

고속 처리를 위한 이진 영상 정규화 하드웨어의 설계 및 구현

(Design and Implementation of Binary Image Normalization Hardware for High Speed Processing)

金赫九**, 姜仙美**, 金惠鎮*

(Hyeog Gu Kim, Sun Mee Kang and Duck Jin Kim)

要約

이진 영상의 정규화 기법은 영상처리 분야에 다각적으로 사용될 수 있으며, 고속 처리를 위한 하드웨어의 구현은 실질적으로 유용하다. 문자 인식 과정 중에는 개별 문자 영상을 일정한 크기로 정규화하는 과정이 필요한데, 이에 많은 처리시간이 소요된다. 따라서, 본 연구는 고속의 문자 인식기(OCR)를 구현하기 위하여 정규화 과정을 하드웨어로 구현함으로써 호스트 컴퓨터와 기능적인 파이프라인(pipeline) 구조를 이루어 시간적 병렬성을 갖도록 하였다.

정규화 하드웨어의 구현은 전체 문자 인식기의 처리 과정에 영향을 미치지 않는 범위내에서 정규화가 이루어 질 수 있도록 범용의 CPU(MC 68000)를 사용하였다. 성능은 초당 140자 이상의 고속 문자 인식기를 구현하는데 사용하여 실험한 결과 요구되는 처리 속도를 얻을 수 있었다.

Abstract

The binary image normalization method in image processing can be used in several fields. Especially, its high speed processing method and its hardware implementation is more useful. A normalization process of each character in character recognition requires a lot of processing time. Therefore, the research was done as a part of high speed process of OCR(optical character reader) implementation as a pipeline structure with host computer in hardware to give temporal parallelism. For normalization process, general purpose CPU, MC68000, was used to implement it. As a result of experiment, the normalization speed of the hardware is sufficient to implement high speed OCR which the recognition speed is over 140 characters per second.

1. 서론

* 正會員, 高麗大學校 電子工學科

(Dept. of Elec. Eng., Korea Univ.)

** 正會員, 情報通信技術共同研究所

(Research Institute for Information and Communication)

※ 본 논문은 삼성전자 위탁과제로 수행되었음.
接受日字 : 1993年 7月 28日

영상처리 분야에 있어서 주어진 영상의 모양을 유지하면서 크기에 대한 정규화를 수행하는 것은 정보의 압축이나, 특정 크기로의 변형등 그 응용 분야가 다양하다. 본 연구는 고속 문자 인식기의 구현을 위해 요구되는 이진 영상의 정규화 과정을 고속 처리하기 위하여 제안된 알고리즘^[1]을 위한 정규화 하드웨

어를 설계 및 구현하였다.

일반적인 문자인식 과정은 스캐너나 카메라를 통해 입력된 영상을 배경 부분과 문자 영상 부분의 명암의 차를 이용하여 이진화를 시킨다. 이진화된 영상으로부터 문자 영역을 분할하고, 분할된 문자 영역으로부터 개별문자를 추출한다. 추출된 개별 문자들은 각각의 문자 영상이 갖는 특징을 이용하여 여러 단계의 처리 과정을 거쳐 하나의 문자 코드를 생성한다.

통계적 방법을 이용한 일반적인 문자 인식 과정은 그림 1에 나타난 것과 같이^[1-5], 특징추출을 위한 전 처리 단계로 개별문자 영상을 특정 크기로 정규화하는 과정이 포함되어 있는데, 이러한 정규화 과정(선형 또는 비선형)^[3,6,7]은 다중의 반복 루프에 의하여 처리되기 때문에 속도면에서 개선의 여지가 있다. 또한, 소프트웨어에 의하여 문자를 인식할 때에는 하나의 문자를 처리하는데 소요되는 전 과정이 호스트 컴퓨터에 의해서만 이루어지기 때문에 각 단계별 처리 과정이 누산된다. 따라서 소프트웨어에 의하여 문자 인식을 고속화 하는데에는 기본적인 한계를 가지고 있다. 본 연구는 이러한 한계를 개선하기 위하여 여러 단계의 인식 과정을 기능별로 분리시켜 호스트 컴퓨터와 병렬 구조로 구성함으로써 고속의 문자 인식기의 구현이 가능하도록 하였다. 그림 2는 문자인식 시스템을 하드웨어의 기능에 따라 분리하여 파이프라인 처리^[8,9]를 통해 시간적 병렬성(Temporal Parallelism)을 얻는 과정을 나타내었다.

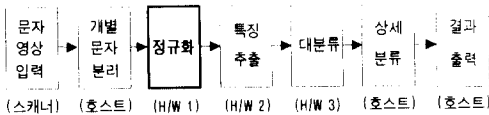


그림 1. 문자 인식 과정 및 하드웨어의 구성
Fig. 1. Character recognition process and its hardware.

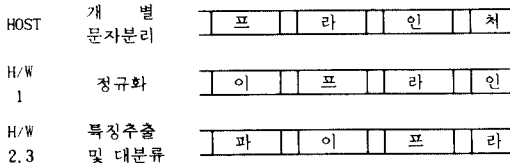


그림 2. 파이프라인 처리과정
Fig. 2. Pipeline process for character recognition.

문자영상의 정규화는 임의의 크기(H×W)를 갖는 2차원의 입력 영상을 일정한 크기(N×M)의 영상이

되도록 선형(또는 비선형)적으로 축소(또는 확대)시키는 것으로, 문자영상의 변화에 의한 문자 특징의 변화를 줄이기 위하여 이용되는데, 다중 크기의 글자체를 인식 대상으로 하는 패턴 매칭 알고리즘에서는 이러한 정규화 과정이 필수적이다.

II. 정규화 알고리즘

일반적인 정규화 알고리즘은 입력 영상을 일정한 크기의 영상으로 선형 변형시키는 선형 정규화와 입력 영상이 가지고 있는 특징을 이용하여 변형시키는 비선형 정규화가 있다. 본 연구에서 제안한 정규화 하드웨어는 범용 CPU를 사용하여 구현하였기 때문에, 소프트웨어에 의해 선형 혹은 비선형 정규화의 구현이 가능하다. 따라서, 일반적인 영상에 대해 특정한 특징을 필요로 하는 정규화 알고리즘에도 이용 가능하다.

1. 선형 정규화 알고리즘

본 논문에서는 인쇄 문자를 정규화 대상으로하여 비교적 간단한 선형 정규화 알고리즘을 이용하였다. 그림 3에서와 같이 크기가 W×H인 입력 영상을 N×N으로 정규화 할 때에는 식 1을 이용하여 입력의 h'는 h로, w'는 w에 위치시킴으로서 정규화가 이루어진다.

$$w = \frac{N}{W} \cdot w', \quad h = \frac{N}{H} \cdot h' \quad (1)$$

여기서 W:입력영상의 너비 H:입력영상의 높이 h:행방향 h'의 대응 위치 w:열방향 w'의 대응 위치

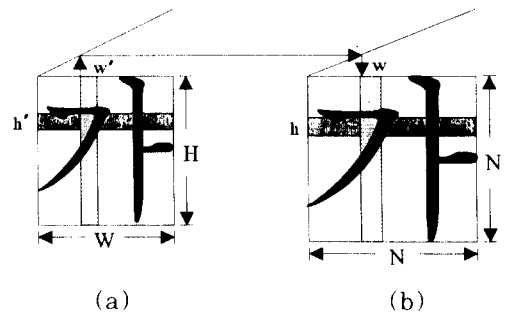


그림 3. 선형 정규화
(a) 입력 문자 영상(W×H)
(b) 정규화된 문자 영상(N×N)

Fig. 3. Linear normalization.
(a) Input character image.
(b) normalized character image.

2. 정규화 하드웨어 알고리즘

2.1절에 소개된 선형 정규화 알고리즘을 실현하기 위한 하드웨어에서의 처리 과정을 그림 4에 나타내었다. 먼저 입력 영상의 높이 정보를 이용하여 행 방향에 대해 정규화를 수행하는데, 영상을 확대할 경우에는 (식 1)에 의해 계산된 위치에 바로 이전 line을 삽입하고, 축소할 경우에는 해당 위치의 line을 삭제한다. 행 방향에 대한 정규화가 완료되면 영상을 종횡 변환 회로를 이용하여 90°로 회전시켜 열 방향에 대해 같은 방법으로 정규화를 수행한다^[10]. 종횡 변환 회로는 열방향의 정규화를 line 단위로 수행하기 위하여 이용된 하드웨어로서 전체 처리 속도의 개선에 많은 영향을 준다.

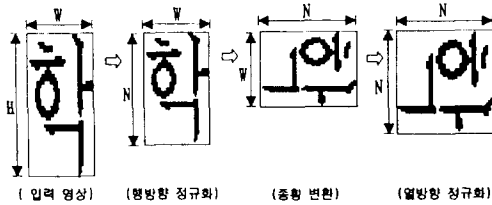


그림 4. 제안된 하드웨어에 의한 정규화 과정
Fig. 4. Normalization process by proposed hardware.

따라서, 소프트웨어에 의한 화소 단위의 정규화 과정에 비해 line 단위로 행(또는 열)을 삽입(또는 삭제)함으로써 정규화 속도를 향상시킬 수 있다. 이상의 과정을 거쳐 정규화가 완료된 문자영상으로부터 문자의 특징을 얻을 수 있도록 소영역으로 분할하여 특징 추출이 용이한 데이터의 형태로 변형시켜 특징 추출부에 출력한다.

Ⅲ. 정규화 하드웨어의 구현 및 실험

본 논문에서 구현한 정규화 하드웨어의 전체적인 구성도를 그림 5에 나타내었다. 그림 2에 제안된 파이프라인 처리 구조에서 살펴보면, 정규화 처리 과정이 호스트 컴퓨터에서 개별문자 영상을 추출하는 시간 이내에 처리가 완료된다면, 호스트 컴퓨터에서 수십 문자 분의 영상을 정규화 하드웨어로 전송한 후, 즉시 인식된 문자를 인출 할 수 있다. 따라서 호스트 컴퓨터의 개별문자 추출 시간내에 정규화 과정이 처리가 가능하면 전체적인 파이프라인 구조를 형성하는데 병목현상이 발생되지 않기 때문에, 이를 기준으로 대상 하드웨어의 속도를 설정하고, 이에 적용 가능한

범용의 CPU (MC68000, 12MHz)를 사용하여 정규화 하드웨어를 구현 하였다.

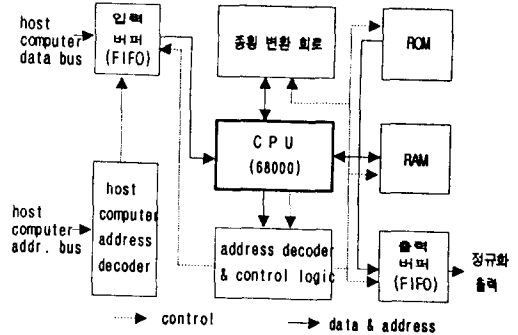


그림 5. 제안된 정규화 하드웨어 구성도
Fig. 5. Block diagram of proposed normalization hardware.

데이터의 통신은 호스트 컴퓨터의 입출력 포트를 이용하여 이루어 질 수 있도록 입출력 버퍼(FIFO : First In First Out)를 사용 하였으며, 데이터의 전송여부는 각 버퍼의 flag를 이용하여 판단할 수 있도록 하였다.

입력 버퍼는 호스트 컴퓨터에서 정규화 하드웨어로 데이터를 전송하기 위하여 사용되었다. 그림 6의 데이터 형식에 따라 호스트 컴퓨터에서 문자 영상의 너비와 높이 정보에 해당하는 개별 문자 영상 데이터를 FIFO에 기록(write)하면, 정규화 하드웨어에서 FIFO의 상태(flag)를 확인 한 후, 해당 문자 영상 데이터를 (식 2)에 계산된 크기만큼 읽어온다. (식 2)에서 행방향 워드의 수는 문자 영상의 너비 정보를 16-bit 단위로 표현한 값이고, 3은 문자의 header 정보이다. 이때 호스트 컴퓨터에서 분리된 하나의 행에 해당되는 개별 문자 (약 30-50자)를 정규화 회로로 전송할 수 있도록 입력 버퍼의 크기(32KByte)를 설정하였다.

$$\text{개별 문자 영상의 데이터 크기(words / 문자)} = \text{문자 높이} \times \text{행방향 워드의 수} + 3 \quad (\text{식} 2)$$

문자 너비(WIDTH)
문자 높이(HEIGHT)
행방향 워드의 수
영상 데이터

그림 6. 입력 데이터 형식
Fig. 6. Input data format.

ROM은 정규화 프로그램을 저장하고 있는데, 그림 4의 정규화 과정을 수행하기 위한 프로그램의 처리과정에 대한 순서도를 그림 7에 나타내었다.

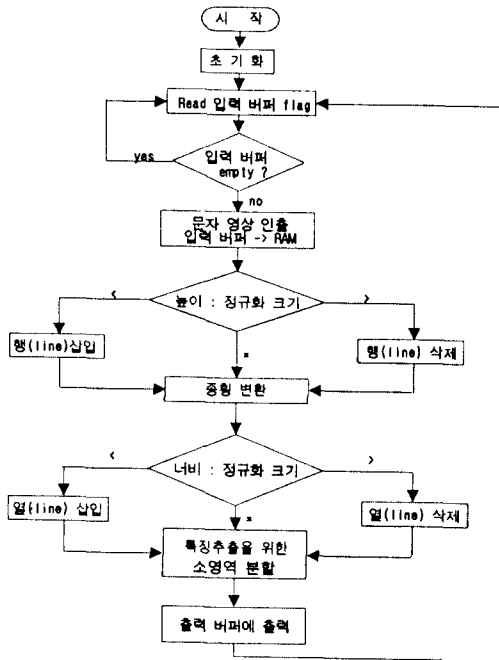


그림 7. 정규화 하드웨어의 내부 프로그램 순서도
Fig. 7. Flowchart of normalization hardware program.

RAM은 중간 처리 과정의 데이터를 저장하기 위하여 사용하였다. 문자 영상의 중형 변환회로는 DPRAM(Dual Port RAM)구조와 유사한데 내부적으로 어드레스 디코더와 플립플롭을 이용하여 16×16 크기의 영상 정보를 90° 회전 시키는 기능을 수행한다. 워드(16bits) 단위로 16개의 데이터를 내부의 플립플롭에 write한 후, read를 하면 90°로 회전된 데이터를 얻을 수 있다. 그러나 1차 행 방향으로 정규화가 행해진 문자이기 때문에 행의 크기는 N(42)개이고 열의 크기는 입력된 너비(W)로 구성되어 있기 때문에 이를 몇개의 16×16 단위의 크기로 나누어 중형 변환을 수행한다. 중형 변환 회로의 구성도를 그림 8에, 데이터의 처리 과정을 그림 9에 나타내었다.

정규화 회로중 중형 변환 회로와 어드레스 디코더, 제어 로직, 플립플롭 및 정규화 회로의 게이트 레벨의 모든 회로를 2개의 USIC(EPLD : EPM5192)을 이용하여 구현함으로써 높은 신뢰성과 간략화를 꾀하였다^[11,12].

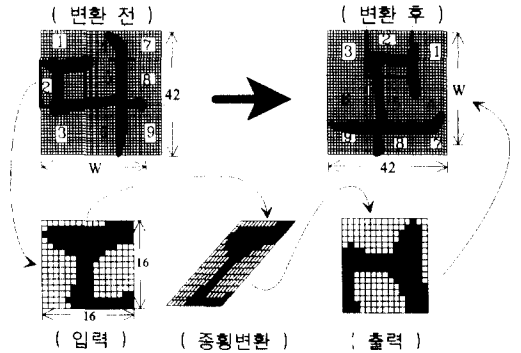


그림 8. 중형 변환 회로 구성도
Fig. 8. Block diagram for column-row rotating circuit.

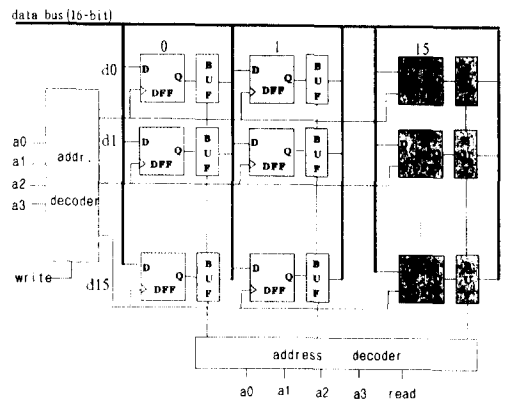


그림 9. 데이터의 중형 변환
Fig. 9. Data rotation process.

출력 버퍼는 정규화가 완료된 문자 영상 데이터를 특징 추출 하드웨어로 전송하기 위하여 사용하였고, 한 문자에 해당되는 영상에 대해 정규화가 완료되면 제어로직은 인터럽트 신호를 발생시켜 다음 단계의 하드웨어에 종료를 알린다. 인터럽트 신호를 인지한 다음 단계의 하드웨어(그림 1 참조)에서 정규화 데이터를 인출하기 시작하면 정규화 하드웨어는 다음 정규화 대상 문자를 입력 버퍼에서 인출하여 정규화를 수행한다. 이상에서 기술된 전체 하드웨어의 address map은 그림 10과 같다.

구현된 하드웨어에서 처리하는 정규화 크기는 입력 문자 영상의 크기에는 제한을 두지 않았으며, 정규화 후의 출력의 크기도 임의로 설정이 가능하나, 본 논문에서는 일반적으로 많이 통용되는 출력물을 300dpi로 스캔한 크기에 많은 변형을 가지지 않는 크기로 설정(42×42 화소)하여 실험 하였다. 실험을

2F0	DATA Write to 정규화 H/W
2F2	Status Read & Write
2F4	DATA Read from hardware

(a)

00000H	R O M
08000H	R A M
10000H	종횡변환 READ
18000H	종횡변환 WRITE
20000H	READ from 입력 버퍼
28000H	WRITE to 출력 버퍼
38000H	STATUS READ

(b)

그림 10. 정규화 하드웨어의 address map

- (a) 호스트 컴퓨터의 입출력 포트.
- (b) 정규화 하드웨어

Fig. 10. Address map of normalization hardware.

- (a) I/O port of host computer.
- (b) normali-zation H/W.

통해 호스트 컴퓨터(IBM PC486-33MHz)에서 정규화에 소요되는 평균 시간과 구현된 하드웨어에 의해 수행된 소요 시간을 표 1에 나타내었으며, 병렬 처리를 위한 소요 시간의 비교를 위해 호스트에서 개별 문자 추출에 소요되는 시간을 함께 기록하였다.

실험 결과로 부터 하나의 개별 문자에 대하여 평균적으로 호스트 컴퓨터에서 정규화하는데에는 약 19.2ms가 소요되었고, 구현된 하드웨어에 의한 처리 시간은 약 2.2ms로 처리 속도가 개선되었다. 또한 호스트 컴퓨터에서 개별 문자를 추출하는데 소요되는 시간(3.6 ms) 이내에 하드웨어에서 정규화 작업이 완료됨으로써 호스트 컴퓨터에서의 정규화 시간인 19.2ms를 줄일 수 있다.

표 1. 처리 시간의 비교

Table 1. Comparison of processing time.

구 분	호스트 컴퓨터 (IBM PC)	구현된 하드웨어
정규화	19.2 ms	2.2 ms
개별 문자 추출	3.6 ms	---

IV. 결론

영상처리 분야에 있어서 임의 크기의 입력 영상에 대하여 원하는 크기로 축소 및 확대가 가능한 고속의 정규화 하드웨어를 구현하였다. 특히 이진 영상 처리 분야인 문자 인식 알고리즘의 개발에 있어서 각 처리 단계를 고속화 시킨 후, 이를 상호간의 파이프라인 구조로 구성함으로써 시간적 병렬성을 갖는 고속의 문자 인식기의 구현 가능성을 제안하였으며, 그 일환으로 정규화 하드웨어를 구현하였다.

구현된 하드웨어는 정규화 크기는 임의의 입력에 대해 42×42 화소로 하였다. 호스트 컴퓨터(IBM PC486-33MHz)에서 개별 문자 추출 및 정규화 과정에서 소요되는 평균 처리시간과 구현된 하드웨어에 의해 수행된 소요시간을 측정한 결과 호스트 컴퓨터에서 개별 문자를 추출하는 시간 이내에 정규화 작업이 완료됨으로써 호스트 컴퓨터의 정규화 시간을 절감할 수 있다. 따라서 본 논문에서 제안한 전체적인 파이프라인 구조를 갖는 시스템 구성에 사용 가능성을 보였다.

參 考 文 獻

- [1] OCR의 기술동향 및 인식기술. 산업연구원, 1989.
- [2] 김진형 외, "문서 인식 및 처리기의 개발에 관한 연구", 연구보고서, 한국 과학 기술원, 1989. 4.
- [3] 김덕진 외, "병렬처리 기술을 이용한 인쇄문자 인식기의 구현에 관한 연구", 연구보고서, 고려대학교 정보통신기술공동연구소, 1992. 6.
- [4] Michio Umeda, "Recognition of Multi-font Printed Chinese Characters", CH1801-0/82/0000/0793 IEEE, pp.793-796, 1982.
- [5] 孫寧, 田原秀, 阿曹弘具, 木村正行, "方向

線素特徴量を用いた 高精度文字認識”, 電子情報通信學會論文誌 Vol.J74-D-II No.3 pp. 330-339, 1991. 3.

[6] H.Yamada, T.Saito, K.Yamamoto, “Line Density Equalization - A Nonlinear Normalization for Correlation Method”, Trans. IECE of JAPAN, Vol. J67-D, No.11, pp.1379-1383, 1984.11.

[7] 김 봉석, “선과 점 밀도를 이용한 문자 비선형 정규화 및 인식”, 석사학위 논문, 고려대학교 대학원, 1992.

[8] 酒卷 久 et. al., “認識速度 70字/秒の日本語 OCR.専用 LSIとDSPの パイプライン處理で高速化”, キヤノン ソフトウェア戦略本部, NIKKEI ELECTRONICS, pp.195-201, 1990. 7.

[9] 弘具, 大町眞一郎, 木村 正行, 勝山 裕, “高速高精度知的認識システム SEIUN”, 電子情報通信學會論文誌 Vol.J76-D-II No.3 pp. 474-484 1993. 3.

[10] Shin 'ichiro OHMACHI, Hirotomoto ASO and Masayuki KIMURA, “A Systolic Array for Preprocessings for Fast Character Recognition”, 電子情報通信學會論文誌 Vol. J 73-D-II No.2 Februar 1990.

[11] MOTOROLA, MC68000 COURSE NOTES.

[12] ALTERA, DATA BOOK, 1992.

著者紹介



金 赫 九(正會員)

1961年 3月 19日生. 1984年 2月 금오공대 전자공학과 졸업(공학사). 1990年 8月 고려대학교 산업대학원 전자통신공학과 졸업(공학석사). 1994年 2月 고려대학교 대학원 전자공학과 졸업(공학박사).

현재 고려대학교 부설 정보 통신 기술공동연구소 연구원. 주관심 분야는 문서인식, 마이크로프로세서 응용 및 영상처리 등임.



姜 仙 美(正會員)

1959年 7月 28日生. 1981年 2月 고려대학교 전자공학과 졸업(공학사). 1988年 6月 에얼랑젠-뉘렌베르그 대학교 전자공학과 졸업(Diplom). 1992年 8月 고려대학교 대학원 전자공학과 졸업(공학박사).

1992年 11月 ~ 1994年 2月 고려대학교부설 정보.통신기술공동 연구소 연구조교수. 현재 고려대학교 산업대학원 객원조교수. 주관심 분야는 패턴인식, 영상처리 분야 등임.

金 惠 鎮(正會員) 第 29卷 A編 第 8號 參照

현재 고려대학교 전자공학과 교수