Water Shed 알고리듬을 이용한 공 찾기

유지철* · 김영길*

*아주대학교

Finding the Ball with WaterShed Algorism

Ji-Chul Yu* · Young-kil Kim*

*Ajou University

E-mail : ugchul@ajou.ac.kr

요 약

본 논문은 Water Shed 알고리듬을 이용하여 여러 공들 중에 숨어있는 공을 찾는 것으로써 실재적인 목적은 Water Shed 알고리듬의 한계를 집어보고 분석해 보고자 함이다. 이로써 세포 분석이나 농산물 관련된 분야에서 사용되던 기존의 Water Shed의 문제점을 보완하는 계기가 될 것이다. 본 연구는 Visual Studio C++을 사용하여 구현해 보았으며, Distance Transform 및 Labelling 등의 알고리듬이 추가로 사용되었다.

ABSTRACT

This is the paper for the finding the hidden ball with the Water Shed Algorism, and for analyzing the weakness. So, With this paper, Other institutions or organizations can improve their researching. For this researching, Visual Studio C++ is used, and some kinds of Algorism is added to the software such as Distance Transform, Labelling which we make.

키워드

Water Shed, Distance Transform, Labelling, Distance Map, Minima

Ⅰ. 서 론

본 논문은 단순이 공을 찾는 것이 목적이 아니라 그림 1에서 보는 것처럼 38개의 공들 중에 34개의 공은 찾았지만 2개의 공은 결굴 찾지 못한 것을 알 수 있다. 이렇게 찾지 못한 2개의 공을 찾기 위함이 목적이다. 다시 말하면, WaterShed 알고리듬을 이용 하였을 때에 공을 찾지 못한 원인을 분석하고 찾지 못한 공을 찾기 위한 대한을 모색해 보기 위함이다.

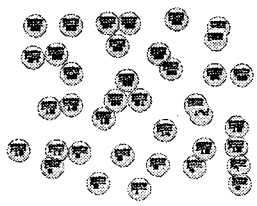

그림 1. 정보통신망

본 논문의 WaterShed 알고리듬에 대하여 간단히 서술하자면, WaterShed란 영상의 gray level 값을 높이로 생각하여 최소 영역에서부터 차츰 물을 채워 나가듯이 영역을 합쳐나가는데 최종적으로 하나의 테두리로 둘러싸인 부분을 균일 영역으로 판단한다.[1] 이때 사용된 또 하나의 알고리듬은 Distance Transform 알고리듬으로써, 그림 2에서 보는 것과 같이, 각 화소로부터 가장 가운데

영이 아닌 화소의 거리로 변환하여 꼭 지점의 거리나 위치를 계산 하는 것이다. 이 외에도 각각의 '0'이 아닌 물체들 사이의 거리를 구하는 데에도 사용 되지만, 본 논문에선 전자의 경우로 Distance Transform 알고리듬을 사용 하였다.[2] 그리고 그것에 대한 간단한 예를 그림 2에서 표현해 보았다.

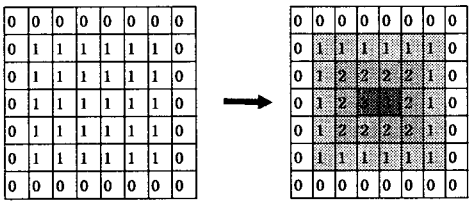

그림 2. Distance Transform

이렇게 구해진 Distance Map을 이용하여 두 물체 사이의 유역능선을 찾아내게 된다.

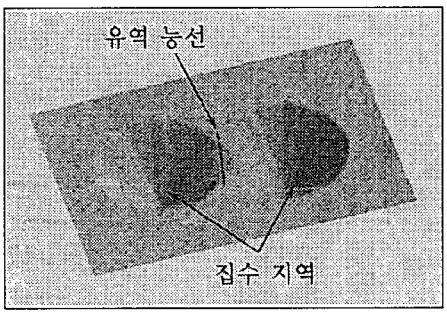

그림 3. 유역 능선(3)

II. 본 론

위에서 서술된 이론을 가지고 영상처리 알고리듬을 구현 하여 공의 개수를 개산하여 볼 경우 거의 완벽할 정도의 정확도를 보였다. 하지만 몇 층으로 싸여 있는 공의 무더기일 경우에선 정확도에서 많이 떨어짐을 확인 할 수 있었다.

그럼 왜 정확도가 떨어지는지에 대하여 예를 들어 보고, 결과를 서술해 볼 것이다.

우선, 그림 1에서, 각각을 구별하지 못한 공처럼 몇 가지의 경우를 포토샵을 통하여 구현 한 다음 영상처리 알고리듬을 사용하여 그 결과를 확인해 보았다. 그림 4로 알 수 있듯이 두 물체가 거의 포개어져 있을 경우, 그림 1에서 각각으로 구분되지 못한 2개의 물체처럼 인식 하고 있음을 알 수 있다. 이에 대한 원인은 다음과 같다.

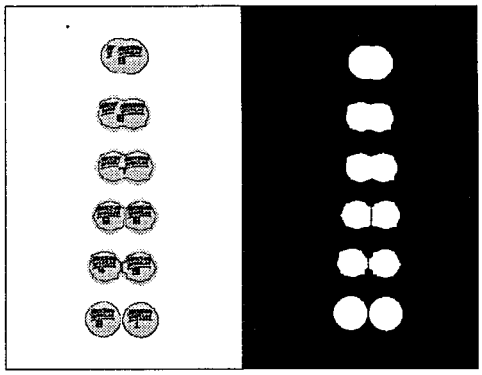

그림 4. 공의 간격에 따른 인식

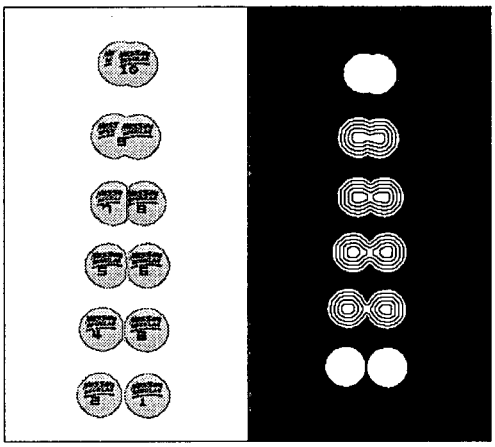

그림 5. Distance Map

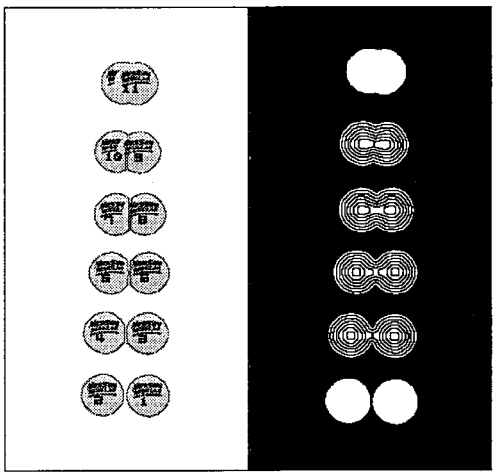

그림 6. Distance Map

위의 그림 6의 Distance Map의 등고선 간격이 그림 5의 등고선 간격 보다 조밀하다. 결과를 보면 Map의 등고선 간격이 조밀 한 그림 6의 결과에서 2번째에 있는 공이 서로 분리 되어 나타나는 것을 알 수 있는데, 다시 말해, 그림 3에서 설명된 집수 지역이 그림 5에서는 두 개로 구분 되지 못하여 유역능선을 만들지 못했고, 반면에 그림 6에선 집수지역이 두개로 구별 되어 유역능선을 찾게 되었다는 것이다.

그러면, 그림 6의 첫 번째 공도 Map의 간격을 높이면 찾아 낼 수 있지 않을까?, 하지만 Map의 간격을 더 이상 줄이게 된다면 4번째와 5번째의 집수지역에 영향을 준다. 결국에는 그림 7의 결과처럼(오른쪽의 결과는 왼쪽의 결과 보다 Map 간격을 더 좁힌 경우에 해당 된다.) 역행되는 결과가 나옴을 알 수 있다.

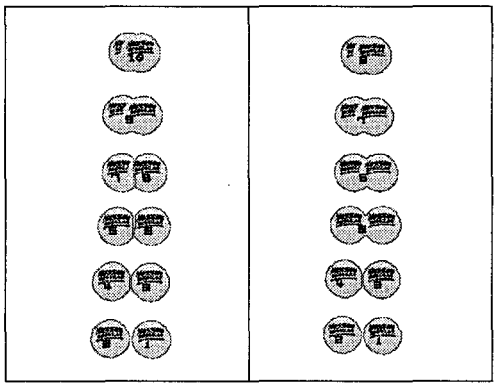

그림 7. Map 간격을 더 좁혔을 때

이처럼 Distance Map의 경우 어느 정도까지의 사용에는 상당히 합당한 결과를 얻을 수 있지만 더욱더 정밀한 곳에는 각각을 나누어 인식 하지 못함을 알 수 있었다. 그럼 이 알고리듬의 한계는 어떤지 더욱더 자세히 분석해 보겠다.

V. 결 론

그림 4나 5,7의 그림에서 맨 첫 번째 공을 서서히 서로 떨어뜨려 두 개의 공으로 인식하는 바로 그 순간을 측정하여 과연 공이, 다시 말해, 두 물체가 몇 %이상 까지 겹치면 각각의 물체로 인식하는 지, 못 하는지를 관찰해 보았다.

그림 8은 두 물체의 간격을 서서히 벌리면서 포개짐의 비율을 측정하였다.

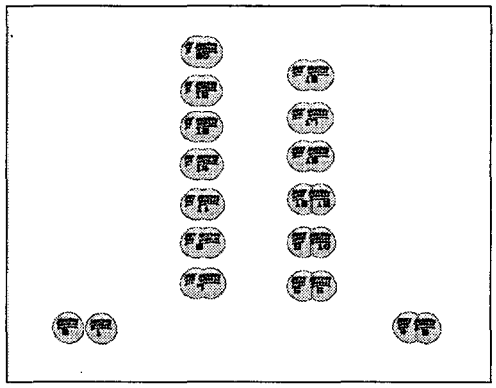

그림 8. 포개어짐 정도에 따른 결과를 알아보기 위한 그림

표 1. 포개어짐 정도 율

공번호 / 비율	19	17	15	12,13	9,10	5,6
	53%	50.3%	45.5%	44.8%	44.1%	43.4%

위의 표 1의 결과로 공의 절반을 가릴 경우, 다시 말해, 한 물체가 다른 물체의 반 이상을 가릴 경우 각각의 객체로 인식하지 못한다는 것이다. 이러한 점이 Water Shed의 한계라 할 수 있다. 또한 12, 13번 공처럼 각각의 객체로 인식하였을 때에도 동그란 공 원래모양이 아닌 집수지역으로 만들어진 유역능선에 의해 나누어지게 됨으로 마치 절단되어진 물체처럼 인식된다는 것도 Water Shed의 또 다른 한계점이라 할 수 있다.

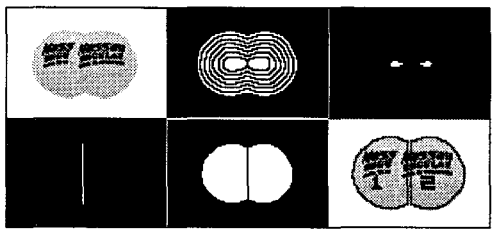

그림 9. 공 찾기까지의 일련의 과정

이러한 한계점을 인지하고 WaterShed를 최대한 잘 활용하기 위해서는 우선, 물체가 반 이상 겹치게 되어서는 안 되며, 또한 분리된 물체라도 그림 9처럼 데이터가 외곡 될 수 있음을 고려해야한다.

그리고 마지막으로 WaterShed의 가장 중요한 부분을 차지하는 Distance Transform에서 집수지역, 다른 말로, Minima를 최대한 분리해 내는 것이 핵심기술이 될 것이다. 이를 위해서는 몇 가지 영상철리 기법이 있을 수 있는데 대표적인 Morphological Image Processing중에 Closing 방법을 사용하여 어느 정도의 Minima를 떨어트릴 수 있을 것이다.

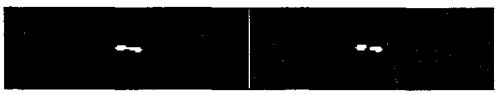

그림 10. Closing을 사용하여 Minima를 분리시킨 결과

그림 11. 분리시킨 Minima에 의한 영상 결과

그림 11은 어느 정도 기대한 결과가 나왔음을 알 수 있다. 하지만 물체의 외곡은 WaterShed에선 최대의 풀어야할 숙제이다.

참고문헌

[1] Lee Vincent and Pierre Soille, "Watersheds in digital spaces: An efficient algorithm", IEEE PAMI, 13, 6, p583~598, 1991

[2] Jain, "Fundamentals of Digital Image Processing", Pearson Prentice Hall, Chap9, 1989

[3] Rafael C.Gonzalez, Richard E.Woods, Steven L.Eddins, "Digital Image Processing", Pearson Prentice Hall, p418, 1992