A TRANSPUTER BASED SYSTEM FOR PARALLEL DYNAMIC TIME WARPING

Giulio Colangeli and Filippo Ardito
ALCATEL-FACE R.C., Via Nicaragua 10
00040 Pomezia, Italy

ABSTRACT
In this paper the use of the Transputer, as a basic processing element, to implement a parallel approach for Dynamic Time Warping algorithm will be described. A practical system, based on two levels hierarchic architecture, for parallel algorithms development in speech recognition field will be also introduced.

-1- Introduction
The Dynamic Time Warping (DTW) technique is a well known method for speech and pattern recognition for its efficiency, but it presents an upper limit from the computing requirements point of view. To face this problem several solutions based on parallel approach have been presented, generally by using dedicated VLSI's [2,3,4]. On the other hand, if that approaches may solve the computing requirements, they don't allow to build a flexible system which is required in the lab environment to investigate on the choice and tuning up of an efficient parallel algorithm.

In this paper the use of the TRANSPUTER processor and the OCCAM programming language to implement a parallel approach for Dynamic Time Warping algorithm will be described.

The availability of both a general purpose VLSI device, the TRANSPUTER [5], which can be used as a building block for parallel and concurrent systems, and a high level language with the related constructs as the OCCAM [6], allows to implement an array architecture and to map the DTW algorithm in a very effective way.

The system we are using in our experiments is based on two levels hierarchic architecture: at the upper level there is one transputer working as the MASTER controller of the system. It is devoted to interface the FRONT END processor, the HOST processor for data and/or program upload/download and to handle the data I/O (unknown and reference patterns) to/from the lower level.

At the lower level there is a ring array of transputers which implements the wavefront computation on the DTW plane of the algorithm [2]. Data flow is circular in the ring from left to right processing elements by using the links of the transputer.

The computation consists of the following sub-tasks:
- the distance calculation between the unknown pattern \( u(j) \) and the reference pattern \( r(i) \).
- the solution of the dynamic programming equation to extend the path on the DTW plane.
- the data transfer from the left to the right processor.

The above computation proceeds as a diagonal wavefront across the DTW plane.

The computation corresponds to a PROCESS in the OCCAM language and the sub-tasks can be executed concurrently on each transputer in the ring array by using the PARallel construct. The number of the processes is the the number of the transputers in the ring array.

The communication between processes is allowed by the CHAN assignment and by the IN, OUT primitive processes handling the channel link.

-2- The DTW algorithm
In speech recognition the comparison between an unknown and a set of reference templates is widely performed by the dynamic time warping (DTW) algorithm [1]. Since this algorithm is a well known technique, here we recall only its basic steps.

Let \( R_i \), \( U_j \) be the reference and unknown frames under processing. Each frame consists of 10 coefficients (Mel-cepstral parameters are used):

\[
\begin{align*}
R_i &= r_i(0), r_i(1), \ldots, r_i(9) \\
U_j &= u_j(0), u_j(1), \ldots, u_j(9)
\end{align*}
\]

The basic operations to execute on \( R_i, U_j \) are:
- Euclidean distance calculation

\[
d(i,j) = \sum_{k} [r_i(k) - u_j(k)]^2 \tag{1}
\]
- Cumulative distance calculation, by using an asymmetric form of DTW

EUROspeech '89, Paris, France, September 1989 1106
The path extension choice on the DTW plane is performed under the following constraints:

- a diagonal path (D) is always allowed
- a vertical path (V) is allowed only if the previous path was diagonal
- a horizontal path (H) is allowed only if the previous path was diagonal

Due to the path constraints the equation (2) become:

\[
g(i,j) = \min \left\{ \begin{array}{l}
g(i-1,j) + \frac{d(i,j) - d(i-1,j)}{2} \\
g(i-1,j-1) + d(i,j) \\
g(i,j-1) + d(i,j)
\end{array} \right. \tag{3}
\]

The (1), (3) are the basic operations used in the DTW we are implemented. By using (1),(3) and the data structure shown in fig. 1, the algorithm can be executed in recursive scheme with the following steps:

FOR each unknown frame
    FOR each reference frame
        - computing the Euclidean distance \(d(i,j)\)
        - read the previous Euclidean distance \(d(i-1,j)\)
        - read the previous cumulative distances \(D,H,V\)
        - read the path history
        - computing the new cumulative distance \(g(i,j)\)
        - write the cumulative distance
        - write the path history
    END
END

2.1 - The wavefront DTW

Looking at the DTW plane, where the reference frames are on the y axis and the unknown frames are on the x axis, the above processing scheme is "column" based, i.e. the execution proceeds along the y axis where, at the step \(i,j\):

- the cumulative distance \(H\) is \(g(i-1,j)\)
- the cumulative distance \(D\) is \(g(i-1,j-1) + H(i-1,j)\)
- the cumulative distance \(V\) is \(g(i-1,j)\)

The distances D, V, which are related to the previous reference frame \((i-1)\) can be stored in the internal registers of the DTW processor, while the distance \(H\) must be read from the reference memory and the new distance \(g(i,j)\) must be written in the same memory to be used when the new unknown frame \(U(j+1)\) will be available.

The amount of the memory accesses, for each reference frame, can be reduced implementing the DTW by a wavefront approach. A certain number of DTW processors, i.e. the processing elements PE, can be placed to calculate the basic operations along a diagonal wavefront on the DTW plane [2].

In this approach the DTW is executed on a "super-column" which size, i.e. the number of unknown frames, corresponds to the number L of PEs in the ring array. This approach can be implemented easily if each PE is a transputer because the transputer link architecture provides ease connection between PE's. By adopting this approach we have to consider the following implementation remarks (fig. 2):

- the unknown frame become the "super-unknown" frame:
  \[U_{kL} = (U_k, U_{k+1}, U_{k+2}, \ldots, U_{k+L})\]
- each diagonal corresponds to one step and the following relations between the step \(n\) and \(n+1\) exist:
  step \(n\) : - the transputer \(K\) executes the equations (1),(3) on \(R_i,U_j\)
        - the transputer \(K+1\) executes the eq. (1),(3) on \(R_{i+1},U_{j-1}\)
  step \(n+1\) : - the transputer \(K\) executes (1),(3) on \(R_{i+1},U_j\)
        - the transputer \(K+1\) executes (1),(3) on \(R_{i+1},U_{j+1}\)

Thus, within the "super-column", the transputer \(K\) processes a known frame which is shifted between the transputers : the next unknown frame is received from the predecessor transputer in the ring array while the currently is sent to the successor transputer.

- the cumulative distances \(D\), \(V\) and \(H\) can be transferred from/to the neighbour PEs in the ring array following the progress of the diagonal wave:
  \[D(n+1)\text{ of transp. } K+1 = H(n)\text{ of transp. } K\]
  \[V(n+1)\text{ of transp. } K+1 = g(n)\text{ of transp. } K\]

3- The system architecture

The overall speech recognition system is broken down into two levels depending from the real-time and computing requirements: at the upper level, the less intensive from the real-time point of view, are placed the MASTER transputer and the FRONT END processor.

The FRONT END is based on a DSP, TMS 320 family, which performs the task of the speech samples acquisition via the A/D device and the features extraction. This is realized by a 16 bands filter bank and a successive Mel-cepstral transformation. The speech signal, acquired with a 8 KHz sampling rate is reduced to 10 Mel-cepstral coefficients every 20 ms. corresponding to one speech frame. The
speech frame is sent to the Master via a parallel I/O port. The Master transputer interfaces the FRONT END on its own parallel bus.

To store the speech reference templates, the vocabulary, 2 Mbytes of external memory is also available. The DTW algorithm is performed at the lower level from a ring array of the transputers.

The communication from upper to the lower level is performed via the Master transputer links.

Because of the limit of the links, maximum four, a Programmable Link Switch is used to expand the link number up to 32. Via these links the reference and the unknown frames are transferred from upper to the lower level. At the lower level each transputer is connected to the transputer link which is right and the upper level via three of the four links available. The fourth link is used for testing purpose.

The system is shown in fig. 3.

The system has been planned to be realised in two versions: the former is built on two commercial boards from INMOS : the B002 board, containing the transputer Master and which allows the user interface with the VAX/VMS host, and the B003 board having four transputers which realise the ring array at the lower level.

The latter version, under realisation, will use both the B003 and B008 boards as host interface, VAX or PC environment, and several slave B008 boards to build the maximum ring array.

-4- The Data Flow and the Transputer implementation

The Transputer, as element of an array of processing elements, can allow only a data driven approach for the parallel implementation of the algorithm. This requires the following "running rule" to be satisfied:

Each transputer, in the ring array, is executing the same program but the sequence of the instructions is synchronised only by the correct sequence of data input and output from the channel links.

This is allowed by the progressive entry of the transputer in the running phase on the effective data. Moreover, the result have been already "produced" when they will be "consumed".

At the beginning, step 1, only the PE1 executes the (1),(3) on R1,U1 while the other ones are executing on dummy frames.

At the step 2, PE1 executes the (1),(3) on R1,U2. PE2, the successor PE in the ring, reads the distance D,V and the unknown U1 from PE1 and executes (1),(3) on R2,U1, while the others PE's are executing on dummy frames.

However, the output/input of the data, dummy or valid, is always necessary to avoid the dead-lock of the PE's.

-4.1- Boundary problems

In the data driven approach, the regular data flow must be guaranteed in spite of the boundary problems. Because of the ring array size, two boundary problems exist:

- within the "super-column"
- across the consecutive "super-columns"

The flexibility of the transputer links and the OCCAM language allows to overcome them.

Considering the "super-column" and the related "super-unknown", there is the following difference in the data flow. This is shown in the fig. 4, where, to simplify, the "super-unknown" consists of four frames: within the "super-column", the columns of the DTW plane can be classified in three classes:

-class A : the first frame of the "super-unknown"
-class B: the intermediate frames
-class C : the last frame of the "super-unknown"

If the class frame is A, the reference frame, the cumulative distance H and the "path history" must be read from the template reference memory.

When the class is C, the new cumulative distance g and the path history must be re-written in the template reference memory to be used in the next "super-column".

At the next step the transputer will run on a class A frame.

Since only the Master transputer can access to the template memory, a data transfer via the link from/to upper to/from lower level will be necessary.

If the frame is intermediate, class B, the transputer executes (1),(3) on the same reference frame while unknown and data are transferred circularly from the left to the right PE in the ring, to be used at the same or at the next step.

The transputer checks the class A, B or C by testing two flags which are shifted circularly in the ring from left to right PE's.

The flags are used as follows:

- R_input : when active, enable the request of a new reference to the master.

-D_save : when active, enable the cumulative distance g and path history transmission to the master.

When a "super-column" cross occurs, the transputer will run on a new "super-unknown". Therefore it must receive a new unknown frame from the master. This is allowed by a check on the flag EOC, i.e. end of column, which is stored in each reference frame.

-5- Conclusions

A transputer based system to be used in the lab environment for parallel algorithms development in the field of speech recognition has been described.

The wavefront approach for DTW algorithm has been implemented. This approach is based on that described in [2].
Up to now, an analysis of its real time features hasn't been performed, because it overcomes the purpose of this implementation.

Overall, the transputer and the OCCAM high level language allow a rapid and very stimulating method to tune up the algorithms and to investigate on parallel architectures [4,7]. The limited number of the transputer links seems not to be a disadvantage if a ring array architecture and/or a programmable link switch are used.

---

REFERENCES