

# 복부 CT 영상에서 빠른 폐 분할을 위한 그래픽 하드웨어 기반 레벨 셋 기법

박성진, 홍헬렌

{sjpark, hihong}@cglab.snu.ac.kr

Hardware-based Level Set Method for Fast Lung Segmentation on CT Abdomen Image

SeongJin Park, Helen Hong

## 요 약

본 논문에서는 복부 CT 영상에서 폐 부위를 빠르게 분할하기 위하여 그래픽 하드웨어를 사용한 레벨 셋 기법을 제안한다. 제안방법은 다음과 같이 세 단계로 구성된다. 첫째, 레벨 셋 기법을 그래픽 하드웨어로 효율적으로 구현하기 위하여 초기 레벨 셋 값 설정과 설정된 레벨 셋 값을 텍스처메모리에 저장한다. 둘째, 레벨 셋 기법의 가장 중요한 부분인 속도함수를 그래픽 하드웨어의 빠른 연산을 이용하여 계산하고, 레벨 셋 값을 갱신한다. 셋째, 갱신된 레벨 셋 값을 통하여 제로-레벨 셋을 찾는다. 본 논문에서는 제안 방법을 평가하기 위하여 일련의 복부 CT 영상을 사용하며, 육안평가 및 수행시간 면에서 기존 소프트웨어 기반 레벨 셋 기법과 비교분석한다. 실험결과 본 제안방법은 소프트웨어 기반 레벨 셋 기법과 분할결과를 동일하게 유지하면서 평균 9배 빠르게 폐 부위를 분할하였다.

## 1. 서 론

의료영상에서 인체 내 특정 관심부위를 추출하는 것은 질환 진단 및 치료를 위해 매우 필수적인 단계이다. 또한, 3차원 모델링이나 가상화 등의 작업에 앞서 처리되어야 하는 중요한 과정이다. 이를 위하여 임계값 기법과 같은 화소기반 방법, 영역확장법과 같은 영역-기반 방법, 경계선 검출과 같은 고전적인 방법들이 이용된다. 이와같은 고전적인 방법들은 지역적인 정보만을 사용하기 때문에 잡음에 민감하고, 불명확한 경계를 생성하며, 전처리나 후처리가 필요하다. 반면 레벨 셋(Level Set) 방법은 잡음이나 틈(hole) 등 비정규적인 특성들로부터 영향을 극복할 수 있을 뿐 아니라 코너 부분을 효과적으로 분할할 수 있고, 초기 곡선의 위치 및 형태에 의존적이지 않게 위상학적 변화(topological changes)가 가능하며, 임의의 차원으로 확장이 용이하여 2차원 단면 영상 뿐 아니라 3차원 볼륨데이터에 대한 분할이 가능하다 [1]. 그러나 레벨 셋 기법은 관심부위 주변 뿐 아니라 영상 전체 화소에 대하여 계산되어 처리시간이 많이 소요되므로 300~500장의 의료영상에서 특정 관심부위를 정확하고 빠르게 분할하기 위해서는 소프트웨어 기반의 레벨 셋 기법에는 한계가 있다.

일반적으로 기존 레벨 셋 기법은 front를 확장시키거나 축소시키는 방법으로 관심부위를 분할하는데 이를 위하여 우선, 초기 레벨 셋 값을 설정하여 front부분은 0의 레벨 값을 할당하고, front의 외부는 양수의 레벨 값을, 내부는 음수의 레벨 값을 각각 할당한다. 그리고 속도함수를 정의하여 front를 확장 또는 축소시키도록 한다. 이 때 정의되는 속도함수에서 영상의 특징이나 곡률같은 기하학적 특징이 사용된다. 마지막으로 정의된 속도함수를 각 화소에 적용하여 각 화소의 레벨 셋 값을 갱신하고, 0의 레벨 값을 갖는 부분을 찾아서 front를 다시 그려준다. front가 분할하려는 관심부위의 경계에

다다르면 확장이나 축소를 종료한다.

기존 레벨 셋 기법의 속도를 개선하기 위하여 Sethian[2] 등은 한 방향으로의 front 전파를 이용한 빠른 전진(Fast Marching) 방법을 제안함으로써 처리 속도를 단축시켰고, Adalsteinnsson[3]은 front로부터의 거리 계산을 통해 협대역(narrow band)을 구축함으로써 처리 속도를 단축시켰다. 이 방법은 각 시간마다 front 주위 몇 개 화소만을 계산에 고려하고 갱신시켜주는 방법으로 기존 방법에 비하여 계산량이 현저히 줄어들긴 했지만 front가 전진해 가면서 새로운 협대역을 구축해야 하는 과정이 발생하고 새로운 협대역을 구축하는 과정에서 복잡한 알고리즘을 이용하여야 하며, 새로 생성된 front로부터 거리 계산하는데 있어서 오류가 발생할 수 있는 문제점이 있다. Lefohn[4]의 경우 그래픽 하드웨어를 이용하여 3차원 볼륨데이터를 가상화하는 방법을 제안하였다. 이 방법은 볼륨데이터가 그래픽 하드웨어의 메모리보다 크기 때문에 메모리 관리 방법을 제시하였으나, 상대적으로 메모리 접근시간에 따른 속도 저하가 발생한다.

본 논문에서는 복부 CT 영상에서 폐 부위를 정확하고 빠르게 분할하기 위하여 그래픽 하드웨어를 이용한 레벨 셋 기법을 제안함으로써 기존의 소프트웨어 기반 레벨 셋 기법의 한계를 극복하고자 한다.

## 2. 그래픽 하드웨어를 이용한 레벨-셋 기법

본 논문에서는 그래픽 하드웨어를 사용하여 레벨 셋 기법을 구현하기 위하여 DirectX9에서 지원하는 셰이더(shader)언어를 사용한다. 또한 본 논문에서 세 개의 텍스처메모리 input, main, temp를 사용한다. input 텍스처와 main 텍스처는 영상의 밝기값(intensity)과 레벨 셋 값을 저장하고, temp 텍스처는 속도함수를 적용해서 갱신된 레벨 셋 값을 저장한다. 그림 1은 본 논문에서

제안한 방법의 파이프라인이다.

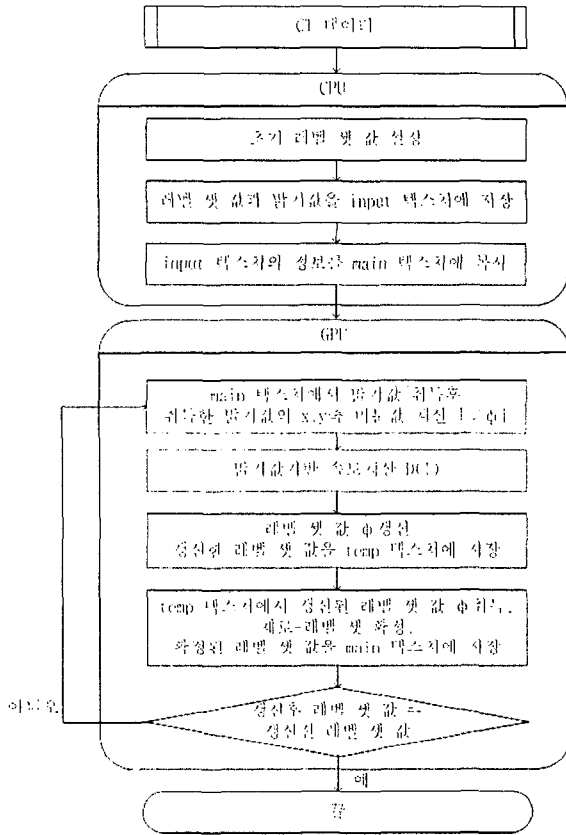


그림 1. 그래픽스 하드웨어를 이용한 레벨 셋 기법 구현단계

2.1 초기 레벨 셋 값 설정

input에 레벨 셋 값과 영상의 밝기값을 저장한다. 이때 input의 RGB채널에는 영상의 밝기값을 저장하고, A채널에 레벨 셋 값을 저장한다.

일반적으로 레벨 셋 값은 front부분을 제로-레벨 셋(zero level set)으로 하여 0의 값을 할당하고, front의 내부에 음수값, 외부에 양수값을 할당한다. 그러나 부호가 없는 자료형의 값을 취급하는 텍스처메모리에 저장하기 위해서 음수를 사용할 수 없으므로 0에서 255사이의 값을 할당해야한다. 이 때, front부분에 127값을 할당하고, 내부에 0값을, 외부에 255값을 할당하고 이를 input에 저장한다. input에 저장되어있는 레벨 셋 값을 렌더 타겟(Render Target) 텍스처인 main에 복사한다.

2.2 속도함수를 이용한 레벨 셋 값 계산

레벨 셋 값은 (식 1)을 통하여 갱신한다.

$$\phi = \phi - |\nabla\phi| \cdot D(I), \quad (1)$$

이 때,  $\phi$ 는 레벨 셋 값을 나타내고,  $|\nabla\phi|$ 는 레벨 셋 값의 x,y축 미분값 크기를 나타내며,  $D(I)$ 는 밝기값기반속도함수를 나타낸다. 따라서 레벨 셋 값의 x,y축 미분값

크기와 밝기값기반 속도함수의 곱을 통해 front의 수축과 팽창여부를 결정한다. 밝기값기반 속도함수는 (식 2)를 통하여 표현한다.

$$D(I) = \epsilon - |I - T|, \quad (2)$$

이 때,  $T$ 는 관심영역의 밝기값의 평균을,  $\epsilon$ 은 표준편차를 나타낸다.

그림 2는  $\epsilon$ 과  $T$ 에 따른 front의 수축과 팽창여부를 표현한 것이다. 그림과 같이 front의 밝기값이  $T - \epsilon$ 과  $T + \epsilon$ 사이라면 front는 팽창하게 되고, 그 외의 영역에서 front는 수축하게 된다.

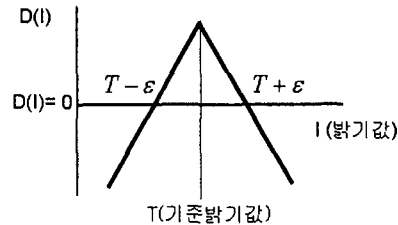


그림 2. 파라미터에 의한 front의 밝기값기반 속도함수

그래픽 하드웨어로 속도함수를 구현하기 위하여 우선 밝기값의 x,y축 미분값을 계산한다. 이를 위하여 버텍스 셰이더(Vertex Shader)에서 이웃하는 4개의 좌표를 계산하고, 픽셀 셰이더(Pixel Shader)에서 계산된 4개의 좌표로 이웃하는 레벨 셋 값을 읽어 중앙수치미분법을 사용하여 x,y축의 미분값을 계산한다. 그리고 main에서 현 좌표의 레벨 셋 값을 읽고  $T$ 와  $\epsilon$ 을 결정해서 속도함수를 계산한다. 계산된 속도함수를 (식 1)에 적용하여 레벨 셋 값을 갱신하고, 이를 렌더타겟 텍스처인 temp에 저장한다.

2.3 제로-레벨 셋 확정

속도함수를 적용하는 과정에서 갱신된 레벨 셋 값은 기정의한 0,127,255 이외의 값을 갖는 경우가 생긴다. 이 때, front가 소멸되지 않게 하려면 이러한 값이 기정의한 값을 갖도록 조정한다. 그림 3은 영교차(zero crossing)를 사용해서 제로-레벨 셋을 확정하고, 0,127,255의 값을 갖도록 조정하는 방법을 표현한 것이다.

123	128	129	135	155	127	255	255	255	255
109	125	129	140	150	0	127	255	255	255
120	125	126	125	128	0	0	127	127	255
123	124	126	126	132	127	127	127	127	255
128	129	130	128	135	255	255	255	255	255

(a) 제로-레벨 셋 확정전 (b) 제로-레벨 셋 확정후  
그림 3. 제로-레벨 셋 확정방법

우선 127보다 작은 값은 0으로 127보다 큰 값은 255로 할당한다. 그리고 (식 3)을 적용해서 제로-레벨 셋을 확정한다.

$$f(\phi(x,y)) = \begin{cases} 127, & \text{if } \phi = 0, \max(\phi(N)) = 255 \\ \phi & \end{cases} \quad (3)$$

이 때,  $f(\phi(x,y))$ 는 확정된 레벨 셋 값이고,  $\phi$ 는 확정되기 전의 레벨 셋 값을 0과 255값으로 분류한 값이다.  $\phi(N)$ 은 4개의 이웃하는 레벨 셋 값을 나타낸다.

그래픽 하드웨어로 위의 과정을 구현하기 위하여 버텍스 셰이더에서 4개의 이웃하는 좌표를 계산하고, 픽셀 셰이더에서 계산된 좌표로 이웃하는 레벨 셋 값을 획득한다. 이웃하는 좌표의 레벨 셋 값과 현좌표의 레벨 셋 값을 비교해서 그림 3과 같이 현 좌표의 제로-레벨 셋 여부를 확정한다. 그 결과를 main에 저장한다.

### 3. 실험 및 결과분석

본 실험은 Intel Pentium 4 CPU 2.6GHz와 1.0GB메모리를 장착한 PC에서 수행하였다. 그래픽 하드웨어는 ATI Radeon 9600이고, 그래픽 하드웨어의 메모리는 256MB이었다. 실험에 사용한 데이터는 512x512인 일련의 폐 영상 데이터를 사용했다.

하드웨어를 이용한 레벨 셋 기법의 영상분할 방법과 육안평가 및 계산시간 비교를 위해 동일조건 하에서 소프트웨어를 이용한 레벨 셋 기법의 영상분할 방법을 사용했다. 폐를 분할하기 위해서 파라미터는  $T$ 는 60,  $\epsilon$ 은 30의 값을 각각 주어서 실험하였다. 표 1은 그래픽 하드웨어를 이용한 제안방법과 소프트웨어를 이용한 방법의 수행시간을 비교한 것이다. 이 때, 실험과정에서 초기 제로-레벨 셋을 수동 설정하였기 때문에 각 방법에 따라 반복횟수에 차이가 발생한다.

표 1. 레벨 셋 기법 구현방법에 따른 수행시간 비교

영상	구현방법	반복횟수	총 수행시간 단위(초)	단위시간당 반복횟수 단위(반복수/초)
A	소프트웨어	187	9.922	18.85
	제안 방법	189	1.093	172.92
B	소프트웨어	237	12.125	19.55
	제안 방법	228	1.328	171.69
C	소프트웨어	310	16.515	18.77
	제안 방법	315	1.828	172.32

총 수행시간을 기준으로 그래픽 하드웨어를 이용해서 영상분할한 결과가 약 9배 빨랐다. 그림 4는 본 논문에서

제안한 방법과 소프트웨어로 구현한 방법의 결과영상을 비교한 것으로 분할결과에는 차이가 없음을 알 수 있다.

### 4. 결론

본 논문에서는 영상 내에서 특정 관심부위를 분할하기 위해 그래픽 하드웨어를 이용한 레벨 셋 기법을 사용하는 방법을 제안했다. 우선, 초기 레벨 셋 값 설정과 설정된 레벨 셋 값을 텍스처메모리에 저장하는 방법을 제시했다. 레벨 셋 기법의 가장 중요한 부분인 속도함수를 계산하고 레벨 셋 값을 갱신하는 방법을 제시했다. 갱신된 레벨 셋 값을 제로-레벨 셋을 확정함으로써 제로-레벨 셋은 127, 내부는 0, 외부는 255의 값을 갖는 원칙을 유지하였다. 제안한 방법으로 영상을 분할했을 때, 소프트웨어로 구현한 것과 똑같은 결과 영상을 얻었고, 속도는 9배정도 빨랐다. 앞으로 그래픽 하드웨어를 이용해서 레벨 셋 기법을 삼차원 볼륨데이터의 분할에도 적용할 예정이다. 이때, 볼륨렌더링과 연계시켜 구현함으로써 가상화도 가능하도록 할 예정이다.

### 5. 참고문헌

- [1] S.Osher and J.A.Sethian, Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulation. Journal of Computational Physics, vol. 79, pp. 12-49, 1988
- [2] J.A.Sethian, A fast marching level set method for monotonically advancing fronts. Proc. Natl. Acad. Sci. USA 93, pp. 1591-1595, 1996
- [3] D.Adalsteinson and J.A.Sethian, A fast level set method for propagating interfaces. Journal of Computational Physics, pp. 269-277, 1995
- [4] A.E.Lefohn, J.M.Kniss, C.D.Hansen, R.T.Whitaker, A streaming narrow-band algorithm: interactive computation and visualization of level sets. IEEE Transactions on Visualization and Computer Graphics 2004.
- [5] Jiangwen Deng, H.T. Tsui, A fast level set method for segmentation of low contrast noisy biomedical images, 2001.



(a) 제안 방법 (b) 소프트웨어로 구현한 레벨 셋 기법  
그림 4. 레벨 셋 기법 구현방법에 따른 결과영상 비교