

3차원 측정 데이터의 중첩 제거 알고리즘 Overlap Removal Algorithm for 3D can Data

**정연찬¹, 장민호²

*Y. C. Chung(ychung@snut.ac.kr)¹, M. Chang²

¹ 서울산업대학교 금형설계학과, ²(주)솔루션닉스

Key words : Overlap Removal, Reverse Engineering, 3D Scanner

1. 서론

3차원 스캐너의 활용이 많아지고 있다. 3차원 스캐너는 문서나 그림을 스캔하듯이 3차원 실물을 스캔(혹은 촬영, 측정)해서 3차원 형상 정보를 얻을 수 있는 도구다. 측정 정밀도를 비롯한 전반적인 성능의 향상으로 얻어진 형상 정보를 제품 검사에 활용할 수 있는 수준에 이르렀다. 결과적으로 영화, 게임과 같은 영상 산업은 물론이고 일반적인 제조 산업에도 그 활용 범위가 넓어지고 있다[1].

백색광(white light)을 이용하는 면 단위 3차원 스캐너(areal 3D scanner)는 측정 대상물의 표면에 2차원 패턴을 가지는 빛을 투사한 후 그 패턴의 변형을 카메라로 관찰한다. 측정 좌표 값의 계산은 3각 측량에 기초한다. 백색광을 이용하는 면 단위 3차원 스캐너는 한 번의 측정 과정으로 한 장의 측정 데이터를 생성한다.

한 번의 측정으로 얻어진 한 장의 측정 데이터는 2차원 패턴이 한쪽 방향에서 투사된 곡면의 정보를 담고 있다. 따라서 완전한 3차원 형상을 측정하기 위해서는 여러 방향에서 2차원 패턴을 투사하는 여러 번의 측정이 필요하다. 여러 번의 측정은 여러 장의 측정 데이터를 생성한다. 여러 장의 측정 데이터는 필연적으로 중첩 영역을 가진다. 그리고 중첩 영역이 있어야 여러 장의 측정 데이터를 하나로 정렬할 수 있다. 측정 데이터의 중첩은 데이터 사이의 빈 공간 혹은 구멍을 방지한다. 일반적인 3차원 스캐닝 시스템은 여러 개의 측정 데이터를 하나 좌표계로 일치시키는 정렬(registration or align) 작업에 측정 데이터의 중첩 성질을 사용한다.

하나의 좌표계로 정렬된 데이터는 중첩이 존재하는 여러 개의 데이터이므로 중첩을 제거하고 하나의 데이터로 병합(merge)해야 한다. 여러 개의 측정 데이터를 하나로 병합하는 방법은 크게 두 가지로 분류된다. W.E. Lorensen과 H.E. Cline에 의해 제안된 marching cube 방법은 점군(point clouds)의 데이터를 하나의 삼각망으로 만든다[2]. 또 다른 방법은 Turk와 Levoy에 의해 제안된 zippering 방법이다[3]. 이 방법은 Zippering 방법은 중첩 영역을 평균화하고 제거한 후 두 측정 데이터를 지퍼를 채우듯이 깐다.

이 논문에서는 측정 데이터의 중첩을 제거하는 알고리즘을 제안한다. 각각의 측정 데이터는 격자 구조의 자료 구조에 넣어서 연산을 수행한다. 이 연구가 다른 연구와 다른 중요한 점이 측정 데이터를 격자 구조로 표현하고 연산을 수행하는 것이다. 기존의 다른 연구들은 측정 데이터를 삼각망으로 표현하고 중첩 제거 알고리즘을 삼각망 자료 구조로 구현하였다. 격자 구조는 삼각망 구조에 비해 더 적은 컴퓨터 메모리를 사용한다.

2. 전체적인 절차

Fig. 1은 제안하는 알고리즘의 4단계 주요 절차를 보여주고 있다. 제안하는 방법의 각 절차의 측정 데이터 단위로 수행된다. 첫 번째 단계에서는 대체 점들보다 더 나쁜 측정점들을 현재 측정 데이터에서 제거한다. 두 번째 단계에서는 첫 번째 단계에서 제거된 점들 중에 측정점 영역의 호수(작은 구멍) 혹은 만(작은 틈새)에 놓이는 점들을 되살린다. 세 번째 단계에서는 대체 점을 가지는 점들 중에 섬(작은 독립 영역) 혹은 곳(좁은 띠 영역)에 놓이는 점들을 제거한다. 마지막 네 번째 단계에서는 다른 측정 데이터에서 각 측정점의 대체점이 현재 측정 데이터에 있다면 그 점을 제거한다.

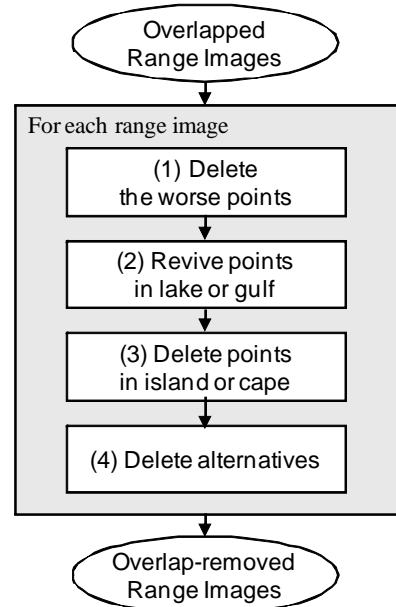


Fig. 1 Overall procedure

3. 측정 데이터 자료 구조

레인지 이미지(range image)는 카메라 초점(focal point)에서 각 이미지 점에 해당하는 물체까지의 거리 값이다. 레인지 이미지를 통해 각각의 이미지 점들은 실제 좌표 값으로 대응된다. Fig. 2는 이미지 점(image point)과 물체 점(object point)의 관계를 보여 준다. 이미지 점은 3차원 스캐너의 카메라 이미지 평면에 놓인다. 물체 점은 측정 대상물의 표면에 놓인 점을 세계 좌표계(world coordinate)로 표현한 것이다. 물체 점의 좌표 값은 이미지 점과 이미지의 거리 값(range data)으로 쉽게 계산된다. 해당 물체 점의 거리 값은 카메라 좌표계의 z좌표 값에 해당한다. Fig. 2에서 z_0 는 카메라 렌즈의 초점 평면에서 P_0 까지의 거리 값이다.

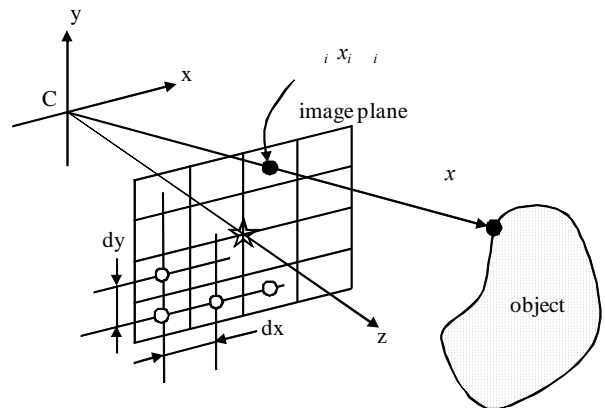


Fig. 2 Image point and object point

물체 점 P_0 는 초점 C와 이미지 점 P_i 가 만드는 직선위에 놓이기 때문에 P_0 의 x, y 좌표값은 식(4)와 같이 계산된다[4]. 식에서 f는 카메라의 초점 거리이고 z_0 는 P_0 의 거리 값이다.

$$P_0 = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = \begin{pmatrix} \frac{x_0 \cdot z_0}{f} \\ \frac{y_0 \cdot z_0}{f} \\ z_0 \end{pmatrix} \quad (1)$$

격자 표현법은 2차원 배열 구조에 각 이미지 점들의 거리 값을 저장한다. 이미지 점은 카메라의 2차원 픽셀에 대응된다. 카메라의 픽셀들은 등간격으로 배치되며 렌즈 중심에서 초점 거리(focal length)만큼 떨어진 평면에 놓인다. 격자 표현법은 픽셀의 위상 구조를 그대로 이어받는다.

4. 중첩 영역 제거

제안하는 알고리즘의 첫 번째 절차는 한 장의 측정 데이터에서 다른 측정 데이터에 있는 대체 점보다 나쁜 점들을 제거하는 것이다. 현재 측정 데이터에 속하는 어떤 점의 대체 점은 다음 두 가지 조건을 만족해야 한다. 먼저 현재 측정 데이터의 점을 다른 측정 데이터에 투영할 때 투영되는 격자에 속하는 점이어야 한다. 그리고 현재 점과 투영 점과의 거리는 미리 정의된 거리보다 적어야 한다. 현재 점과 대체 점의 비교를 위한 점의 평가 점수는 그 점의 법선 벡터와 카메라 벡터의 함수로 주어진다. 일반적으로 두 벡터의 사이각이 클수록 나쁜 점이며 사이각이 작으면 더 좋은 점이다.

측정 데이터에서 첫 번째 절차로 제거되지 않은 점들의 영역은 Fig. 3에서 보듯이 호수(lake, 작은 구멍)와 좁은 만(gulf, 좁은 틈새)을 가진다. 이것은 첫 번째 절차에서 인접 점들의 분포를 고려하지 않기 때문이다. 두 번째 절차에서는 첫 번째 절차에서 제거된 점들 중에 작은 호수 혹은 좁은 만에 속하는 점들을 되살린다. 어떤 제거된 점이 호수 혹은 만에 속하는지 여부는 그 점에서 제거되지 않은 점까지의 좌우(혹은 상하) 거리로 판단한다. 즉 좌로도 일정 거리 이내에 제거되지 않은 점이 존재하고 우로도 일정 거리 이내에 제거되지 않은 점이 존재한다면 그 점은 작은 호수 혹은 만에 속하는 점이다.

세 번째 절차에서는 현재 측정 데이터에서 대체 점을 가지는 점들 중에 섬(island, 독립된 작은 영역) 혹은 곳(cape, 좁은 띠 영역)에 속하는 점들을 제거한다. 이 절차에서는 첫 번째 절차에서와 같이 점들의 평가 점수를 고려하지 않는다. Fig. 4는 이 절차의 결과를 보여준다.

마지막 절차에서는 다른 측정 데이터에 존재하는 현재 측정 데이터의 대체점들을 제거한다.

제안된 중첩제거 알고리즘은 구현되어 다양한 예제들로 테스트되었다. Fig. 5는 중첩 제거의 한 예다.

후기

이 연구는 산업자원부 중기거점사업의 지원을 받았습니다.

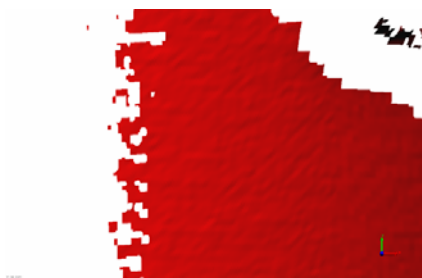


Fig. 3 After deleting points worse than alternatives



Fig. 4 After reviving lakes and deleting islands



a) overlapped data



b) overlap removed data

Fig. 5 Typical example of overlap removal

참고문헌

1. T. Varady, R.R. Martin and J. Cox, "Reverse engineering of geometric models -- and introduction," Computer-Aided Design, Vol. 29, pp. 255-268, 1997
2. W.E. Lorensen and H.E. Cline, "Marching cubes: a high resolution 3D surface construction algorithm," SIGGRAPH, pp. 163-169, 1987
3. G. Turk and M. Levoy, "Zippered polygon meshes from range images," SIGGRAPH, pp. 311-318, 1994
4. A. Srivastava, X. Liu and C. Heshner, "Face recognition using optimal linear components of range images," Image and Vision Computing, Vol. 24, pp. 291-299, 2006