

유연한 뉴런의 적응적 SOG*

이찬희*, 이상훈**, 장수미**, 정순호
*부경대학교 전자계산학과
**부경대학교 전산정보학과
부경대학교 전자컴퓨터정보통신공학부
e-mail:chlee@aisol.pknu.ac.kr

Adaptive SOG* with Flexible Neurons

Chan-Hee Lee*, Sang-Hoon Lee**, Soo-Mi Jang*, Soon-Ho Jung

*Dept of Computer Science, PuKyong National University

**Dept of Computer and Information, PuKyong National University

Faculty of Electronics, Computer, Telecommunication Engineering, PuKyong National University

요약

본 논문은 기존의 신경망을 이용한 세션화 기법 중 빠른 속도와 우수한 세션화 결과를 가지는 SOG* 세션화 기법에서 해결하지 못한 특수한 경우를 해결하는 적응적 SOG* 세션화 기법을 제안한다. 실험 결과로써 숫자와 문자에 대해서 유사한 수행 속도와 향상된 결과를 나타내었다. 따라서 제안된 방법은 숫자 또는 문자 인식에 있어 특징 추출의 빠른 전처리 과정으로 사용할 수 있다.

1. 서론

문자 인식에 있어 인식 성능 향상을 위해 실제 인식 단계에서 입력으로 사용하게 될 인식 문자 패턴에 대한 전처리 및 특징 추출에서 인식에 필요한 정보를 최소화하는 방법을 고려하게 된다. 전처리 단계의 세션화 과정을 통하여 방대한 이미지 정보를 축소시켜 이 축소된 정보가 인식을 위한 다음 단계로 사용됨으로서 효율을 높이게 된다[1].

지금까지 최적 세션화 알고리즘은 크게 간접 기법과 직접 기법으로 나눌 수 있다. 간접 기법은 이미지의 특성 유형을 반영하는 골격선을 생성해 나가는 것이고, 직접 기법은 원래 이미지로부터 픽셀을 하나씩 제거해 나가므로써 세션화를 수행하는 것이다 [2]. 두 가지 기법 중에서 간접 기법이 직접 기법에 비해 우수한 성능을 갖는다[3].

이러한 간접 기법 중에서 신경망을 이용한 방법으

로는 이미지의 구조에 따라서 자신을 조직하는 특성을 갖는 Kohonen의 자기 구성 특징 지도(Self-Organized feature Map : SOM)를 이용한 세션화 연구가 진행되었고, 이 방법의 단점을 개선한 자기 구성 특징 그래프(Self-Organized feature Graph : SOG)가 제안되었고, SOG에서 학습 시간이 오래 걸리는 단점을 개선한 SOG* 세션화 기법이 제안되었다[4].

이 SOG* 세션화 기법은 우수한 세션화 결과와 빠른 속도를 가지지만 숫자와 문자 각 이미지에 대해 항상 고정된 수의 뉴런을 가짐으로 경쟁층의 뉴런이 부족하거나 불필요하게 많은 경우가 발생할 소지가 생긴다.

따라서 본 논문에서는 문자 이미지의 크기와 픽셀의 분포에 따라 경쟁층의 뉴런을 변화시킴으로써 기존 SOG와 동일한 우수한 세션화 결과를 내는 개선된 SOG* 세션화 기법을 제안하고 그 실험 결과를

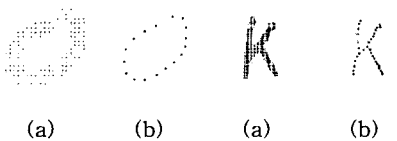
보인다.

본 논문의 2장에서는 관련 연구로 SOG* 세선화 기법을 설명하고, 3장에서는 개선된 적응적 SOG* 방법을 소개하며, 4장에서는 실험 및 실험 결과 분석을 하며, 마지막 5장에서는 결론을 맺는다.

2. 관련 연구(SOG*)

SOG* 세선화 방법은 세선화를 위한 학습 과정에서 경쟁층의 뉴런을 재정돈 하는 과정을 거쳐 승자 뉴런을 선택할 때 모든 뉴런을 한번에 모두 재정돈 하는 것이 아니라 재정돈 할 뉴런의 수를 단계별로 증가 시켜가며 학습시키는 점층적 학습 알고리즘과 학습 과정에서 뉴런들이 각 입력 패턴의 구조를 찾아내는 위치를 안정화되는 시점을 파악하여 이후 학습에서 기존 정렬을 그대로 사용하는 SOG의 변형 방법을 사용한 SOG 기반의 고속 세선화 알고리즘이다. 단계는 1단계부터 $\log_2(M-1)$ 단계까지이고, 각 단계별로 학습할 뉴런의 수는 $2^{\text{level}+1}$ 이다. 여기서 M은 맵의 크기를 의미한다[4].

다음 그림 1에서 SOG* 세선화의 결과를 보여주고 있다. 각 세선화의 결과는 원 이미지의 특성을 유지하는 우수한 결과를 나타내었고, 수행 속도 또한 기존의 방법보다 우수한 $O((\log M)^3)$ 을 나타내었다. 세선화 과정의 경쟁층 뉴런의 수는 숫자는 17개, 한글이나 영문자와 같은 문자는 33개로 고정한다. 그 이유는 SOG* 세선화는 각 단계별로 학습할 뉴런의 수가 $2^{\text{level}+1}$ 로 고정되어 있어서 3, 5, 9, 17, 33, 65, ...로 증가하기 때문에 실험적으로 단순한 숫자는 17개로 문자는 33개로 고정을 한다.



(a) 원 이미지 (b) SOG* 세선화 결과
(그림 1) SOG* 세선화 결과

3. 적응적 SOG*

기존 SOG* 세선화 기법은 숫자 17개, 문자는 33개라는 고정된 뉴런의 수를 가짐으로써 원 이미지가 크거나 픽셀의 분포가 조밀한 숫자의 경우는 골격선의 유지가 잘 되어 있으나 결과 이미지의 픽셀의 간

격이 조밀하지 못하고, 원 이미지가 작거나 픽셀 분포가 조밀하지 못한 문자의 경우는 앞의 예와 반대의 문제를 가지고 있다. 그림 2에서 이러한 예를 보여준다. 먼저 그림 2의 (a)에서는 크기가 크면서 픽셀의 분포가 조밀한 숫자의 경우 17개의 뉴런으로 세선화 한 결과를 보여주고 있으며, (b)에서는 반대의 경우인 33개의 뉴런으로 문자를 세선화 한 결과를 보여주고 있다.



(a) 원 이미지 (b) SOG* 세선화 한 결과
(그림 2) 특별한 경우의 SOG* 세선화 결과

위의 그림 2와 같은 특별한 경우의 이미지에 대해서도 좋은 세선화 결과를 얻기 위하여 기존의 SOG* 세선화 알고리즘을 다음 그림 3과 같이 수정하여 일반화시킨다.

```

/* 맵 크기 선택 : 설정된 기준 이하면 17, 초과면 33으로 초기화 */
for 단계 := 1 to log2(M-1) do
  begin
    /* 학습 할 뉴런 선택; 원 이미지의 픽셀 분포와 크기를 조사하여 설정한 기준에 맞게 학습할 뉴런을 추가 */
    node := 2level+1+추가노드;
    /* 가중치 부여 */
    give the weight of step;
    for repeat := 1 to 학습계수*단계 do
      begin
        for input := 1 to 단계 do
          begin
            /* 입력 패턴과 경쟁층 뉴런의 거리 정렬 */
            if 현재정렬=이전정렬
              then Ki := 이전정렬;
            else 입력과 노드들의 거리에 의한 정렬;
            /* 학습 */
            승자 뉴런 가중치 조절;
            이웃 뉴런 가중치 조절;
          end; // end for input
        end; // end for repeat
      end; // end for level
    end;
  end;
  (그림 3) 개선된 SOG* 학습 알고리즘
  
```

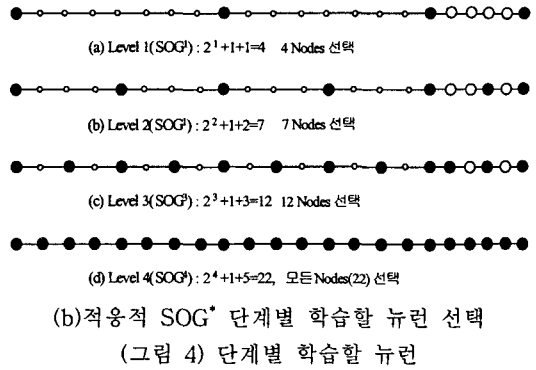
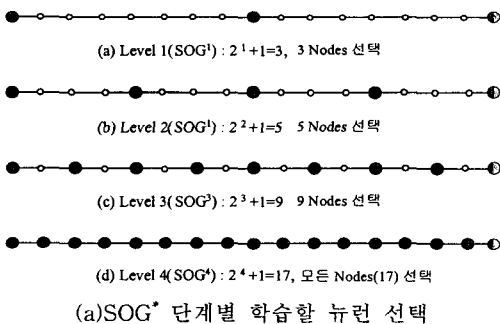
그림 3의 알고리즘은 기존 SOG* 알고리즘을 기반으로, 먼저 맵의 크기와 추가할 뉴런의 수를 결정하는데, 이때는 원 이미지의 픽셀의 분포도와 크기를

기준으로 17 또는 33으로 결정한다. 픽셀 분포만으로 볼 때는 450픽셀을 초과하여 존재하는 경우는 33개의 맵 크기가 적당하고 그 이하는 17개의 맵이 적당하다. 그러나 픽셀 분포만으로 결정하게 되면 이미지의 크기가 작으면서 조밀하게 분포된 경우 또는 그 반대의 경우에 적용적이지 못하기 때문에 이미지의 크기를 같이 고려하게 되는데, 원 이미지의 가로×세로의 크기가 6000미만인 경우는 17 맵으로 선택하고 그 이상이면 33으로 선택한다. 맵의 크기가 결정이 되면 단계별로 학습을 시키는데 기존 SOG* 알고리즘에서는 고정된 $2^{level}+1$ 개의 뉴런만을 학습에 참여시키지만 개선된 방법에서는 $2^{level}+1$ 에다가 추가할 뉴런의 수를 단계별로 추가하여 학습을 시킨다. 만일 4개의 뉴런이 더 추가되어야 한다면 각 단계별로 1개씩을 추가하여 $2^{level}+1+1$ 개의 뉴런이 학습에 참여하게 되는 것이다. 나머지 부분의 알고리즘은 기존 SOG* 기법과 모두 동일하다.

그림 4에서는 기존의 SOG* 방법에서 각 단계별로 학습할 뉴런을 선택하는 과정과 적응적 SOG* 방법에서 각 단계별로 학습할 뉴런을 선택하는 과정을 나타내고 있다. 그림 4의 (a)에서는 SOG*에서 17개의 고정된 뉴런을 1단계부터 4단계($\log_2(17-1)=4$)까지 선택하는 과정을 설명하고 있으며, (b)에서는 적응적 SOG*에서 22개의 뉴런이 단계별로 선택되어지는 과정을 설명하고 있다. 적응적 SOG* 방법에서는 단계는 SOG*와 동일하면서 추가되는 뉴런들만 각 단계별로 배치하게 된다.

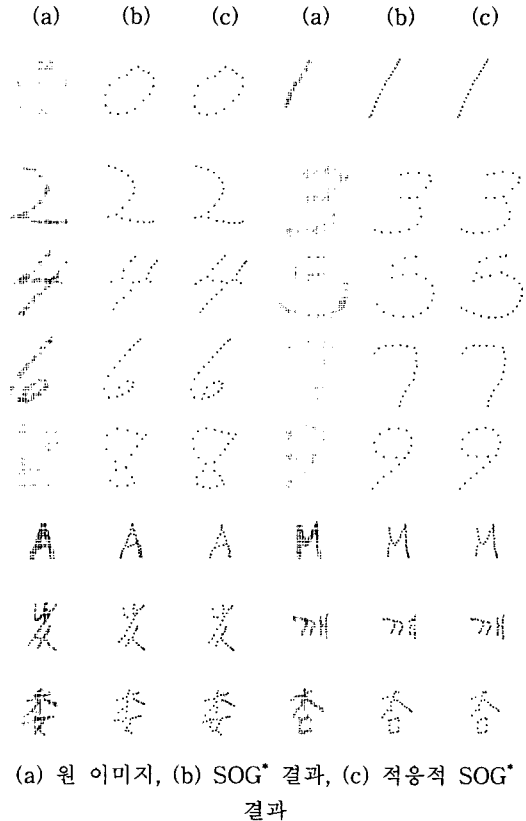
4. 실험 및 실험 결과

실험은 P-III 1GHz와 Windows 2000 환경에서 VC++ 6.0으로 구현하여 실험하였고, 사용된 이미지들은 한글, 영문자와 숫자들이다. 모든 이미지들은



국내의 여러 대학과 연구소의 샘플 데이터와 임의의 필기자가 필기한 데이터를 사용하였다.

그림 5에서는 기존 SOG* 세션화 기법의 결과와 적응적 SOG* 세션화 기법을 비교하여 나타내었다.



(그림 5) SOG*와 개선된 SOG*의 세션화 결과

그림 5에서 숫자 0, 1, 7은 경쟁층 뉴런의 추가나 삭제가 필요 없이 17개의 뉴런 그대로 사용하여도

무방한 일반적인 경우를 보이고 있으며, 나머지 숫자들은 17개의 뉴런으로 나타내어도 무방하지만 세션화 결과의 질을 더 높이기 위하여 경쟁층의 뉴런이 몇 개 추가된 경우이다. 영문자 A, M과 한글 '났', 한문 촛자 같은 경우는 33개의 뉴런으로는 오히려 결과의 질이 좋지 않아 몇 개의 뉴런을 뺀 경우이고 한글 '깨', 한문 촛자는 33개의 뉴런에 조금 더 추가되어 세션화 결과의 질을 높인 예를 보여주고 있다.

고정된 수의 경쟁층 뉴런으로도 SOG* 세션화 기법은 양질의 세션화 결과와 빠른 수행 속도를 가지지만 그림 5의 영문자 A와 같은 특수한 경우에는 기존 방법으로는 적용적이지 못하다. 본 실험 결과 기존 SOG*의 빠른 수행 속도와 우수한 세션화 결과를 가지면서 특수한 경우의 이미지에 대해서도 잘 적응됨을 보았다.

5. 결론

기존의 SOG* 세션화 기법은 개선된 SOG와 수행 속도를 빠르게 하는 점증적 정돈 기법을 이용한 고속 세션화 기법이다. SOG*의 세션화 결과는 우수하지만 고정된 경쟁층 뉴런의 수를 가짐으로써 본 논문에서 특수하다고 말하고 있는 픽셀의 범위와 이미지의 크기에 잘 적응을 하지 못한다. 본 논문에서는 이러한 점을 개선하기 위하여 원 이미지의 픽셀의 범위와 이미지의 크기를 파악하여 동적으로 경쟁층의 뉴런을 생성하여 기존의 방법보다 우수한 세션화 결과를 내었다.

문자 인식의 진척리로 사용되는 세션화에서 본 논문의 세션화를 사용한다면 이미지의 특성을 유지하는 우수한 세션화 결과와 빠른 속도를 기대할 수 있다.

참 고 문 헌

[1] Holt. C. M., A. Stewart, "A parallel thinning algorithm with fine grain subtasking," Parallel Comput. 10, pp.329-334, 1989.
 [2] L. Lam., S. Lee and C. Y. Suen, "Thinning methodologics a comprehensive survey," IEEE Trans. Pattern Analysis and Machine Intelligence, 14(9), 869-885, 1992.
 [3] P. Ahmed, "A neural network based dedicated

thinning method," Pattern Recognition, 16, pp.585-590, 1995.

[4] 이찬희, 정순호, "개선된 SOG 기반 고속 세션화 알고리즘(SOG*)", 한국정보처리학회논문지, 제 8-B권, 제 6호, pp.651-656, 2001.