Implementation of a Parallel Hough Transform Processor

W. Conen[1)], F. Klefenz[1)], R. Männer[1,2)], R. Zoz[1)]

[1)] Chair of Computer Science V, University of Mannheim, Mannheim, Germany
[2)] Interdisciplinary Center for Scientific Computing, Heidelberg, Germany

ABSTRACT

Systolic arrays belong to the class of pipelined array architectures where many identical processing elements (PE´s) are interconnected locally so that data can be passed from all PE´s to their respective neighbors synchronously and in parallel. In principle, all of them perform the same basic operation on their current operands in one clock cycle. At the University of Mannheim a systolic processor array is under development specialized to a specific pattern recognition task that has extreme high-speed requirements. The system is a systolic processor for the identification of circular particle tracks in a 2D projection. For each well defined track, the starting angle and the radius of curvature is computed in less than 5 μs. The system consists of a Hough transform processor that determines well defined tracks, and an Euler processor that counts their number by applying the Euler relation to the thresholded result of the Hough transform. A systolic processor consisting of 35×32 processing elements is being implemented in programmable gate arrays. It is scalable, so that the processor can easily be adapted to different generalized Hough transforms. The data and control flow in the systolic processor have been simulated completely.

1. Introduction

Many pattern recognition tasks can be solved by mapping the input data to a feature space in which a feature vector indicates a set of feature attributes. These attributes span a *n*-dimensional coordinate system for a number *n* of attributes. The coordinate transform can often be executed by a systolic array of identical PEs which operate in parallel on *n* data streams that are synchronized by a sequential ordering parameter. The coordinate transform can therefore be expressed by a kind of a space-time equation and can be formulated as a lock-step process, i.e., a data flow scheme on a clock cycle basis. Here the data arrive just-in-time at the right place where the right operation is executed. The input data are assumed to be asserted in the right sequential order. This has to be assured by appropriate preprocessing where necessary.

The systolic processor described here finds and counts in an image all circle segments with a common vertex point. It is designed to identify particle tracks in the OPAL wire chamber at CERN[1] which originate from the interaction zone and are given as circle segments of an arbitrary starting angle and a minimal radius of curvature. The

complete operation has to be finished within 5 µs. Fig. 1 shows one sector with detector data simulated by the software package GEANT.

Fig. 1: Typical track distribution in one sector of the OPAL detector

As discussed below, the pattern recognition process consists of four steps,

- preprocessing of the detector data for adaption to the systolic architecture,
- a Hough transform[2] that maps the detector coordinates in the (r,ϕ)-plane onto new co-ordinates in a $(1/r_C,\phi_S)$-plane where r_C is the radius of curvature and ϕ_S is the starting angle of a track,
- thresholding the transform result by a global threshold, and
- applying the Euler relation to compute the number of peaks in the resulting image.

The system therefore consists of a preprocessor, a Hough transform processor that determines well defined tracks, and an Euler processor that counts their number. For the latter two processors a systolic architecture is presented that is being implemented in field programmable gate arrays.

2. Principle of Operation

The OPAL detector consists of 24 sectors which operate independently of each other. Each track *i* is defined by a set of track points $(r,\phi)_i$. Because interesting tracks come from the origin, each one is determined by its starting angle ϕ_S and its radius of curvature r_C. Together with the origin every pixel (r,ϕ) in the detector image specifies a class of tracks with different ϕ_S and r_C[3]. From Fig. 2 one can derive the relation between (ϕ_S,r_C) and (r,ϕ) as

$$\frac{r/2}{r_c} = \sin(\phi - \phi_s) \qquad (1)$$

The histogramming method for identifying particle tracks used is based on accumulating the number of pixels that vote for every possible track (ϕ_S,r_C). As can be seen from Eq. (1), each pixel (r, ϕ) is mapped onto a sine curve with amplitude 2/r and a phase shift of ϕ. This mapping is a special type of a Hough transform. Applying this transform to every detector pixel yields one curve per pixel in the $(1/r_C,\phi_S)$-plane (Fig. 3). Pixels belonging to a track with parameters $(1/r_C,\phi_S)$ have a common intersection point. The result of this Hough transform therefore gives a probability measure for the existence of all possible tracks. Fig. 4 shows the representation of an ideal track with r_C = 10 m and ϕ_S = 0

degree in the $(1/r_C,\phi_S)$-plane. The sharp peak falls exactly in the right bin (r_C = 10 m and ϕ_S = 0 degree).

Fig. 2: Geometrical relationship between (r,ϕ) and $(1/r_C,\phi_S)$. Fig. 3: Hough space.

Fig. 4: Representation of an ideal track in the $(1/r_C,\phi_S)$-plane

3. The data flow architecture

3.1 Data preprocessing

In Fig.5 the data flow scheme and the involved pipelined processing stages are given. In the preprocessing stage a readout unit is assigned to every detector wire. The signal is digitized by an 100 MHz FADC system and the signal distribution is stored in RAM successively by incrementing the adress generation unit (AGU) (see Fig. 5). The hit detector (HDT) determines the drift time t_i from the signal distribution. These drift times are directly transformed by table-lookup (LUT) to the corresponding values ϕ_i. The list of values ϕ_i is stored in ascending order in a buffer (FIFO). This list of values ϕ_i is converted by a digital-to-time unit (DTC) into bit serial hit streams synchronized by a common clock, which are directly injected into the activation units of the Hough

transform processor. By appropriate preprocessing all detector pixels with a certain ϕ, i.e., up to 160 r coordinates per sector, are supplied in parallel and handled by the Hough transform processor in one cycle.

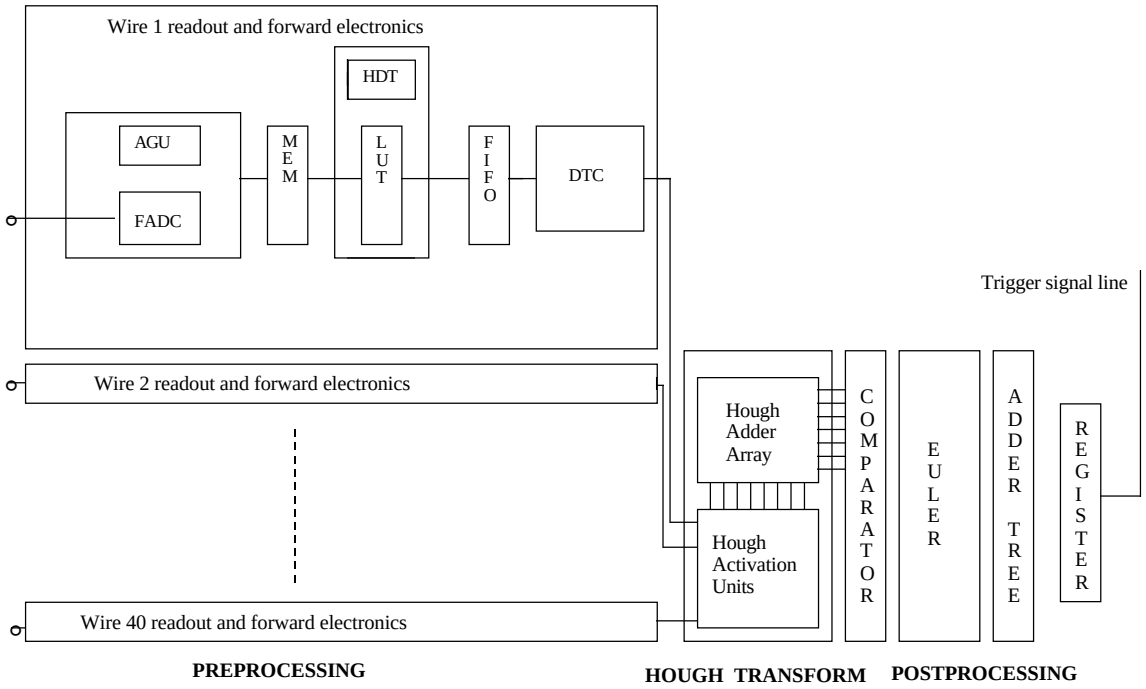

Fig. 5: System overview

3.2 The Hough transform processor

The Hough transform corresponds to the mapping described by Eq. (1). For every input pixel it is required to increment all bins of an accumulator plane which are covered by a certain sine curve. If all detector pixels had to be handled in parallel, all possible sine curves, i.e., sine curves with all possible combinations of amplitudes and phase shifts, had to be used in parallel. This multiple update operation is necessary because of the multiple intersections of sine curves. It can be avoided, if only the r coordinate is handled in parallel and the processing in the ϕ coordinate is serialized. This kind of serialization is suggested by the time-of-flight determination of ϕ in the OPAL detector (pixels with a larger distance to the center plane are read out later due to the detector set-up). Therefore only one class of sine curves - called mask below - with different amplitudes but fixed phase shift has to be handled at every instant t.

The essential operation is to generate for each r_i the corresponding sine at the right place in the $(1/r_C,\phi_S)$-plane at activation time t. Now instead of shifting the mask synchronously with ϕ over the accumulator plane, it is more appropriate to shift the contents of the accumulators synchronously with ϕ column by column in the opposite direction.

In this case the mask can be held fixed and every line of it will be activated individually. After all ϕ are handled sequentially, the Hough transform has been computed. This is described below in detail.

Because the detector is assembled of 48 half-sectors that are handled individually, only ϕ values from 0 to 7.5 degree have to be processed so that the sine curves can be approximated by straight lines. Thirty-five different lines with different slopes have to be generated and stored in the accumulator plane. All lines are generated by incremental moves in upward and sideward direction. Fig. 6 demonstrates the line generation process for one line with a slope of 45 degrees. Since sideward moves are restricted to one step because the contents of the accumulator array are shifted only one column at a time, all slopes are realized by the appropriate choice of upward moves. Slopes greater than 45 degrees are achievable, if several PEs are activated simultaneously; slopes less than 45 degrees, if delays are introduced in the vertical shift register. The singular point of the mask at $(1/r_C,\phi_S)=(0,0)$ has been avoided by shifting successive lines one pixel apart on the $1/r_C$ axis. To ensure proper updating in the accumulator plane the introduced spatial offset is compensated in accumulating all tokens at their actual position r_C row by row at a time (Fig. 7).

Fig. 6: Line generation by appropriate control token flow.

The Hough processor[4] is subdivided into two building blocks. First, it consists of a 35×32 array of counter elements. The array tiling is oriented in rows. The second part is a 32×35 activation matrix for the counters. This time the activation matrix tiling is oriented in columns. The 32 parallel count-enable lines of each column are connected to an activation unit for each column (Fig. 7). Each activation unit which is realized as a serial shift register with parallel output (SIPO) creates the right activation pattern of the PEs for one line. The serial bit stream is transformed into the successive activation of PEs with a fixed arrangement of delay elements and a fixed mapping of activation lines in every stage. Fig. 7 shows schematically several processing elements of the Hough adder array and the connection of the processing elements to the corresponding activation units.

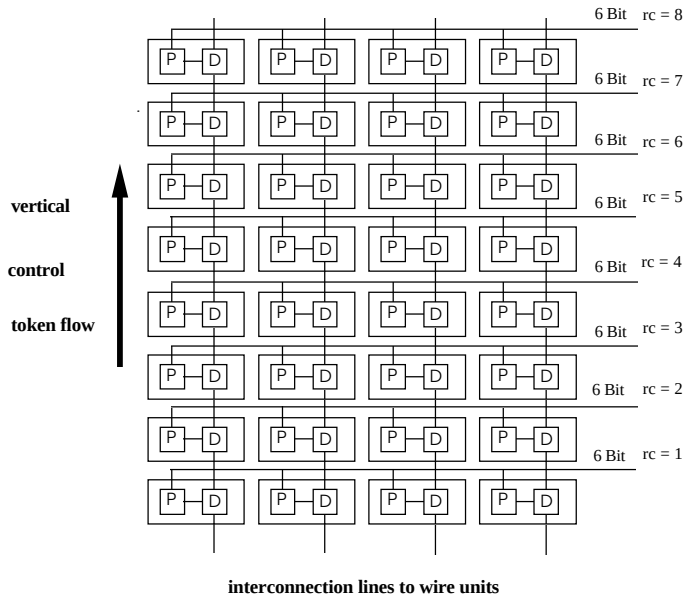

Fig. 7: Interconnection of the Hough adder array and the activation units (only a few PEs are shown).

3.3 The Euler Processor

The Hough transform processor outputs a 2D histogram which contains peaks at those positions in the $(1/r_C,\phi_S)$-plane that correspond to well defined tracks. By setting a global threshold in the 2D histogram a binary image is created that contains clusters of adjacent pixels at these positions. For a trigger decision these clusters have to be detected and counted. Counting of the clusters can easily be done with the help of the Euler relation:

$$\text{connectivity number} = n \bullet \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} - m \bullet \begin{bmatrix} x & 1 \\ 1 & 0 \end{bmatrix}.$$

The connectivity number is computed as the difference of how often the two 2×2 patterns appear in the image. Since the operation is fully local this updating can again easily be done in parallel by a systolic processor. The binary image that has been created by thresholding is shifted into the pipeline one column at every time step in parallel. The processor consists of thirty-two pattern matching units (PMUs) for parallel matching of the left pattern wherever it occurs along one column and thirty-two pattern matching units (PMUs) for the right pattern. A PMU sets the registration line "n" to high if the left pattern is detected or line "m" to high if the right pattern is detected. Each PMU is realized by simple logic elements. Subsequently an accumulator sums up the number of occurrences of n and m, which are delivered by the output signals of the PMUs. Then an integrator unit sums up the partial results and computes after *n* steps the difference, which represents the number of tracks which is taken as the trigger criterion.

4. Implementation by field programmable gate arrays

All processing stages (activation units, adders, comparators, Euler processor, integrators, subtractors) have been realized with Xilinx field programmable gate arrays with a supposed clock rate of 20 MHz derived from worst-case delay studies.The first prototype (for one detector sector) consists of 21 Xilinx chips in total.

For the Hough unit[5] the mapping of the PEs to the chips is dictated by the row-oriented data flow and the vertical control flow. The activation matrix is divided into sixteen columnwise connected units of 35 elements × 2 columns. Each of the 35 parts of a unit generates the input for two adder elements. One 35 input adder tree per row is used to sum up the activation data in parallel. Finally, the result is thresholded. To cover the whole array for one detector sector, sixteen XC3042 chips are needed. Each XC3042 contains two adder trees, two comparators, and one unit of the activation matrix.

The flow patterns of the distributed serial shift registers can be considered as the program of the systolic array. Since all other parts are uniform and static components which never have to be redefined this flow programming has been automated. The control flow pattern extracted from the simulation is directly converted to logic equations, the Xilinx programming cycle is initiated, and the resulting bit stream downloaded to the Xilinx chips. The threshold used by the comparators is reprogammable in a similar manner.

The Euler processor has been mapped into one XC3042 chip whose I/O pins are fully used. It consists of 32 pattern matching units analyzing one column at a time in parallel. A second implementation of the Euler chip is cascadable in both directions, so that larger arrays of size 2^n ($n^3 5$) can be handled in parallel.

5. Status

The systolic array described here is a special-purpose processor for low-level pattern recognition tasks. Due to the extreme speed requirements given, massive parallelism has been exploited by assigning one PE to every pixel of the images to be processed. Due to the high regularity of systolic arrays and the relative simple structure of the PE´s a VLSI implementation could be done easily. The system can be tailored and customized to a broad class of special configurations because the architecture is flexible, scalable, reconfigurable and reprogrammable.

References

1. Heuer R.D., Wagner A.: The OPAL Jet Chamber; Nucl. Instr. Meth., Vol. A265 (1988) p. 11-19
2. Davies E.R.: A Modified Hough Scheme for General Circle Location; Patt. Recog. Lett., Vol. 7 (1988) p. 37-43

3. Klefenz F., Männer R.: Echtzeit-Spurerkennung mit einem systolischen Arrayprozessor; in Burkhardt H., Höhne K.H., Neumann B. (Eds.): Mustererkennung 1989, *Informatikfachberichte 219*, Springer, Berlin (1989) p. 401-405
4. Klefenz F., Männer R.: A Systolic Track Finding Trigger Processor; *Comp. in High Energy Physics ´91*, Tsukuba City, Japan (1991) p. 219-222
5. Männer R., Gläß J., Klefenz F.: Massively Parallel Systolic Processors for High-Speed Recognition of Simple Patterns; *Proc. Parallel Computing Technologies ´91*, Novosibirsk, USSR, World Scientific, Singapure (1991) p. 98-107