

Bit-mapped 이미지 상의 정보 추출의 구현 및 분석

김 민호, 송 인희

한국 산업 기술 대학교 컴퓨터 공학과

{prendure, ihsong}@kpu.ac.kr

Implementations and analysis to extract information on a bit-mapped image

Minho Kim, Inhee Song

Dept. of Computer Engineering, Korea Polytechnic University

요약

Bit-mapped 형태로 저장되어진 측량 정보에서 원하는 정보를 추출하는 것은 여전히 중요하고 해 볼 만한 문제이다. 본 논문에서는 bit-mapped화 된 지도 이미지에서 주어진 두 지점간의 경로를 다양한 종류의 search 기법으로 추적하여 비교 분석했다.

1. 서론

기하학적인 자료들은 교통 관계 기관, 공공 기관, 또 정부의 기관 등 여러 방면에서 사용되고 또 처리되고 있다. 이런 기하학적 자료들은 bit-mapped 상태의 지도로써도 저장되는 경우가 많이 있다. 그러므로, bit-mapped 상태의 지도에서 요구되는 자료들을 적절히 추출하고 분석하는 것은 꼭 필요한 기능이다. 하지만, bit-mapped 상태의 이미지를 처리하는 것은 상당히 어려운 문제 중의 하나이다. 본 논문에서 우리는 이러한 자료들을 추출, 분석하는 문제를 실험했다. 일반적으로 사용되는 지도를 bit-mapped 형태로 저장하여 두 지점간을 연결하는 최상의 경로를 찾아냈다. 이 경로를 찾아내는 문제는 두 가지 중요한 문제를 다루고 있다. 하나는 지도 이미지에서 도로를 나타내는 부분을 추출하는 것이고, 다른 하나는 그 도로를 따라서 정해진 두 지점간의 경로를 trace 하는 부분이다. 지도에서 도로는 그것의 색은 다른 부분과는 다르다는 것을 이용하여 도로를 이미지에서 추출해 낼 수 있다.

도로를 추출하기 위하여 몇 가지의 알고리즘을 사용하여 지도의 다른 정보를 제거했다. 또한 추출된 도로의 정보를 따라서 원하는 최상의 경로를 찾아내기 위하여 휴리스틱 찾기 알고리즘을 사용했다. 경로를 추출하기 위하여 두 가지 종류의 criteria를 사용하는데, 경로를 따라 여행하는데 걸리는 시간을 주 요소로 사용했다. 지도는 대개 여러 개의 다른 종류의 도로형태로 표시된다. 즉, 고속도로, 주간선 도로, 등으로 나타내어진다. 각각의 도로에 다른 weights를 배당하면, 두 지점간의 거리와 도로의 종류에 따라서 예상 시간을 계산할 수 있다.

2. 지도 이미지에서 도로 추출하기

Thresholding 알고리즘은 배경에서 주요 물체를 추출하는데 사용되는 segmentation 알고리즘 중의 한 가지이다. 원하는 정보, 즉 도로의 값을 추출하기 위하여 도로의 값과 구분되는 배경들을 threshold시켜서 원하는 정보만을 저장할 수 있다. 그 결과가 효과적이기 위해서는 threshold 값, T 가 원하는 자

료의 값은 잃어버리지 않고, 배경의 값들은 제거하여 원하는 자료, 즉 도로의 형태를 가장 잘 보존할 수 있는 값으로 선택되어야 한다. 이 threshold 값은 지도의 종류, digitize 시 빛의 상태나 사용되는 카메라 상태에 따라서 다양하게 선택되어질 수 있다. 이 값은 많은 실험에 의해서 결정되어질 수 있는데, 한번 결정된 값은 비슷한 형태의 digitized 지도에서는 계속적으로 사용될 수 있다. 본 논문에서는 tester가 여러 번 테스트하여 가장 적절한 값을 선택할 수 있도록 구현되어 있다. 다음 단계에서 우리는 추출된 도로에서 주어진 두 지점의 경로를 찾아내야만 한다. Threshold된 지도 이미지에 는 도로를 나타내는 자료가 남겨져 있고, 주어진 두 지점 사이를 연결하는 도로를 추적하여야 한다. 실제 이미지에서 도로는 대부분 두께가 한 pixel이상의 형태로 남겨져 있는데, 도로의 두께를 한 픽셀로 줄이도록 하여야 실제로 도로를 따라가는 형태로 추적될 수 있다. 도로를 나타내는 물체에서 뼈대만 남을 때까지 돌레를 지워나간다. 여기서는 threshold한 후 bi-color의 이미지로 저장되어 있으므로, 도로가 가장 얇은 두께로 남을 때 까지 도로의 외곽 부분을 배경색으로 바꾼다. 결국 도로는 한 픽셀정도의 두께의 물체로 남겨진다.

3. 지도 이미지에서 두 지점의 연결 찾기

이제 threshold되고 thinning된 이미지에서 주어진 두 지점간의 연결된 도로를 trace 하는 것이 남아있다. 도로가 converge, diverge 또는 dead-end되는 것을 고려해서 연결 도로를 찾아야 한다. 본 논문에서는 3가지 다른 접근 방식으로 연결된 도로를 찾았다. 알고리즘의 기본 방식은 아래와 같다.

Place the starting point onto the OPEN list

REPEAT

Remove the first point, P, from the OPEN list

IF P is the goal point

THEN Exit with success

ELSE Add Point P to the CLOSED list

Add all of the unvisited neighbors of point P to the OPEN list

IF the OPEN list is empty

THEN Exit with failure

INFINITELY

위의 접근 방식은 어떻게 OPEN 리스트를 구성하느냐에 따라서 달라진다. 결국 찾는 도로의 결과가 달라진다.

3.1 Depth-First-Search

OPEN 리스트의 처음 부분에 찾아가지 않은 주변의 점들을 저장한다. 결과는 다음의 경우 중 하나가 참 일 때 까지 계속된다: dead-end에 도달 할 때거나, 반복이 감지되거나, 아니면 목표 지점에 도달할 때까지 계속된다.

3.2 거리를 사용한 방식

OPEN list의 점들은 아래의 평가 함수를 기본으로 ascending 순서로 저장되어진다:

$$f(P) = g(P) + h(P)$$

이 방식에서, $g(P)$ 는 시작점에서 임의의 점 P까지의 가장 짧은 경로의 길이를 나타내는 값이고, $h(P)$ 는 점 P에서 목표점까지의 Cartesian 거리이다. $h(P)$ 는 점 P에서 목표점까지의 최단 경로의 길이의 최하 bound이다.

3.3 추적 시간을 사용한 방식

다시 한번 OPEN list의 점들은 아래의 평가 함수를 기본으로 상승 순서로 저장되어진다:

$$f(P) = g(P) + h(P)$$

이 방식에서는 $g(P)$ 는 시작점에서 임의의 점 P에 도달하는데 걸리는 예상시간을 나타내는 값이고 (도로의 형태와 거리에 따라 다른 값을 가진다.) $h(P)$ 는 P에서 목표점까지 현재의 도로 타입에 따라 걸리는 예상시간을 나타내는 값이다. 여기서 $h(P)$ 는 위의 방식과는 달리 점 P에서 목표점까지 걸리는 예상시간의 최하 bound가 아닐 수도 있다. 함수 $g(P)$ 는

시작점에서 현재의 점 P까지의 최단 경로를 구성하는 다양한 도로를 거치는데 걸리는 시간의 합으로 계산되어 진다. 함수 $h(P)$ 는 점 P에서 목표점까지의 Cartesian 거리와 현재의 점이 놓여진 도로의 속도 요인에 따라서 계산되어 진다. 도로의 속도 요인의 도로의 종류에 따라서 결정되는데, 대개 도로의 색에 의해서 구분되어진다.

4. 구현 결과

본 논문에서는 같은 시작점과 끝점을 사용하여 세 개의 다른 방식을 적용하여 결과를 측정했다. 다양한 threshold 값과 thinning 횟수를 적용하였다. 사용한 지도 이미지는 인디애나 지도를 디지털이

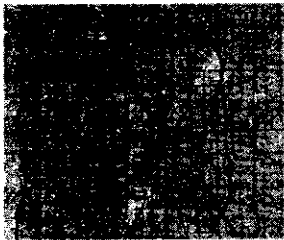
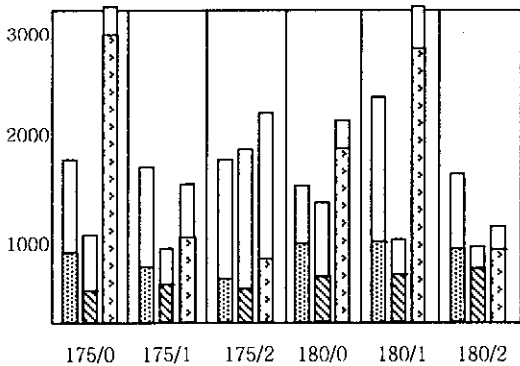


그림 1 그림 2(↓)



하여 사용하였다. 1번 이미지는 사용한 인디애나 지도이고, 2번은 3가지 방식의 결과를 나타내는 차트이다. 차트에서 x축은 다양한 종류의 threshold (첫번째 값)와 thinning값(두번째 값)이고 y축은 search space와 최종 경로를 나타낸다. 첫번째 그래프는 시간을 두 번째는 거리를 세 번째는 DFS 방식의 서치이다. 아래의 음영 부분은 최단 거리 경

로, 위 부분은 모든 서치 공간을 나타낸다. 차트를 보면 DFS 서치보다 휴리스틱 서치가 서치하는 부분이 작을 뿐 만 아니라 결과로 찾은 경로 역시 월등히 짧은 것을 알 수 있다. 그리고 휴리스틱 서치 방법이 DFS 보다 여러 가지 처리에 대해 덜 영향을 받는다는 것을 알 수 있다.

5. 결론

본 논문에서는 전통적인 이미지 프로세싱 알고리즘과 휴리스틱 서치 방법에 의해서 bit-mapped된 지도 이미지에서 성공적으로 최적 경로를 찾아낼 수 있다는 것을 보였다. Bit-mapped화 된 이미지에는 도로 외에 많은 데이터들이 포함되어 있어서 도로만을 trace하는데 많은 어려움이 있다. 또한 최적의 경로를 찾는데 많은 다른 경로를 거쳐서 찾아내는 문제점도 있다. 앞으로 우리는 더 나은 결과를 얻기 위하여 이미지를 처리하는 동안 생기는 이미지의 여러 가지 문제점을 제거하는데 연구를 계속할 계획이다. 또한, 여러 방면에 적용할 수 있도록 개선시킬 계획이다. 본 논문이 bit-mapped 이미지에서 처리하는 여러 종류의 방법에 도움이 되기를 바란다.

6. 참고 문헌

- [1] Gonzalez, Rafael and Paul Wintz, Digital Image Processing, Addison-Wesley, New York,
- [2] Nilsson, Nils, Problem-Solving Methods in Artificial Intelligence, McGraw-Hill, New York
- [3] Zhang, T. Y. and C. Y. Suen, " A Fast Parallel Algorithm for Thinning Digital Patterns," , Communications of the ACM, vol.27, no. 3, pp. 236-239, 1984
- [4] Baxes, G.A., " Digital Image Processing" , Wiley
- [5] Sedgewick, R., Fajolet, P. " Introduction to the Analysis of Algorithms" , Addison-Wesley Pub Co.