

확장된 퍼지 클러스터링 알고리즘을 이용한 영상 분할

(Image Segmentation Using an Extended Fuzzy Clustering Algorithm)

金 秀 桓*, 姜 景 辰**, 李 太 遠**

(Su Hwan Kim, Kyung Jin Kang, and Tae Won Rhee)

要 約

최근들어 영상 처리 분야의 응용에 퍼지 이론이 활발히 적용되고 있다. 특히 퍼지 클러스터링 알고리즘이 영상 분할에 응용되어 영상내에 포함되어 있는 애매함과 잡음의 영향을 줄이기 위해 적용되고 있으나 영상내에 포함되어 있는 방대한 양의 데이터 때문에 많은 메모리와 실행 시간을 필요로 한다. 따라서 영상 자체의 특성을 변화시키지 않으면서 메모리와 실행 시간을 줄일 수 있을 뿐 아니라 기존의 퍼지 클러스터링 알고리즘과 같은 영상 분할 효과를 갖는 새로운 영상 분할 알고리즘이 필요하다. 본 논문에서는 이를 위하여 주어진 영상내 특징들을 직접 사용하는 대신 같은 특징값을 갖는 데이터들의 수를 그 특징값의 가중치로 사용하는 확장된 퍼지 클러스터링 알고리즘을 사용할 경우 기존의 퍼지 클러스터링 알고리즘에 의한 영상분할 보다 메모리와 실행 시간을 감소시킬 수 있음을 입증하였다.

Abstract

Recently, the fuzzy theory has been adopted broadly to the applications of image processing. Especially the fuzzy clustering algorithm is adopted to image segmentation to reduce the ambiguity and the influence of noise in an image. But this needs lots of memory and execution time because of the great deal of image data. Therefore a new image segmentation algorithm is needed which reduces the memory and execution time, doesn't change the characteristics of the image, and simultaneously has the same result of image segmentation as the conventional fuzzy clustering algorithm. In this paper, for image segmentation, an extended fuzzy clustering algorithm is proposed which uses the occurrence of data of the same characteristic value as the weight of the characteristic value instead of using the characteristic value directly in an image and it is proved the memory reduction and execution time reduced in comparison with the conventional fuzzy clustering algorithm in image segmentation.

I. 서 론

*準會員, **正會員, 高麗大學校 電子工學科
(Dept. of Elec. Eng., Korea Univ.)
接受日字: 1991年 10月 31日

영상 분할은 주어진 영상을 몇개의 서로 다른 의미를 갖는 영역으로 분할하는 것이다. 이는 영상을 분석하는데 필수적인 과정으로 최근에는 의학 분야

에도 응용되어 상당한 효과를 보고 있다.^[3]

지난 20여년 동안 많은 영상 분할 방법이 제안되어 왔다. 이들 영상분할 방법들은 크게 3가지로 분류될 수 있는데, 이는 영상 특징의 임계화 및 클러스터링, 에지검출, 그리고 영역 추출이다.^[4] 영상 특징의 임계화 및 클러스터링, 영역추출에 의한 영상분할 방법이 영상내 화소들의 유사성을 이용하고 에지 검출에 의한 영상분할 방법은 영상내 화소들의 불연속성과 관계가 있다. 즉, 전체적인 관점에서의 영상 분할은 영상 특징의 임계화 및 클러스터링, 영역 추출에 의한 영상분할 방법을 이용함으로써 구현할 수 있는 반면에 국부적인 관점에서의 영상분할은 에지검출에 의한 영상 분할 방법을 이용함으로써 구현할 수 있다.^[4] 한편, 클러스터링에 의한 영상분할 방법은 유사한 특징을 갖는 화소들을 하나의 영역으로 분할하므로 개념적으로 매우 간단할 뿐 아니라 다른 영상 분할 방법에 비하여 잡음에 더 강하다.^[5] 그러나 이러한 접근 방법은 영상내 다른 영역은 특징 공간에서 명확하게 구분지어 진다는 가정에 기반을 두고 있기 때문에 이 가정에 벗어나는 경우 영상 분할에 실패할 가능성이 높다.^[6]

한편 최근들어 영상처리 분야의 응용에 퍼지 이론이 활발히 적용되고 있다. 특히 퍼지 클러스터링 알고리즘이 영상 분할에 응용되어 영상내에 포함되어 있는 애매함과 잡음의 영향을 줄이기 위해 적용되고 있으나 영상내에 포함되어 있는 방대한 양의 데이터 때문에 많은 메모리와 실행 시간을 필요로 한다.

이러한 문제점을 해결하기 위한 대표적인 방법으로는 피라미드 데이터 구조에 의한 접근 방법이 있다. 그러나 이러한 접근 방법은 몇가지 한계를 가지고 있는데 특히 평균 효과는 본질적인 영상 특징을 파괴하여 같은 물체가 점차적으로 해상도를 높여감에 따라 동일한 특징을 가지지 않을 수 있으며 연속된 무늬의 해석에 적용되었을 때는 치명적일 수 있다.^[7]

따라서 영상 자체의 특성을 변화시키지 않으면서 메모리와 실행 시간을 줄일 수 있을 뿐 아니라 기존의 퍼지 클러스터링 알고리즘과 같은 영상분할 효과를 갖는 새로운 영상 분할 알고리즘이 필요하다.

본 논문에서는 이를 위하여 주어진 영상내 특징들을 직접 사용하는 대신 같은 특징값을 갖는 데이터들의 수를 그 특징값의 가중치로 사용하는 확장된 퍼지 클러스터링 알고리즘^{[1],[11]}을 사용할 경우 최적 퍼지 클러스터링 알고리즘^[17]에 의한 영상분할 보다 메모리와 실행 시간을 감소시킬 수 있음을 실험적으로 입증하였다.

II. 퍼지 클러스터링 알고리즘

1. 크리스프 클러스터링과 퍼지 클러스터링

크리스프 클러스터링 알고리즘은 각 데이터들이 하나의 클러스터에 할당되도록 분할한다. 그러나 각 데이터가 엄격하게 하나의 클러스터에 할당되는 것이 적절하지 못한 경우가 있다. 이와 같은 경우 퍼지 클러스터링 알고리즘을 실제 데이터의 구조를 나타내는데 더 많은 유연성을 갖는다. 그림1은 15개의 점으로 구성되는 데이터 집합 X를 나타내고 있다. 크리스프 클러스터링 알고리즘에 의해 이들 점들은 그림2에 나타낸 바와 같이 분류된다. 여기서 '1'은 왼쪽 클러스터에 속하는 것을 표시하며 '0'은 오른쪽 클러스터에 속하는 점을 표시한다. 반면 그림3은 각각 퍼지 클러스터링 알고리즘에 의해서 각 점들의 왼쪽과 오른쪽 클러스터에 속하는 귀속 정도를 나타낸다. 그림2에서는 한 가운데 점이 반드시 왼쪽 또는 오른쪽 클러스터 중 하나에 귀속되어야 하기 때문에 비대칭이지만 그림3에서는 이 점의 귀속 정도가 각각 0.5이므로 대칭이다.^[8]

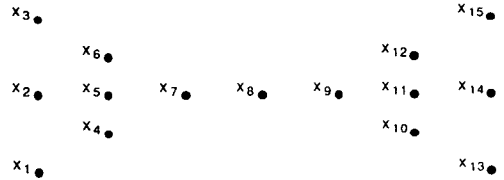


그림 1. 나비 모양 모델
Fig. 1. A butterfly model.

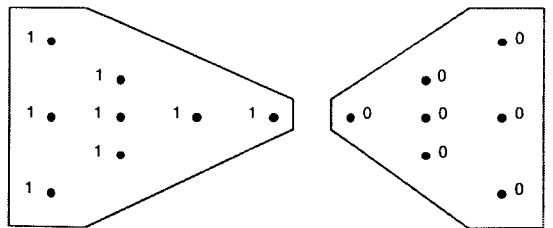


그림 2. 나비 모양 패턴의 크리스프 클러스터들
Fig. 2. Two clusters of the butterfly in crisp clustering.

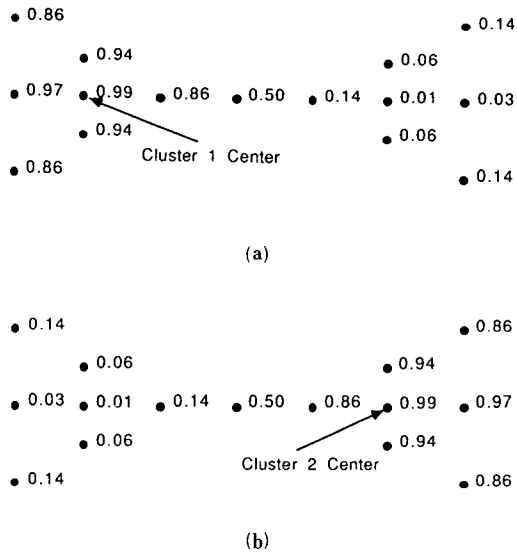


그림 3. 나비 모양 패턴의 퍼지 클러스터들
 (a) 첫번째 클러스터 (b) 두번째 클러스터들
 Fig. 3. Two clusters of the butterfly in fuzzy clustering.
 (a) a cluster 1 of the butterfly,
 (b) a cluster of the butterfly.

초기 중심값 설정 알고리즘이 필요하다.

3. 초기 중심값 설정 알고리즘

기본적인 개념은 데이터가 c 개의 클러스터로 분할되어있을 때 낮은 정도의 소속함수들이 있는 영역에 $(c+1)$ 번째 초기 중심값을 위치시킨다는 것이다. 최적 퍼지 클러스터링 알고리즘을 위한 초기 중심값 설정 알고리즘은 다음과 같다.⁷⁾

- 1) 전체 데이터의 평균과 표준편차를 구한다.
- 2) 전체 데이터의 평균에 첫번째 초기 중심값을 위치 시킨다.
- 3) 전체 데이터로 부터 동일한 거리에 또 하나의 초기 중심값을 위치시킨다.
- 4) 변형된 퍼지 c -means 알고리즘과 변형된 퍼지 MLE 알고리즘을 이용하여 데이터를 분할한다.
- 5) 클러스터 수, c 를 증가시키고 최적의 분할이 얻어질 때까지 3)에서 5)까지의 과정을 반복한다.

4. 변형된 퍼지 c -means (MFCM) 알고리즘

MFCM 알고리즘은 각 클러스터의 중심으로 부터 데이터 사이의 유사도 측정에 기반을 둔 목적 함수의 반복적 최적화를 이용한다. 목적함수는 다음과 같다.

$$J_m(U, V) = \sum_{k=1}^n \sum_{i=1}^c (u_{ik})^m (d_{ik})^2 \tag{1}$$

여기서 u_{ik} 는 i 번째 클러스터 안에서 데이터 x_k 의 소속함수이고 지수 m 은 클러스터링의 성질을 결정한다. 특히 $m=1$ 일 때 절대적인 하드 클러스터링이 되며 m 이 증가하면 더 유연성이 있는 퍼지 클러스터링이 된다.

거리 측정값 $d^2(x_k, v_i)$ 가 식(2)와 같이 정의된다고 할때 행렬 A 는 통계학적 특징에 따른 가중치로 작용한다.⁷⁾

$$d^2(x_k, v_i) = (x_k - v_i)^T A (x_k - v_i) \tag{2}$$

행렬 A 가 단위행렬일 때 이 거리는 유클리디안 거리가 되는데 MFCM 알고리즘은 이 거리를 사용한다. 즉, 식(2)로 부터 k 번째 데이터 x_k 와 i 번째 클러스터의 중심값 v_i 사이의 거리 d_{ik} 는 다음과 같다.

$$d_{ik} = d(x_k, v_i) \tag{3}$$

$$= \|x_k - v_i\| \tag{4}$$

$$= \left\{ \sum_{j=1}^p (x_{kj} - v_{ij})^2 \right\}^{1/2} \tag{5}$$

MFCM 알고리즘은 목적함수 $J_m(U, V)$ 을 최소화할 수 있도록 U 와 V 를 적절하게 선택하는 것에 의존한다. 여기서 V 는 모든 클러스터의 중심값이다.

2. 최적 퍼지 클러스터링 알고리즘

클러스터 모양과 밀도의 다양성과 초기 중심값 결정의 문제점에 대한 만족할만한 해결책을 얻기 위하여 최적 퍼지 클러스터링이라 불리는 2단계 퍼지 클러스터링 알고리즘이 제안되었다!¹⁾ 이 알고리즘은 클러스터 수를 증가시키면서 각 단계의 분할 정도를 평가하고 최적의 분할이 얻어질 때까지 다음을 반복한다.⁷⁾

- 1) II-3에 있는 초기 중심값 설정 알고리즘을 이용하여 각 클러스터의 중심값을 결정한다.
- 2) II-4에 있는 변형된 퍼지 c -means 알고리즘을 이용하여 데이터를 분할한다.
- 3) II-5에 있는 변형된 퍼지 MLE 알고리즘을 이용하여 데이터를 분할한다.
- 4) II-6에 있는 성능평가 함수를 이용하여 분할 정도를 평가한다.
- 5) 클러스터 수, c 를 증가시키고 최적의 분할이 얻어질 때까지 1)에서 5)까지의 과정을 반복한다.

일반적으로 퍼지 클러스터링 알고리즘은 임의로 초기 중심값을 정하며 다른 초기 중심값 설정은 다른 분할 효과를 나타낼 수 있다. 실제 경우에 있어서 대략적인 초기 중심값에 대한 사전정보를 얻을 수 없으므로 최적의 분할을 얻기 위해서는 효과적인

이 알고리즘은 $1 \leq i \leq c$ 와 $1 \leq k \leq n$ 에 대하여 다음과 같다.⁷⁾

1) $2 \leq c \leq n$ 에 대하여 클러스터 수, c 와 $1 \leq m \leq \infty$ 에 대하여 지수 m 를 선택한다.

2) II-3에 있는 초기 중심값 설정 알고리즘을 이용하여 초기 중심값들을 초기화 한다.

3) i 번째 클러스터에 대한 모든 데이터들의 소속 함수를 구한다.

$$u_{ik} = \frac{1}{\sum_{j=1}^c (d_{1k}/d_{jk})^{2/(m-1)}} \quad (6)$$

4) i 번째 클러스터의 중심값을 구한다.

$$v_i' = \frac{\sum_{k=1}^n (u_{ik})^m x_k}{\sum_{k=1}^n (u_{ik})^m} \quad (7)$$

5) i 번째 클러스터에 대한 모든 데이터들의 소속 함수를 구한다.

$$u'_{ik} = \frac{1}{\sum_{j=1}^c (d_{1k}/d_{jk})^{2/(m-1)}} \quad (8)$$

6) U 와 U' 를 비교한다. $\|u_{ik} - u'_{ik}\| \leq \epsilon$ 의 조건을 만족하면 멈추고 그렇지 않으면 4)부터 6)까지의 과정을 반복한다. ϵ 은 멈춤 조건으로 0과 1사이의 값을 가지며 보통 0에 가까운 값을 갖는다.

5. 변형된 퍼지 각 클러스터에 다른 수의 데이터가 존재할 뿐 아니라 다양한 밀도가 존재할 때에 대하여 exponential 거리, $d_e^2(x_k, v_i)$ 가 MLE 알고리즘에 근거하여 정의되었다.¹¹⁾ 이 거리는 k 번째 데이터가 주어졌을 때 i 번째 클러스터에 속하는 정도, $h(i|x_k)$ 를 계산하기 위하여 사용된다.

$$h(i|x_k) = \frac{1/d_e^2(x_k, v_i)}{\sum_{k=1}^n 1/d_e^2(x_k, v_i)} \quad (9)$$

$$d_e(x_k, v_i) = \frac{|F_i|^{1/2}}{P_i} \exp[(x_k - v_i)^T F_i^{-1}(x_k - v_i)/2] \quad (10)$$

여기서 F_i 는 i 번째 클러스터의 퍼지 공분산 행렬을 나타내고 P_i 는 i 번째 클러스터를 선택할 전확률이다.

$m=2$ 에 대하여 식(9)의 $h(i|x_k)$ 는 식(6)의 u_{ik} 와 비슷하다. 그래서 MFCM 알고리즘의 식(6)에 대신 식(9)를 대입하면 MFMLE 알고리즘이 된다. MFMLE 알고리즘은 i 번째 클러스터에 속할 전확률, P_i 의 계산과 퍼지 공분산 행렬, F_i 의 계산을 포함한다.

$$P_i = \frac{1}{n} \sum_{k=1}^n h(i|x_k) \quad (11)$$

$$F_i = \frac{\sum_{k=1}^n h(i|x_k) (x_k - v_i)^T (x_k - v_i)}{\sum_{k=1}^n h(i|x_k)} \quad (12)$$

6. 최적 클러스터 수의 결정

아직까지는 최적 클러스터 수의 결정에 대한 일반적인 해결 방법은 제시되고 있지 못하고 있으나¹⁴⁾ 최적 퍼지 클러스터링 알고리즘에서 사용한 데이터의 최적 분할에 대한 평가 기준은 다음과 같다.⁷⁾

- 분할된 클러스터들간의 경계가 확실한지 여부
- 클러스터들의 크기가 최소가 되는지 여부
- 클러스터의 중심부분에 데이터 밀도가 최대가 되는지 여부

지금까지 설명한 바와 같이 MFCM 알고리즘과 MFMLE 알고리즘에 근거를 둔 최적 퍼지 클러스터링 알고리즘은 반복적인 계산과 데이터 및 소속 함수의 양 때문에 많은 메모리와 실행 시간을 필요로 한다. 일반적으로 데이터와 소속 함수가 전체 메모리의 대부분을 차지하며 소속 함수의 계산이 전체 실행 시간의 대부분을 차지한다. 다음장에서는 최적 퍼지 클러스터링 알고리즘에 근거를 둔 확장된 퍼지 클러스터링 알고리즘이 이러한 문제점들을 해결하기 위하여 서로 다른 특징 값을 가진 데이터들과 이에 따른 소속 함수만을 고려하도록 제안하였다.

III. 확장된 퍼지 클러스터링 알고리즘

확장된 퍼지 클러스터링 알고리즘은 같은 특징값을 갖는 데이터들의 발생 빈도를 고려하고 이를 그 특징값에 대한 가중치로 사용하였다. 이는 어떤 특징에 대하여 같은 특징값을 갖는 경우가 한번 이상 발생할 수 있으며 같은 특징값들을 갖는 데이터들은 같은 소속 함수를 갖기 때문이다. 따라서 같은 특징값을 갖는 데이터들의 소속 함수는 한번만 계산하고 이 특징값을 갖는 데이터에 대한 가중치를 적절하게 이용함으로써 서로 다른 특징값을 갖는 데이터들과 이들 데이터들의 가중치만으로도 기존의 방법과 같은 데이터 분할이 가능하다. 결국 이러한 접근 방법은 데이터와 소속함수의 크기를 줄인다. 데이터와 이들의 소속 함수가 전체 메모리의 대부분을 차지하며 계산의 복잡성을 결정하므로 메모리와 실행 시간의 관점에서 이 알고리즘이 효과가 있다.

최적 퍼지 클러스터링 알고리즘에 근거를 둔 확장된 퍼지 클러스터링 알고리즘은 다음과 같다.

1) III-1에 있는 새로운 초기 중심값 설정 알고리즘을 이용하여 각 클러스터의 초기 중심값을 결정한다.

2) III-2에 있는 확장된 퍼지 c-means 알고리즘을 이용하여 데이터를 분할한다.

3) III-3에 있는 확장된 퍼지 MLE 알고리즘을 이용하여 데이터를 분할한다.

4) II-6에 있는 성능 평가 함수를 이용하여 분할 정도를 평가한다.

5) 클러스터 수, c를 증가시키고 최적의 분할이 얻어질 때까지 1)에서 5)까지의 과정을 반복한다.

1. 새로운 초기 중심값 설정 알고리즘

다음은 확장된 퍼지 클러스터링 알고리즘을 위한 새로운 초기 중심값 설정 알고리즘이다.

1) 전체 데이터의 평균과 표준편차를 구한다.

2) 전체 평균으로부터 서로 반대 방향으로 같은 거리(표준편차)만 좀 떨어진 위치에 두개의 초기 중심값을 위치시킨다.

3) 확장된 퍼지 c-means 알고리즘과 확장된 퍼지 MLE 알고리즘을 이용하여 데이터를 분할한다.

4) i번째 클러스터의 데이터중 중심으로 부터 T_p 만큼의 거리안에 포함되는 데이터 비 ρ_i 를 계산한다. $1 \leq k' \leq n' \leq n$ 에 대하여 서로 다른 특징값을 갖는 데이터들의 집합 $X' = \{x_1, \dots, x_{k'}, \dots, x_{n'}\}$ 이 전체 데이터의 집합 $X = \{x_1, \dots, x_k, \dots, x_n\}$ 의 부분 집합이라 하자. $x_{k'}$ 는 서로 다른 특징값을 갖는 집합의 데이터이고 이 데이터의 발생 빈도는 $\omega_{k'}$ 이다. 여기서 $\omega_{k'} \geq 1, n = \sum_{k'=1}^{n'} \omega_{k'}$ 는 정수이다. i번째 클러스터에 대한 비 ρ_i 는 다음과 같다.

$$\rho_i = \frac{\sum_{k'=1}^{n'} e_{ik'} \omega_{k'} u_{ik'}}{\sum_{k'=1}^{n'} \omega_{k'} u_{ik'}} \quad (13)$$

여기서

$$e_{ik'} = \begin{cases} 1 & \text{만약 } (x_{k'} - v_i)^T \sigma_i^{-1} (x_{k'} - v_i) \leq T_p \\ 0 & \text{그렇지 않은 경우} \end{cases}$$

한편 i번째 클러스터의 중심 부분에 많은 데이터가 집중되어 있으면 다른 클러스터에 비해 ρ_i 의 값이 상대적으로 크다. 특별한 경우에는 i번째 클러스터의 분포가 평균이 v_i 이고 분산 σ_i 인 가우시안 분포를 따른다고 할 때 T_p 는 $x_{\alpha}(\alpha)$ 와 같으며 이 범위안에 $(1-\alpha) \times 100\%$ 만큼의 데이터를 포함한다⁶⁾

5) 클러스터 수, c를 증가시키고 최적의 분할이 얻어질 때까지 3)에서 5)까지의 과정을 반복한다.

2. 확장된 퍼지 c-means (EFM) 알고리즘
식(2)로 부터 목적함수 $J_m(U, V)$ 는 다음과 같이 변형된다. 여기서 $d_{ik'} = d(x_{k'}, v_i)$ 이다.

$$J_m(U, V) = \sum_{k'=1}^{n'} \sum_{i=1}^c \omega_{k'} (u_{ik'})^m (d_{ik'})^2 \quad (14)$$

$$= \sum_{k'=1}^{n'} \omega_{k'} \sum_{i=1}^c (u_{ik'})^m (d_{ik'})^2 \quad (15)$$

EFM 알고리즘은 II-4에 있는 MFCM 알고리즘의 변형에 의하여 이루어지며 $1 \leq i \leq c$ 와 $1 \leq k' \leq n'$ 에 대하여 다음과 같다.

1) $2 \leq c \leq n'$ 에 대하여 클러스터 수, c와 $1 \leq m \leq \infty$ 에 대하여 지수 m을 선택한다.

2) III-1에 있는 초기 중심값 설정 알고리즘을 이용하여 초기 중심값들을 초기화 한다.

3) i번째 클러스터에 대한 모든 데이터들의 소속 함수를 구한다.

$$u_{ik'} = \frac{1}{\sum_{j=1}^c (d_{ik'}/d_{jk'})^{2/(m-1)}} \quad (16)$$

4) i번째 클러스터의 중심값을 구한다.

$$v'_i = \frac{\sum_{k'=1}^{n'} \omega_{k'} (u_{ik'})^m x_{k'}}{\sum_{k'=1}^{n'} \omega_{k'} (u_{ik'})^m} \quad (17)$$

5) i번째 클러스터에 대한 모든 데이터들의 소속 함수를 구한다.

$$u'_{ik'} = \frac{1}{\sum_{j=1}^c (d_{ik'}/d_{jk'})^{2/(m-1)}} \quad (18)$$

6) U와 U'를 비교한다. $\|u_{ik'} - u'_{ik'}\| \leq \epsilon$ 의 조건을 만족하면 멈추고 그렇지 않으면 4)부터 6)까지의 과정을 반복한다. ϵ 은 멈춤 조건으로 0과 1 사이의 값을 가지며 보통 0에 가까운 값을 갖는다.

3. 확장된 퍼지 MLE (EFMLE) 알고리즘

비슷하게 EFMLE 알고리즘은 식(9)와 식(10)으로부터 다음과 같이 정의된다.

$$h(i|x_{k'}) = \frac{1/d_e^2(x_{k'}, v_i)}{\sum_{k'=1}^{n'} 1/d_e^2(x_{k'}, v_i)} \quad (19)$$

$$d_e(x_{k'}, v_i) = \frac{|F_1|^{1/2}}{P_i} \exp[(x_{k'} - v_i)^T F_1^{-1} (x_{k'} - v_i)/2] \quad (20)$$

여기서

$$v_i = \frac{\sum_{k'=1}^{n'} \omega_{k'} h(i|x_{k'})^m x_{k'}}{\sum_{k'=1}^{n'} \omega_{k'} h(i|x_{k'})^m} \quad (21)$$

또한 식(11)과 식(12)로부터 i 번째 클러스터에 대한 전확률, P_i 와 퍼지 공분산 행렬, F_i 는 다음과 같이 정의된다.

$$P_i = \frac{1}{n} \sum_{k'=1}^{n'} \omega_{k'} h(i|x_{k'}) \quad (22)$$

$$F_i = \frac{\sum_{k'=1}^{n'} \omega_{k'} h(i|x_{k'}) (x_{k'} - v_i)^T (x_{k'} - v_i)}{\sum_{k'=1}^{n'} \omega_{k'} h(i|x_{k'})} \quad (23)$$

최적 퍼지 클러스터링 알고리즘과 비교하여 확장된 퍼지 클러스터링 알고리즘의 차이는 서로 같은 특징값을 갖는 데이터들의 발생 빈도를 고려한 것이다.

4. 모의 실험

모의 실험을 위해 표1의 4개의 가우시안 분포들로 이루어진 1150개의 난수들을 전체 데이터로 이용한다. 이 난수들은 0과 200사이에서 발생한다. 이 데이터 모델에서 전체 데이터 수, n 은 1150이고 서로 다른 특징값을 갖는 데이터 수, n' 는 116이다. 최적 클러스터 수는 $2 \leq c \leq 5$ 의 범위에서 고려된다.

표 1. 모의 실험을 위한 4개의 가우시안 분포 데이터 모델
Table 1. Four gaussian distribution data models for an simulation.

The Number of Data	An Average	A Standard Deviation
250	39.52	5.28
300	64.79	10.48
400	99.99	5.27
200	170.05	5.32

그림4는 데이터의 리스트이며 이 때 (a)는 최적 퍼지 클러스터링 알고리즘을 위한 데이터들이며 (b)는 확장된 퍼지 클러스터링 알고리즘을 위한 데이터들과 이들의 가중치이다.

그림5는 제안된 확장된 퍼지 클러스터링 알고리즘에 의한 데이터 분할 과정을 보여준다. 데이터를 2개의 클러스터로 나누고자할 때 두개의 초기 중심값들은 {46.22, 133.58} (흰색 점들)이 전체 데이터들의 평균으로부터 전체 거리의 표준편차 만큼 서로 반대 방향으로 떨어진 위치에 둔다. 데이터를 두개

k	x_k	k'	$x_{k'}$	$\omega_{k'}$
1	39	1	39	15
2	42	2	42	22
3	34	3	34	11
4	40	4	40	23
5	45	5	45	19
.	.	6	38	16
.	.	7	44	19
114	35	8	43	16
115	45	9	35	15
116	41	10	41	15
117	40	.	.	.
118	38	.	.	.
.	.	110	166	6
.	.	111	162	4
1146	172	112	179	2
1147	176	113	186	1
1148	160	114	182	1
1149	166	115	159	1
1150	169	116	158	1

그림 4. 모의 실험을 위한 데이터 목록
(a) 최적 퍼지 클러스터링 알고리즘을 위한 데이터 목록
(b) 확장된 퍼지 클러스터링 알고리즘을 위한 데이터 목록

Fig. 4. Data list for an simulation.
(a) data list for optimal fuzzy clustering algorithm,
(b) data list for an extended fuzzy clustering algorithm.

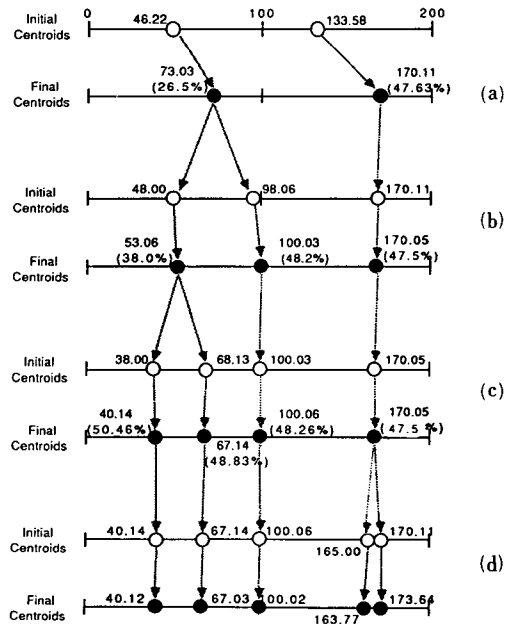


그림 5. 모의 실험 결과
(a) 2개의 클러스터 (b) 3개의 클러스터
(c) 4개의 클러스터 (d) 5개의 클러스터
Fig. 5. Results of an simulation.
(a) two clusters, (b) three clusters,
(c) four clusters, (d) five clusters.

로 나누고자할 때 최종 중심값들 {73.03, 170.11} 을 EFCM 알고리즘과 EFMLE 알고리즘에 의해 결정한다. 식(13)에 의하여 2개의 클러스터들의 ρ_1 값 {26.5%, 47.63%}를 계산한다. ρ_1 값이 26.5%인 클러스터를 2개의 클러스터로 나눈다. 그림 3 개의 초기 중심값{170.11} (미리 결정되어 있음)과 {48.00, 98.06} (ρ_1 값이 26.5%인 클러스터의 최종 중심으로부터 그 클러스터의 표준편차 만큼 서로 반대 방향으로 떨어진 위치에 새롭게 결정함)을 위치시킨다. 비슷하게, 클러스터 수가 3, 4, 5인 경우에 대해서도 확장된 퍼지 클러스터링 알고리즘이 적용된다. 그림 4에서 최적 클러스터 수 c_{opt} 가 4이다.

IV. 성능 분석

1. 메모리

확장된 퍼지 클러스터링 알고리즘은 최적 퍼지 클러스터링 알고리즘에 비하여 메모리에서 감소 효과가 있다. 확장된 퍼지 클러스터링 알고리즘은 $(p+c_{opt}+1) \times n'$ 만큼의 메모리를 사용하고 최적 퍼지 클러스터링 알고리즘은 $(p+c_{opt}) \times n$ 만큼의 메모리를 사용한다. 여기서 c_{opt} 는 최적 클러스터수를 나타내고 p 는 데이터의 차수를 나타낸다. 그러므로 식(24)가 만족된다면 확장된 퍼지 클러스터링 알고리즘은 최적 퍼지 클러스터링 알고리즘에 비하여 메모리에서 감소 효과가 있다.

$$\frac{(p+c_{opt}+1) \times n'}{(p+c_{opt}) \times n} < 1 \tag{24}$$

다시 말하면

$$n' < \frac{(p+c_{opt}) \times n}{(p+c_{opt}+1)} \tag{25}$$

그림6은 식(26)를 사용하여 메모리 감소 효과를 그래프로 나타낸 것이다.

$$y(\%) = \left(1 - \frac{(p+c_{opt}+1)}{p+c_{opt}} \times \frac{n'}{n}\right) \times 100 \tag{26}$$

실제 응용들에서 데이터 수와 그들의 소속 함수의 감소는 IV-2에서 계산되는 것보다 실행 시간에서 더 좋은 효과를 나타낸다. 이는 주 메모리와 보조 메모리 사이의 swap 시간이 크게 줄기 때문이다.

확장된 퍼지 클러스터링 알고리즘 메모리 감소 효과를 최적 퍼지 클러스터링 알고리즘과 비교하기 위하여 표1의 모델을 사용하였다. 식(26)에 의하면 데이터 차수, p 가 1이고 전체 데이터 수에 대한 서로 다른 특징값을 갖는 데이터의 비, n'/n 가 0.10이며 최

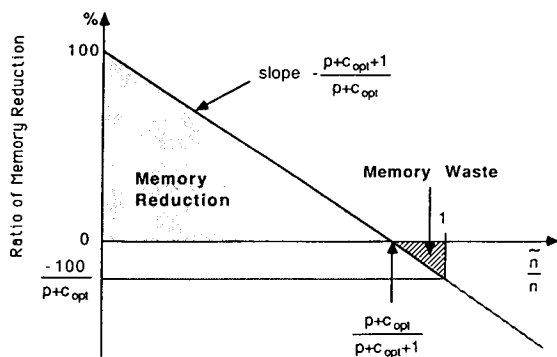


그림 6. 메모리 감소 효과
Fig. 6. The effectiveness of memory reduction.

적 클러스터 수, c_{opt} 는 4일때 메모리 감소 효과는 89.53%이다.

한편 데이터 차수, p 와 최적클러스터수, c_{opt} 가 고정되어 있을 때 전체 데이터 수에 대한 서로 다른 특징값을 갖는 데이터의 비, n'/n 가 감소하면 메모리 감소 효과는 커지며 전체 데이터 수에 대한 서로 다른 특징값을 갖는 데이터의 비, n'/n 가 고정되어 있을 때 $(p+c_{opt})$ 가 증가하면 $(p+c_{opt})/(p+c_{opt}+1)$ 이 1에 접근하기 때문에 메모리 감소효과는 데이터 감소 효과에 접근한다.

2. 실행 시간

실행 시간의 성능을 평가하기 위하여 다음과 같은 변수들을 고려한다.

- 더하기를 위한 실행시간 : Δ_{add}
- 곱하기를 위한 실행시간 : Δ_{mul}
- 나누기를 위한 실행시간 : Δ_{div}
- 거듭 제곱을 위한 실행시간 : Δ_{pow}
- 제곱근을 위한 실행시간 : Δ_{sqrt}
- 지수 함수를 위한 실행시간 : Δ_{exp}

• 최적 퍼지 클러스터링 알고리즘에서 MFCM 알고리즘과 확장된 퍼지 클러스터링 알고리즘에서 EFCM 알고리즘의 반복 횟수와 반복 끝냄 조건

$$: \zeta, \epsilon_{FCM}$$

• 최적 퍼지 클러스터링 알고리즘에서 MFMLE 알고리즘과 확장된 퍼지 클러스터링 알고리즘에서 EFMLE 알고리즘의 반복 횟수와 반복 끝냄 조건 : η, ϵ_{FMLE}

- 클러스터 수 : c
- 지수와 데이터의 차수 : m, p

표2는 최적 퍼지 클러스터링 알고리즘에서 포함되

표 2. 최적 퍼지 클러스터링 알고리즘내의 MFCM, MFMLE 알고리즘과 확장된 퍼지 클러스터링 알고리즘내의 EFCM, EFMLE 알고리즘 내의 실행 시간

Table 2. Execution time for a MFCM, a MFMLE in fuzzy optimal algorithm and an FECM, EFMLE in an extended fuzzy clustering algorithm.

Computation Objects	Algorithm	Execution Time
Membership Matrix, $U=[u_{ik}]$	MFCM	$cn(\zeta+1)(3p+3cp+2c) \times \Delta_{add} + cnp(\zeta+1)(c+1) \times \Delta_{mui}$ $+ cn(\zeta+1)(2c+1) \times \Delta_{div} + c^2n(\zeta+1) \times \Delta_{pow}$ $+ cn(\zeta+1)(c+1) \times \Delta_{sqrt}$
	EFCM	$cn(\zeta+1)(3p+3cp+2c) \times \Delta_{add} + cnp(\zeta+1)(c+1) \times \Delta_{mui}$ $+ cn(\zeta+1)(2c+1) \times \Delta_{div} + c^2n(\zeta+1) \times \Delta_{pow}$ $+ cn(\zeta+1)(c+1) \times \Delta_{sqrt}$
	MFMLE	$\{ \eta nc(c+1) \{ 3p^2+2p-1 + \frac{p(p-1)(2p-1)}{6} \} + \eta nc^2 \} \times \Delta_{add}$ $+ \eta nc(c+1) \{ 2p^2+3p+1 + \frac{p(p-1)}{2} + \frac{p(p-1)^2(2p+1)}{6} \} \times \Delta_{mui}$ $+ \{ \eta nc(c+1)(p+2) + \eta nc^2 + \eta nc \} \times \Delta_{div} + \eta nc^2 \times \Delta_{pow}$ $+ 2\eta nc(c+1) \times \Delta_{sqrt} + \eta nc(c+1) \times \Delta_{exp}$
	EFMLE	$\{ \eta nc(c+1) \{ 3p^2+2p-1 + \frac{p(p-1)(2p-1)}{6} \} \times \Delta_{add}$ $+ \eta nc(c+1) \{ 2p^2+3p+1 + \frac{p(p-1)}{2} + \frac{p(p-1)^2(2p+1)}{6} \} \times \Delta_{mui}$ $+ \{ \eta nc(c+1)(p+2) + \eta nc^2 + \eta nc \} \times \Delta_{div} + \eta nc^2 \times \Delta_{pow}$ $+ 2\eta nc(c+1) \times \Delta_{sqrt} + \eta nc(c+1) \times \Delta_{exp}$
Cluster Center Vector, V	MFCM	$2\zeta cnp \times \Delta_{add} + \zeta cnp \times \Delta_{mui} + \zeta cnp \times \Delta_{div} + 2\zeta cnp \times \Delta_{pow}$
	EFCM	$2\zeta cnp \times \Delta_{add} + 3\zeta cnp \times \Delta_{mui} + \zeta cnp \times \Delta_{div} + 2\zeta cnp \times \Delta_{pow}$
	MFMLE	$2\eta cnp \times \Delta_{add} + \eta cnp \times \Delta_{mui} + \eta cnp \times \Delta_{div} + 2\eta cnp \times \Delta_{pow}$
	EFMLE	$2\eta cnp \times \Delta_{add} + 3\eta cnp \times \Delta_{mui} + \eta cnp \times \Delta_{div} + 2\eta cnp \times \Delta_{pow}$
Termination Criteria, $\epsilon_{FCM}, \epsilon_{FMLE}$	MFCM	$2\zeta cn \times \Delta_{add} + \zeta cn \times \Delta_{mui} + \zeta cn \times \Delta_{sqrt}$
	EFCM	$2\zeta cn \times \Delta_{add} + \zeta cn \times \Delta_{mui} + \zeta cn \times \Delta_{sqrt}$
	MFMLE	$2\eta cn \times \Delta_{add} + \eta cn \times \Delta_{mui} + \eta cn \times \Delta_{sqrt}$
	EFMLE	$2\eta cn \times \Delta_{add} + \eta cn \times \Delta_{mui} + \eta cn \times \Delta_{sqrt}$
A Priori Proability, P	MFMLE	$2\eta cn \times \Delta_{add} + \eta c \times \Delta_{div}$
	EFMLE	$2\eta cn \times \Delta_{add} + \eta cn \times \Delta_{mui} + \eta c \times \Delta_{div}$
Fuzzy Covariance Matrix, F	MFMLE	$\eta nc(3p^2+1) \times \Delta_{add} + 2\eta ncp^2 \times \Delta_{mui} + \eta cp^2 \times \Delta_{div}$
	EFMLE	$\eta nc(3p^2+1) \times \Delta_{add} + 3\eta ncp^2 \times \Delta_{mui} + \eta cp^2 \times \Delta_{div}$

어 있는 MFCM 알고리즘, MFMLE 알고리즘과 확장된 퍼지 클러스터링 알고리즘에 포함되어 있는 EFCM 알고리즘, EFMLE 알고리즘의 실행시간을 $\Delta_{add}, \Delta_{mui}, \Delta_{div}, \Delta_{pow}, \Delta_{sqrt}, \Delta_{exp}, \zeta, \eta, c$ 그리고 $m=2$ 의 항으로 나타낸 것이다.

확장된 퍼지 클러스터링 알고리즘의 메모리 감소 효과를 최적 퍼지 클러스터링 알고리즘과 비교하기 위하여 표1의 모델을 사용하였다. 이를 위해 사용한 $\Delta_{add}=1450ns, \Delta_{mui}=1450ns, \Delta_{div}=2833ns, \Delta_{pow}=6250ns, \Delta_{sqrt}=1533ns, \Delta_{exp}=12167ns, p=1, \epsilon_{FCM}=0.005, \epsilon_{FMLE}=0.05$ 라고 가정하였으며 $\Delta_{add}, \Delta_{mui},$

$\Delta_{div}, \Delta_{pow}, \Delta_{sqrt}, \Delta_{exp}$ 의 값들은 16MHz의 MIPS R2030 RISC 컴퓨터 환경하에서 C프로그램을 이용하여 실수에 대한 계산시간을 10⁷번 만큼 실행시킨 후 구한 평균값이다.¹²⁾

표3은 표1의 데이터 모델에서 최적 퍼지 클러스터링 알고리즘과 확장된 퍼지 클러스터링 알고리즘의 실행시간을 보여주고 있으며 그림7은 이 표를 그래프로 나타낸다. 이 그래프에 의하면 데이터 차수, p가 1이고 전체 데이터 수에 대한 서로 다른 특징값을 갖는 데이터의 비, n'/n가 0.10이며 최적 클러스터수, c_{opt} 가 4일때 실행 시간의 감소 효과는 89.79%

이다.

한편 데이터 차수, p 와 전체 데이터 수에 대한 서로 다른 특징값을 갖는 데이터의 비, n'/n 가 고정되어 있을 때 최적 클러스터 수, c_{opt} 가 클수록 전체 실행 시간 중 소속 함수를 계산하기 위한 실행 시간의 비가 증가하기 때문에 실행 시간의 감소 효과는 데이터의 감소 효과에 접근한다.

다음장에서는 제안한 확장 퍼지 클러스터링 알고리즘을 이용하여 칼라 영상을 분할한다.

표 3. 모의 실험에서의 실행시간
Table 3. Execution time for an simulation

c	7	ξ	Execution Time(n)		Reduction (%)
			Optimal Fuzzy Clustering	Extended Fuzzy Clustering	
2	13	5	7,344,047	756,408	89.70
3	1	8	10,816,980	1,108,452	89.75
4	3	6	15,558,298	1,589,757	89.78
5	35	4	61,315,233	6,257,765	89.79

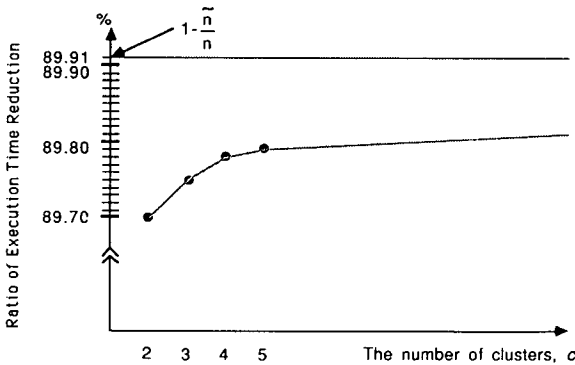


그림 7. 모의 실험의 실행시간 감소 효과
Fig. 7. The effective of execution time reduction for an simulation

V. 실험 및 검토

칼라 영상분할시 다양한 종류의 칼라 특징들이 Red, Green, 그리고 Blue에서 계산된다. 칼라 영상을 분할하기 위하여 가장 일반적으로 사용되는 칼라 영역은 RGB 칼라 영역과 OHTA 칼라 영역이다.^[24]

칼라 영상을 분할하기 위하여 RGB 칼라 영역을 사용하는 경우는 R, G 그리고 B의 특징 공간을 어떤 변환도 하지 않고 직접 사용한다. 그러나 R, G 그리고 B는 모두 밝기와 깊은 관계가 있을 뿐 아니

라 각각은 밝기에 있어서 상관 관계가 있기 때문에 무의미한 영상분할이 발생할 수 있다.^[2]

칼라 영상을 분할하기 위하여 OHTA 칼라 영역을 사용하는 경우는 R, G, 그리고 B의 변환에 의해 계산되어 지는 $I1 = (R+G+B)/3$, $I2' = (R-B)$, 그리고 $I3' = (2G \times R - B)/2$ 의 특징 공간을 사용한다. 이들 세개의 특징들의 순서는 의미가 있으며 많은 경우에 있어서 단지 처음 두개의 특징만을 사용하여서도 칼라 영상 분할시 좋은 결과를 얻을 수 있다. 또한 R, G, B 특징들로부터 이들 특징으로의 변환이 단순하다.^[2]

확장된 퍼지 클러스터링 알고리즘에 의한 영상 분할이 메모리와 실행시간에서 감소 효과가 있음을 증명하기 위하여 256×256 의 크기를 갖는 girl 칼라 영상을 사용하였다. 또한 주어진 영상을 몇개의 클러스터로 나누어야 하는 문제는 아직 일반화된 해결방법이 제안되어 있지 않고 주어진 상황에 따라 적절한 방법을 사용하고 있으므로⁽⁴⁾ 본 논문에서는 최적 클러스터 수, c_{opt} 를 9개로 가정하기로 한다.

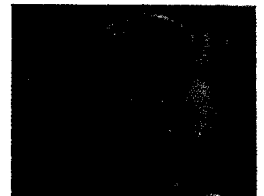
그림8은 girl 칼라 영상의 흑백 사진이다. 이들은 256 그레이 레벨을 갖는 Red, Green, 그리고 Blue 영상이다. 그림9와 그림10은 확장된 퍼지 클러스터링 알고리즘을 사용하여 girl 칼라 영상을 RGB 칼라 영역과 OHTA 칼라 영역에서 분할한 결과이다. 각각의 경우 (a)는 스카프, 꽃의 일부, 그리고 입술 부분이 분할 결



(a)



(b)



(c)

그림 8. Girl 칼라 영상
(a) red 영상 (b) green 영상 (c) blue 영상
Fig. 8. Girl color image.
(a) red image, (b) green image, (c) blue image.

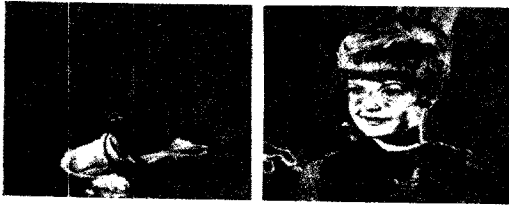


(a) (b)

그림 9. RGB 칼라 영역에서 girl 영상의 분할 결과 (c=9, m=2)

- (a) 스카프, 꽃의 일부, 그리고 입술에 대한 소속함수의 그레이 레벨 표현
- (b) 얼굴에 대한 소속함수의 그레이 레벨 표현

Fig. 9. Segementation result for girl image in RGB color space (c=9, m=2) gray level scale representation of fuzzy membership for (a) scarf, part of flower lip and, (b) face.



(a) (b)

그림 10. OHTA 칼라 영역에서 girl 영상의 분할 결과 (c=9, m=2)

- (a) 스카프, 꽃의 일부, 그리고 입술에 대한 소속함수의 그레이 레벨 표현
- (b) 얼굴에 대한 소속함수의 그레이 레벨 표현

Fig. 10. Segementation result for girl image in OHTA color space (c=9, m=2) gray level scale representation of fuzzy membership for (a) scarf, part of flower, lip and, (b) face.

과로 나타나고 (b)는 얼굴 부분이 분할 결과로 나타난다. 이 결과는 최적 퍼지 클러스터링 알고리즘에 의한 분할 결과와 같다.

그러나 최적 퍼지 클러스터링 알고리즘을 사용하여 girl 영상을 분할하고자 할 경우 RGB 칼라 영역과 OHTA 칼라 영역에서 모두 3차원의 데이터가 영상의 크기 만큼인 65,536개가 필요한 반면에 확장된 퍼지 클러스터링 알고리즘을 이용할 경우 RGB 칼

라 영역에서는 3차원의 데이터 32,519개와 1차원의 가중치 32,519개가 필요하고 OHTA 칼라 영역에서는 3차원의 데이터 29,082개와 1차원의 가중치 29,082개가 필요하다. 즉, 확장된 퍼지 클러스터링 알고리즘에서 전체 데이터 수에 대한 서로 다른 특징값을 갖는 데이터 수의 비 n'/n 은 RGB 칼라 영역의 경우 0.496이고 OHTA 칼라 영역의 경우 0.443 이다. 그림11은 클러스터 수, c의 변화에 따른 메모리 감소 효과를 나타내며 그림12는 클러스터의 수, c의 변화에 따른 실행시간 감소 효과를 나타낸다.

따라서 앞에서 가정한 최적 클러스터 수, c_{opt} 가 9 이므로 이에 대한 메모리 감소 효과는 식(26)에 의하여 각각 46.24%와 51.92%이고 표2에 의하여 실행

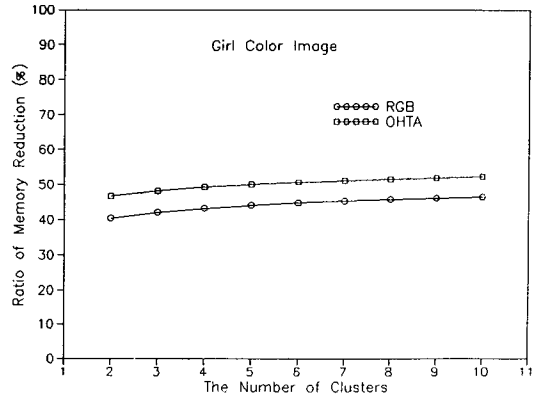


그림 11. Girl 칼라 영상에서의 메모리 감소 효과
Fig. 11. The effectiveness of memory reduction in girl color image.

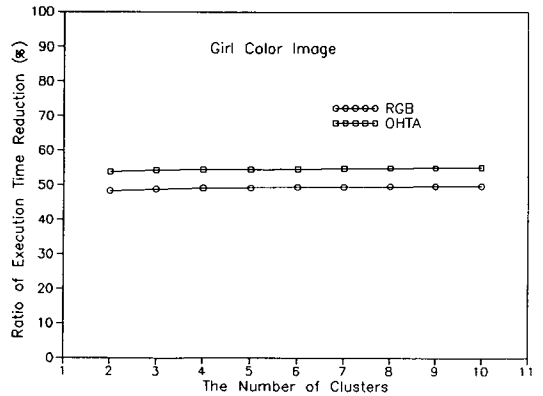


그림 12. Girl 칼라 영상에서의 실행시간 감소 효과
Fig. 12. The effectiveness of execution time reduction in girl color image.

시간 감소 효과는 각각 49.66%와 54.98%이다.

이러한 실험 결과를 일반화 하면 확장된 퍼지 클러스터링 알고리즘을 사용하여 칼라 영상을 분할할 경우 영상의 크기에 관계없이 전체 영상 데이터에 대한 서로 다른 특징값을 갖는 영상 데이터의 비, n'/n 이 0.833이고 최적 클러스터 수, c_{opt} 가 2일 경우에 최적 퍼지 클러스터링 알고리즘에 비하여 메모리와 실행 시간에서 최소의 동시 감소효과를 얻는다. 이 경우에 메모리 감소 효과는 0.04%이고 실행 시간 감소효과는 13.45%이다. 따라서 영상의 크기에 관계없이 전체 영상 데이터에 대한 서로 다른 특징값을 갖는 영상 데이터의 비, n'/n 이 0.833 보다 작고 최적 클러스터 수, c_{opt} 가 2보다 클 경우 메모리 감소효과와 실행 시간 감소 효과는 이들 값보다 크다.

한편, 다른 표준 영상에서 전체 영상 데이터에 대한 서로 다른 특징값을 갖는 영상 데이터의 비, n'/n 를 구해보면 lena 칼라 영상에서 RGB 칼라 영역의 경우 0.761이고 OHTA 영역의 경우 0.642이며 blonde 칼라 영상에서는 RGB 칼라 영역의 경우 0.478이고 OHTA 칼라 영역의 경우 0.396이다.

VI. 결 론

저자들은 확장된 퍼지 클러스터링 알고리즘을 제안하고 이 알고리즘이 자동 목표물 탐색과 다중 침투 탐색에 효과적으로 이용될 수 있음을 증명한 바 있다. 본 논문에서는 영상을 분할하고자할 때 영상 데이터를 직접 사용하는 대신 같은 특징값을 갖는 영상 데이터들의 수를 그 특징값의 가중치로 사용하는 확장된 퍼지 클러스터링 알고리즘을 사용할 경우 영상 자체의 특성을 변화시키지 않으면서 메모리와 실행 시간을 줄일 수 있을 뿐만 아니라 기존의 퍼지 클러스터링 알고리즘과 같은 분할 효과를 가짐을 입증하였다. 한편 이러한 접근 방법은 칼라 영상의 분할 방법과 같은 개념인 다중 주파수에 의한 리모트 센싱에도 효과적으로 이용되어질 수 있다.

參 考 文 獻

[1] J. Bezdek and J. Dunn, "Optimal puzzy partition: a heuristic for estimating the

parameters in a mixture of normal distributions," *IEEE Trans. Computer*, vol. C-24, pp.835-838, August 1975.

- [2] Yu-ichi Ohta, Takeo Kande, and Toshiyuki Sakai, "Color information for region segemntation," *Computer Graphics, and Image Processing*, vol. 13, pp. 222-241, 1980.
- [3] K. Fu and J. Mui, "A survey on image segmementation," *Pattern Recognition*, vol. 13, pp. 3-16, 1981.
- [4] T. Huntsberger, C. Jacobs and R. cannon, "Iterative fuzzy image segmentation," *Pattern Recognition*, vol. 12, pp. 131-138, 1985.
- [5] M. Trivedi and J. Bezdek, "Low-level of aerial images with fuzzy clustering," *IEEE Trans. Systems, Man, and Cybernetics*, vol. SMC-16, no. 4, pp. 589-598, July/August, 1986.
- [6] R. Johnson and D. Wichern, *Applied Multivariate Statistical Analysis*, Prentice-Hall, 1988.
- [7] I. Gath and A. Geva, "Unsupervised optimal clustering," *IEEE Trans. Pattern Analysis and machine Intelligence*, vol. PAMI-11, no. 7, pp. 773-781, July 1989.
- [8] H. Zimmermann, *Fuzzy Set Theory and its Applications*, Dordrecht, Netherlands: Kluwer-Nijhoff Publishing, 1984.
- [9] W. Pratt, *Digital Image Processing*, pp. 606-608, John Wiley & Sons, Inc.
- [10] 김수환, 강경진, 이태원, "확장된 퍼지 클러스터링 알고리즘을 이용한 자동 목표물 탐색" 대한 전자공학회 논문지 제28-B권, 제10호, pp. 842-851, 1991년 10월.
- [11] 김수환, 조창호, 강경진, 이태원, "확장된 퍼지 클러스터링 알고리즘을 이용한 다중 침투 검출" 대한전자공학회 논문집 제29-B권, 제1호, pp. 102-112, 1992년 1월.
- [12] RISC/os programmer's reference. MIPS computer system Inc., February 5, 1989.

 著 者 紹 介

金 秀 桓 (準會員) 第28卷 B編 第10號 參照
 현재 고려대학교 대학원 전
 자공학과 디지털시스템연구실
 연구원

李 太 遠 (正會員) 第18卷 B編 第10號 參照
 현재 고려대학교 전자공학과
 교수



姜 景 辰 (正會員) 第28卷 B編 第10號 參照
 현재 고려대학교 대학원 전
 자공학과 박사과정 재학중.