

Watershed 알고리즘을 통한 non-rigid object의 효율적인 영역 분할 방식에 관한 연구

이인재*, 김용호*, 김중규*, 전준근**, 이명호**, 안치득**
성균관대학교 전기전자컴퓨터공학부*, 한국전자통신연구소 방송기술연구부**

Effective segmentation of non-rigid object based on watershed algorithm

In-Jae Lee*, Yong-Ho Kim*, Joong-Kyu Kim*, Jun-Geun Jeon**, Myoung-Ho Lee**, Chieteuk Ahn**
The School of Electrical and Computer Science Engineering, SKKU*,
Broadcasting Technology Department ETRI**

Abstract

본 논문에서는 구름이나 연기와 같은 non-rigid object에 대한 영역 분할 방식에 대해 연구하였다.

Non-rigid object의 효과적인 영역 분할을 위해서 object의 윤곽선을 정확히 파악해 낼 수 있는 장점을 가진 watershed 알고리즘을 사용하였다. 하지만 이 알고리즘은 object가 많은 영역으로 분할되는 oversegmentation 현상이 발생하여 본 논문에서는 pre, post-processing을 통해 이 oversegmentation 현상을 극복하고자 하였다.

Pre-processing에서는 noise를 제거하고 영상을 단순화하면서 정확한 gradient magnitude를 구할 수 있는 방법에 대해서, post-processing에서는 통계적인 분석을 통한 region merging을 이용하여 object를 최적화 상태로 찾아줄 수 있는 방법에 대하여 연구하였다.

I. 서론

본 논문에서 연구 대상이 되는 non-rigid object는 형태가 변하는 object를 의미하는데 특히 구름, 연기 등과 같이 일정한 형태가 없어 시간에 따른 형태 변화를 예측하기 어려운 object를 대상으로 하였다.

현재 많이 연구되어지고 있는 알고리즘 중 seeded region growing과 같이 원영상에서 영역을 병합하는

방법은 threshold를 적절히 하지 못하였을 경우 시작영역에서 나아가지를 못하거나 범람해 버리는 undersegmentation 현상이 발생할 수 있다. 이러한 현상은 non-rigid object의 경우 다른 영상에 비하여 boundary에서의 intensity 차이가 작고, 내부에서의 변화가 심한 특징을 가지기 때문에 특히 더 심하게 나타날 수 있다. 따라서 non-rigid object를 보다 정확히 segmentation해주기 위해 watershed 알고리즘을 사용하였다. Watershed 알고리즘은 gradient 영상 내에서 최소영역을 찾아낸 후 화소값을 증가시켜 가면서 분할된 영역들을 얻어내는 방법이다. 이 방법을 적용할 경우에는 영역을 먼저 분할한 후에 merging하는 과정을 거치기 때문에 undersegmentation은 방지할 수 있다. 하지만 반대로 결과 영상에서 oversegmentation 현상이 나타나는 단점이 있어 pre-processing과 post-processing을 통하여 이를 줄여 주었다.

본 논문에서는 먼저 pre-processing 단계로 noise를 줄이고, mathematical morphology 기법을 이용하여 영상을 단순화시킨 후 gradient magnitude를 구하였다. Post-processing에서는 color정보와 통계적인 분석을 적용한 region merging을 이용하여 object를 최적의 상태로 찾아줄 수 있도록 하였다.

II. 영상의 단순화 및 gradient 계산

Pre-processing과 post-processing의 목적은

watershed로 인해 발생하는 oversegmentation을 줄이는데 있다. 그러나 post-processing에서 사용하는 region merging의 경우 영역의 개수가 많아질수록 처리 시간이 길어지므로 먼저 영역을 단순화시켜줄 필요가 있다. 또한 좋은 watershed 결과를 얻기 위해서는 edge 부분을 부각시킬 필요도 있다. 따라서 pre-processing은 크게 두 가지 단계로 나누어 줄 수가 있다.

1. Noise 제거 및 영상의 단순화

Watershed 알고리즘을 원영상에 적용하였을 때 원하지 않는 미세 영역까지 segmentation 되는 문제점이 나타난다. 따라서 먼저 filtering 과정을 통해 noise를 줄여주고 morphological filter를 사용하여 영상을 단순화시켜줌으로써 object의 edge는 보존해 주면서 불필요한 영역은 제거시켜 줄 수 있다. 영상의 noise 성분을 줄여 주기 위해 일반적으로 linear filter, median filter, Gaussian 등 여러 가지 방법들이 사용되어지고 있으며 가능한 edge의 손실없이 원영상을 단순화시켜 주기 위해서는 morphology 기법이 많이 사용되어지고 있다. 이 mathematical morphology 기법의 기본 연산자로는 dilation, erosion이 있고 더 나아가 opening, closing 연산자도 보편적으로 많이 사용되고 있다.

입력 영상을 f 라 하고 structure element를 k 라 했을 때 dilation과 erosion에 관한 식은 다음과 같다.

$$f \oplus k(x) = \max_{x-z \in F, z \in K} \{f(x-z) + k(z)\} \quad (1)$$

$$f \ominus k(x) = \max_{x+z \in F, z \in K} \{f(x+z) - k(z)\} \quad (2)$$

이 식을 이용하여 opening과 closing 또한 다음과 같이 표현할 수 있다.

$$f \circ k = (f \ominus k) \oplus k \quad (3)$$

$$f \cdot k = (f \oplus k) \ominus k \quad (4)$$

Opening 연산자는 structuring element를 이용하여 주위에 비해 intensity가 밝은 미세한 부분들을 제거해주는 반면 closing 연산자는 어두운 intensity를 가진 미세한 부분들을 제거시킨다. 즉 이 연산들을 통해 미세한 부분에 영향을 주어 미세 영역에 대한 과분할 현상을 줄여줄 수 있다.

2. Gradient 계산

일반적으로 gray-level이 빨리 변하는 지점들이 영역들의 edge 부분과 일치하므로 watershed 알고리즘의 입력 영상으로 gradient 영상을 사용하면 segmentation을 하는데 있어 보다 효율적이다. 따라서 첫 번째 단계에 의해 단순화된 영상에 대한 gradient 영상을 구해 이것을 watershed에 적용시키기 위한 입력 영상으로 사용해준다. 일반적으로 Sobel, Prewitt, Gaussian, Gaussian derivative 등을 이용해 gradient 영상을 구해줄 수 있다.

III. Watershed 알고리즘과 효율적인 region merging 기법

1. Watershed 알고리즘

Watershed 알고리즘에서 영상의 data는 gray level 값을 높이로 하는 지형학적인 표면으로서 다루어진다. 가령 가장 낮은 바닥에서부터 물을 채워간다고 가정했을 때 물은 가장 낮은 높이에 있는 minima에서부터 출발하여 영상의 catchment basin을 점차적으로 채워나간다. 그리고 두 개의 다른 minima로부터 채워지는 물이 서로 합쳐지려고 하는 지점에는 dam을 세워놓는다. 이 과정이 끝난 후 각각의 catchment basins은 dam에 의해 둘러싸여지며 이 dam들의 집합은 watershed line과 일치한다. 이 과정을 통해 우리는 입력 영상을 여러 개의 다른 catchment basins으로 나누어 줄 수 있다.

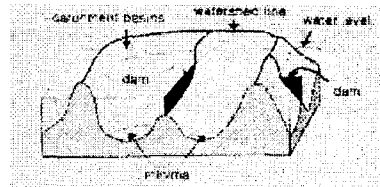


그림 1 Watershed 알고리즘의 개념도

2. 효율적인 region merging

Region merging의 최종 목적은 하나의 object에 하나의 label을 할당시키는 데 있다. 이 과정에서 판단 기준이 되는 각 영역의 유사성을 결정하기 위해 영역들 간의 mean, variance 등 통계적인 방법들을 이용한다.

일반적으로 gray level값을 사용해 merging 시키기 위한 조건들을 설정해 주는데, RGB 혹은 YCbCr 등 다양한 color 정보를 조건 설정시 적절하게 첨가시켜 준

다면 보다 효율적인 region merging이 가능하다. 그래서 유사성을 판단할 때 이웃하는 영역간의 RGB나 YCbCr 등의 mean값의 차를 이용하면 보다 효율적인 결과를 얻을 수 있다.

Hierarchical Stepwise Optimization(HSWO) 알고리즘은 stepwise criterion $C_{i,j}$ 의 값이 가장 작은 두 개의 영역을 선택해 merging시키는 최적화 기법이다. 원영상 $f(x, y)$ 에서 segmentation된 영역들을 S_i 라 하고 이 영역들이 평균값 μ_i 를 가진다고 할 때 각 영역들의 근사화 에러는 다음과 같이 표현될 수 있다.

$$H(S_i) = \sum_{(x,y) \in S_i} (f(x,y) - \mu_i)^2 \quad (5)$$

HSWO 알고리즘에서 영역들간의 유사성을 판단하기 위해 사용되어지는 stepwise criterion $C_{i,j}$ 은 다음과 같이 정의할 수 있다.

$$C_{i,j} = H(S_i \cup S_j) - H(S_i) - H(S_j) \quad (6)$$

이것을 다시 정리해 보면

$$C_{i,j} = \frac{N_i \cdot N_j}{N_i + N_j} (\mu_i - \mu_j)^2 \quad (7)$$

여기서 N_i 와 N_j 는 각 영역 S_i , S_j 의 크기이다. 이 $C_{i,j}$ 를 계산한 후 이 값이 가장 작은 이웃하는 영역을 찾아 merging시켜준다. 이 과정을 반복해 나가면서 object를 최적의 상태로 찾아준다.

Color의 mean값을 이용하여 merging한 방법은 처리 시간도 짧고 효율적이지만 object를 최적의 상태로 segmentation 시키는데 있어서 한계가 있다. 반면 HSWO 알고리즘의 경우는 mean 값 외에 영역의 크기 정보도 반영해 주므로 object를 좀더 최적의 상태로 찾아줄 수 있지만 대신 처리 시간이 많이 소요된다. 따라서 두 가지 방법을 접목시켜 먼저 color의 mean값 차를 이용하여 입력영상을 그에 상응하는 최적의 상태로 segmentation해 준 후 여기에 HSWO 알고리즘을 적용해 주면 보다 정확하고 효율적인 segmentation 결과를 얻을 수 있다.

IV. 실험 결과 및 고찰

제안한 방법의 효율성을 확인하기 위해 176×144 크

기의 Y,Cb,Cr 구름 영상을 가지고 실험하였다. 그림 2-(b)는 pre-processing 단계를 거친 결과 영상으로서 Gaussian filter를 이용하여 noise를 제거시켜 주고 closing을 이용하여 미세한 영역을 줄여준 후 sobel edge detector를 통해 gradient magnitude를 구한 결과이다. 그림 2-(c)는 그림 2-(b)를 입력영상으로 하여 watershed 알고리즘을 적용시켜준 결과이다. 그림 2-(d)는 원영상에 watershed line을 흰색으로 표시한 결과 영상으로, 먼저 mean 값의 차이가 Y의 경우 5, Cb,Cr의 경우에는 10보다 작은 조건을 만족하면서 서

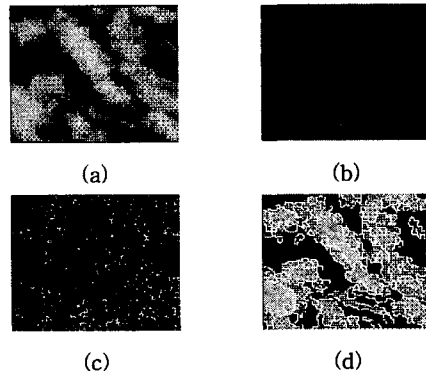


그림 2. 구름 영상에 대한 image segmentation 실험 (a)구름 원영상 (b)경사 영상 (c)Watershed 알고리즘을 적용한 결과 영상(613 regions) (d)Region merging한 결과 영상(Y=5, Cb=Cr=10; 156 regions; 160 iterations)

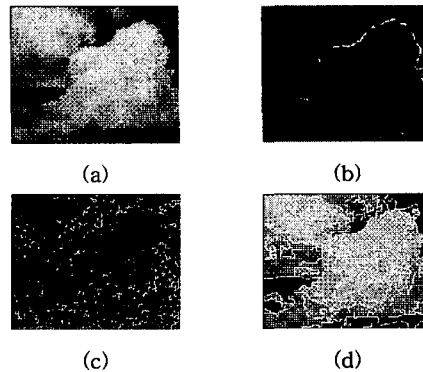


그림 3. 구름 영상에 대한 image segmentation 실험 (a)구름 원영상 (b)경사 영상 (c)Watershed 알고리즘을 적용한 결과 영상(761 regions) (d)Region merging한 결과 영상(Y=5, Cb=Cr=10; 62 regions; 250 iterations)

로 이웃하는 영역들을 merging시켜주었다. 그 후 160번의 iteration을 통해 stepwise criterion C_{ij} 를 만족하는 영역들을 찾아 merging시켜 주었다. 그림 3은 다른 구름 영상에 대해 같은 방법으로 실험한 결과로서 그림 3-(d)의 경우에는 250번의 iteration을 통해 merging시켜 주었다.

그림 2의 경우 watershed 알고리즘을 적용한 결과 object가 613개의 영역들로 segmentation되었고 이것을 color값을 이용하여 merging시켜 주면 316개의 영역으로, 여기에 HSWO 알고리즘을 적용해 주면 156개의 영역으로 감소됨을 알 수 있었다.

그림 3의 경우에는 watershed 알고리즘을 적용했을 때 761개의 영역으로 segmentation 되었던 것이 두 단계의 merging 후에는 62개의 영역으로 현저히 감소하였다. 또한 원영상과 결과 영상을 비교하였을 때 중요한 edge 부분의 손실 없이 효율적으로 oversegmentation 현상을 극복하였음을 알 수 있었다.

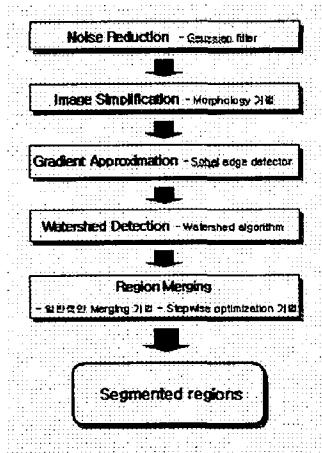


그림 4 제안된 segmentation의 흐름도

V. 결론

본 논문에서는 구름과 같이 형태가 일정하지 않은 non-rigid object의 정지영상에 대하여 기존의 방법보다 정확하면서 효율적인 segmentation 방법에 대해 연구하였다. Pre-processing 과정에서는 영상의 noise를 제거, 단순화시키고 좀 더 정확한 edge를 찾아내 이를 입력 영상으로 사용하여 watershed 알고리즘을 적용해주었

다. Post-processing 과정에서는 color정보, 영역의 크기 그리고 영역간의 mean 값의 차이를 반영하여 차이가 작은 것부터 merging할 수 있도록 하였으며 또한 동적 메모리 할당과 배열을 이용함으로써 속도와 메모리를 줄일 수 있도록 하였다. 이 과정을 통해 중요한 edge의 손실 없이 oversegmentation 현상을 효율적으로 극복해 보다 정확한 결과를 얻을 수 있었다. 이런 non-rigid object에 대한 segmentation 기술은 위성영상 분석 등의 기상분야, 혈류 분석과 속도 측정 및 피부병의 병위 진행 정도 파악 등의 의료분야, 적외선 감시 시스템 등의 방위 분야, 대기 오염 측정 등의 환경분야에 다방면으로 이용될 수 있다. 또한 정지영상에 그치지 않고 본 연구 결과를 토대로 더 빠르고 효율적인 방안을 연구하여 나아가 동영상에도 적용시킬 수 있을 것이다.

참고문헌

- [1] Luc Vincent and Pierre Soille, "Watersheds in Digital Spaces : An Efficient Algorithm Based on Immersion Simulations", IEEE Trans. Pattern Anal. Machine Intell., Vol. 13, No. 6, pp. 583-597, 1991
- [2] Kostas Haris, Serafim N. Efstratiadis, "Hybrid Image Segmentation using Watersheds and Fast Region Merging", IEEE Trans. Image Processing, Vol. 7, No. 12, pp. 1684-1699, 1998
- [3] Jean-Marie Beaulieu and Morris Goldberg, "Hierarchy in Picture Segmentation : A Stepwise Optimization Approach", IEEE Trans. Pattern Anal. Machine Intell., Vol. 11, No. 2, pp. 150-163, 1989
- [4] Munchuri Kim, Jae Gark Choi, "A VOP Generation Tool: Automatic Segmentation of Moving Objects in Image Sequences Based on Spatio-Temporal Information", IEEE Trans. Circuits Syst. Video Technol., Vol. 9, No. 8, pp. 1216-1226, 1999
- [5] Milan Sonka, Vaclav Hlavac, Roger Boyle, "Image Processing, Analysis, and Machine Vision", Brooks/Cole Publishing Company, 1999
- [6] Ramesh Jain, Rangachar Kasturi, Brian G. Schunck, "Machine Vision", McGraw-Hill Inc. 1995