

## 이미지 코드 인식을 위한 개선된 전처리 알고리즘

임상오<sup>0</sup> 김동철 정철호 한탁돈

연세대학교 컴퓨터과학과 미디어시스템연구실

{solim55, dong}@yonsei.ac.kr, balgeum00@korea.com, hantack@kurene.yonsei.ac.kr

### Enhanced Preprocessing Algorithm for Image Code Recognition

Sang-Oh Lim<sup>0</sup> Dong-Chul Kim Cheolho Chung Tack-Don Han

Media System Lab., Dept. of Computer Science, Yonsei University

#### 요 약

본 논문에서는 코드 영역을 분리하기 위한 전처리 과정 중 코드 추출에 적합한 자동 이진화 알고리즘을 제안하여, 반복과정을 제거하고 정확한 코드영역 추출로 인식률 및 속도를 향상 시켰다. 배경이 복잡한 이미지가 들어 올 경우 기존의 전역 평균 임계값이나 클래스간의 분산을 이용한 방법으로는 이미지 코드 영역을 찾아 낼 수 없었던 문제를 해결하기 위하여 이미지 코드 주변에 배경과 구분을 두기 위한 흰색 영역이 있다는 점을 착안, 상하좌우 방향 바깥쪽에서 안쪽으로 탐색하여 가장 밝은 값을 갖는 값을 찾아내고 찾아낸 그룹 중 가장 낮은 값을 임계값으로 선택하여 최적의 임계값을 찾아 내었고 이를 통해 복잡한 영상 내에서도 이미지 코드 영역을 찾아 낼 수 있다. 제안된 이진화 알고리즘의 성능을 평가하기 위하여 2000장의 테스트 이미지에 적용한 결과, 기존의 이진화 알고리즘들 보다 정확성뿐만 아니라 속도 면에서도 우수한 것을 확인하였다.

#### 1. 서 론

최근 컴퓨팅 환경이 유비쿼터스 환경으로 변화해 감에 따라 물리적인 현실 세계와 컴퓨터상의 가상 세계를 연결시켜주는 태그 인터페이스의 기술이 대두되고 있다. 태그 인터페이스로는 크게 무선 기술을 이용한 RFID와 영상처리 기술을 이용한 이미지 코드로 나눌 수 있다. 이미지 코드는 바코드와 같이 실생활에서 다양한 분야에 사용되고 있으며 모바일 시대에 접어들면서 모바일 환경에 적합한 모바일 코드들이 개발되고 있다. 현재 국내에서 개발된 모바일 코드에는 흑백을 색상을 이용하고 국내 SK 와 KTF 이동통신사에서 네이트코드 및 핫코드로 서비스 되고 있는 아이콘랩에서 개발한 코드가 있고, 연세대학교 미디어시스템 연구실에서 개발하여 활발히 연구되고 있고 많은 논문들이 발표되고 있는 컬러 색상을 사용한 컬러코드(ColorCode)가 대표적인 예이다[1].

모바일 환경에서 시스템을 구축하기 위해서는 일반 컴퓨터 시스템과는 달리 저사양에서 실시간 작동 가능하도록 알고리즘의 최적화가 필요하다. 모바일 환경에서의 이미지 코드 인식 시스템을 구축하는데 필요한 핵심적인 기술로는 전처리 기술(Preprocessing), 이미지 복원(Image Restoration), 색상 분류(Color Classification), 색상 인식(Color Recognition) 등의 기술이 요구된다. 각각의 기술들에는 여러 가지 주요한 연구주제들이 있으며 이번 논문에서는 전처리 과정 중 배경이 복잡한 경우 처리를 하지 못하고 실패가 일어나는 문제점에 초점을 맞춘다.. 모바일 환경에서의 카메라를 이용하여 이미지 코드를 인식하려면 조명, 인쇄매체, 인식 기기 등과 같은

다양한 요소를 고려해야 하고 따라서 조명과 인쇄매체에 강건한 컬러 코드 인식 알고리즘 필요하다. 본 논문에서는 코드 영역을 분리하기 위한 전처리 과정 중 코드 추출에 적합한 자동이진화(Automatic Thresholding) 알고리즘[2]을 개선하고, 반복과정을 제거하고 정확한 코드 영역 추출로 인식률 및 속도를 향상 시켰다.

#### 2. 관련 연구

##### 2.1 이진화 기법

이진화는 입력된 영상을 흰색과 검정색의 두 가지 색상으로 표현하여 물체와 배경을 분리해 내는 기법으로 그림 1 과 같이 분류할 수 있다. 하나의 임계값을 가지는 Single threshold와 2개 이상의 임계값을 가지는 Multi-threshold 방법으로 분류할 수 있으며, Single threshold를 이용하여 전체 이미지를 이진화하는 Global threshold와 전체 영상을 작은 블록으로 나누어서 각 블록마다 하나의 임계값을 사용하는 Local threshold 기법으로 나눌 수 있다[3].

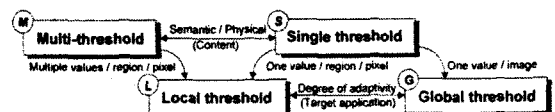


그림 1. 이진화 기법 분류 기준

Global threshold에서는 임계값이 영상 전체에 대하여 적용되는 전역적 값이기 때문에 임계값을 어떻게 결정하

는가에 따라서 이진화 성능이 크게 좌우된다. Global 방법의 경우 알고리즘 자체가 간단하며, 단순한 영상에서는 효율적으로 사용될 수 있으나, 잡영이 심한 영상이나, 전경과 배경간의 명암 변화가 있는 영상에서는 효율이 떨어진다. Global 방법은 임계값을 결정하는 방식에 따라 클래스간 분산을 이용한 방법[4], 엔트로피를 이용한 방법[5,6,8], 히스토그램 변형을 이용한 방법, 모멘트를 유지하는 방법[7], 평균을 이용한 방법, 반복선택법(Iterative threshold selection)[11,12], 최소-최대법(Min-Max Average)[13] 등으로 나누어진다. 클래스간 분산을 이용한 Otsu 방법은 그레이 영상으로부터 히스토그램을 얻어 전경 클래스와 배경 클래스간의 분산이 가장 큰 level을 임계치로 선택하는 방법이다. 히스토그램에서 peak와 valley가 분명한 이미지에서 잘 작동되며, 다시 말해 히스토그램이 bimodal이나 multimodal을 보일 때 좋은 성능을 보인다. 반면에 히스토그램이 unimodal형태이거나 그에 가까운 분포를 보일 경우에는 성능이 떨어진다. 반복선택법의 경우 대략적인 임계 값에서 출발하여 점차 반복적으로 이 추정 값을 향상시키는 방법으로서 임계 값의 처음 추정치 T를 선정한다. 영상의 평균 밝기는 좋은 출발점이 된다. 추정 임계 값 T를 이용하여 영상을 2개의 영역 R1과 R2으로 구분하여, 영역 R1과 R2의 평균 그레이 값  $u_1$ 과  $u_2$ 를 구한뒤, 새로운 임계값을  $T = (u_1 + u_2)/2$  식을 이용하여 결정한다.  $u_1$ 과  $u_2$ 이 더 이상 변하지 않을 때까지 이를 되풀이 하여 최종 T를 구한다. 조명이 균일하지 않은 경우에 반복선택법은 효율이 떨어진다. 최소-최대법은 매우 간단한 자동이진화 방법으로서 명암 값의 최소값과 최대값을 찾아 (최소값+최대값) / 2를 임계 치로 선택하는 방법이다.

Global 방법과는 다르게 이진화하고자 하는 픽셀의 주변 픽셀 값까지 고려하여 그 픽셀의 이진화를 결정하는 방법을 Local 방법[12]이라 한다. 즉, 중심의 픽셀이 주변보다 어두우면 black으로, 주변보다 밝으면 white로 변환시키는 방법으로서 움직이는 윈도우를 이용한 방법[9], Adaptive Thresholding 방법[8,9,10] 등이 있다. 지역적인 특징들에 주목할 수 있는 반면, 이미지의 전체적인 특징을 이용할 수 없는 없으며 실행속도가 전역적 방법에 비해 상당히 느리다는 단점이 있다.

2.2 컬러기반 이미지 코드

일반적인 2D 이미지 코드와 차별성을 갖는 컬러코드는 1999년 연세대학교에서 개발한 2D 이미지 코드로서 범용 장비인 웹 카메라, 모바일 폰 등에서 인식되며 컬러를 이용한 방대한 코드 영역과 향상된 디자인을 가지고 있다. 코드를 인식하는 방법, 매트릭스 형태 그리고 Parity 형태에 따라 다양한 종류의 코드가 있으며 적용되는 분야의 특성에 맞게 코드가 사용된다. 컬러코드는 컬러 셀들로 이루어진 2차원 매트릭스형 코드로 데이터 영역, 오류탐지 영역, 방향 탐지 영역으로 구성되어 있다. 컬러코드 데이터 영역에는 문자, 숫자, 기호 등이 정보가 포함되어 있으며 코드 변환 표에 따라 컬러 값으로 맵핑된다. 오류 검증 영역은 데이터 셀들의 컬러 값에

대한 오류 검증 연산을 수행한다. 코드 방향 탐지 영역은 오류 검증 영역 셀들의 정보를 연산함으로써 코드의 방향성을 탐지한다.[14,15]

컬러코드 인식 과정 중 전처리 과정에서 임계값을 정하는 단계는 전체 프로세스에 영향을 줄 만큼 매우 중요하다. 임계값을 정하는 방법으로는 전체 이미지를 그레이 레벨 이미지로 변환 후 전체 이미지 픽셀의 평균 값을 임계값으로 하는 Global 방법과 두 가지 고정 값이 사용된다. Global 평균 기법만으로는 평균 91.2%의 성공률을 보이고 있고 두 가지 고정 값을 추가적으로 사용할 경우 평균 96.4%의 성공률을 보여준다. 실패율을 보이는 3.6%의 이미지를 분석해본 결과 배경이 복잡하거나 이미지 평균에 영향을 미칠 만큼의 큰 물체가 존재하여 기존의 임계값 결정 방법으로는 찾아내기가 힘든 이미지임을 알 수 있었다.

3. 코드주변의 밝기 정보를 이용한 개선된 전처리 기법

어떠한 종류의 이미지 코드이던지 코드영역과 배경을 분리하기 위해서는 두 영역간의 경계가 명확히 구분되도록 배경색이 사용되어야 한다. 컬러코드의 경우 코드영역에 Red, Green, Blue, Black을 사용하기 때문에 이들 색이 사용되는 코드영역을 배경으로부터 분리해 내기 위해서는 이들 색과 구분이 잘 되는 흰색계통의 밝은 색이 배경전체에 사용되거나 코드주위 잉여여백에 사용될 수 밖에 없다.

따라서 본 논문에서는 이러한 잉여여백의 밝기 값을 구하여 이 값을 임계값으로 하여 이진화를 실시함으로써, 배경이 아무리 복잡하고 다양한 색으로 구성되어 있을지라도 정확하고 빠르게 코드영역을 분리해 낼 수 있는 기법을 제안하였다. 제안한 방법은 이미지 코드 영역 주변에는 흰색의 잉여여백이 반드시 존재하며, 코드 영역은 대부분 이미지의 중앙부분에 위치한다는 가정 하에 다음과 같은 전처리 과정을 수행한다.

Step 1: 이미지의 바깥 4지점(좌,우,상,하)에서 중심으로 탐색해 가면서 명도 값을 비교하여 각 구간에서 흰색 영역의 명도 값을 구한다. 좌, 우, 상, 하 각각의 바깥 방향에서 중앙으로부터 탐색하여 구한 명도 값은 각각  $t_1, t_2, t_3, t_4$  로 매칭 된다.

$$t_1 = \text{Max}_{0 \leq j \leq n-1} \left\{ \frac{1}{n+1} \sum_{i=h}^{i+n} g(i, j) \right\} \quad (1)$$

$$t_2 = \text{Max}_{0 \leq j \leq n-1} \left\{ \frac{1}{n+1} \sum_{i=h}^{i+n} g(i, j) \right\} \quad (2)$$

$$t_3 = \text{Max}_{0 \leq j \leq h/2-1} \left\{ \frac{1}{n+1} \sum_{i=j}^{i+n} g(i, j) \right\} \quad (3)$$

$$t_4 = \text{Max}_{h/2 \leq j \leq n-1} \left\{ \frac{1}{n+1} \sum_{i=j}^{i+n} g(i, j) \right\} \quad (4)$$

여기서  $g(i,j)$ 는  $(i,j)$ 좌표를 갖는 픽셀의 gray영상의 명도 값을 나타내며,  $w$ 는 이미지의 너비,  $h$ 는 이미지의 높이,  $n$ 은 명도값 계산시 고려할 픽셀 개수를 나타낸다.

Step 2: 위에서 구한 4가지 명도 값 중 가장 낮은 명도 값을 이진화를 위한 임계값  $T$ 로 설정하여 Global threshold를 수행한다.

그림 2에서 볼 수 있듯이 불균등한 조명에 의해 발생하는 집중조명(spot)이나 그라데이션 때문에, 코드 주위가 같은 흰색이라 할지라도 조명을 많이 받은 영역과 적게 받은 영역의 명도 값은 다르게 나타난다. 그래서 제안한 방법에서는 이러한 불균등한 조명의 영향을 고려하여 탐색방향을 4방향으로 나누어 각각의 방향에서 흰색 잉여여백의 명도 값을 구하도록 하였다.

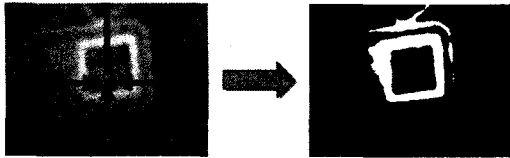


그림 2. 탐색방향 및 이진화 수행결과

4. 실험 및 평가

본 논문은 제안된 알고리즘의 성능을 평가하기 위하여 Intel Pentium-4 2.8GHz CPU와 1GB RAM이 장착된 PC상에서 C++로 구현하였다. 제안된 알고리즘과 최대-최소 평균방법, 전체 평균방법, Ostu 방법들 사이의 이진화 성능을 비교, 평가하기 위하여 5가지 종류의 대체에 인쇄된 코드 이미지를 5가지의 조명 환경 하에서 4가지의 디바이스로 촬영한 총 2000개의 실영상을 사용하였다. 표 1은 실험환경과 대체들을 자세히 나타낸 것이다.

표 1. 실험 환경 및 대체

조명	종류	Kelvin	설명
	D65	6590	흐린날 그늘
TL84	3980	10~15시 사이 햇빛	
CWF	3995	아침저녁 야외	
INC-A	2785	텁스텐	
Horizon	2400	새벽	

코드매체	종류	코드 크기	배경색
	A4	1.1cm*1.1cm	white
잡지	0.8cm*0.8cm	white	
신문	0.65cm*0.65cm	gray	
명함	0.8cm*0.8cm	white	
플라스틱카드	0.75cm*0.75cm	light pink	

디바이스	종류	모델	해상도
	휴대폰	LG Cyon	KP3800
휴대폰	LG Cyon	SV510	1.3 M pixels
PDA	HP iPAQ	RW-6100	1.1 M pixels
휴대폰	Samsung Anycall	SPH-E3700	1 M pixels

표 2. 알고리즘별 이진화 수행결과

원본	Proposed	Min-Max	Global Av	Ostu

표 2는 샘플영상에 대한 이진화 수행결과를 나타낸 것이다. 전처리가 성공하려면 이진화를 수행했을 때 코드 영역이 주위 배경으로부터 분리될 수 있어야 한다. 또한 코드영역이 손상되어서도 안 된다. 실험결과에서 보면, 코드주위의 흰색 잉여여백이 적고 배경이 복잡한 영상에서는 Global Average와 Ostu 알고리즘의 경우, 영상이 전체적으로 어두울 경우에는 Min-Max Average 알고리즘의 성능이 좋지 못했다. 반면에 본 논문에서 제시한 알고리즘을 적용했을 경우에는 대부분의 영상에서 대체적으로 좋은 성능을 보였다.

표 3. 알고리즘 별 전처리 성공률 및 처리속도

illum.	device	Algorithm							
		Proposed		Min-Max Av.		Global Av		Ostu	
		succ(%)	time(ms)	succ(%)	time(ms)	succ(%)	time(ms)	succ(%)	time(ms)
D65	KP3800	94	9	90	10	92	11	93	15
	SV510	98	9	98	10	96	13	100	14
	RW6100	92	9	78	11	59	12	62	16
	SPH-E3700	97	9	97	11	87	13	88	15
TL84	KP3800	98	9	96	10	95	12	98	15
	SV510	95	8	100	10	96	13	99	14
	RW6100	97	9	94	11	90	11	92	15
CWF	SPH-E3700	97	9	98	11	98	12	99	15
	KP3800	94	9	92	10	91	12	100	14
	SV510	95	8	98	10	92	10	98	14
	RW6100	98	9	97	11	98	11	98	15
INC-A	SPH-E3700	100	9	98	11	94	12	91	15
	KP3800	100	9	79	10	80	12	75	16
	SV510	90	10	94	12	85	13	85	16
HORIZON	RW6100	99	9	100	12	99	12	100	14
	SPH-E3700	99	9	97	17	98	12	97	15
	KP3800	94	10	99	16	89	13	97	15
Average		96.1	9.1	95.0	11.5	91.2	12.2	93.1	14.9

표 3은 다섯 가지 표준조명하에서 2000개의 영상에 대해서 수행한 전처리 알고리즘의 성능을 측정된 것이다. 알고리즘들 간의 성능을 비교하기 위하여 각 알고리즘의 전처리 성공률과 처리시간을 측정하였다. 실험결과 제안된 알고리즘이 96.1%의 전처리 성공률을 보여 가장 우수한 성능을 보였다. 제안된 프로세서 성능과 저용량 메모리를 요하는 모바일 컴퓨팅 환경에서 매우 중요한 처리 속도 면에 있어서도 다른 알고리즘에 비해 우수한 성능을 보였다. PDA에서 조명의 종류에 따른 전처리 성공률을 나타낸 그래프를 그림 3에 나타내었다.

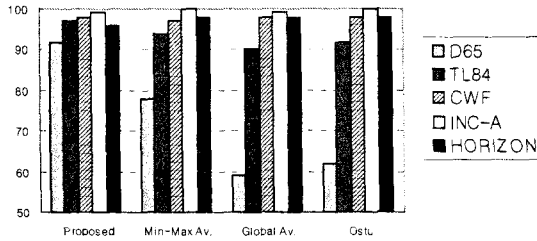


그림 3. 조명의 영향에 따른 성공률

제안한 방법은 모든 조명에 대해 92%이상의 성공률을 보여 조명의 종류에 큰 영향을 받지 않음을 알 수 있으나, 다른 방법들은 조명에 따라 59% ~ 100%의 심한 변이를 보여 조명의 종류에 영향을 많이 받음을 알 수 있다. 그림 4는 D65표준조명하에서 디바이스의 종류에 따른 전처리 성공률을 나타낸 그래프이다.

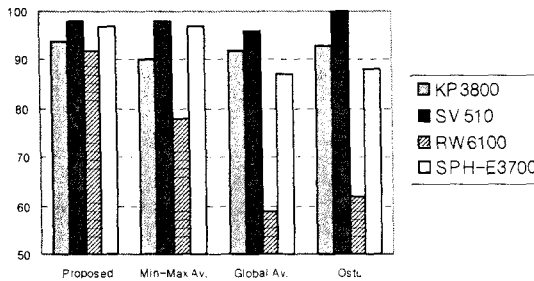


그림 4. 디바이스의 영향에 따른 성공률

동일한 조명하에서, 제안한 방법은 모든 디바이스에 대해서 92%이상의 성공률을 보여 디바이스의 종류에도 큰 영향을 받지 않음을 알 수 있었다. 그러나 다른 방법들, 특히 Global 평균과 Ostu 방법은 장비에 따라 62%이하의 성공률을 보이는 등 디바이스의 종류에 따라서도 영향을 많이 받음을 알 수 있다. 인쇄매체의 종류에 따른 전처리 성공률은 그림 5에 나타내었다.

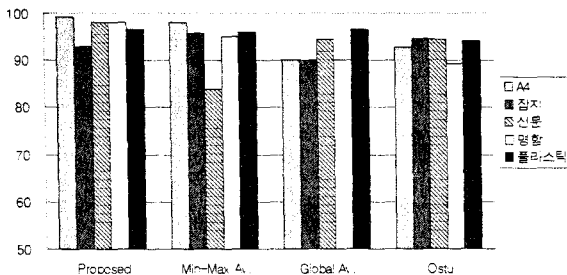


그림 5. 인쇄매체의 영향에 따른 성공률

인쇄매체의 종류에 대해서도 제안된 방법은 모든 매체에 대해 93%이상의 성공률을 보여 가장 우수한 성능을 보였다. 신문과 영합이 일부 알고리즘에서 성공률이 다소 낮게 나타났으나, 전반적으로 볼 때, 인쇄매체의 종류에 대해서는 모든 알고리즘들이 크게 영향을 받지 않는 것으로 보인다.

5. 결 론

본 논문에서는 배경이 복잡한 이미지에서 코드 영역을 분리하기 위한 전처리 과정 중 코드 영역 추출에 적합한 자동 이진화 알고리즘을 제안하였다. 제안된 알고리즘은 임계값을 찾기 위한 반복과정을 제거하고 정확한 코드 영역 추출로 인식률 및 속도를 향상 시켰다. 기존의 전역 평균 임계값이나 클래스간의 분산을 이용한 방법으로는 이미지 코드 영역을 찾아 낼 수 없었던 문제를 해결하기 위하여 이미지 코드 주변에 배경과 구분을 두기 위한 흰색 영역이 있다는 점을 착안, 상하좌우 방향 바깥쪽에서 안쪽으로 탐색하여 가장 밝은 값을 갖는 값을 찾아내고 찾아낸 그룹 중 가장 낮은 값을 임계값으로 선택하여 최적의 임계값을 찾아내었고 이를 통해 복잡한 영상 내에서도 이미지 코드 영역을 찾아낼 수 있다.

제안된 이진화 알고리즘의 성능을 평가하기 위하여 2000장의 테스트 이미지에 적용한 결과, 이미 알려진 기존의 기법인 Ostu, Min-Max Average, Global Average 방법에 비하여 정확성뿐만 아니라 속도 면에서도 우수한 것을 확인하였다. 또한 다양한 조명과 디바이스, 인쇄매체가 존재하는 모바일 컴퓨팅 환경 하에서도 안정감 있는 알고리즘임을 확인할 수 있었다. 끝으로 본 실험에서 얻은 이진화 수준도 이미지 코드 인식을 위한 전처리 단계인 코드 영역분리를 위해선 충분하지만, 전처리 성공률을 더 높이기 위해선 그림 6에서와 같이 불균등한 조명으로 인하여 그라데이션이 심한 이미지를 이진화 하는 전처리 기법에 대해 연구가 필요하다.

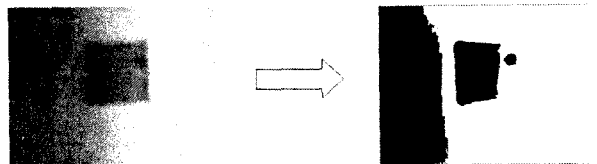


그림 6. 그라데이션 문제

감사의 글

본 연구는 교육인적자원부 BK21 및 한국과학재단 특정기초연구(R01-2005-000-10898-0) 지원으로 수행되었음.

6. 참고문헌

- [1] Tack-Don Han, Cheolho Cheong, "Implementation of new services to support ubiquitous computing for campus life", Software Technologies for Future Embedded and Ubiquitous Systems IEEE, 2004.
- [2] Abutaleb A.S. (1989) "Automatic Thresholding of Gray-Level Pictures Using Two-Dimensional Entropy" Computer Vision Graphics and Image Processing, 47: 22-32
- [3] J.Sauvola, M. Pietikäinen, "Adaptive document image binarization", Pattern Recognition, vol.33, pp.225-236, 2000.
- [4] Otsu N. (1979) "A threshold selection method from gray level histograms" IEEE Trans. on Systems, Man and Cybernetics, 9: 62-66
- [5] Pun T. (1980) "A New Method for Gray-Level Picture Threshold Using the Entropy of the histogram" Signal? Processing,? 2( 3): 223-237
- [6] Pun T. (1981) "Entropic Thresholding: A New Approach" Computer Graphics and Image Processing, 16: 210-239
- [7] Tsai W.H. (1985) "Moment-preserving thresholding: A new approach" Graphical Models and Image Processing, 19: 377-393
- [8] Bernsen J. (1986) "Dynamic Thresholding of Grey level Images" Proc. of the 8th Int. Conf. on Pattern Recognition, pp. 1251-1255
- [9] Duc D.A., Du T.L.H., and Duan T.D. (2004) "Optimizing Speed for Adaptive Local Thresholding Algorithm Using Dynamic Programming " 2004 International Conference on Electronics, Information, and Communications, pp. 438-441
- [10] Sauvola J. and Pietaksinen M. (2000) "Adaptive document image binarization" Pattern Recognition, 33: 225-236
- [11] Ridler T.W. and Calvard S. (1978) "Picture thresholding using an iterative selection method" IEEE Trans. on System, Man and Cybernetics, 8(8): 630-632
- [12] Li C.H. and Tam P.K.S. (1998) "An iterative algorithm for minimum cross-entropy thresholding" Pattern Recognition Letters, 19: 771-776
- [13] Rafael C.Gonzalez and Richard E. Woods, Digital Image Processing, Prentice Hall Inc. 2004
- [14] 정철호, "네트워크 환경에서 정보처리를 위한 입력 인터페이스 연구", 연세대학교 대학원 컴퓨터과학산업시스템 공학과 석사학위 논문, 2001
- [15] 정철호, 이남규, 신은동, 한탁돈, "명함 정보처리를 위한 칼라코드 시스템 설계", 한국정보과학회 HCI2000