k-DOP을 이용하여 2차원 볼록 다각형간의 Hausdorff 거리를 계산하는 효율적인 알고리즘

(An Efficient Algorithm for Hausdorff Distance Computation of 2D Convex Polygons using k-DOPs)

이 지 은 † 김 용 준 ††
(Jieun Lee) (Yong-Jun Kim)

요 약 본 논문에서는 두 개의 이차원 볼록 다각형간의 Hausdorff 거리를 계산하는 효율적인 알고리즘을 제안한다. 볼록 다각형을 k-DOP으로 바운딩하고, k-DOP의 방향성과 계층적인 특성에 따라 관심영역만을 추적하는 방법으로, 본 논문에서 제안하는 알고리즘은 평균적으로 $O(\log n)$시간에 수행되며, 최악의 경우에도 $O(n)$의 수행성능을 보인다.

키워드 : Hausdorff 거리, k-DOP, 볼록 다각형, 기하학적 알고리즘

Abstract We present an efficient algorithm for computing the Hausdorff distance between two 2D convex polygons. Two convex polygons are bounded by k-DOPs and the regions of interest are traced using the orientational and hierarchical properties of k-DOP. The algorithm runs in a logarithmic time in the average case, and the worst case time complexity is linear.

Key words : Hausdorff distance, k-DOP, convex polygon, geometric algorithm

1. 서 론

자연과학이나 공학에서 주어지는 문제의 해결에 필요한 기하학적 거리는 최단거리(minimum distance)[1-4], Minkowski 거리[5], Hausdorff 거리[4,6-10], 측지거리(geodesic distance)[11] 등 여러 가지가 있다. 컴퓨터 게임 및 물리기반 시뮬레이션, 로보틱스, 가상현실 등에서 많이 사용되는 충돌감지 기술에는 최단거리가 주로 사용된다[3]. Hausdorff 거리는 형상 유사도 계산, 형상 인식 등에서 많이 사용되는데[12-15], 웹 2.0 시대를 맞이하여 2차원 혹은 3차원 형상 검색에 대한 요구가 많아지면서, 더욱 중요한 거리 개념으로 자리 잡고 있다. 따라서 이와 같이 다양한 형태의 거리 계산에서 공통으로 사용될 수 있는 기하학적 물체 표현과 이에 대한 효율적인 알고리즘을 개발하는 것은 향후 중요한 연구방향이다.

최근의 컴퓨터 시스템 성능의 획기적인 발전과 특히 프로그래밍이 가능한 그래픽스 하드웨어 가속기인 GPU의 성능 향상에 크게 힘입어, 여러 가지 형태의 기하학적 계산들을 하드웨어로 처리하는 접근 방법이 다각도로 시도되고 있다[16-18]. PLAYSTATION®3나 Xbox에서 NURBS로 표현된 3차원 물체를 다각형 근사없이 직접 렌더링 하는 기능들이 새로 개발되었으며[19,20], 이는 향후 하드웨어 가속에 기반을 둔 기하학적 처리의 기술 발전을 더욱 더 촉진시킬 전망이다. 구체적인 예를 들자면, 매우 복잡한 3차원 자유형상 모델에 대하여, 렌더링뿐만 아니라 물리기반 시뮬레이션, 충돌감지, Voronoi 영역 계산 등 여러 가지 기하학적 계산들을 직접 하드웨어로 처리하는 과정이 일반화될 전망이다.

하드웨어 처리의 핵심은 주어진 문제를 병렬적으로 분석하여 이에 적합한 고속 알고리즘의 개발과 시스템

† 정 회 원 : 조선대학교 컴퓨터공학부 교수
JieunJadeLee@gmail.com
†† 학생회원 : 서울대학교 전기컴퓨터공학부
kyj24182@3map.snu.ac.kr
논문접수 : 2008년 7월 31일
심사완료 : 2009년 1월 21일

정보과학회논문지: 시스템 및 이론 제36권 제2호(2009.4)

수행 성능을 최대화시키는데 있다. 물체를 미리 정해진 k개의 방향에서 바운딩하는 최소영역으로 근사하는 k-DOP은 하드웨어 가속을 위한 병렬처리에 적합한 특성을 가지고 있다[21]. 최단거리 계산이나 충돌감지에서는 이미 물체를 k-DOP으로 표현하여 병렬 수행의 장점을 활용하고 있으며[21,22], k-DOP을 여러 가지의 기하학적 문제의 해결에 응용하면, 이와 관련된 중요한 응용 문제들을 하드웨어를 이용하여 고속으로 처리할 수 있는 가능성이 커진다[23]. 본 논문에서 제안하는 Hausdorff 거리 계산 알고리즘은 아직 하드웨어 가속을 위한 구체적인 시스템의 구현을 염두에 두고 개발되지는 않았으나, 본 연구에서 제안하는 접근 방식의 기본적인 계층 구조를 이용하여 향후 하드웨어를 이용한 가속화가 쉽게 이루어 질 수 있을 것으로 기대된다.

본 논문에서는 k-DOP을 이용하여 2차원 볼록 다각형간의 Hausdorff 거리를 계산하는 효율적인 알고리즘을 제시한다. 제안하는 알고리즘은 볼록 다각형들이 이미 k-DOP으로 바운딩되어 계층적인 구조로 표현되었다고 가정한다. 전처리 과정에서 이러한 데이터 표현을 처리한다면 $O(n+k)$의 시간이 소요될 것이다. 본 논문에서는 이미 k-DOP으로 표현된 입력 물체를 가정하며, k-DOP의 방향적인 특성과 계층적인 특성을 활용하여, 이전 알고리즘이 $O(n)$ 시간에 수행되던 것에 비해, 평균적으로 $O(\log(n+k))$ 시간에 수행되는 성능 향상을 보인다. k의 값은 임의로 선택할 수 있으나, 원래 볼록 다각형이 n각형일 때 k는 $O(n)$인 값으로 택하는 것으로 가정한다. 이 경우 제안하는 알고리즘은 평균적으로 $O(\log n)$ 시간에 수행되고, 최악의 경우 $O(n)$ 시간의 복잡도를 가진다. 알고리즘의 핵심은 k-DOP을 계층적으로 처리하여 Hausdorff 거리가 측정될 부분만을 남기고 나머지 불필요한 부분을 효과적으로 제거하는데 초점을 두고 있다. 또한 본 논문에서 다루는 볼록 다각형은 그 경계와 내부를 모두 포함하는 영역을 의미하는 것으로 가정한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련된 이전 연구들에 대해 간략하게 요약하고, 3장에서 Hausdorff 거리의 정의와 오프셋을 이용한 Hausdorff 거리 개념을 설명하며, 4장에서는 k-DOP을 소개한다. 두 개의 볼록 다각형에 대하여 k-DOP을 이용한 효율적인 Hausdorff 거리 계산 알고리즘을 5장에서 제시하고, 다양한 적용 결과들을 예시한다. 끝으로 6장에서 본 논문의 결론을 내리고 향후 연구방향에 대하여 논한다.

2. 관련 연구

Atallah[6]는 두개의 볼록 m각형과 n각형에 대해, Hausdorff 거리를 계산하는 알고리즘을 소개하였다. 두 볼록 다각형을 A와 B라고 할 때, A에서 임의로 선택된 하나의 꼭지점에서 B까지의 최단거리를 계산하고, A의 나머지 꼭지점들에 대해 순서대로 최단거리를 계산하는데, A와 B가 볼록 다각형이므로 A의 다음 꼭지점에 대하여 B에 내리는 수선의 발을 찾는 과정을 B의 경계 곡선을 따라가면서 스캔하는 과정으로 해석하였다. B가 볼록 다각형이므로 이 과정은 B의 경계를 최대 2번까지 스캔하므로 $O(m+n)$의 수행시간을 갖는다. 이에 비해 본 논문에서 제안하는 알고리즘은 평균적으로 $O(\log n)$의 수행시간을 보이며(단, $n \geq m$), 최악의 경우에도 $O(n)$의 시간복잡도를 갖는다.

Hausdorff 거리 계산을 위한 알고리즘은 볼록 다각형뿐만 아니라 일반 다각형, 다면체, 곡선, 곡면, 점들의 집합 등 여러 가지 물체를 대상으로 개발되었다. Belogay 등[24]은 2차원 곡선 사이의 Hausdorff 거리를 계산하였고, Elber와 Grandine[4]은 3차원 곡선 사이의 Hausdorff 거리, 3차원 곡선과 곡면 사이의 Hausdorff 거리, 3차원 곡면 사이의 Hausdorff 거리를 정확하게 계산하는 대수적인 방법을 제안하였다. 이러한 연구는 알고리즘의 수치적 안정성과 정확성에 치중되어 있으므로 향후 계산의 효율성을 향상시키기 위한 새로운 접근 방법을 개발하는 것이 요구된다. 본 연구에서 제안하는 알고리즘의 기본적인 접근 방법은 이와 같은 알고리즘의 효율성 개선 측면에서 활용될 수 있다.

Llanas[9]는 주어진 이산적인 점들의 집합에 대하여 각 점에서 다각형에 이르는 최단거리가 최대가 되는 점을 찾아서 Hausdorff 거리를 계산하는 방법을 제안하였다. 임의로 선택된 한 점으로부터 다각형까지의 최단거리를 구한 후, 그 수선의 발을 중심으로 하고 현재까지 구해진 최단거리의 최대값을 반지름으로 하는 원을 만들고 이 원 안에 들어가는 점들을 모두 제거한다. 왜냐하면, 이런 점들로부터 주어진 다각형까지의 거리는 현재까지 구한 최단거리의 최대값보다 크지 않기 때문이다. 제거되고 남은 점들중에서 임의로 한 점을 택하여 최단거리를 구한 후, 그 거리가 현재까지 구한 최단거리의 최대값보다 짧으면 버리고, 더 길면 위의 과정을 계속 반복한다. 이렇게 하여 남은 점이 하나도 없을 때까지 Hausdorff 거리 계산 과정을 반복한다. Llanas가 제안하는 방법은 일종의 트리밍(trimming, 불필요한 부분들을 계산 대상에서 제거함) 방법으로 계산속도를 향상시킬 수 있다. 하지만, 여러 가지 측면에서 개선되어야 할 부분들이 많다. 예를 들자면, 수선의 발들에서만 트리밍 원들을 만들지 말고 이보다 더 조밀한 경계선 상의 점들을 택하여 더 많은 트리밍 원들을 만들어서 불필요한 점들을 제거하면 초기 단계에서도 더 많은 점들을 쉽게 제거하여 알고리즘의 전반적인 효율성을 높일

수 있다. 본 연구에서 제안하는 방법도 트리밍을 목적으로 하며, Llanas의 방법보다 훨씬 더 체계적이고 기하학적인 접근 방법이며 Hausdorff 거리 계산뿐만 아니라 보다 다양한 기하학적 문제의 해결에 이용될 수 있다.

k-DOP은 임의의 물체에 대한 기하학적 문제에 있어서 법선벡터와 관련이 있는 계산들을 비교적 간단하게 근사 계산할 수 있는 구조를 제공해 준다. 특히 두 물체가 접촉을 하는 경우, 그 접촉점에서 두 물체의 법선벡터의 방향이 정반대가 된다는 사실을 고려하여, 충돌감지의 문제를 단순화시킨다. Klosowski 등[21]은 이러한 k-DOP의 장점을 이용하여 충돌감지를 효과적으로 하기 위한 Bounding Volume Hierarchy 트리를 구성할 때 k-DOP을 이용하였으며, 그 결과로 시스템의 성능을 크게 향상시킬 수 있음을 보였다. 본 논문에서는 k-DOP의 물체 표현 구조가 충돌감지 문제뿐만 아니라 보다 일반적인 기하학적 계산에도 효율적으로 이용될 수 있음을 보인다. 구체적으로 k-DOP을 이용하여 Hausdorff 거리 계산이 효율적으로 이루어짐을 보인다.

3. 오프셋을 이용한 Hausdorff 거리 계산

일반적으로 기하학적 문제에 많이 사용되는 최단거리는 물체의 전반적인 형태를 고려할 수 없고, 물체간의 위치 관계 또한 고려할 수 없다. N. Grégoire와 M. Bouillot[25]는 아래의 두 그림을 이용하여 최단거리의 개념이 갖는 한계를 직관적으로 설명하였다. 그림 1(a)의 두 삼각형 사이의 최단거리 $d(p_2,q_0)$와 그림 1(b)의 두 삼각형 사이의 최단거리 $d(p_0,q_0)$는 같다. 하지만 그림 1(b)의 경우, 두 삼각형 사이의 관계는 그림 1(a)와 비교하여 더욱 더 가깝게 느껴진다. 이러한 차이를 나타낼 수 있는 보다 일반적인 거리의 개념은 무엇일까?

Hausdorff 거리는 이상에서 언급한 바와 같은 최단거리의 한계를 극복하기 위하여 고안되었다. 주어진 두 물체 A와 B에 대하여, A로부터 B까지의 Hausdorff 거리는 수학적으로 아래와 같이 정의된다:

$$D(A,B) = \max_{x\in A}\min_{y\in B} d(x,y).$$

여기서 $d(x,y)$는 어떠한 거리 함수도 사용될 수 있으나, 일반적으로 유클리드 거리를 사용한다. 본 논문에서는 $d(x,y)$를 유클리드 거리로 한다. 마찬가지로 B로부터 A까지의 Hausdorff 거리는

$$D(B,A) = \max_{y\in B}\min_{x\in A} d(y,x)$$

로 정의된다. 그리고 A와 B사이의 Hausdorff 거리는

$$H(A,B) = \max\{D(A,B),\ D(B,A)\}$$

로 정의된다.

그림 1(a)에서 $D(A,B)$는 점 p_0에서 삼각형 B에 최단거리의 수선을 내린 경우에 구해지고, $D(B,A)$는 점 q_1에서 삼각형 A에 최단거리의 수선을 내린 경우에 구해진다. 이 두 거리를 비교하면 두 번째 거리가 더 크므로, $H(A,B)$는 $d(p_2,q_1)$으로 구해진다. 그림 1(b)의 경우 $D(A,B)$는 점 p_1로부터 선분 $\overline{q_0q_1}$에 내린 수선의 길이와 같고, $D(B,A)$는 점 q_2에서 선분 $\overline{p_2p_0}$에 내린 수선의 길이와 같다. 따라서 $H(A,B)$는 $distance(q_2,\overline{p_2p_0})$로 주어진다. 우리는 그림 1(b)의 Hausdorff 거리가 그림 1(a)의 Hausdorff 거리보다 더 짧다는 것을 쉽게 알 수 있다.

Hausdorff 거리를 정의하는 또 다른 방법은 오프셋(offset)의 개념을 이용하는 것이다. 물체 A에 대한 거리 r 만큼의 오프셋 A_r은 A로부터 거리 r 이내에 있는 모든 점들의 집합으로서 $A_r = p \mid d(p,A) \le r$로 정의된다. 여기서 $d(p,A)$는 점 p로부터 A까지의 최단거리를 나타내고, p가 A에 속하는 경우 $d(p,A)=0$이다. 두 물체 A와 B에 대하여 Hausdorff 거리 $H(A,B)$는 아래와 같이 정의된다[10]:

$$H(A,B) = inf\{r>0 \mid A\subset B_r \text{ and } B\subset A_r\}.$$

그림 2(a)의 예에서 $0<r_A<r$인 모든 r에 대하여 $B\subset A_r$이고, $0<r<r_A$인 모든 r에 대해서는 $B\not\subset A_r$이므로 $D(B,A)=r_A$이다. 마찬가지로 $0<r_B<r$인 모든 r에 대하여 $A\subset B_r$이고 $0<r<r_B$인 모든 r에 대해서는 $A\not\subset B_r$이므로 $D(A,B)=r_B$이다. 따라서 Hausdorff 거리 $H(A,B)$는 $\max(r_A,\ r_B)$이 된다.

여기서 한 가지 중요한 기하학적 사실은 그림 2(b)에서 알 수 있듯이, 두 물체 A, B의 Hausdorff 거리를 결정하는 두 지점 p_A와 p_B는 법선을 공유한다는 사실이다[4,8]. 이러한 기하학적인 사실에 근거하여 Hausdorff

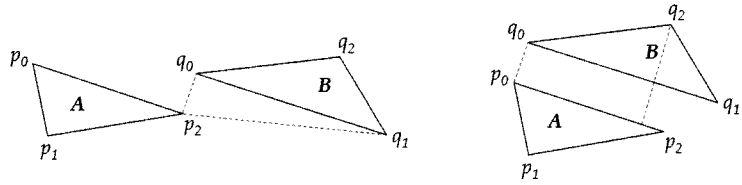
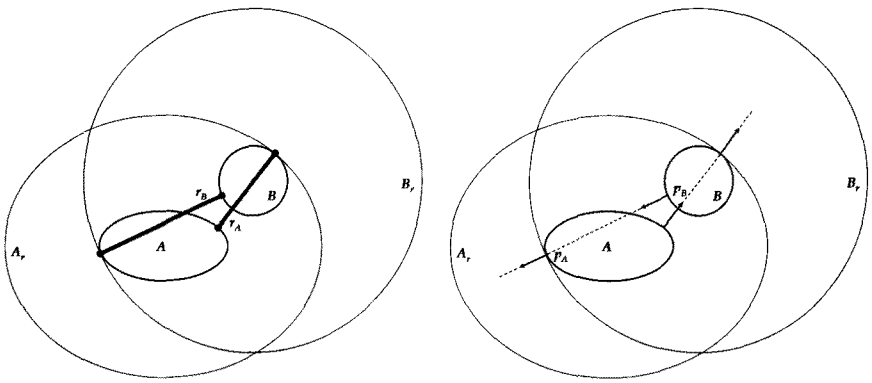
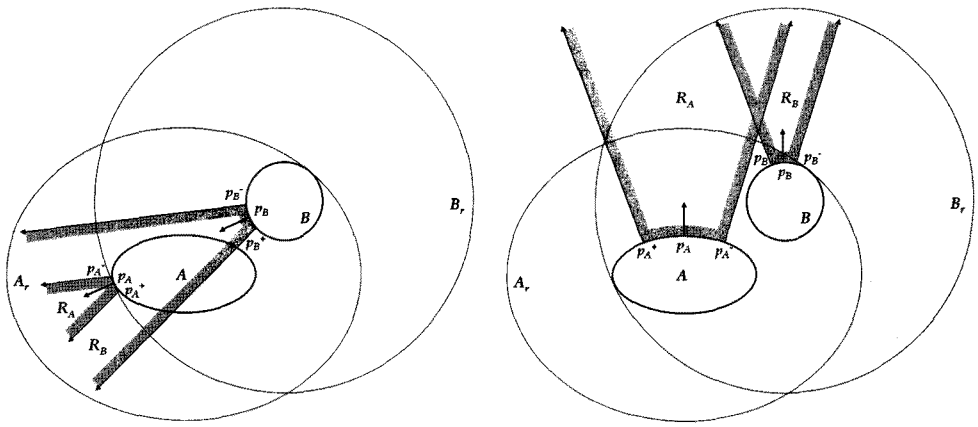


그림 1 두 삼각형 A와 B의 위치에 따른 최단거리와 Hausdorff 거리 비교



(a) (b)

그림 2 오프셋을 이용한 Hausdorff 거리 계산



(a) (b)

그림 3 곡선 세그먼트와 법선벡터의 방향 및 포함 관계: (a) Hausdorff 거리를 결정하는 경계점을 포함하는 경우, (b) Hausdorff 거리를 결정하는 경계점을 포함하지 않는 경우

거리를 계산하기 위한 효율적인 알고리즘을 개발할 수 있다. 알고리즘의 성능을 높이기 위하여, 정확한 해를 포함하는 부분 곡선분을 효율적으로 찾는 방법의 개발이 중요하다. 이에 대하여 아래에서 자세하게 살펴 본다.

본 논문의 서론에서 이미 언급하였듯이 볼록 다각형은 경계뿐만 아니라 내부의 영역까지 포함한 개념으로 가정한다. 따라서 볼록 다각형의 법선벡터는 다각형의 외부로 나가는 방향만을 고려한다. 그림 3(a)와 같이 Hausdorff 거리가 나타나는 지점 p_A와 p_B 근처의 부분 곡선들을 법선벡터의 값을 기준으로 대응시켰다고 가정해 보자. 즉 부분 곡선 $\overset{\frown}{p_A^- p_A^+}$와 부분 곡선 $\overset{\frown}{p_B^- p_B^+}$의 양끝점에서의 법선벡터가 일치한다고 가정하자. $\overset{\frown}{p_A^- p_A^+}$의 양끝점에서의 법선방향으로 그은 반직선들과 $\overset{\frown}{p_A^- p_A^+}$로 둘러싸인 영역을 R_A라 하고, 이와 비슷하게 $\overset{\frown}{p_B^- p_B^+}$와 양끝점에서의 법선방향으로 그은 반직선들로 둘러싸인 영역을 R_B라고 하자.

필요조건 A. $\overset{\frown}{p_A^- p_A^+}$와 $\overset{\frown}{p_B^- p_B^+}$가 A로부터 B까지의 Hausdorff 거리 $D(A,B)$를 결정하는 점 p_A와 p_B를 포함하기 위하여 $\overset{\frown}{p_A^- p_A^+} \cap R_B \neq \varnothing$이어야 한다. 마찬가지로 B로부터 A까지의 Hausdorff 거리 $D(B,A)$를 결정하는 점들을 포함하기 위해서는 $\overset{\frown}{p_B^- p_B^+} \cap R_A \neq \varnothing$인 조건을 만족해야 한다.

증명. A로부터 B까지의 Hausdorff 거리 $D(A,B)$를 결정하는 점 p_A와 p_B에 대하여, R_B는 p_B의 법선을 포함

하며 p_A와 p_B는 법선을 공유하므로, p_B의 법선은 p_A를 지난다. 그러므로 R_B는 적어도 p_A를 포함하며 $\overset{\frown}{p_A^- p_A^+}$도 p_A를 포함하므로, $\overset{\frown}{p_A^- p_A^+} \cap R_B$는 공집합이 아니다. 마찬가지로 B로부터 A까지의 Hausdorff 거리 $D(B,A)$를 결정하는 점 p_B와 p_A에 대하여, R_A는 p_A의 법선을 포함하며 p_B와 p_A는 법선을 공유하므로, p_A의 법선은 p_B를 지난다. 그러므로 R_A는 적어도 p_B를 포함하며 $\overset{\frown}{p_B^- p_B^+}$도 p_B를 포함하므로, $\overset{\frown}{p_B^- p_B^+} \cap R_A$는 공집합이 아니다.

[필요조건 A]는 Hausdorff 거리가 나타나는 지점들이 반드시 만족해야 하는 필요조건이다. 반대로 그림 3(b)와 같이 $\overset{\frown}{p_A^- p_A^+} \cap R_B = \varnothing$이고 $\overset{\frown}{p_B^- p_B^+} \cap R_A = \varnothing$이면, $p_A^-p_A^+$와 $p_B^-p_B^+$ 사이에는 Hausdorff 거리가 생기지 않으므로 이 영역들은 계산에서 제외할 수 있다.

완료조건 A. 위에서 언급한 [필요조건 A]를 만족하는 부분 곡선들에 대해서, 곡선의 구간을 줄여가며 문제를 해결하는 방법이 가능하다. 구간을 줄여 가다가 각 곡선상에 하나의 점이 남게 되면, 이 두 점 사이의 거리가 Hausdorff 거리가 된다. 실제 구현에서는 해당 곡선분이 다각형의 꼭지점(곡선의 일종으로 간주함), 선분, 간단한 형태의 곡선분 등으로 되는 단계에서 수치적인 방법으로 Hausdorff 거리가 나타나는 두 지점을 찾는다.

본 논문에서는 위의 접근 방법을 효율적으로 적용하기 위하여, 볼록 다각형을 k-DOP으로 근사하여 법선벡터들이 대응되는 구간들을 찾는다. 두 다각형을 k-DOP으로 근사한 후 k-DOP의 각 방향에 대해, k-DOP 세그먼트의 포함관계를 조사하고 한쪽의 세그먼트가 다른 쪽의 해당 영역에 포함되는 경우만을 대상으로 Hausdorff 거리를 계산한다. 이는 [필요조건 A]를 토대로 한 것이다. 또한 제안하는 알고리즘에서 k-DOP 구간을 분할해 가면서 필요조건을 만족하는 구간을 계산할 때, k-DOP 구간에 하나의 정점만 남는 경우에는 더 이상 계산을 진행하지 않고, 정점간의 거리를 Hausdorff 거리 후보로 제시한다. 이는 [완료조건 A]에 근거를 둔 것이다. 볼록 다각형의 경우, 각 선분에서는 하나의 고정된 법선벡터를 갖는다. 하지만 각 정점에서는 이웃한 두 선분의 법선벡터들 사이의 모든 방향벡터를 법선벡터로 갖는 것으로 간주한다. 따라서 본 논문에서는 다각형을 각 꼭지점의 위치는 고정되어 있고 접선과 법선방향만 연속적으로 바뀌는 곡선분의 특수한 경우로 간주한다.

4. k-DOP

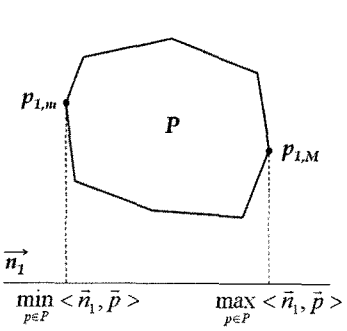
k-DOP(k-Discrete Oriented Polytope)은 AABB(Axis-Aligned Bounding Box)나 OBB(Oriented Bounding Box) 등과 같이 물체를 감싸는 바운딩 영역(bounding region, 3차원에서는 bounding volume)의 일종이다. AABB의 확장된 개념으로서, 주어진 물체에 대한 k-DOP은 $k/2$개로 균등하게 분할된 방향에 대해, 아래와 같이 물체와 접하는 접점을 구하여 계산한다:

$$p_{i,M} = \{p \mid \max_{p \in P} < \vec{n_i}, \vec{p} >\},$$
$$p_{i,m} = \{p \mid \min_{p \in P} < \vec{n_i}, \vec{p} >\}.$$

여기서 $\vec{n_i}$ (단, i=1, 2, 3, ..., $k/2$)는 k-DOP의 방향을 나타내는 단위 벡터이며, $\vec{p}$ 는 다각형 P의 각 정점 p에 대한 위치 벡터이다(그림 4 참조).

이와 같이, k-DOP은 미리 정해진 방향에 대한 접점들의 위치를 순서대로 나열하여 정의할 수 있다. 이러한 자료 구조의 장점은 k-DOP의 계층적인 특성을 바탕으로 하여, 다음 레벨의 k'-DOP (단, $k'=k/2$, $k/2^2$, $k/2^3$, ...)을 자동으로 계산할 수 있다는 것이다. 그림 5의 예에서 4-DOP은 8-DOP의 1,3,5,7번째 접점값만을 취하여 쉽게 만들어지며, 그림 5의 8-DOP과 4-DOP은 그림 6과 같이 저장된다.

k-DOP은 곡선 물체도 바운딩할 수 있다. 그림 7은 타원에 대한 4-DOP과 8-DOP의 예시이다.



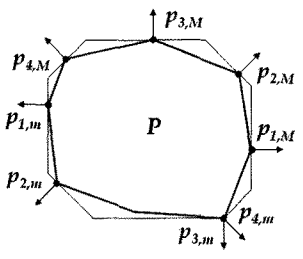


그림 4 다각형의 k-DOP 근사 방법

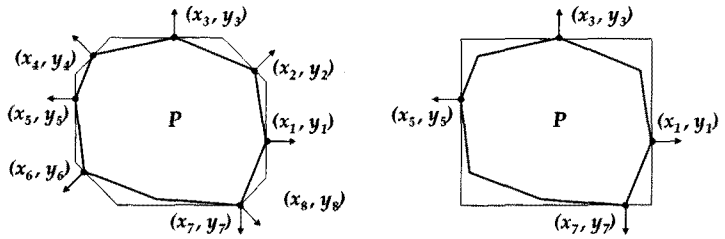


그림 5 다각형 P에 대한 8-DOP과 4-DOP

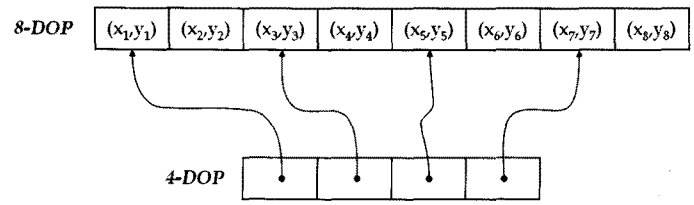


그림 6 그림 5의 8-DOP과 4-DOP에 대한 자료 구조

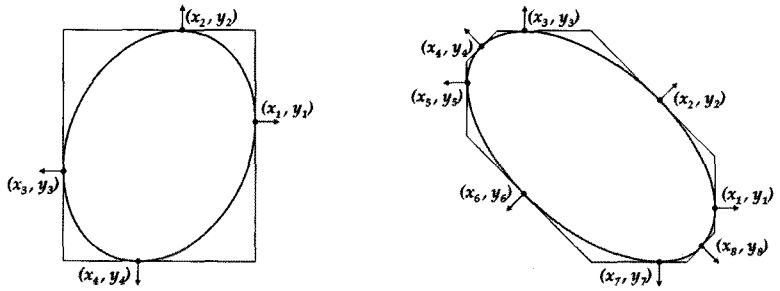


그림 7 타원에 대한 4-DOP과 8-DOP

5. k-DOP을 이용한 두 개의 볼록 다각형간의 Hausdorff 거리 계산

5.1 k-DOP을 이용한 다각형 트리밍

효율적으로 Hausdorff 거리를 계산하기 위해서는 이 거리를 결정하는 다각형의 일부분을 빨리 찾아내어 이 부분만을 남기고 필요없는 나머지를 빨리 제거하는 것이 필수적이다. 본 논문은 k-DOP의 방향적인 특성과 계층적인 특성을 이용하여, 불필요한 부분을 빠르고 효과적으로 제거한다.

다각형에서 불필요한 부분을 제거하는 과정은 크게 두 가지 기준을 따른다. 우선, 3장에서 설명한 [필요조건 A]에 부합하지 않는 부분을 먼저 제거한다. 그 다음, 아래의 5.1.3절에서 설명하는 k-DOP을 이용한 [Hausdorff 거리 범위 예측]에 의해 불필요한 부분을 추가로 제거한다.

5.1.1 k-DOP 세그먼트와 k-DOP 삼각형

k-DOP에 의한 트리밍을 설명하기 전에 k-DOP 세그먼트와 이를 포함하는 k-DOP 삼각형을 다음과 같이 정의한다. 하나의 k-DOP 세그먼트 $SEG(k,i)$는 k-DOP의 연속된 두 접점 p_i와 p_{i+1}을 양끝점으로 하고 그 사이에 포함된 다각형의 경계를 의미한다. 두 접점 p_i와 p_{i+1}에서의 두 접선들의 교점을 a_i라고 할 때, k-DOP 삼각형 $TRI(k,i)$는 삼각형 $\triangle p_i a_i p_{i+1}$으로 정의되며 그 경계와 내부를 모두 포함한다.

5.1.2 k-DOP 세그먼트의 방향과 포함 관계에 의한 트리밍([필요조건 A]의 k-DOP 적용)

이 절에서는 볼록 다각형과 k-DOP 표현에 대하여 3장에서 기술한 [필요조건 A]를 적용하여 구체적인 알고리즘으로 제시한다. 즉, 곡선의 법선벡터 방향이 대응되는 부분 곡선들 사이의 관계를 조사하는 것과 같이, k-DOP 표현에서는 각 k-DOP 접점을 기준으로 하여 k-DOP 방향이 대응되는 k-DOP 세그먼트들 사이의 관

계를 조사할 수 있다.

A의 선분 $\overline{p_ip_{i+1}}$와 접점 p_i에서 법선벡터 방향으로 그은 반직선과 접점 p_{i+1}에서 법선벡터 방향으로 그은 반직선으로 둘러싸인 영역을 $R_A(k,i)$라 하고, 이와 비슷하게 B의 선분 $\overline{\hat{p}_i\hat{p}_{i+1}}$와 접점 $\hat{p}_i$에서 법선벡터 방향으로 그은 반직선과 접점 $\hat{p}_{i+1}$에서 법선벡터 방향으로 그은 반직선으로 둘러싸인 영역을 $R_B(k,i)$라 하자. 영역 $R_A(k,i)$와 $R_B(k,i)$는 3장에서 정의한 영역 R_A와 R_B를 각각 포함하는 영역이다. 3장의 정의에 충실하자면 영역의 경계가 $SEG(k,i)$를 포함하게 되는데, 경우에 따라서는 여러 개의 선분이 $SEG(k,i)$에 포함될 수 있으므로, 알고리즘을 단순하게 만들기 위해서 $SEG(k,i)$를 하나의 선분으로 교체한 $R_A(k,i)$와 $R_B(k,i)$를 이용하여 트리밍을 한다.

필요조건 A1. (k-DOP 세그먼트의 방향과 포함 관계에 의한 트리밍 조건)

$SEG_A(k,i)$와 $SEG_B(k,i)$가 A로부터 B까지의 Hausdorff 거리 $D(A,B)$를 결정하는 점들 p_A와 p_B를 포함하기 위해서는 $TRI_A(k,i)\bigcap R_B(k,i)\neq\varnothing$이어야 한다. 마찬가지로 B로부터 A까지의 Hausdorff 거리 $D(B,A)$를 결정하는 점들을 포함하기 위해서는 $TRI_B(k,i)\bigcap R_A(k,i)\neq\varnothing$인 조건을 만족해야 한다. 이는 Hausdorff 거리가 나타나는 지점들이 반드시 만족해야 하는 필요조건이다. 반대로 $TRI_A(k,i)\bigcap R_B(k,i)=\varnothing$이고 $TRI_B(k,i)\bigcap R_A(k,i)=\varnothing$이면, $SEG_A(k,i)$와 $SEG_B(k,i)$에 대응되는 A와 B의 다각형 경계 부분에서는 Hausdorff 거리가 생기지 않으므로 이 부분들은 계산에서 제외할 수 있다. 여기서도 알고리즘을 단순하게 하기 위하여 3장에서 다룬 곡선분들 $\widehat{p_A^-p_A^+}$와 $\widehat{p_B^-p_B^+}$에 해당하는 $SEG_A(k,i)$와 $SEG_B(k,i)$ 대신 $TRI_A(k,i)$와 $TRI_B(k,i)$를 이용하여 트리밍을 한다.

그림 8을 예로 들어 보면, 선분 $\overline{p_1p_2}$과 접점 p_1에서 법선방향으로 그은 반직선과 접점 p_2에서 법선방향으로 그은 반직선으로 둘러싸인 영역에 $TRI_B(4,1)$이 겹치는 부분이 있으며, 선분 $\overline{\hat{p}_3\hat{p}_4}$과 접점 $\hat{p}_3$에서 법선방향으로 그은 반직선과 접점 $\hat{p}_4$에서 법선방향으로 그은 반직선으로 둘러싸인 영역에 $TRI_A(4,3)$이 겹치는 부분이 있으므로 두 세그먼트 $SEG(4,1)$과 $SEG(4,3)$은 남겨 두고, $SEG(4,2)$와 $SEG(4,4)$는 필요조건을 만족하지 않으므로 계산에서 제외시킨다. 단, 여기서 $SEG(k,i)$는 $SEG_A(k,i)$와 $SEG_B(k,i)$를 동시에 나타내는 표현이다.

5.1.3 k-DOP 삼각형에 의한 Hausdorff 거리 범위 예측과 트리밍

우리는 k-DOP 삼각형을 가지고 실제 다각형 부분 경계 사이의 Hausdorff 거리 값의 범위를 예측할 수 있다. $SEG(k,i)$는 $TRI(k,i)$에 항상 포함되므로, $SEG_A(k,i)$와 $SEG_B(k,i)$ 사이의 Hausdorff 거리는 $TRI_A(k,i)$에 속한 점 x와 $TRI_B(k,i)$에 속한 점 y 사이의 거리 $d(x,y)$가 가질 수 있는 거리의 최소값과 최대값 사이에 있다. 즉, $D(A,B)=\max_{x\in A}\min_{y\in B}d(x,y)$를 기준으로 수식화하면 다음과 같다:

$$\min_{x\in T_A}\min_{y\in T_B}d(x,y) \leq \max_{x\in T_A}\min_{y\in T_B}d(x,y) \leq \max_{x\in T_A}\max_{y\in T_B}d(x,y)$$

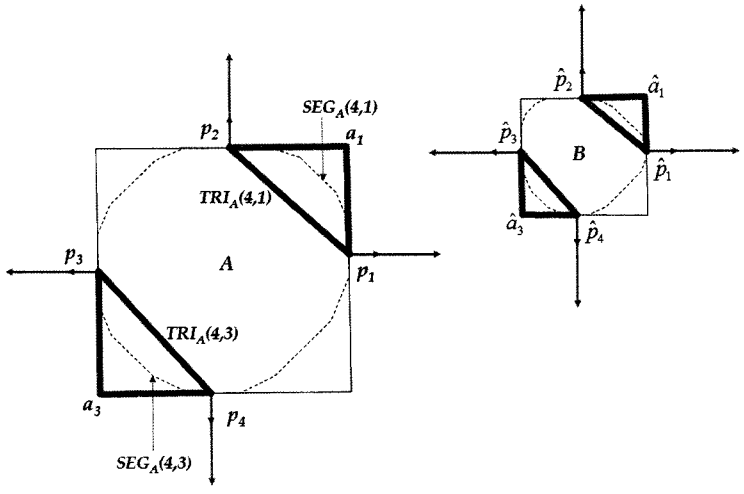


그림 8 k-DOP 세그먼트, k-DOP 삼각형

여기서 T_A는 $TRI_A(k,i)$를 나타내고 T_B는 $TRI_B(k,i)$를 나타내며, 삼각형의 경계와 내부를 모두 포함한다.

그림 8의 예시에서 [필요조건 A1]을 만족하는 두개의 k-DOP 세그먼트의 쌍이 있다고 하자. 하나는 $D(A,B)$에 대응되는 $SEG(4,3)$이고, 마찬가지로 $SEG(4,1)$은 $D(B,A)$에 대응되는 k-DOP 세그먼트이다. $SEG_A(4,1)$와 $SEG_B(4,1)$ 사이의 Hausdorff 거리는 $TRI_A(4,1)$과 $TRI_B(4,1)$에 속한 점들 사이의 거리의 최소값과 최대값 범위내에 있다. 여기서 최소값은 $TRI_A(4,1)$의 꼭지점 a_1에서 $TRI_B(4,1)$의 밑변 $\overline{\hat{p}_1\hat{p}_2}$ 에 수선을 내려 최단거리가 나오는지 조사하며, 그림과 같이 수선의 발이 내려지지 않는 경우 수선의 발과 가까운 밑변의 끝점 $\hat{p}_2$까지의 거리를 조사하여 구해진다. 마찬가지로 최대값은 $TRI_B(4,1)$의 꼭지점 $\hat{a}_1$에서 $TRI_A(4,1)$의 밑변 $\overline{p_1p_2}$에 수선을 내려 조사하며, 그림과 같이 수선이 내려지지 않는 경우, 수선의 발과 가까운 쪽 밑변의 끝점 p_1까지의 거리를 측정한다. 즉, $D(B,A)$는 a_1에서 $\hat{p}_2$까지의 거리를 최소로 하고, $\hat{a}_1$에서 p_1까지의 거리를 최대로 하는 범위에 포함된다. 이와 유사하게 $SEG_A(4,3)$과 $SEG_B(4,3)$ 사이의 Hausdorff 거리는 $\hat{a}_3$와 p_4 사이의 거리를 최소값으로 하고, a_3와 $\hat{p}_3$ 사이의 거리를 최대로 하는 범위에 포함된다.

그런데 $SEG(4,1)$에 해당하는 Hausdorff 거리 범위의 최대값이 $SEG(4,3)$의 Hausdorff 거리 범위의 최소값보다 작을 경우, 최종적으로 구해지는 Hausdorff 거리는 $SEG(4,3)$에서 나올 것이므로, 우리는 $SEG(4,1)$를 계산 대상에서 제거할 수 있다. 이와 같이 [필요조건 A1]을 만족하는 여러 개의 k-DOP 세그먼트 쌍들의 Hausdorff 거리 범위의 최소값과 최대값을 예측하여 Hausdorff 거리가 나올 수 없는 세그먼트 쌍을 제거하는 것을 [Hausdorff 거리 범위 예측]에 기반한 트리밍이라고 한다.

5.2 k-DOP을 이용한 두 개의 볼록 다각형간의 Hausdorff 거리 계산

아래의 알고리즘에서 소개하는 k-DOP을 이용한 영역 트리밍은 5.1.2절의 [필요조건 A1]에 의한 트리밍과 5.1.3절의 [Hausdorff 거리 범위 예측]에 기반한 트리밍을 번갈아 수행한다.

알고리즘 1. 두 개의 볼록 다각형간의 Hausdorff 거리 계산

INPUT : 두 개의 볼록 다각형은 두 개의 k_n-DOP으로 바운딩된 형태로 주어진다.

STEP 1: 두 개의 k_n-DOP 물체를 k_0-DOP으로 표현한다. 여기서 k_0는 k_n보다 작은 값으로 사용자가 지정할 수 있으며, k_n-DOP으로부터 자동으로 계산된다.

STEP 2: 같은 방향의 k-DOP 세그먼트 쌍에 대해 [필요조건 A1]에 의한 트리밍을 수행한다.

STEP 3: STEP 2에서 남은 k-DOP 세그먼트 쌍에 대하여, [Hausdorff 거리 범위 예측]에 의한 트리밍을 수행한다.

STEP 4: k-DOP 세그먼트에 꼭지점만 남거나 k가 k_n이 될 때까지, k를 증가시켜 k-DOP 세그먼트를 갱신하고 STEP 2-4를 반복한다.

STEP 5: STEP 4에서 남은 k-DOP 세그먼트들에 대해서 최종적으로 Hausdorff 거리를 계산한다. (5.3절 참조)

그림 9는 이 과정을 단계적으로 보인다. 그림 9(a)는 Hausdorff 거리를 계산하고자 하는 두개의 볼록 다각형이 16-DOP으로 표현된 A와 B이다. 초기 k값을 4로 하여 두개의 16-DOP을 4-DOP으로 표현하면 그림 9(b)와 같다. 4-DOP 세그먼트중 $SEG(4,1)$과 $SEG(4,3)$이 [필요조건 A1]을 만족하며, $SEG(4,2)$와 $SEG(4,4)$는 계산에서 제외된다. 다시 k-DOP 세그먼트의 [Hausdorff 거리 범위 예측]에 의한 트리밍을 시행하여, $SEG(4,1)$의 Hausdorff 거리 범위 예측의 최대값이 $SEG(4,3)$의 Hausdorff 거리 범위 예측의 최소값보다 작으므로 $SEG(4,1)$를 계산에서 제외하고, $SEG(4,3)$만이 남는다 (그림 9(b)의 굵은 회색선 부분 참조). 하나의 4-DOP 세그먼트 $SEG(4,3)$은 두개의 8-DOP 세그먼트 $SEG(8,5)$와 $SEG(8,6)$으로 분할된다(그림 9(c) 참조). 이 중 $SEG(8,6)$는 [필요조건 A1]을 만족하지 않으므로 제거되고, $SEG(8,5)$만이 남는다. 또한 하나의 세그먼트만 남았으므로 k-DOP 세그먼트의 [Hausdorff 거리 범위 예측]에 의한 트리밍은 생략된다(그림 9(c)의 굵은 회색선 부분 참조). 다시 $SEG(8,5)$만을 대상으로 8-DOP을 16- DOP으로 갱신하여, $SEG(16,9)$과 $SEG(16,10)$로 분할한다. 이 경우 두 세그먼트는 모두 [필요조건 A1]을 만족하고, k-DOP 세그먼트의 [Hausdorff 거리 범위 예측]에 의해서도 제거되지 않는다(그림 9(d)). 그리고 k값이 원래 주어진 물체의 k_n인 16에 도달하였으므로 더 이상 k-DOP 세그먼트를 분할하지 않으며, 최종적으로 남은 k-DOP 세그먼트들에 대하여 Hausdorff 거리를 직접 계산한다(그림 9(e) 및 5.3절 참조).

5.3 최종적으로 남은 영역의 Hausdorff 거리 계산

이제 우리는 남은 부분에 대해서 실제 다각형 경계 사이의 Hausdorff 거리를 계산한다. 5.2절의 알고리즘에서 최종적으로 남은 k-DOP 세그먼트의 쌍은 그림 10와 같이 4가지 경우이다. 그림 10(d)와 같이 k-DOP 세그먼트의 선분들이 사라지고 한 점만 남은 경우, 원래

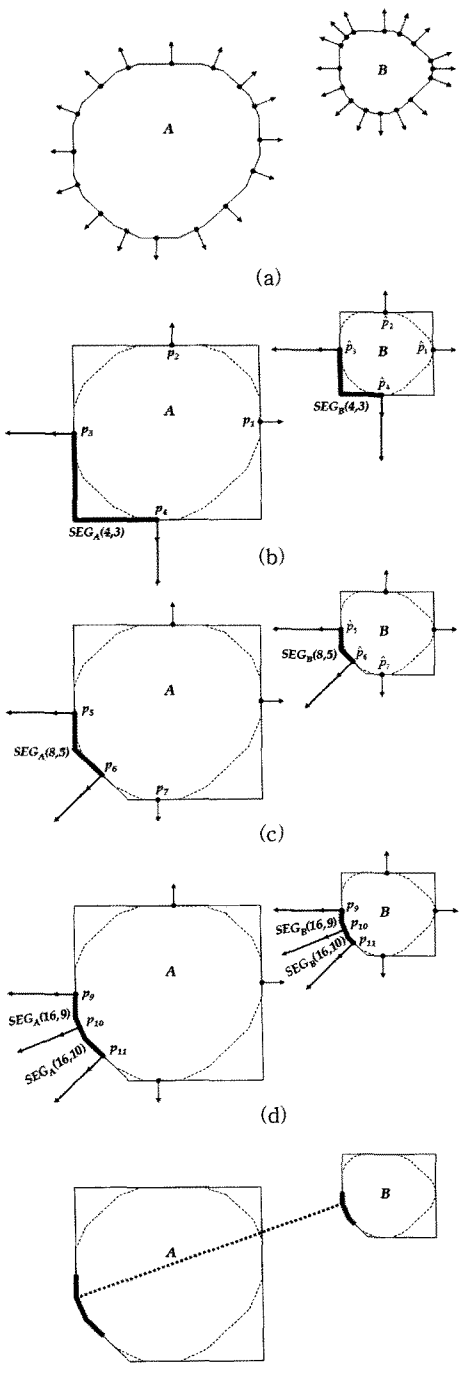


그림 9 (a) 두 개 볼록 다각형의 16-DOP 표현 A와 B, (b) 4-DOP 표현 및 트리밍, (c) 8-DOP 갱신 및 트리밍, (d) 16-DOP 갱신 및 트리밍, (e) 최종적으로 남은 16-DOP 세그먼트들과 Hausdorff 거리 계산

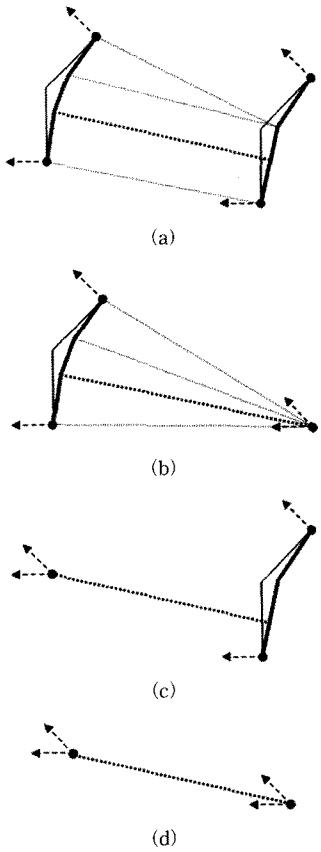


그림 10 최종적으로 남은 k-DOP 부분과 다각형 경계에 대한 4가지 경우와 Hausdorff 거리 계산(가는 선분이 k-DOP 선분이고 굵은 선분이 실제 볼록 다각형의 경계)

다각형에서 남은 부분도 하나의 정점이다. 그리고 그림 10(a)와 같이 k-DOP 세그먼트의 선분들이 남아 있는 경우 원래 다각형의 경계 선분들이 남게 된다. 이 단계에서는 트리밍 되고 남은 실제 다각형 경계 사이의 Hausdorff 거리를 직접 계산한다.

그림 10의 경우 k-DOP 방향을 보면 왼쪽 다각형에서 오른쪽 다각형까지의 Hausdorff 거리를 계산해야 함을 알 수 있다. 그림 10(a)의 왼쪽 다각형에 세 개의 선분이 남아 있으므로 네 개의 꼭지점에서 오른쪽 다각형까지의 최단거리를 구한다. 그림 10(a)의 왼쪽 다각형에서 오른쪽으로 이어지는 4개의 점선이 각 점에서 구한 최단거리이며, 이중 굵은 점선이 가장 긴 최단거리이다. 그림 10(b)-(d)에 해당하는 경우에도 각각 가장 긴 최단거리를 구한다.

5.1절의 k-DOP을 이용한 다각형 트리밍 과정을 통해 남은 k-DOP 세그먼트 쌍은 하나 이상이 될 수 있으며, 각

쌍에 대해 그림 10과 같이 각 세그먼트에 해당하는 Hausdorff 거리를 구한다. 각 쌍에서 구해진 Hausdorff 거리들 중 최대값이 바로 두 물체 사이의 Hausdorff 거리이다.

5.4 겹쳐진 k-DOP 물체의 경우

본 논문에서 제안하는 다각형 트리밍 알고리즘은 두 개의 겹쳐진 k-DOP 물체에 대해서도 적용된다. 단, 겹쳐지지 않은 경우보다 제거되는 k-DOP 세그먼트의 수가 작아질 가능성이 높다. 두 k-DOP 물체가 많이 겹쳐져 있을수록 제거되는 세그먼트의 수는 줄어들게 된다 (그림 11 참조).

5.5 계산 복잡도 분석

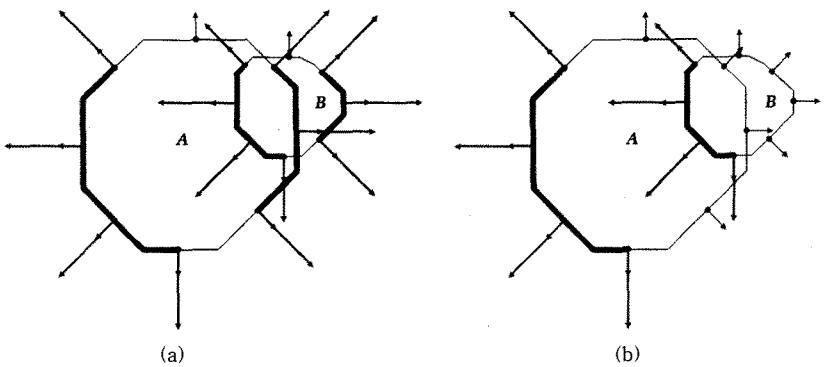
[알고리즘 1]에서 k-DOP을 k = 4, 8, 16, 32, ..., k_n 등과 같이 2의 지수배로 갱신한다고 가정하자. STEP 1은 명확하게 O(1)이다.

STEP 2-4의 시간복잡도를 계산하기 위해 하나의 k-DOP 세그먼트를 추적하는 과정을 살펴보자. 먼저 최선의 경우의 시간복잡도를 분석한다. 최선의 경우는 하위레벨 k-DOP의 한 세그먼트를 다음 레벨 k-DOP 세그먼트로 분할하였을 때, 두 개의 세그먼트중 하나만이 [필요조건 A1]에 의한 트리밍과 [최단거리 범위 예측]에 의한 트리밍 단계에서 제거되지 않고 남는 경우로서, 하나의 세그먼트에 대한 트리밍 검사와 분할 과정이 반복되는 경우이다. 그러므로 최선의 경우의 시간복잡도는 O(log$k$)이다. 일반적으로 하나의 k-DOP 세그먼트에서 분할된 두 개의 k-DOP 세그먼트가 모두 제거되지 않을 수도 있으나, 평균적으로는 k를 증가시켜 가면서 각 k-DOP 세그먼트는 결국 하나의 점으로 수렴하기 때문에, 최선의 경우와 같은 시간복잡도를 갖는다.

최악의 경우는 유사한 형태의 두 다각형이 겹쳐 있는 상황의 경우로서 이 때 STEP 2-4은 O(k)의 시간복잡도를 갖는다. 즉 하나의 k-DOP 세그먼트를 분할하였을 때 생기는 두개의 k-DOP 세그먼트가 모두 [필요조건 A1]을 만족하고 [Hausdorff 거리 범위 예측]에 의한 트리밍 단계에서도 제거되지 않는 경우가 이에 해당한다. 이 때 이진트리 방식으로 확장되는 모든 k-DOP 세그먼트들에 대해서 [필요조건 A1]과 [Hausdorff 거리 범위 예측]에 의한 트리밍을 수행하고 k-DOP 갱신을 수행해야 하므로, 결과적으로 O(k)의 수행 시간이 요구된다.

STEP 5에서 최종적으로 남은 k-DOP 세그먼트의 경우, 다각형의 선분들이 모든 방향으로 비슷하게 분포하는 평균적인 경우에는 상수개의 다각형 선분이나 점을 포함하므로 O(1)이고, 최악의 경우에는 O(n)이다. 최악의 경우는 최종적으로 남은 k-DOP 세그먼트에 입력 다각형에 속한 대부분의 선분들이 남게 되는 경우로, 남은 세그먼트에 대하여 Hausdorff 거리를 직접 계산하기 위해서는 O(n)의 계산시간이 요구된다.

이상의 분석을 종합하여 볼 때, [알고리즘 1]이 평균적으로 STEP 2-4에서 소요하는 시간은 O(log$k$)이다. 그리고 최악의 경우, O(n)이 될 수 있다. 일반적으로 n각형을 최적으로 바운딩하는 k-DOP은 다각형 각 선분의 법선방향에 해당하는 n개의 방향을 고려하면 되므로, 2n-DOP으로 표현할 수 있다. 하지만 이 경우 법선 방향들이 균등하게 배분되지 않으므로, 주어진 입력 다각형에 따라 k-DOP의 형태가 바뀐다는 단점이 있다. 따라서 본 논문에서는 균등한 방향으로 분할된 k-DOP을 사용한다. 본 논문에서 제안하는 알고리즘은 평균적으로 O(log$k$) 시간에 수행되므로, k의 값을 적절하게 결정하는 것이 중요하다. k는 m과 n의 값을 고려하여 결정하는데, k가 m과 n에 비하여 지나치게 큰 값일 경우, O(log$k$)의 수행시간이 기존 알고리즘의 성능 O(n)보다 더 커질 수도 있다. 반면, k가 m과 n에 비하여 지나치게 작은 값일 경우, 한 개의 k-DOP 세그먼트 내에 상대적으로 많은 수의 선분과 꼭지점들이 포함되므로 k-DOP에 의해 트리밍되는 성능이 전반적으로 저하되며,



(a) (b)

그림 11 겹쳐진 8-DOP 다각형의 (a) 필요조건 A1에 의한 트리밍 결과와 (b) 최단거리 범위 예측에 의한 트리밍 결과

따라서 트리밍 이후의 계산량이 많아질 수 있다. 본 논문에서는 k의 값이 $O(n)$으로 주어졌다고 가정한다. 따라서 본 논문의 수행시간은 평균적인 경우에 $O(\log n)$으로 주어지며, 최악의 경우에는 $O(n)$이 된다. 이 결과는 평균적인 경우의 시간복잡도를 비교할 때 기존의 $O(n)$ 시간 알고리즘에 비하여 매우 효율적인 알고리즘이다.

5.6 구현 결과

본 논문에서 제시한 알고리즘을 이용하여 계산된 두 볼록 다각형간의 Hausdorff 거리의 예들이 그림 12에 제시되었다. 두 개의 다각형이 분리되어 있는 경우(그림 12(a)), 일부가 겹쳐 있는 경우(그림 12(b)), 그리고 한 다각형이 다른 다각형의 내부에 포함되어 있는 경우

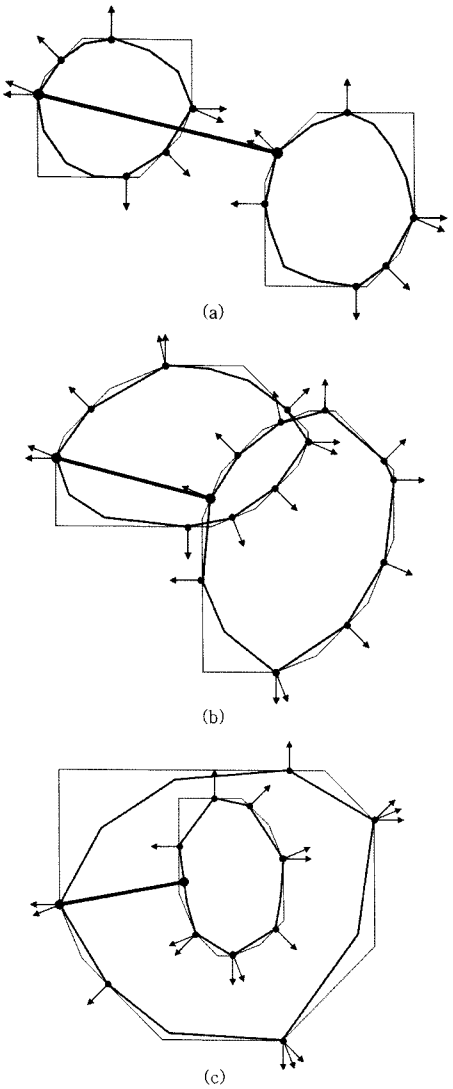


그림 12 볼록 다각형간의 Hausdorff 거리 계산의 결과

표 1 [6]의 방법과 본 연구의 방법의 Hausdorff 거리 계산을 위해 수행된 연산 횟수

시행번호	다각형A의 선분수	다각형B의 선분수	수행된 연산의 횟수	
			[6]의 방법	본 논문의 방법
1	11	13	113	15
2	11	13	103	50
3	6	6	21	48
4	6	13	56	15
5	5	6	34	29
6	13	6	71	21
7	11	6	62	22
8	13	13	104	29
9	6	7	23	58
10	11	6	34	15
...				
190	7	11	62	23
총 연산횟수			9224	5430

(그림 12(a))에 대하여 구현된 알고리즘이 정확한 해를 구할 수 있음을 확인할 수 있다. 또한 이 예제는 4-DOP에서 16-DOP까지 갱신하며 트리밍을 수행 하는데, 초기단계에서 트리밍이 되는 세그먼트들에서는 다음 단계에서의 k-DOP 표현으로 더 이상 진행되지 않으므로, 그림에 나타난 화살표의 총 개수로 전체 수행 시간을 예측할 수 있다.

표 1은 본 논문의 방법과 [6]에서 제시한 방법을 구현하여 실제 수행결과를 비교한 것이다. 실험은 5각형부터 13각형 사이의 볼록다각형 20개를 생성한 후, 임의의 2개를 골라 Hausdorff 거리를 계산할 때 수행된 연산 횟수를 측정하였다. 또한 20개의 다각형 집합에서 만들어지는 총 190가지의 다각형 쌍을 모두 계산할 때 수행된 연산 횟수도 측정하였다. 그 결과, [6]의 방법이 9224회 연산이 수행된 것이 비해, 본 논문에서 제시하는 방법은 5430회의 연산만 수행되어, 보다 빠르고 효율적으로 Hausdorff 거리를 계산할 수 있음을 보였다.

6. 결 론

본 논문에서는 두 개의 볼록 다각형간의 Hausdorff 거리를 효율적으로 계산하기 위하여, k-DOP을 이용한 트리밍 기법을 제안하였다. 본 논문의 알고리즘은 기존 방법의 평균적인 수행 시간 $O(n)$에 비해 월등히 향상된 $O(\log n)$의 수행 성능을 보인다. 최악의 경우에도 기존의 알고리즘과 같은 $O(n)$의 시간복잡도를 갖는다. 제안된 방법은 형상 매칭, 형상 검색 등의 응용분야에 적용되어 Hausdorff 거리를 측정하는 계산 시간을 크게 단축시킬 것이다.

향후 본 연구를 다각형뿐만 아니라 다면체, 곡선, 곡

면 등에 대한 효율적인 Hausdorff 거리 계산 알고리즘으로 확장할 계획이다. 또한 볼록하지 않은 물체에 대하여 Hausdorff 거리를 계산하는 효율적인 방법을 개발할 것이다. 볼록하지 않은 물체의 경우 Hausdorff 거리를 결정하는 두 점 p_A와 p_B 사이에 법선 공유를 하지 않는 경우가 생길 수 있으므로[4], [필요조건 A]보다 더 일반적인 트리밍 조건의 개발이 요구된다.

참 고 문 헌

[1] J. T. Schwartz, "Finding the minimum distance between two convex polygons," Information Processing Letters, Vol.13, pp. 168, 1981.

[2] F. Chin, and C. A. Wang, "Optimal algorithms for the intersection and the minimum distance problems between planar polygons," IEEE Trans. on Computers, Vol.32, No.12 pp. 1203-1207, 1983.

[3] E. Gilbert, D. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three dimensional space," IEEE Journal of Robotics and Automation, Vol.4, No.2, pp. 193-203, 1988.

[4] G. Elber and T. Grandine, "Hausdorff and minimal Distances between parametric freeforms in R^2 and R^3," Lecture Notes in Computer Science: Advances in Geometric Modeling and Processing, Proc. of 5th Int. Conf. GMP 2008, Vol.4975, pp. 191-204, 2008.

[5] H. P. Moon, "Minkowski Pythagorean hodographs," Computer Aided Geometric Design, Vol.16, No.8, pp. 739-753, 1999.

[6] M. J. Atallah, "A linear time algorithm for the Hausdorff distance between convex polygons," Information Processing Letters, Vol.17, pp. 207-209, 1983.

[7] G. Rote, "Computing the minimum Hausdorff distance between two point sets on a line under translation," Information Processing Letters, Vol.38, pp. 123-127, 1991.

[8] H. Alt and L. Scharf, "Computing the Hausdorff distance between sets of curves," Proc. of the 20th European Workshop on Computational Geometry, pp. 233-236, 2004.

[9] B. Llanas, "Efficient computation of the Hausdorff distance between polytopes by exterior random covering," Computational Optimization and Applications, Vol.30, pp. 161-194, 2005.

[10] J.-M. Morvan, Generalized curvatures, Springer, 2008.

[11] M. P. Do Carmo, Differential geometry of curves and surfaces, Prentice-Hall, 1976.

[12] H. Alt, B. Behrends, and J. Blömer, "Approximate matching of polygonal shapes," Technical report B 93-10, Institut für Informatik, Freie Universität Berlin, 1992.

[13] M. P. Dubuisson, and A. K. Jain, "Modified Hausdorff distance for object matching," Proc. of 12th Int. Conf. on Pattern Recognition, pp. 566-568, 1994.

[14] D. P Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.15, No.9, pp. 850-863, 1993.

[15] W. J. Rucklidge, Lecture Notes in Computer Science: Efficient visual recognition using the Hausdorff distance, Vol.1173, Springer-Verlag, New York, 1996.

[16] N. K. Govindaraju, S. Redon, M. C. Lin, and D. Manocha, "CULLIDE: Interactive collision detection between complex models in large environments using graphics hardware," In Graphics Hardware 2003, pp. 25-32, July 2003.

[17] A. Gress and G. Zachmann, "Object-space interference detection on programmable graphics hardware," In SIAM Conf. on Geometric Design and Computing, 2003.

[18] T. Klein, S. Stegmaier, and T. Ertl, "Hardware-accelerated reconstruction of polygonal isosurface representations on unstructured grids," Pacific Conf. on Computer Graphics and Applications, pp. 186-195, 2004.

[19] M. Guthe, Á. Balázs, and R. Klein, "GPU-based trimming and tessellation of NURBS and T-Spline surfaces," ACM Trans. on Graphics, Vol.24, No.3, pp. 1016-1023, 2005.

[20] C. Loop and J. Blinn, "Real-time GPU rendering of piecewise algebraic surfaces," ACM Trans. on Graphics, Vol.25, No.3, pp. 664-670, 2006.

[21] J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral, and K. Zikan, "Efficient collision detection using bounding volume hierarchies of-DOPs," IEEE Trans. on Visualization and Computer Graphics, Vol.4 No.1, pp. 21-36, 1998.

[22] G. Zachmann, "Rapid collision detection by dynamically allgned DOP-trees," Proc. of Virtual Reality Annual Int. Symp., pp. 90-97, 1998.

[23] A. Raabe, B. Bartyzel, J. K. Anlauf, and G. Zachmann, "Hardware accelerated collision detection - an archtecture and simulation results," Proc. of Design, Automation and Test in Europe, Vol.3, pp. 130-135, 2005.

[24] E. Belogay, C. Cabrelli, U. Molter, and R. Shonkwiler, "Calculating the Hausdorff distance between curves," Information Processing Letters, Vol.64, pp. 17-22, 1997.

[25] N. Grégoire, and M. Bouillot, "Hausdorff distance between convex polygons," http://cgm.cs.mcgill.ca/~godfried/teaching/cg-projects/98/normand/main.html.

이 지 은

1997년 이화여자대학교 컴퓨터공학과 공학사. 1999년 포항공과대학교 컴퓨터공학과 공학석사. 2007년 서울대학교 전기컴퓨터공학부 공학박사. 1999년~2002년 LG전자기술원 정보기술연구소 연구원 2008년~현재 조선대학교 컴퓨터공학부 조교수. 관심분야는 컴퓨터그래픽스, 기하모델링, 멀티미디어정보처리

김 용 준

2008년 서울대학교 수학교육과 이학사 2008년~현재 서울대학교 전기컴퓨터공학부 석사과정. 관심분야는 컴퓨터그래픽스, 기하모델링, 사용자인터페이스