COMPUTER VISION IN PARALLEL COMPUTING - REDUCING COMMUNICATION TIME

Michal Podpora
Doctoral Degree Programme, Opole University of Technology
E-mail: ravyr@klub.chip.pl

Supervised by: Jan Sadecki
E-mail: jsad@po.opole.pl

ABSTRACT

In this paper author considers using a cluster in a computer vision project, he explains some ideas about minimisation of communication time between nodes of a cluster in a specific approach.

1. INTRODUCTION

Sight is one of the most important senses for almost every being on our planet. Although we already know how retina works and more or less how the brain identifies the visual "signal" [2], our knowledge is still far too insufficient to build a complete virtual model.

Nowadays, computers are becoming more and more powerful, but a single PC seems to be still not enough for all the real-time processing. Video capturing, converting and filtering at a minimal frame rate of e.g. 20 frames per second take so much computation time that creating a substantial artificial intelligence algorithm becomes a very difficult programming challenge.

2. PARALLEL COMPUTING

Despite the fact that parallel computing provides much more CPU power, scientists and hobbyists involved in computer vision seem to avoid parallel computing. This tendency is fully justified, because of a huge problem with communication time. It is not possible for a parallel algorithm to be more efficient than a sequential one if the communication between nodes of a cluster is based on passing video frames or transmitting real-time video stream.

In comparison with the human brain, most computers are definitely sequential machines. On a serial computer parallelism could be simulated [4] to perform some incoherent computations, but it is still a single, sequential machine.

2.1. CAN PARALLEL COMPUTING IN COMPUTER VISION BE FASTER THAN SEQUENTIAL?

In my opinion yes, but we should stick to following ideas:

- Video capturing is processed in real-time only on the input node of the cluster. Other nodes should not be synchronised with the input node.

- Use of a synchronous algorithm [1] for the communication causes sequential dependencies between the processes, so it would be better to use asynchronous algorithm.
- Minimisation of amount of transmitted data must be made – transmitting video stream or sending separate video frames is unthinkable. (There are better ways of achieving this goal than image compression).
- The algorithm should divide the whole task between available nodes "wisely", i.e. not only taking into consideration load balancing, but also the virtual architecture of the application.

2.2. COMMUNICATION TIME IS THE KEY

Let us focus on the problem of minimisation of the communication time. In my example application the input node was powerful enough to cope with real-time video capturing (1), some simple operations on frames like e.g. comparison of two subsequent frames (1) and finally preparation of output data for other nodes.

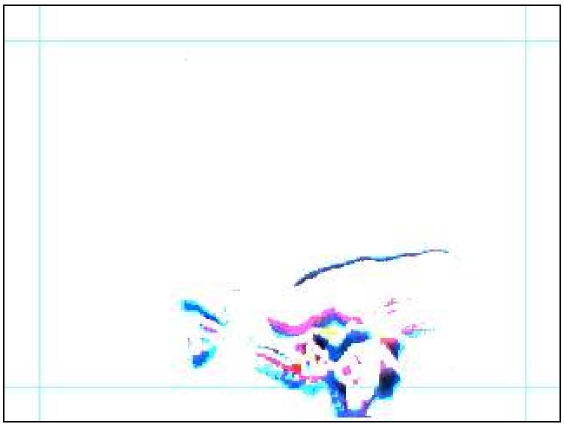

Figure 1: An input frame (on the left) and filtered difference between two frames [5] (on the right).

Although the application is not complete yet (and neither is the minimisation algorithm), there are some interesting results. All nodes of a cluster (except the input node) contain artificial neural networks "specialised" to distinguish a specified feature like colour, brightness or movement. These neural networks do not require a series of full-size, full-colour frames, and that's where some minimisation can be made.

Neural networks send their "decisions" back to the input node asynchronically and the input node defines the "object of interest" of neural networks. The coordinates of the centre of this object are called a "virtual yellow spot" (analogous to the *fovea centralis* – a small depression in retina providing the most detailed information [2]). This "virtual yellow spot" is marked with a cross in fig. (2).

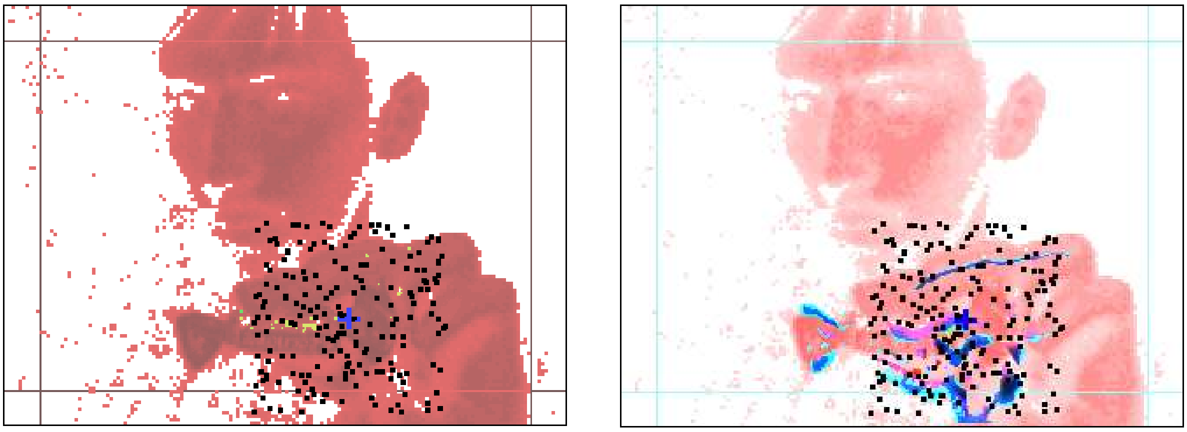

Figure 2: The candy I'm holding has obviously attracted the networks' attention.

Once the "virtual yellow spot" is defined, it can be used by all neural networks in nodes to specify which part of the scene contains the "object of interest".

3. PROMISING DIRECTIONS AND FUTURE WORK

The application is not finished yet, but I already know that my idea of reducing communication time is much more efficient than e.g. compression of video stream. In previous implementations I was either using a single PC or two computers in client-server architecture but results were not satisfactory. Real-time video capturing and processing concluded in time-critical code sections of applications. Moreover, the greater part of the code was indeed a sequence of operations. An application processing the input video stream, with user interface, supporting all essential functions and options, was so complex that there was not enough CPU time left (for the "AI part") before the next frame. This kind of problem does not exist in parallel computing – the algorithm doesn't have to be sequential or even synchronous. An asynchronous algorithm and a judicious cluster topology create brand new possibilities for a programmer.

The most important impediment is the communication time between nodes. I hope that my ideas about reducing it are sufficient. I am currently writing procedures to measure the communication time of this specific application:

- running it on a single PC / on a cluster,
- transmitting video stream / passing video frames / passing selected data in the neighbourhood of the yellow spot.

Afterwards I am going to write an example application to supply documentary evidence and present the possibilities of employing my idea.

4. SUMMARY

Although computer vision is a very popular issue [3], it is still mostly terra incognita. In my opinion parallel computing offers new possibilities. Also, new possibilities for myself – I am currently working towards the Ph.D. degree and I am planning to continue exploring this issue for at least three years.

REFERENCES

[1] Karbowski, A., Niewiadomska-Szynkiewicz, E., Obliczenia Równoległe i Rozproszone, Oficyna Wydawnicza Politechniki Warszawskiej (2001)

[2] Hecht, E.: Optics, Addison Wesley, 4th edition (2002)

[3] Fisher, R.B., CVonline – Compendium of Computer Vision, available: http://homepages.inf.ed.ac.uk/rbf/CVonline/all.htm (2006)

[4] Ballard, D., Brown, C., Computer Vision, Prentice Hall Professional Technical Reference (1982)

[5] Mather, G., Introduction to Motion Perception, available: http://www.lifesci.sussex.ac.uk/home/George_Mather/Motion/ (2006)