

점증적 정돈기법의 SOG를 이용한 개선된 세선화 알고리즘

정선정⁰ 이찬희⁰ 정순호⁰
부경대학교 전자계산학과⁰ 부경대학교 전자컴퓨터정보통신 공학부
sijung@mail1.pknu.ac.kr shjung@pine.pknu.ac.kr

Improved Thinning Algorithm using SOG with Incremental Ordering method

Sun-Jung Jung⁰ Chan-Hee Lee⁰ Soon-Ho Jung⁰
Dept. of Computer Science, Pukyong National University⁰
Faculty of Electronics, Computer, Telecommunication Engineering,
Pukyong National University

요 약

세선화 알고리즘의 간접 기법으로 제시된 자기구성 특징 그래프(Self-Organizing feature Graph : SOG) 기법은 안정된 세선화 결과를 가지는 장점이 있으나 학습 알고리즘에서 전체 노드를 재정돈하는 과정이 내포되어 있다. 본 논문에서는 학습 알고리즘의 재정돈 과정을 대신하는 점증적 정돈기법을 제안하고 이 기법을 세선화 알고리즘에 결합하여 실험하고 분석하였다. 제안된 알고리즘은 기존의 SOG를 적용한 결과와 같은 우수한 세선화 결과를 얻으며 학습시간은 $O((\log M)^3)$ 인 복잡도를 가진다.

1. 서론

컴퓨터 인식 기술에서 사용되는 세선화는 이미지에 서 골격을 찾는 기법으로 크게 직접 기법과 간접 기법으로 나눌 수 있다. 직접 기법은 픽셀들을 직접 제거하여 골격선을 찾아나가는 기법이고 간접 기법은 이미지의 구조를 파악하여 골격선을 생성하는 기법이다 [1][2]. 두 기법 중 간접 기법이 직접 기법에 비해 더 나은 성능을 가진다[3]. P. Ahmed은 간접 기법으로서 Self-Organizing feature Map(SOM)을 변형하여 학습결과가 우수한 Self-Organizing feature Graph(SOG)를 제안하였다[3][4].

그러나 SOG를 이용한 세선화 알고리즘은 세선화 결과에서는 우수한 반면 학습 단계에서 뉴런들의 재정돈 과정 때문에 학습시간이 오래 걸리게 된다. 본 논문에서는 이런 단점을 보완하기 위해 점증적 정돈기법을 적용하여 우수한 세선화 결과를 가지면서 학습시간이 감소된 개선된 SOG 세선화 알고리즘을 제안하였다.

이 논문의 2장에서 SOG의 학습 방법과 SOM에 점증적 정돈기법을 적용한 L* 알고리즘을 소개하고, 3장에서 제안하는 알고리즘을 기술하였고 4장에서는 제안한 알고리즘을 평가하기 위해 숫자, 알파벳 대소문자, 한글 등의 이미지를 세선화하는 실험을 통해 그 결과를 분석

하고, 두 알고리즘의 시간 복잡도를 계산하여 분석하며 5장에서 결론을 언급한다.

2. 관련연구

2.1 자기구성 특징 그래프(Self-Organizing feature Graph)

SOG는 SOM의 변형으로서 자가학습 과정에서 뉴런의 재정돈에 의해 인접성이 변화하여 학습목표 영역으로의 매핑 능력이 향상되어 우수한 학습결과를 가지는 신경망 모형이다[3][5]. 그 학습 과정은 SOM과 유사하며, 경쟁과 가중치 수정의 두 단계로 크게 나눌 수 있다 [4][6][7]. 경쟁 단계에서는 입력과의 거리가 가장 작은 뉴런을 찾는 과정이며, 가중치 수정 단계에서는 식 (1)과 같이 뉴런의 가중치들을 수정한다.

$$w_i(t+1) = w_i(t) + \zeta(t, K_i) \times (X - w_i(t)) \quad ,$$
$$\zeta(t, K_i) = \exp\left[-\frac{\ln(K_i)}{a(t)}\right]^2 \quad \text{식(1)}$$

여기서 X 는 입력 벡터이고, K_i 는 정돈된 뉴런들의 초기 인덱스값이며, $a(t)$ 는 학습률이다.

2.2. L* 알고리즘

L* 알고리즘은 SOM에서 경쟁 단계에서 입력과의 거리가 가장 작은 뉴런인 승리자를 찾는데 걸리는 시간을 줄여 전체적인 학습 시간을 단축시키는 알고리즘이다[8]. 학습을 할 때 단계별로 뉴런들의 수를 점차 늘려가면서 승리자를 찾기 때문에 기존의 SOM에서 항상 모든 뉴런을 대상으로 승리자를 구하는 방법에 비해 계산시간이 단축된다.

3. 점증적 정돈기법을 이용한 세선화 알고리즘

3.1. 점증적 정돈기법

기존의 SOG를 이용한 세선화 알고리즘은 학습을 할 때마다 뉴런들을 재정돈하기 때문에 학습시간이 오래 걸리게 된다. 이를 개선하기 위해서 본 논문에서는 점증적 정돈기법을 적용한 세선화 알고리즘을 제안한다. 점증적 정돈기법이란 세선화를 위해 재정돈 과정을 하되 한꺼번에 전 뉴런을 재정돈 하는 것이 아니라 재정돈할 뉴런의 수를 단계별로 증가시켜가면서 세선화하는 방법이다. 처음 단계에서는 일부의 뉴런들만을 학습시키고, 다음 단계에서는 이전 단계에서 학습한 뉴런에 학습이 되지 않은 뉴런들 중 일부를 추가시켜 학습을 진행시킨다. 이때 단계별로 추가되는 뉴런은 이전 단계의 뉴런들을 이용한다.

3.2. 점증적 정돈기법을 이용한 세선화 알고리즘

점증적 정돈기법을 적용한 세선화 알고리즘을 구현하기 위해서는 먼저 전 뉴런을 몇 단계로 나누어 학습해야 하는지를 결정해야 한다. 단계는 1부터 $\log_2(M-1)$ 까지로 나누어지며, 여기서 M은 뉴런의 총 개수이다. 그런 다음 각 단계별로 학습에 참여할 뉴런의 수를 결정해야 한다. 단계에 따른 뉴런의 수는 $2^{level}+1$ 이고 이때 level은 각 학습 단계를 의미한다.

예를 들어 17개의 뉴런을 학습하기 위해서는 한꺼번에 17개의 뉴런을 모두 학습하는 것이 아니라 단계를 4 단계로 나누고 각 단계별로 뉴런의 수를 각각 3, 5, 9, 17로 증가시켜 가면서 학습을 한다.

다음으로 뉴런과 입력과의 거리를 계산하고 이 값들을 다시 정돈한다. 마지막으로 이 정돈한 값들의 순서를 반영하여 가중치를 수정한다. 가중치를 수정하는 방법은 기존의 SOG와 같이 식(1)을 이용한다.

한 단계가 끝나면 다음 단계에 학습되어야 할 뉴런들을 추가해야 한다. 이때 추가하는 뉴런들은 전 단계에서 학습된 뉴런들을 이용한다. [그림 1]은 점증적 정돈기법을 이용한 세선화 알고리즘을 간략히 기술한 것이다.

이 알고리즘을 8의 숫자에 적용한 예는 [그림 2]와 같고 이때 M은 17이다.



원이미지 level:1 level:2 level:3 level:4

[그림 2] 단계별 세선화 결과

```

procedure 개선된 SOG()
for level=1 to log(M-1) by 1
    lev_max_node<= 2level+1
    repeat level당 학습횟수
        sorting(lev_max_node);
        alpha 설정
        // weight update
        for x<-0 to x<-lev_max_node by 1
            wi(t+1)= wi(t)+{(t, Ki)×(X- wi(t))}
        end repeat
    다음 level에서 사용될 weight 부여
end for level
end procedure
    
```

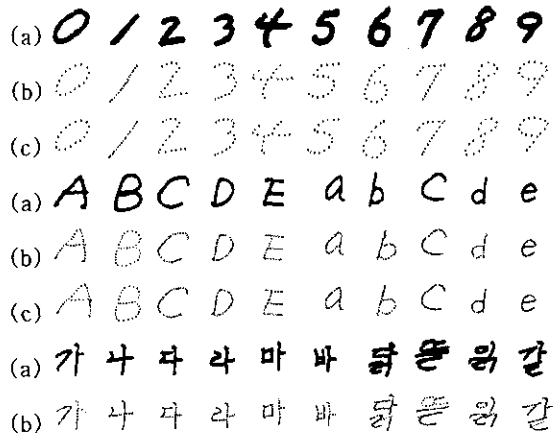
[그림 1] 점증적 정돈기법 알고리즘

4. 제안한 세선화 알고리즘의 실험 및 분석

4.1. 실험

본 논문에서는 기존의 SOG를 이용한 세선화 알고리즘과 점증적 정돈기법을 적용한 세선화 알고리즘을 비교하기 위해 몇 가지 실험을 수행한다. 이 실험은 셀러론 550 MHz의 IBM PC에서 이루어지고 프로그래밍 언어로는 Visual C++을 사용하여 구현되었다.

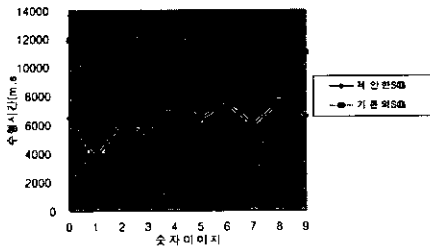
먼저 두 알고리즘의 세선화 결과를 비교하기 위해 각 알고리즘에 숫자, 알파벳 대소문자, 한글 이미지를 적용하여 실험한다. [그림 3]는 각 알고리즘에 대한 숫자, 알파벳 대소문자, 한글 이미지의 세선화 결과이다. 숫자 이미지에서는 단계별 학습횟수가 2000, 뉴런의 수는 17이고, 알파벳 대소문자와 한글 이미지에서는 단계별 학습횟수가 3000, 뉴런의 수는 33으로 고정시켜 실험한다. (a)는 세선화하기 전의 원 이미지를 보여주고 (b)는 기존의 알고리즘의 세선화 결과이고 (c)는 본 논문에서 제안한 알고리즘의 세선화 결과이다.



(c) 가 나 다 각각 바둑판의 결과

[그림 3] (a) 원 이미지 (b) 기존의 SOG
(c) 점증적 정돈기법을 이용한 SOG

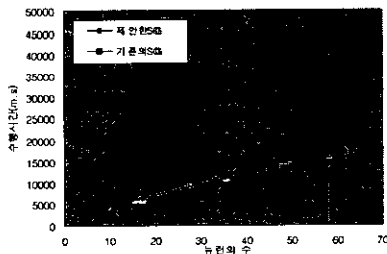
[그림 3]에서와 같이 두 알고리즘의 세선화 결과가 모두 우수하다는 것을 알 수 있다. 다음으로 두 알고리즘의 수행시간을 비교하기 위해 실험한 데이터 중 숫자 이미지에 대한 수행시간을 비교하고 그 결과는 [그림 4]과 같다.



[그림 4] 각 알고리즘의 수행 시간

이 그래프에서도 알 수 있듯이 제안한 알고리즘이 기존의 알고리즘보다 평균적으로 약 1.67배 빠르다는 것을 알 수 있다.

다음으로 뉴런의 수에 따른 두 알고리즘의 수행 시간을 비교해 보면 [그림 5]와 같다.



[그림 5] 뉴런의 수에 따른 각 알고리즘의 수행시간

뉴런의 수에 따른 결과 역시 제안한 알고리즘이 기존의 알고리즘보다 속도 면에서 우월하다는 점이 밝혀졌고, 또한 뉴런의 수가 증가할수록 제안한 알고리즘의 수행속도가 기존의 알고리즘보다 비례적으로 빨라진다는 것을 알 수 있다.

4.2. 시간 복잡도 계산

기존의 알고리즘과 제안한 알고리즘을 수학적으로 분석하기 위해 두 알고리즘의 시간 복잡도를 계산한다. 기존의 알고리즘은 항상 모든 뉴런을 전부 학습에 참여시키기 때문에 시간 복잡도는 $O(M \log M)$ 인 반면, 제안한 알고리즘은 단계에 따라 학습할 뉴런의 수가 점차 증가하면서 학습하므로 총 학습시간은 다음과 같다.

$$C \cdot \sum_{level=1}^{\log(M-1)} (2^{level} + 1) \cdot \log(2^{level} + 1)$$

$$= C \cdot (3 \cdot \log 3 + 5 \cdot \log 5 + 9 \cdot \log 9 + \dots + M \cdot \log M) \\ \leq C \cdot \left(\sum_{k=1}^{\log(M-1)} k \cdot \log k \right) \leq C \cdot \left(\sum_{k=1}^{\log(M-1)} k^2 \right) \\ = O(\log(M-1)^3)$$

여기서 C 는 level당 학습횟수이다. 따라서 시간 복잡도는 $O((\log M)^3)$ 이고 이것은 제안한 알고리즘이 실험의 결과와 마찬가지로 기존의 SOG를 적용한 세선화 알고리즘보다 우수하다는 것을 증명한다.

5. 결론

기존의 SOG를 이용한 세선화 알고리즘은 세선화 결과에서는 우수하지만 학습 단계에서의 뉴런들의 재정돈 과정 때문에 학습시간이 오래 걸린다는 단점을 가지고 있다. 본 논문에서는 이런 단점을 개선하기 위해 점증적 정돈기법을 적용한 세선화 알고리즘이 제안되었고, 기존의 알고리즘과 제안한 알고리즘을 비교한 결과 제안한 알고리즘의 세선화 결과가 우수할 뿐만 아니라 속도 또한 약 1.67배 빠르다는 것이 확인되었으며, 시간 복잡도 계산에서도 제안한 알고리즘은 $O((\log M)^3)$ 으로서 제안한 알고리즘의 수행속도가 기존의 알고리즘보다 우수하다는 것이 증명되었다.

그러므로 세선화 알고리즘의 적용에서 본 논문에서 제안한 점증적 정돈기법의 SOG를 이용한 세선화 알고리즘을 채택하면 좀 더 빠르며 좋은 세선화 결과를 얻을 것이다.

6. 참고 문헌

- [1] Holt, C. M., A. Stewart, "A parallel thinning algorithm with fine grain subtasking", *Parallel Comput.* 10, pp. 329-334, 1989.
- [2] S. Mahmood, L. S. Abuliba, R. G. Green, "Skeletonization of arabic characters using clustering based skeletonization algorithm(CBSA)", *Pattern Recognition*, 24(5), pp. 453-464.
- [3] P. Ahmed, "A neural network based dedicated thinning method", *Pattern Recognition*, 16, pp. 585-590, 1995.
- [4] T. Kohonen, "The self-organizing map", *Proc. IEEE* 78(9), pp. 146-180, 1990.
- [5] 강효진, 정순호, "개선된 SOG를 이용한 세선화 알고리즘", *한국정보처리학회 춘계학술발표 논문집*, 1999.
- [6] S. Haykin, "Neural Networks", Prentice Hall, 1994.
- [7] T. Kohonen, "Self-organized formation of topologically correct feature maps", *Biol. Cybern.*, vol. 43, pp. 59-69, 1982.
- [8] Y. P. Jun, H. Yoon, J. W. Cho, "L* Learning: A Fast Self-Organizing Feature Map Learning Algorithm Based on Incremental Ordering", *IEICE Trans. Inf. & Sys*, vol. E76-D, No. 6, June. 1993.