

거리 변환을 이용한 형태학적 골격화 알고리즘

유 선 국 · 강 병 옥

¹연세대학교 의과대학 의학공학교실

Using Distance Transformation on Morphological Skeletonization Algorithm

Sun K. Yoo, B.W. Kang

= Abstract =

Skeletonization is an algorithm to find the skeleton of the center obtain images of the object image. Through a distance of Morphological transformation functions provided in MATLAB and in order to see how to implement the Skeletonization. Programs Sources mathworks and were changed to suit the vascular image analysis modeling to analyze them.

Key words: Skeletonization, Distance Transformation, vascular image, MATLAB

서 론

디지털 영상에서 2차원 물체를 표현하고 기술하는 기법은 여러 가지 방법이 사용되고 있으며, 그 기법들 중의 하나가 물체 내의 화소에서 배경까지의 최단 거리로 표현하는 거리 변환(distance transform) 방법이 있다. 이는 물체의 특성을 분석하는데 중요한 실마리를 제공한다. 디지털 영상 처리에서의 거리를 나타내는 척도(measure)로는 마름모꼴(city-block), 체스판꼴(chessboard) 및 유클리디안(Euclidean) 거리의 세가지가 가장 많이 사용되고 있다. 세 가지 거리들 중에서 마름모꼴, 체스판꼴 또는 그 둘의 결합형인 팔각형(octagonal) 거리가 주로 많이 연구되어 왔다. 그 이유는 그 거리 변환들은 2회의 스캔(scan)으로 끝낼 수 있어 거리 변환을 구하는데 걸리는 시간이 적게 들며, 2진 형태학 오전 연산을 반복시켜 단계적으

로 거리를 누산할 수도 있다. 반면에 이들 거리 척도들은 물체의 방향에 매우 민감하다는 단점을 가지고 있다. 2차원 오브젝트의 기술, 표현 및 인식하는 문제에 있어서는 물체의 영역 정보와 윤곽선 정보 등이 이용된다.

본 론

서론에 언급한 거리 변환을 간단히 표현하면 하기 그림(1)과 같다.

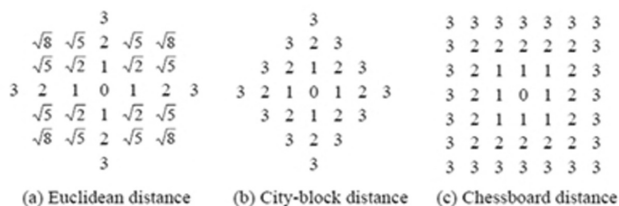


그림 1. 거리 변환 방식

통신저자: 유선국, (120-752) 서울시 서대문구 연세로 35 연세대학교 의과대학 의학공학교실

이중 MATLAB 프로그램으로 구현하게 될 방식은 chessboard distance 방식으로 아래 그림(2)와 같이 나타낸다.

$$x_0[m, n] = \begin{cases} 1 & \text{if pixel } [m, n] \text{ is inside the shape} \\ 0 & \text{otherwise} \end{cases}$$

사각형의 이미지가 있다고 가정한다면, 이미지가 있는 부분(1로 나타냄)에서 자신의 주위, 네 방향을 본다.

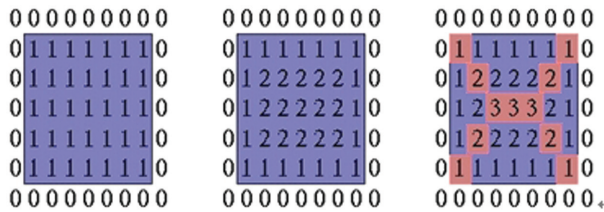


그림 2. chessboard distance

for each $x_{k-1}[m, n] = k$, do $x_k[m, n] = \min\{x_{k-1}[i, j]\} + 1$

here $[i, j]$ is the four closest neighbors of $[m, n]$.

네 방향 중에 가장 큰수와 자신이 처음으로 가지고 있던 초기값을 더하면 위 그림(2)의 가운데와 같은 형태가 된다. (2,2)에서의 거리값을 본다면, 자신의 주위가 모두 1이다. 즉, 1이 가장 큰 수이고 (2,2)에서의 초기값인 1과 더하면 2가 되는 것이다.

$$\{[m, n] \mid x_k[m, n] = \max\{x_k[i, j]\}\}$$

이러한 방식으로 여러 번 거치면, 그림(2)의 맨 오른쪽과 같이 나타난다. 그러면 이미지의 거리 변화는 끝나게 된다.

실 험

구현에 사용할 알고리즘 프로그램은 MATHWORK - Skeleton 2D by Roberto Herrera-Lara가 구현된 것을 바탕으로 혈관 이미지 분석에 맞도록 변경을 하여 진행하였다.

그림(3)은 프로그램의 분석을 위하여 변경 없이 구동한 결과이다. 변환 방식은 첫번째로 소스인 컬러 이미지를 지정된 Threshold 값에 맞춰 binary 이미지로 변환하여 우측 상단과 같이 된다. 이때 함수는 im2bw를 사용한다.

이후 imcomplement 명령으로 흑백 반전을 시켜주고, 그림(3)의 우측 상단 이미지의 붉은색 원 부분처럼 Threshold값이 맞지 않는 부분은 채워지지 않고 구멍이 뚫린 것처럼 되게 된다. 프로그램 작성자는 imfill 함수의 holes 명령을 사용해 그

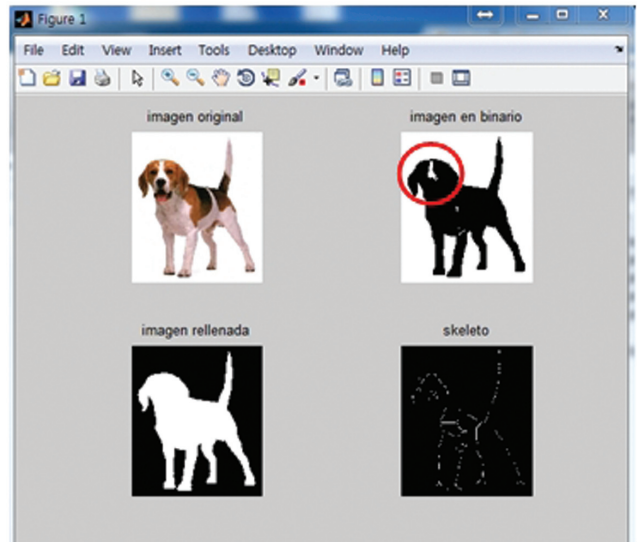


그림 3. 프로그램 구동

림(3)의 좌측 하단 처럼 구멍을 매꾼다.

마지막으로 바이너리 이미지 모폴로지 함수인 bwmorph의 skel 명령을 실행해 골격화를 완성했다.

이미지를 혈관조형이미지로 변경하여 실행해본 결과는 검은색의 이미지만을 얻을 수 있었다. 이유는 기존 프로그램의 Threshold값이 높게 책정되어 있어 이를 변경해야만 했다.

im2bw
Convert image to binary image, based on threshold

Syntax

BW = im2bw(I, level)
BW = im2bw(X, map, level)
BW = im2bw(RGB, level)

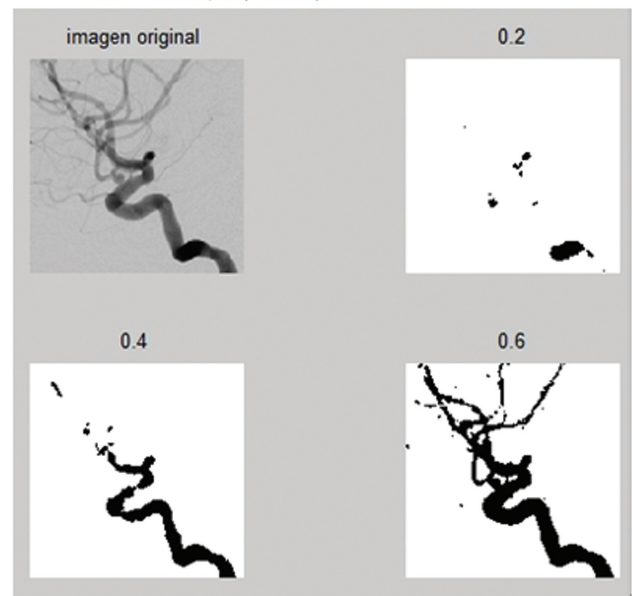


그림 4. im2bw 변경 적용

그림(4)처럼 Threshold 값을 조정하여 이미지의 출력상태를 확인하여 알맞은 값을 지정해 주었다. Threshold 값을 잘 지정해 주는 것이 최종 이미지에 가장 큰 영향을 미치는 것을 확인 할 수 있었다.

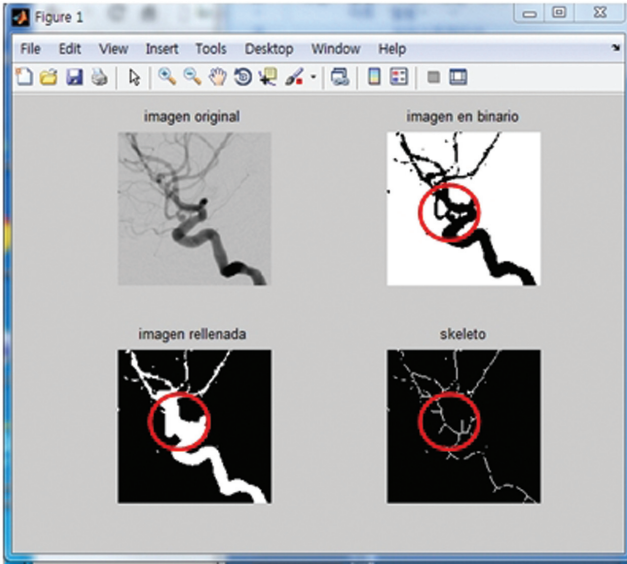


그림 5. 혈관조형이미지 실행 결과

위 그림(5)의 붉은색 원 부분처럼 imfill 기능을 사용시에 서로 다른 혈관 사이를 매꾸를 결과를 초래해 결론적으로 혈관을 관통해 지나가는 skeleton 이미지가 만들어지게 된다. 노이즈가 생기는 것 보다 관통하는 것은 더 많은 문제가 발생하기에 imfill 함수는 사용을 배제하고 진행했다.

그림(6)은 bwmorph 함수의 skel 명령을 테스트해본 그림이다. 우측 상단 1번, 좌측 하단 5번, 우측 하단 inf(이미지에 더 이상 변화가 없을 때 까지 skel 명령을 반복)를 실행한 결과이다. skel이 n번 진행될때마다 변하는 이미지를 확인 할수 있었으며 최종적으로 혈관에 구멍이 뚫리지 않고 잘 변환된 것을 확인 하였다.

결 론

기존 프로그램을 분석하여 방식과 핵심 함수, 인자들을 찾아

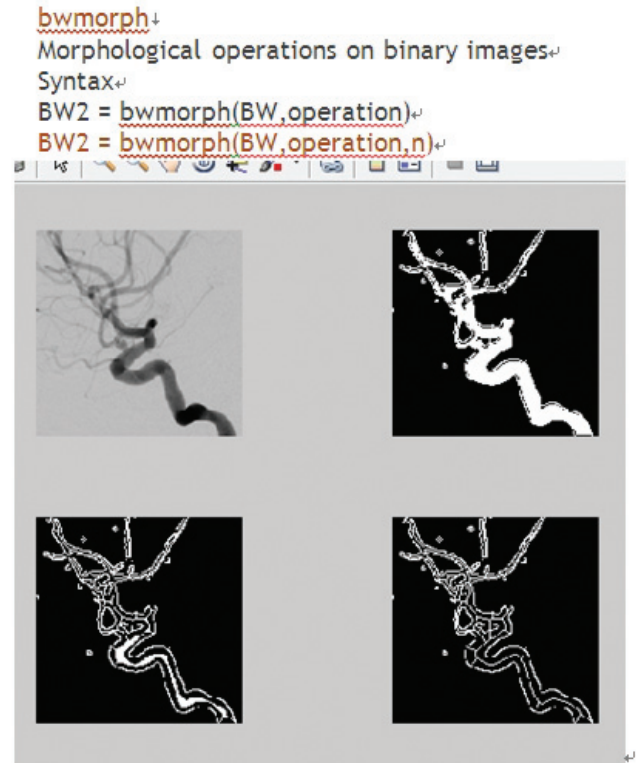


그림 6. bwmorph 테스트 화면

확인할 수 있었다. 비록 4세대 언어인 matlab 프로그램에 막혀서 직접 알고리즘을 작성하지는 못했지만 거리변환을 이용한 skeleton 방식에 대한 이해를 할 수 있었기에 기본 구현은 어렵지 않을 것으로 예상 된다. 하지만 이를 빠르고 정확하게, 원하는 대로 사용 할 수 있도록 알고리즘을 추가 개선하는 것이 핵심으로 보인다.

참 고 문 헌

1. 권준식, 최종수, "2-D object recognition using distance transform on morphological skeleton", 電子工學會論文誌, v.33B no.7, 1996년, pp.138-146
2. <http://pulsebeat.tistory.com/29>
3. <http://fourier.eng.hmc.edu/e161/lectures/morphology/node3.html>
4. <http://www.mathworks.com/matlabcentral/fileexchange/30298-skeleton-2d>

= 초 록 =

Skeletonization은 오브젝트 이미지를 해당하는 이미지의 뼈대를 구해서 중심점을 찾아내는 알고리즘이다. MATLAB에서 제공하는 Morphological 함수의 거리변환을 통해서 Skeletonization을 구현하는 방법을 살펴보는 것을 목적으로 한다. 프로그램을 mathworks에서 찾아 이를 분석해 혈관조형이미지 분석에 맞게 변경하였다.