문자열 패턴 매칭 (SPM:String Pattern Matching) 프로세서의 설계

국 일 호* 조 원 경
경희대학교 전자공학과

Design of String Pattern Matching (SPM) Processor

Il-Ho Kook , Won-Kyung Cho
Kyung Hee University, Electronics

ABSTRACT- SPM is MDC Processor for string pattern expressed in directional chain code. In this paper we consider the string pattern matching algorithm (Leve-nstein Algorithm) whitch is portion of Dynamic Programing, and propose architecture of SPM and simulate it on the R-T level to evaluate its architecture. We used the C language as the hardware description language, and developed it on the IBM PC/AT Zenix system V OS environment.

1. 서 론

MDC(Minimum-Disatance Classification) 알고리즘은 통계적인 패턴 인식으로 널리 알려진 방법으로[1][3], 본 논문에서는 방향성 체인코드 또는 심볼로 표현된 문자열 패턴에 대한 MDC 알고리즘을 이용한다. 문자열 패턴을 위한 MDC 알고리즘은 입력문자열패턴 X와 표준문자열패턴 M^l(l=0,1,...,n) 간의 레벤스타인 디스턴스 $d^l(X,M^l)$를 구하여 최소 거리를 갖는 표준문자열패턴의 라벨 l 을 구한다[1][2][3].

본 논문에서 제안한 SPM 은 문자열패턴의 MDC 알고리즘을 위한 프로세서로서, 문자인식에서 문자 구성의 최소단위인 문자열패턴으로 표현한 스트로크의 인식에의 응용을 위하여 설계된 것이다

2. 알고리즘 (Algorithm)

다이나믹 프로그램밍[5] 의 한 방법으로 잘알려진 레벤스타인 디스턴스 알고리즘에 의하여 체인코드로 표현된 두 문자열 패턴간의 패턴매칭 및 디스턴스의 계산이 이루어 진다[2][3][4][5]. 두 문자열의 매칭의 과정에서 입력문자열 X 의 i 번째 심볼 x_i 와 표준문자열 M 의 j 번째심볼 m_j 을 비교하여 $x_i \neq m_j$ 일경우 x_i 를 에러변환하게되며 변환의 종류에 따른 디스턴스 또는 거리값의 계산이 이루어진다. 이때, (i,j)에서의 부분거리값은,

$$S_{i,j} = \min \begin{vmatrix} S_{i-1,j} + Wd \\ S_{i-1,j-1} + Ws \\ S_{i,j-1} + Wi \end{vmatrix}$$

가 된다. 이때 Wd, Ws, Wi는 각각 삭제변환, 대체변환, 삽입변환에 대한 가중치가 된다. 단, $x_i = m_j$ 일경우 Ws = 0이다. 그림 1. 에 두 문자열간의 매칭과정 및 에러변환의 종류를 나타내었다.

또한 이 알고리즘은 변환 가중치[6] 에 따라, 비가중치 (Unweighted)알고리즘, 준가중치(Semi-Weighted) 알고리즘, 전가중치(Full-Weighted) 알고리즘으로 나누어고찰 하여본다.

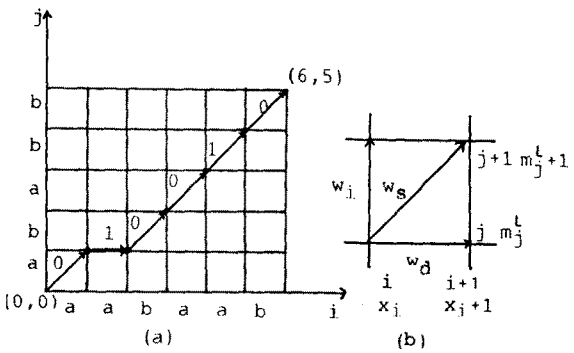

그림 1. 두 문자열의 매칭과정(a) 및 에러변환종류 (b)

Fig. 1. String Matching Process (a) and Error Transform (b)

ㅇ 비 가중치 레벤스타인 알고리즘
(Unweighted Levenstein Algorithm)
에러변환에대한 가중치가 다음과 같을경우,
Wi=Ws=Wd=1

ㅇ 준 가중치 레벤스타인 알고리즘
(Semi-Weighted Levenstein Algorithm)
에러변환에 대한 가중치가 다음과 같을경우,
Wi ≠ Ws ≠ Wd

ㅇ 전 가중치 알고리즘
(Full-Weighted Levenstein Algorithm)
방향성코드 a_i (i=0,1,2,...,m) 에 대하여 변환에 따라 가중치가 모두 다를경우. 즉,

$Wi(a_i) \neq Ws(a_i) \neq Wd(a_i)$

그리고,

$Wi(a_i) \neq Wi(a_j)$

$Ws(a_i) \neq Ws(a_j)$

$Wd(a_i) \neq Wd(a_j)$

이때, $a_i \neq a_j$ 이다.

3. SPM 의 설계

(1) SPM 의 구조

그림 2는 SPM 의 구조를 기능적인 유니트별로 나타낸것이다. 그림2 에서와 같이 SPM 은 크게 심볼 비교및 가중치 가산부(Symbol Compare & Weight Adder Unit), 최소거리값 비교부 (Minimum Distance Unit) , 최소거리라벨부(Argmin Unit) , 데이타 어드레스발생부(Data Address Generation Unit)등 4개의부분 으로 나뉘어진다.

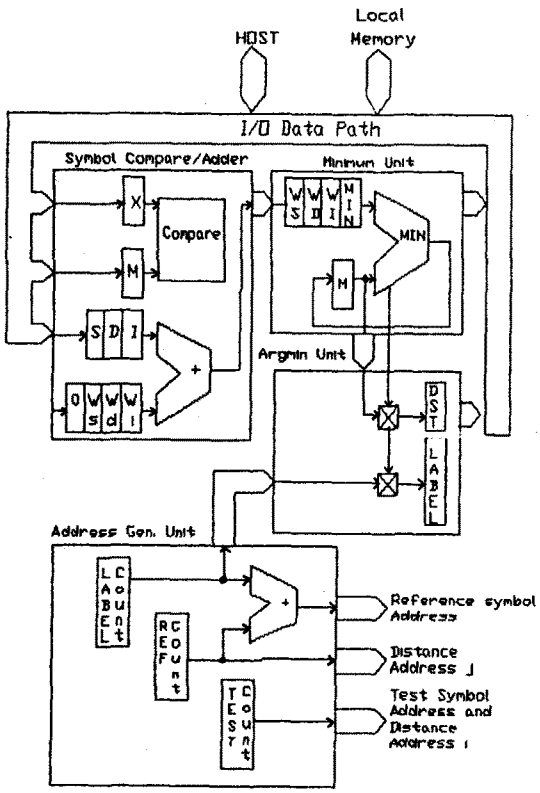

그림 2. SPM 의 내부 구조도

Fig. 2. SPM Internal Block Diagram

각 부분별 기능은 다음과 같다.

○ 가중치 가산부

(Symbol Compare & Weight Addition Unit)

표준패턴과 입력패턴의 심볼, m_j ,x_i 및 (i,j) 이전까지의 각각 변환에 대한 부분거리값 Ws, Wi 및 Wd를 모드한후 두 심볼을 비교하여 에러변환에 따라 가중치를 더하므로서 (i,j) 에서의 가능한 변환에 대한 부분 거리값 WS, WI및 WD를 구한다.

$WD = S(i-1,j) + Wd$

$WS = S(i-1,j-1) + Ws$

$WI = S(i,j-1) + Wi$

○ 최소거리값 비교부 (Minimum Distance Unit)

(i,j) 에서의 각변환에 대한 거리값중 가장작은 값을 취하여 최소거리를 구한다. i=| X | 및 j=| M^l |일때 최종적인 두문자열간의 최소거리(Minimum Distance) 값 MIN이 된다.

$MIN=min(WD,WS,WI)$

○ 최소거리값 라벨버퍼부 (Argmin Unit)

l 번째 표준문자열패턴 M 과 입력 문자열 패턴 X 의 모든 심볼을 비교후 최소거리값인 MIN ($= d^l(M^l,X)$)와 l-1 까지의 최소거리값 DST ($= d^{l-1}(M^{l-1},X)$) 와 비교하여 MIN < DST 일 경우 DST 및 LABEL 을 Up-Date 한다.

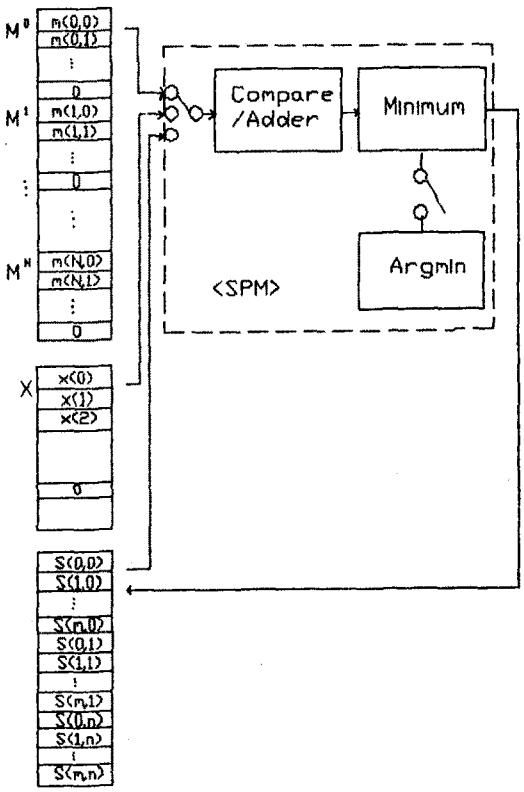

그림 3. SPM 의 주변 데이타 기억장치의 구조

Fig. 3. local data memory structure of SPM

○ 데이타 어드레스 발생부

(Data Address Generation Unit)

그림 3에서 SPM 의 주변 데이타 기억장치(Local Data Memory) 구성을 나타내었다. SPM 은 l 번째 표준문자열패턴 $M^l (=m_0^l, m_1^l, m_2^l, \ldots, m_j^l, \ldots)$ 과 입력

문자열패턴 $X (= x_0, x_1, x_2, \ldots, x_i, \ldots)$와의 패턴매칭 과정중 부분 거리값 S(i,j), 표준문자열패턴 $M^l (= M^0, M^1, M^2, \ldots)$ 의 심볼 데이타 m_j^l , 입력문자열패턴 x_i의 저장을 위한 기억장치 등 3 가지의 주변 데이타 기억장치를 갖게 되며 이들 기억장치의 엑세스를위한 어드레스를 발생한다. 어드레스의 발생을위한 제어흐름은 그림 4에 나타내었다.

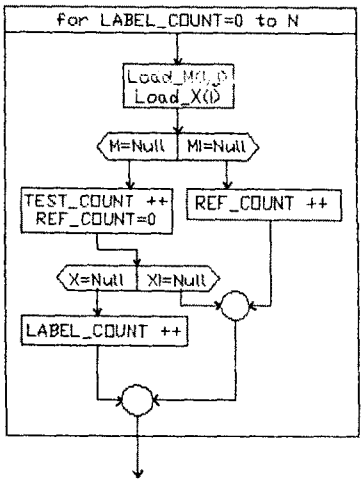

그림 4. 어드레스 발생부의 제어 흐름도
Fig. 4. Control flow of Address generation unit

SPM 의 구조적인 특징은 문자열중 널(Null)데이타에 의하여 어드레스의 발생이 이루어지는등 데이타의 흐름에 의하여 전체 프로세스의 제어가 이루어 진다. 그러므로, SPM 고유의 인스트럭션은 갖지않는다. 또한 표준 문자열 패턴의 갯수 및 변환 가중치의 값은 프로그램머블 하다.

(2) SPM 의 아키텍춰의 기술 (Description)

SPM 구조의 평가 (Architecture Evaluation)를 위하여 C 언어에 의한 아키텍춰를 기술함으로써 SPM 아키텍춰 시뮬레이터(Arichitecture Simulator)를 개발하였다. C 언어에 의한 마이크로 아키텍처 시뮬레이터의 기술은 참고문헌[7] 에 의한다.

SPM 아키텍춰 시뮬레이션은 각 유니트의 기능에 따라 입출력 신호간의 관계를 기술함으로써 가능하며, 아키텍처기술 형식은 다음과 같다.

```
/************************************
    SPM : String Pattern Matcher
 ************************************/
struct {
        unsigned  X:5;
        unsigned  M:5;
          } Reg; /* Register Def. */
                 :
SPM(IO)     /* System & IO Declare */
struct  {
                 :
  /* IO & Local Memory Definition */
                 :
        } IO;

/* Process & Control Flow */

Address_Gen();
        :
Load(l,i,j);
Compare_Add();
        :
if (X!=Null)
        MIN=min(WS,WI,WD)
        Store(i,j);
else
        Argmin();
        :
```

4. 결론

SPM 아키텍춰 시뮬레이터는 IBM PC/AT 의 Zenix System V OS환경에서 C 언어에 의하여 개발되었다. SPM 구조의 기술부분의 프로그램의 크기는 약 300line정도로서 시뮬레이터전체 프로그램의 약 2/3 정도를 차지하며, 레지스터의 정의 및 각 기능별 유니트등 하드웨어 기술을 위한 것이다. SPM 아키텍춰의 평가는 데이타 어드레스발생이 올바른가 살펴보고 데이타의 흐름에따른 각유니트별 레지스터 값의 변화를 살펴 보았으며, 끝으로 두 문자열 패턴간의 거리값이 올바른가를 조사함으로써 마쳤다. SPM 은 문자인식 시스템의 한부분으로서 문자구성의 최소단위인 스트로크의 인식을 위하여 개발된 것으로 비교적 단순한 획을갖는 한글필기체에 대하여 응용되었으나, 좀더 복잡한 도형 또는 수학적인 도형등의 인식에 적용해볼 계획이다

참고문헌

[1] J. T. Tou and R.C. Gonzalez, Pattern Recognition Principles, Chap. 3, Addition-Wesley, 1974

[2] L. Miclet, Structural Methods in Pattern Recognition, Chap. 1, Springer-Verlag, 1986

[3] K. S. Fu, Syntactic Pattern Recognition and Application, Chap. 7, Prentice-Hall, 1982

[4] H. H. Liu and K. S. Fu, "VLSI Arrays for Minimum -Distance Classifications", VLSI for Pattern Recognition and Image Processing, Springer-Verlag, 1984

[5] H. Ney, "Dynamic Programing as a Technique for Pattern Recognition", _____

[6] M. H. Ackroyd, "Isolated Word Recognition Using the Weighted Levenstein Distance", IEEE Trans. on ASSP, Vol. ASSP-28, No.2, Apr.,1980

[7] 박병관, 배상덕, 서대화, 윤용호, " 마이크로 아키텍춰 시뮬레이터", 전자공학회논문지, 제24권 제3 호, 5 월 1987 년

[8] S. C. Glinski et al, "The Graph Search Machine (GSM) ", Proc. IEEE, Vol. 75, No.9, Sep.,1987