

평면 점집합에서 정렬을 고려한 개선된 컨벡스 헐 알고리즘

박병주*, 이재홍*, 강병익**

*한밭대학교 컴퓨터공학과

**건양대학교 의료IT공학과

e-mail:dinobei89@gmail.com

An Improved Convex Hull Algorithm Considering Sort in Plane Point Set

Byeong-Ju Park*, Jae-Heung Lee*, Byung-Ik Kang**

*Dept of Computer Engineering, Hanbat National University

**Dept of Medical Information Technology Engineering, Konyang University

요 약

본 논문에서는 임의의 정렬되지 않은 평면 점집합(Plane Point Set)에서 정렬을 고려한 개선된 Convex Hull 알고리즘을 제안한다. 이 알고리즘은 Convex Hull의 극점(Extreme Point) 특성을 이용하여 처리 데이터를 한정하기 때문에 계산복잡도를 낮춘다. 각 단계마다 볼록 정점(Convex Vertex)만을 판별하는 조건을 이용하여 한 번의 스캔으로 온전한 Convex Set이 구한다. 알고리즘 초기에 점집합의 정렬이 필요한데, 이때 걸리는 시간이 알고리즘 전체 동작시간의 대부분을 차지하는 만큼, 특성에 맞는 방법을 사용하여 빠르게 정렬하였다. 일반적인 상황을 가정하고 점집합을 랜덤하게 구성하여 실험하였으며 기존의 알고리즘에 비해 약 두 배의 속도 향상이 있음을 확인하였다.

1. 서론

Convex Hull 문제는 점집합 혹은 도형이 주어졌을 때 그것을 포함하는 최소 면적의 Convex Polygon을 구하는 문제이다[1]. 연산 결과인 Polygon은 기존 점집합의 서브셋이며 Convex Set이라고도 한다. 이 알고리즘은 계산 기하(Computational Geometry)에서 기본적인 문제로 다양한 분야에서 응용된다. 가령 Computer Graphics, GIS (Geographic Information System), HCI (Human Computer Interaction), Robotics 같은 분야에 응용될 수 있다[1]. Convex Hull을 SVM과 같은 인식기에서 사용할 불변 특징 추출[2]이나 빠른 인식기 학습을 위해 응용하는 연구[3] 등 패턴인식 분야에서의 연구나 모핑 알고리즘[4], 볼륨 렌더링 알고리즘[5]과 같이 영상처리 분야에서의 연구 등 다양하게 응용하는 실제 연구 사례를 찾아볼 수 있다.

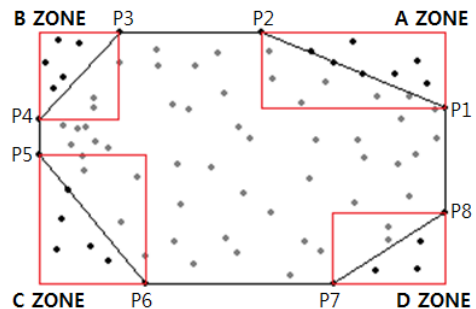
현재 많이 사용하는 Convex Hull 알고리즘은 탐색이 불필요한 모든 정점도 탐색하는데, Convex Polygon의 특성을 이용하면 점집합의 탐색 영역을 줄일 수 있다. 본 논문에서는 특정 크기의 2차원 영상에서 Convex Set을 구하기 위한 개선된 Convex Hull 알고리즘을 제안하였으며, 기존의 알고리즘과 수행시간을 비교하여 성능을 측정하였다.

2. 관련 연구

현재까지의 Convex Hull 알고리즘은 크게 두 가지 방법으로 나눌 수 있다. 이들의 주된 차이는 주어진 점집합

에 대해 정렬을 하는가에 있다.

우선, 널리 알려진 기존의 알고리즘을 시간 복잡도와 함께 살펴보면 Graham's Scan[6], Quickhull[7]과 같은 알고리즘은 $O(n \log n)$, Gift-wrapping 알고리즘은 $O(nh)$ 의 시간복잡도를 가진다. 여기서 h 는 극점의 개수를 뜻한다. Quickhull 알고리즘은 Divide and Conquer 방식을 이용하여 Convex Set을 계산하기 때문에 점집합을 정렬하지 않고 알고리즘이 수행된다. 이들 알고리즘 자체는 최적이지만 비싼 계산 때문에 다른 최적 알고리즘들이 제안되었다.



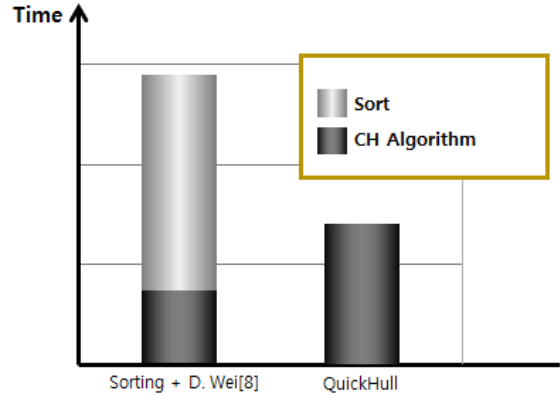
(그림 1) 점집합의 극점과 영역 분할

Dong Wei[8]의 알고리즘과 Xianquan Zhang[9]의 알고리즘은 극점을 통해 계산복잡도를 줄인 알고리즘이다. 극점을 사용하는 알고리즘은 특정 범위 내에 속하는 좌표만을 신속히 탐색해야 하기 때문에 초기에 점집합을 정렬해야 한다. 따라서 알고리즘의 시간복잡도는 정렬의 시간복

※ 본 연구는 교육과학기술부와 한국연구재단의 지역 혁신인력양성사업으로 수행된 연구결과임

잡도에 지배되어 정렬 알고리즘의 이론상 최소 시간복잡도인 $O(n \log n)$ 이 되게 된다.

극점을 사용하는 알고리즘[8]을 조금 더 자세히 살펴보면, 극점에 의해 점집합은 다섯 개의 영역으로 나뉘어 갈 수 있다. 최소 2개에서 최대 8개로 얻어진 극점을 모두 이어 Polygon을 구성하면 이 Polygon 내부에 포함된 정점들은 모두 볼록 정점이 될 수 없음을 알 수 있다. 따라서 선분에 의해 만들어진 Polygon 외부의 점들만이 후보가 됨을 알 수 있다. 4개의 영역 각각에 대해 볼록 정점 후보를 추려낸 뒤, 반시계 방향으로 Graham's Scan의 Circulation과 Back Tracking 방법을 사용하여 오목 정점 (Concave Vertex)을 제거하게 되고 남은 정점들이 Convex Set이 된다.



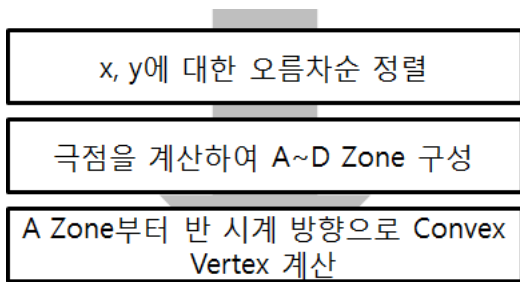
(그림 3) 정렬을 포함한 Convex Hull Algorithm 동작 시간 비율

3. 개선된 Convex Hull 알고리즘

3.1 알고리즘 구성

극점을 이용하는 기존의 알고리즘에서는 정렬 문제에 대한 별도의 언급 없이 기존의 정렬 알고리즘을 사용하도록 하고 있다. 하지만 실제 Convex Hull 알고리즘에서 정렬에 걸리는 시간의 비중이 매우 높기 때문에 정렬 방법에 대한 고려가 있어야 한다. 만약 $O(n \log n)$ 으로 동작하는 정렬 알고리즘을 이용하면 기존의 Convex Hull 알고리즘의 전체 수행 시간보다 더 오래 걸린다. 이러한 문제를 알고 있지만 정렬 알고리즘의 동작속도 향상에 대한 근본적인 개선은 이론상 불가능하다. 따라서 입력으로 주어질 점집합의 좌표를 특정한 크기의 영상에 속한 것으로 한정하여 빠르게 정렬을 수행하고 Convex Set을 계산하도록 하였다. 일단 점집합이 정렬된 이후에는 극점을 탐색하여 볼록 정점의 후보 영역만으로 한정하게 된다. 이후에는 Dong Wei[8]와 다르게 한 번의 탐색으로 온전한 Convex Set을 구하게 된다.

본 논문에서 제안하는 알고리즘은 (그림 2)와 같이 크게 세 단계로 이뤄진다.

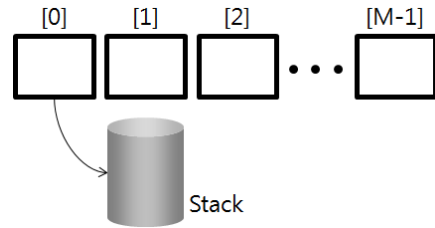


(그림 2) 알고리즘 구성도

3.2 제안하는 알고리즘

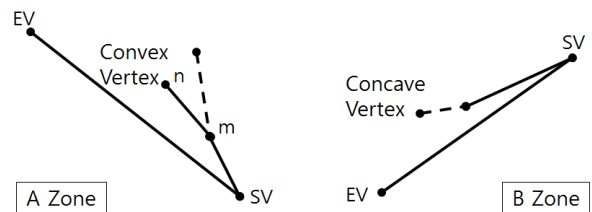
$W \times H$ 크기의 영상에서 임의의 점집합이 입력으로 주어질 때 이를 먼저 정렬해야 한다.

이때 정점 좌표 값의 범위가 특정한 크기의 영상으로 한정된 상황에서 빠르게 정렬하는 방법을 이용한다. 정렬할 원소의 개수가 N 이고 값의 범위가 $0 \sim (M-1)$ 인 점집합을 G 라고 하고, 길이가 M 인 포인터 배열을 pG 라 하자. 포인터 배열 pG 의 각 인덱스에는 스택이 들어간다.



(그림 4) 스택 포인터 배열 pG의 구성

$G[0] = g_0, G[1] = g_1, \dots, G[N-1] = g_{N-1}$ 일 때, 모든 i 에 대해 $pG[g_i] = i$ 를 할당한다. pG 에는 G 의 인덱스 값이 들어있다. 즉 값과 인덱스를 대치한 상태가 된다. 이 상태에서 $pG[0]$ 부터 $pG[M-1]$ 까지 순서대로 스택에 있는 모든 데이터를 POP하면 점집합 G 의 인덱스 i 가 나온다. 각각의 i 에 대해서 값으로 대치($G[i]$)하면 정렬된 데이터를 얻을 수 있다.



(그림 5) Convex/Concave 판단

이렇게 x, y 좌표에 대해서 정렬한 뒤 극점을 계산하게 된다. x 가 max일 때 y 가 min인 정점을 P1으로 할당하고 P8까지의 극점을 구한다. 이렇게 얻은 8개의 극점에서 P1과 P2로 만들어지는 영역부터 반 시계방향으로 4개의 Zone(A ~ D)을 할당한다. 각 Zone에서의 연산은 방향만

다르고 내용은 동일하다.

각 Zone에서의 볼록 정점을 보면, 반시계 방향으로 윗방향직선(directed line)들의 기울기는 점점 커짐을 알 수 있다. 이와 같은 특성을 이용하여 볼록 정점을 한 번의 스캔으로 판별한다. 각 Zone에서의 볼록 정점 탐색을 알고리즘은 다음과 같다.

- Step 0) $SV=P_i, EV=P_{i+1} (i=1\sim 8)$: region initialized
- Step 1) Select m, n (use CCW condition)
- Step 2) renewal m (m')
- Step 3) IF $m==n$ THEN ch.PUSH(m), go to Step 0)
- Step 4) IF slope of $\overline{SVm} < \overline{mm'}$ THEN ch.PUSH(m)
- Step 5) $SV=m, m=m'$, go to Step 1)

4. 실험 결과

본 연구에서는 기존의 Convex Hull 알고리즘을 개선하여 구현하고 기존의 알고리즘들과 연산 속도를 측정하여 비교하였다. 정점 8000개를 Quick Sorting 하였을 때 0.69ms의 시간이 걸렸다. 만약 기존의 정렬 알고리즘을 이용하면 총 2번 정렬하기 때문에 1.38ms 동안 정렬하게 된다. 같은 조건에서 정렬을 사용하지 않는 Convex Hull 알고리즘에서는 0.420ms로 3배가량 빨랐다. 따라서 점집합의 정렬을 포함한 알고리즘은 비교대상에서 제외하였다.

입력 점집합 P는 $P=\{(x, y) | 0 < x < 1023, 0 < y < 767\}$ 의 범위에 대해 랜덤하게 입력받았고 테스트는 2.66 Dual Core CPU의 PC에서 측정하였다. 측정된 시간은 1,000회 수행의 평균값을 사용하였다.

<표 1> 알고리즘 동작 시간 비교

Points Number	Quickhull[7]	Graham's Scan[6]	본 논문의 방법
500	0.032	0.411	0.023
1000	0.054	2.054	0.037
2000	0.114	5.76	0.064
4000	0.224	26.02	0.115
8000	0.420	108.313	0.221
16000	0.831	419.16	0.426
32000	1.653	1783.25	0.861
64000	3.356	7481.09	1.721

단위: ms

5. 결론

본 논문에서는 주어진 점집합에 대한 정렬을 고려한 개선된 Convex Hull을 계산하는 알고리즘을 제안하였다. 크게 점집합에 대한 정렬과 볼록 정점을 탐색하는 연산이 있는데 같은 개수의 점집합에 대해 정렬 알고리즘보다도 기존의 Convex Hull 알고리즘의 동작속도가 더 빠르기 때문에 정렬 문제를 가장 중요한 요인으로 보았다. 이를

해결하기 위해 특정 크기의 2차원 평면에 존재하는 점집합을 입력으로 받는 제한된 환경에서 동작하는 알고리즘으로 한정하였고 여분의 메모리를 사용하여 빠르게 정렬하였다.

또한, 기존의 극점을 이용하는 알고리즘을 개선하여 한번의 탐색으로 온전한 Convex Set을 했다. 그 결과 점집합을 이루는 정점의 개수와 무관하게 기존의 알고리즘에 비해 평균 2배의 속도로 동작하는 것을 확인하였다.

참고문헌

- [1] 차혜경, 신성용, "효율적인 Convex Hull 알고리즘과 그응용", 정보과학회지 제 5 권 제 4 호, 1987
- [2] Minhas. R, Wu, J, "Invariant Feature Set in Convex Hull for Fast Image Registration" Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on, vol, no, pp.1557-1561, 7-10 Oct. 2007
- [3] Chongming Wu, Xiaodan Wang, Dongying Bai, Hongda Zhang, "A Fast Training Algorithm for SVM Based on the Convex Hulls Algorithm" Signal Processing, 2008. ICSP 2008. 9th International Conference on, vol, no, pp.1578-1581, 26-29 Oct. 2008
- [4] Chun-Qiong Gong, "A New Method for the Morph of Planar Polygons Based on Shape Feature" Information Technology, Computer Engineering and Management Sciences (ICM), 2011 International Conference on, vol.3, no, pp.7-10, 24-25 Sept. 2011
- [5] He Fei, Li Xia, "A Improved Volume Rendering Algorithm Based on Convex Hull" Information Science and Engineering (ICISE), 2010 2nd International Conference on, vol, no, pp.1-3, 4-6. Dec. 2010
- [6] R.L. Graham, "An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set", Information processing letters pp.132-133, 1972
- [7] W Eddy, "A New Convex Hull Algorithm for Planar Sets", ACM Transactions on Mathematical Software, vol. 3, no. 4, pp.398-403, Dec 1977
- [8] Dong Wei, XingHua Liu, "Improved Algorithm for Computing Convex Hull of Plane Point Set", Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on , vol., no., pp.1-4, 11-13. Dec. 2009
- [9] X Zhang, Z Tang, J Yu, M Guo, "A Fast Convex Hull Algorithm for Binary Image", Informatica: An International Journal of Computing and Informatics, 2010