IMAGE ANALYSIS AND COMPUTER VISION FOR UNDERGRADUATES

Andrea Cavallaro

Multimedia and Vision Laboratory
Queen Mary, University of London
Mile End Road, London E1 4NS (United Kingdom)
Email: andrea.cavallaro@elec.qmul.ac.uk

ABSTRACT

Real hands-on experience can help students gain a better understanding of theoretical problems in image analysis and computer vision and allows them to put in practice and improve their knowledge in digital signal processing, mathematics, statistics, perception and psychophysics. However, important efforts are necessary to enable students to develop a computer vision application because of the lack of extensively tested and well documented software platforms.

In this paper, we describe our experience with an open source library addressed to researchers and developers in computer vision, the OpenCV library, its limits when used by students, and how we adapted it for teaching purposes by producing a set of appropriate tutorials. These tutorials help the students reduce the average time for installation and setup from 1 week to 4 hours and help them design an end-to-end image analysis and computer vision project. Finally, we discuss our experience of using this framework for undergraduate as well postgraduate student projects.

1. INTRODUCTION

Video-based systems are becoming an important part of many applications around us, such as automated video surveillance, video-based human machine interfaces, and immersive gaming. Such a development has created a market demand for students trained in this area and in fact an increasing number of universities are proposing courses in image analysis and computer vision.

Although it is easy to generate interest in students while teaching these subjects, it is much more difficult to help them have a practical experience in the design of a computer vision application by means of a project. Computer vision projects allow students to put in practice and develop their knowledge in mathematics, statistics, digital signal processing, perception and psychophysics. Although it is possible to implement MATLAB laboratory sessions for low-level image analysis functions, it is much more difficult to allow the students to design and to implement a real-world, real-time application such as object detection and recognition, video-based human-computer interfaces, mixed reality, or object tracking.

We experienced some problems in letting students start an undergraduate computer vision project from scratch or with software borrowed from our research projects: it took them too long to get started, thus reducing the time available to really learn about computer vision. The ideal solution to this problem would have been a well commented software platform and a library providing all the basic data structures and their operators, basic functions or classes for reading data from a video camera or from a file, displaying a video or writing a video file, functions for computing the optical flow, filters for tracking and edge detection, and so on, as well as more advanced functionalities to demonstrate the capabilities of the library to the students. The Intel OpenCV library [1] appeared to be the best candidate for such a platform. However, we realized that the software was not adequately commented and documented for our purpose (the library is in fact addressed to researchers and professionals, not to undergraduate students) and the time taken by the average student to correctly install the software package, additional software required, and to understand how to use it was still prohibitively high.

Given the large number of useful functions of the OpenCV library, we decided to improve the documentation in order to enable undergraduate students to use the potentiality of this library for their projects. This resulted in three main documents that we now use to guide the students in the installation of the OpenCV software, the additional packages needed to receive data from a video camera and a tutorial on how to write a simple computer vision application. We will discuss these documents and the way we organize the projects in Section 2. Section 3 reports some examples of projects, whose reports in turn become part of the library documentation. In Section 4 we evaluate our experience with the platform. Finally, in Section 5 we draw the conclusions.

2. STUDENTS' PROJECTS WITH OPENCV

We describe here the structure of the image analysis and computer vision project module and the documents containing tutorials that made the OpenCV library accessible to undergraduate students. The objective of this module is a design, development, or research project during which students produce a real-time application. The project enables students to appreciate the components of a computer vision system, to develop their skills in signal processing and in C and C++ programming. In addition to this, the module gives students experience of managing their own time to complete a project and of developing their communication skills, both written and oral, to a standard expected by industry of a new graduate. The outputs of the project are a report, a demonstration, a presentation, and an oral examination. The project normally occupies about 240 hours, spread over 2 semesters, and can take the form of an individual or a group project. In the case of a group project, the aim is also to give students experience in working as a team, communicate among each others and manage a more complex project. In fact, individual projects usually start from the information available in the tutorials, whereas group projects (in general 6 people) start from already existing projects and build up more complex applications. In such a way, students also appreciate the importance of good commented and documented software.

Students are first given an introductory lecture on the objective of the project and the OpenCV library. Then they are given three tutorials produced to help them start. The first tutorial is the installation guide, which helps install the OpenCV library for use with Visual Studio and Direct X SDK. The second is a guide to creating the workspace and project for OpenCV. The final guide shows how to write a real-time application, a background subtraction algorithm using the library and a web camera. Thanks to the three tutorials, the average undergraduate student is able to have his/her first real-time computer vision program running in about 4 hours. In addition to the tutorial and the OpenCV documentation, students have also access to previous project reports which complete and enrich the basic documentation of the library.

3. PROJECT EXAMPLES

In this section we describe and discuss examples of projects developed by the undergraduate students based on the framework described in Section 2. The projects are based on real-world applications, such as video surveillance, video production (e.g., weather forecast programs with the presenter superimposed on a map), interactive and immersive games.

The first example of project is object detection based on background subtraction. *Background subtraction* allows one to detect moving objects using a static camera. This is a simple and effective example of real-time application where students appreciate and learn about problems caused by different lighting conditions, camera noise, and shadows. Starting from a simple thresholded pixel-by-pixel image difference (Figure 1), the students understand how to use morphological and low-pass filters [2,3], re-sampling and statistical analysis [4]. In addition to the above, they learn how to use statistical approaches for the generation of a background frame.

The completion of the previous example enables the students to develop other applications, such as *object tracking* (Figure 2). Working on object tracking, students understand the importance of feature representations and of color space conversions, the use of distance functions in the feature space, and different tracking algorithms. Furthermore, they learn how to disambiguate between similar objects by choosing the appropriate set of features to describe them. In particular, they experience several histogram comparison functions, such as the χ^2, the correlation methods, and the Bhattacharyya distance. They learn and test the difference between color spaces and the properties of photometric invariant color features [5].

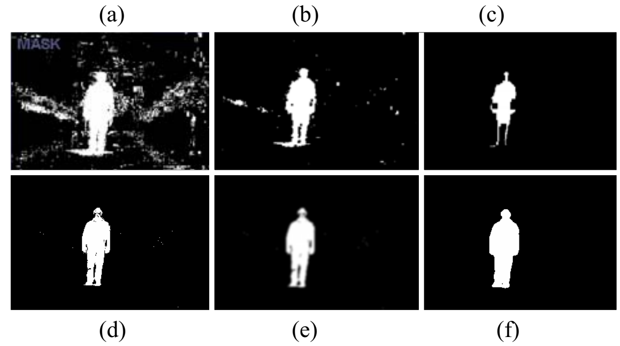


Figure 1. Example of thresholded background subtraction result (a) undergoing different morphological filtering: (b) single erosion, (c) recursive erosion. (d) After down-sampling, (e) after down-sampling followed by up-sampling. (f) After statistical analysis.

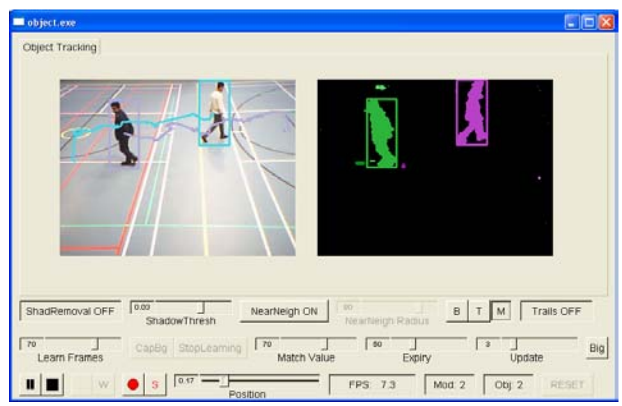


Figure 2. Example of object detection and tracking based on background subtraction and color histogram distance.

This allows them to design algorithms that differentiate objects from their shadows.

Another application enabled by the results of background subtraction is *mixed reality* (Figure 3). This application allows students to understand how to write algorithms for video production and how to solve different problems related to ambient lighting and to the composition of several inputs.

Other projects that do not necessarily require the use of background subtraction are based on the design of *perceptual human-computer interfaces* (Figure 4). Figure 4 (a) shows an example of a *remote control* simulated by tracking hand gestures. Here, the tracking is based on a model of the hand defined by color and edge information. Moreover, face detection [6] is used to disambiguate between the hands and the face. The project shown in Figure 4 (b), called the *virtual artist*, allows one to draw lines on the screen without using any device. In both applications, students learn how to define and use a model of an object and how to track it over time. A particular aspect covered when developing these applications is the analysis of the

(a) (b)

(c) (d)

Figure 3 Example of real-time mixed reality application. (a) Background frame learned and updated over time; (b) current frame; (c) new background frame (synthetic or real); (d) object-based scene composition.

precision of the tracking. Students evaluate the results in terms of user satisfaction and in terms of objective metrics (accuracy).

We conclude this section presenting two applications that are usually very motivating for students and enable them to use their creativity, namely the design of special visual effects and of video animations. *Special visual effects* can be generated based on background subtraction and on the use of video object memory (Figure 5 (a)). *Video animation* is based on tracking: moving objects can control the movement of avatars or simple symbols (Figure 5 (b)).

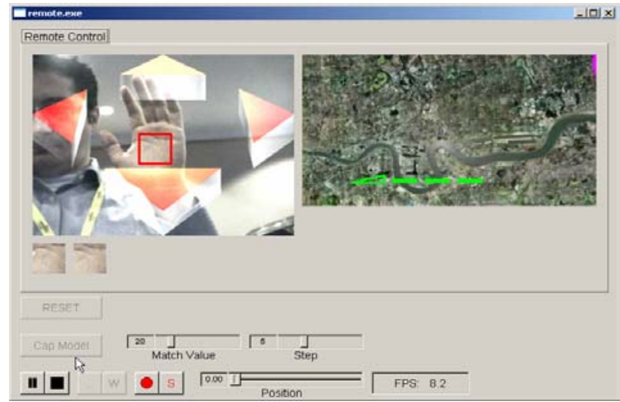
4. EVALUATION AND ASSESSMENT

4.1. Learning outcomes

Learning outcomes for the image analysis and computer vision projects can be divided into two groups, namely subject specific skills and transferable skills. In terms of learning outcomes in subject specific skills, students learn to employ signal processing, mathematical and software 'tools' to a familiar or unfamiliar situation. In particular, we experienced that the opportunity to develop ‘good looking’ applications, which hide a good amount of theoretical studies and concepts, helped motivating the students. Interestingly, this aspect also motivated some of the students to go far beyond their project specifications and to learn more on the subject. Furthermore, the possibility of developing real-time computer vision applications seems to be very rewarding for the students and at the same time allows them to have fun (see examples in Section 3).

In terms of learning outcomes in transferable skills, students learn to manage time effectively and produce written progress reports and a final report on time. In addition to the

(a)



(b)

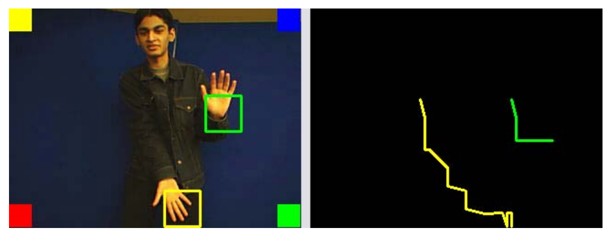


Figure 4 Examples of perceptual human-computer interaction. (a) Video-based remote control: a vertical movement of the hand changes channel (Up: next channel, down: previous channel; Left (right) decrease (increase) the volume). (b) Virtual artist: with the movement of the hands the student can draw in different colors that can be selected in the corners of the image.

experience with signal processing and computer vision, the projects allow the students to appreciate the importance of writing software that can be easily reused. Students working in group build their work on top of other students’ work and therefore appreciate the problems one encounters when software is not well commented and documented.

4.1. Assessment

OpenCV is becoming widely used by the research community. Since its release as freeware, the OpenCV library has been downloaded over half a million times and the official *yahoo* group has over 5000 members. However, when we joined the group, we found no answers to our queries and we received ourselves a large number of questions. One of the reasons for this is that the OpenCV library is aimed at users with an in-depth understanding of computer vision. Furthermore, there is little documentary support for the functions.

The creation of the tutorials and the use of students’ report as additional documentation to the library facilitate the use of OpenCV. Although students find it hard to start their image analysis and computer vision project, they usually get

(a)

(b)

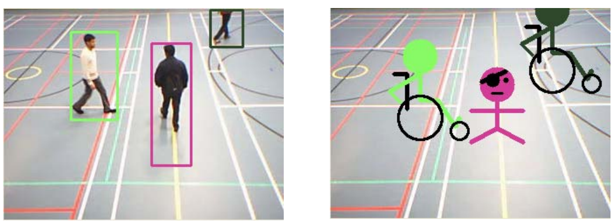


Figure 5 (a) Example of special visual effects generated with background subtraction and the use of object memory. (b) Example of object animation based on tracking. The motion of the tracked people (left) controls the movement of symbols (right)

enthusiastic when they succeed in showing to their peers what they were able to design and with a common personal computer and a web camera. This in turn generates interest for the subject in other students and could also be used as part of universities strategies for widening participation. For example, a group of students found it very rewarding to present their work at the University Open Day and their demos had a very good appeal to high school students.

An important advantage of adopting OpenCV for teaching purposes is that it is open source. This not only reduces costs but also allows the increasing number of students with their own computer and a web cam to work at home and to continue using the library after the project.

In addition to the above, the platform is very useful for providing the students with practical examples in class during the lectures to support the theoretical part of the course.

To conclude, we report here some feedbacks received from the students after completing their projects:

"This project had the "excitement" factor, being able to work with live camera feeds and manipulating the input to achieve desired effects as well as for fill our goals."

"I feel a little extra background information on OpenCV could have helped us achieve a little more."

"Although the project was a learning curve for me it was also a fun and enjoyable at the same time."

"I think it was well worth the effort, and we were all very pleased with what we had produced."

5. CONCLUSIONS

Laboratory hands-on experiments and projects are a very effective way to learn subjects such as signal processing and computer vision. We presented the framework that we developed and we use to enhance the quality of learning in image analysis and computer vision at undergraduate level based on the OpenCV library. Providing additional documentation to the library and project examples opened a window of opportunity for student projects that was not available at an undergraduate level.

All the projects produce a user guide and functional documentation in order to complement and enhance the functions available in the OpenCV library. The framework and the documentation are continuously updated with new projects and additional tutorials are provided as the number of applications increases.

6. ACKNOWLEDGMENTS

We would like to acknowledge the effort of the students Navin Kerai, Bhavin Padhiar, Lad Ketanbhai, Randip Singh Bahra, Muhammad Razwan Aslam, and Khalid Saeed Allahawala who helped set up the framework described in this paper.

7. REFERENCES

[1] G. Bradski, "The OpenCV Library", *Dr. Dobb's Journal November 2000, Computer Security*, 2000.

[2] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1988.

[3] A. Bovik, *Handbook of Image and Video Processing*, Academic Press, 2000.

[4] A. Cavallaro, T. Ebrahimi, "Interaction between high-level and low-level image analysis for semantic video object extraction", *Journal of Applied Signal Processing*, No. 6, pp. 786-797 June 2004.

[5] E. Salvador, A. Cavallaro, T. Ebrahimi, "Shadow identification and classification using invariant color models", in Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Salt Lake City (Utah-USA), pp 1545-1548, 2001.

[6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", in Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2001.