코스트 최소화법에 의한 문자영역의 추출

김 석 태†

요 약

범용성을 지닌 문자영역의 추출을 위해서는 대상화상에 의존하지 않는 정보를 활용할 필요가 있다. 본 논문에서는 문자영역의 추출문제를 코스트 최소화 개념으로 접근하여, 문자의 일반적 특징들을 종합적으로 고려하는 결과를 얻을 수 있는, 범용성을 띤 영역추출방법을 제안한다. 구체적으로는, 문자의 형상과 배치에 관한 규칙성을 구하고자 하는 해에 대한 조건으로 설정, 그 조건을 충족시키는 해가 최소값을 갖는 코스트 함수로 도입하고, 이 함수를 Simulated Annealing법에 의해 최소화하여 영역추출을 한다. 본 방법은 코스트 함수를 정의한다는 점에서 다른 방법과 확연한 차별성을 갖는다. 본 코스트 함수를 이용한 영역추출실험 결과, 실험가설에 부합되는 결론을 얻어 제안방법의 유효성이 확증되었다.

On Character Region Extraction by Cost Minimization Method

Seok-tae Kim†

ABSTRACT

If a method of character region extraction will have general purposes, it could not but make use of common features which all target images have. This paper suggests these common features should be considered as the conditions for the region to be extracted within a framework of the cost minimization. The method suggested above could be effective by minimizing a cost function estmating the extent that character regions satify quantitatively the features, through Simulated Annealing Method. This method has an uniqueness in that it defines the cost function.

Experimental results verify the usefulness of this cost minimization approach to characer region extraction.

1. 서 론

최근 문서작성 지원 환경은 크게 정비되고 있다. 따라서 "정보의 생산" 환경도 종전에 비해 크게 개선되고 있는 것이다. 하지만, 정보전달 매체는 종이 인쇄물인 문서가 대종을 이루고 있고, 그 축적된 량 또한 방대한 상태이다. 따라서 "정보 이용", 이를테면 데이터베이스화나 미디어 변환 수준에서 보면, 기존의 방대한 문서정보들을 보다 효율적으로 활용하고 관리할 필요성이 제기된다.

그림, 표, 문자 들을 포함하는 문서화상을 처리하는데 필수적인 요소기술인 문자영역 추출에 관해서는 이제껏 많은 방법들이 제안되고 있다[1-2]. 이들 방법들의 대부분은 주변분포[3-4], 레이아웃 정보[5-6], 배경정보[7]와 같은 대상화상에 대한 경험적 지식을 이용하는 연유로 범용성에 많은 제약이 있었다. 또한 한정된 언어의 문자에만 적용되는 한계성을 가지고

† 종신회원 : 부산수산대학교 정보통신공학과
논문접수 : 1995년 9월 23일, 심사완료 : 1996년 1월 12일

있었다[8-10].

범용성을 가진 문자영역 추출방법을 구안하기 위해서는, 대상화상에 의존하지 않는 정보인 문자영역의 선분 근접성을 이용하는 방법이 제안되고 있다[11]. 그러나 선분 근접성이라는 개념 자체가 애매한가하면, 그 방법이 잡음에 약하며 비정형화된 문서에는 제대로 적응하지 못하는 문제점들이 대두되고 있다.

따라서 문서 내의 선분 근접성 이외에, 문자열 내의 문자 배치에 관한 규칙성을 적극적으로 이용하여 문자영역에 대한 충족도를 검토함으로써 비정형화된 서식 문서에도 효율적으로 적응 대처할 수 있는 대안에 관한 검토가 요구되고 있다.

이에, 본 논문에서는 문자영역 추출을 코스트 최소화 관점에서 접근하여, 비정형화된 문서에서도 문자와 문자열을 동시에 추출할 수 있는 방법을 제안한다. 문자의 정방성, 문자의 근접성, 크기의 일치, 등간격 배치, 직선적 배치 등을 일반적인 문자 특징의 준거로 삼으려 한다. 이것은 상기 문자 특징들을 문자영역이 충족시켜야 할 조건으로 설정하여, 이 조건을 종합적으로 충족시키는 영역을 문자영역으로 추출해내는 방법이다. 구체적으로 말해서, 이들 조건들이 종합적으로 충족되는 해를 최소값을 갖는 코스트 함수로 도입하고, 이 함수를 Simulated Annealing법[12-15]을 이용해 최소화함으로써 영역추출을 완성한다. 모든 영역추출의 문제가 코스트 최소화의 일종이라고 생각할 수 있으나, 본 방법은 코스트 함수를 정의한다는 점에서 다른 방법들과 완전히 구별된다.

2장에서는 한글문서의 문자영역 특징들에 대해서 기술하고, 3장에서는 코스트 최소화에 의한 문자영역의 추출방법을 상세히 설명한다. 4장에서는 영역 추출실험을 통하여 상기 방법의 유효성을 검증한다.

2. 한글문서의 문자영역 특징

2.1 문자의 일반적 특징

한글문서내에 존재하는 문자(한글, 한자, 숫자, 영문자 등)의 형상은 다종다양하지만, 모든 문자는 정사각형 내의 짧은 선분이 집적되어 만들어진다는 점에서는 동일한 특징을 갖는다. 이들 개개의 문자들이 여러 개의 문자집합을 형성하고, 또 그것들이 문자열을 구성한 결과, 의미와 내용을 갖는 최소단위를 이루게 된다.

또한 문자열은 임의수의 문자에 의해 만들어지지만 문자열 중의 한 문자에만 촛점을 맞출 경우 그 문자의 영향 범위는 그 문자의 양옆 두 문자에 국한 될 뿐이다. 따라서 문자의 배치는 인접하는 2~3 문자의 문자 배치에 관한 규칙성에 의해 결정된다.

본 논문에서는 비정형화된 한글문서에 있어서의 문자가 갖는 특징을 각 문자의 형상에 관한 특징과 문자열내의 문자배치에 관한 특징으로 나누어 설명한다. 그 구체적 예는 표 1이다.

이들 특징들은 문서내에 존재하는 모든 문자열에도 해당한다. 그러나 역으로 이들 특징을 충족시키고 있는 문서내의 연결성분이 반드시 문자열을 나타낸다고는 할 수 없다. 즉, 상기 6 가지의 특징들은 문자열 특징을 충족시키는 문자영역 결정에 있어서 필요조건은 되지만 충분조건은 되지 않는다.

〈표 1〉 문자의 일반적 특징
〈Table 1〉 Common Features of Characters

문자의 형상 특징	
정방성	문자의 외접 구형의 형상은 정방형에 가깝다
밀집성	문자의 선분들은 집적되어 있다.
문자의 배치 특징	
근접성	문자열 내에 존재하는 문자들은 화상내의 다른 문자들의 간격에 비해 보다 근접되어 있다.
크기 일치성	문자열 내에 존재하는 문자는 쉼표 등을 제외하면 거의 동일한 크기를 갖는다.
직선적 배치	문자열 내에 연속해 존재하는 3문자들은 거의 동일한 선상에 놓인다.
등간격배치	동일한 문자열 내에 연속해 존재하는 3문자는 거의 동일한 간격으로 떨어져 있다.

2.2 문자의 특징량의 추출

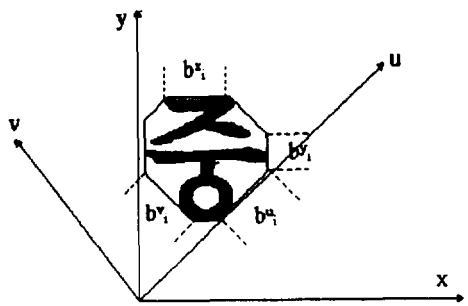
문자 특징들을 코스트 함수에 반영시키기 위해서는 문서 내의 이들 특징들을 수량화할 필요가 있다. 2.1에서 언급한 특징량의 정의 및 수량화는 다음과 같다.

2.2.1 1문자에 관련된 특징량

정방성율 [q]: 문자영역(연결성분들이 결합된 영역)의 외접 구형(外接矩形)이 정방형에 근접하는 정도에

대한 검증은 외접 구형의 가로폭과 세로폭의 비를 조사하므로써 가능하다. 이 점을 고려해 i번째 문자영역의 정방성율을 다음식으로 정의한다.

$$q_i = \min\{q_i^{xy}, q_i^{uv}\} \tag{1}$$



(그림 1) 문자영역의 폭
(Fig. 1) Width of a character region

단

$$q_i^{xy} = \begin{cases} \dfrac{b_i^x}{b_i^y} & b_i^x < b_i^y \\ 1 & b_i^x = b_i^y \\ \dfrac{b_i^y}{b_i^x} & b_i^x > b_i^y \end{cases}$$

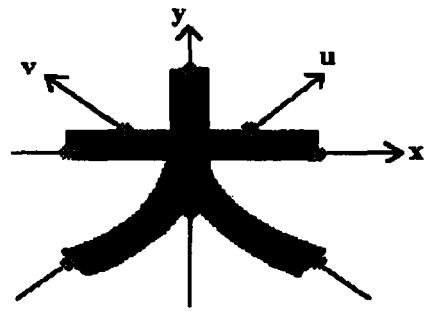
$$q_i^{uv} = \begin{cases} \dfrac{b_i^u}{b_i^v} & b_i^u < b_i^v \\ 1 & b_i^u = b_i^v \\ \dfrac{b_i^v}{b_i^u} & b_i^u > b_i^v \end{cases}$$

여기에서 b_i^x, b_i^y, b_i^u, b_i^v는 그림 1과 같이 각 좌표 방향에서 외접 구형의 폭(width)이다.

선분 밀집율 [w]: 문자 영역의 선분 밀집율을 각 연결 영역의 밀집율의 선형합으로 표시하여 다음의 식으로 정의한다.

$$w_{Ci} = \frac{w_{Ci}^x + w_{Ci}^y + w_{Ci}^u + w_{Ci}^v}{4} \tag{2}$$

여기에서 w_{ci}^x, w_{ci}^y, w_{ci}^u, w_{ci}^v는 그림 2와 같이 영역 c_i의 중심을 원점으로 하는 x축 y축 u축 v축을 가로 지르는 경계의 갯수이다. 그림 2에서는 $w_{ci}^x = w_{ci}^y = w_{ci}^u = w_{ci}^v = 2$ 이다(◎의 갯수).



(그림 2) 선분의 밀집율
(Fig. 2) Line segment concentration

2.2.2 2문자에 관련된 특징량

근접율 [c_{ij}]: 2개의 영역의 근접율은 영역간의 거리와 영역의 크기에 의존한다. 즉, 2개의 문자 영역간의 거리가 같아도 이들의 크기에 따라 그 거리감은 다르다. 이 점을 고려해 문자 영역 c_i와 c_j의 근접율을 다음식으로 정의한다.

$$c_{ij} = \frac{s_i + s_j}{d_{ij}} \qquad (0 < c_{ij} < \infty) \tag{3}$$

단, s_i는 $s_i = \min\{s_i^{xy}, s_i^{uv}\}$이고 s_i^{xy}, s_i^{uv}는 각각 문자영역 c_i의 xy축 및 uv축에 대한 구형의 면적을 나타낸다. 문자영역의 중심을 외접구형의 중심으로 하면 d_{ij}는 문자영역 c_i와 c_j의 중심간의 뉴클릿거리를 나타내고, $d_{ij} = d_{ji}$가 성립한다(그림 3 참조). 또 c_{ij} 값의 크기와 그 근접성 정도는 비례하며 $c_{ij} = c_{ji}$가 성립한다.

크기 일치율 [m]: 2개의 문자영역의 크기 일치율은 영역 크기의 比를 이용한다. 문자영역 c_i와 c_j의 크기 일치율은 다음 식으로 정의한다.

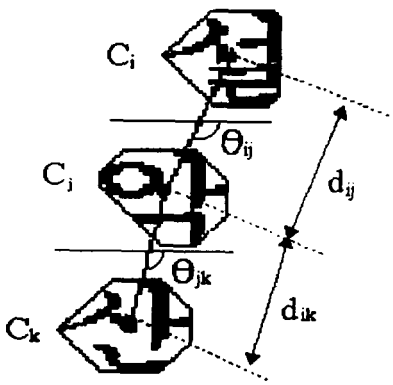
$$m_{ij} = \begin{cases} \dfrac{s_i}{s_j} & s_i < s_j \\ 1 & s_i = s_j \\ \dfrac{s_j}{s_i} & s_i > s_j \end{cases} \qquad (0 < m_{ij} < 1) \tag{4}$$

m_{ij}는 값이 1에 가까울수록 크기가 일치함을 나타내고, $m_{ij}=m_{ji}$가 성립한다.

2.2.3 3문자에 관한 특징량

직선율 [*l*]: 문자영역이 동일선 상에 직선적으로 배열되어 있는가에 대한 검증은 문자영역의 중심선을 연결하는 벡터로 조사한다. 문자영역 c_i, c_j, c_k에서의 이들 직선율은 다음 식으로 나타낸다.

$$l_{ijk}=\frac{1}{\pi}\ |A(\theta_{ij}-\theta_{jk}+\pi)| \qquad (0\le l_{ijk}\le 1) \qquad (5)$$



(그림 3) 문자영역의 배치특징
(Fig. 3) Arrangment regularity of the characters

여기서 함수 A는 주어진 각도를 $-\pi$에서 π까지를 범위로 하고 $A(\theta)=\theta_0$(단, $-\pi\le\theta_0\le\pi$이고 $\theta=\theta_0+2\pi n$ $(n=0, \pm1, \pm2, ...)$)을 충족시키는 함수이다. θ_{ij}는 c_i의 중심과 c_j의 중심을 연결하는 벡터와 x축이 이루는 각도이고 $-\pi\le\theta_0\le\pi$의 크기를 갖는다. 또 $\theta_{ij}=A(\theta_{ji}+\pi)$가 성립한다(그림 3참조). l_{ijk}가 나타내는 직선율은 문자영역이 c_i, c_j, c_k의 순으로 배열되어 있다고 가정한 경우이고, 그 값이 1에 가까울수록 동일 직선상에 존재하고 있음을 나타낸다. 또 $l_{ijk}=l_{kji}$이 성립한다.

등간격율 [*e*]: 동일한 문자열 내에 연속해 존재하는 문자들이 거의 동일한 간격으로 떨어져 있는가에 대한 검증은 문자영역간의 간격의 비를 이용한다. 문자영역 c_i, c_j, c_k에서의 문자배치의 등간격율을 다음 식으로 나타낸다.

$$e_{ijk}=\begin{cases}\dfrac{d_{ij}}{d_{jk}} & d_{ij}<d_{jk}\\ 1 & d_{ij}=d_{jk}\\ \dfrac{d_{jk}}{d_{ij}} & d_{ij}>d_{jk}\end{cases} \qquad (0<e_{ijk}<1) \qquad (6)$$

e_{ijk}의 값이 1에 가까울수록 문자영역 c_i, c_j, c_k가 순차적으로 등간격 배치되어 있다. 또 $e_{ijk}=e_{kji}$가 성립한다.

3. 코스트 최소화에 의한 문자영역의 추출

3.1 해(解)공간의 정의

코스트 함수가 정의된 공간을 '해 공간'이라고 부르기로 한다. 함수의 최소값을 나타내는 해 공간 중의 어느 한 점이 본 방법에서 구하고자 하는 해이다. 즉, 해 공간 상에서 구해진 해는 추출하려는 영역이 문자의 형상 특징과 문자의 배치 특징을 최대로 충족시키는 상태를 나타낸다. 해 공간 P, R를 다음과 같이 정의한다.

$$P=\{p_i\ ;\ i=0, 1, ..., N-1\}$$
$$R=\{r_{ij}\ ;\ i, j=0, 1, ..., N-1\} \qquad (i\ne j)$$

N은 화상 중에 존재하는 문자영역의 총 개수이다. p_i는 제 i번째의 문자영역이 실제로 문자일 확률을 나타내는 변수이고 $0\le p_i\le 1$의 범위를 갖는다. 1에 가까울수록 문자일 가능성이 크고 0에 가까울수록 그 반대이다. r_{ij}는 문자영역 c_i와 c_j의 동일한 문자열내의 인접성에 관한 변수이고, $r_{ij}=r_{ji}$이 성립하고, $0\le r_{ij}\le 1$의 범위를 갖는다. 또 그 값이 1에 가까울수록 동일한 문자열에 속하고 또 문자가 인접하고 있을 가능성이 높다. 그리고 r_{ij}의 값이 0에 가까울 경우는 그 반대이다.

3.2 코스트 함수의 정의

정의된 해 공간에 문자 특징을 반영시킨 코스트 함수를 정의한다. 코스트함수는 목적함수와 제약함수의 선형합으로 나타낸다. 목적함수는 일반적 문자특징을 총괄적으로 만족할 때, 즉 문자영역의 각 변수가 특징량과의 관계가 적절할 때 최소값을 가지도록 정의한다.

한편, 제약함수는 각 변수의 값에 의해 나타내는 영역이 문자영역으로서 적절할 때만 최소값을 취하도록 정의되어 있다.

3.2.1 목적함수

목적함수는 문자의 일반적 특징을 반영시키기 위해 문자의 형상에 관한 특징량 및 배치에 관한 특징량과 상태공간의 각 변수의 관계에 주목해 정의한다. 우선, 문자의 형상을 나타내는 정방성율과 선분 밀집율을 충족시키면 시킬수록 그 문자영역이 실제적으로 문자영역일 가능성이 높아지도록 한다. 그리고 문자배치에 관한 특징을 총괄적으로 만족하면 할수록 그 문자영역이 실제적으로 문자열의 일부분일 확율이 높아지도록 한다. 이러한 조건들을 충족하는 경우에 최소값을 갖도록 목적함수를 정의한다. 본 방법에서는 1개의 문자영역이 존재하는 경우, 2문자가 인접하는 경우, 3문자가 서로 인접하는 경우로 각각 나누고 그것들에 대해 목적함수를 정의한다.

a.1 문자의 목적함수

본 목적함수는 문자영역 c_i의 정방성율 q_i, 선분 밀집율 w_i가 아주 클때 $p_i=1$이 되는 경우와 q_i, w_i가 아주 작은 값을 가질때 $p_i=0$이 되는 경우에 최소값을 갖도록 다음의 식으로 정의한다.

$$F_a(P)=\frac{1}{N}\sum_{i}^{N}[p_i \oplus \{T(q_i, t^q_{\min}, t^q_{\max})T(w_i, t^w_{\min}, t^w_{\max})\}] \quad (7)$$

식(7)에서 역치(域値/threshold value)함수 T는 다음식으로 정의되고 그림 4와 같다.

$$T(x, x_{\min}, x_{\max})=\begin{cases}0 & x < x_{\min}\\ \dfrac{x-x_{\min}}{x_{\max}-x_{\min}} & x_{\min}\le x\le x_{\max}\\ 1 & x > x_{\max}\end{cases}$$

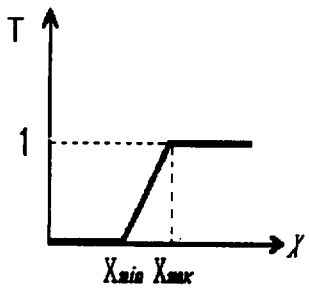
여기서 변수 $t^q_{\min}$, $t^q_{\max}$, $t^w_{\min}$, $t^w_{\max}$는 각각 특징량 q_i, w_i의 역치를 결정하는 파라미터이다. 그리고 연산자 $\oplus$는 다음식에서 정의되는 연산을 한다.

$$x \oplus y=\begin{cases}x+y-2xy & 0\le x, y\le 1\\ M & \text{otherwise}\end{cases}$$

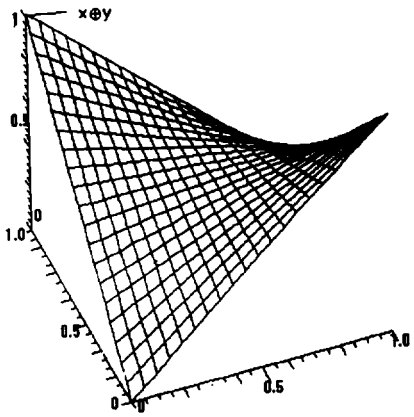
위의 식에서 M은 충분히 큰 값을 갖는다. 연산자 $\oplus$는 배타적 논리합과 같은 $x=0$, $y=0$일 때, 혹은 $x=1$, $y=1$일 때 0의 값을 가진다. 또 $x=1$, $y=0$일 때, 혹은 $x=0$, $y=1$일 때 1의 값을 가진다. x, y가 그 이외의 중간값을 가질 때는 그림 5와 같은 값을 갖는다.

식(7)이 최소일 때는, 모든 i에 대해 $T(q_i, t^q_{\min}, t^q_{\max})$ $T(w_i, t^w_{\min}, t^w_{\max})$가 0.5 이상이며 $p_i=1$이 되는 경우, 그리고 $T(q_i, t^q_{\min}, t^q_{\max})$ $T(w_i, t^w_{\min}, t^w_{\max})$가 0.5 이하이고 $p_i=0$이 되는 경우 들이다. 즉, 이것은 문자영역 c_i는 현재 고려하고 있는 2 개의 특징(q_i, w_i)을 어느 정도 충족시키고 $p_i=0$이 되면 F_a의 최소값이 얻어짐을 의미한다.

또 $T(q_i, t^q_{\min}, t^q_{\max})$ $T(w_i, t^w_{\min}, t^w_{\max})$가 0이나 1에 가까운 값을 가지면 p_i값의 변화가 함수 F_a값에 커다란 영향을 미치지만, $T(q_i, t^q_{\min}, t^q_{\max})$ $T(w_i, t^w_{\min}, t^w_{\max})$가 중간값을 가지는 경우는 p_i값의 변화가 함수 F_a의 값



(그림 4) 역치함수 T
(Fig. 4) Threshold Function T



(그림 5) 연산자 ⊕
(Fig. 5) Operator ⊕

에 그다지 영향을 미치지 않는다. 이것은 특징량 q_i, w_i의 값으로써 형상특징의 충족 여부를 판단할 수 있을 때는 p_i가 함수에 많은 영향을 미치고, q_i, w_i의 값이 애매한 값을 취할 때는 함수에 큰 영향을 미치지 않는 것을 의미한다. 이와 같이 함수 F_a는 문자의 형상특징을 충족시키는 정도에 따라 효율적인 적응대처가 가능하다.

b.2 문자의 목적함수

2개의 문자영역에 촛점을 맞추는 경우, 문자의 배치 특징으로는 문자의 근접성(c_{ij})과 문자의 크기 일치율(m_{ij})을 준거로 삼는다. 이들 c_{ij}, m_{ij}가 큰 값을 가지고 $r_{ij}=1$ 일때, 혹은 c_{ij}, m_{ij}가 작은 값을 가지고 $r_{ij}=0$ 일때 구하는 목적함수가 최소치를 갖도록 정의한다. 2 문자에 대한 목적함수 F_b를 다음식으로 나타낸다.

$$F_b(R)=\frac{1}{N(N-1)}\sum_{i}^{N}\sum_{j,j\neq i}^{N}$$

$$[r_{ij}\oplus\{T(c_{ij},\,t^{c}_{\min},\,t^{c}_{\max})T(m_{ij},\,t^{m}_{\min},\,t^{m}_{\max})\}] \quad (8)$$

위의 식에서 변수 $t^{c}_{\min}$, $t^{c}_{\max}$, $t^{m}_{\min}$, $t^{m}_{\max}$는 각각 특징량 c_{ij}, m_{ij}의 역치를 결정하는 파라미터이다. 그리고 역치함수 T는 식(7)에서 정의한 것과 같은 성질을 갖는다.

c.3 문자의 목적함수

주목하는 문자 영역이 3개인 경우 준거로 삼는 문자의 배치 특징들은 문자의 직선성(l_{ijk})과 등간격성(e_{ijk})이다. 이들 l_{ijk}, e_{ijk}가 큰 값을 가지는 경우 문자영역 c_i, c_j, c_k가 동일한 문자열 내에 연속적으로 존재하며 동일 직선상에 놓인 문자일 가능성이 높다. 역으로, 그렇지않을 경우는 이들 3개의 문자영역중 적어도 하나는 다른 문자들과 무관할 가능성이 높다. 이와 같이 3 문자의 특징에 해당하는 특징량 l_{ijk}, e_{ijk}가 큰 값을 가지고, $r_{ij}\,r_{jk}=0$ 일 때에, 구하는 목적함수가 최소값을 가지면 된다. 이상의 점들을 고려한 3문자 영역에 대한 목적함수 F_c를 다음과 같이 정의한다.

$$F_c(R)=\frac{1}{N(N-1)(N-2)}\sum_{i}^{N}\sum_{j,j\neq i}^{N}\sum_{k,k\neq i,k\neq j}^{N} \quad (9)$$

$$[(r_{ij}\,r_{jk})\oplus\{T(l_{ijk},\,t^{l}_{\min},\,t^{l}_{\max})T(e_{ijk},\,t^{e}_{\min},\,t^{e}_{\max})\}]$$

위의 식에서 변수 $t^{l}_{\min}$, $t^{l}_{\max}$, $t^{e}_{\min}$, $t^{e}_{\max}$는 각각 특징량 l_{ijk}, e_{ijk}의 역치를 결정하는 파라미터이다. 그리고 역치함수 T의 기본적인 개념은 식(7)에서 정의한 F_a와 같다.

단, 3개의 문자영역을 대상으로 하고 있기 때문에 문자영역 c_i, c_j, c_k에서 배치특징이 만족되면 $r_{ij}=1$이고 $r_{jk}=1$이어야 한다. 역으로, 만족하지 않는 경우는 r_{ij}와 r_{jk}중 하나는 적어도 0이어야 한다. 목적함수 F_c는 그 점에서 식(7)와는 차별성을 갖는다.

3.2.2 제약함수

제약함수는 문자가 최소한 만족시켜야 할 조건을 나타낸다. 즉 상태공간 (P, R) 중의 각 변수가 적절한 문자영역을 나타내고 있을때 제약함수는 0에 가까운 값을 갖는다. 본 방법에서는 최소한 만족시켜야 할 조건으로 다음과 같은 것을 도입한다.

a. 각 변수의 범위에 대한 제약함수 :

앞에서 정의한 것과 같이 각 변수 p_i, r_{ij}의 범위는 0에서 1까지의 값을 갖는다. 따라서 모든 변수는 0에서 1의 범위의 값을 갖고, 그 이외의 값은 갖지않는 조건을 나타내는 제약함수를 다음과 같이 정의한다.

$$F_d(P,\,R)=\frac{1}{N}\sum_{i}^{N}H_1(p_i)+\frac{1}{N(N-1)}\sum_{i}^{N}\sum_{j,j\neq i}^{N}H_1(r_{ij}) \quad (10)$$

위의 식에서 함수 H_n은 다음식과 같이 정의하고 그림 6에 나타낸다.

$$H_n(x)=\begin{cases}0 & 0\le x\le 1\\ M & \text{otherwise}\end{cases}$$

b. r_{ij}의 제약함수 :

이 제약함수는 문자열중의 문자의 배치특징을 고려해 변수 r_{ij}를 도입한다. 주어진 한 문자에 촛점을 맞출 경우 그 문자와 인접한 문자의 수는 1~2 개이다. 또 문자가 문자열중에 존재하지 않고 단독으로 있을 때는 인접하고 있는 문자가 존재하지 않는다. 이 조건을 고려한 제약함수를 다음과 같이 정의한다.

$$F_r(R)=\frac{1}{N(N-1)}\sum_{i}^{N} H_2\Big(\sum_{j,j\neq i}^{N} r_{ij}\Big) \qquad (11)$$

여기에서 함수 H_2은 식(10)의 H_n과 동일하다. 함수 F_r은 모든 문자영역이 오직 하나의 문자영역에 속하고, 2개 이상의 문자열내에서 인접하는 것을 허용하지 않는 함수이다.

(그림 6) 함수 H(x)
(Fig. 6) Function H(x)

c. p_i에 관한 제약함수 :

문자열을 구성하는 각 대상영역은 실제로 문자영역이어야 한다. 2개의 문자 영역 c_i와 c_j가 문자열에 인접해 있는 경우, 이들이 문자일 정도를 나타내는 변수 p_i와 p_j가 동시에 1이 되도록 다음과 같은 제약함수를 정의한다.

$$F_p(P,\ R)=\frac{1}{N(N-1)}\sum_{i}^{N}\sum_{j,j\neq i}^{N}\{r_{ij}(1-p_i\,p_j)\} \qquad (12)$$

3.2.3 코스트함수

3.2.1과 3.2.2에서 정의한 목적함수와 제약함수의 선형합에 의해 코스트함수 F_{cost}를 정의한다.

$$F_{cost}(P,\ R)=S_{obj}(S_a F_a+S_b F_b+S_c F_c)+S_{con}(S_d F_d+S_r F_r+S_p F_p) \qquad (13)$$

상기의 식에서 $s_a+s_b+s_c=1$, $s_d+s_r+s_p=1$, $s_{obj}+s_{con}=1$로 한다. s_{obj}하고 s_{con}은 목적함수와 제약함수의 가중치를 나타내는 파라미터이다. 또, $s_x(x\in\{a, b, c, d, r, p\})$는 각각 함수 F_x의 가중치를 나타내는 파라미터이다. 이 코스트함수는 한글의 일반적 문자특징 및 한글이 충족시켜야 할 최소한의 조건을 총괄적이고 상호 보완적으로 나타내고 있다. 가중치 s_a, s_b, s_c, s_r, s_d, s_p, s_{obj}, s_{con}의 구체적인 값에 대해서는 4장에서 검토한다.

3.3 최소화법

상기의 코스트함수의 최소해가 본 방법에서 구하는 해, 즉 문자영역이 된다. 그러나 상기의 코스트함수는 많은 변수를 포함하고 있기 때문에 해석적 방법으로 최소해를 구하는 것은 쉽지가 않다. 따라서 본 방법에서는 조합문제의 최적화에 자주 이용되는 근방탐색법인 Simulated Annealing법을 이용해 최소해를 구한다[12-15]. 최소화 알고리듬을 다음에 나타낸다.

〈step 1〉 모든 r_{ij}, p_i에 대해 초기값을 랜덤하게 준다.
〈step 2〉 계산온도 T에 초기치 $T=T_0$를 준다.
〈step 3〉 (a)(d)까지의 처리를 x회 반복한다.
(a) 현재의 코스트함수 $F_{cost}(P, R)$를 계산한다.
(b) r_{ij}, p_i중에서 대상으로 하는 변수를 랜덤하게 선택한다.
(c) 선택한 변수 r_{ij}, p_i의 변화량 Δr, Δp의 변화에 따른 코스트함수 $F_{cost}(P, R)$의 변화량 $\Delta F_{cost}(P, R)$을 구한다.
(d) 다음의 규칙에 따라 대상으로 하고 있는 변수 r_{ij}, p_i를 변화시킨다.
· $\Delta F_{cost}(P, R)<0$이면 $r_{ij}\leftarrow r_{ij}+\Delta r$, $p_i\leftarrow p_i+\Delta p$
· $\Delta F_{cost}(P, R)\geq 0$이면
$p(\Delta F_{cost}(P, R))=\exp(-\frac{\Delta F_{cost}(P, R)}{T})$ 의 확률로 $r_{ij}\leftarrow r_{ij}+\Delta r$, $p_i\leftarrow p_i+\Delta p$
〈step 4〉 x회에 걸친 〈step 3〉의 처리에서 변수 r_{ij}, p_i가 전혀 변화하지 않으면 종료한다. 그렇지 않는 경우는 계산온도 T를 Simulated Annealing에 따라 T←DT로 낮추고 〈step 3〉으로 간다 (단, D<1).

여기서 변수 r_{ij}, p_i의 범위($0\leq r_{ij}$, $p_i\leq 1$)는 변화량의 간격으로 양자화되어 있다고 가정한다. 〈step 1〉에 의해 설정된 r_{ij}, p_i의 초기값도 양자화된 값을 갖는다. 〈step 3〉의 처리횟수를 나타내는 x는 충분히 큰 것이

바람직하다. 여기서는 모든 r_{ij}에 대해 $+\Delta r$ 혹은 $-\Delta r$의 변화가 $\frac{1}{\Delta r}$ 회 발생할 횟수인 $\frac{2N(N-1)}{\Delta r}$을 x의 값으로 한다.

4. 추출실험 및 검토

4.1 추출실험

실험에 사용된 대상문서로는 신문, 논문, 잡지, 도서카드, 상품 선전문, 비정형 필기 문서 등 50 종류의 문서들이다. 이들 문서는 문자크기가 일정치 않고 가로쓰기, 세로쓰기, 간단한 그림 등이 혼재된 구조를 가지고 있다. 실험에 사용한 화상데이터는 이미지 스케너를 이용해 150~400 dpi로 받았고 Sun-SPARC-classic 워크스테이션의 x-window환경에서 C 언어를 이용해 처리하였다. 프로그램은 약 6천 행 정도이다.

대상문서의 내용을 대상별, 종류별로 나누어 정리하면 표 2와 같다. 표 2에서 화상수는 실험에 사용된 대상 문서화상의 수를 나타내고, 그림수는 그림이 들어있는 대상 문서화상의 수를 나타낸다. 그리고 문자수와 문자열의 수는 대상문서 화상내에 존재하는 내용의 평균 갯수를 나타내고 있다. 특히 문자열은 띄어쓰기의 공백을 기준으로 계산했고, 그림은 선, 사진 등 문자가 아닌 일정한 영역을 가진 것을 기준으로 계산했다. 쉼표, 마침표 등 일정한 영역을 갖지 않는 것은 계산에서 제외하였다. 추출 실험결과 문자는 94.2%, 문자열은 84.5%의 추출율을 얻었다.

〈표 2〉 대상화상의 내용.
〈Table 2〉 Specification of the experimental document image

대상 / 종류	신 문	잡 지	비정형 문서	기 타
화상수	20	10	10	10
문자수	45.2	65.0	40.4	52.6
문자열수	14.8	17.7	14.9	15.1
그림수	6	1	3	6
가로, 세로쓰기	가로·세로	가로·세로	가로·세로	가로·세로

4.2 검 토

목적함수의 역치로 사용하는 파라미터는 $t^q_{\min}=0.1$, $t^q_{\max}=1$, $t^w_{\min}=0.1$, $t^w_{\max}=1$, $t^c_{\min}=0.6$, $t^c_{\max}=2$, $t^m_{\min}=0.7$, $t^m_{\max}=1$, $t^l_{\min}=0.7$, $t^l_{\max}=1$, $t^e_{\min}=0.4$, $t^e_{\max}=1$ 로 실험을 통해 얻었다. 또 최소화 알고리듬 중의 파라미터는 계산온도 T의 초기값으로 $T_0=1$로, T를 감소시키는 비율은 $D=0.9$로, 또 변수 p_i, r_{ij}의 양자화 간격은 0.2로 했다.

그리고 코스트함수 $F_{cost}(P, R)$는 초기 상태에서 유사한 값을 갖는 것이 바람직하기 때문에 각 목적함수, 제약함수가 균등하게 참조되도록 s_a, s_b, s_c, s_d, s_r, s_p, s_{obj}, s_{con}의 값을 정했다. 가중치 s_a, s_b, s_c의 값은 다음의 식에서 결정했다.

$$S_a = \frac{F_a}{F_a + F_b + F_c}$$

$$S_b = \frac{F_b}{F_a + F_b + F_c}$$

$$S_c = \frac{F_c}{F_a + F_b + F_c}$$

그리고 가중치 s_r과 s_d에 대해서는 $F_d(P_0, R_0)$, $F_r(R_0)$, $F_p(R_0)$가 항상 0을 취하기 때문에 s_a, s_b, s_c와 같이 정할 수는 없다. 또 $F_d(P_0, R_0)$, $F_r(R_0)$, $F_p(R_0)$의 값은 0와 무한대 뿐이기 때문에 s_r, s_d, s_p값은 의미를 갖지 못한다. 따라서 여기서는 $s_r=s_d=s_d=1/3$로 주었다. 같은 이유에서 $s_{obj}=s_{con}=1/2$로 하였다.

그림 7과 그림 8은 실험에 사용된 데이타의 일부로써 문자와 문자열이 추출된 결과를 나타내는 예이다. 이 결과는 최소화 알고리즘에 의해 코스트함수를 최소로 하는 해공간 중의 해로써, $p_i \geq 0.5$를 만족하는 영역을 문자영역으로 추출한 것이다. 또 $p_i \geq 0.5$, $p_j \geq 0.5$를 만족하고 $r_{ij} \geq 0.5$를 만족하는 문자영역의 링크를 문자열로써 추출한 결과이다. 그림에서 각 문자를 둘러싸고 있는 외접선은 문자영역을 나타내고 각 문자영역의 중심을 연결한 선은 동일 문자열에 속함을 나타내고 있다.

그림 7(a)는 문자열을 자유롭게 배치해 임의로 만든 비정형화 문서이다. 그림이 존재하고, 각 문자열의 방향과 문자의 크기가 일정하지 않지만 문자와 문자열을 바르게 찾아내고 있다. 표 3에 그림 7(a)에서 "모십니다" 부분에 해당하는 각 문자의 특징량(q, w, c, m, l, e)과 p_i, r_{ij}의 값을 나타낸다.

그림 7(b)는 신문의 사회면의 일부로 가로쓰기와

〈표 3〉 Fig. 7(a)의 "모십니다"에 대한 실험 결과
〈Table 3〉 Experimental results for "모십니다" in Fig. 7(a)

1문자 특징량	대상문자	q(정방율)	w(선분밀집율)	P_i
	모	0.6200	5	1.0
	십	0.7895	5	1.0
	니	0.5397	3	1.0
	다	0.5090	4	1.0
2문자 특징량		c(근접율)	m(크기일치율)	r_{ij}
	모-십	0.7635	0.7332	1.0
	십-니	0.6906	0.9169	1.0
	니-다	0.8410	0.7319	1.0
3문자 특징량		l(직선율)	e(등간격율)	
	모-십-니	0.9295	0.9784	
	십-니-다	0.9567	0.9872	

세로쓰기가 뒤섞여 있고 문자크기도 다양함에도 불구하고 문자영역과 문자열이 정확히 추출되었다.

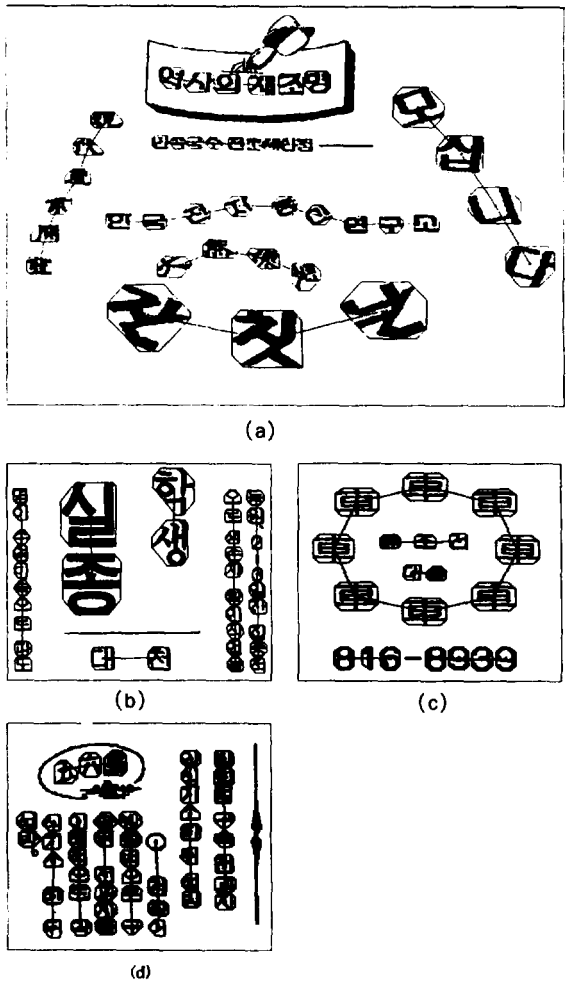
그림 7(c)는 신문의 자동차 판매 광고면의 일부분이다. 비정형화된 형식으로 문자열이 곡선을 이루고 있고 글자의 크기가 다양하지만 문자와 문자열이 바르게 추출되었다. 특히, 전화번호 '1' 부분은 문자의 가로 세로의 비의 차가 매우 큰데도 불구하고 정확하게 추출된 예이다.

그림 7(d)는 부산일보 사회면의 일부분으로 간단한 그림이 들어 있다. 그림은 문자영역의 조건을 충족시키지 못하므로 제외되었으나, 그 속의 글자는 바르게 추출되었다. 또 문자열도 가로로 정확히 추출되었다.

그림 8에는 문자와 문자열 추출이 실패한 대표적인 예를 나타낸다.

그림 8(a)는 신문 광고란의 일부이다. 문자영역은 대부분 바르게 추출되었지만 문자열 일부를 바르게 추출하지 못하였다. 문자들이 가로, 세로 양 방향으로 잘 정열되어 있어 본 함수의 문자 배치특징을 두 방향 모두 충족시킨 경우이다. 본 예에서 실제의 문자열을 구하려면 언어지식이나 문맥의 정보 등을 이용해야 한다.

그림 8(b)는 문자들이 서로 밀착되어 있어 다른 문자의 자소(字素)들이 접속되어 하나의 문자로 추출된 경우이다(그림 8(b)에서 문자 "나"와 "서", "리"와 "며", "가"와 "피"의 경우). 이와 같은 경우는 조합문자인 한글문자가 자소의 2차원 공간배치에 의해 생성

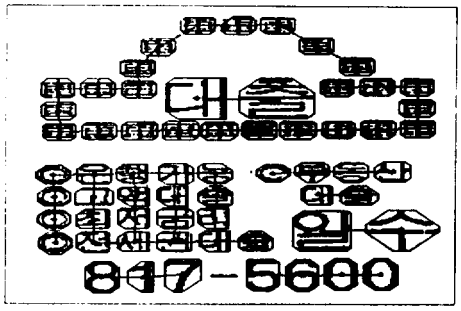


(a)
(b)
(c)
(d)

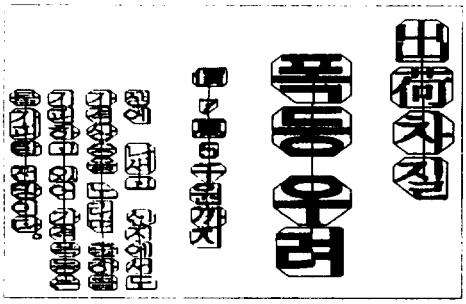
(그림 7) 추출에 성공한 예
(Fig. 7) Examples correctly extracted from a document image

되는 특수성 때문에 빈번히 발생한다. 이것은 본 실험에서 추출에 실패한 것중 가장 빈도수가 높은 한 예이다. 개선책으로는 한글문자의 구조정보를 코스트함수에 적극적으로 반영시키는 방법, 문서의 거시적인 특징을 이용하는 방법 등을 생각할 수 있다.

그림 8(c)는 특징추출의 단계에서 충분한 정보를 얻지못하면 그 결과가 최종추출의 결과에까지 영향을 미치기 때문에 그로 인해 충분한 대처가 안된 경우이다. 예를 들면 파선으로 둘러싸인 부분은 대부분의 문자가 굵고 큰데 비해 상대적으로 해상도가 낮아



(a)



(b)

(c)

(그림 8) 추출에 실패한 예
(Fig. 8) Examples erroneously extracted from a document image

문자선이 결함되어 문자영역으로의 조건을 충족시키지 못해 문자영역으로 바르게 추출되지 못했다.

마지막으로 각 대상문서에 대해 추출시간을 검토하였다. 문자의 갯수가 많고 또 떨어진 자소의 갯수가 많을 수록 많은 시간을 필요로 했다. 그림 7(a) (b) (c) (d)는 각각 Sun-SPARCclassic의 x-window환경에서 3분 40초, 3분 24초, 1분 31초, 5분 10초씩 걸렸다. 시간이 많이 걸린 이유로는 최소화 과정에서 화상전체의 연결영역에 대해 문자영역의 가능성, 동일 문자열의 가능성을 서로 비교하기 때문이다. 이들을 해결하기 위해서는 해 공간을 부분공간으로 분할해 각 부분공간에 대해 병렬적으로 비교를 하므로써 해결되리라 생각된다. 이 방법에 대해서는 현재 검토중이다.

5. 결 론

범용적인 문자영역의 추출을 위해서는 대상문서에 의존하지 않는 정보를 이용할 필요가 있다. 본 논문에서는 이와 같은 정보인 문자의 정방성, 문자의 근접성, 크기의 일치, 등간격 배치, 직선적 배치 등을 일반적인 문자특징의 준거로 삼았다. 본 방법은 상기 문자특징들을 문자영역이 충족시켜야 할 조건으로 설정하여, 이 조건을 종합적으로 충족시키는 영역을 문자영역으로 추출해 내는 방법이다. 구체적으로 말해서, 이들 조건들이 종합적으로 충족되는 해를 최소값으로 갖는 코스트함수로 도입하고, 이 함수를 Simulated Annealing법을 이용해 최소화함으로써 영역추출을 완성하는 방법이다.

본 방법의 특징은 영역추출의 문제를 코스트 최소화 문제로 간주해 코스트함수를 정의한다는 점에서 여타 방법들과 구별된다.

정형화된 문서 뿐만이 아니라, 비정형화된 문서, 그림 등이 혼재하는 문서들에 대해 추출실험을 한 결과, 높은 추출율을 얻어 본 방법의 유효성을 확증하였다. 또, 추출에 실패한 문서에 대한 검토를 통해서 그 원인을 명확히 규명했다.

문서가 가지고 있는 일반적인 특징을 종합적이며 상호보완적으로 고려할 수 있는 코스트함수를 이용한 결과, 다양한 문서 형식, 다종의 문자크기 등에 대해 높은 추출율을 가지는 것으로 보아, 일반 화상의 이해를 위한 정보처리에도 코스트최소화 개념이 유효할 것임이 시사되었다. 그러나 보다 실용적인 추출능력을 가지기 위해서는 적절한 코스트함수의 도입 등이 요구된다.

문제점으로는 대상문서의 문자수가 많아질 경우, 그 처리시간이 많이 소요된다는 점이다. 이를 해결하기 위해서는 코스트함수의 최소화법을 개선하는 방향도 고려할 수 있지만, 그 보다는 병렬적으로 처리

하는 등의 새로운 방법 모색이 더 적합하리라 판단된다. 또 다른 문제점으로는, 본 방법이 문자의 특징추출 과정에서의 오류가 최종결과에까지 영향을 미칠 수 있는 보텀업(bottom-up)적인 접근이기 때문에 거시적인 시점에서 얻어진 특징도 코스트함수에 반영할 필요가 있다.

참 고 문 헌

[1] 도정인, "한글문서 인식 시스템의 개발," 정보과학회지, vol.9, No.1, pp.23-32, 1991.

[2] M.Minoh, "Document Image Analysis," The Journal of the Institute of Electronics, Information and Communication Engineers (IEICE Journal), vol.76, No.5, pp.502-509, 1993.

[3] T. Akiyama and I. Masuda, "A Method of Document-image Segmentation Based on Projection Profiles, Stroke Densities and Circumscribed Rectangles," IEICE Trans., vol.J69-D, No.8, pp. 1187-1195, 1986.

[4] G.Nagy and S.Seth, and M.Viswanathan, "A Prototype Document Image Analysis System for Technical Journals," IEEE Comput. Special issue on Document Image Analysis System, pp.10-22, 1992.

[5] K.Kise, J.Sugiyama, N.Babaguchi and Y.Tezuka, "Layout Model Based Analysis of Document Structure," IEICE Trans., Vol.J72-D-2, No.7, pp. 1029-1039, 1989.

[6] T.Akiyama and N.Hagita, "Automated Entry System for Printed Documents," Pattern Recognition, Vol.23, No.11, pp.1141-1154, 1990.

[7] A.Antonacopoulos and R.T.Ritchings, "Flexible Page Segmentation Using the Background," IAPR, 1994.

[8] O'Gorman L.: "The Document Spectrum for Page Layout Analysis," IEEE Trans. Pattern Anal. & Mach. Intell., Vol.15, No.11, pp.1162-1173, 1993.

[9] K.Goyhten, N.Babaguchi and T.Kitahashi, "Constraint Satisfaction Approach to Extraction of Japanese Character Region from Unformatted Document Image," IEICE Trans. Inf. & Syst., Vol. E78-D, No.4, pp.466-475, 1995.

[10] H.Goto and H.Aso, "Robust and Fast Text-Line Extraction Using Local Linearity of the Text-Line," IEICE Trans., vol.78-D-II, No.3, pp.465-473, 1995.

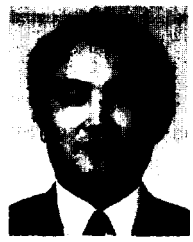
[11] 岩城, 久保田, 荒川, "近接線密度法による文字·圖形分離抽出," 日信學論(D), J68-D, 4, pp.821-828, 1985.

[12] B.E.Rosen, "Simulated Annealing and its Applications," Technical Report of IEICE, AI93-59, pp. 1-8, 1993.

[13] H.L.Tan, S.B.Gelfand and E.J.Delp: "A Cost Minimization Approach to Edge Detection using Simulated Annealing," IEEE Trans. Pattern Anal. & Mach. Intell., Vol.14, No.1, pp.3-18, 1991.

[14] X.Xie, R.Sudhakar and H.Zhuang, "Corner Detection by a Cost Minimization Approach," Pattern Recognition, Vol.26, No.8, pp.1235-1243, 1993.

[15] G.A.Tagliarini, J.F.Christ and E.W.Page: "Optimization Using Neural Networks," IEEE Trans. Comput., Vol.40, No.12, pp.1347-1358, 1991.



김 석 태

1983년 光云大學校 電子工學科 卒業(學士)

1988년 京都工藝纖維大學 大學院 電子工學科 卒業(工學碩士)

1991년 大阪大學 通信工學科 卒業(工學博士)

1991년~現在 釜山水産大學校 情報通信科 助教授

관심분야: 화상처리, 패턴인식, 멀티미디어통신, 지적 CAI등