

모바일 장치에서의 경계 곡면 가시화 A Visualization Process of Bounded Surfaces on Mobile Devices

*양상욱¹, #최영²

*S.W. Yang(araato@gmail.com)¹, #Y. Choi(yychoi@cau.ac.kr)²

¹ 중앙대학교 기계공학부, ² 중앙대학교 기계공학부

Key words : Mobile 3D, Constrained Delaunay Triangulation, Visualization

1. 서론

모바일 장치 상에서 3D 데이터를 직접 렌더링 하기 위한 도구가 등장하기 전의 모바일 3D 구현의 초기에는 매우 간소화된 3D 형상 데이터를 이용하거나^[1], 모델링과 삼각화 및 렌더링 등의 많은 계산이 필요한 것은 원격 서버에서 수행하고 모바일 장치에서는 이미지를 보여주는 방식^[2]의 연구가 진행되었다. 이후 OpenGL의 임베디드 시스템 버전인 OpenGL/ES^[3]의 규격과 함께 공개된 API가 제공되고, 마이크로 소프트의 Windows Mobile 5.0과 더불어 DirectX Mobile^[4] API도 공개됨에 따라 모바일 장치에서 삼각화된 3D 형상의 직접 렌더링이 편리해지고 3D 게임 등의 다양한 응용프로그램이 개발되고 있다.

CAD나 엔지니어링 소프트웨어 분야에서 3D 형상은 일반적으로 B-rep 솔리드와 같은 경계 곡면 기반의 데이터로 표현되는데, 경계 곡면 데이터가 가시화되기 위해서는 삼각화 과정을 거쳐야 한다. 삼각화 과정은 계산량과 메모리공간을 많이 필요로 하기 때문에 현재까지 모바일 장치에서의 3D 가시화는 미리 삼각화된 데이터를 다루는 것이 일반적이며, 이를 보다 효율적으로 구현하기 위하여 삼각형의 수와 해상도를 조절하여 수행 속도와 데이터의 크기를 조절하는 연구가 진행된 바 있으나^[5] 모바일 장치에서 직접 삼각화를 수행하는 연구는 찾아보기 어렵다. 본 연구팀의 선행연구^[6]에서는 PDA 상에서 보로노이 다이어그램을 먼저 계산하고 상동 관계로부터 디루니(Delaunay) 삼각형들을 구하여 경계 기반의 STEP 데이터를 가시화 하였으며, 본 연구에서는 구속 조건 삼각화(CDT, Constrained Delaunay Triangulation)를 직접 구하고 그 과정에서 경계 데이터에 대해서 상수시간 알고리즘을 적용하여 보다 효율적인 삼각화가 수행되도록 하였다. Fig.1은 모바일 장치에서 B-rep 데이터를 읽어 들이고 가시화하는 과정을 나타내고 있다.

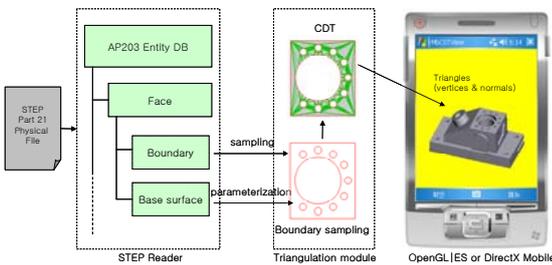


Fig. 1. Block diagram of entire visualization process of bounded surfaces on a PDA.

2. 경계기반 삼각화

경계 곡면의 형상은 기저 곡면(base surface)과 경계 곡선으로 표현되며, 이를 가시화 하기 위해서 2D 매개변수 공간에서 곡면 경계와 곡면의 곡률(curvature)을 고려하여 CDT를 수행하고, 경계 내부의 삼각형만을 3D 공간에서 렌더링한다. 전 가시화 과정에서 대부분의 계산 시간과 메모리 공간은 CDT에서 소요되며, 이 과정이 모바일 환경에서 수행되기 위해서는 시간과 메모리 공간을 효율적으로 trade-off 할 수 있는 삼각화 방법이 필요하다. 경계

곡선으로부터 삼각화 할 때의 입력데이터는 샘플링된 순차적 점 데이터이다. Fig.2는 샘플링 점들을 삽입하면서 삼각화를 수행하는 과정을 보여주고 있다. 각 과정을 진행 할 때 먼저 삽입된 점에 대한 정보를 활용 할 수 있으므로 점진적 구성(incremental construction) 방식이 적합하다.

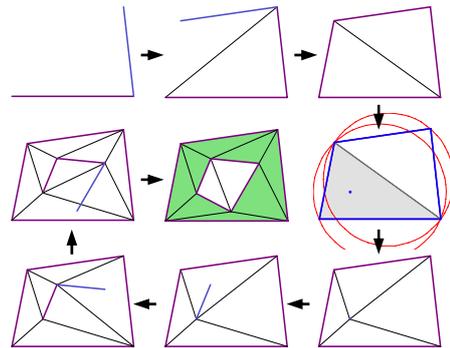


Fig. 2. Incremental construction with bounding edges.

점진적 구성 알고리즘의 복잡도는 일반적으로 n 개의 점에 대해서 $O(n \log n)$ 이며 검색(Locate)과 수정(Update)의 두 단계로 수행된다^[7,8]. 검색 단계에서는 새로이 추가될 점을 포함하는 기존 삼각형을 찾고, 수정 단계에서 새 점의 영향을 받는 기존 삼각형과 그 이웃들을 수정한다. 수정 단계는 상수 혹은 일차의 복잡도만을 소요하기 때문에 실질적으로 검색 단계가 전체 수행시간의 병목(bottle-neck)이 된다. 검색의 효율을 향상시키기 위한 여러 가지 방법들이 연구되어왔으며 Su 와 Drysdale^[7]은 여러 삼각화 알고리즘에 대해 분석적/실험적 비교를 수행하기도 하였다. Guibas 와 Stolfi^[9]는 고르게 분포되어 있는 점들의 집합에서 기존 삼각형의 임의의 에지부터 새 점 방향으로 삼각형들을 찾아가면서 검색을 수행할 때 $O(\sqrt{n})$ 시간이 소요됨을 보였다. 본 논문에서는 이를 응용하여 순차적으로 경계 에지를 추가할 때, 임의의 에지가 아닌, 바로 전에 삽입된 에지로부터 새로이 추가될 점까지 검색하여 하나의 점이 삽입될 때 상수 복잡도로 검색이 가능한 알고리즘을 개발하였다.

모든 점들을 포함할 수 있는 충분히 큰 초기 삼각형을 가정한다면, 추가되는 점들은 항상 기존의 한 삼각형 T 를 수정하게 된다. 이러한 가정은 전체 점집합의 불록 다각형 안과 밖에 대해서 따로 생각할 필요를 없애주기 때문에 점진적 구성에서 널리 사용된다. 점진적 구성은 검색된 T 로부터 수정되어야 할 주위의 삼각형들을 업데이트한다.

제한된 알고리즘은 경계 곡선을 순차적으로 따라가면서 삼각화하기 위한 방법이다. 이전의 점이 없을 때는 kd-tree를 이용하여 삽입될 T 를 찾으며 경계 곡선을 따라가면서 검색할 경우는 Fig.3의 왼쪽 그림과 같은 절차를 거치는데, 형상의 경계 루프를 따라 삼각화 할 경우가 이에 해당된다. Fig.3은 이미 어느 정도 삼각화가 이루어진 상태에서 내부 루프를 정의하는 과정에서 점 P_{n-1} 을 삽입한 이후 P_n 을 삽입할 때 P_n 을 포함하는 기존 삼각형 T 를 찾는 과정을 보여주고 있으며 검색 수행 절차는 다음과 같다.

- ① P_{n-1} 의 이웃 삼각형 중 $\overline{P_{n-1}P_n}$ 의 방향을 포함하는 삼각형 T_i 를 찾는다(초기 $i=1$).
- ② T_i 가 P_n 을 포함한다면 $T=T_i$, 과정 종료.
- ③ T_i 의 에지 중 P_n 방향으로 만나는 에지를 찾아 에지의 반대 삼각형을 T_i 로 한다.

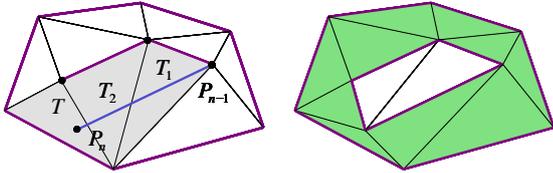


Fig. 3. 'Locate & Update' process during incremental addition of bounding edges.

3. 삼각화 수행 속도 분석

Table.1은 사이트의 수가 충분히 많으면서 복잡한 형상에 대한 삼각화를 할 때 제한된 검색 프로세스가 사이트수에 무관하게 상수 복잡도로 수행되는지를 검증하기 위하여 Fig.4의 데이터를 대상으로 삼각화 수행 속도를 측정 한 결과이다.

Table. 1. CDT building time experiment result (ms, tree depth)

# of initial Random pts.	394 edges (2 loops)	994 edges (17 loops)	1594 edges (33 loops)	2194 edges (48 loops)
1000	32 (48)	125 (57)	203 (57)	266 (57)
2000	31 (46)	94 (49)	172 (49)	234 (49)
3000	47 (35)	109 (49)	172 (49)	234 (49)
4000	47 (37)	109 (39)	172 (39)	234 (39)
5000	31 (33)	93 (41)	172 (41)	234 (41)
6000	47 (35)	109 (45)	156 (49)	218 (49)
7000	31 (39)	109 (46)	172 (46)	219 (46)
8000	47 (36)	94 (37)	172 (37)	235 (39)
9000	47 (37)	109 (43)	157 (43)	235 (43)

각 행은 경계 데이터에 의한 삼각화를 하기 전에 미리 생성해 놓은 랜덤한 삼각형의 숫자이며 각 열은 Fig.4의 데이터에서 홀의 수를 변화시켜서 경계 에지의 개수를 증가시킨 경우이다. 테이블 내의 숫자는 1/1000초(ms) 단위이며, 괄호 안의 항목은 kd-tree의 최대 깊이(maximum depth)이다. 각 경우의 수행 시간을 보면 초기 사이트 수에 관계없이 입력 데이터의 수에만 영향을 받는다는 것을 알 수 있다. 또한 트리의 깊이가 균형적이지는 않지만 수행시간은 트리 깊이의 영향 크게 받지 않는 것을 볼 수 있다. 계산 시간 측정은 모바일 장치에서 수행된 것은 아니고, Core2Duo 2.13GHz PC에서 수행되었다.

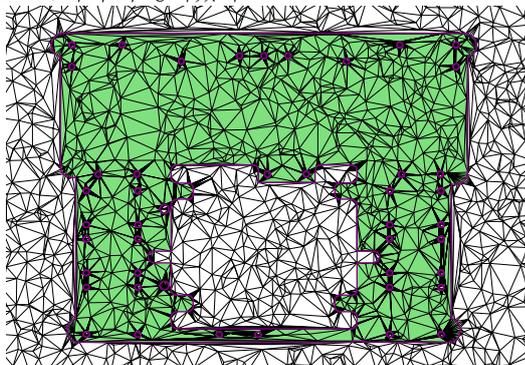


Fig. 4. Test data for the experiment described in Table.1.

4. PDA에서의 가시화

Fig.5는 제시된 알고리즘을 적용하여 모바일 환경용으로 구현된 프로그램을 PDA 장치에서 실행한 결과를 보여 주고 있다. 첫 번째 그림은 STEP 파일을 읽기 위한 대화상

자 화면이고 두 번째와 세 번째는 STEP 파일을 가시화한 결과인데, HP사의 hx2700 PDA 상에서 각각 1060개와 3567개의 삼각형을 생성하는데 4.1초와 14.6초가 소요되었으며 이 데이터에 대해서 선행연구^[6]와 비교한다면 약 2 배 빨라진 속도이다. 개발환경은 Microsoft Visual Studio 8.0 이며, Windows Mobile 5.0 Pocket PC SDK 와 OpenGL|ES 1.1을 이용하였다.



Fig. 5. Mobile STEP data viewer on a PDA

4. 결론

일반적으로 삼각화된 3D 데이터는 빠르게 렌더링 가능한 장점 때문에 경량 뷰어나 모바일 응용 프로그램에서 많이 사용되고 있다. 그러나 뷰어나 모바일 프로그램에서 직접 삼각화가 가능하다면 삼각화된 데이터 보다 훨씬 컴팩트한 형식으로 3D 표현을 할 수 있으므로 3D 데이터의 전송을 위한 네트워크의 사용량과 시간을 줄이고, 제한적인 모바일 장치의 저장공간을 효율적으로 사용하는데 도움이 될 것이다. 본 논문에서 제안한 상수 시간 CDT 알고리즘은 가시화 과정과 같이 연속된 경계 점의 입력이 있는 경우 매우 효율적으로 사용될 수 있을 것으로 생각되며 구현된 모바일 뷰어를 통해서 실용적인 면에서 검증하였다.

후기

본 연구는 서울시 산학연협력사업과 한국과학재단 특정기초 연구지원사업(과제번호 R01-2006-000-10327-0)의 지원으로 이루어졌으며 이에 감사 드립니다.

참고문헌

1. Woodward, C., Valli S., Honkamaa P., and Hakkarainen, M., "Wireless 3D CAD Viewing on a PDA Device", Proceedings of the AMOC, 14-17, 2002.
2. Su, X., Prabhu, B.S., Chu, C.C., and Gadh, R., "Middleware for Multimedia Mobile Collaborative System", Proceedings of WTS, 14-15, 2004.
3. Khronos Group, "OpenGL ES Overview", <http://www.khronos.org/opengles/>
4. msdn.microsoft.com, "Direct3D Mobile for Windows Mobile-based Devices"
5. Zunino, C., Lamberti, F., and Sanna, A., "A 3D Multiresolution rendering engine for PDA devices", SCI 2003 Proceedings, 5, 538-542, 2003.
6. 양상욱, 최영, "모바일 장치에서의 가시화를 위한 경계 기반 삼각화", 한국 CAD/CAM 학회 학술발표회 논문집, 487-494, 2007.
7. Su, P., and Drysdale, R., L., "A Comparison of Sequential Delaunay Triangulation Algorithms", Symposium on Computational Geometry, 61-70, 1995.
8. Boissonnat, J., D., and Teillaud, M., "On the randomized construction of the Delaunay tree", Theoretical Computer Science, 112, 339-354, 1993.
9. Guibas, L. and Stolfi, J., "Primitives for the manipulation of general subdivisions and the computation of Voronoi Diagrams", ACM Transactions on Graphics, 4, 75-123, 1985.