

# 깊이정보 생성을 위한 영상 분할에 관한 연구

임재성  
국방기술품질원 유도전자센터

## A study on image segmentation for depthmap generation

Jae Sung Lim

DTaQ(Defence agency for Technology and Quality)

**요약** 디스플레이 기기들이 고도화 되면서, 사용자의 목적에 부합하는 영상이 요구되어져 가고 있다. 따라서, 3D 디스플레이에서 필요하게 되는 깊이 정보가 요구될 때 디스플레이 기기들은 객체 기반의 영상 정보를 제공 할 수 있어야 한다. 따라서, 본 논문에서 깊이 정보 생성을 위한 히스토그램 기반의 영상분할 알고리즘을 제안한다. 기존의 K 군집 알고리즘에서 군집의 수를 파라미터화 하여, 영상에 적응적으로 군집 수를 결정할 수 없게 되는 한계를 지닌다. 또한, k 군집 알고리즘이 지니고 있던 지역 최소점에 빠져 영상 분할에 있어 과분할을 야기하는 지역 최소점에 빠지게 되는 경향이 있다. 반면에, 제안하는 알고리즘은 분할해야할 군집 선정에서 계산량을 고려하여 적응적으로 선택 가능할 수 있게 하는 히스토그램 기반의 알고리즘을 설계하여 적응적으로 선택 가능하게 하였다. 기존 알고리즘이 가지고 있었던 지역 최소점에 빠지지 않도록 방지하게 하여 결과 영상에서 객체 기반의 결과를 보여줄 수 있도록 설계 했다. 이 후 연결요소 알고리즘을 통해 과분할 요소를 제거 했다. 따라서, 제안하는 알고리즘은 객체 기반의 깊이 정보 결과를 보여 줄 뿐만 아니라, 벤치마크 방법에 비해 확률 랜드 인덱스, 분할 커버링 측면에서도 각각 벤치마크 방법에 비해 0.017, 0.051으로 향상된 결과치를 보여준다.

**Abstract** The advances in image display devices necessitate display images suitable for the user's purpose. The display devices should be able to provide object-based image information when a depthmap is required. In this paper, we represent the algorithm using a histogram-based image segmentation method for depthmap generation. In the conventional K-means clustering algorithm, the number of centroids is parameterized, so existing K-means algorithms cannot adaptively determine the number of clusters. Further, the problem of K-means algorithm tends to sink into the local minima, which causes over-segmentation. On the other hand, the proposed algorithm is adaptively able to select centroids and can stand on the basis of the histogram-based algorithm considering the amount of computational complexity. It is designed to show object-based results by preventing the existing algorithm from falling into the local minimum point. Finally, we remove the over-segmentation components through connected-component labeling algorithm. The results of proposed algorithm show object-based results and better segmentation results of 0.017 and 0.051, compared to the benchmark method in terms of Probabilistic Rand Index(PRI) and Segmentation Covering(SC), respectively.

**Keywords** : 3D display, Depthmap generation, Fast K-means based on histogram image segmentation, Image segmentation, Probabilistic rand index(PRI), Segmentation covering(SC)

### 1. 서론

최근 3D display 기술의 발달로 인해서 다양한 3D 영상이 제작되고 많은 관심을 끌고 있다. 이와 함께 3D

TV가 널리 보급되면서 3D 영상은 이제 우리의 생활과 매우 밀접한 display 형태로 바뀌어 가고 있다. 이에 따라서 3D 콘텐츠에 대한 요구가 크게 증대 되었고 3D 영상의 화질, 즉 입체감의 정확도 및 편안함, 입체 물체의

\*Corresponding Author : Jae Sung Lim(DTaQ)  
Tel: +82-10-4151-0210 email: jslim@dtaq.re.kr

Received July 12, 2017  
Accepted October 13, 2017

Revised (1st September 19, 2017, 2nd September 26, 2017)  
Published October 31, 2017

경계에서의 매끄러운 분할 등에 대한 요구도 함께 증대하고 있다. 3D 영상의 경우, 깊이 맵(depth map)생성을 통해서 객체간의 상대 깊이 정보를 도출하고 이에 따라서 입체감을 표현한다. 따라서, 영상 내에 존재하는 객체를 정확히 분류해 내고 이들 간의 깊이(depth)를 정확히 구분해 낸다는 것은 우수한 화질 생성에 중요한 영향을 미치게 된다. 정확한 깊이 맵 생성을 위해서는 동일한 객체 내에서 동일한 깊이값을 가질 수 있도록 하는 것이 매우 중요하다. 가장 기본적인 초기 깊이 맵 생성을 위한 스테레오 매칭(stereo matching)의 경우 위와 같은 객체별 깊이 값을 동일하게 제공하기에는 그 정확도가 매우 떨어진다. 따라서, 입력 영상을 의미 있는 다수의 픽셀 집합으로 나누는 작업을 의미하는 영상 분할(image segmentation)의 결과 Fig.1.은 이러한 스테레오 매칭을 통한 깊이 맵 생성에서 매우 유용하게 사용 될 수 있다.



(a) Original image (b) Segmented image  
**Fig. 1.** Example of image segmentation

영상 분할을 기반으로 한 깊이 맵 생성의 경우, 영상 분할의 정확도가 깊이 맵의 정확도에 큰 영향을 줄 수 있고, 이는 곧 3D 영상의 화질을 좌우 하게 된다. 따라서, 효과적인 영상 분할은 3D 영상 화질을 향상 시키는 데 큰 역할을 수행 하게 된다. 이외에도 영상 분할의 결과는 내용기반 영상 검색, 이미지 내에서의 객체 검출, 객체 인식, 의료관련 객체 분할등의 다양한 분야에도 적용되어 사용될 수 있다. 따라서, 우수한 영상 분할을 위한 다양한 연구들이 진행되어 왔다.

기존의 영상 분할의 경우 계산량이 높고 그에 비례하여 하드웨어 리소스 사용으로 실시간 처리 및 일반 TV 시스템에서 사용하기에 많은 비용이 요구되는 단점이 존재 했다. 따라서, 본 연구에서는 기존 영상 분할 알고리즘들을 살펴본 뒤 이와 비교하여 계산량을 줄이면서 향상된 영상 분할 결과를 제공할 수 있는 알고리즘을 제안하고자 한다.

## 2. 관련 연구

### 2.1 Conventional image segmentation algorithms

군집화를 이용한 알고리즘은 특징이 유사한 데이터를 같은 범주로 지정하고 각 범주를 통하여 분할된 영상을 표현하는 방법으로 가장 널리 쓰이는 알고리즘은 K-means clustering[1]과 mean shift 알고리즘[2]이다.

K-means 군집화의 경우 최초 K개의 중심을 지정한다. 이후 각 중심에 데이터를 종속시켜 군집을 형성하면서 비용 함수를 최소화하는 군집을 생성한다. 비용함수는 식(1)의 형태로 나타난다.

$$E = \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2 \quad (1)$$

여기서  $\mu_i$ 는  $i$ 번째 군집의 중심값이다. 이후 생성된 클러스터의 중심값을 재 생성하고 라벨링을 수행하며 그 이후 중심의 위치를 정하는 형태로 중심의 위치가 수렴할 때까지 군집화를 수행한다. 이 방법은 군집화 수행 시 K 값을 설정하여야 하고 초기 중심의 위치에 따라 분할 결과가 다양하게 변화하는 단점을 지니기 때문에 안정적인 결과를 얻기 힘들다.

다음으로, mean shift 알고리즘의 경우는 탐색 윈도우를 설정하여 윈도우 내의 포함되는 데이터의 평균으로 윈도우를 이동시켜 나가면서 윈도우 이동이 없을 때까지 평균값을 찾아 나가는 알고리즘이다. 이를 영상 분할에 적용하게 되면 먼저, 다수의 탐색 윈도우를 이미지 전체에 균일하게 위치시킨다. 이후, 각 윈도우를 평균의 위치로 이동시켜 가고 최종적으로 같은 평균으로 수렴된 윈도우들은 윈도우가 이동해 오면서 거친 데이터들은 같은 군집으로 합병되어 최종 분할을 수행하게 된다. K-means 군집화에 비해서 K값 설정과 같은 변수 설정은 없지만, 탐색 윈도우의 초기 위치에 따라 성능이 변화할 수 있고 계산량이 높은 단점을 가지고 있다.

그래프 컷(Graph cut)을 이용한 영상 분할 알고리즘은 이미지를 각 픽셀들이 연결된 가중치 그래프( $G = \langle V, E \rangle$ )로 나타내고 각 엣지(E)를 끊어가면서 분할을 수행하는 알고리즘을 의미한다. 다시 말해, 이미지 Z의 각 픽셀 Zi를 노드(V)로 나타내고 각 노드간의 관계를 엣지로 나타낸다. 이후 식(2)에서 표현되는 에너지를

함수를 최소화 할 수 있는 최적의 컷(cut)을 찾아내는 것이 그래프 컷의 수행과정이다.

$$E_n(Z) = \sum_{i=1} E_1(z_i) + \lambda_1 \sum_{i,j \in E} E_2(z_i, z_j) \quad (2)$$

여기서  $E_1$ 은 노드  $Z_i$ 가 전경과 배경과 같은 영역 모델에 지정되었을때 배정된 영역 모델과  $Z_i$ 와의 유사도 차이를 나타내고,  $E_2$ 의 경우  $Z_i$ 와  $Z_j$ 의 연결성 혹은 유사도를 나타내는 것으로  $Z_i$ 와  $Z_j$ 사이의 연결을 끊을 때 발생하는 에너지를 나타낸다. 따라서, 이 에너지 함수를 최소화 하는 그래프 컷을 찾는 것은 최적의 영상 분할과 연결되는 것이다.

하지만, 에너지 함수가 엣지 수에 비례함에 따라 본래 목적으로 하는 이상적인 것이 이루어지지 않는 경우가 생길 수 있다.

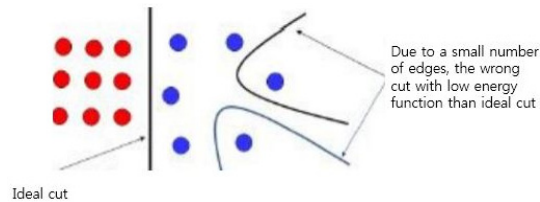


Fig. 2. Example of incorrect graph cut result

이에 그림 Fig. 2와 같은 잘못된 컷 결과를 예방하기 위해 정규화 컷(normalized cut)[3]이 제안되었다. 정규화 컷에서는 아래 식(3)에서와 같이 A, B로 그래프를 컷하여 나눌 때, 발생하는 에너지 함수에 각 A, B그룹이 가지고 있는 각 그룹의 엣지 총합을 이용하여 엣지 수가 매우 작은 그룹으로 컷이 이루어지는 것을 방지한다.

$$N_{cut}(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \quad (3)$$

여기서  $N_{cut}$ 은 A그룹과 B그룹의 차이를 나타내고,  $cut(A, B)$ 는 A그룹과 B그룹의 연결 엣지를 나타낸다. 마지막으로  $assoc(A, V)$ 와  $assoc(B, V)$ 는 각각 A와 B그룹에 연결된 모든 엣지 값의 합을 나타낸다. 앞서 살펴본 군집화 및 그래프 기반 영상 분할 알고리즘의 경우 계산량 및 하드웨어 리소스를 고려했을때 실시간 처리에 적합하지 않기 때문에 이를 그대로 실시간으로 요구되는 깊이 정보 생성을 위한 영상 분할에 이용하기에는 어려

움이 예상된다.

앞서 밝혔듯이 기존 K-means 군집화, 그래프 기반 영상 분할의 경우 비용 측면에서 TV 시스템에 쓰기에는 부적합한 계산량과 하드웨어 리소스가 크게 소요된다는 점과 앞서 소개된 각 방법이 가지는 한계점을 볼 수 있었다. 이에 본 연구에서는 계산 및 하드웨어 비용 면에서 우수한 영상 분할 알고리즘을 제안하고자 한다.

### 3. 제안 알고리즘

앞선 연구들을 통해서 알 수 있듯이 영상 분할의 경우 프레임 전체(global)정보와 지역(local)정보를 적절히 이용하는 것이 매우 중요하다. 또한, 기존 방식에서 프레임 정보를 모두 저장해서 이용하기에는 프레임 메모리 및 비용 측면에서 문제가 있으며, 데이터를 양자화해서 처리하지 않을 경우 계산량 크기가 커져 실시간 처리에는 적합하지가 않다. 따라서 연구의 목표는 라인버퍼(line buffer)를 9개 이하로 사용하면서 실시간 처리가 가능한 계산량을 지닌 영상 분할 알고리즘의 설계에 부합하면서도 우수한 성능을 위해서 다음과 같은 설계전략을 사용하고자 한다.

프레임 버퍼 없이 전체 정보를 이용하기 위해서 프레임 정보를 히스토그램(histogram)으로 치환하여 사용하되 픽셀(pixel)정보를 적절히 양자화하여 계산량을 감소시킨다. 히스토그램을 양자화하여 얻은 전체정보는 다음 프레임이 입력으로 들어 올때 지역정보와 함께 사용되어 최종 결과를 도출하게 된다. 이를 통하여 우수한 영상 분할 결과를 도출 할 수 있도록 하였다. 최종 결과는 픽셀 단위의 라벨링을 통하여 분할 결과를 제공하게 된다.

Fig. 3은 전체 알고리즘의 동작을 흐름도(flow chart)로 나타낸 것이다.

간단히 동작을 요약하면, N-1번째 프레임의 RGB 정보를 CIELab 컬러 공간으로 변환한 후 일정 간격으로 양자화한 결과를 이용하여 3D 히스토그램으로 생성한다. 이후 3D 히스토그램의 최고점을 이용하여 군집화를 수행하여 각 컬러에 해당하는 컬러 인덱스(color index)를 도출한다. 이후 N번째 프레임을 처리할 때 이전 프레임에서 도출된 컬러 인덱스와 지역적 영역에서의 이미지 디테일과 컬러 차이를 이용하여 변형된 8 connected component labeling을 적용하여 분할된 이미지를 도출 한다.

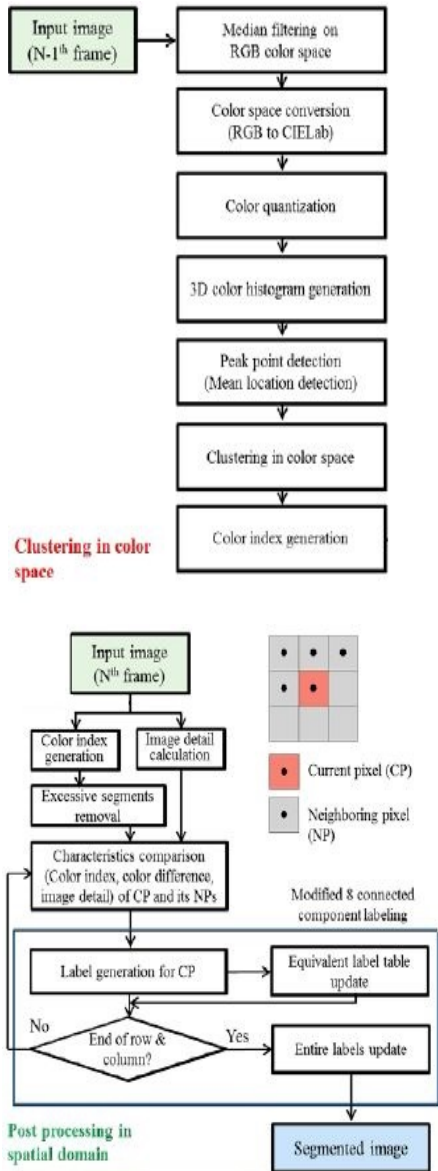


Fig. 3. Flow chart of the proposed method

### 3.1 Clustering in color space

제안하는 영상 분할 알고리즘은 가장 먼저 입력 이미지의 컬러 공간을 일정 개수의 군으로 군집화 하여 1차적인 분할을 수행한다.

#### 3.1.1 Median filtering on RGB color space

효과적인 군집화를 위해서 median filtering을 RGB평면에 적용하여 존재할 수 있는 의미없는 노이즈 성분을

제거한다.

#### 3.1.2 Color space conversion and quantization

이 후 필터링된 RGB 컬러 공간을 CIE Lab color 공간으로 변환한 후 각 L, a, b평면을 각각 일정 크기의 level(10, 10, 10 or 10, 12, 12)로 양자화를 수행한다. 이를 통해서 전체 계산량의 숫자와 히스토그램 bin수를 줄이고자 의도하였다. 또한, L, a, b의 level수가 매우 작기 때문에 CIE Lab 변환시 Look Up Table(LUT)를 이용하는데 도움이 될 수 있다. LUT를 이용함으로써 CIE Lab으로 컬러 공간을 변환하는데 필요한 복잡도가 높은 연산들을 피할 수 있다. 일련의 양자화까지의 과정을 그림으로 나타내면 아래 Fig.4와 같다.

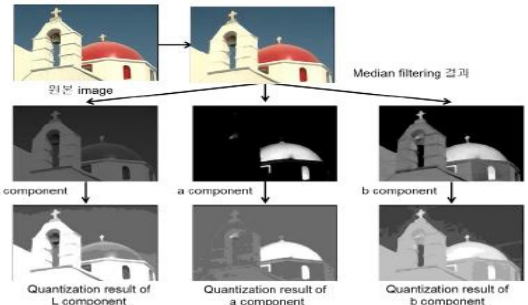


Fig. 4. The procedures of color space conversion

#### 3.1.3 3D color histogram generation

양자화된 L, a, b 컬러 성분에 대해서 3차원 히스토그램을 생성한다. L, a, b의 level수가 (10,10,10)일 경우에는 총 히스토그램 bin개수는 1000개가 된다.

#### 3.1.4 Initial mean location detection for clustering

이 후 수행 될 군집화를 위해서 초기 평균의 위치와 개수를 3D 히스토그램의 최고지점들을 이용하여 결정하게 된다. 컬러 공간에서 각각 컬러 K의 군집으로 분류하게 된다고 할 때, 각 K개의 군집이 가우시안분포(Gaussian distribution)를 가진다고 가정하여, 3D 히스토그램에서의 모든 방향에서의 큰 값을 가지는 고점을 군집의 중심으로 초기 위치를 도출하게 된다. 아래 Fig. 5에서 볼 수 있듯이 고점값을 이용한 평균값 설정은 간편하면서도 가장 가까운 평균의 bin을 군집화하는 군집화 기반 영상 분할에 매우 적합한 형태를 띈다.

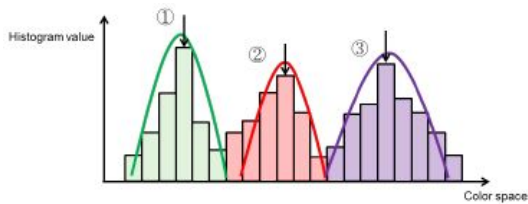


Fig. 5. Example of the peak selection from 1D histogram

3D 히스토그램에서의 고점은 아래 Fig. 6.에서와 같이 총 26개의 방향과 비교하여 가장 큰 히스토그램값을 가지는 지점이 고점 후보가 된다.



Fig. 6. Peak selection from 3D histogram

26개 방향을 비교하여 고점으로 지정된 bin의 히스토그램값이 주어진 이미지 전체 픽셀의 0.1% 이상일 경우만 평균으로 설정하여 노이즈 성분 혹은 무시할만한 작은 군집의 생성을 방지한다. 이렇게 입력 프레임의 3D 히스토그램을 분석하여 고점값을 도출하면 고점의 개수가 컬러 공간에서의 군집 수가 된다. 만약 고점의 개수가 2개 이하인 경우에는 under segmentation되었을 가능성이 있다고 판단하여 추가적인 중심(평균)을 도출하게 된다. 실제로 아래 Fig. 7.에서 2D 히스토그램의 예를 들어 보면 한쪽 y 차원에서는 히스토그램의 형태가 2개의 가우시안 형태를 띄고 있지만, 다른 한쪽 x 차원에서는 한쪽 클러스터에서 뻗어 나온 성분에 의해서 다른 클러스터 후보의 평균이 고점이 되지 못하는 것을 볼 수 있다. 이러한 경우는 3D에서 더욱 많을 것이다. 따라서, 이러한 경우 25개 방향에서 최대값을 가지는 컬러 인덱스를 기존 고점에 추가하여 평균으로 이용한다.

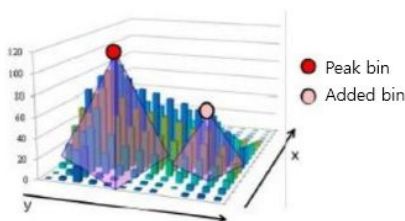


Fig. 7. Example of under segmentation in 2D histogram

하지만, 추가되는 평균의 경우, 평균으로 도출된 두 bin간의 유클리디안 거리가 1이하인 경우와 평균으로 추가된 후보 bin의 히스토그램값이 주어진 이미지 전체 픽셀의 0.1%이하일 경우에는 추가 평균을 설정하지 않아서 실제 2개 이하로 분할 될 확률이 있는 이미지의 경우는 평균의 개수를 유지할 수 있도록 한다.

### 3.1.5 Clustering in color space

초기 평균이 결정된 이후에는 식(4)와 같이 각 히스토그램 bin을 가장 가까운 평균으로 배정하여 군집화를 수행한다.

$$C_{idx}(i) = \operatorname{argmin} \Delta E^2(i, \text{mean}(j)), w.r.t. j \quad (4)$$

$$\Delta E^2(x, y) = (L(x) - L(y))^2 + (a(x) - a(y))^2 + (b(x) - b(y))^2 \quad (5)$$

여기서  $i$ 는 히스토그램 bin 인덱스를  $j$ 는 평균의 인덱스를 나타낸다.



Fig. 8. Example of clustering result

실제 군집화의 결과를 나타내면 Fig. 8.와 같다. 히스토그램 고점값을 이용한 군집화의 경우 K-means 군집화와 같이 반복적으로 평균값을 이동시켜 수렴할 때까지 반복할 필요 없이 한 번의 군집지정으로 우수한 분할 결과를 도출 할 수 있다. 컬러 공간에서의 결과를 도출하면 Table. 1.과 같다.

Table 1. Example result of clustering

Color space (L,a,b)	Color index (C <sub>idx</sub> )	Color space (L,a,b)	Color index (C <sub>idx</sub> )
(1,1,1)	1	(3,1,10)	3
(1,1,2)	1	(4,1,9)	3
(1,2,1)	1	(4,2,10)	3
...	...	(5,2,2)	4
(1,9,12)	2	...	...
(1,9,13)	2	(10,10,9)	k
...	...	(10,10,10)	k

표 1.에서 볼 수 있듯이 L, a, b 컬러 공간에서 각 컬러 성분은 자신과 유사한 성질을 가지는 군집형태가 되어 하나의 컬러 인덱스를 가지게 된다.

### 3.1.6 Post processing in spatial domain

컬러 인덱스로 군집화된 결과만으로 영상분할을 수행할 경우 컬러 인덱스는 공간정보를 포함하고 있지 않기 때문에 공간적으로 떨어진 객체들도 컬러만 동일하다면 같은 객체로 분류될 수 있다. 따라서, 이러한 공간적인 정보를 이용하여 픽셀 간의 연결성을 도출할 수 있는 8 connected component labeling 알고리즘을 이용하여 최종 분할 결과를 도출 한다. 이 과정에서 컬러 인덱스가 가지는 단점을 보완할 수 있도록 추가적인 정보를 도출하여 최종 픽셀간의 연결을 결정하게 된다.

### 3.1.7 Removing excessive segments

Connected component labeling 알고리즘을 적용하기 전에 공간적으로 과도하게 분할된 경우에는 식 (6)과 같이 인접된 객체로 병합되어 과분할(over-segmentation)을 방지하고자 한다.

$$C_{idx}(x,y) = \operatorname{argmax} H(z), \text{ w.r.t. } j \quad (6)$$

$$H(z) = \sum_{\substack{C_{idx}(x',y')=z \\ (x',y') \in M(x,y)}} 1 \quad (7)$$

여기서 x, y는 픽셀 인덱스를 나타내고 M은 x,y를 중심으로 하는 9×9 윈도우를 나타낸다.



Fig. 9. Example of excessive segments removal

Fig. 9.는 과도하게 분할된 성분을 제거한 결과를 나타냈다. 실제로 의미 없이 경계에서 생성된 과분할 요소들이 제거된 것을 볼 수 있다.

### 3.1.8 Modified 8 connected component labeling based on color difference and image details

이 후 컬러 인덱스를 기반으로 8 connected component labeling을 적용하여 공간적으로 인접하면서 같은 성질의 픽셀을 같은 분할객체로 지정하게 된다. 본래 connected component labeling의 경우 이진 영상을 입력으로 이용하여 전경에만 라벨링을 수행하나 제안하는 알고리즘에서는 모든 픽셀에 대해서 라벨을 부여해야 하므로 아래와 같이 그 형태를 변형하였다.

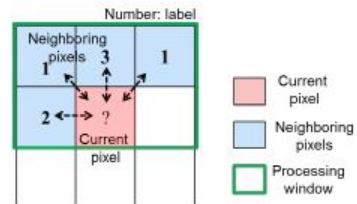


Fig. 10. Processing window of modified 8 connected component labeling

Fig. 10은 픽셀 단위로 컨볼루션을 수행하면서 현재 픽셀에 라벨링을 부여할 때 사용되는 2×3윈도우를 나타냈다. 여기서 기본적으로 설정된 수는 컬러 인덱스를 나타낸다고 할 수 있다. 만약, 컬러 인덱스만을 사용하여 connected component labeling을 수행한다면 Fig. 10에서 컬러 인덱스의 동일 여부만 판단하여 연결성을 판단하면 되나, 컬러 인덱스의 경우 컬러 인덱스 생성시 수행한 양자화 과정으로 인해 과분할이 발생하는 경우가 존재한다. 즉 컬러 인덱스는 공간정보를 고려하지 않고 생성되었기 때문에 점진적인 컬러 변화의 경우에 아래 Fig. 11과 같이 같은 객체로 설정되어야 할 영역이 분할 될 수 있다. 또한 de-focused 배경에도 깊이 맵 생성시 분할되지 않아야 하므로 이러한 특성을 반영하여야 한다.



Fig. 11. Examples of clustering results(over-segmentation)

이에 제안하는 알고리즘은 픽셀간의 연결성을 컬러 인덱스 이외에 현 픽셀과 주변 픽셀 각각을 중심으로 한 5×5픽셀 패치 간의  $\Delta E$ 의 합( $\Delta E_{block}$ )과 라플라시안 필터링(Laplacian filtering)결과를 임계값(Th=8)하여 이진 값으로 도출한 이미지 디테일( $I_{detail}$ ) Fig. 12을 이용한다.



Fig. 12. Idetail calculated by Laplacian filtering

컬러 인덱스( $C_{idx}$ )를 이용하여 1차적으로 현재 픽셀과 주변 픽셀간의 연결성을 나타내는 마스킹 행렬(masking matrix,  $M_{masking}(i,j)$ )를 도출하면 아래 식과 같이 나타낼 수 있다.

$$M_{masking}(i,j) = \begin{cases} 1, & C_{idx}(i) = C_{idx}(j) \\ 0, & otherwise \end{cases} \quad (8)$$

여기서  $i$ 는 현재 픽셀 인덱스,  $j$ 는 주변 픽셀 인덱스를 나타낸다. 컬러 인덱스의 특성으로 인한 과분할을 방지하기 위해서 인접 픽셀간의 컬러 인덱스( $C_{idx}(i)$ )가 다른 경우에도 Fig. 13.과 같이 현재 픽셀과 주변픽셀간의 컬러 차이가 매우 작고( $\Delta E_{block} < 0.05$ ), 영역간의 경계가 없는 경우(현재 픽셀을 중심으로 5×5 픽셀 패치내의  $I_{detail}$ 이 모두 0인경우)동일 객체로 병합하기 위해서 연결성을 부여한다.

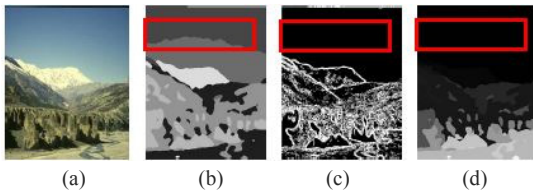


Fig. 13. Segment merge by modified 8 connected component labeling based on color difference and image details.((a)input image (b)Cidx (c)Idetail (d)Segment result using Mmasking)

이를 수식으로 나타내면 식(9)와 같다.

$$M_{masking}(i,j) = \begin{cases} 1, & C_{idx}(i) = C_{idx}(j) \text{ or } (I_{detailsum}(i) = 0 \text{ and } \Delta E_{block}(i,j) < 1) \\ 0, & otherwise \end{cases} \quad (9)$$

$$I_{detailsum}(i) = \sum_{x=-2}^2 \sum_{y=-2}^2 I_{detail}(r+x, c+y) \quad (10)$$

이를 이용하여 마스킹 매트릭스의 생성 예를 살펴보면 Fig. 14.와 같다.

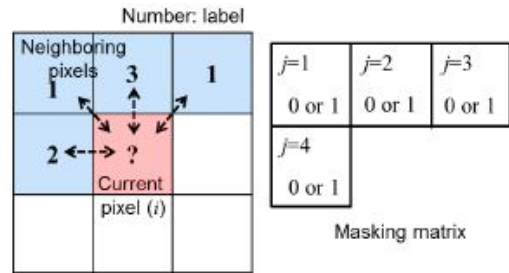


Fig. 14. Example of masking matrix( $M_{masking}$ )

인접 픽셀 라벨과 마스킹 매트릭스와의 컨볼루션을 통해 연결성이 없는 픽셀의 라벨을 0으로 마스킹하고 이후 0을 제외한 인접 라벨값들 중 가장 작은 값의 라벨을 현재 픽셀에 부여하는 방식으로 현재픽셀의 라벨을 정해 나간다.

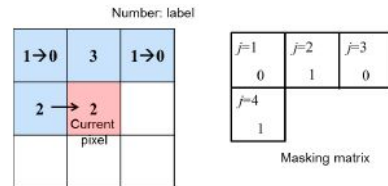


Fig. 15. Labeling at a current pixel in the modified 8 connected component labeling

현재 수행하고 있는 8 connected component labeling의 경우 2개의 line buffer만을 이용하여 순차적으로 라벨링을 수행하는 알고리즘이다. 따라서, 프레임 전체를 참조하여 최종 연결성 도출을 위해서 동치 라벨표의 생성이 필요하다. 즉 프로세싱시에는 연결성을 확인 할 수 없었던 라벨끼리의 연결성을 프레임 전체 프로세싱이 끝난 후에 확인하기 위함이다. 이를 통해서 프레임 버퍼를 사용하지 않고 동치 라벨 표를 활용하여 전체 연결성을 반영한 객체들을 도출 할 수 있게 된다.

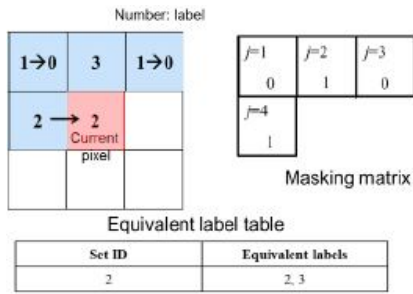


Fig. 16. Equivalent label table configuration

Fig. 16에서와 같이 현재 픽셀의 라벨은 2로 결정되지만, 2,3은 연결성이 존재하므로 최종 라벨 결정시에 2,3을 동일한 객체로 결정하기 위해서는 위와 같은 동치라벨표를 생성한다. 최종적으로 동치라벨표를 이용하여 라벨들을 정리하여 최종 결과를 도출 하게 된다. 결과 예시를 보면 Fig.17과 같이 최종 결과를 확인해 볼 수 있다.

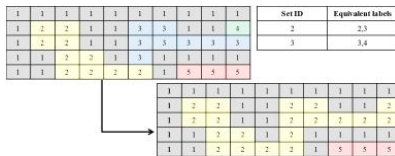


Fig. 17. Final result of 8 connected component labeling

### 4. 실험 및 분석

영상 분할 결과를 평가하기 위해서 버클리에서 제공하고 있는 Berkeley Segmentation Dataset 500(BSDS) [4]을 이용하였다. BSDS에는 피 실험자 5-6명이 매뉴얼하게 작성한 영상 분할 결과를 ground-truth로 이용하였다.

#### 4.1.1 Objective evaluations metrics for image segmentation quality

영상 분할의 질을 평가하기 위한 객관적인 방법으로는 현재 관련 분야에서 널리 쓰이고 있는 probabilistic rand index(PRI)[5]와 segmentation covering(SC)[6]를 사용하였다.

PRI의 경우 아래 식(11)으로 표현될 수 있다.

$$PRI(S, G_k) = \frac{1}{T} \sum_{i < j} [c_{ij} p_{ij} + (1 - c_{ij})(1 - p_{ij})] \quad (11)$$

여기서  $S$ 는 분할결과,  $G_k$ 는 ground-truth 분할결과,  $T$ 는 총 픽셀쌍의 수(total number of pixel pairs),  $c_{ij}$ 는 픽셀  $i$ 와  $j$ 가 같은 라벨을 갖는 이벤트,  $p_{ij}$ 는  $c_{ij}$ 의 확률을 나타낸다. 즉 PRI는 입력 영상에서 가능한 모든 경우의 픽셀 쌍을 만들었을 때 이들 픽셀 쌍의 동일 객체 여부가 ground-truth와 영상 분할 결과가 얼마나 일치하는가를 나타낸 것이라고 할 수 있겠다.

다음으로 SC의 경우는 식(12)으로 표현될 수 있다.

$$o(R, R') = \frac{|R \cap R'|}{|R \cup R'|}, C(S' \rightarrow S) = \frac{1}{N} \sum_{R \in S} |R| \max_{R' \in S'} o(R, R') \quad (12)$$

여기서  $S$ 는 ground-truth 분할결과,  $S'$ 는 분할결과,  $R$ 은  $S$ 내의 분할결과(segment in  $S$ ),  $R'$ 은  $S'$ 내의 분할 결과(segment in  $S'$ ),  $N$ 은 전체 픽셀 개수를 나타낸다. 즉 ground-truth에서 하나의 합에 대비하여 공통 영역이 가장 큰 객체를 찾아 내어, 이 값들을 모두 더해가는 방식으로 ground-truth의 모든 객체에대해서 반복한 값이 SC라고 하겠다.

이들 PRI와 SC를 이용하여 객관적인 영상 분할 퀄리티 평가 결과를 도출 하였고 이들 평가 방법의 계산량이 매우 높은 관계로 ground-truth와 영상 분할 결과 모두 1/4크기로 다운스케일링 하여 진행하였다. 실험에 사용되는 벤치마크 방법은 비교적 계산량이 낮고 프레임 버퍼를 사용하지 않고 우수한 성능을 제공하는 fast image segmentation based on K-means clustering with histograms in HSV color space[7]를 사용하였다.

#### 4.1.2 Evaluations for image segmentation quality(color index)

벤치마크 알고리즘과 공정한 평가를 위해서는 제안하는 알고리즘에서 추가적으로 수행한 modified 8 connected component labeling based on color difference and image details을 적용하기 전까지의 결과를 비교하는 것이 올바른 방법이다. 따라서, color index 결과를 이용하여 일반적인 8 connected component를 양쪽 방법에 모두 적용하여 평가하였다.



Table. 2. PRI and SC results of benchmark and proposed algorithms

Division		Benchmark			Proposed	
		Th*=30 (1478bins)	Th*=30 (1478bins)	Th*=40 (1478bins)	(10,12, 10)* (1000 bins)	(10,12, 12)* (1440 bins)
BSDS (500)	Avg. PRI	0.757	0.756	0.747	0.756	0.764
	Avg. SC	0.470	0.467	0.472	0.504	0.521

표 2.는 벤치마크와 제안하는 방법의 PRI와 SC 결과를 나타낸 표이다. Th\*는 최대최소 알고리즘의 임계값 (Llevel, alevel, blevel) \*은 L, a, b의 양자화 레벨 수, (#bins)는 총 히스토그램 bin수를 나타낸다. 최대최소 알고리즘은 앞선 절에서 간단히 소개했던 방법으로 K-means 군집화를 수행하기 위해서 최초 중심(평균)을 도출하는 방식 중 하나이다. 벤치마크에서는 최대최소 알고리즘을 이용하여 컬러 히스토그램의 최대값 위치를 첫 번째 평균으로 지정하고 각 히스토그램 bin을 가장 가까운 평균에 군집화 하고 이후 지정된 평균에서 가장 먼 곳의 컬러 위치를 추가 평균으로 지정해 간다. 추가할 평균의 위치와 기존의 평균들과의 최대 거리가 Th\*값보다 작을 경우 동작을 마무리하게 된다. 벤치마크의 경우 다양한 Th\*를 적용하여 결과를 도출하였고, 제안하는 알고리즘의 경우는 bin의 수를 1000개와 1440개의 2가지 경우로 L, a, b의 레벨 수를 조정하여 실험을 진행하였다. 총 500장의 BSDS에 대한 평균 PRI와 SC의 값을 살펴보게 되면 제안하는 방법이 더 적은 히스토그램 bin을 사용하면서도 우수한 결과를 나타내는 것을 볼 수 있다. 실제 대표적인 영상들에 대한 결과를 살펴보면 아래와 같이 과분할 요소가 보이지 않으며 객체 단위로 군집화 된 것을 확인 할 수 있다.



Fig. 18. Comparison of image segmentation(1)

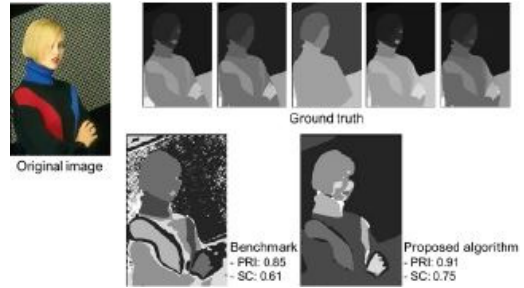


Fig. 19. Comparison of image segmentation(2)

#### 4.1.3 Evaluations for effectiveness of modified 8 connected component labeling based on color difference and image details

마지막으로 컬러 인덱스의 단점을 보완하기 위해서 적용하였던 modified 8 connected component labeling의 효과를 평가하였다. 그 결과를 Fig. 20, Fig. 21의 예를 들어서 나타내었다. 결과를 살펴보면 알 수 있듯이 컬러 인덱스가 가지고 있었던 단점으로 인해 과분할 되었던 영역들이 병합되어 우수한 결과를 보이는 것을 볼 수 있다. 다만, 실제 영상 분할의 경우 피 실험자들은 단순히 다르게 보이는 객체들은 다른 객체로 나눠 놓았기 때문에 깊이 맵 생성에 적합한 ground-truth와는 거리가 있는 영상들로 존재하여 PRI나 SC값이 떨어지는 것을 볼 수 있다.

따라서, Fig. 21와 같은 경우 2번째 ground-truth와 실제 깊이 맵 생성에 더 적합한 영상을 도출하여 결과를 도출하는 것도 깊이 맵 생성을 위한 영상 분할 평가 방법이 될 수 있을 것으로 예상된다.

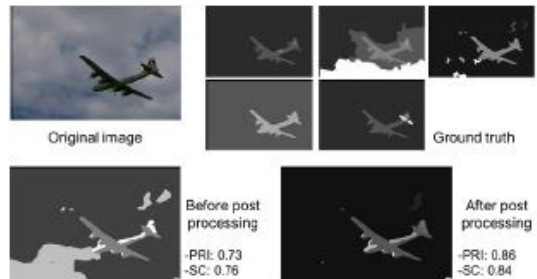


Fig. 20. Effectiveness of modified 8 connected component labeling based on color difference and image detail

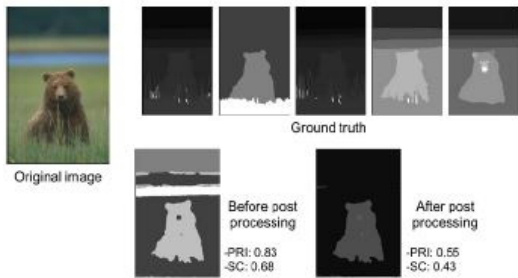


Fig. 21. Effectiveness of modified 8 connected component labeling based on color difference and image detail(2)

## 5. 결론

이 논문에서, 우리는 깊이 맵을 생성하기 위한 영상분할 알고리즘을 제시하였다. 제안하는 방법은 군집을 이루는 파라미터를 적응적으로 할당하여, 계산량을 고려한 의미있는 객체로 분할하는 알고리즘을 제시하였다. 실험 결과에서 볼 수 있듯이 제안한 방법은 타 벤치마크 방법에 비해 과분할 결과를 억제하면서 군집화 하는 것을 PRI와 SC의 지표로 검증할 수 있어 이 후 분할 요소에 대한 정보를 활용 시 쓰임이 있을 것으로 판단된다.

## References

- [1] K. Alsabti, et. al, "An efficient K-means clustering algorithm", *Electrical Engineering and Computer Science*, 1998.
- [2] D. Comaniciu, P.Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603-619, 2002.  
DOI: <https://doi.org/10.1109/34.1000236>
- [3] J.Shi, et. al, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, Aug. 2000.  
DOI: <https://doi.org/10.1109/34.868688>
- [4] Berkeley Segmentation Dataset and Benchmark : <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds>
- [5] R.Unnikrishnan, et. al, "Toward objective evaluation of image segmentation algorithms," *Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 929-944, Jun. 2007.  
DOI: <https://doi.org/10.1109/TPAMI.2007.1046>
- [6] T. malisiewicz, A. A. Efros, "Improving spatial support for objects via multiple segmentations," *Proc. British Machine Vision Conf.*, 2007.

DOI: <https://doi.org/10.5244/C.21.55>

- [7] T. chen, et. al, "Fast image segmentation based on K-means clustering with histograms in HSV color space," *Multimedia Signal Processing*, pp. 322-325, Oct. 2008.

DOI: <https://doi.org/10.1109/MMSP.2008.4665097>

임재성(Jae Sung Lim)

[정회원]



- 2012년 2월 : 한양대학교 공과대학 전자통신공학부
- 2014년 2월 : 포항공과대학교 전자전기공학과 석사
- 2015년 9월 ~ 현재 : 국방기술품질원(DTaQ) 연구원

<관심분야>

영상, 비디오신호처리, 국방신뢰성공학