

# 개선된 SOG 기반 고속 세선화 알고리즘(SOG\*)

이 찬 회<sup>†</sup> · 정 순 호<sup>††</sup>

## 요 약

본 논문은 기존의 신경망을 이용한 세선화 방법 중에서 자기 구성 그래프(Self-Organized Graph : SOG) 세선화 기법의 우수한 세선화 결과를 유지하면서, 수행 속도를 향상시키기 위하여 Kohonen Features Map의 새로운 점층 기법을 변형된 SOG에 적용한 개선된 SOG(Improved SOG : SOG\*) 세선화 기법을 제안한다. 실험 결과로써 숫자와 문자 모두 기존의 SOG와 같은 우수한 세선화 결과를 나타내며,  $O((\log M)^3)$ 의 시간 복잡도를 가지는 속도 향상을 이루었다. 따라서 본 논문에서 제안한 방법은 숫자 또는 문자 인식에 있어 특징 추출의 빠른 전처리 과정으로 사용할 수 있다.

## Fast Thinning Algorithm based on Improved SOG(SOG\*)

Chan-Hee Lee<sup>†</sup> · Soon-Ho Jung<sup>††</sup>

### ABSTRACT

In this paper, we propose Improved Self-Organized Graph(Improved SOG : SOG\*) thinning method, which maintains the excellent thinning results of Self-organized graph(SOG) built from Self-Organizing features map and improves the performance of modified SOG using a new incremental learning method of Kohonen features map. In the experiments, this method shows the thinning results equal to those of SOG and the time complexity  $O((\log M)^3)$  superior to it. Therefore, the proposed method is useful for the feature extraction from digits and characters in the preprocessing step.

**키워드 :** SOG\*, 세선화(thinning), Kohonen 네트워크(Kohonen network), 자기구성그래프(SOG), L\* 알고리즘

### 1. 서 론

문자 인식에 있어 인식 성능 향상을 위하여 일반적으로 두 가지 측면을 고려하게 된다. 하나는 실제 인식 단계에서 입력으로 사용하게 될 인식 문자 패턴에 대한 전처리 및 특징 추출에서 인식에 필요한 정보를 최소화하는 것이고, 다른 하나는 특정 패턴에 국한되지 않는 일반화 및 객관적 적응 능력을 가지면서 유사 패턴에 대한 인식률을 높일 수 있는 분류기 적용에 관한 것이다[1-5].

이 두 가지 측면 중에서, 전자의 측면에서는 전처리 단계의 세선화 과정을 통하여 방대한 이미지 정보를 축소시켜 이 축소된 정보가 인식을 위한 다음 단계로 사용됨으로써 효율을 높이게 된다. 따라서 세선화에 있어 가장 중요한 것은 원래 이미지의 구조적인 특징을 유지하는 것이다[6].

지금까지 최적 세선화 알고리즘은 크게 직접 기법과 간접 기법으로 나눌 수 있다. 직접 기법은 원래 이미지로부터 픽셀을 하나씩 제거해 나감으로써 세선화를 수행하는 것이고,

간접 기법은 이미지의 특성 유형을 반영하는 골격선을 생성하는 것이다[6, 7]. 두 가지 기법 중에서 간접 기법이 직접 기법에 비해 우수한 성능을 갖는다[8].

이러한 간접 기법 가운데 신경망을 이용한 방법으로는 이미지 구조에 따라서 자신을 조직하는 특성을 갖는 Kohonen의 자기 구성 특징 지도(Self-Organized feature Map : SOM)를 이용한 세선화 수행 연구가 진행되었고, 이 방법의 단점을 개선한 자기 구성 특징 그래프(Self-Organized feature Graph)가 제안되었다[8, 9]. 이 SOG 방법은 템플릿 윈도우를 이용한 기법[10]과 ART2 신경망을 이용한 클러스터링 방법[11]에 비하여 그 과정이 단순하고, 세선화 결과가 우수하다. 그러나, 학습 단계에서 뉴런들을 재정돈하는 단계로 인하여 학습 시간이 오래 걸리는 단점을 갖는다.

따라서 본 논문은 SOG의 우수한 세선화 결과를 가지면서 전체 수행 시간의 감소를 가져오는 SOG\* 세선화 방법을 제안하고 그 실험 결과를 보인다.

본 논문의 구성은 2장에서 관련된 연구를 설명하고, 3장에서는 SOG와 L\* 알고리즘을 적용한 개선된 SOG\* 방법을 소개하며 4장에서는 실험 및 실험 결과 분석을 하며, 마지막 5장에서는 결론을 맺는다.

† 준 회 원 : 부경대학교 대학원 전자계산학과  
 †† 정 회 원 : 부경대학교 컴퓨터멀티미디어공학부 교수  
 논문접수 : 2001년 10월 23일, 심사완료 : 2001년 11월 12일

2. 관련 연구

2.1 자기 구성 그래프(Self-Organized Graph : SOG)

선형 네트워크에서 SOM의 경쟁층의 각 뉴런들은 두 개의 인접 뉴런들을 가지며 그 인접성은 항상 고정되어 있다. 이차원 입력 벡터와 일차원 경쟁층을 갖는 SOM에서 입력층에 입력 패턴이 들어오면 경쟁층의 각 뉴런들은 식 (1)에 의해 승자 뉴런을 찾게 된다[12, 13].

$$winning\ unit\ c = \min_{1 \leq i \leq n} \|X - W_i\| \quad (1)$$

여기서 X는 입력 벡터이고 W<sub>i</sub>는 경쟁층의 i 뉴런에 연결된 가중치 벡터이다. 승자 뉴런이 결정되면 승자 뉴런과 그 이웃에 속하는 뉴런들의 가중치들은 식 (2)에 따라 학습된다.

$$W_i(t+1) = \begin{cases} W_i + \alpha(t)(X - W_i(t)), & i \in N_c(t) \\ W_i(t), & otherwise \end{cases} \quad (2)$$

식 (2)에서 N<sub>c</sub>(t)는 t시간에서 뉴런 c의 이웃을 말하며, α(t)는 학습률을 나타낸다[12, 13].

SOG는 SOM에서 학습되는 동안 고정된 인접성을 입력 벡터로부터의 거리에 따라 뉴런들을 재정렬함으로써 수정하게 된다. 재정렬 된 뉴런들은 네트워크의 재조직에 있어 향상된 능력을 가지며, 이 효과를 학습 알고리즘에 적용하기 위하여 SOM의 가중치 수정 방법인 식 (2)를 다음의 식 (3)과 같이 수정하여 사용하게 된다[8].

$$w_i(t+1) = w_i(t) + \zeta(t, K_i) \times (X - w_i(t))$$

$$\zeta(t, K_i) = \exp - \left[ \frac{\ln(K_i)}{\alpha(t)} \right]^2 \quad (3)$$

식 (3)에서 X는 입력 벡터이고, K<sub>i</sub>는 정돈된 뉴런들의 초기 인덱스 값이며, α(t)는 학습률이다.

SOG의 기본 구조는 SOM과 동일하며, 단지 가중치 수정에 있어 식 (3)을 사용하여 각 뉴런들의 정렬 순서도 가중치 수정의 우선 순위에 영향을 미치도록 한다[8].

2.2 L\* 학습 알고리즘

L\* 학습 알고리즘은 기존 SOM에서 경쟁층의 승자 뉴런을 찾을 때 소요되는 시간을 줄여 전체 학습 시간을 감소시키는 알고리즘이다[14].

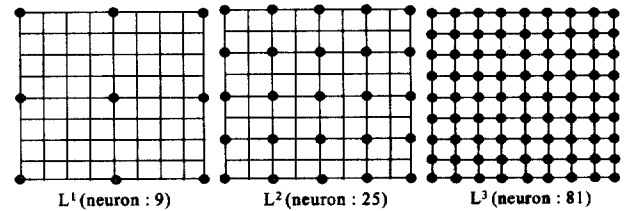
SOM이 승자 뉴런을 찾는 방식은 모든 뉴런을 대상으로 하지만, L\* 알고리즘은 승자 뉴런을 찾을 때 모든 뉴런을 대상으로 하는 것이 아니라 단계별로 뉴런들의 수를 점차 늘여 가면서 승자 뉴런을 찾기 때문에 기존의 SOM 방식보다 계산 시간이 단축되는 효과를 얻을 수 있다[14].

L\* 알고리즘은 1단계부터 log<sub>2</sub>(M-1) 단계까지 학습을 한다. 여기서 M은 경쟁층 맵의 크기를 의미한다. 이때 각 단계

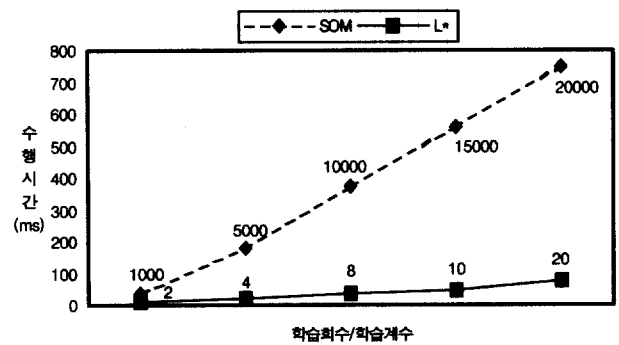
별로 학습할 뉴런의 개수는 1차원 맵의 경우 2<sup>level</sup>+1이고, 2차원 맵인 경우는 (2<sup>level</sup>+1)<sup>2</sup>이 된다. level은 각 단계를 의미한다. 단, 각 단계별로 학습할 뉴런의 개수에 실험치로 결정되는 학습 계수를 곱한 만큼을 학습 횟수로 결정한다. 학습 계수가 20 이상이면 완전한 자기 구성 특징 맵을 구성한다[14].

한 단계가 끝나면 다음 단계에 학습해야 할 뉴런을 추가해야 하는데, 이때 추가하는 뉴런들은 이전 단계에서 학습된 뉴런들을 이용한다.

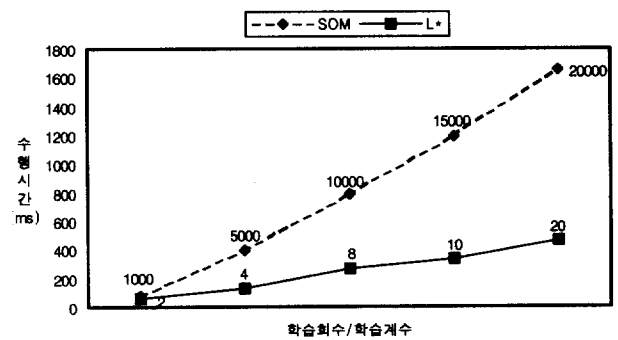
다음 (그림 1)에서 이차원 9×9(전체 81개의 경쟁층 뉴런) 맵에서 단계별로 학습해야 할 뉴런을 나타내고 있다. 1단계에서는 (2<sup>1</sup>+1)<sup>2</sup>=9의 수만큼 뉴런을 선택하는데, 맵에서 코너에 해당하는 뉴런과 그들 중심에 해당하는 뉴런이 선택된다. 2단계 이후부터는 이전 단계에서 선택된 뉴런들 사이에서 선택되며, 최종 단계에서는 전체 뉴런이 선택된다[14].



(그림 1) 9×9 맵에서 단계별로 학습 할 뉴런



(a) 9×9 map 수행 시간 비교



(b) 17×17 map 수행 시간 비교

(그림 2) SOM과 L\* 수행 시간 비교

(그림 2)에는 각각 기존의 SOM 방식과 L\* 방식에 대한

이차원 9×9, 17×17 맵에 대한 시간 측정 결과를 나타내었다. SOM은 학습 횟수가 20,000번일 때 완전한 맵을 구성하고, L\* 알고리즘은 학습 계수(A)가 20일 때 즉, 학습횟수가

$$\sum_{level=0}^{\log_2(M-1)} A \times (2^{level} + 1)^2 \text{이 될 때 완전한 맵을 이루었다.}$$

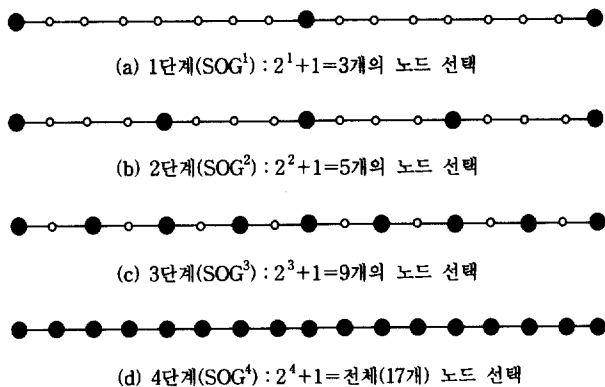
(그림 2)에 보이는 것처럼 실험 결과, L\*를 적용한 방식이 기존 SOM보다 시간을 많이 감소시키면서 자기 구성 조직 맵을 구성함을 알 수 있다.

### 3. 개선된 자기 구성 그래프(Improved SOG : SOG\*)

#### 3.1 SOG\*의 학습 알고리즘

SOG 알고리즘은 학습이 끝날 때까지 각 단계에서 입력이 들어올 때마다 입력 패턴과 경쟁층의 뉴런들과의 거리를 정렬해야 한다[8]. 이 학습 과정에서 뉴런들이 각 입력 패턴의 구조를 나타내는 위치를 찾아 안정화되는데, 이렇게 안정화를 이루게 되면 더 이상 학습할 필요가 없음에도 불구하고 기존 SOG는 안정화 후에도 계속된 학습으로 많은 시간을 소비하는 단점을 가지게 된다. 이러한 단점을 개선하기 위하여 SOG의 변형을 사용하는데, 이는 SOG에서 안정화되는 시점을 찾아 이후 학습에서는 기존 정렬을 그대로 사용하는 방법이다((그림 4)의 ㉔). 또한, 세션화를 위한 학습 과정에서 경쟁층의 뉴런을 재정돈하는 과정을 거쳐서 승자 뉴런을 선택할 때 모든 뉴런을 한번에 모두 재정돈 하는 것이 아니라 재정돈 할 뉴런의 수를 단계별로 증가 시켜가며 학습시키는 점진적 학습 알고리즘[14]을 사용하여 세션화를 수행하고자 한다((그림 4)의 ㉕).

이 과정은 (그림 3)과 같이 나타낼 수 있다. 첫 번째 단계인 SOG<sup>1</sup>((그림 3) (a)에서는 제일 끝의 두 노드와 그들 중심의 노드((그림 3)의 검은색 노드들)가 선택되어 학습을 하게 되며, 두 번째 이후 단계부터는 이전 단계에서 선택되었던 n개의 노드들 사이에서 선형 보간법을 이용하여 n-1개의 새로운 노드를 선택하게 된다. (그림 3)에서 흰색으로 표시된 뉴런은 현 단계에서 학습에 참여하지 않는 노드들이다.



(그림 3) SOG\* 학습 과정

이러한 SOG\* 알고리즘은 (그림 4)와 같고, 이는 L\* 알고리즘을 SOG 변형에 적용한 방식이다.

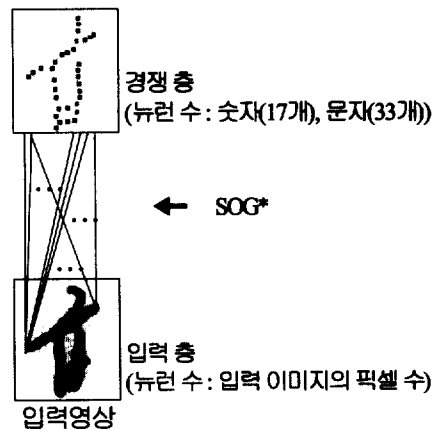
알고리즘의 ㉔는 가중치를 수정하는 부분이며, 이 가중치는 최초 단계에서는 random 값을 부여하게 되지만, 두 번째 단계 이후부터는 이전 단계에서 선택된 뉴런들을 기반으로 가중치를 부여하게 된다. ㉕ 부분은 경쟁층 뉴런들을 재정렬하는 부분이며, 입력 패턴이 들어오게 되면 승자 뉴런이 선택되고 나머지는 입력 패턴과의 거리에 따라 정렬되는데 이때 이전의 정렬 순서와 현재의 정렬 순서가 같다면 가중치들이 안정화된 것을 의미하므로 더 이상 정렬을 수행하지 않는다. ㉕는 실제 학습하는 부분이며 승자 뉴런과 이웃 뉴런의 가중치를 갱신하며 학습을 수행하게 된다.

```

for level := 1 to log2(M-1) do
begin
/* 학습 할 뉴런 선택 */
node := 2level + 1;
/* 가중치 부여 */
give the weight of step ;
for repeat := 1 to delta * level do
begin
for input := 1 to level do
begin
/* 입력 패턴과 경쟁층 뉴런의 거리 정렬 */
if sorted orders = previous orders
then Ki := previous orders ;
else 입력과 노드들의 거리에 의한 정렬 ;
/* 학습 */
update weight of winning unit ;
update weight of neighborhood unit ;
end ; // end for input
end ; // end for repeat
end ; // end for level
    
```

(그림 4) SOG\* 알고리즘

이 방법에 사용되는 신경망의 구조는 (그림 5)와 같은 이차원 입력 벡터와 일차원 경쟁층을 가진 Kohonen network 이다.

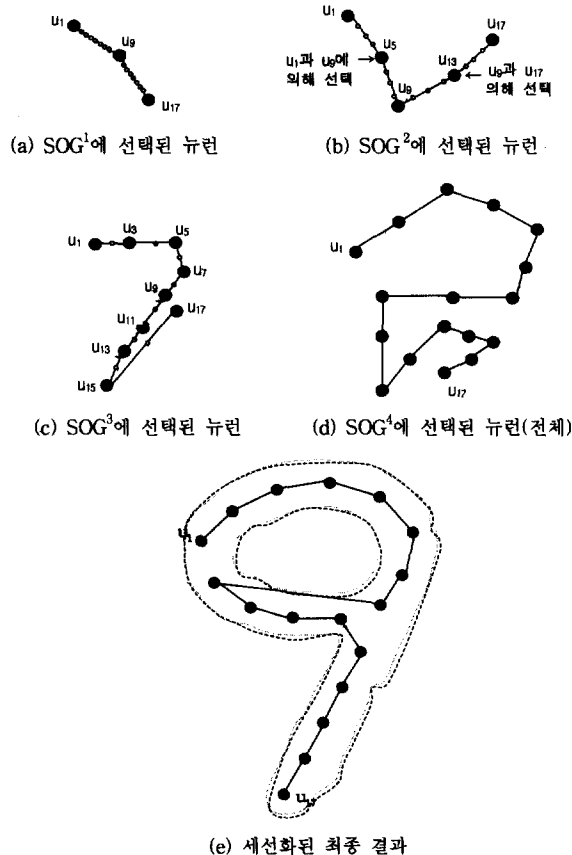


(그림 5) SOG\* 구조

SOG\*에서 입력 영상은 이차원 입력 구조로서 0과 1사이의 이차원 좌표로 변환되어 사용되며, 경쟁층은 실험을 통하여 숫자 이미지는 17개, 문자 이미지는 33개의 고정된 수의 뉴런들이 일차원으로 이루어진다.

학습을 위해서는 각 단계별로 뉴런의 수를 증가시켜 가면서 학습을 하며 다음으로 뉴런과 입력과의 거리를 계산하고 이 값들을 다시 정돈한다. 마지막으로 이 정돈된 값들의 순서를 반영하여 가중치를 수정한다. 가중치를 수정하는 방법은 식 (3)과 같다[8].

숫자 이미지를 예로 들면, 17개의 뉴런으로 세선화를 수행하므로 총 4단계로 이루어진다. 각 단계별로 학습할 뉴런의 개수는 3, 5, 9, 17이 된다. 최초에 학습할 3개의 뉴런은 random 가중치로써 결정하여 학습을 한다. 다음 단계에서는 5개의 뉴런이 학습에 참가하게 되는데, 이때는 이전에 학습한 3개의 뉴런과 각 뉴런들 사이에 연결되어 있는 뉴런들이 추가되어 학습하게 된다. 두 번째 단계부터는 선형 보간법[15]을 이용하여 새로 추가되는 경쟁층 뉴런들의 가중치를 결정한다. 선형 보간법을 사용하는 이유는 이전 단계에서 학습된 뉴런들의 가중치를 그대로 유지함으로써 전체 뉴런들의 가중치들이 안정화되는 시점을 빨리 찾아 학습에 소요되는 시간을 단축하기 위해서이다. 세 번째 이후 단계도 두 번째 단계와 동일한 방법으로 수행을 한다.



(그림 6) 숫자 9가 학습되는 과정

(그림 6)에서 숫자 이미지 9에 대해 학습하는 단계를 설명하고 있으며, 그림에 표시된  $u_n$ 은 뉴런의 번호이다. 최초 학습 단계에서는 (그림 6) (a)와 같이 3개의 뉴런이 선택되어 학습을 하게 되는데, 이때 선택되는 뉴런은 외곽의 두 뉴런과 그들의 중심 뉴런인  $u_1, u_9, u_{17}$ 이 선택되게 된다. 두 번째 단계는 (그림 6) (b)와 같이 이전 단계에서 선택되었던 뉴런들 사이에 있는 뉴런들을 선택하여 학습에 참여하게 된다. 이후 단계도 두 번째 단계와 동일한 방법으로 (그림 6) (c), (d)와 같이 수행하게 된다. (그림 6) (e)는 4단계까지 모든 과정이 끝난 최종 세선화 결과를 나타내고 있다.

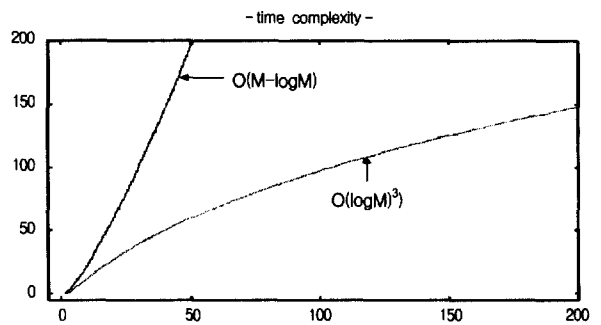
(그림 6)에서 검은색으로 표시된 뉴런은 현 단계에서 선택되어 학습에 참여할 뉴런이고, 작은 점으로 표시된 뉴런은 현 단계에서 선택되지 않은 뉴런을 나타낸다.

### 3.2 SOG\* 알고리즘 분석

기존의 SOG는 학습시에 항상 모든 뉴런들을 참가시키기 때문에  $O(M \log M)$ 의 시간 복잡도를 나타낸다. 반면, SOG\* 세선화기법은 학습시에 집중적으로 뉴런들을 참여시킨다. 단계는 1단계부터  $\log_2(M-1)$ 단계까지이고, 각 단계별로 학습 계수를 곱한 만큼을 학습 회수로 수행하게 된다. 따라서 (그림 7)과 같은 수식을 유도할 수 있다. 여기서 level은 단계를 말하며, C는 단계당 학습 회수를 의미한다. 유도된 수식에 의해서 SOG\* 세선화기법의 시간 복잡도가  $O((\log M)^3)$ 임을 알 수 있다.

$$\begin{aligned}
 & C \cdot \sum_{level=1}^{\log(M-1)} (2^{level} + 1) \cdot \log(2^{level} + 1) \\
 &= C \cdot (3 \cdot \log 3 + 5 \cdot \log 5 + \dots + M \cdot \log M) \\
 &\leq C \cdot \left( \sum_{k=1}^{\log(M-1)} k \cdot \log k \right) \\
 &\leq C \cdot \left( \sum_{k=1}^{\log(M-1)} k^2 \right) \\
 &= O((\log(M-1))^3) = O((\log M)^3)
 \end{aligned}$$

(그림 7) SOG\* 세선화기법의 시간 복잡도



(그림 8) 두 알고리즘의 시간 복잡도

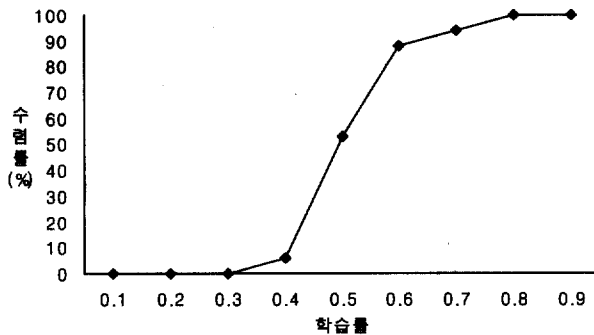
기존 SOG와 SOG\* 세선화기법의 시간 복잡도를 비교한

그래프를 (그림 8)에 나타내었다. SOG는 뉴런의 수가 증가함에 따라 기하급수적으로 증가하는 반면 SOG\* 세션화기법은 증가량이 아주 적음을 알 수 있다. 따라서 문자 인식의 기본이 되는 전처리 과정의 세션화에서 많은 시간의 감소를 가져 올 수 있다.

4. 실험 및 실험 결과

실험은 펜티엄-III 1GHz와 Windows 2000 환경에서 VC++ 6.0으로 구현하여 실험되었고, 사용된 이미지들은 (그림 10) (a)와 같은 한글, 한자, 일문자, 영문자와 숫자들이다. 이들 이미지들은 국내외 여러 대학 연구실의 샘플 데이터와 임의의 필기자가 필기한 데이터를 사용하였다.

숫자 이미지, 문자 이미지 각각 17, 33개의 경쟁층 뉴런을 사용하였고, 단계별 학습 계수는 20으로 고정하였으며 학습 횟수는 최대 680회가 된다. 초기 경쟁층의 가중치는 0.4부터 0.6 사이의 random weight로 부여하였다[8, 12, 13]. 학습률은 (그림 9)의 그래프와 같이 실험에 의해서 0.8로 결정하였는데, 실험적으로 학습률 0.8에서 완전한 맵을 구성하였다. 각 이미지의 크기는 30×30~200×200픽셀 범위에서 실험을 하였다.



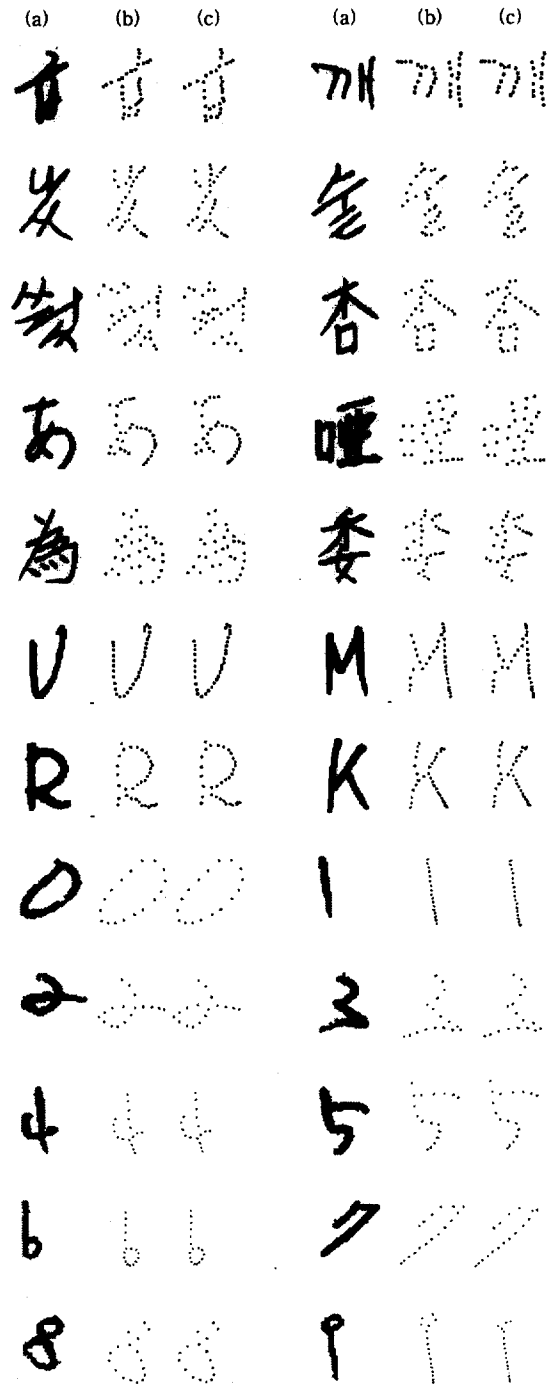
(그림 9) 학습률에 따른 수렴률

(그림 10)의 (b)와 (c)는 SOG\* 세션화기법의 결과를 기존의 SOG와 비교해서 나타내었다. SOG\* 세션화기법은 각 숫자와 문자 이미지에 대한 세션화의 결과가 기존의 SOG 방식과 같이 우수한 결과를 나타내고 있으며, 세션화 결과뿐만 아니라 그 수행 속도 면에서 기존의 방식보다도 훨씬 우수함을 나타내었다.

(그림 11)은 이 방법을 적용한 방식이 기존의 SOG 방식보다 시간적인 측면에서 우수함을 증명해 준다. 먼저 (그림 11)의 (a)에서는 실험의 결과로 얻어진 각 숫자 이미지에 대한 수행 시간을 비교하였고, (b)에서는 뉴런의 수에 따른 두 방법의 수행 시간을 비교하였다.

숫자에 대한 수행 시간과 경쟁층의 전체 뉴런의 개수에 따른 수행 시간을 비교해 본 결과 기존에 모든 뉴런을 정돈하는 SOG 방법 보다 새롭게 제시한 SOG\* 세션화기법이 우수

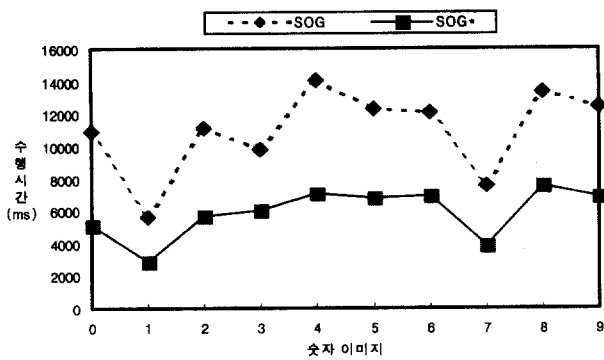
함을 나타내었다.



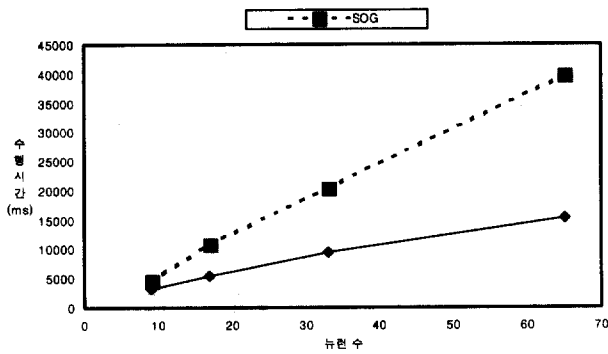
(a) 원 이미지, (b) SOG\* 결과, (c) SOG 결과

(그림 10) 알고리즘별 세션화 결과

또한, 세션화 결과가 우수하여 문자의 특성을 그대로 유지하고 있다. 문자 인식에 있어서 세션화는 문자의 골격 특성을 유지하여 실제 인식의 중요한 입력 정보로 사용되기 때문에 특성이 그대로 유지되는 SOG\* 세션화 기법이 효과적으로 사용될 수 있다.



(a) 숫자별 알고리즘 수행 시간



(b) 뉴런 수에 따른 수행 시간

(그림 11) SOG와 SOG\*의 수행 시간 비교

## 6. 결 론

본 논문은 기존 SOG 방법의 우수한 세선화 결과를 동일하게 가지면서 세선화 처리 속도를 빠르게 하는 SOG\* 알고리즘을 제안하였다.

이 방법은 학습 속도를 증가시키는 Kohonen Features Map의 L\* 학습 알고리즘과 세선화 결과가 우수한 SOG의 변형을 사용한 것으로 알고리즘 분석 결과 기존의 SOG가  $O(M \log M)$ 의 시간 복잡도를 가지는 반면에 제안한 방법은  $O((\log M)^3)$ 의 시간 복잡도를 가짐을 알 수 있었다. 또한 숫자 이미지와 문자 이미지에 대한 실험 결과로 이 알고리즘의 시간 복잡도에 따르는 성능 평가를 확인할 수 있었고, 각 이미지에 대해 그 특성을 그대로 유지하면서 세선화 과정을 기존의 SOG 방법보다도 빠르게 수행하는 우수한 결과를 얻었다.

따라서 문자 인식의 전처리로 사용되는 세선화에서 본 논문의 세선화 방법을 사용한다면 이미지의 특성을 유지하는 우수한 세선화 결과와 빠른 속도를 낼 수 있고, 숫자 또는 문자 인식의 특징 추출의 전처리 과정으로 유용하게 사용할 수 있다.

## 참 고 문 헌

[1] Keiji Yamada, "Inverse Recall Neural Network Model and Feedback Pattern Recognition," IEEE International Conference on Neural Networks, pp.399-406, 1993.  
 [2] Lianwen JIN, Kwokping CHAN and Bingzheng XU, "Off-line Chinese Handwriting Recognition using Multi-Stage

Neural Network Architecture," IEEE International Conference on Neural Networks, pp.3083-3088, 1995.

[3] 방극준, 조남신, 강창언, 홍대식, "필기체 숫자인식을 위한 병렬 자구성 계층 신경회로망", 전자공학회논문지, 제33권 B편 제7호, pp.1193-1202, 1996.  
 [4] 이성환, "패턴 인식의 원리 2", 홍릉과학출판사, 1997.  
 [5] 이성환, "문자 인식 이론과 실제 II", 홍릉과학출판사, 1997.  
 [6] Holt. C. M., A. Stewart, "A parallel thinning algorithm with fine grain subtasking," Parallel Comput. 10, pp.329-334, 1989  
 [7] L. Lam., S. Lee and C. Y. Suen, "Thinning methodologics - a comprehensive survey," IEEE Trans. Pattern Analysis and Machine Intelligence, 14(9), pp.869-885, 1992.  
 [8] P. Ahmed, "A neural network based dedicated thinning method," Pattern Recognition, 16, pp.585-590, 1995.  
 [9] M. Altuwaijri, M. Bayoumi, "A New Thinning Algorithm for Arabic Characters Using Self-Organizing Network," Circuits and Systems, ISCAS '95, IEEE International Symposium on, Vol.3, pp.1824-1827, 1995.  
 [10] John Cowell, Fiaz Hussain, "Thinning Arabic Characters for Feature Extraction," IEEE Information Visualization, Proceedings, Fifth International Conference, pp.181-185, 2001.  
 [11] Majid M. Altuwaijri, Magdy A. Bayoumi, "A Thinning Algorithm for Arabic Characters Using ART2 Neural Network," IEEE Transactions on Circuits and Systems-II, Vol.45, No.2, 1998.  
 [12] J. E. Dayhoff, "Neural Network Architectures," Van Nostrand Reinhold, 1990.  
 [13] S.Haykin, "Neural Networks," Prentice Hall, 1994.  
 [14] Y. P. JUN, H. S. Yoon, J. W. CHO, "L\* Learning : A Fast Self-Organizing Feature Map Learning Algorithm Based on Incremental Ordering," IEICE Trans. INF & SYST., Vol.E76-D, No.6, June, 1993.  
 [15] I. Gyoum, P. Albrecht, Y. Le Cun, J. Denker, W. Hubbard, "Design of a Neural Network Character Recognizer for a Touch Terminal," Pattern Recognition, Vol.24, No.2, pp. 105-119, 1991.

### 이 찬 희

e-mail : chlee@aisol.pknu.ac.kr  
 2000년 부경대학교 전자계산학과 학사  
 현재 부경대학교 전자계산학과 석사과정  
 관심분야 : 인공지능, 신경망, 패턴인식, 컴퓨터 비전

### 정 순 호

e-mail : shjung@pine.pknu.ac.kr  
 1980년 서울대학교 수학교육과 졸업  
 1982년 한국과학기술원 전산학과 석사  
 2000년 한국과학기술원 전산학과 박사  
 1982년~1986년 삼성전자 종합연구소 주임 연구원

1986년~1987년 C&B Technology Co. 개발부장  
 1987년~현재 부경대학교 컴퓨터멀티미디어공학부 교수  
 관심분야 : 인공지능, 신경회로망, 패턴인식, 컴퓨터 시각, 컴퓨터 보안