

An Image Processing Online Laboratory within the iLab Shared Architecture

[doi:10.3991/ijoe.v6i2.1289](https://doi.org/10.3991/ijoe.v6i2.1289)

Michael Niederstätter, Thomas Klinger and Danilo Garbi Zutin
Carinthia University of Applied Sciences, Villach, Austria

Abstract—This paper focuses on a new way for students to deepen their knowledge in image processing and analysis with the help of an online laboratory. The existing solution provides a remotely controlled visual presenter which is used to project a document or book to various kinds of screens. The visual presenter can be controlled remotely and it provides the possibility to design, create and upload your own autofocus algorithm. Via the serial port and an image acquisition card the visual presenter can be controlled with the LabVIEW 8.6 software (National instruments, USA). Because of numerous advantages such as single login and reservation system for more online laboratories the whole experiment is realized within the iLab shared architecture.

Index Terms— image processing, autofocus, LabVIEW, iLabs, remote laboratory

I. INTRODUCTION

Using online laboratories has the potential of significantly reducing obstacles related to cost, time-inefficient use of facilities, inadequate technical support and limited access to laboratories. Nowadays there exist a large number of online laboratory solutions, but expenditure is growing and the life cycle of online laboratories software and hardware is becoming shorter.

It is still difficult to share instrumentation and experiments among laboratories. Each one has its own security policy and adopts a proper technology in accessing and controlling real devices. A common integrated framework, offering indexing facilities, unique logins, file sharing and the seamless access and run of experiments, is the main challenge in order to create a network of online laboratories. Grid technologies can be used to set up an effective network of remote laboratories for education purposes by sharing instrumentation and resources. However, the evolution of remote laboratories from the current client/server architecture to grid-based architecture requires well-defined tools for location, security, and integration of resources, and further research is currently being conducted to examine this issue [1-3].

As Universities and other institutions are likely to develop their own solutions and standards to deliver online laboratories to their users, no trend to a unique standard is observed, creating an obstacle against sharing these online laboratories. Considering the current scenario, a migration towards standardized solutions for delivering online laboratories becomes necessary to ensure software reusability and therefore facilitate online laboratory development and sharing.

At this point, the ISA (iLab Shared Architecture [4]) comes into play. ISA is a software architecture developed at the MIT (Massachusetts Institute of Technology) that

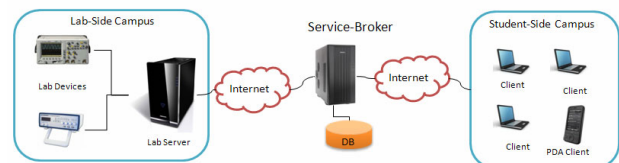


Figure 1. The iLab Shared Architecture [4]

facilitates a cross institution sharing and management of online laboratories. It offers a framework that establishes clear rules governing the communication between clients and their respective online laboratory servers by means of an API (Application programming Interface) based on Web services SOAP calls.

ISA proposes a classification for Online Experiments, on the one side are the batched experiments and on the other the interactive experiments. Due to their differences on the implementation, operation and technical requirements, a different architecture (ISA) exists for each one of them, although the iLab team at the MIT has already released a merged architecture in August 2008.

The performed work focuses on an online laboratory where users monitor and control various aspects of the experiment during its execution. This requires the implementation as ISA Interactive experiment which can be accessed only by one user at a time.

When accessing the online laboratory a user has to authenticate through a middleware server (Service Broker), which handles then all further connection with the Lab Server, where the hardware is connected and the Software is running. Before getting access on to the laboratory a reservation for the laboratory has to be made. Based at the settings various number and lengths of time slots can be reserved. When successfully reserved and being within this timeslot the reservation can be redeemed and a connection to the laboratory will be established. This connection is directly between lab server and Client. Via a ticketing service the service broker informs the lab server when the reserved time slot expires and the connection will be cancelled. [5]

All interfaces of the iLab architecture are realized with web services so that a high flexibility is given and that they can run autonomously on different places.

II. THE INTENTION

The objective of this work was to provide an online laboratory for image processing. When performing experiments in image processing and image analysis mostly expensive equipment like cameras, acquisition cards and various other components is required. Another circumstance is that the devices can be used mainly by one per-

son or group at a time, which makes it very difficult to organize hands on laboratories in a bigger student group. A common solution for this problem was mostly to use sample pictures or videos of pre-set problems to apply various image processing and image analysis methods in practice.

Because of the permanently high interest of students to work with real devices and to implicate real world side-effects this remote laboratory offers a significant alternative to classical teaching methods.

The overall idea was to offer the student a visual presenter which can be controlled completely remote over the internet. So students can experiment from wherever they want and whenever they have reserved the access to the laboratory. As can be seen on figure 3 the main functionalities are to zoom and to focus the image. Next to this also an autofocus-functionality is provided. Exactly this autofocus functionality represents the most absorbing part of the laboratory exercise. When pushing the button on the graphical user interface a subroutine with autofocus algorithm will be executed. This algorithm has to be designed, developed and uploaded before by the student itself using a pre-defined template which can be edited within the LabVIEW development environment.

To prove the result and the quality of the algorithm it can be chosen through different analysis types of the image like:

- Original image
- Edge detection of image
- FFT of image
- Or simply the standard deviation

With the help of the different analysis methods it should be an easy task for the students to prove the quality of their autofocus algorithm and to become motivated to tune and improve their algorithm.

III. THE REMOTE LABORATORY

A. Hardware

A visual presenter from the company ELMO (HV-5500XG [5]) illustrates the main device used for this online laboratory. This presenter uses a 1/3 progressive scan CCD to acquire the image and offers an analog RGB-output to communicate with the PC. Furthermore the serial port (RS 232) can be used to control the main functionalities like zoom, focus and to switch through different illuminations. To acquire the images on the laboratory server a PCI card from National Instruments [6] is used. (NI PCI - 1411). Software

As central development software LabVIEW 8.6 [6] is used. LabVIEW (short for Laboratory Virtual Instrumentation Engineering Workbench) is a platform and development environment for a visual programming language from National Instruments [7]. Additionally to the base software framework also some toolkits are used. For instance the software toolkit Vision Development Module provides several libraries for image processing and analysis which can be used to acquire, process, analyze and visualize the raw picture from the camera [8]. Furthermore LabVIEW offers also libraries (VISA) for the communication over the serial port, which makes it a quite simple task to control the device.



Figure 2. ELMO – visual presenter [5]

The front panel is designed within LabVIEW and can be used directly to control the different actions of the laboratory. Via the build-in web server the front panel can be used in combination with the iLab shared architecture as main user interface of the laboratory.

As explained in the previous section next to the control of the device there is the functionality to upload and test a self developed autofocus algorithm. Therefore first of all the use of a template virtual instrument is required, because it has some pre-defined inputs and outputs which have to be considered in the right way so that the autofocus functions remains executable. When the autofocus subroutine within the template is finished it can be uploaded via a php script to the server. The script checks whether the file is a valid SubVI (virtual instrument) – file. Furthermore the generated name of the last uploaded algorithm will be stored separately in a file and based on this value the subroutine is called dynamically from the main application. If the autofocus algorithm is called by pushing the button on the user interface the code a student uploaded before will be executed on the server and the results can be observed directly within the web browser. For the case an autofocus algorithm works not well in the first tries an additional “Stop Autofocus” button was implemented. This functionality is very useful if the algorithm is working incorrectly or when haven’t chosen the right break condition. When a user modifies the template in the wrong way so that the main application cannot execute it anymore a message in the status field informs the user about the current state.

Because of the use of LabVIEW on the client side there is some additional software installation required. The so called runtime engines for LabVIEW and for the Vision toolkit have to be installed in order to use the laboratory from any client. The main user interface can be seen on figure 3.

B. Overall structure

After the technical aspects discussed more detailed in the previous session this session will give you an overview how the laboratory is implemented within the ISA framework.

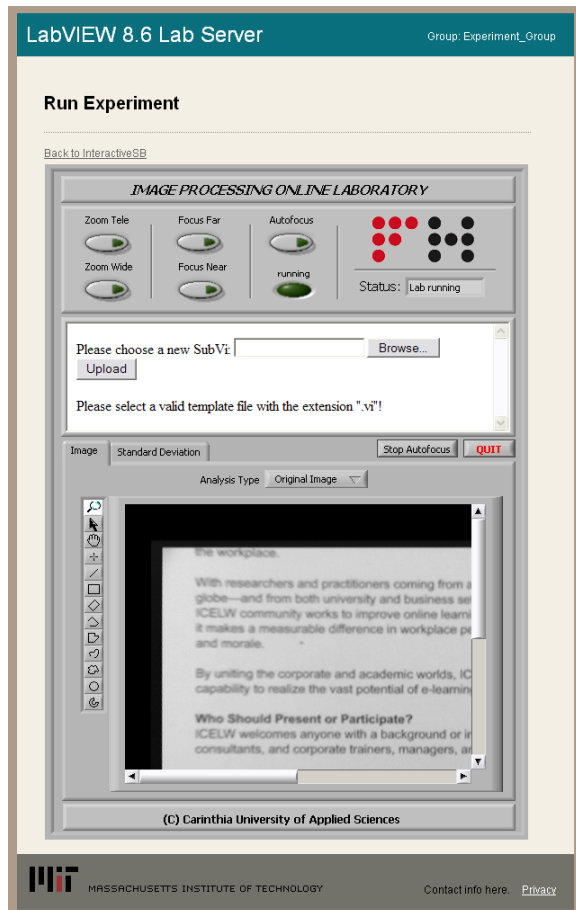


Figure 3. Graphical user interface

As can be seen in figure 4 the whole application can be divided in 5 sections, the client side, the iLab service broker and the iLab lab server, the LabVIEW application and the scheduling services. The service broker is responsible for authentication, reservation and for the maintenance of the lab session with the laboratories behind. Therefore the service broker has to communicate with the scheduling services, because they keep all data of the available time blocks in a laboratory and about user rights and user reservations. The lab server is running apart from the service broker and the authorization to access the lab must be granted through the service broker.

As you may have seen on the block diagram the process starts when someone authenticates on the service broker. All user rights, group memberships and bookings are stored on the user side scheduling server from which the service broker has to retrieve this information. When authentication is successful the user gets access to a list of laboratories (depending on the rights of the specific user) which can be selected from different experiment groups. The user has to select a laboratory, in this case the autofocus laboratory, and a reservation can be requested. The available time slots will be fetched from the Lab Side Scheduling server, which keeps all available time slots, the maximal time frame length and some other details. After selecting a time frame the user has the possibility to redeem the reservation within the reserved time frame. When redeeming a session the ticketing service between the service broker and the lab server is used to check whether the user has a valid reservation at the actual time. In this case the lab server returns a coupon and the

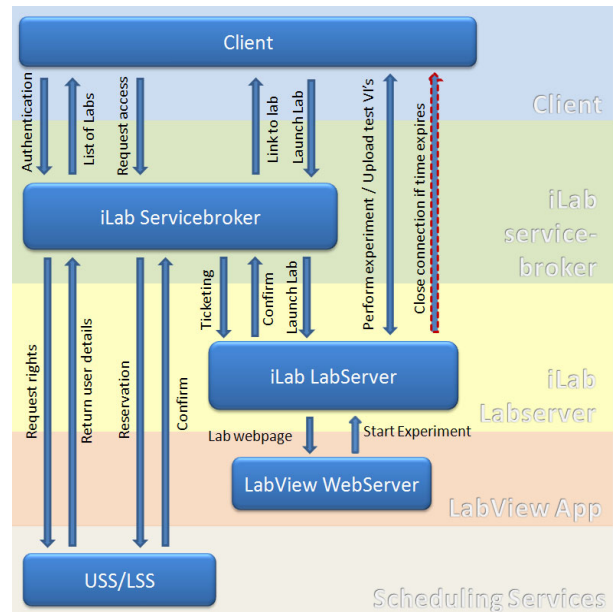


Figure 4. Block diagram of the online laboratory within the ISA

session is redeemed correctly. In this case the direct link to the main interface on the lab server appears and the experimentation can start. In this case, as long as the reservation time window does not expire, directly access is granted to the lab server and the remaining time will be displayed in the status bar of the web browser. During this stage the ASP engine will execute the LabVIEW main application and the front panel which is published with the LabVIEW web server will be embedded as object into the html page of the interactive lab server. When the session expires a message appears and the connection will be closed so that the user with the next reservation can access the laboratory.

IV. PLANNED EXTENSIONS

To increase the operability and the difficulty of the laboratory some extensions are currently under development. First tests of the online laboratory reflected that the task can be performed really good, but additionally it would be interesting to test the autofocus algorithm on different patterns with maybe different conditions. This would foster the students to adapt their developed autofocus algorithm to different situations and to reflect problems depending on different factors in their resulting laboratory report.

To change the pattern for example two paper rolls can be used in combination with a motor (see figure 5). Different patterns could be printed on the paper roll and by controlling the rolls with a motor the pattern used to test your autofocus algorithm can be changed easily.

A further interesting investigation would be to offer the possibility to change the lighting conditions and maybe to integrate also a light as a disturbance source. This helps to demonstrate many side effects and their influence on the result of the experiment.

With these extensions the laboratory could be divided into two classes, one for beginner level and another for experienced users in the image processing field. This leads to a categorization of the experiment and to a better acceptance from different users with different background.

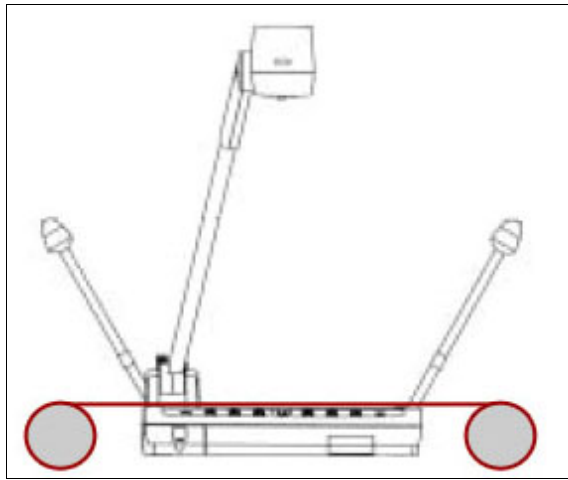


Figure 5. Planned extension to change the pattern [5]

V. CONCLUSION

Image processing and image analysis is nowadays widely used in various applications all over the world and became an irreplaceable method to solve certain problems. Important circumstances of the increasing circulation of image processing applications are the falling prices of cameras and numerous advantages in using them as “universal sensors”. With modern software tools the development of applications becomes easier and more effective [8].

Because of the importance of image processing it is consequential to foster the education within this science field with new interesting learning materials and methodologies. The presented online laboratory contributes with a highly flexible framework in terms of time and place of the learner to foster the interests in studying and experimenting within this science field. Combined with the

iLab shared architecture (ISA) the experiment gets integrated into a consistent access framework where users can reserve and run autonomously their experiments of interest.

REFERENCES

- [1] James E. Corter, Jeffrey V. Nickerson, Sven K. Esche, Constantin Chassapis, “Remote Versus Hands-On Labs: A Comparative Study”, 34th ASEE/IEEE Frontiers in Education Conference, Session F1G, 20–23 October 2004, Savannah, GA, USA.
- [2] Qin Shuren, Bo Lin and Liu Xiaofeng, “Development of the Networked Virtual Instrument Lab for Vibration Measuring Based on Microsoft.Net”, Instrumentation and Measurement Technology Conference, IMTC 2004, Como, Italy, 18-20 May 2004.
- [3] MIT iLab URL: <https://wikis.mit.edu/confluence/display/ILAB2/Home> - Last visited in 03/2010.
- [4] J. Harward, J. A del Alamo, iLab: A Scalable Architecture for Sharing Online Experiments, ICEE 2004
- [5] ELMO VISUAL PRESENTER HV-5500 XG, Handbook
- [6] URL: <http://www.ni.com>. last visited 10.03.2010
- [7] URL: <http://en.wikipedia.org/wiki/LabVIEW> - last visited 10/03/2010
- [8] Image Processing with LabVIEW and IMAQ Vision. (National Instruments Virtual Instrumentation Series), Thomas Klinger

AUTHORS

M. Niederstätter is with System Engineering Department, Carinthia University of Applied Sciences, Villach, Austria (e-mail: M.Niederstaetter@fh-kaernten.at).

T. Klinger is head of System Engineering Department, Carinthia University of Applied Sciences, Villach, Austria (e-mail: T.Klinger@fh-kaernten.at).

D. G. Zutin is a senior researcher at the Carinthia University of Applied Sciences, Villach, Kaernten A-9500 Austria (e-mail: D.GarbiZutin@fh-kaernten.at).

This work was supported in part by Carinthia University of Applied Sciences - Austria. Manuscript received March 26, 2010. Published as resubmitted by the authors 25 April 2010.