Client/Server구조를 이용한 PDA기반의 문자 추출 시스템

(PDA-based Text Extraction System using Client/Server Architecture)

박 안 진 † 정 기 철 ††
(Anjin Park) (Keechul Jung)

요 약 최근, PDA를 이용한 모바일 비젼 시스템에 관한 많은 연구가 진행되고 있다. 대부분의 PDA에서 사용하는 CPU는 실수 연산 구성요소(floating-computation component)가 없는 정수(integer)형 CPU를 사용하므로, 실수 연산이 많은 영상 처리 및 비젼 시스템에서는 많은 시간이 소요되는 단점이 있다. 본 논문에서는 이를 해결하기 위해 무선 랜(LAN)으로 연결된 Client(PDA)/Server(PC)구조를 이용한 시스템을 제안하며, 연속 영상에서 Client(PDA)와 Server(PC) 각각의 CPU를 이용하여 파이프라이닝 형식으로 시스템을 구축함으로써 수행 시간을 단축한다. Client(PDA)는 에지 밀도(edge density)를 이용하여 대략적인 문자 영역을 추출하며, Server(PC)는 Client(PDA)에서 대략적으로 검출된 결과를 바탕으로 정밀한 문자 영역을 추출하기 위해, MLP(multi-layer perceptron) 기반의 텍스춰 분류 방법과 연결 성분(connected component:CC) 기반의 필터링 방법을 이용한다. 본 실험에서 제안한 방법은 MLP와 CC를 이용함으로써 효과적인 문자 추출 결과를 보였으며, 파이프라이닝 형식의 Client(PDA)/Server(PC)구조를 이용함으로써 빠른 수행 시간을 보였다.

키워드 : 모바일 비젼(PDA), 문자 추출, 다층 신경망, 텍스춰, 연결 성분

Abstract Recently, a lot of researches about mobile vision using Personal Digital Assistant(PDA) has been attempted. Many CPUs for PDA are integer CPUs, which have no floating-computation component. It results in slow computation of the algorithms performed by vision system or image processing, which have much floating-computation. In this paper, in order to resolve this weakness, we propose the Client(PDA)/Server(PC) architecture which is connected to each other with a wireless LAN, and we construct the system with pipelining processing using two CPUs of the Client(PDA) and the Server(PC) in image sequence. The Client(PDA) extracts tentative text regions using Edge Density(ED). The Server(PC) uses both the Multi-Layer Perceptron(MLP)-based texture classifier and Connected Component(CC)-based filtering for a definite text extraction based on the Client(PDA)'s tentatively extracted results. The proposed method leads to not only efficient text extraction by using both the MLP and the CC, but also fast running time using Client(PDA)/Server(PC) architecture with the pipelining processing.

Key words : Mobile Vision(PDA), Text Extraction, MLP, Texture, Connected Component(CC), CAMShift

1. 서 론

언제, 어디서, 누구나 대용량 네트워크를 사용하는 유비쿼터스(ubiquitous) 시대가 다가오면서, 카메라가 장착된 무선 통신이 가능한 PDA(personal digital assistant)와 같은 모바일 기기가, 가까운 미래에는 일상의 한 부분이 될 것이다.

최근에는 디지털 카메라를 이용한 자연 영상에서의 문자 추출 및 인식에 관한 연구가 많이 진행되고 있다. 특히 모바일 기기에 장착된 카메라는 작고, 가벼우며, 네트워크와 쉽게 연동할 수 있다. 이런 장점으로 인하여, 모바일 기기를 이용한 자연 영상에서의 문자 추출에

· 이 논문은 2003년도 한국학술진흥재단 지원에 의하여 연구되었음 (KRF-2003-041-D00526)
† 학생회원 : 숭실대학교 미디어학부
anjin@ssu.ac.kr
†† 종신회원 : 숭실대학교 미디어학부 교수
kcjung@ssu.ac.kr
논문접수 : 2004년 8월 18일
심사완료 : 2004년 11월 19일

관한 연구가 다양한 분야에서 진행되고 있으나[1-5], 양질의 문서의 OCR(optical character recognition)과는 달리 상당히 어려운 문제로 인식되고 있다.

모바일 기기에서 입력 받은 영상은 자연 영상이며, 기존의 자연 영상에서 문자를 찾는 방법(*text detection*)은 크게 문자 검출(text region localization) 방법과 문자 추출(배경과 분리:text extraction) 방법으로 나눌 수 있다. 문자 검출 방법으로 Zhang[1] 등은 PDA 카메라를 이용하여 표지판(sign)에 있는 간단한 문자를 영어로 번역하는 시스템을 제안하였으며, 문자 검출을 위해 다중-해상도(multi-resolution)을 이용하였다. Li[2] 등은 디지털 카메라에서 입력 받은 영상에서의 문자 검출을 제안하였으며, 연결 성분의 MBR(minimum bounded rectangle)의 배치 분석(alignment analysis)을 이용하였다. 최영우[3] 등은 카메라에서 입력 받은 자연영상에서의 문자 검출을 제안하였으며, 문자의 색 연속성, 밝기 변화 및 색 변화와 같은 낮은 수준의 영상 특징을 이용하여 문자 후보 영역을 찾고, 다중-해상도 웨이블릿(wavelet) 변환을 이용하여 높은 수준의 문자 특징인 획의 구성여부로 검증하는 계층적인 구조를 이용하였다.

문자 추출 방법으로 *Watanabe*[4] 등은 PDA를 이용하여 간판, 표지판과 같은 영상 내의 문자를 영어로 번역하는 시스템을 제안하였으며, 사용자가 PDA에서 직접 문자 영역을 검출하고 기존의 히스토그램을 이용한 자동 임계값 선택 방법(threshold selection method)을 이용하여 문자를 추출한다. Haritaoglu[5]는 PDA 카메라를 이용하여 영상 내의 문자를 영어로 번역하는 시스템을 제안하였으며, PDA에서 문자 영역을 사용자로부터 입력 받고, 문자를 배경과 분리하는 방법으로 SNF (symmetric neighborhood filter)를 이용한 영상 향상(enhancement)과 계층 연결 성분(hierarchical connected component)을 이용한 영상 분할(segmentation) 방법을 이용하였다. 박현일과 김수형[6]은 핸드폰 카메라를 이용하여 입력 영상에서 전화번호를 추출하는 시스템을 제안하였다. 전화 번호의 위치는 이미 알고 있다는 조건 하에 배경에서 전화번호를 추출하였으며, 전화번호를 배경과 분리하는 이진화 방법으로 필터링(*filtering*)과 클러스터링(clustering)을 이용하였다.

본 논문은 PDA를 이용하여 문자 추출을 수행하는데, PDA는 일반적인 PC와 달리 크게 두 가지 문제점을 가지고 있다[1].

첫째, 제한된 연산 자원이다. 대부분의 PDA는 실수 연산 구성요소(floating-computation component)가 없는 정수(integer)형 CPU를 사용하며, 실수 연산을 수행하기 위해서는 실수 대행 라이브러리(*float emulation* library)를 사용하므로 오랜 수행 시간이 소요된다. 예를 들어, 본 논문에서 사용한 MLP(multi-layer perceptrons) 기반의 텍스춰(texture) 분류 방법을 이용하여 240×320 영상에 대해 PDA에서 문자 추출을 수행하면 4분 정도의 수행 시간이 소요되며, 이는 본 실험에서 사용한 PC에서 수행한 시간에 비해 120배 느리다.

둘째, 제한된 저장 공간(memory)이다. 예를 들어, 본 논문에서 사용한 Pocket PC 2002 기반의 PDA는 16~64MB 저장공간을 가지며, 일반적인 Palm OS 기반의 PDA는 보통 8~16MB 저장공간을 가진다. 이 공간은 프로그램과 데이타의 저장공간을 함께 사용하며, 문자 추출 시스템과 OCR을 수행하기에는 너무나 작은 공간이다.

이런 문제점을 해결하기 위해 Zhang[1] 등은 실수 연산을 일반화된 정수 연산으로 바꾸어 문자를 검출하였다. 그러나, 이 방법은 수행속도는 빨라졌으나 정확도의 손실을 가져왔다. Haritaoglu[5]는 이런 문제를 해결하기 위해 부동 연산이 많이 수행되는 정밀한 문자 추출을 Server(PC)에서 수행하는 방법을 이용하였다. 그러나 이 방법은 PDA의 연산 자원을 이용하지 않으며, PDA는 단지 Server의 주요 프로세스를 도와 주기 위한 입력과 출력 역할만을 담당하였다.

우리는 제한된 연산 자원, 제한된 저장 공간과 같은 PDA의 문제점을 해결하기 위해, Client/Server구조를 이용한 문자 추출 시스템을 제안한다(그림 1). 연속 영상[1]에서 효과적으로 문자를 추출하기 위해, Server(PC)의 프로세스와 Client(PDA)의 프로세스를 시간적으로 중첩되게 수행하고[2], Server(PC)의 프로세스가 끝나면 바로 다음 프로세스(Client(PDA)의 결과를 이용한)를 수행하는 파이프라이닝 방식의 시스템을 구축한다. Client(PDA)는 Client(PDA)/Server(PC) 사이의 전송 시간을 줄이기 위해, 에지 밀도(edge density:ED)를 이용하여 검출된 대략적인(tentative) 문자 영역을 JPEG 포맷(format)으로 압축하여 Server(PC)에 전송한다. Server(PC)는 Client(PDA)의 결과를 바탕으로 정밀하게(definite) 문자를 추출한다. 우리는 MLP 기반의 텍스춰 분류 방법과 연결 성분(connected component: CC) 기반의 필터링 방법을 동시에 이용하며, 각각의 장점을 이용하여 단점을 보완한다. MLP 기반의 텍스춰 분류 방법은 문자와 비-문자 영역을 분류하는 텍스춰 분석기를 생성하기 위해 MLP를 이용함으로써, 문자의 크기나 모양, 배경과 같은 다양한 환경에서 적응성을 높일 수 있으며, 검출률(recall rate)을 향상시킬 수 있는

1) 단일 영상에서는 Client(PDA)와 Server(PC) 프로세스를 순차적으로 수행한다.

2) Client(PDA)와 Server(PC)는 각각 CPU를 가지고 있기 때문에 동시에 프로세스를 수행한다.

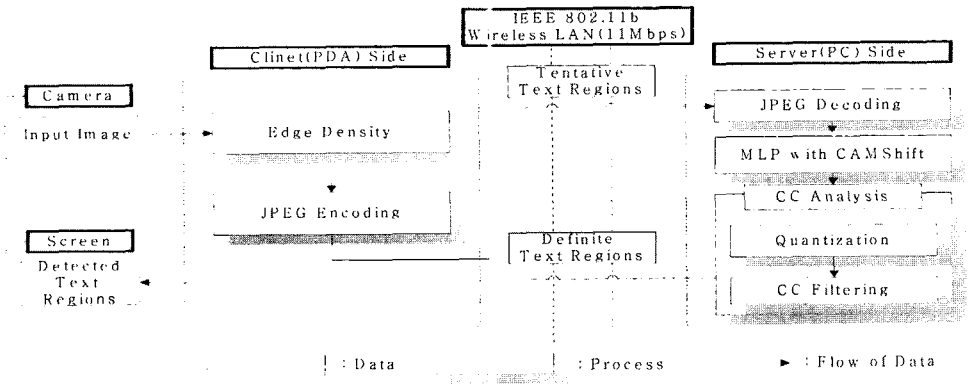


그림 1 제안된 문자 추출 시스템의 전체 흐름도

장점을 가지고 있지만, 이에 따른 불가피한 오 추출(false alarm)의 증가로 정확도(precision rate)가 떨어지는 단점을 가지고 있다. 이런 단점을 해결하기 위해 연결성분 기반의 필터링 방법을 이용하여 정확도를 적정선으로 유지할수 있다. 또 다른 단점으로 MLP의 결과는 문자의 위치만을 나타내며, 이 역시 연결성분 기반의 필터링 방법을 이용하여 문자를 배경과 분리하는 문제를 해결할 수 있다. 이와 같이 MLP와 CC 방법을 동시에 이용하면 검출률과 정확도를 향상시킬 수는 있지만, 수행시간이 증가하는 단점이 있다. 이러한 수행시간의 대부분은 MLP 수행 단계에서 사용되며, 본 논문에서는 MLP 수행 단계에서의 많은 계산량으로 인한 속도 저하를 CAMShift 알고리즘을 이용하여 불필요한 부분에 대한 처리를 하지 않음으로써 해결한다.

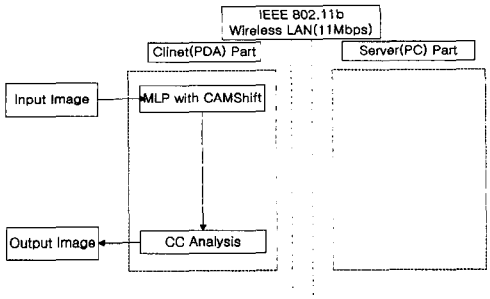
본 논문의 구성은 다음과 같다. 제 2장에서 Client(PDA)와 Server(PC)사이의 시스템 구조에 대해 기술하고, 제 3장에서는 Client(PDA), 제 4장에서는 Server(PC)에서의 문자 추출 방법을 기술하며, 제 5, 6장에서 실험결과 및 향후 연구 방향을 기술한다.

2. 시스템 구조(System Architecture)

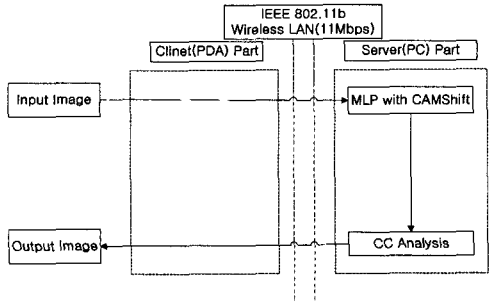
우리는 PDA의 단점을 해결하기 위해 Client/Server 구조를 이용하며, 연속 영상에서 Client(PDA)와 Server(PC)의 자원을 최대한 활용하여 효과적으로 문자를 추출하기 위해 파이프라이닝 방식으로 시스템을 구축한다.

2.1 Client/Server구조

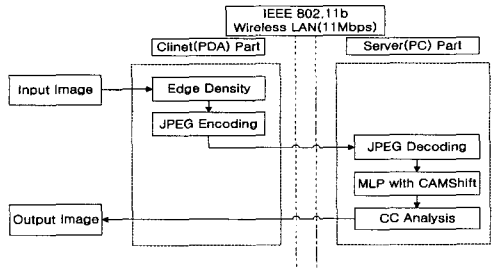
PDA를 이용하여 영상 처리와 컴퓨터 비젼을 수행하는 시스템은 크게 두 가지 방법으로 분류된다. 첫째, PDA만으로 시스템이 구성된 방법이다[1,7](그림 2(a)). 이 방법은 PDA에서 모든 일을 처리하며 앞에서 언급한 PDA의 문제점으로 인하여 수행 시간 및 저장 공간에서 큰 문제점이 발생하였다. 둘째, PDA가 입력과 출력만을



(a) Client(PDA)만으로 구성된 방법



(b) Client(PDA)가 입력과 출력만을 담당하는 방법



(c) 제안된 방법.

그림 2 PDA를 이용한 영상처리 및 컴퓨터 비젼을 수행하는 시스템의 유형

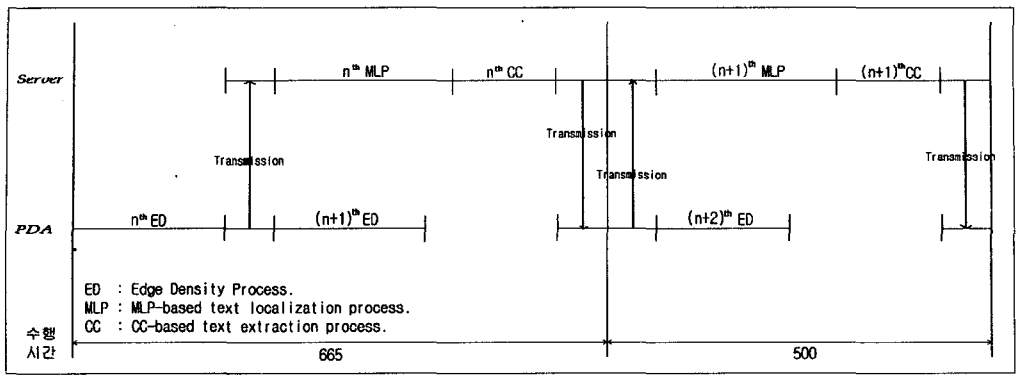


그림 3 제안된 시스템의 파이프라이닝 처리

담당하는 방법이다[4,5](그림 2(b)). 이 방법은 영상처리 및 컴퓨터 비젼과 같은 시간이 많이 소요되는 주요 프로세스를 Server에서 수행하는 방법으로, PDA는 Server의 주요 프로세스를 도와 주기 위한 하나의 장치로 입력과 출력만을 담당한다.

본 논문에서는 위의 두 가지 방법을 절충하여 Client (PDA)와 Server(PC)의 자원을 최대한 활용하는, 빠르고 효과적인 시스템을 제안한다(그림 2(c)). 우리가 제안한 시스템에서 Client(PDA)는 입력과 출력만을 담당하는 장치가 아니라, PDA의 자원을 이용하여 대략적인 문자 영역을 검출하며, Server(PC)는 Client(PDA)의 결과를 바탕으로 정밀하게 문자를 추출한다.

2.2 파이프라이닝 처리(pipelining processing)

연속 영상에서 Client(PDA)와 Server(PC)의 자원을 활용하여 효과적으로 문자를 추출하기 위해, Client (PDA)와 Server(PC), 각각의 CPU를 이용하여 파이프라이닝 방식으로 시스템을 구축한다.

그림 3은 제안된 Client/Server구조에서의 파이프라이닝 방식의 시스템을 보여준다. 본 논문에서의 파이프라이닝 처리란, Server(PC)와 Client(PDA) 프로세스를 시간적으로 중첩되게 병렬 처리하여 한 프로세스가 끝나면 바로 다음 프로세스가 수행되는 것을 말하는 것으로, Server(PC)의 CPU가 MLP와 CC 프로세스를 수행하는 동안, Client(PDA)의 CPU는 에지 밀도(ED) 프로세스를 수행하여 중첩되는 시간을 줄인다. 결과적으로, 한 프레임이 수행되는 시간은 Server(PC)[3]의 수행 시간과 Client(PDA)와 Server(PC) 사이의 전송시간을 합친 것과 같으며,[4] Server(PC)는 Client(PDA)의 예비 문자 영역에 대해서만 문자 추출을 수행하므로, PDA를 이용하지 않고 단지 PC만으로 MLP와 CC를 이용하여 문자 추출을 수행하는 시간보다 빠른 결과를 가진다.

그림 4는 제안된 시스템의 흐름도를 나타내며, 그림 4의 (1), (2)는 각각 Client(PDA), Server(PC) 사이의 데이타 전송을 나타낸다. 그림 4의 (1)은 Client(PDA)의 결과를 Server(PC)에 전송하는 부분이며, 그림 4의 (2)는 Server(PC)의 결과를 Client(PDA)의 스크린에 보여주기 위해 전송하는 부분이다. 그림 3에서 보는 바와 같이 Client(PDA)는 에지 밀도(ED)를 수행하고 Server(PC)의 결과가 Client(PDA)에 전송되길 기다린다(그림 4의 (4)). 그림 4의 (3)은 Server(PC)의 결과가 Client(PDA)에 전송되었음을 알려주는 부분이며, Server(PC)의 결과가 Client(PDA)에 전송되면, Client (PDA)는 Server(PC)에 Client(PDA)의 결과를 전송하고, 다음 주기를 위한 Client(PDA)의 프로세스를 수행한다.

3. Client(PDA) Side

Client(PDA)는 에지 밀도를 이용하여 대략적인 문자 영역을 추출하고, 전송 속도를 줄이기 위해 JPEG포맷으로 결과 영상을 압축, 전송한다.

영상에서 문자의 외곽은 에지로 이루어져 있다는 점을 이용하여, 영상의 에지 밀도를 계산하여 대략적인 문자 영역을 검출한다. 에지 밀도는 각각 N×N 윈도우(window)내의 에지 픽셀(pixel)의 개수를 말하며, 에지 밀도가 임계값(threshold) 이상이면 대략적인 문자 영역으로 생각한다. 소벨(sobel operator)을 이용하여 에지 픽셀을 구하였으며, 21×21 윈도우를 사용하고, 임계값으로 50을 사용하였다. 240×320 영상에 대해 PDA에서 에지 밀도를 수행하면 프레임 당 약 150ms의 수행시간이 소요되며, 간단한 계산량에 비해 많은 수행시간을 가지지만, 에지 밀도를 수행하지 않고 입력 영상을 그대로 Server(PC)에 전송하는 시스템에 비해 전체적인 시스템

3) Client의 수행 시간보다 Server가 더 오래 소요되므로 Server의 수행 시간을 합친다.

4) JPEG Encoding 시간은 Edge Density에 포함되어 있으며, JPEG Decoding 시간은 MLP에 포함되어 있다.

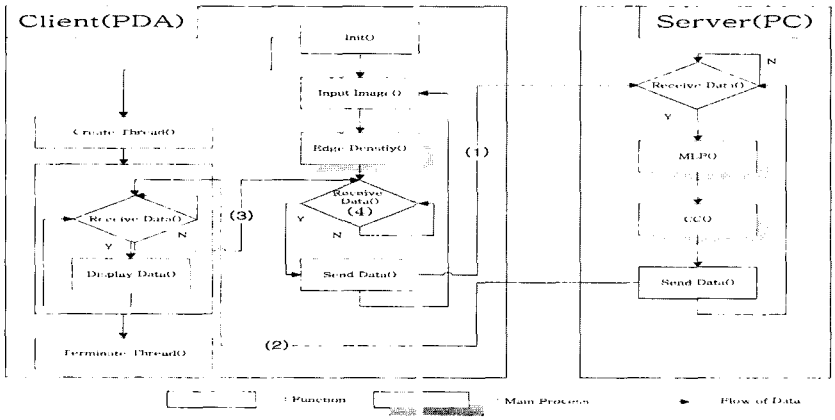
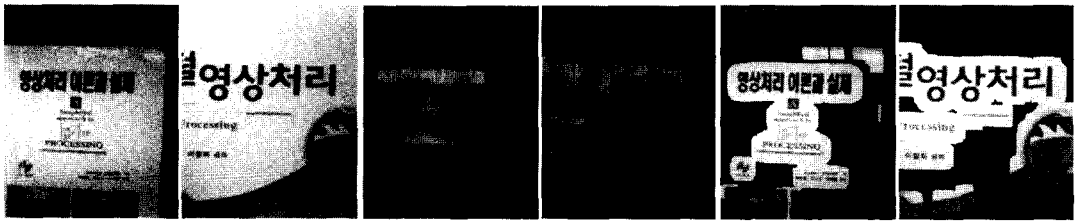
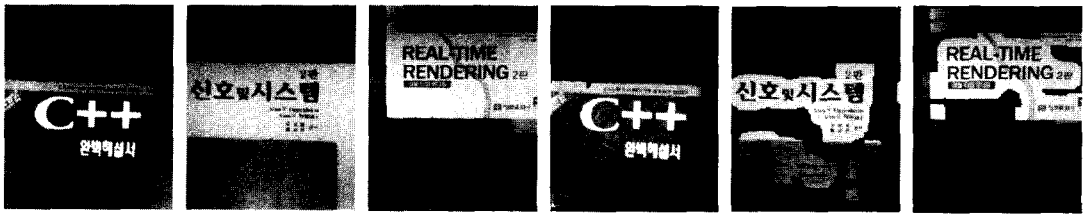


그림 4 제안된 시스템의 흐름도.



(a) 입력 영상 (b) 에지 밀도 영상 (c) 에지 밀도를 이용하여 획득한 결과 영상

그림 5 에지 밀도를 이용한 결과 영상



(a) (b) (c) (d) (e) (f)

그림 6 에지 밀도를 이용한 결과 영상: (a)-(c) 입력 영상, (d)-(f) 결과 영상

의 수행시간을 줄일 수 있다.

그림 5는 에지 밀도를 이용하여 대략적으로 문자 영역을 검출한 결과를 보여준다. 그림 5(a)는 에지 밀도 영상을 나타내며, 하얀색은 임계값 이상일 때 에지 픽셀의 개수를 표시한 것이며, 검은색은 문자 영역이 아님(임계값 이하)을 표시한다. 그림 5(b)는 에지 밀도를 이용하여 획득한 결과 영상을 나타낸다. 그림 5(b)에서 보는 바와 같이 문자 영역에 대해 높은 검출률을 가지지만, 에지가 많이 포함되어 있는 영상에서 많은 오 검출이 생기는 단점을 가지고 있다.

Server(PC)에서는 그레이 영상(gray image)만으로 문자 추출을 수행하므로, 에지 밀도를 이용한 문자 추출 결과 영상을 그레이 영상으로 변환하여 압축한다. 우리는 영상을 압축하는 방법으로 JPEG 알고리즘을 이용하였다. JPEG 알고리즘은 효율적인 압축률을 보이며, 일반적으로 1/20까지 압축률을 높여도 원래의 영상과 비교해서 큰 차이가 없다[8]. 그림 6은 에지 밀도를 이용한 결과 영상을 나타내며, 표 1은 그림 6의 (d)-(f)를 이용하여 JPEG과 Run-Length Encoding(RLE)을 비교한 실험 결과이다. 표 1의 수행 시간은 JPEG과 RLE가 수행되는 시간을 나타내며, 전송 시간은 수행결과를 Server(PC)로 전송하는 시간을 나타낸다. 표 1에서 보는 바와 같이 JPEG은 전체 수행 시간과 압축률에서 RLE에 비해 좋은 성능을 보인다.

표 1 압축률(%)[5]과 영상 압축 알고리즘의 수행 시간(ms) 비교

	압축률				수행 시간	전송 시간	전체 수행 시간
	(d)	(e)	(f)	평균			
JPEG	79.13	79.42	79.77	79.44	30	50	80
Run-Length	62.60	63.60	66.50	64.23	15	80	95

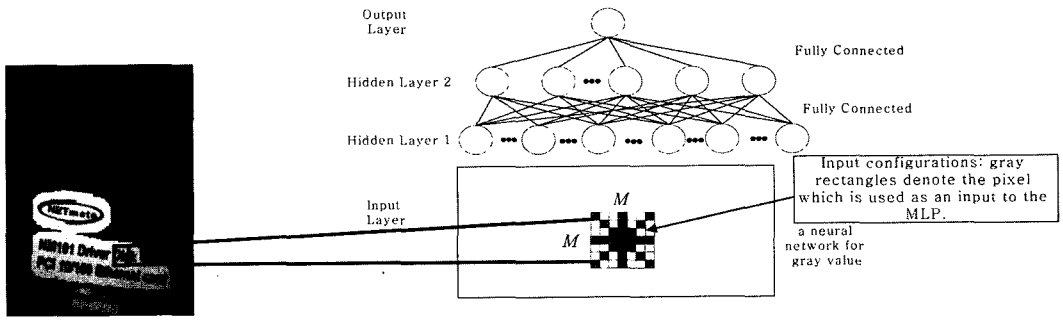


그림 7 신경망의 개략적인 구조

4. Server(PC) Side

Server(PC)는 Client(PDA)의 결과를 바탕으로 정밀하게 문자를 추출하며, MLP 기반의 텍스춰 분류 방법과 연결 성분 기반의 필터링 방법을 이용한다. PDA에 장착된 카메라에 의해 획득된 영상은 저 해상도이며, 다양한 배경을 포함한다. MLP 기반의 텍스춰 분류 방법은 이런 영상에서도 높은 검출률을 보이며, CAMShift 알고리즘을 같이 이용함으로써, MLP 방법의 문제점으로 지적되는 입력 영상 전역 탐색에 관한 문제점을 해결한다. 또한, MLP의 오 검출에 대한 문제, 문자를 배경과 분리하는 문제를 해결하기 위해 연결 성분 기반의 필터링을 이용한다[10].

4.1 MLP 기반의 텍스춰 분류 방법

문자 영역과 비-문자 영역을 구분하는 텍스춰 분석기를 구성하는 어려움을 극복하기 위해, 본 논문에서는 MLP를 사용하여 다양한 크기나 모양의 문자와 배경에 적응할 수 있는 텍스춰 분석기를 구성한다. MLP는 2개의 은닉층, 1개의 출력노드로 구성되며 인접한 노드들은 모두 연결되어 있고, 입력층은 그림 7과 같이, 입력 영상에서 M×M 크기의 입력층 내의 검은색으로 표시된 픽셀들의 그레이 값을 이용한다. MLP를 이용하여 입력 영상을 처리함으로써 생성된 MLP의 출력 영상(TPI: Text Probability Image)의 각 픽셀은 대응하는 입력 영상 내의 해당하는 픽셀의 문자 여부를 나타낸다. 그림 7은 신경망의 입력층이 생략되어 있고, 대신 입력 픽셀의 모양으로 이를 대신한다. 이러한 입력 구조는 텍스춰 분석 분야에서 성능과 속도 향상에 좋은 것으로 알려져 있다[12].

4.2 CAMShift 알고리즘

기존의 텍스춰 분류 방법은 입력 영상의 전체 영역을 탐색하여 많은 수행 시간을 가지는 단점이 있다. 우리는 이런 문제를 해결하기 위해 변형된 CAMShift 알고리즘을 사용하였다[13].

본 논문에서는 TPI 상에서 CAMShift 알고리즘을 반복 수행함으로써 영상 내의 문자를 검출한다. 시작 단계에서 TPI 상의 각 검색창이 문자 영역을 포함하고 있는지를 결정하며, 연속된 일련의 단계에서 2차원 모멘트(moment) 계산을 통해서 문자 영역의 크기와 위치를 구하고 인접한 픽셀을 병합하면서 문자 영역을 찾게 된다. 매 번의 반복 수행과정에서 검색창의 크기를 문자 영역의 크기에 비례해서 수정하고, 겹치는 노드들을 제거하기 위해 노드 병합과정을 수행한다. 수렴된 후 각 문자 영역들은 크기와 가로 세로 비율을 이용하여 필터링을 거친다. 그림 8은 문자 검출을 위한 CAMShift 알고리즘을 나타낸다. 그림 8에서 x, y는 x, y축, i는 i번째 검색창, 0,t,t+1은 실행순서(iteration), ε_x, ε_y는 임계값을 표시한다.

문자 영역의 위치와 크기를 구하기 위해 연산이 간단하고 잡음에 효과적인 모멘트를 사용하였다. 이차원 p+q차 모멘트는 다음과 같이 기술할 수 있다.

$$M_{pq}(i) = \sum_x \sum_y x^p y^q TPI(x,y) \quad (1)$$

문자 영역의 중심 좌표(sample mean location)는

$$mean_X(i)_t = \frac{M_{10}}{M_{00}}, \quad mean_Y(i)_t = \frac{M_{01}}{M_{00}}. \quad (2)$$

5) 압축률(%) = $\frac{O-C}{O} \times 100$ (O는 원본 파일의 비트 수, C는 압축 파일의 비트수)[9].

■ Set up the initial locations ($mean_x(i)_0$, $mean_y(i)_0$) and sizes ($\lambda_x(i)_0$, $\lambda_y(i)_0$) of search windows Ws
■ Do
For each search window $W(i)$
- Generate the TPI within $W(i)$ using MLP.
- Based on the mean shift vector, derive the new position and size of a text region.
- Modify $W(i)$ according to the derived values.
Merge overlapping windows.
Increment the iteration number t.
While($||\ mean_x(i)_t - mean_x(i)_{t+1}\ ||\ > \varepsilon_x$ or $||\ mean_y(i)_t - mean_y(i)_{t+1}\ ||\ > \varepsilon_y$)
■ Filter localized text regions using area and aspect ratio of the region-bounding rectangles

그림 8 문자 검출을 위한 CAMShift 알고리즘

로 설정할 수 있고, 문자 영역의 폭(width)과 높이(height)는 다음과 같이 표현할 수 있다.

$$width(i)_t = \sqrt{2(a+c)+2\sqrt{b^2+(a-c)^2}},$$
$$height(i)_t = \sqrt{2(a+c)-2\sqrt{b^2+(a-c)^2}} \quad (3)$$

where

$$a = M_{20}/M_{00} - (M_{10}/M_{00})^2,$$
$$b = 2(M_{11}/M_{00} - M_{10}M_{01}/M_{00}^2)$$
$$and\ c = M_{02}/M_{00} - (M_{01}/M_{00})^2.$$

문자 영역의 폭과 높이를 계산한 후에, 각 노드를 새로운 위치로 옮기고, 검색창의 크기를 다음의 식처럼 변경한다.

$$mean_X(i)_{t+1} = mean_X(i)_t,\ mean_Y(i)_{t+1} = mean_Y(i)_t, \quad (4)$$

$$\lambda_x(i)_{t+1} = width(i)_t,\ \lambda_y(i)_{t+1} = height(i)_t \quad (5)$$

반복 수행 과정에서 불필요한 중복 수행을 하지 않기 위해서, 노드들의 겹쳐진 비율이 일정 이상이면 두 노드를 병합한다. 이와 같은 병합과정은 더 이상의 병합할 노드가 없을 때까지 수행한다. D_α와 D_β를 각각 두 개의 문자 영역 α와 β에 의해 점유된 영역이라고 할 때, 두 노드의 겹쳐진 정도는 아래와 같다.

$$\Lambda(\alpha,\beta) = \max(\text{size}(D_\alpha \cap D_\beta)/\text{size}(D_\alpha),\ \text{size}(D_\alpha \cap D_\beta)/\text{size}(D_\beta)) \quad (6)$$

이때 size(λ)는 영역 λ내의 픽셀의 개수이다. 경계치 T_O를 0.8로 설정하였을 때, 두 노드 α와 β가 $T_O \le \Lambda(\alpha,\beta)$조건으로 병합을 결정한다.

그림 9는 CAMShift를 이용하여 문자를 검출한 결과를 보여준다. 그림 9(b)에서 검은색 화소들은 MLP에 의해 문자 영역으로 표시된 부분이며, 회색 부분은 MLP가 수행되지 않은 영역들이다. 이러한 CAMShift 알고리즘으로 기존의 텍스춰 기반의 방법들에서 문제점으로 지적되는 입력 영상 전역 탐색에 관한 문제점을 어느 정도 해결할 수 있다. 그림 10은 검색창의 변화에 따른 검출된 문자 영역을 나타낸다.

4.3 연결 성분 기반의 필터링

에지 밀도를 이용하여 대략적으로 문자를 검출하면 높은 검출률을 가지지만, 에지가 많은 영상에서 높은 오검출을 보인다. 이런 오 검출을 줄이기 위해 MLP 기반의 텍스춰 분류 방법을 이용하여 문자 검출을 수행하였으나, 그림 9, 10에서 보는 바와 같이 텍스춰 기반의 문자 검출 방법은 고-대비(high-contrast), 고-주파(high-frequency) 영역, 문자와 비슷한 텍스춰 특성을 가진 영역에서 오 검출을 가지게 된다. 또한, MLP의 결과는 문자의 위치만을 나타내며, 문자를 배경과 분리하는 별

(a) 마킹된 문자 영역 (b) 신경망의 결과 영상.

그림 9 CAMShift를 이용한 문자 검출 예

(a) (b) (c) (d) (e) (f)

그림 10 검색창의 변화에 따라 검출된 문자 영역: (a) 부터 (f) 순서

(a) MLP 결과 영상 (b) 양자화 영상

그림 11 Single-link 클러스터링을 이용한 양자화 영상

도의 작업이 필요하다. 우리는 오 검출에 대한 문제, 문자를 배경과 분리하는 문제를 해결하기 위해 연결 성분 기반의 필터링을 이용하여 문자를 추출한다.

본 논문에서는 연결 성분을 만들기 위해 single-link 클러스터링(clustering) 알고리즘을 이용하여 양자화를 수행한다[13]. Single-link 클러스터링을 수행하기 위해 우리는 256×256 행렬(matrix)[6]을 구성하였으며, 각 클러스터링 단계에서는 가장 비슷한 두 개의 그레이 값을 병합해 가며, 두 개의 병합된 그레이 값은 낮은 값으로 대체한다. 실험을 통한 결과, 4개의 그레이 값으로 클러스터링 한다. 각각의 그레이 값에서 연결 성분을 구하고 특징값(크기, 면적, 가로 세로의 길이 등)을 추출한다. 이러한 전처리 과정을 거친 후 다음의 2단계 필터링을 수행한다.

단계 1: 면적, 크기, 가로 세로의 길이와 같은 연결 성분의 특징값을 이용: MAX(MIN)_AREA, MAX(MIN)_WIDTH, MAX(MIN)_HEIGHT와 같이 미리 정한 연결 성분의 특징값을 이용하여 필터링을 실행한다.

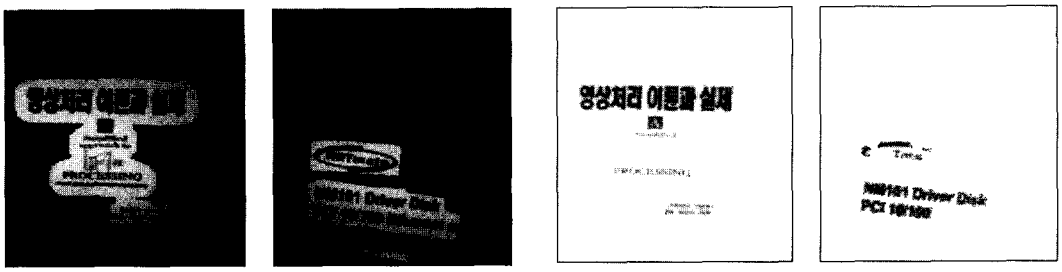
단계 2: 기하학적 배열: 근접한 연결 성분의 수를 확인한다. 문자는 2개 이상의 근접한 연결성분을 가져야 하며, 가로, 세로의 길이가 유사해야 한다.

6) 입력 영상이 그레이 영상이므로 256×256(28) 행렬을 구성한다.

그림 11은 single-link 클러스터링을 이용한 양자화 결과 영상이며, 그림 12는 MLP의 수행 결과에 연결성분의 두 단계 필터링을 적용한 결과를 보여준다. 그림 11, 12에서 보는 바와 같이 고-대비의 문자 영역에 대해서는 높은 정확도를 가지지만, 저-대비의 문자 영역은 양자화 후에 문자 영역이 배경에 포함되는 현상 때문에, 결과 영상에서 저-대비의 문자 영역은 추출되지 않은 단점이 있다. 그림 13은 MLP를 이용할 때와 이용하지 않았을 때의 CC 필터링 결과를 비교한 영상이다. 그림 13(a)는 PDA에서 전송된 대략적인 문자 영역, 그림 13(b)는 MLP를 수행한 결과 영상, 그림 13(c)는 대략적인 문자 영역에 대해 CC 필터링을 적용한 결과, 그림 13(d)는 MLP의 수행 결과에 CC 필터링을 적용한 결과이다. 그림 13(c)와 13(d)에서 보는 바와 같이 CC 필터링 이전에 MLP를 수행함으로써 많은 오 추출을 줄일 수 있다.

5. 실험 결과

본 실험에서 Client(PDA)는 HP iPAQ h5450 모델, everCAM LMC2001B 카메라, 그리고 무선 랜 모듈로 구성되어있다. HP iPAQ h5450 모델은 Pocket PC 2002 기반이며, 64MB SDRAM/48MB Flash ROM을



(a) MLP기반의 수행 결과 (b) 연결 성분 기반의 수행 결과

그림 12 연결 성분 필터링에 대한 결과

(a) 에지 밀도를 이용한 대략적인 문자 영역 (b) MLP를 이용한 문자 검출 결과 영상 (c) (a)영상에 대해 CC 필터링을 수행한 결과 (d) (b)영상에 대해 CC 필터링을 수행한 결과

그림 13 연결 성분 필터링에 대한 결과

사용한다. everCAM LMC2001B 카메라는 32만 화소를 가진다. Server(PC)는 Pentium IV 2.66GHz시스템을 사용한다.

픽셀 단위에서 문자 추출률을 평가하였으며, 아래의 두 식(정확도(precision)와 검출률(recall))을 이용하였다. 평가를 위해 DB는 안내 표지판, 책 표지 등을 주로 사용하였다. 표 2는 50장의 영상에 대해 각 단계에서 검출률과 정확도를 보이며, CC 방법을 사용하면 MLP만을 사용하였을 때 보다 높은 정확도를 보인다. 검출률이 다소 떨어지는 이유는 PDA에서 입력 받은 영상의 낮은 질(quality)로 인하여, 문자의 크기가 작은 영역에서 제대로 추출하지 못하는 경우가 생기기 때문이다. 예를 들어 그림 14에서 보는 바와 같이 문자의 크기가 작은 영역에 대해서는 제대로 문자를 추출하지 못하는 결과를 보인다. 실험 결과를 토대로 실제 처리 가능한 문자의 범주를 테스트해 본 결과 세로 길이가 30픽셀 이상의

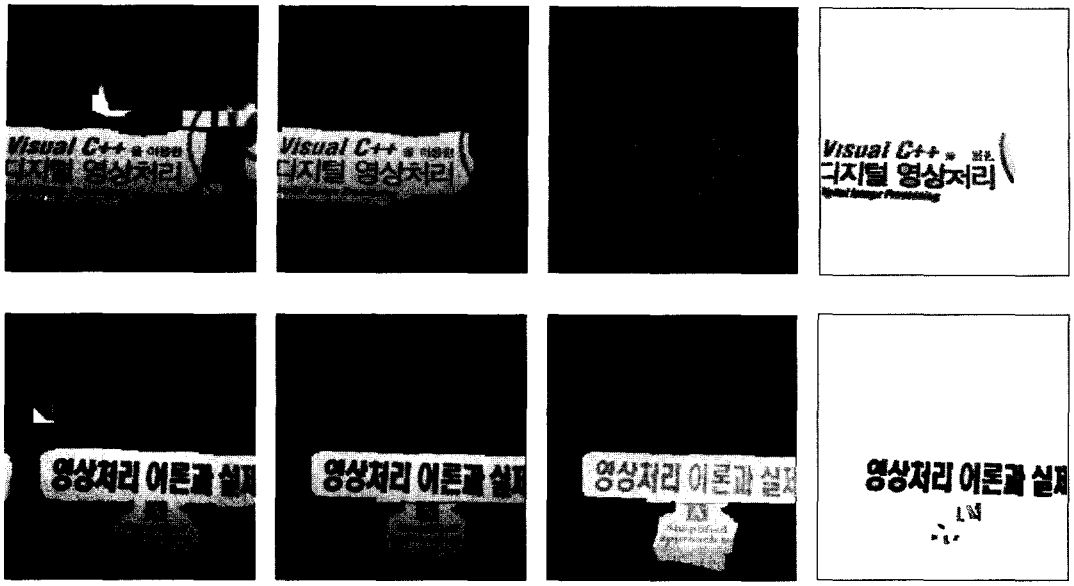
표 2 단계별 정확도와 검출률 비교

	Edge Density	MLP	CC
Precision rate(%)	48.00	73.48	91.12
Recall rate(%)	98.00	92.50	84.34

문자에 대해서는 대부분 추출이 가능함을 확인할 수 있다. 그림 15(d)에서 추출된 최소의 글자(Ladies)는 32픽셀이며, 이 이상의 크기를 가진 문자는 대부분 추출이 가능하다.

$$precision(\%) = \frac{\#of \text{ correctly detected text pixels}}{\#of \text{ detected text pixels}} \times 100 \quad (7)$$

$$recall(\%) = \frac{\#of \text{ correctly detected text pixels}}{\#of \text{ text pixels}} \times 100 \quad (8)$$



(a) PDA에서 전송된 영상 (b) MLP를 이용한 문자 검출 결과 영상 (c) single-link 클러스터링 (d) CC필터링 후의 결과 영상

그림 14 Server(PC)에서의 단계별 결과영상의 예

표 3 각 단계별 수행 시간(ms)

	Edge Density(ED)	JPEG Encoding	PDA->PC 전송시간	MLP with CAMShift[8]	CC	PC->PDA[9] 전송시간	총 수행 시간
Only-PC	0	0	0	475	125	0	600
Only-I/O	0	30	100	475	125	10	740
제안한 방법	150	30	50	300	125	10	665

본 실험에서는 240×320 해상도의 영상을 이용하였다. 그림 14는 Server(PC)에서의 단계별 결과 영상을 보여준다. 그림 14(a)는 PDA로부터 전송 받은 대략적인 문자 영역, 그림 14(b)는 MLP를 이용하여 문자를 검출한 결과, 그림 14(c)는 single-link 클러스터링을 이용하여 양자화를 수행한 결과, 그림 14(d)는 CC를 이용하여 문자를 추출한 결과이다. 그림 14(d)의 배경색은 추출된 문자 영역의 평균 그레이 값을 이용하며 128이상이면 배경을 검은색으로 127이하이면 배경을 하얀색으로 선택한다.

표 3은 각 단계별 수행 시간을 나타낸다. 우리가 제안한 방법은 PDA에서 대략적인 문자를 검출함으로써 PDA의 연산 자원을 사용하지 않는 시스템(Only-I/O)[7]에 비해 Client(PDA)에서 Server(PC)로의 전송 시간이 줄었으며, 대략적인 문자 영역에 대해서만 MLP를 이용하여 문자를 검출함으로써 전체적인 수행 시간을 줄일 수 있다. 그러나, PDA가 없이 PC에서만 MLP와 CC를 수행하는 방법(Only-PC)에 비해, 우리가 제안한 방법은 Client(PDA)의 수행시간, Client(PDA)와 Server(PC)의 전송시간 때문에 더 오랜 수행 시간을 가진다. 우리는 더 빠른 수행 시간을 위해 연속 영상에서 파이프라이닝 형식으로 시스템을 구축한다.

그림 15는 건물 내 안내 표지판에 대해 문자 추출을 수행한 결과를 보여준다. 그림 15(a)는 PDA를 이용하여 입력 받은 영상, 그림 15(b)는 에지 밀도를 이용한 대략적인 문자 영역, 그림 15(c)는 MLP를 이용한 문자 검출 결과, 그림 15(d)는 CC 필터링을 이용한 문자 추출 결과이다. 그림 15(d)처럼 문자와 비슷한 특성을 가진 영역은 오 추출이 발생한다.

그림 16~19는 파이프라이닝 처리를 이용한 연속 영상에서의 문자 추출 결과이다. 그림 16은 Server(PC)에서 수행한 문자 추출 결과이다. 그림 16(a)는 MLP의 결과 영상, 그림 16(b)는 CC 필터링을 수행한 후의 결

7) 이 시스템은 PDA의 연산 자원을 이용하지 않으며, PDA는 단지 JPEG Encoding과 입·출력만을 담당하는 장치이다.

8) MLP with CAMShift 에 JPEG Decoding 시간이 포함되어 있다.

9) PC에서 PDA로 문자 픽셀(pixel)인지 아닌지를 나타내는 이진 영상(binary image)을 전송한다.

(a) PDA에서 입력 받은 영상

(b) 에지 밀도를 이용한 대략적인 문자 영역

(c) MLP를 이용한 문자 검출 결과

(d) CC 필터링을 이용한 문자 추출 결과

그림 15 안내 표지판에 대한 문자 추출 결과

과 영상을 나타낸다. 그림 17은 파이프라이닝 형식의 시스템을 수행했을 때의 프레임 별 결과 영상과 수행 시간이다. 그림 17(a)의 6개 영상에 대해, 그림 17(b)는 MLP와 CC의 수행시간과 한 프레임에 소요되는 전체 수행 시간(TOTAL[10])을 나타낸다. 표 4는 파이프라이닝 방식을 이용한 시스템에서 처음 입력 받은 영상의 수행 시간을 제외한 시스템의 전체 수행 시간이다. 표 4의 PDA가 없는 PC만으로 MLP와 CC를 이용하여 문자를 추출하는 시스템(Only PC)은 CPU가 하나뿐이기 때문에 파이프라이닝 처리가 되지 않으며, 표 3과 같은 수행 시간을 가진다. PDA의 연산 자원을 사용하지 않는 시스템(Only I/O)은, 표 3에 비해 Client(PDA)에서 입력 영상을 압축하는 시간(JPEG Encoding)만큼 수행 시간을 줄인다. 표 5는 Only-PC 시스템에서 에지 밀도를 추가하여 실험한 수행 시간이다. 표 5에서 에지 밀도 수행 시간은 PDA에서 수행하는 시간 보다 빠르며, 전체 수행 시간 역시 에지 밀도를 이용하지 않은 방법에 비해 빨라졌다. 표 4와 표 5에서 파이프라이닝 방식의 시스템(제안한 방법)은 표 5의 Only-PC 시스템에 비해 Client(PDA)의 대략적인 문자 추출, Client(PDA)와 Server(PC)사이의 전송에서 시간을 소모하지만, 비슷한 수행시간을 가진다.

10) TOTAL에는Client(PDA)와 Server (PC)사이의 전송 시간이 포함되어 있다.

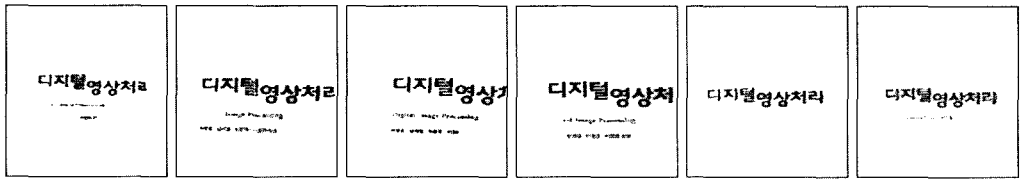
표 4 파이프라이닝 처리를 이용한 수행 시간

	Only PC (Without PDA)	제안한 방법	Only I/O
전체 수행 시간(ms)	600	499	710

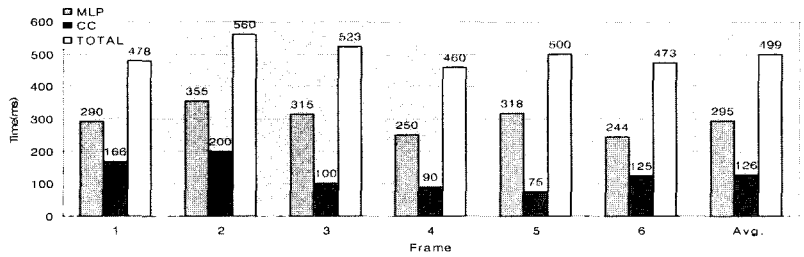
(a) MLP의 결과 영상

(b) CC필터링 결과 영상

그림 16 서버에서 실행한 연속 영상의 문자 추출 결과



(a) 프레임 별 결과 영상



(b) 프레임 별 수행 시간

그림 17 프레임 별 수행 시간

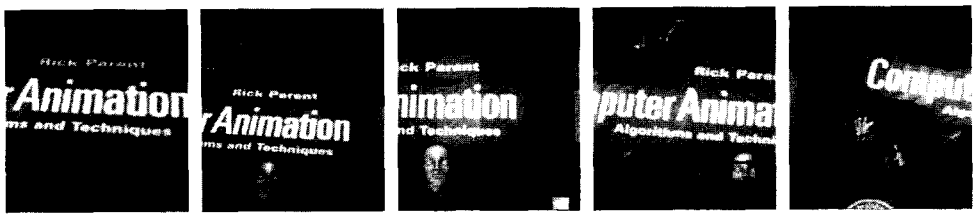
표 5 Server(PC)에서 에지 밀도를 이용한 문자 추출 수행 시간

	ED	MLP with CAMShift	CC	Total
수행 시간(ms)	100	300	100	500

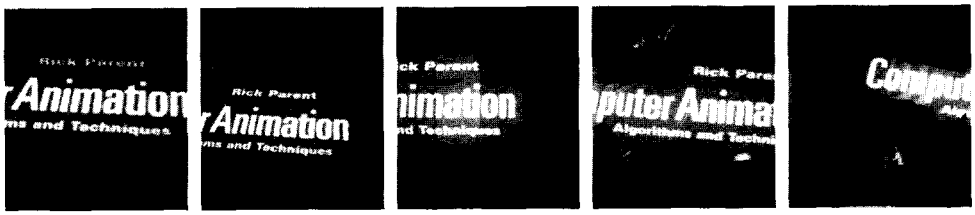
그림 18은 연속영상에서 에지 밀도, MLP, CC의 수행 결과를 보여준다. 그림 18(a)는 에지밀도를 이용한 대략적인 문자 영역, 그림 18(b)는 MLP를 이용한 문자 검출 결과, 그림 18(c)는 CC 필터링을 수행한 후의 문자 추출 결과를 나타낸다. 그림 19는 연속 영상에서 Client와 Server의 결과를 보여준다. 그림 19(a)는 PDA에서 장착된 카메라로부터 입력 받은 입력 영상, 그림 19(b)는 Client(PDA)에서의 결과 영상이며, 그림 19(c)는 Server(PC)에서의 결과 영상을 나타낸다.

6. 결 론

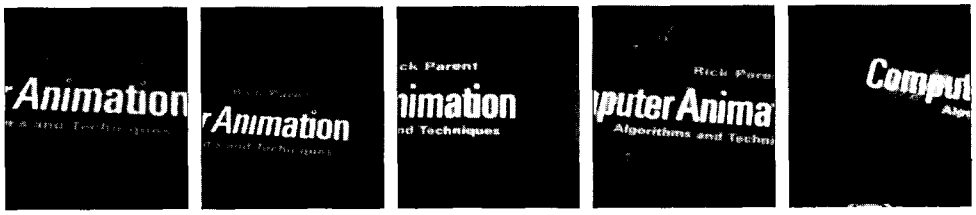
본 논문에서 사용한 MLP와 CC를 동시에 이용한 문자 추출 방법은 PC에서 효과적인 문자 추출 방법이다. 그러나 이 방법을 PDA에서 사용하면 실수 연산에서 많은 수행 시간이 소요된다. 이 문제를 해결하기 위해 우리는Client(PDA)/Server(PC) 구조를 이용하였으며, 연



(a) PDA에서의 에지 밀도를 이용한 대략적인 문자 영역

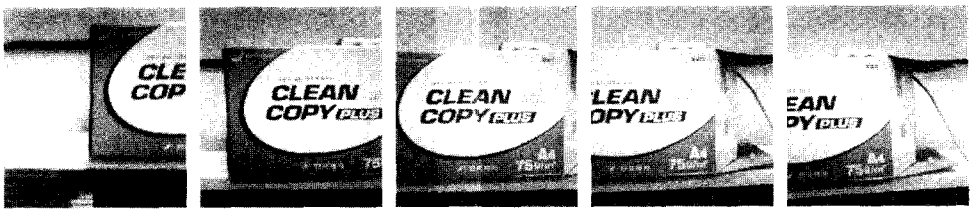


(b) MLP를 이용한 문자 검출 결과



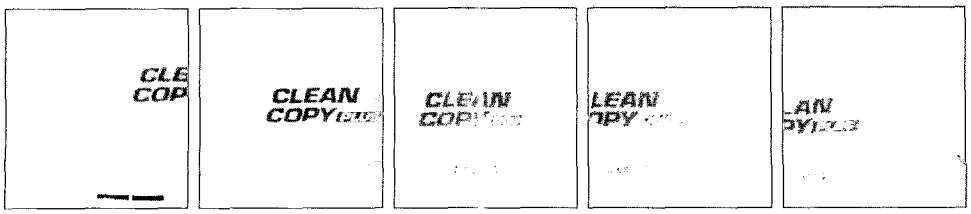
(c) CC를 이용한 문자 추출 결과

그림 18 연속영상에서의 문자 추출 결과



(a) PDA의 입력 영상

(b) PDA 결과 영상



(c) PC 결과 영상

그림 19 연속영상에서의 문자 추출 결과

속 영상에서 수행 시간을 단축하기 위해 파이프라이닝 형식의 시스템을 제안하였다. 결과적으로, 본 실험에서 제안한 방법은 MLP와 CC를 이용하여 효과적인 문자 추출결과를 보였을 뿐만 아니라, Client(PDA)와 Server(PC)의 병렬 처리로 인해 빠른 수행 속도를 보였다.

본 시스템에서는 PDA에서 입력 받은 저해상도의 영상에서 두 가지 문제점이 발생한다. 1) 문자가 배경에 비해 낮은 대비를 가지면 양자화 후에 문자 영역이 배경에 포함되는 현상이 생긴다. 2) 몇몇의 글자가 양자화 후에 하나의 연결 성분을 형성하는 현상이 생긴다. 그러나, 대부분의 응용 분야 또는 시스템에서는 크고 고-대비를 가진 중요한 문자의 인식을 주로 필요로 한다. 그러므로, 본 시스템은 인식이 가능한 중요한 문자만을 추출 대상으로 설정하였다.

향후의 연구과제로써 저해상도 영상에서 더욱 정확하게 문자 영역을 추출하는 방법에 관한 연구가 필요하며, 연속 영상을 고려한 문자 인식 시스템과의 연계에 관한 연구도 필요하다.

참 고 문 헌

[1] Jing Zhang, Xilin Chen, Jie Yang and Alex Waibel, "A PDA-based Sign Translator," Proceedings of the 4th IEEE International Conference on Pattern Recognition, pp. 216-219, Oct. 2002.

[2] Chuang Li, Xiaoqing Ding and Youshou Wu, "Automatic Text Location in Natural Scene Images," Proceedings of the 6th International Conference on Document Analysis and Recognition, pp. 1069-1073, Sep. 2001.

[3] 최영우, 김길천, 송영자, 배경숙, 조연희, 노명철, 이성환, 변혜란, "계층적 특징 결합 및 검증을 이용한 자연 이미지에서의 장면 텍스트 추출", 정보과학회 논문지, 제 31권, 제 5호, pp. 420-438, 2004.

[4] Yasuhiko Watanabe, Kazuya Sono, Kazuya Yokomizo and Yoshihori Okada, "Translation Camera on Mobile Phone," Proceedings of the IEEE International Conference on Multimedia & Expo, Vol. 2, pp. 6-9, July 2003.

[5] Ismail Haritaoglu, "Scene Text Extraction and Translation for Handheld Devices," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 408-413, 2001.

[6] 박현일, 김수형, "휴대폰 카메라로 획득한 저해상도 영상에서의 전화번호 인식", 제 31회 정보과학회 춘계학술대회, 제 B권, pp. 691-693, Apr. 2004.

[7] Xilin Chen, Jie Yang, Jing Zhang and Alex Waibel, "Automatic Detection and Recognition of Signs From Natural Scenes," IEEE Transactions on Image Processing, Vol. 13, No. 1, Jan. 2004.

[8] Randy Crane, "A Simplified Approach to Image Processing," Prentice Hall PTR, 1997.

[9] R. J. Marshall, "The Determination of Peaks in Biological Waveforms," Computers and Biomedical Research, Vol. 19, pp. 319-329, 1986.

[10] 정기철, 김광인, 한정현, "신경망 기반의 텍스춰 분석을 이용한 효율적인 문자 추출", 정보과학회 논문지, 제 29권, 제 3호, pp. 180-191, 2002.

[11] Anil. K Jain and Bin Yu, "Automatic Text Location in Images and Video Frames," Pattern Recognition, Vol. 31, No. 12, pp. 2055 -2076, 1998.

[12] Anil K. Jain and Kalle Karu, "Learning Texture Discrimination Masks," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 2, pp. 195-205, 1996.

[13] Gary R. Bradski and Vadim Pisarevsky, "Intel's Computer Vision Library: Application in Calibration, Stereo, Segmentation, Tracking, Gesture, Face and Object Recognition," Proceedings of IEEE Conference of Computer Vision and Pattern Recognition, Vol. 2, pp. 796-797, 2000.

[14] Huiping Li, David Doerman and Omid Kia, "Automatic Text Detection and Tracking in Digital Video," IEEE Transactions on Image Processing, Vol. 9, No, 1, pp. 147-156, Jan. 2000.

박 안 진

2004년 인제대학교 정보컴퓨터공학과 학사. 2004년~현재 숭실대학교 정보과학대학 미디어학과 석사과정. 관심분야는 패턴 인식, Mobile Vision

정 기 철

1996년 경북대학교 컴퓨터공학과 공학석사. 2000년 경북대학교 컴퓨터공학과 공학박사. 1999년 방문 연구원, Machine Understanding Division, Electro Technical Laboratory, Japan. 1999년 방문 연구원, Intelligent User Interfaces Group, DFKI(The German Research Center for Artificial Intelligence GmbH), Germany. 2001년 PRIP Lab., Michigan State University, U.S. 박사후연구원. 2003년~현재 숭실대학교 정보과학대학 미디어학부 교수 관심분야는 HCI, Interactive Contents, 영상 처리, 패턴 인식, Augmented Reality, Mobile Vision