

# 영상분할을 위한 밀도추정 바탕의 Fuzzy C-means 알고리즘

## A Density Estimation based Fuzzy C-means Algorithm for Image Segmentation

고정원, 최병인, 이정훈

Jeong-Won Ko, Byung-In Choi, and Frank Chung-hoon Rhee

한양대학교 전자 컴퓨터 제어 계측 공학과

### 요 약

Fuzzy C-Means (FCM) 알고리즘은 probabilistic 멤버십을 사용하는 클러스터링 방법으로서 널리 쓰이고 있다. 하지만 이 방법은 노이즈에 대하여 민감한 성질을 가진다는 단점이 있다. 따라서 본 논문에서는 이러한 노이즈에 민감한 성질을 보완하기 위해서 데이터의 밀도추정을 이용하여 새로운 FCM 알고리즘을 제안한다. 본 논문에서 제안된 알고리즘은 FCM과 비슷한 성능의 클러스터링 수행이 가능하며, 노이즈가 포함된 데이터에서는 FCM보다 더 나은 성능을 보여준다.

키워드 : Fuzzy c-means, 노이즈, 밀도추정, Parzen-window

### Abstract

The Fuzzy C-means (FCM) algorithm is a widely used clustering method that incorporates probabilistic memberships. Due to these memberships, it can be sensitive to noise data. In this paper, we propose a new fuzzy C-means clustering algorithm by incorporating the Parzen Window method to include density information of the data. Several experimental results show that our proposed density-based FCM algorithm outperforms conventional FCM especially for data with noise and it is not sensitive to initial cluster centers.

Key Words : Fuzzy c-means, Noise, Density Estimation, Parzen-window

## 1. 서 론

Fuzzy C-Means (FCM)은 유클리디안 공간상에서 거리를 이용하여 퍼지 멤버십을 할당함으로써 클러스터링을 수행한다[1]. 하지만 노이즈가 섞여 있는 데이터를 클러스터링 하는 경우는 결과가 좋지 않을 수 있다. 이러한 이유는 패턴과 클러스터 센터간 거리에 따라 클러스터들간의 상대적인 멤버십을 할당하기 때문이다. 이러한 단점을 보완하기 위하여 PCM이나 CFCM등의 노이즈에 강한 클러스터링 방법들이 제안되었다[2][3]. FCM이 패턴과 클러스터 센터간의 상대적인(relative) 멤버십을 사용했다면 PCM은 실제적인(actual) 멤버십을 사용한 클러스터링 방법이다. CFCM은 패턴과 클러스터들의 거리에 따라 멤버십에 가중치를 주는 방법이고, PCM, CFCM 두 방법 모두 노이즈에 강한 특성이 있다. 하지만 PCM과 같은 경우 결과값이 초기센터에 많은 영향을 받고 또한 클러스터 센터들이 한곳으로 겹치는 현상이 일어나기도 한다[4]. CFCM 역시 초기 센터값에 따라 결과가 안 좋게 나타날수가 있다. 따라서 본 논문에서는 패턴의 밀도추정을 이용한 Density Estimation Based FCM (DBFCM) 클

러스터링 방법을 제안하고 있다. 제안하는 방법은 먼저 Parzen-window 방법을 사용하여 패턴의 밀도 추정값을 구한후 그 값이 작으면 그것은 다른 데이터들과 멀리 떨어져 있는 노이즈 패턴으로 판단하여 상대적으로 작은 퍼지 멤버십을 할당하고, 밀도 추정값이 높으면 좋은 패턴이라 판단하여 높은 멤버십을 할당한다. 따라서 노이즈가 포함된 데이터에 대해서는 FCM보다 더 뛰어난 클러스터링 결과를 나타낼 수 있다. 본 논문은 다음과 같이 구성된다. 두 번째 절에서는 다른 클러스터링 알고리즘들에 대하여 소개하고 세 번째 절에서는 여기서 제안하는 DBFCM에 대하여 설명한다. 네 번째 절에서는 제안하는 알고리즘과 다른 알고리즘의 실험결과를 비교하고, 마지막으로 결론을 맺겠다.

## 2. FCM 와 CFCM 알고리즘

### 2-1. Fuzzy C-Means (FCM)

일반적으로 잘 알려진 FCM은 다음과 같은 목적함수를 최소화 하는 것을 기초로 한다.

$$J_{FCM} = \sum_{k=1}^N \sum_{i=1}^C (u_{ik})^m \| \mathbf{x}_k - \mathbf{v}_i \|^2 \quad (1)$$

접수일자 : 2006년 10월 21일

완료일자 : 2007년 1월 30일

감사의 글 : 본 연구는 한국과학기술원 영상정보특화연구센터를 통한 국방과학연구소의 연구비 지원으로 수행되었습니다.

$$\text{subject to } \sum_{i=1}^{i=C} u_{ik} = 1 \quad (2)$$

여기서  $k$ 는 패턴의 개수를 나타내고  $i$ 는 클러스터 센터의 개수를 나타낸다. 그리고  $u_{ik}$ 는 패턴  $k$ 가 클러스터  $i$ 에 소속되는 멤버십 값을 나타내고,  $m$ 은 퍼지화 상수로서  $m \in (1, \infty)$  인 조건을 만족한다. 위의 목적함수가 최소값을 갖도록 해주는 소속도와 클러스터 센터는 다음의 식에 의해서 구할 수 있다[5].

$$u_{ik} = \left( \sum_{j=1}^{j=C} \left( \frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}} \right)^{-1} \quad (3)$$

$$\text{and } \mathbf{v}_i = \frac{\sum_{k=1}^{k=N} (u_{ik}^m \mathbf{x}_k)}{\sum_{k=1}^{k=N} (u_{ik}^m)} \quad (4)$$

하지만 FCM은 노이즈에 대해 민감하다는 단점이 있다. 왜냐하면 FCM의 특성상 노이즈 패턴에 대해서도 높은 멤버십 할당을 할 수 있기 때문이다.

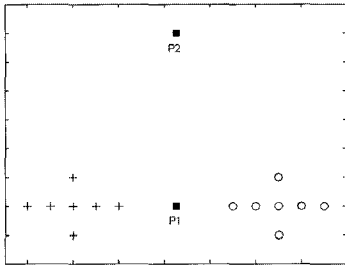


그림 1. FCM에서의 멤버십  
Fig. 1. Membership for FCM

그림과 같이 양쪽에 두개의 클러스터가 있을 경우 패턴 P2의 경우 패턴P1에 비해 각 클러스터들과 멀리 떨어져 있기 때문에 각 클러스터에 대한 멤버십은 P1보다 작아야 한다. 하지만 FCM의 특성상 패턴 P1, P2 모두 각 클러스터에 대하여 0.5 라는 멤버십을 갖는다. 따라서 P2와 같이 노이즈에 가까운 패턴도 높은 멤버십을 가질수 있고 이러한 부분이 클러스터링 결과에 나쁜 영향을 줄 수 있다.

### 2-2. Credibility Fuzzy C-Means (CFCM)

CFCM은 노이즈에 민감한 FCM의 약점을 보완할수 있는 방법이다. 제안된 알고리즘에서는 credibility 라는 새로운 변수를 사용하여, 노이즈에 민감한 기존의 FCM의문제점을 극복하고 있다[3].

패턴  $\mathbf{x}_k$ 의 credibility  $\Psi_k$ 는 다음과 같다.

$$\Psi_k = 1 - \frac{(1-\theta)\alpha_k}{\max_{j=1..n}(\alpha_j)}, \quad 0 \leq \theta \leq 1 \quad (5)$$

where,  $\alpha_k = \min_{i=1..c}(d_{ik}) \quad (6)$

위의 식에서 볼수 있듯이 데이터군에 가까우면 높은 credibility를 가지고, 멀리 떨어져 있으면 낮은 credibility를

가진다. 일반적으로 노이즈 패턴은 다른 패턴들과 멀리 떨어져 있기 때문이다. 즉 credibility 라는 변수를 사용하여 정상적인 패턴과 노이즈 패턴을 분류를 할 수가 있다. CFCM의 목적함수는 다음과 같다.

$$J_{\text{CFCM}} = \sum_{k=1}^{k=N} \sum_{i=1}^{i=C} (u_{ik})^m \|\mathbf{x}_k - \mathbf{v}_i\|^2, \quad (7)$$

$$\text{subject to } \sum_{i=1}^{i=C} u_{ik} = \Psi_k \quad (8)$$

목적함수를 최소화 시켜주는 패턴과 클러스터간의 멤버십은 다음 식에 의해 구할 수 있다.

$$u_{ik} = \frac{\Psi_k}{\sum_{j=1}^{j=c} \left( \frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}}} \quad (9)$$

클러스터 센터에 대한 업데이트 식은 FCM과 거의 같기 때문에, FCM과 같이 식(4)를 사용하여 클러스터 센터를 구한다. 제안된 CFCM에서는 FCM과는 달리 클러스터와 멀리 떨어진 노이즈 패턴에 대해서는 낮은 멤버십 가중치를 준다. 따라서 클러스터들에 멀리 떨어져 있는 노이즈 패턴들은 클러스터링 하는데 있어서 영향을 덜 미치게 되므로 노이즈에 강한 클러스터링 결과를 얻을수 있다.

### 3. Density Based FCM (DBFCM)

CFCM에서는 패턴과 데이터군의 거리를 계산하여 credibility 라는 변수의 식에 사용함으로써, 멤버십 계산에 사용했다. 본 논문에서 제안하는 방법에서는 각 패턴의 밀도를 이용한다. 일반적으로 정상적인 패턴은 그 주위에 다른 패턴들이 많이 분포해있지만, 노이즈 패턴의 경우는 그 주위에 다른 패턴들이 없고 홀로 떨어져 있다. 따라서 밀도가 높은 패턴에 대해서는 높은 멤버십을 할당하고, 밀도가 낮은 패턴에 대해서는 상대적으로 낮은 멤버십을 할당한다. 따라서 정상적인 패턴은 높은 멤버십을 가지게 되고 노이즈 패턴의 경우는 낮은 멤버십을 가지게 되어 노이즈에 강한 결과를 얻을 수 있다. 그리고 패턴의 밀도추정 방법으로는 Parzen-window 를 사용한다.

#### 3-1. Parzen-window method

패턴  $\mathbf{x}_k$ 을 포함하는 영역  $R$ 이 있을 때, 고정된 크기의 영역  $R$ 안에 존재하는 패턴들의 개수를 계산하여 밀도 추정값  $p_k(\mathbf{x})$ 를 얻을수있다. 이때 다음의 식을 이용하여  $p_k(\mathbf{x})$ 를 구한다[6].

$$p_k(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{V} \varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) \quad (10)$$

- $h$  : 다차원 영역  $R$ 의 선분길이
- $V = h^d$  : 다차원 영역  $R$ 의 부피
- $n$ : 데이터의 개수

이때 window 함수  $\varphi(u)$ 로는 일반적으로 많이 쓰는 가우시안 함수를 사용한다.

$$\varphi(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2} \quad (11)$$

$p_k(x)$  값은 smoothing parameter인  $h$ 에 영향을 받는다.  $h$ 가 커지면 데이터의 밀도분포가 완만해지고,  $h$ 가 작아지면 밀도분포가 날카로워진다. 따라서 적절한  $h$  값을 선택하는 것이 중요하다.  $h$  값은 데이터의 표준편차와 연관이 있기 때문에, 다음과 같이 적절한  $h$  값을 구한다[7].

$$h = \sigma \left( \frac{4}{(m+2)n} \right)^{\frac{1}{m+4}} \quad (12)$$

여기서  $n$ 은 데이터의 차수,  $m$ 은 벡터공간의 차수,  $\sigma$ 는 전체 데이터의 표준편차를 나타낸다.

### 3-2. Proposed DBFCM

식(10),(11),(12)를 사용하여  $p_k(x)$ 를 구한다. 이때 DBFCM은 다음의 목적함수를 가진다.

$$J_{DBFCM} = \sum_{k=1}^N \sum_{i=1}^C u_{ik}^m \| \mathbf{x}_k - \mathbf{v}_i \|^2, \quad (13)$$

$$\text{subject to } \sum_{i=1}^C u_{ik} = p_k \quad (14)$$

목적함수를 최소화 하는 멤버쉽은 다음의 식으로 구할 수 있다.

$$u_{ik} = \frac{p_k}{\sum_{j=1}^C \left( \frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}}} \quad (15)$$

CFCM과 마찬가지로 클러스터 센터에 대한 업데이트 식은 FCM과 거의 같기 때문에, 역시 식(4)를 사용하여 클러스터 센터를 구한다. DBFCM의 전반적인 알고리즘은 다음과 같다.

### Density Based Fuzzy C-Means Clustering

#### Step1. Initialization

Initialize random initial center  $\mathbf{v}$  and  $m$

Compute the initial membership  $u_{ik}$  using (3)

#### Step2. Minimization of the objective function

Do:

Update  $\mathbf{v}^t$  using (4)

Compute  $u_{ik}^t$  using (15)

Until:  $u_{ik}^t - u_{ik}^{t+1} < \epsilon$

## 4. 시뮬레이션 및 결과 고찰

본 절에서는 본 논문의 타당성을 보이기 위해 FCM, CFCM, 그리고 제안하는 DBFCM 알고리즘으로 데이터들을 실험 후, 그 결과를 비교해 보겠다. 모든 실험에 fuzziness parameter인  $m$  값은 2를 사용하였다.

### 4-1. Spherical Data

이번 실험에서 구형의 두 개의 클러스터로 나뉘는 데이터를 클러스터링 하였다. 클러스터당 27개씩 총 54개의 패턴으로 이루어진 데이터이다. 그 결과 FCM, CFCM, DBFCM 방법 모두 좋은 결과를 나타내고 있다.

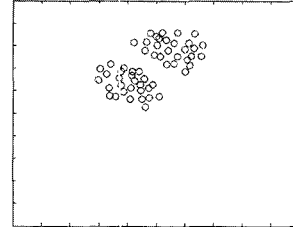


그림 2. 노이즈가 없는 구형의 데이터

Fig. 2. Spherical Data (FCM, CFCM, DBFCM)

### 4-2. Spherical Data with Noises

그럼 이번에는 앞의 데이터에 14개의 노이즈를 포함시켰을 경우 결과를 보자. 노이즈에 민감한 FCM은 역시 아래쪽에 위치한 노이즈들에 영향을 받아 두 클러스터가 만나는 지점에서 결과가 좋지 않게 나왔다. 하지만 CFCM과 DBFCM의 경우에는 노이즈에 영향을 받지 않고 좋은 클러스터링 결과를 보여준다.

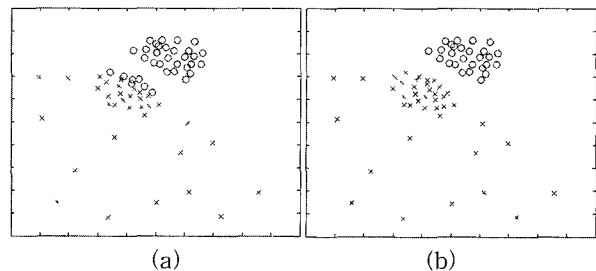
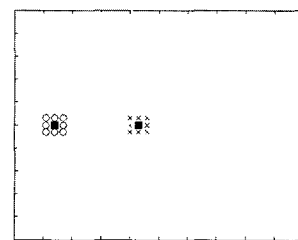


그림 3. 노이즈가 포함된 구형의 데이터

Fig. 3. Spherical Data with Noises (a) FCM (b) CFCM, DBFCM

### 4-3. Same volume "squares" Data

각각 9개의 개수를 가지는 같은 부피의 두 개의 클러스터로 나뉘어지는 데이터를 FCM, CFCM, DBFCM 방법으로 클러스터링을 하여 결과를 비교해본다. 실험 후 그림과 같은 결과를 볼 수 있는데 세가지 방법 모두 올바른 클러스터 센터를 얻을 수 있었다.



■ : 최종 클러스터 센터

그림 4. 같은 부피를 가진 사각형 모양의 데이터

Fig. 4. Same Volume "Squares" Data (FCM, CFCM, DBFCM)

4-4. Different volume "squares" Data

같은 개수, 다른 부피를 가진 사각형 모양의 데이터에 대한 클러스터링 결과는 다음과 같다. FCM과 CFCM에서는 클러스터 센터가 오른쪽으로 치우치는 현상이 나타난다. 때문에 큰 사각형의 왼쪽에 위치한 패턴 세개는 작은 사각형 쪽으로 잘못 클러스터링 된 것을 볼 수 있다. 하지만 DEFCM의 경우에는 가장 올바른 클러스터 센터를 찾을 수 있었고 때문에 모든 패턴들을 올바르게 분류할 수 있었다.

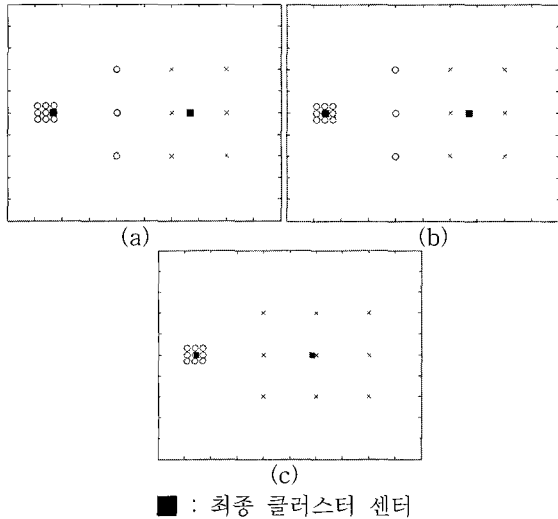
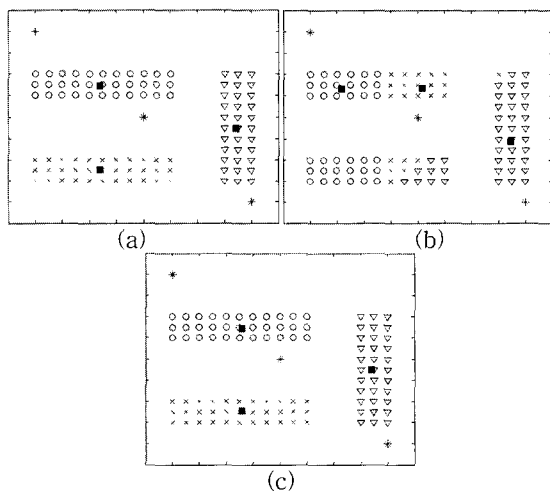


그림 5. 다른 부피를 가진 사각형 모양의 데이터  
Fig. 5. Different Volume "Squares" Data (a) FCM (b) CFCM (c) DEFCM

4-5. 3-Lines Data

이번엔 라인당 33개씩 총 99개의 패턴을 가진 라인 데이터의 결과를 보겠다. 실험 결과 FCM과 DEFCM은 초기 클러스터의 위치에 관계없이 올바른 클러스터링을 수행 하지만 CFCM의 경우 초기 클러스터 센터를 잘못 주었을 경우 좋지 않은 결과를 보여준다. 이는 CFCM에서 불안정한 초기센터가 credibility에 영향을 주면서 수행이 되기 때문이다.



\* : 초기 클러스터 센터 ■ : 최종 클러스터 센터  
그림 6. 세 개의 선 모양의 데이터에 대한 클러스터링 결과  
Fig. 6. Clustering Results for 3-Lines Data (a) FCM (b) CFCM (c) DEFCM

4-7. Image Data ( road, vegetation, sky )

이번 실험에서는 실제 영상의 분할에 대한 각 알고리즘의 성능의 차이를 비교해 보도록 한다. 사용되는 두 개의 속성 영상은 excess green 영상과 median 필터에 의해 필터링 된 영상이다.

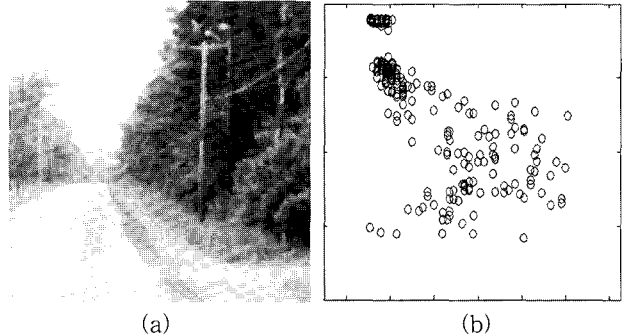


그림 7. 노이즈가 포함되지 않은 영상  
Fig. 7. Noiseless Image (a) Real Image (b) Extracted Data

먼저 200x200의 사이즈를 갖는 실제 영상을 분할하기 위해서 각 영역에서 100개씩 샘플 데이터를 추출하여 학습시킨 후 그 결과에 의해서 영상을 분할하도록 한다. 추출된 300개의 데이터는 그림 7.(b)와 같다. 샘플 데이터의 클러스터링 결과와 실질적인 영상분할 결과는 다음 그림과 같다.

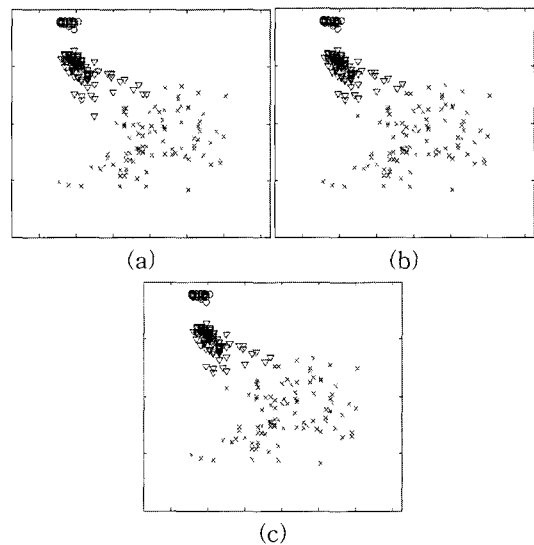


그림 8. 노이즈 없는 영상의 샘플데이터에 대한 클러스터링 결과  
Fig. 8. Clustering Results for Noiseless Image (a) FCM (b) CFCM (c) DEFCM

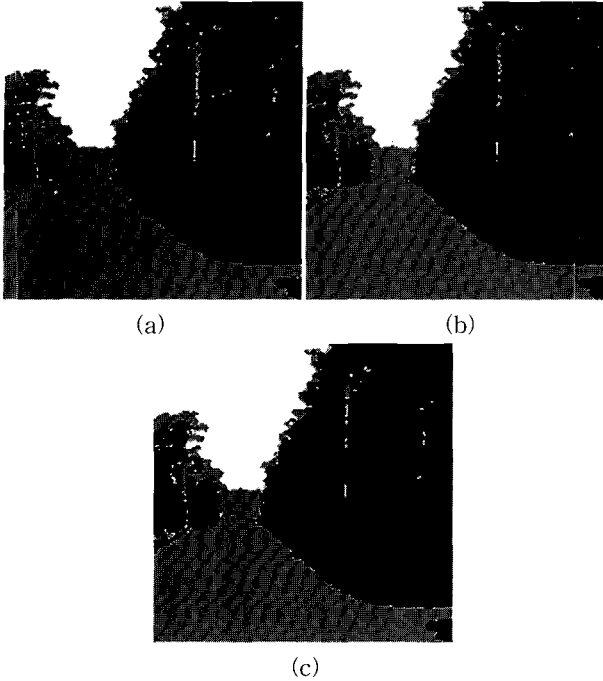


그림 9. 노이즈 없는 영상의 영역분할 결과

Fig. 9. Image Segmentation Results for Noiseless Image (a) FCM (b) CFCM (c) DBFCM

표 1. 노이즈가 없는 영상의 인식률 비교  
Table 1. Confusion Matrix for Noiseless Image.

|    | FCM        |       |      | CFCM       |       |      | DBFCM      |       |      |
|----|------------|-------|------|------------|-------|------|------------|-------|------|
|    | 인식률=86.55% |       |      | 인식률=87.70% |       |      | 인식률=88.42% |       |      |
|    | 길          | 숲     | 하늘   | 길          | 숲     | 하늘   | 길          | 숲     | 하늘   |
| 길  | 8219       | 0     | 1    | 8220       | 0     | 0    | 7538       | 25    | 657  |
| 숲  | 4548       | 20074 | 824  | 4112       | 20533 | 801  | 3836       | 20820 | 790  |
| 하늘 | 5          | 0     | 6329 | 5          | 0     | 6329 | 5          | 0     | 6329 |

세가지 방법으로 영상을 분할한 결과 DBFCM, CFCM, FCM 순으로 인식률이 좋았다. 하지만 노이즈가 없는 경우에 FCM도 상당히 좋은 결과를 나타낸다.

4-8. Noisy Image Data (road, vegetation, sky)

이번에는 영상에 노이즈를 주었을 경우 세가지 방법의 분할 결과를 알아본다. 앞에서 사용한 길, 숲, 하늘 부분으로 이루어진 영상에 5%의 노이즈를 준 영상은 다음과 같다.

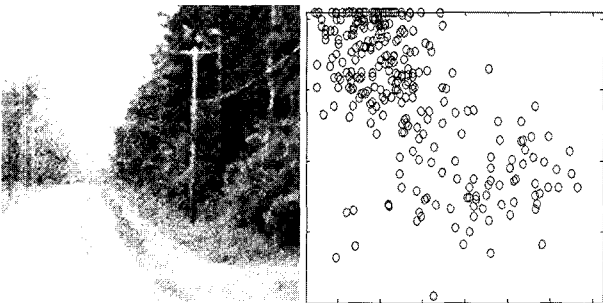


그림 10. 노이즈가 포함된 영상  
Fig. 10. Noisy Image (a) Real Image (b) Extracted Data

역시 앞의 실험과 마찬가지로 위의 영상에서 각 클러스터마다 100개씩의 샘플데이터를 추출하여 학습시킨 후, 그 결과에 의한 영상 분할의 결과를 비교한다. 그림 11은 각 알고리즘에 의하여 샘플 데이터를 클러스터링 한 결과를 나타낸다.

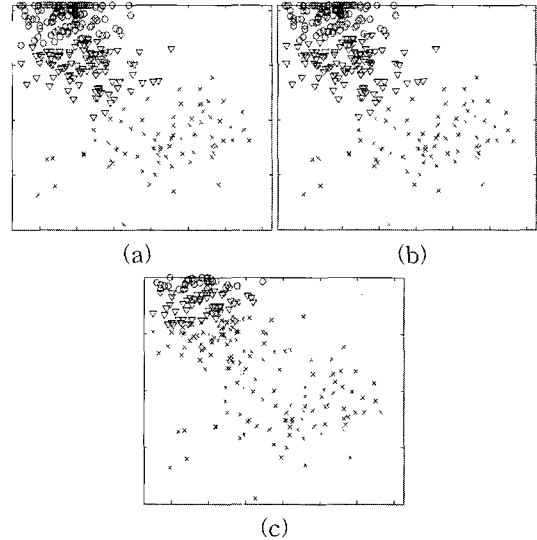


그림 11. 노이즈 있는 영상의 샘플 데이터에 대한 클러스터링 결과

Fig. 11. Clustering Results for Noisy Image (a) FCM (b) CFCM (c) DBFCM

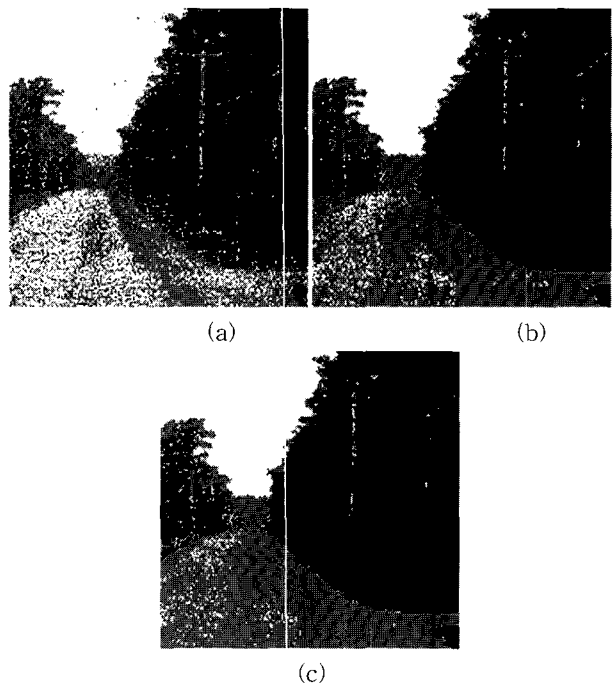


그림 12. 노이즈가 있는 영상의 영역분할 결과

Fig. 12. Image Segmentation Results for Noisy Image (a) FCM (b) CFCM (c) DBFCM

그림 12에서 보듯이 FCM의 경우 결과가 가장 나쁘고 하늘과 길 부분에서 잘못 분할된 픽셀이 상당히 많았다. 그리고 세가지 방법중에서 DBFCM의 결과가 가장 좋음을 알 수 있다.

표 2. 노이즈가 포함된 영상의 인식률 비교  
Table 2. Confusion Matrix for Noisy Image.

|    | FCM<br>인식률=72.74% |       |      | CFCM<br>인식률=84.15% |       |      | DBFCM<br>인식률=85.21% |       |      |
|----|-------------------|-------|------|--------------------|-------|------|---------------------|-------|------|
|    | 길                 | 숲     | 하늘   | 길                  | 숲     | 하늘   | 길                   | 숲     | 하늘   |
| 길  | 4624              | 18    | 3578 | 7818               | 30    | 672  | 7538                | 25    | 657  |
| 숲  | 5817              | 18147 | 1482 | 4737               | 19812 | 897  | 4720                | 19832 | 894  |
| 하늘 | 10                | 0     | 6324 | 5                  | 0     | 6329 | 5                   | 0     | 6329 |

표에서 볼 수 있듯이 노이즈가 포함되었을 경우 FCM의 인식률은 CFCM과 DBFCM에 비해 12%정도 낮게 나왔다. 그리고 DBFCM이 CFCM보다 인식률이 1%정도 높음을 알 수 있다. 결과에서 볼 수 있듯이 영상에 노이즈가 포함 되었을 때 FCM의 경우 인식률이 현저하게 떨어진다. 반면에 CFCM과 DBFCM의 경우 인식률이 조금밖에 안 떨어짐을 알 수 있다. CFCM과 DBFCM를 비교해 보면 DBFCM이 CFCM보다 인식률 결과가 더 좋음을 알 수 있다.

### 5. 결 론

본 논문에서는 Parzen Window방법을 사용한 DBFCM 알고리즘을 제안하였다. 결과에서 보았듯이 제안하는 알고리즘은 기본적으로 FCM과 비슷한 성능을 보이면서, 노이즈가 포함된 데이터에 대해서는 FCM 보다 우수한 성능을 보였다. 그리고 CFCM과 달리 초기 센터값에 민감하다는 단점을 보이지도 않는다. 하지만 제안하는 DBFCM 알고리즘에서는 원하는 결과값을 얻기 위해, 최적의 smoothing parameter h 값을 구하는 것이 매우 중요하다. 파라미터 h 를 어떻게 정해주냐에 따라 결과값이 다양하게 나타나기 때문이다. 따라서 어떠한 형태의 데이터가 입력되더라도, 최적의 h 값을 구할 수 있는 방법을 찾는 것이, 향후 해결해야 할 과제이다.

### 참 고 문 헌

- [1] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.
- [2] R. Krishnapuram and J. Keller, "A Possibilistic Approach to Clustering," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 2, pp. 98-110, 1993.
- [3] K. Chintalapudi and M. Kam, "A Noise Resistant Fuzzy C means Algorithm for Clustering," *IEEE Int. Conf. Fuzzy Systems (FUZZIEEE1998)*, Anchorage, AK, May 4-9, 1998.
- [4] N. Pal, K. Pal, and J. Bezdek, "A Mixed C-Means Clustering Model," *IEEE Int. Conf. Fuzzy Systems*, 1997, Spain, pp. 11-21.
- [5] L.E.Scales, *Introduction to Non-Linear Optimization*. New York: Plenum Press, 1981.
- [6] H. Duda, P. Hart, and D. Stork, *Pattern Classification*. Second ed. John Wiley & Sons, 2001.
- [7] A. Bowman and A. Azzalini, *Applied Smoothing Techniques for Data Analysis*. London: Oxford University Press, 1997.

## 저 자 소 개



고정원 (Jeong-Won Ko)

2005년 : 한양대학교 전자컴퓨터 공학부 졸업

2005년~현재 : 동 대학원 메카트로닉스 공학과 석사과정

관심분야 : Fuzzy Systems, Pattern Recognition.

E-mail : [jwko@fuzzy.hanyang.ac.kr](mailto:jwko@fuzzy.hanyang.ac.kr)



최병인 (Byung-In Choi)

2001년: 한양대학교 전자공학과 졸업

2001년~2003년: 동 대학원 전자전기제어 계측공학과 석사

2003년~현재: 동 대학원 전자전기제어계 측공학과 박사과정

관심분야 : Fuzzy Systems, Pattern Recognition.

E-mail : [bichoi@fuzzy.hanyang.ac.kr](mailto:bichoi@fuzzy.hanyang.ac.kr)



이정훈 (Frank Chung-Hoon Rhee)

1981년~1985년 : University of Southern California (USC), Dept. of Electrical Eng., BSEE.

1985년~1987년 : University of Missouri, Dept. of Electrical & Computer Eng., MSEE.

1990년~1993년 : University of Missouri,

Dept. of Electrical & Computer Eng., Ph.D.

1994년~1995년 : 한국전자통신연구원

1995년~현재 : 한양대학교 전자컴퓨터공학부 부교수

2003년~현재 : 세계 퍼지 학회(IFSA) 한국 지부 대표

관심분야 : Computer Vision, Pattern Recognition, and all aspects of Computational Intelligence.

E-mail : [frhee@fuzzy.hanyang.ac.kr](mailto:frhee@fuzzy.hanyang.ac.kr)