

잉여수체계를 이용한 MDC 프로세서의 설계에 관한 연구

김형민\* , 조원경  
경희대학교 전자공학과

A study on the Design of MDC Processor using the Residue Number System

Hyeong-Min Kim\* , Won-Kyung Cho  
Kyung Hee University, Electronics

ABSTRACT

This paper proposes the Minimum-Distance Classification(MDC) processor using the Residue Number System(RNS). The proposed MDC Processor in this paper is efficient for real-time pattern clustering application and illustrate satisfiable error rate in application experiments of image segmentation but error rate increase as cluster number do.

I. 서론

패턴인식과 영상처리 등의 응용분야에 자동오차 패턴 클러스터링 방법을 이용한 고속의 패턴 클러스터링 프로세서가 요구된다. 그러나 자동오차 패턴 클러스터링 방법은 특징벡터행렬 연산의 반복적인 특성 때문에 상당한 처리시간을 요구한다. 본 논문에서는 잉여수체계를 이용하여, 자동오차 패턴 클러스터링을 실행하는 MDC(Minimum-Distance Classification) 프로세서를 병렬처리(Parallel Processing)와 파이프라인 구조(Pipelined structure)로 설계하여 실시간 패턴 클러스터링이 가능하도록 한다. 프로세서 설계시 연산회로는 일반적으로 2진수체계를 이용하는 데 2진수체계에서는 고속의 병렬승산기를 구성하기가 어렵다. 그러나 잉여수체계(Residue Number System)에서는 연산모듈을 이용하여 승산기를 구성하면 고속의 병렬승산기를 쉽게 구성할 수 있고 고속의 병렬승산기는 대규모의 데이터를 실시간으로 처리할 수 있도록 하며 MDC 프로세서에 이용하면 고속의 실시간 패턴 클러스터링이 가능하다. 잉여수체계는 2진수체계와는 다른 언웨이트드(Unweighted) 수체계로 연산시 캐리정보가 필요없기 때문에 캐리로 인한 지연시간의 필요없이 연산의 고속화가 가능하고 연산모듈이 서로 독립적이므로 병렬형 연산회로의 구성에 적합하다. 또한, 잉여수를 이용한 연산회로의 설계는 동일한 구조를 갖는 연산모듈도 가능하기 때문에 용이하게 VLSI로 제작할 수 있다 [1]. 제안된 MDC 프로세서를 영상분할 실험에 응용한 결과 2진수체계에서 실수연산을 이용하여

영상분할을 실시한 결과에 대하여 적은 오차를 가지며 만족할 만한 영상분할 결과를 얻었다. 또한, 기존의 연산회로와 비교한 결과 대규모의 데이터를 실시간으로 처리할 수 있음을 확인할 수 있었다.

II. MDC(Minimum-Distance Classification)의 기본개념

MDC 알고리즘은 자동오차 패턴 클러스터링 방법으로, 유클리디언 거리를 이용한 최소거리 판단에 바탕을 둔 발견적(heuristic) 과정으로 통계적 패턴인식과 영상처리에 이용되는 기술이며 라벨할당 과정과 Cluster Center Updating 과정으로 이루어진다. 라벨할당과정에서는 입력패턴행렬 $P(N \times D)$ 과 표준패턴행렬 $C(K \times D)$ 간의 거리값에 의하여 각 입력패턴을 가장 가까운 클러스터로 할당한다. 여기서  $N$ 은 패턴수,  $D$ 는 특징차수이며  $K$ 는 클러스터수이다. 라벨할당과정은 (1) 식에 의한 입력패턴과 표준패턴간의 유클리디언 거리(ED) 계산과

$$ED_n = d^2(P^n, C^k) \quad (1)$$

(2) 식의 각 입력패턴을 가장 가까운 클러스터로 라벨을 할당시키는 두 단계로 볼 수 있다.

$$L(n) = k, \quad (2)$$

$$d^2(P(n), C(k)) = \min\{ED_0^n, ED_1^n, \dots, ED_{K-1}^n\}$$

$L(n)$ 은  $n$  번째 입력패턴에 할당된 라벨이다. cluster center updating 과정에서는 라벨할당과정에서 생성된 각 클러스터내의 패턴들에 의하여 새로운 클러스터 중심값들을 (3) 식으로 구한다.

$$C(k, d) = \frac{1}{TS(k)} \sum_{n \in S(k)} P(n, d) \quad (3)$$

$|S(k)|$ 는  $k$  번째 클러스터  $S(k)$ 의 패턴 수 MDC 알고리즘은 원하는 최종결과를 얻기까지 위의 과정을 반복하며 흐름도는 그림 1과 같다.

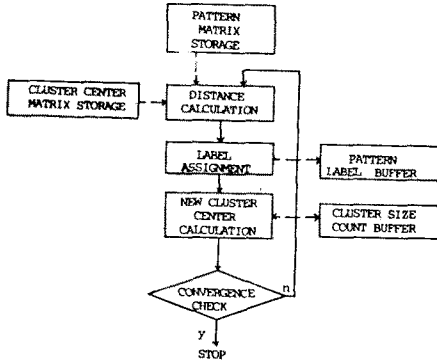


그림 1 자동오차 패턴 클러스터링의 흐름도

III. MDC 프로세서의 설계

1. 기본구성

잉여수체계를 이용한 MDC 프로세서의 기본 구성도는 그림 2와 같다. 그림 2의 MDC 프로세서는 유클리디언 거리를 구하는 데 이용되는 DSA, 최소거리값을 갖는 클러스터를 선택하는 데 필요한 비교기와 기타 필요한 정보의 기억을 위한 FIFO 형 라벨버퍼 및 클러스터 카운터로 구성되며 전채적으로 파이프라인 구조를 갖는다.

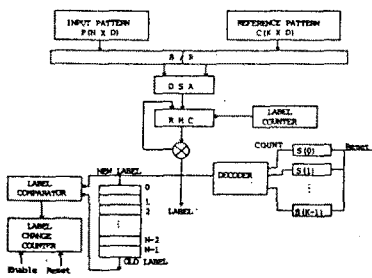


그림 2 라벨할당 프로세서의 구성도

2. 표준패턴 기억장치

고속의 연산에 필요한 데이터의 원활한 공급을 위한 기억장치는 그림 3과 같다. 표준패턴 기억

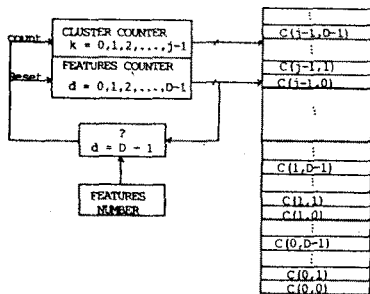


그림 3 표준패턴 기억장치

장치는 j 개의 세그먼트로 나누어지고 각 세그먼트에는 D 차의 특징 데이터를 저장할 수 있다. 한 세그먼트는 2의 멱수 (2^n)로 구성되며 2^n >= D를 만족한다. 이와 같은 기억장치로부터 데이터를 액세스하기 위하여 2 개의 어드레스 카운터가 필요하다. 세그먼트 카운터는 각 세그먼트를 어드레스 (k = 0, ..., K-1) 하는데 이용된다. 특징차수 카운터는 각 차수의 특징 데이터를 어드레스 (d = 0, ..., D-1)하는데 이용된다. 특징차수 카운터의 어드레스는 특징차수 (D 차)를 기억하고 있는 레지스터의 값과 비교하여 D - 1 에 이르면 특징차수 카운터는 리셋되고 세그먼트 카운터는 1씩 증가하게 되어 다음 세그먼트의 표준패턴 데이터를 출력시킨다.

3. DSA (Difference-Squared-Accumulator)

DSA 는 입력패턴과 표준패턴 사이의 유클리디언 거리값을 구하는 데 이용된다. 잉여수체계에서의 거리값은 법 mi 일 경우 (4) 와 같이 계산된다.

$$ED_k^n \text{ mod } m_i = \left( \sum_{d=0}^{D-1} (P(n, d) \text{ mod } m_i - C(k, d) \text{ mod } m_i)^2 \text{ mod } m_i \right) \text{ mod } m_i \quad (4)$$

DSA 연산을 위한 기본 연산모듈을 잉여수체계를 이용하여 구성한 예는 그림 4와 같다. 그림 4의 점선 내의 뺄셈과 제곱 연산은 2진수체계를 이용할 경우 순서적으로 두 단계의 연산을 거쳐야 하지만 잉여수체계에서는 (a\_j - b\_j)^2 형태의 연산이 하나의 연산모듈로 가능하므로 하드웨어의 크기와 연산시간을 감소할 수 있다. 따라서 그림 4의 연산모듈을 이용하는 경우 K 개의 표준패턴과 D 차의 특징차수를 갖는 N 개의 입력패턴을 처리하는 데 NKD 클럭이 소요된다.

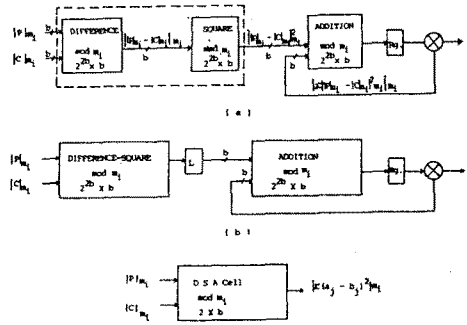


그림 4 DSA 의 블록 다이어그램

4. 혼합기수변환을 이용한 비교기

잉여수체계는 비중가 표현이므로 수의 부호, 오버플로우(overflow) 검출과 크기비교를 위하여는 비중가(Unweighted) 표현을 중가(Weighted) 표현으로 변환하여야 한다. 이에 적합한 알고리즘이

혼합기수변환 알고리즘이며 병렬성을 높인 병렬 혼합기수변환 알고리즘이 제안되고 있다.[3] 본 논문에서 제안한 MDC 프로세서는 최종 결과로 라벨만을 출력하므로 잉여수를 2진수체계로 변환할 필요가 없으며 잉여수의 부호검출만을 필요로 한다. 따라서 여기서는 혼합기수변환 알고리즘을 이용한 부호검출기를 이용하였다. 구성된 비교기의 하드웨어 구성도는 그림 5와 같다. 비교기는

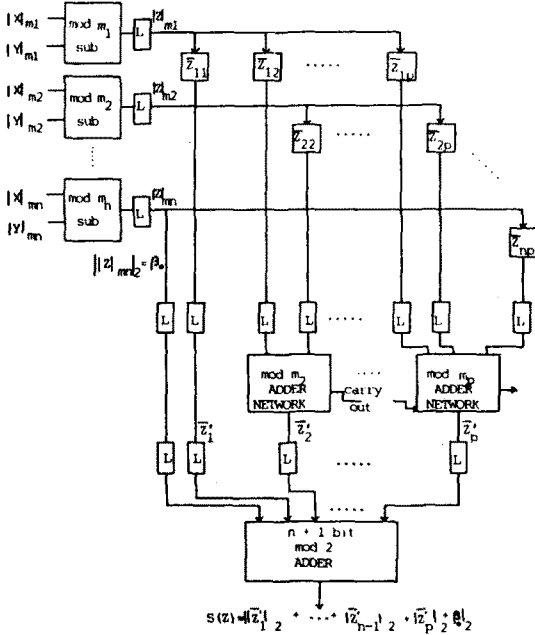


그림 5 혼합기수변환을 이용한 비교기

출력단에 래치를 이용하여 한계값으로 초기화시키고 D 클럭마다 현재까지의 최소 거리값을 출력한다. 따라서 오버플로우가 발생하지 않는다. DSA 와 비교기 등을 이용하여 구성된 MDC 프로세서의 입력패턴 처리시간은 다음과 같다.

초기 지연시간 :  $t_c + t_{DSA} + t_{RMC}$   
 한 입력패턴 처리시간 :  $T(1, K, D) = t_c + t_{DSA} + t_{RMC} + KD$   
 N 개의 입력패턴 처리시간 :  $T(N, K, D) = t_c + t_{DSA} + t_{RMC} + (N - 1) t_d + NKD$

단위는 클럭 사이클이며 기본 연산모듈의 연산시간에 의하여 결정된다. 여기서

- $t_c$  : B / R 변환기의 소요 클럭 사이클 수
- $t_{DSA}$  : DSA 의 소요 클럭 사이클 수
- $t_{RMC}$  : 비교기(RMC) 의 소요 클럭 사이클 수
- $t_d$  :  $t_{RMC} - 1$  이다.

IV. 실험 및 고찰

잉여수체계를 이용하여 구성된 MDC 프로세서의 이론적인 타당성과 응용실험을 위하여 디지털 컴퓨터를 이용하여 시뮬레이션을 실행하였다. 실험 방법은 XENIX system V에서 C 언어를 이용하여 B/R 변환기, DSA 와 비교기를 부함수로 작성하여 각 모듈의 동작을 정의하고 MDC 프로세서의 시뮬레이션을 진행하였다. 또한, MDC 프로세서의 응용실험을 위하여 영상분할에 필요한 특징을 Sobel 연산자를 이용하여 추출하였다. 초기 클러스터의 중심값으로 영상의 입력화소중 임의의 화소를 취하였다. MDC 프로세서의 각 패스에 소요되는 처리시간은 입력패턴의 수가 많을 경우에 초기 지연시간은 무시할 수 있으므로 (5) 식과 같이 간략화할 수 있다.

$$T(N, K, D) \approx (N-1)t_d + NKD \quad (5)$$

단위는 클럭 사이클이다.

기존 연산회로의 비교를 위하여, 10MHz 의 클럭을 이용하여 80287 Coprocessor 를 이용할 경우와 제안한 MDC 프로세서의 Cronkite 영상에 대한 처리시간 비교를 표 1과 그림 6에 보인다. 여기서 Coprocessor 에 의한 처리시간은 1/20배로 축소하여 그래프에 나타내었다. 영상분할 실험의 실험결과 클러스터의 수가 적을 경우에는 2진수체계를 이용한 실수 연산결과와 잉여수체계를 이용한 결과는 거의 동일하였다. 그러나 클러스터의 수

| CLUSTER | Unit (Sec.)        |                    |
|---------|--------------------|--------------------|
|         | PROPOSED PROCESSOR | 80287 CO-PROCESSOR |
| 2       | 0.0246             | 5.50               |
| 4       | 0.0442             | 11.10              |
| 8       | 0.0836             | 22.31              |
| 16      | 0.1622             | 44.72              |
| 32      | 0.3190             | 88.55              |

표 1 처리시간의 비교

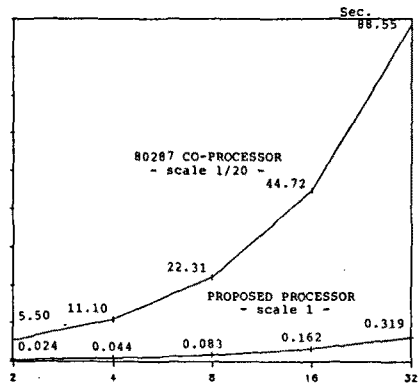


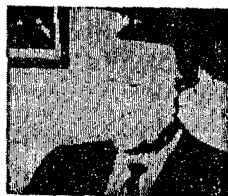
그림 6 처리시간의 비교 그래프

| Image       | Grey Level | Cluster Number | Pattern Number | Misclassified Pattern Number | Error (%) |
|-------------|------------|----------------|----------------|------------------------------|-----------|
| "LINCOLN"   | 64         | 2              | 4096           | 5                            | 0.12      |
|             |            | 3              | 4096           | 70                           | 1.70      |
| "HOUSE"     | 64         | 2              | 4096           | 0                            | 0.        |
|             |            | 3              | 4096           | 0                            | 0.        |
|             |            | 4              | 4096           | 160                          | 3.90      |
| "CHRONKITE" | 256        | 2              | 16384          | 0                            | 0.        |
|             |            | 3              | 16384          | 72                           | 0.43      |
|             |            | 4              | 16384          | 612                          | 3.73      |

표 2 영상분할 실험결과



(a) original image



(b) 2 cluster of float



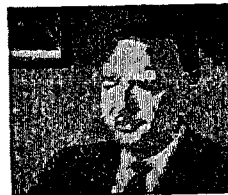
(c) 3 cluster of float



(d) 4 cluster of float



(e) 2 cluster of RNS



(f) 3 cluster of RNS



(g) 4 cluster of RNS

그림 7 Cronkite 영상의 영상분할 결과

가 증가함에 따라 오차도 증가하였으며 실험결과 는 표 2와 같다. 그림 7은 실험에 의하여 구해 진 결과의 한 예이다. 실험결과 제안된 MDC 프로세서는 많은 데이터를 실시간으로 처리하여야 하는 영상처리 분야에 효과적으로 이용할 수 있음을 확인할 수 있었다.

V. 결 론

본 논문에서는 잉여수체계와 MDC 알고리즘을 이용하여 통계적 패턴 클러스터링을 실행하는 MDC 프로세서를 설계하였다. 설계된 MDC 프로세서를 대규모의 특징벡터행렬 데이터를 실시간으로 처리하여야 할 필요가 있는 영상분할에 적용한 결과 패턴인식과 영상처리 분야에 효과적으로 이용할 수 있음을 확인 하였다. 제안된 MDC 프로세서는 병렬처리와 파이프라인 구조로 설계하였으며, 잉여수체계를 이용하여 고속의  $(a_j - b_j)^2$  형태의 연산을 실행할 수 있기 때문에 실시간 처리를 필요로 하는 분야에 이용할 수 있을 것으로 기대가 된다. 또한, 연산모듈의 규칙적인 배열에 의하여 회로를 구성할 수 있으므로 VLSI로 제작하기가 용이한 장점이 있다.

참 고 문 헌

- [1]. 조 원경, "RNS 를 이용한 연산 프로세서의 설계에 관한 연구", 한양대학교 박사학위 논문, 6. 1986.
- [2]. L. M. Ni and A. K. Jain, "A VLSI Systolic Architecture for Pattern Clustering", IEEE Trans. on PAMI, Vol. PAMI-7, No.1, Jan.1985
- [3]. C. H. Huang, "A Fully Parallel Mixed-Radix Conversion Algorithm for Residue Number Applications", IEEE Trans. on Comp, Vol. C-32, No.6, pp. 398 ~ 403, April. 1983.
- [4]. Z. D. Ulman, "Sign Detection and Implicit-Explicit Conversion of Numbers in Residue Arithmetic", IEEE Trans. on Computers Vol.C-31, No. 6, pp. 590 ~ 594, June. 1983.
- [5]. H. H. Liu and K.S.Fu, "VLSI Architecture for Minimum-Distance Classification", Proc. Intern. Conf. on Computer Design: VLSI in Computers, 1983.
- [6]. N. S. Szabo and R. I. Tanaka, "Residue Arithmetic and Its Application to Computer Technology", McGraw-Hill, 1967.
- [7]. G. B. Coleman and H. C. Andrews, "Image Segmentation by Clustering", IEEE proc., Vol. 67, No. 5, pp. 773 ~ 785, May. 1979