

# 3 차원 광학식 스캐너 시스템의 격자형 자료 구조

\*정연찬<sup>1</sup>, 장민호<sup>2</sup>

<sup>1</sup>서울산업대학교 금형설계학과, <sup>2</sup>솔루션닉스

## Structured Data Structures for a 3D Scanner System

\*Yun Chan Chung<sup>1</sup>, Minho Chang<sup>2</sup>

<sup>1</sup> Dept. of Die and Mould Design, Seoul National Univ. of Technology, <sup>2</sup> Solutionix

Key words : Optical 3D Scanner, Data Structure, Structured Data

### 1. 서론

광삼각법을 이용해서 곡면 단위로 측정하는 3 차원 스캐너의 활용이 많아지고 있다. 곡면 단위로 측정하는 3 차원 스캐너 시스템은 한번에 많은 양의 측정 데이터를 생성한다. 컴퓨터 메모리는 제한적이므로 보다 적은 메모리로 측정점 데이터를 표현할 필요가 있다. 또 효율적인 연산을 위한 고려가 필요하다.

대용량의 측정점을 다루는 자료 구조와 관련한 기존 연구는 유한 요소 해석, 쾌속 조형(rapid prototyping), 가상 현실 비롯한 컴퓨터 그래픽스, 3 차원 측정 등의 연구 분야에서 주로 다루어졌다. 이들 자료는 정렬되지 않은 점을 다루기 때문에 점의 좌표와 함께 모서리, 면 등의 위상 정보도 함께 저장하여 렌더링 및 인접 관계 연산을 지원한다.

최근에는 정렬된 데이터를 생성하는 3 차원 스캐너가 많이 사용되는데, 본 연구에서는 정렬된 측정 데이터를 생성하는 3 차원 스캐너 시스템에서 측정 데이터를 적은 메모리 양으로 효율적으로 처리할 수 있는 컴퓨터 자료 구조를 비교 제안하려고 한다.

### 2. 측정 데이터 자료 구조의 특성과 조건

#### 2.1 측정점 계산 방법에 따른 데이터 특성

본 연구에서 고려하는 3 차원 스캐너는 위상이동 광삼각법으로 측정점을 계산하며 카메라의 픽셀 별로 측정 데이터를 생성한다. 따라서 측정 데이터는 다음과 같은 특성을 갖는다. 첫째, 이미지 평면상의 한 점  $P_i$  에 해당하는 측정점은 렌즈 중심과  $P_i$  를 잇는 연장선 상에 존재한다. 따라서,  $P_i$  에 해당하는 측정점의 깊이 정보만 정해지면, 원근 투영에 의하여 직교 좌표계상의 측정점이 유일하게 결정된다. 둘째, 카메라의 영상 소자들은 균일한 크기를 가지고 렌즈 중심으로부터 초점거리  $f$  만큼 떨어진 이미지 평면에 규칙적인 배열되어 있기 때문에, 인접한 두 개의 측정점을 이미지 평면에 원근 투영시키면, 이들 사이의 거리는 항상 영상소자의 크기와 같다. 셋째, 픽셀에 대응하는 각 측정점들은 픽셀이 가지고 있는 좌우 및 상하의 사각 격자 구조의 인접 관계(topology)를 이어 받는다. 측정점들은 카메라의 영상 소자처럼 물리적인 사각 격자 구조를 가지고 있지는 않지만, 인접한 두 픽셀에 해당하는 측정점 사이에는 다른 측정점이 존재하지 않기 때문에 측정점들도 인접 관계상으로는 사각 격자 구조를 갖는다.

#### 2.3 측정 데이터 처리 연산의 특성

3 차원 스캐너 시스템에서 계산된 측정점 데이터의 처리 방법과 그 순서는 측정의 목적에 따라 달라질 수 있다. 3 차원 스캐너 시스템에서 수행되는 측정 데이터 처리 연산의 일반적인 흐름을 Fig. 1 에 도시하였다. 데이터 처리 흐름에서 ‘깎기(zippering)’ 혹은 ‘합병(merge)’ 이전의 연산

들은 데이터의 구조적 특성(사각 격자 위상)을 유지할 수 있다. 그러나 ‘깎기’ 혹은 ‘합병’ 이후에는 삼각망 자료 구조가 일반적이다.

#### 2.4 측정 데이터 자료 구조의 요구 조건

측정 데이터 자료 구조는 요구 메모리 양이 적어야 한다. 한번의 측정으로 얻어지는 측정점이 백만 개 이상이고 완전한 3 차원 형상 정보를 얻기 위해서는 수십 혹은 수백 번의 측정이 필요하다. 따라서 가능한 메모리 양이 적은 자료 구조를 채택해야 한다.

연산 효율이 높아야 한다. 관련 기술의 발전으로 측정점의 개수가 지속적으로 늘어나고 사용자의 기대 수준이 높아지고 있다. 수억 개의 측정점을 빠른 시간에 처리 하기 위해서는 연산 효율을 고려한 자료구조가 필요하다.

측정점의 위상 정보를 유지해야 한다. 3 차원 스캐너에서 얻어지는 측정점 데이터는 개념적으로 사각 격자 구조를 가진다. 따라서 특정한 측정점의 좌우 혹은 상하 방향으로 인접한 측정점을 직접적으로 파악할 수 있다. 사각 격자 구조의 위상 정보를 가진 측정 데이터는 삼각화, 인접 점 찾기 등의 다양한 연산에 효율적이다.

### 3. 측정 데이터 자료 구조 비교

자료 구조의 소요 메모리 양을 편리하게 비교하기 위해 한번의 측정으로 최대 백만 개의 측정점이 얻어지는 3 차원 스캐너를 가정하자. 측정점의 좌표 값은 정밀도가 요구 되는 데이터이므로 배정도 실수(8 바이트)에 저장한다.

#### 3.1 측정점 배열 구조

측정점의 좌표 값을 3 차원 스캐너의 픽셀 수와 일치하는 2 차원 배열에 저장하는 방식이다. 이 방법은 2 차원 배열 구조가 측정점의 위상을 표현하고 각 배열 요소가 측정

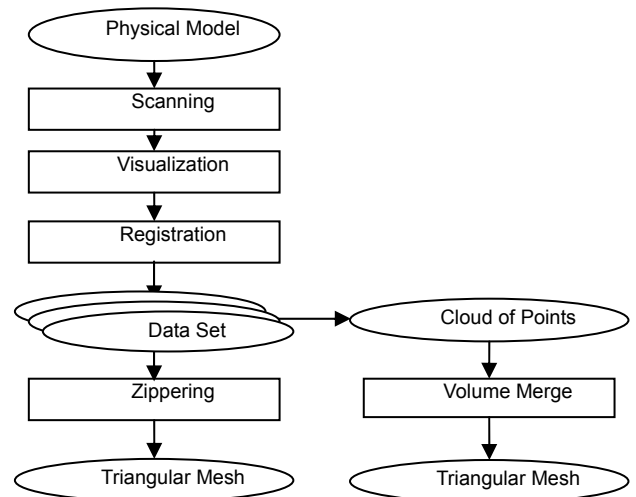


Fig.1 Process flow of the scanned data

점의 좌표 값을 표현한다. 측정점의 좌표 값을 추가적인 계산이나 메모리의 번지 이동 없이 배열에서 직접 조회할 수 있다.

각각의 배열 요소는 24 바이트이며 백만 개의 측정점을 저장하려면 24MB의 메모리가 필요하다. 한번에 백만 개의 측정점을 획득하는 3 차원 스캐너의 측정 데이터 한 세트를 저장하기 위해서는 24MB의 메모리가 소요된다.

### 3.2 포인터 배열 구조

측정점은 별도의 구조에 저장하고 측정점을 가리키는 포인터를 2 차원 배열에 저장하는 방식이다. 측정 데이터의 사각 격자 위상은 2 차원 배열이 표현하고 측정점의 좌표 값은 별도로 저장한다. 이 방식은 포인터를 저장할 4MB의 별도 메모리가 필요하다. 측정점을 저장하기 위해 24MB가 필요하므로 총 28MB의 메모리가 필요하다.

본 연구에서 대상으로 하는 3 차원 스캐너는 카메라의 매 픽셀마다 측정 데이터를 얻을 수 있지만, 일반적으로 3 차원 스캐너에서 얻어진 측정점의 개수는 최대 측정점의 수보다 작다. 이는 스캐너의 FOV를 벗어난 부분, 카메라의 사각 부분에서 데이터가 얻어지지 않기 때문이다. 다수의 측정 데이터를 분석한 결과 측정점의 유효율은 평균 35%로 나타났다. 유효한 측정점이 없는 2 차원 배열 요소에는 Null 포인터를 저장하고 유효한 측정점만 별도로 저장하면 필요 메모리 양이 줄어든다.

측정점의 유효율을  $e$ 라고 할 때 소요 메모리는 식(1)과 같다. 측정점의 유효율이 100%면 28MB의 메모리가 소요된다. 유효율이 35%라면 3.1 절에서 소개한 ‘측정점 배열 구조’가 요구하는 메모리의 절반인 12MB로 측정 데이터를 저장할 수 있다.

$$4MB + 24B \times 1M \times e = (4+24e)MB \quad (1)$$

### 3.3 깊이 값 배열 구조

스캐너 렌즈의 초점에서 측정점까지의 깊이를 2 차원 배열에 저장하는 방식이다. 측정점의 z-좌표 값만을 2 차원 배열에 저장한다. 측정점의 x-, y-좌표 값은 2 차원 배열에서 그 측정점이 저장된 인덱스를 이용해서 계산한다. 픽셀의 간격과 기준 픽셀의 위치를 알면 식(2)와 같이 2 차원 배열의 (I, J) 인덱스를 x-, y-좌표 값으로 쉽게 변환할 수 있다.

$$\begin{aligned} X_0 &= X_i \times Z / F \\ Y_0 &= Y_i \times Z / F \\ \text{Where } X_i &= X_0 + dX \times I \\ Y_i &= Y_0 + dY \times J \end{aligned} \quad (2)$$

$Z$ : stored depth value  
 $F$ : focal length  
 $X_0, Y_0$ : coordinate of pixel(0,0)  
 $dX, dY$ : pixel interval

이 방법은 2 차원 배열에 실수 하나만 저장하기 때문에 측정 데이터의 유효율에 관계없이 8MB의 메모리에 백만 개의 측정점을 저장할 수 있다.

### 3.4 격자형 자료 구조 비교

사각 격자형 자료를 저장할 세 가지 배열 자료 구조의 메모리 소요량과 하나의 측정점 계산에 필요한 연산의 회수를 표 1에 정리하였다. 메모리 공간 기준으로는 ‘깊이

값 배열’ 구조가 가장 적절한 자료 구조다. 연산 효율은 ‘측정점 배열’ 구조가 가장 좋다. ‘측정점 배열’ 구조는 추가적인 메모리 번지 이동이나 실수 연산 없이 측정점의 좌표 값을 직접 접근할 수 있다. ‘포인터 배열 구조’는 특정 측정점에 접근하기 위해서 항상 추가적인 메모리 번지 이동이 필요하다. ‘깊이 값 배열’ 구조에서 x, y, z 세 좌표 값을 얻기 위해서는 식(2)에서 보듯이 5 번의 실수 곱셈이 수행된다. 자료 구조의 연산 능력을 비교하기 위해 실제 구현을 통해 몇 가지 기본적인 연산을 수행해보았다. 측정 데이터의 크기와 연산의 종류에 따라 상이한 결과가 나타나 자료 구조의 성능을 평가하기 어려웠다. 그러나 세 자료 구조의 성능이 2 배 이상의 뚜렷한 차이는 없었다.

Table 1 Comparison of the data structures

| Type of Data Structure             | Memory for 1 million points | Operation for a point |
|------------------------------------|-----------------------------|-----------------------|
| Point Array                        | 24MB                        | direct access         |
| Pointer Array<br>(e = valid ratio) | e = 1.0                     | 1 memory address      |
|                                    | e = 0.35                    |                       |
| Depth Array                        |                             | 5 multiplications     |

## 4. 결과 및 과제

측정 데이터의 자료 구조가 갖춰야 할 요구 조건을 만족하는 세 가지 격자형 자료 구조를 제시 비교 하였다. 제시된 세 가지 자료 구조는 모두 사각 격자 위상을 가지기 때문에 위상에 근거한 연산을 빠르게 처리할 수 있다. 그리고 2 차원 배열 구조를 사용하기 때문에 원근 투영에 의한 인접 관계 계산도 가능하다. 제시된 세 가지 자료 구조에서 ‘깊이 값 배열’ 구조가 가장 적은 메모리 공간을 사용한다. 이 자료 구조는 한번에 최대 백만 개의 측정점을 획득하는 스캐너의 경우 2GB 메모리 한계를 갖는 일반적인 MS Windows PC에서 200 개 이상의 데이터 셋을 한꺼번에 처리할 수 있다. 논리적으로는 ‘깊이 값 배열’ 구조가 연산에 가장 많은 시간이 걸린다. 그러나 몇 가지 간단한 연산을 구현해서 비교한 결과 연산의 성능은 세 가지 자료 구조가 큰 차이를 보이지 않았다.

제시된 자료 구조를 이용해서 측정 데이터 처리에 필요한 연산을 구현하면 자료 구조의 성능과 적합성을 보다 잘 분석할 수 있을 것이다. 특히 원근 투영에 근거한 인접 관계 연산을 이용하면 측정 데이터를 보다 빠르고 정밀하게 처리할 수 있을 것이다.

### 후기

본 연구는 중기거점 차세대신기술개발사업의 연구비 지원으로 수행되었습니다.

### 참고문헌

1. Sun, W., Bradley, C., Zhang, Y.F. and Loh, H.T., "Cloud data modeling employing a unified, non-redundant triangular net," Computer-Aided Design, **33**, 183-193, 2001
2. Hradek, J., Kuchar, M. and Skala, V., 2003, "Hash functions and triangular mesh reconstruction," Computers and Geosciences, **29**, 741-751, 2003