Fast 2D Convolutions and Cross-Correlations Using Scalable Architectures

Cesar Carranza*, Daniel Llamocca†, and Marios Pattichis‡

*Sección Electricidad y Electrónica, Pontificia Universidad Católica del Perú, Lima-32, Perú
†Electrical and Computer Engineering Department, Oakland University, Rochester, MI, 48309, USA
‡Department of Electrical and Computer Engineering, The University of New Mexico, Albuquerque, NM, USA

Emails: acarran@pucp.edu.pe, llamocca@oakland.edu, pattichis@ece.unm.edu

Abstract

The manuscript describes fast and scalable architectures and associated algorithms for computing convolutions and cross-correlations. The basic idea is to map 2D convolutions and cross-correlations to a collection of 1D convolutions and cross-correlations in the transform domain. This is accomplished through the use of the Discrete Periodic Radon Transform (DPRIT) for general kernels and the use of SVD-LU decompositions for low-rank kernels. The approach uses scalable architectures that can be fitted into modern FPGA and Zynq-SOC devices. Based on different types of available resources, for $P \times P$ blocks, 2D convolutions and cross-correlations can be computed in just $O(P^2)$ clock cycles up to $O(P^3)$ clock cycles. Thus, there is a trade-off between performance and required numbers and types of resources. We provide implementations of the proposed architectures using modern reprogrammable devices (Virtex-7 and Zynq-SOC). Based on the amounts and types of required resources, we show that the proposed approaches significantly outperform current methods.

Background

Fig. 1: 2-D Linear convolution using the Discrete Periodic Radon Transform and 1-D Circular convolutions

Proposed Methods

Methods

Conclusion

Fig. 2: Architecture for computing the 1D circular convolution.

Fig. 4: Family of fast and scalable architectures for $N = 127 (N = 128$ for FFTr2) in terms of Running time versus the required number of 1-bit additions. Similar plots are obtained for other resources.

Fig. 5: Performance comparison between SliWin (for references), FastConv and FastScaleConv. To measure performance, we consider the number of Frames Per Second (FPS) to perform the convolution between an image of 4096x4096 and a kernel of size 19 x 19.

References
