M. Cline. Aspect of Software Adaptability. Communications of ACM, Vol. 39, No. 10, pp.58-59, 1996

[7] http://nuke.eidon.it/

[8] http://en.wikipedia.org/wiki/Smart_camera

[9] Sandor Szabo, David Coombs, Martin Herman, Ted Camus and Hongche Liu, “A Real-time Computer Vision Platform for Mobile Robot Applications”, Real-Time Imaging Volume 2, Issue 5, October 1996, Pages 315-327