

모바일 환경의 OCR Anyword Anyword OCR in Ubiquitous Computing

박종경, 음봉규, 권용식, 진성아
성결대학교 멀티미디어학부

Park Jong-Kyeong, Eum Bong-Kyu, Kwon Young-Sik,
Chin Seong-Ah
Sungkyul Univ.

요약

최근 모바일기기에 유비쿼터스 콘텐츠를 구현하는 시도가 활발히 진행되고 있다. 핸드폰을 이용해 위치를 판단한다거나, 핫코드를 찍어 상품을 구매하는 등의 콘텐츠가 개발되었다. 또한, 모바일기기도 발전하여 핸드폰, PDA 같은 모바일기기의 내장 카메라모듈은 필수사항이 되었다. 본 연구는 모바일기기의 내장 카메라모듈을 이용한 모바일 환경에 적합한 한글 문자인식 시스템을 제안한다. 본 연구의 시스템은 모바일기기로 PDA를 사용하였으며, PDA의 카메라모듈을 통하여 인쇄체 한글 영상을 입력받고, 모바일기기의 느린 연산속도를 보완하기 위하여, 서버로 이미지와 기율기 정보를 전달한 후, 서버에서 기울어진 인쇄체 문자영상을 보정하고, 프로젝션을 통해 문자를 추출한 후, 차연산을 이용한 매칭 방법으로 인쇄체 한글을 인식한다. 인식한 문자들은 사용자의 수정을 거쳐 텍스트 문서로 저장할 수 있다.

I. 서론

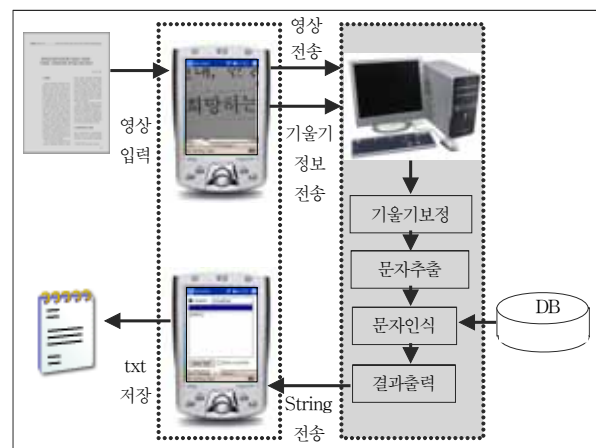
국내에서 2001년부터 보급되기 시작한 모바일기기의 내장 카메라 모듈은 급속도로 보급되어 2005년 현재는 거의 모든 모바일 기기가 내장 카메라 모듈을 가지고 있다. 그 기능 역시 눈부신 발전을 이루어, 10만 화소에서 500만화소로, 고정초점방식에서 오토포커스방식으로 지속적인 발전을 보이고 있다. 모바일 기기의 영원한 과제 중 하나가 입력 장치에 관한 부분이다. 특히 PDA는 큰 화면을 제공하기 위해 터치스크린 방식을 채택하는 대신, 버튼을 줄였는데, 이 때문에 문자입력 속도가 핸드폰이나 노트북에 비해 느리다는 단점을 지니게 되었다.

따라서 본 논문에서는 Rubber Band기법, 양선형 보간법과 적응적 이진화, 프로젝션을 이용한 문자추출법, 세선화를 활용한 차연산 방법을 이용하여 OCR기술을 PDA에 적용하여 구현하였다.

II. 시스템 구현

1. 시스템 구성

본 시스템은 장치에 따라 두 부분으로 나눌 수 있다. 첫 번째 부분은 이미지를 입력하고 결과값을 출력하는 PDA Client 부분이고 두 번째 부분은 OCR연산을 수행하는 PC Server 부분이다. 두 장치는 TCP/IP 통신으로 데이터를 주고받는다.



▶▶ 그림 1. 시스템 구성도

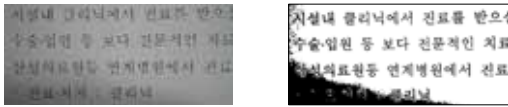
PDA에 Stand alone 방식으로 개발도 가능하지만, 현재 시판된 PDA(500MHz)는 PC(3GHz) 보다 OCR 연산 능력이 5분의 1 수준이며 결과도 만족스럽지 못하여, 서버/클라이언트 방식으로 시스템을 구현하였다. 아래 그림 1.에서는 흰색 배경은 PDA Client를 나타내고, 음영처리 된 순서도 부분은 PC Server 부분을 나타낸다.

2. 이진화 기술

컴퓨터가 문제를 디지털 정보로 인지하게 만드는 이진화 기술은 OCR기술에서 핵심 기술 중 하나로, 이를 수행하기 이전에 먼저 이미지를 회색조로 만들어 주어야 한다. 이를 위해 쉘

* 본 연구는 학술진흥재단 이공계교육과정개발 연구지원사업 (KRF-2005-081-D00012)에 의해 수행되었음

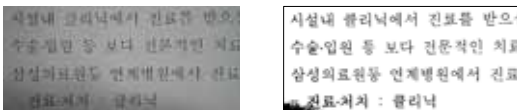
러 영상으로부터 회색조로 변환한 후 임계치를 이용하여 이진화 영상을 획득하였다.



가. 적응적 이진화 전 나. 적응적 이진화 후

▶▶ 그림 2. 적응적 이진화 적용의 예

본 논문에서는 시스템적으로 평균적 이진화외에도 사용자가 90, 120, 200의 임계값을 따로 설정하는 방법을 제안한다. 이 방법을 이용하면 이미지에 맞는 임계값을 적용하기가 용이하게 된다. [그림 3]은 밝음(90), 보통(120), 어두움(200) 세 가지 모두를 적용하여 이진화한 결과를 보여준다.



가. 이진화 전 나. 어두움 (임계값 90)



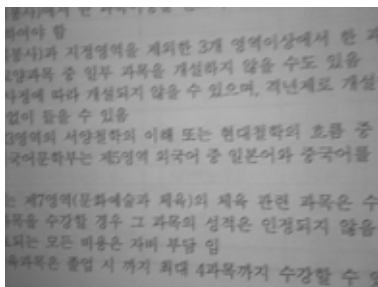
다. 보통 (임계값 120) 라. 밝음 (임계값 200)

▶▶ 그림 3. 3가지 설정에 따른 이진화의 예

3. 기울기 보정

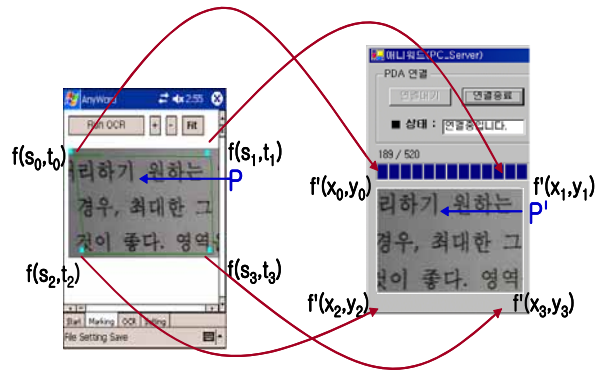
모바일 카메라 모듈을 통해 획득한 문서 이미지의 대부분이 다양한 형태의 사변형으로 왜곡되어 [그림 4]와 같이 OCR에 사용하기 어려운 형태를 지니고 있다. 따라서 사변형을 직사각형으로 변환해주는 전처리 과정이 필요하다. 본 시스템에서는 양선형 변환 알고리즘을 사용하여 획득한 사변형 영상을 직사각형 영상으로 변환하였다[1].

본 시스템에서는 양선형 변환 알고리즘을 모바일환경에 적합하게 재구성했다. 기존의 방식은 연결화소를 이용하여, 근접 문자들간의 관계를 규명하므로써 사변형의 꼭지점 4개를 찾아 내었다[1].



▶▶ 그림 4. 모바일 카메라로 획득한 기울어진 영상의 예

하지만, 이 방법을 PDA에 적용할 경우 연산시간이 오래 걸린다. 따라서, 본 시스템에서는 사용자가 OCR 연산을 원하는 사변형의 4개에 꼭지점을 스타일러스 펜을 이용하여 Rubber Band 방식으로 끌어 옮겨 설정하도록 했다. 이 방식을 사용하면 PDA에서는 별도의 연산을 하지않고 사용자가 OCR 연산을 원하는 영역만 PC에게 전달하여 연산을 수행함으로 수행시간을 단축할 수 있다.



사변형(f) ← 직사각형(f')

▶▶ 그림 5. 기울기 보정의 예와 좌표

[그림 5]에서 $f(s_0,t_0) \sim f(s_3,t_3)$ 는 변환 전 사변형의 네 꼭지점의 좌표이고 $f'(x_0,y_0) \sim f'(x_3,y_3)$ 는 변환 후 직사각형의 네 꼭지점 좌표이다. 보정하여 PC에서 연산에 사용할 직사각형의 크기는 보정 전 사변형의 네 꼭지점 $f(s_0,t_0) \sim f(s_3,t_3)$ 에 값들 중에서 최대값을 사용한다. PDA에서 보내준 최대값으로 만들어진 직사각형 f' 에 들어갈 컬러 정보를 구하기 위해 임의의 점 P' 의 좌표에 해당하는 사변형 f 의 임의의 점 P 의 좌표를 구한다. 수식(1)~수식(8)을 이용하여 변환후의 좌표 $P'(x,y)$ 에 해당하는 변환전의 좌표 $P(s,t)$ 를 구한다. 좌표 $P(s,t)$ 의 컬러 정보값을 다시 좌표 $P'(x,y)$ 에 넣어주면 기울어짐 보정이 완료된다.

$$s = s_{01} + (s_{23} - s_{01}) \times dy \dots\dots\dots (1)$$

$$t = t_{03} + (t_{12} - t_{03}) \times dx \dots\dots\dots (2)$$

$$s_{01} = s_0 + (s_1 - s_0) \times dx \dots\dots\dots (3)$$

$$s_{23} = s_3 + (s_2 - s_3) \times dx \dots\dots\dots (4)$$

$$t_{03} = t_0 + (t_3 - t_0) \times dy \dots\dots\dots (5)$$

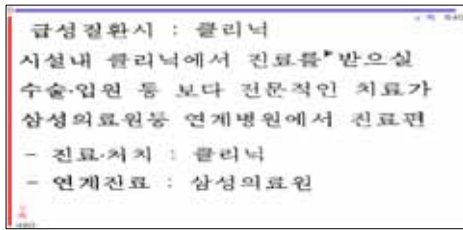
$$t_{12} = t_1 + (t_2 - t_1) \times dy \dots\dots\dots (6)$$

$$dx = \frac{x - x_0}{x_1 - x_0} \dots\dots\dots (7)$$

$$dy = \frac{y - y_0}{y_3 - y_0} \dots\dots\dots (8)$$

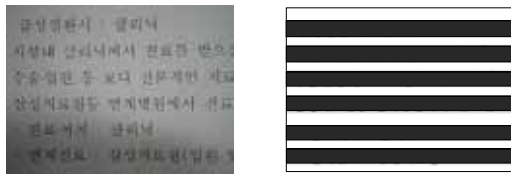
4. 문자추출

본 시스템에서는 문자추출을 위해 프로젝션 방식을 사용한다. [그림 6]과 같이 이미지의 X축, Y축에 방향으로 프로젝션을 적용하여 검은 픽셀이 존재하는지를 검사한 후 문자를 추출하게 된다.



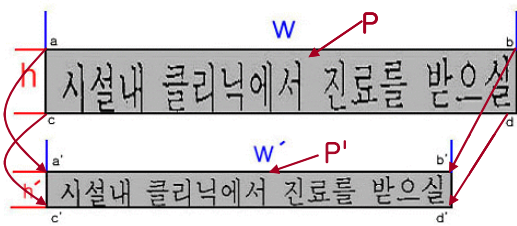
▶▶ 그림 6. 프로젝션을 확대 적용

이러한 프로젝션은 3번의 절차를 거치게 되는데, 첫 번째로 아래 [그림 7]과 같이 세로방향으로 프로젝션을 적용하여 대상 이미지에 글자가 존재하는 행을 검출하게 된다.



가. 적용 전 나. 적용 후
▶▶ 그림 7. 세로방향 프로젝션

두 번째 절차로 각 줄을 매칭에 사용하는 DB 이미지의 높이인 25픽셀에 맞추어 축소하게 된다.



▶▶ 그림 8. 잘라진 한 줄의 이미지와 축소의 예

이를 위해, [그림 8]과 같이 자른 한 행의 이미지에 가로, 세로 값을 각각 w, h로 지정하고 줄이고자 하는 목표 사이즈인 h'과 w'을 구해야 한다. 이때, h'의 크기는 25 Pixel로 고정하는데, 그 이유는 매칭에 사용할 DB의 세로가 25 Pixel이기 때문이다. 이미지를 축소하기 위한 가로, 세로 비율은 수식(9)와 수식(10)을 이용하여 구한다.

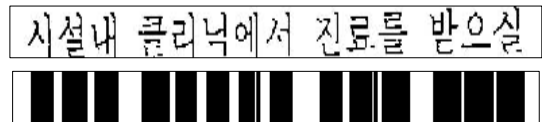
$$\alpha = h' / h \tag{9}$$

$$\beta = w' / w \tag{10}$$

원본 이미지의 임의의 점을 P라 하고 축소되어진 이미지의 임의의 점을 P'이라고 했을 때, 수식(9)와 (10)에 의해 구해진 가로의 비율 α 와, 세로의 비율 β 를 이용하여, 원본 이미지를 축소한다. 임의의 점 P를 축소하는 수식은 수식(11)의 축소 공식과 같다.

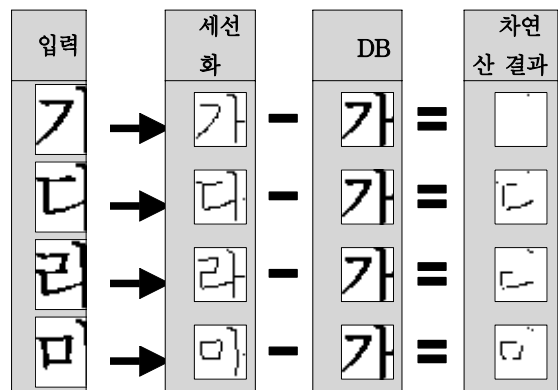
$$P' = \begin{bmatrix} x & 0 & 0 \\ 0 & y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix} \dots\dots\dots \tag{11}$$

[그림 8]과 같이 각 줄에 대해 축소작업을 모든 줄에 적용하면, 이미지안의 모든 줄의 글자가 표준 높이로 고정되게 된다. 그리고 세 번째 절차로 [그림 9]와 같이 가로 방향으로 프로젝션을 적용하여 문자의 정확한 위치를 알아 낼 수 있다. 찾아낸 위치로 문자를 추출하고 라벨링 한 후, 매칭 시 인식률을 높이기 위해 여백을 잘라낸다.



▶▶ 그림 9. 각 행에서 문자를 추출하는 예

5. 차연산을 이용한 문자인식



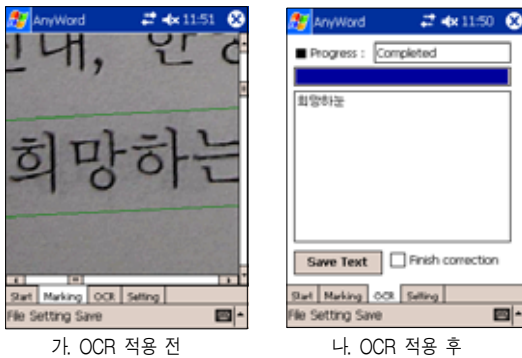
▶▶ 그림 10. 차연산이 수행되는 예

기존의 방식들은 한글의 구조적 특징을 분해하여 문자를 찾아내는 방식을 사용하였는데, 본 시스템에서는 차연산을 사용하여 문자를 매칭한다. 원본 이미지에서 기울기 보정과 프로젝션, 세션화 과정을 거쳐 얻은 각 문자에 대해 크기를 맞춘 후 [그림 10]과 같이 DB와 차연산을 수행하면 일치하지 않는 화소들이 남게 되는데, 남은 검정 화소의 합이 정확도가 된다. 세션화 된 이미지가 원본 이미지와 유사할수록 더 적은 화소가

남게 되므로, 정확도는 수치가 낮을수록 입력 영상과 일치하는 문자가 된다.

III. 실험 결과 및 고찰

[그림 11]과 같이 사용자의 입력에 따라 OCR을 실시하면 PC를 거쳐 연산한 후 PDA에 결과값이 출력된다.



▶▶ 그림 11. 세로방향 프로젝트션

실험은 한글 빈도순 상위 520자[7]에 바탕체를 대상으로 640×480Pixel 크기의 이미지에서 각기 다른 5가지 형태의 영역을 가지고 테스트한 결과, 속도는 전처리에서 20초, 매칭에서 글자당 2초의 시간이 소요되었다. 기울기의 경우 20도를 넘어서면 잡음이 생겨 OCR 결과에 영향을 주므로, 사용자에게 재촬영을 요구하는 것이 효과적이었다.

IV. 결론 및 향후 연구

본 연구는 TCP/IP를 활용하여 PDA의 연산 속도를 PC로 보완한 한글 OCR시스템을 제안하였다. 지금까지 제안된 모바일 환경의 OCR시스템은 기울어진 문서에 대한 인식률이 떨어지고, 모바일 환경에서 연산 오래 걸리는 문제점을 가지고 있었다. 본 연구의 시스템의 양선형 기울기 보정과 TCP/IP 시스템은 이러한 문제점을 보완하였다. 하지만, 차연산을 이용한 문자인식이 노이즈에 민감하여 이점을 SUV를 이용한 방법으로 보완하고, 모바일 환경에 적합한 이진화, 기울기 보정기술을 연구할 계획이다.

■ 참고 문헌 ■

[1] 장대근, 전병태, “카메라 문서 영상의 이진화 및 기울어짐 보정 방법”, 한국컴퓨터정보학회 제10권 제3호, pp.143-150, 2005.

- [2] J.R.Parker "Algorithms For Image Processing And Computer Vision", Wiley, 1996.
- [3] 박현일, 김수형, “휴대폰 카메라로 획득한 저해상도 영상에서의 전화번호 인식”, 제 31회 정보과학회 춘계학술대회, 제B권, pp.691-693, 2004.
- [4] 이성환, “문자인식 :이론과 실제 1, 2권”, 홍릉과학출판사, 1994.
- [5] 전종익 외, “한글 shape문자 pattern 에서의 구조적 정보를 이용한 형식 분류와 인식에 관한 연구”, 한국통신학회 논문지 제 16권 2호 pp.180-195, 1991.
- [6] 조성배, 김진형, “인쇄체 한글문자의 인식을 위한 계층적 신경망”, 한국정보과학회 논문지 제 17권 제 3호 pp.306-316, 1990.05
- [7] 최기선, “국어 정보 베이스”, KAIST, <http://kibs.kaist.ac.kr/>, 2000. 07