Pergamon

Pattern Recognition, Vol. 31, No. 6, pp. 821–822, 1998

0031–3203/98 $19.00 + .00

PII: S0031-3203(97)00097-6

COMMENTS ON "MESH AND PYRAMID ALGORITHMS FOR ICONIC INDEXING": AUTHORS' REPLY

Dear Ms. Mossman,

We received comments by Drs Arndt, Costagliola and Chang (referred to by "authors" for the rest of this note) about our paper "Mesh and Pyramid Algorithms for Iconic indexing". We would like to begin by thanking the authors for pointing out several inadvertent errors. However, we do not agree with several comments. These are described briefly as follows:

1. Our definition of the different types of matching were derived from references 4, 6* and elaborated in Section 4. Reference 6 was omitted inadvertently in the introduction and in Section 4. Most of our work was done during the 1989–1990 academic year. At that time we only had the manuscript version of Ref. [1] available to us. As noted in our paper, we only presented a restricted form of type-2 matching.
2. We agree with the authors that the description of our algorithms assumes that each position is a symbol rather than a set of symbols. However, our algorithms can easily be generalized by replacing a symbol by a symbol bit vector (representing a symbol set). The length of this vector will be equal to the size of the vocabulary; the complexity of all the algorithms will typically be multiplied by the length of this bit vector. In fact, the algorithms described by the authors in their paper [1] also use symbols for the description of their algorithms. The algorithms in the authors' paper [2] use symbol or set of symbols for their description. However, there does not seem to be any major algorithmic changes due to this generalization. The only changes are in the matching function used. Other representations such as linked lists can be used instead of the bit vectors to represent a set of symbols. However, it is unclear how processing of such representations can be done effectively on SIMD machines.
3. We agree with the author's comments regarding Fig. 6. The text referencing the figure should be reworded to the following:

   Some of the matchings which can be potentially found in a picture for a given type-1 query are given in Fig. 6.
4. The use of hashing to reduce two-dimensional pattern matching to one-dimensional pattern matching was described in reference 19. It assumes that q is chosen to be a large enough constant prime number such that the probability of a collision is negligible. Our algorithm for exact matching (i.e. matching without any wildcards) uses this technique to reduce the time to $O(m)$. The amount of memory for storing the hash value is $O(1)$ as it is assumed to be independent of m (as in reference 19). In fact, if the size of the hash value is assumed to be $O(m)$, the time requirements of the algorithm will increase to $O(m^2)$ as each match will require $O(m)$ time. Hashing cannot be used when wildcards are present in the pattern.
5. We have assumed that broadcasting takes $O(1)$ time on a SIMD mesh and pyramid. SIMD machines assume that instructions are broadcast from the controller in constant time to all the processing elements and all the processing elements are synchronized at every step. Hence, broadcasting the same value from the controller to all the processors in unit time is not an unrealistic assumption. In fact, several commercial SIMD machines have this feature available.

   For the algorithms described in our paper, broadcasting from the controller to processing elements in unit time is not a requirement for the pyramid architecture. Pipelining can be used to broadcast a sequence of values from the apex such that a pattern value reaches every processor after unit time gaps after an initial delay equal to the height of the pyramid ($\log n$); a similar strategy was described in our paper for summing in Section 5.3. Such broadcasting can be achieved by the apex sending each pattern symbol to its descendants every time step rather than waiting for the pattern symbol to reach the leaf nodes. Each of the descendants perform a similar operation. All our algorithms can make use of this feature. Hence an additive rather than multiplicative factor of $\log n$ is required. This strategy can also be used even if the pattern is initially stored in the leaf nodes by first pipelining the symbols from the leaf nodes to the apex followed by the apex sending it to all the processors.

*These reference numbers are based on the reference list from our paper. We have used additional references which are denoted by Refs [1, 2] and appear at the end of this letter.

6. The presence of wildcards does not affect the algorithm for type-0 matching as described in our paper (Step 3 of Fig. 9); all that is required in Step 3 is to ignore the wildcards.

We would like to point out that the initial manuscript was written more than six years ago; subsequently, we have concentrated our effort on other research areas.

We will be happy to discuss any of the above comments with the editor or the authors. We can be reached at (847) 467 4129 or (352) 392 1526, respectively.

A. CHOUDHARY
ECE Department, Northwestern University

S. RANKA
CISE Department, University of Florida

REFERENCES

1. G. Tortora, G. Costagliola, T. Arndt and S. K. Chang, Pyramidal algorithms for iconic indexing, *Proc. IEEE Confon Systems. Man and Cybernetics*, Alexandria, VA, 1987, pp. 1004–1010.
2. G. Tortora, G. Costagliola, T. Arndt, S. K. Chang, Pyramidal algorithms for iconic indexing, *Comput. Vision, Graphics and Image Process.* **52,** 26–56 (1990).