11. Image Processing Techniques for Distributed Grid Applications

Piotr Brudło

Gdansk University of Technology,
Faculty of Electronics, Telecommunication and Informatics,
Computer System Architecture Department
e-mail: pebrd@eti.pg.gda.pl

Abstract

In this chapter, parallel approaches to 2D and 3D convolution processing of series of images have been presented. A distributed, practically oriented, 2D spatial convolution scheme has been elaborated and extended into the temporal domain. Complexity of the scheme has been determined and analysed with respect to coefficients in convolution kernels. Possibilities of parallelisation of the convolution operations have been analysed and the results presented. Serial and parallel variants of 2D convolution schemes are proposed and their time-cost trade-offs are discussed. Flexibility of the solution with regard to scalable size of the kernel has been highlighted. The image processing techniques are analysed with respect to be applied in active distributed grid processing systems in the Internet, and their direct orientation toward the Comcute system has been deliberatively spotlighted.

11.1. Introduction

Spatial as well as temporal convolutions are important techniques for video and image processing. In general, spatial convolution is performed on an input image, and then is being extended into the temporal domain for series of consecutive input images. Among various applications of the convolution, one can mention image filtering or noise processing, object detection and identification, pattern recognition, and many others. Typical input data are series of 2D colour or grey-scale pictures sequentially delivered from a video camera or another on-line device, or off-line video source like video DVD player. In the case of on-line processing, the main challenge is to complete processing of a currently convolved image before the delivery of a consecutive picture from an input device. Real-time requirements of 30 or more pictures per second with HD parameters practically direct (confine) design choices either to high performance computing implementations or distributed Internet grid or cloud systems [1], [2]. In the chapter, direct orientation of the proposed processing toward the Comcute system is spotlighted [3], [4]. Since potential computational power the Comcute system satisfies required efficiency, this approach is practically applicable and reasonable [5], [6].

11.2. Mathematical foundations

Formally, n-dimensional convolution is defined by generalisation of (n-l)-dimensional convolution into the n-th dimension [7], [8], [9]. One-dimensional convolution can be defined as is given in formula (1).

$$O[n] = \sum_{i=1}^{r} K[i] \cdot I[n-i] \qquad (1)$$

where:

O - 1D vector of the output data,

K - 1D kernel of length r,

I - 1D vector of input data.

By generalisation into 2D we obtain two-dimensional convolution – formula (2).

$$O[n_1, n_2] = \sum_{j=1}^{r_2} \sum_{i=1}^{r_1} K[i,j] \cdot I[n_1 - i, n_2 - j] \qquad (2)$$

where:

O - 2D matrix of output data,

K - 2D kernel of size r1 x r2,

I - 2D matrix of input data.

In particular, input image I is usually represented by matrix N_1 x N_2 (e.g. 800 x 600) of colour or grey-scale pixels. Practically applicable 2D kernels have dimensions of r_1 by r_2, where r_1 and r_2 are of size up-to 17-19. If one does not consider border conditions, output image O gets the size of N_1-r_1+1 by N_2-r_2+1. The complete formula for 2D convolution of an N_1 x N_2 image is given in (3) and the expression for an individual pixel O[x,y] is given in formula (4). Indices in formulas (3) and (4) have been scaled to start from 1. The indices are in ranges of $<l..r_1, 1..r_2>$, $<l..N_1, l..N_2>$ and $<l..N_1-r_1+l, 1..N_2-r_2+l>$ for kernel K, input image I and output image O, respectively.

$$O = \left\{ O[x,y] \;\middle|\; x \in \langle 1, N_1 - r_1 + 1 \rangle,\ y \in \langle 1, N_2 - r_2 + 1 \rangle \right\} \qquad (3)$$

$$O[x,y] = \sum_{j=1}^{r_2}\sum_{i=1}^{r_1} K[i,j]\cdot I[x+i-1, y+j-1] \qquad (4)$$

2D convolution, as an image transformation, has its geometrical interpretation, and is shown in Fig. 11.1. It is noticeable, that the additions according to indices i and j, as well as inner multiplications are independent from one another, and this is the point where parallelisation can potentially be introduced in calculations.

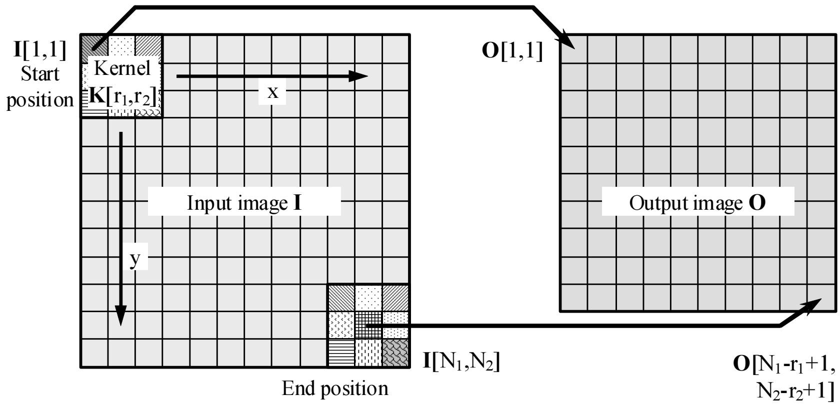

Fig. 11.1. Geometrical interpretation of 2D convolution

In software, formulas (3) and (4) can be implemented directly. In order to calculate convolution this way, numbers of basic mathematical operations (multiplications and additions) equal $(N_1-r_1+l)*(N_2-r_2-l)*r_1r_2$ and $(N_1-r_1+l)*(N_2-r_2-l)*(r_1r_2,-1)$, respectively. In practice, some of kernel coefficients can be zeroes, what produce idle operations. In order to eliminate the idle operations, a different but equivalent formula for 2D convolution is proposed. Kernel decomposition is the result of its interpretation. Since all the additions in (4) are independent, one can transform formulas (3) and (4) into equivalent formulas (5) and (6). The practical implications are that first the complete input image I is multiplied by all kernel coefficients separately, producing r_1r_2 component images $O_{i,j}$ ($i=1..r_1$, $j=1..r_2$). Next, component images $O_{i,j}$ ($i=1..r_1$, $j=1..r_2$) are being added together with the spatial shifts of (r_1-i, r_2-j), respectively. In the proposed scheme, in case a zero coefficient in the kernel is detected, input image I is simply not being multiplied by it. As the result, only significant (not zeroed) component images $O_{i,j}$ are being added (with the spatial shifts of (r_1-i, r_2-j)), finally producing convolved output image O.

$$O_{i,j} = \left\{ O_{i,j}[x, y] = K[i, j]\cdot I[i, j] \;\middle|\; x \in \langle 1, N_1\rangle,\ y \in \langle 1, N_2\rangle \right\} \qquad (5)$$

$$O[x,y]=\sum_{j=1}^{r_2}\sum_{i=1}^{r_1} O_{i,j}[x+r_1-i,\ y+r_2\text{ - }j] \qquad (6)$$

11.3. Practical example of 2D convolution

Convolution techniques may be used in many practical applications of image processing. One can enumerate filtering, object detection, pattern recognition, and broad class of transformations in general. Practical example of convolution use is presented in Fig. 11.2. A picture of a Gripen multi-role fighter [10] (top) has been initially filtered to eliminate unnecessary noises. Next, convolution has been applied in order to detect characteristic edges of the structure (bottom). The final image can be processed further e.g. for feature detection and identification. In general, edge detection is being carried out in three steps: vertical edge detection, horizontal edge detection, and out of the two component results the final image is combined.

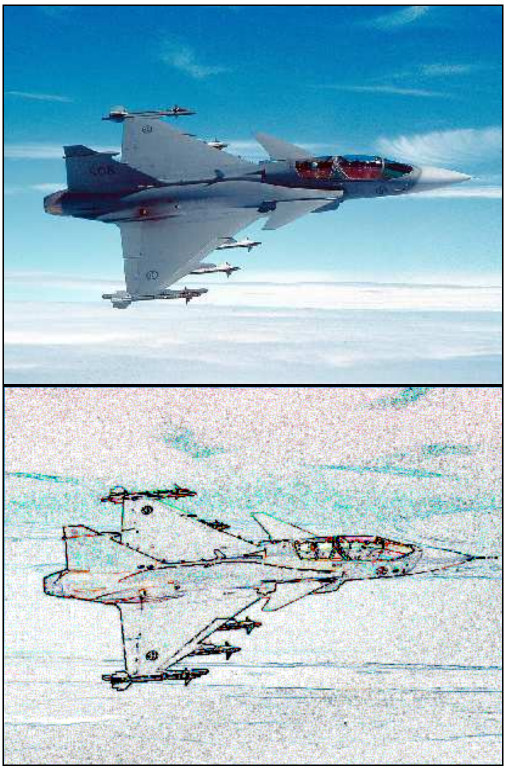

Fig. 11.2. An example of 2D image convolution – edge detection

Two-dimensional convolution technique separately and independently processes individual images (one by one). One can generalise 2D convolution into temporal domain, and obtain spatio-temporal (3D) convolution. Geometric interpretation of 3D convolution has been presented in Fig. 11.3. Using 3D convolution techniques one may perform temporal filtrations like eliminations of selected objects from video transmission, generation of the picture on the base of preceding screenshots, etc. At this stage, however, other advanced image processing techniques have to be used.

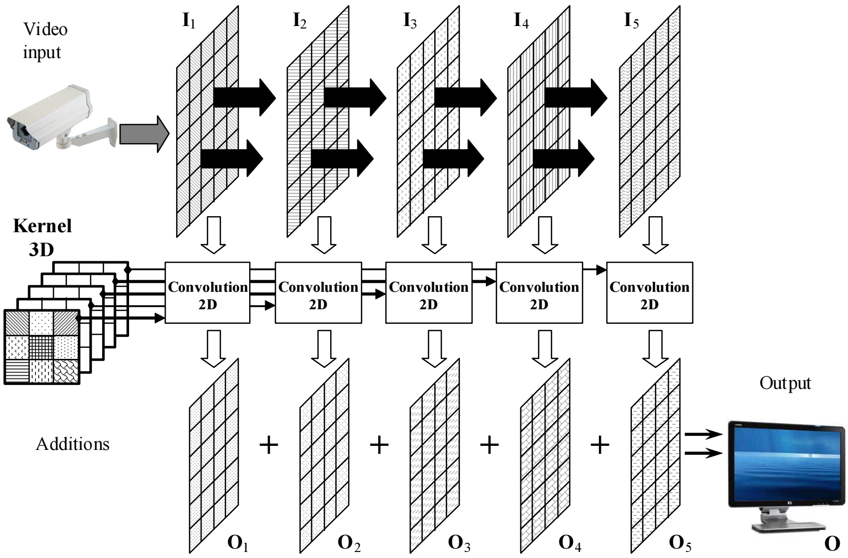

Fig. 11.3. Geometrical interpretation of three-dimensional convolution (spatio-temporal)

11.4. Applications

An example of practical application of the image processing techniques (convolution) is presented in Fig. 11.4. In the example, video input has been taken from a recording stored in mass memory (hard disk). A pattern of a car has been looked for. The pattern has been correlated (convolved) with consecutive video images. At the moment the car has appeared, the correlation function has given significant maximum response. As long as the car has been moving along the screen (along the highway) the correlation maximum has closely followed the pattern of the car. Consecutive maxima have been connected by a white curve to show the trace of the car. In the example, the video film has been recorded by a stationary stable camera.

Fig. 11.4. Tracing of a car pattern on recording from a stationary stable camera

Image processing techniques find their applications in many fields. Security is one of the most representative [11]. Potentially dangerous object or person, hazardous or unexpected behaviours or situations can be automatically detected and, to some extent, identified by applications of this kind, and next security guards can be alerted. In Fig. 11.5, an example of smoke detection has been presented. The pattern of smoke has been prepared earlier. The place on the image at what pattern matching gives the maximum response has been marked by a rectangle (Fig. 5). As the movie has been progressing, the rectangle has followed the place of most superior pattern matching. In general, convolution allows detection of the area of the smoke. In this case, one should analyse the places on the image at what pattern matching response exceeds adopted correlation value. In the presented solution, one can notice that algorithmic (convolution) approach to image processing possesses potentialities to be effectively applied in video monitoring systems of practically any kind. However, one should remember that the required mass computations can be effectively performed on processing systems like HPC or distributed grids in the Internet (e.g. the Comcute system).

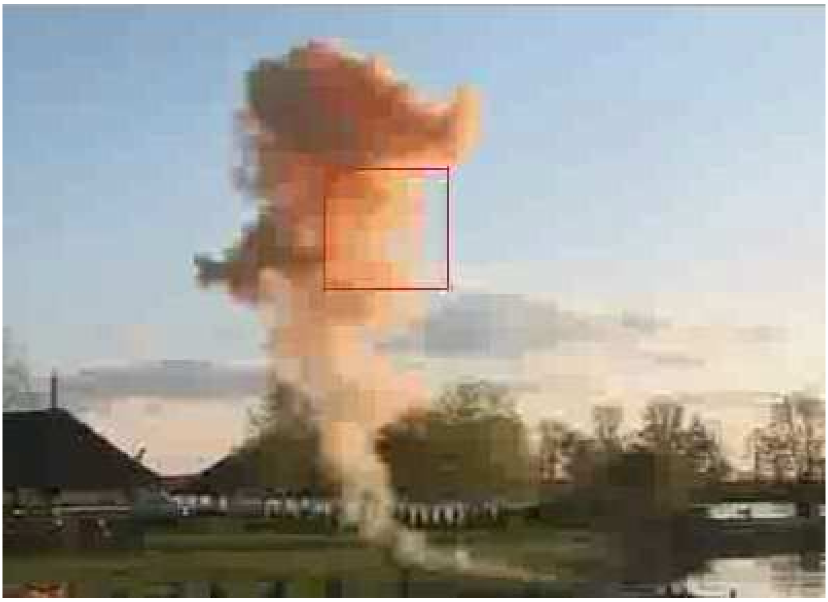

Fig. 11.5. Detection and identification of smoke – monitoring application

11.5. Conclusions and results

Convolution scheme oriented for practical implementations has been elaborated and presented. Two-dimensional and three-dimensional convolutions have been described and analysed in detail. Practical applications of two-dimensional convolutions have been illustrated. Convolution as the approach for detection and identification of selected patterns on dynamic images has been verified. The focus has been put onto active monitoring for security applications. Two instances - detection of a car and detection of smoke – have been exemplified. Potential development directions of this kind of applications have been highlighted. Flexibility and openness of the solution are deliberatively spotlighted. Practical importance and meaning of automated active monitoring have been discussed.

Computational issues of mass processing for image convolution are addressed. The scale of processing requirements directs practical implementations towards two solutions: to high performance computers (supercomputing) on one hand, and to distributed grid processing systems in the Internet on the other. Direct orientation of the solution to the Comcute system is suggested as reasonable choice in efficiency and cost aspects. Scalability of convolution image processing techniques is innately compatible with inner structure of Comcute processing paradigm, and this may constitute a solid ground of the success of practical implementations in the active monitoring field.

References

1. Brudło P.: Berkeley Open Infrastructure for Network Computing, Chapter in monograph: "Distributed Calculations in Computer Systems of Grid Class Architecture", Publisher: Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Gdańsk, Poland, 2012, pp. 39-45.
2. Balicki J., Brudło P., Czarnul P., Kuchta J., Matuszek M., Szpryngier P., Szymański J.: *Functional Design of the Comcute System*, R&D technical report 33/2011, Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Gdansk, Poland, 2011.
3. Brudło P., Kuchta J., Szpryngier P., Szymański J.: *Requirements for Implementation of Selected Computational Methods for the Comcute System*, R&D technical report 36/2011, Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Gdansk, Poland, 2011.
4. Brudło P., Czarnul P., Kuchta J., Szpryngier P., Szymański J.: *Characteristics of Distributed Computations for the Comcute System,* R&D technical report 41/2011, Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Gdansk, Poland, 2011.
5. Balicki J., Bieliński T., Brudło P., Paluszak J., Szymański J: *Dissemination of Computations with Distribution Servers*, Chapter in monograph: „Distributed Calculations in Computer Systems of Grid Class Architecture", Publisher: Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Gdańsk, Poland, 2012, pp. 113-126.
6. Brudło P.: *Implementation Issues in the Comcute System*, Chapter in monograph: "Distributed Calculations in Computer Systems of Grid Class Architecture", Publisher: Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Gdańsk, Poland, 2012, pp. 127-136.
7. Jahne B.: Spatio-Temporal Image Processing. Theory and Scientific Applications, Springer Verlag, Heidelberg, ISBN 3-540-57418-2.
8. Brudło P., Kuchcinski K.*: Parallel Spatio-Temporal Convolution Scheme Oriented for Hardware Real-Time Implementation*, Proc. 23rd EUROMICRO Conf.: New Frontiers of Information Technology, Budapest, Hungary, September 1997, Short Contributions, pp. 190-195.
9. Granlund G., Knutsson H.: *Signal Processing for Computer Vision,* Kluwer Academic Press, ISBN 0-7923-9530-1.
10. Picture of Gripen multi-role fighter: http://www.gripen.saab.se/image_gallery/
11. Brudło P.: Security in Monitoring, Chapter in monograph: "Distributed Calculations in Computer Systems of Grid Class Architecture", Publisher: Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Gdańsk, Poland, 2012, pp. 175-184