

Accelerating statistical texture analysis with an FPGA-DSP hybrid architecture

Ibarra Picó, F.; Cuenca Asensi, S.; Córcoles, V.

Universidad de Alicante. Depto. de Tecnología Informática y Computación. Alicante. Spain
{ibarra, sergio, corcoles}@dtic.ua.es

Abstract.- Nowadays, most image processing systems are implemented using either MMX-optimized software libraries or, when time requirements are limited, expensive high performance DSP-based boards. In this paper we present a texture analysis co-processor concept that permits the efficient hardware implementation of statistical feature extraction, and hardware-software codesign to achieve high-performance low-cost solutions. We propose a hybrid architecture based on FPGA chips, for massive data processing, and digital signal processor (DSP) for floating-point computations. In our preliminary trials with test images, we achieved sufficient performance improvements to handle a wide range of real-time applications.

1. INTRODUCTION

Texture analysis is an important method for image classification and segmentation in a wide range of applications e.g. medical imaging, remote sensing, industrial inspection, etc... [1]. A wide variety of measures have been proposed related to texture properties, among them, statistics measures are widely used in the classification of textured surfaces [1]. Although the performance of such algorithms is usually very good [2], their structure is complex and the process data flow is large. Consequently, the computation cost is high, and for more demanding time requirements, e.g. quality control in high-speed production lines, it is necessary to use specialized hardware and architectures. This customized hardware usually increases the cost (>10K\$), reduces the flexibility and limits the applications of the system. In this work we propose a hybrid architecture based on reconfigurable chips (FPGAs) and digital signal processor (DSP) to achieve high-performance low-cost solutions (<5K\$) in real-time texture classification systems.

2. STATISTICAL TEXTURE ANALISYS

The statistics measures are based on the distributions of single occurrences (first order histograms) or joint occurrences (second order histograms) of pixels features. A careful analysis of the algorithms shows that all of them go through three main stages:

Image Pre-processing (IP): this stage extracts the features (e.g.: gray level, gradient, gray level co-occurrence, gray level difference, etc..) from the pixels of the image. Although pre-processing depends on the specified

algorithm, in all cases this task involves elementary operations on the pixel gray level in a 3x3 neighborhood.

Histogram computation (HC): this stage computes the histogram of the features previously extracted and stores the histogram values in a temporary buffer.

First order histograms: computes the probabilities $P(i)$ of occurrence of each feature in the image, where; $i=0, 1, 2 \dots G$, and G is the number of features. E.g. *Gray level histogram (GLH)* or *Edginess Histogram (EH)*.

Second order histograms:

- *Gray level cooccurrence histogram (GLCH)*. It is based on cooccurrence matrix [1]. Each element $C_{q,d}(i,j)$ of the matrix represent an estimate of the probability that a pair of pixels with a specified separation (q, d) , have levels of gray i and j .

- *Gray level sum and difference histograms (GLSH, GLDH)*. These are the histograms of the sum and difference of all pixels d_x and d_y apart. The probability distribution of GLDH can also be used for texture classification. *DIFFX* and *DIFFY* are histograms of absolute feature differences between neighbouring pixels computed in horizontal and vertical directions, respectively, while *DIFF2* and *DIFF4* accumulates absolute differences in two or four principal directions, in a single histogram.

Statistics calculation (SC): this stage calculates the statistics from the histogram. A large number of texture statistics have been proposed [1], however only some of these are in general use: *Energy*, *Entropy*, *Maximum Probability*, *K moments*, *K Inv. Moments*, *ClusterShade*, *Clust. Prominence* and *Haralick's Correlation*.

For texture analysis, all of these algorithms are usually applied on square image sub-windows, mainly with 32x32 or 64x64 pixels and $G=256, 32, 16$ features.

3. DESIGN OF THE ARCHITECTURE

IP and HC stages perform data intensive tasks but only require elementary operations (difference, absolute value, sum, threshold, concatenation, etc...). Due to the simplicity of the operations and taking advantage of the parallelism of the logical blocks in the FPGA, the pixel

data stream can be processed by fixed-point arithmetic units in a pipelined fashion. SC stage performs all operations with floating-point arithmetic on a reduced set of data, DSPs are particularly suited to this kind of calculation, since they can make several floating-point products and accumulations in just one clock cycle. This stage works asynchronously on data stored in a local buffer memory. There are two main data streams: a video stream for pixel data coming from host, digitizer or digital camera, and a floating-point stream for returning the statistical measures to the host. In addition, a local bus is used to transfer data between the blocks.

In the proposed architecture (Fig1) a Shift Registers Module (SRM) is used for temporary storage of the pixels rows, this way one pixel can be processed by the Pre-processing module (PPM), every clock cycle. Histogram computation is similar for all algorithms; the histograms values (bins) are stored in an external memory (Histogram Buffer). The feature obtained from the PPM is used by the Address Generator to create the address of the bin that has to be incremented. An incrementer is used to carry out this operation and return the new bin value to the corresponding histogram buffer location. This task has to take into account the sub-window where the current pixel is included, hence all the histograms (one per sub-window) are calculated at the same time.

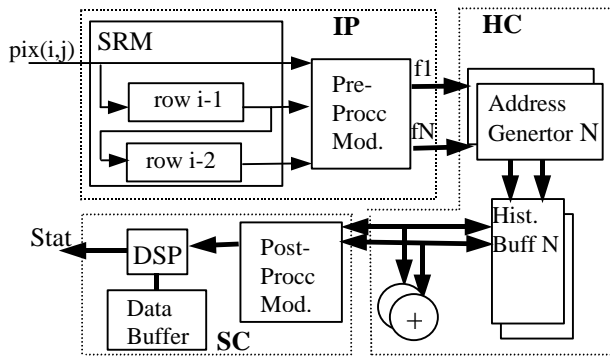


Figure 1: Architecture overview

Once the histograms computation has finished, the histograms bins are read and sent to the SC stage for floating-point computing. The Post-processing module permits the concatenation or accumulation of different histograms of the same sub-window. In this way rotation-invariant features can be obtained. Using independent Histogram Buffers, IP+HC stage can be overlapped with the SC stage. Several parameters of the architecture have to be set to fit the different algorithms. E.g. to implement GLCH with rotation invariance over images of 512x512 and taking windows of 32x32, the selected parameters are: $S=32$, $N=4$ ($d=1$ and $q=0^\circ, 45^\circ, 90^\circ, 135^\circ$), Pre_proc =concatenation, $Post_proc$ =Mean (acc+shift) of the four histograms bins of every sub-window.

4. IMPLEMENTATION ISSUES AND PRELIMINARY RESULTS

To validate the proposed architecture we used the reconfigurable board Mirotech Aristotle. Aristotle is a PCI board composed of two main components: a X4036 Xilinx FPGA with 1MB local SRAM and a TMS320C44 (@60MHz) Texas Instruments DSP with 1Mb global SRAM. IP and HC stages were implemented in the FPGA using the local SRAM as histogram buffers. Built-in synchronous RAM of the FPGA was used for implementing the SR module, and pre and post-processing operations were implemented using the arithmetic facilities of the logic blocks. With no optimization applied to the circuit design, the pipeline can work at 30MHz. The SC stage has been programmed on the DSP using the GNU C compiler. The most time-consuming parts of the program were written with assembly functions and parallel DSP instructions to improve the performance of the asynchronous computation.

The preliminary implementation achieves a performance of 25/30 frames per second (512x512 pixels) depending on the sub-window size and the number of statistics calculated (really DSP is the bottleneck of the system because of his obsolescence). This output is sufficient for a wide range of applications, however in order to compare the results with the new generation of general purpose microprocessors (which include SIMD and SIMD2 instructions), the introduction of new FPGA (e.g. Virtex), and DSP (e.g. TMS320C6X) technologies is necessary.

5. CONCLUSIONS

The hybrid architecture (pipeline/asynchronous) proposed achieves two main goals. Firstly, the architecture can be adapted to perform different texture analysis, thus providing a flexible enough alternative to traditional software implementations. Secondly, the performance of the implementation validates the perspective of using this architecture for real-time applications, while keeping costs down at the same time. In addition, this work shows that second order statistics analysis does not necessarily imply a complex and expensive implementation, when used in industrial applications.

6. REFERENCES

- [1] R.M. Haralick, L.Saphiro. Computer and Robot Vision. Vol I. Addison-Wesley, New York, 1992.
- [2] T. Ojala, M. Pietikäinen and D. Harwood. A comparative study of texture measures with classification based on feature distributions. Pattern Recognition 29(1):51-59, 1996.