원격 작업을 위한 3 차원 점군 데이터기반의 실시간 폴리곤 생성 및 텍스처 맵핑 기법

Real-time Polygon Generation and Texture Mapping for Tele-operation using 3D Point Cloud Data

장 가 람, 신 용 득, 윤 재 식, 박 재 한, 배 지 훈, 이 영 수, 백 문 홍*

(Ga-Ram Jang¹, Yong-Deuk Shin¹, Jae-Shik Yoon², Jae-Han Park¹, Ji-Hun Bae¹, Young-Soo Lee¹, and Moon-Hong Baeg¹)

¹Korea Institute of Industrial Technology (KITECH)

²University of Science and Technology (UST)

Abstract: In this paper, real-time polygon generation algorithm of 3D point cloud data and texture mapping for tele-operation is proposed. In a tele-operation, it is essential to provide more highly realistic visual information to a tele-operator. By using 3D point cloud data, the tele-operator can observe the working environment from various view point with a reconstructed 3D environment. However, there are huge empty space in 3D point cloud data, since there is no environmental information among the points. This empty space is not suitable for an environmental information. Therefore, real-time polygon generation algorithm of 3D point cloud data and texture mapping is presented to provide more highly realistic visual information to the tele-operator. The 3D environment reconstructed from the 3D point cloud data with texture mapped polygons is the crucial part of the tele-operation.

Keywords: real-time polygon generation, texture mapping, 3D point cloud data, tele-operation

I. 서론

원격 작업이란 로봇에 장착된 카메라 등의 센서를 통해 원격에 있는 작업자가 작업지의 환경 정보를 파악하고 작업지의 로봇을 제어하며 작업을 수행하는 것을 말한다. 지금까지의 원격 작업에서는 환경 정보를 얻기 위하여 영상 정보가 주요 수단으로 활용되었다. 영상 정보는 원격 작업자가 환경을 파악하기에 중요한 정보이지만 원활한 작업을 위한 영상 시점의 이동에 제약이 있으며 공간에 대한 거리 정보를 제공하지 않아 영상 정보만으로 원격 작업을 수행하기에 어려움이 있었다. 이러한 영상 정보의 한계를 극복하기 위해 3차원 레이저 스캐너 및 깊이 감지 카메라 등을 통해 획득한 3차원 공간 정보를 원격 작업에 적용하려는 시도가 많은 연구자들에 의해 활발하게 진행되고 있다[1-3].

3차원 레이저 스캐너 및 깊이 감지 카메라와 같은 3차원 공간 센서는 중심 좌표계를 기준으로 각 지점에 대한 공간의 거리를 측정하여 그에 해당하는 3차원 위치를 포함하는 점군(Point Cloud) 형태의 정보를 제공한다. 3차원 공간 센서가 제공하는 점군 데이터를 그대로 원격 작업자에게 3차원 형태로 시각화하면 자유로운 시점 이동이 가능해지고 공간에 대한 측정 정보를 제공함으로써 영상 정보의 문제점을 개선할 수 있으나 점들 사이에 빈 공간이 존재하는 점군 데이터의 특성으로 인해 가까운 시점에서는 환경의 식별이 어려운

문제점이 있다. 예를 들어 원격 작업자가 점군 데이터로 이루어진 재구성된 원격 작업지의 환경에 존재하는 물체를 관찰하며 상호작용을 하고자 한다면 해당 물체 가까이 시점을 이동해야 한다. 하지만 가까운 시점으로 이동할수록 물체를 이루는 점들 사이의 빈 공간이 더 크게 보이므로 물체의 식별이 어려워지고 원활한 원격 작업이 어렵게 된다. 만약 물체가 점군 데이터가 아닌 폴리곤으로 이루어져 있다면 가까운 시점으로 이동하더라도 물체의 모습을 관찰하고 상호작용을 할 수 있으므로 원활한 원격 작업을 가능하게 한다[4].

점군 데이터로부터 폴리곤의 생성은 컴퓨터 그래픽스 분야에서 많이 연구되고 있다[5,6]. 상기 분야에서의 폴리곤 생성 알고리즘은 연산 시간이 길고 원활한 작업을 위한 높은 하드웨어 성능이 요구된다. 특히 최근 출시된 Microsoft KINECT [7]와 함께 제공되는 SDK에서도 폴리곤을 생성하여 출력하는 샘플을 제공하고 있으나 Direct3D11 [8]에서 사용 가능한 지오메트리 셰이더(Geometry Shader)를 통하여 구현되고 있으며 이를 사용하기 위해 고성능의 최신 그래픽카드가 요구된다.

실시간으로 이루어지는 원격 작업은 원활한 작업을 위하여 원격 작업자에게 원격 작업지의 환경 정보를 실시간으로 처리하여 제공해야 한다. 컴퓨터 그래픽스에서 연구되는 알고리즘은 그 연산 시간이 길기 때문에 실시간으로 이루어지는 원격 작업에 적합하지 않으며 원격 작업에 사용되는 로봇의 내부 공간의 제약, 배터리의 제약등과 같은 하드웨어적인 제약으로 인해 고성능의 CPU와 그래픽 카드를 장착하여 사용할 수 없다.

본 논문에서는 원격 작업에 적합하고 점군 데이터의 제한성을 극복하여 원활한 원격 작업을 가능케 하기 위해 점군 데이터로부터 이미지 그리드에 기반한 실시간 폴리곤 생성

* 책임저자(Corresponding Author)

Manuscript received April 12, 2013 / revised August 19, 2013 / accepted September 2, 2013

장가람, 신용득, 박재한, 배지훈, 이영수, 백문홍: 한국생산기술연구원 (riverj@kitech.re.kr/syd@kitech.re.kr/hans1024@kitech.re.kr/joseph@kitech.re.kr/yslee@kitech.re.kr/mhbaeg@kitech.re.kr)

윤재식: 과학기술연합대학원대학교(jsyoon3@kitech.re.kr)

※ 이 논문은 2013 제 28회 ICROS 학술회에 초안이 발표되었음.

알고리즘을 제안한다. 생성한 폴리곤에 환경의 컬러 이미지 정보를 추가하여 폴리곤으로 재구성된 원격 작업지의 환경에 질감을 줄 수 있으며 원격 작업자에게 향상된 입장감을 제공할 수 있다. 또한 점군 데이터로부터 폴리곤의 생성 및 컬러 이미지 정보의 추가는 실시간으로 수행되므로 실시간으로 이루어지는 원격 작업에 적용하여 원활한 원격 작업을 가능하게 한다.

II. 실시간 폴리곤 생성 알고리즘

1. 점군 데이터의 획득

점군 데이터는 깊이 감지 카메라로부터 획득한 2차원 디스패리티(Disparity) 이미지로부터 깊이 보정, IR 카메라 보정 그리고 직교 좌표계 변환을 통하여 획득할 수 있다[9].

1.1 깊이 보정

깊이 감지 카메라로부터 획득한 거리 이미지는 각 픽셀들에 대한 공간의 상대적 거리정보인 디스패리티 d 를 가지고 있으나 디스패리티 정보는 정확한 거리정보가 아니므로 식 (1)의 보정 함수를 이용하여 실제 거리 정보를 구해야 한다 [10].

$$f(d) = \frac{Z_0}{1 + \frac{Z_0}{f_0 \cdot b} d} = \frac{1}{\frac{1}{Z_0} + \frac{1}{f_0 \cdot b} \cdot d} \quad (1)$$

여기에서 Z_0 는 기준평면, f_0 는 깊이 감지 카메라의 초점거리 그리고 b 는 깊이 감지 카메라의 IR 송신부와 수신부사이의 거리를 의미한다.

깊이 보정을 위해 실제 거리에 대한 디스패리티를 측정하여 깊이 보정 함수 내의 파라미터를 구해야 한다. 이를 위해 레이저 거리 측정기를 이용하여 정밀한 실제 거리 정보와 측정된 디스패리티의 관계 데이터를 구했으며 그 결과는 그림

그림 1. 디스패리티 값과 레이저 거리 측정기로 측정한 거리 값.

Fig. 1. Disparity vs. Measured distance.

표 1. 디스패리티 모델의 파라미터 값.

Table 1. Estimated parameters of the disparity model.

	$1/Z_0$	$1/(f_0 \cdot b)$
Value	3.1674	-0.0029
Standard Error	0.00295	2.96884E-6

1과 같다. 그림 1의 측정 정보와 최소 제곱 최적화 방법을 적용한 식 (1)의 모델을 통해 깊이 보정 함수의 파라미터를 구할 수 있었으며 그 결과는 표 1과 같다.

1.2 IR 카메라 보정

깊이 보정을 통하여 획득한 z [m]로부터 x [m], y [m]를 획득하기 위하여 IR 카메라의 초점거리 ($c_{ir,x}, c_{ir,y}$), 광축의 중심 좌표 ($f_{ir,x}, f_{ir,y}$) 값들이 필요하며 이 값들은 IR 카메라 보정을 통하여 획득할 수 있다[11]. 본 논문에서는 Bouguet이 개발한 카메라 캘리브레이션 소프트웨어[12]를 통해 위 값들을 구했으며 그 결과는 표 2와 같다.

1.3 직교 좌표계 변환

깊이 보정과 IR 카메라 보정을 통하여 획득한 값들은 식 (2)의 핀홀(Pin-Hole) 카메라 모델을 이용하여 직교 좌표계로 변환할 수 있다[9,13].

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = f(d) \cdot \begin{bmatrix} \frac{1}{f_{ir,x}} & 0 & -\frac{c_{ir,x}}{f_{ir,x}} \\ 0 & \frac{1}{f_{ir,y}} & -\frac{c_{ir,y}}{f_{ir,y}} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} \quad (2)$$

여기에서 (i, j) 는 2차원 디스패리티 이미지의 픽셀 좌표이다.

그림 2는 2차원 디스패리티 이미지로부터 획득한 점군 데이터를 보여준다.

2. 이미지 그리드 기반 실시간 폴리곤 생성 알고리즘

폴리곤(Polygon)은 3개의 정점(Vertex)들로 구성된 삼각형이며 이러한 폴리곤들이 모여 복잡한 물체를 표현할 수 있는 메쉬(Mesh)를 구성한다. 하나의 정점은 점군 데이터를 이루는 하나의 점(Point)에 대응되며 그림 3은 3개의 정점들로 구성된 폴리곤과 이러한 폴리곤들이 모여 구성된 메쉬(자동차)를 보여준다.

하나의 폴리곤을 생성하기 위하여 점군 데이터로부터 3개의 정점 후보가 필요하며 이러한 후보는 깊이 감지 카메라로

표 2. IR Camera 보정 값.

Table 2. Calibrated IR camera variables.

	$c_{ir,x}$	$c_{ir,y}$	$f_{ir,x}$	$f_{ir,y}$
Value	320.17	260.00	582.64	586.97

그림 2. 3차원 점군 데이터.

Fig. 2. 3D Point Cloud Data.

그림 3. 정점, 폴리곤, 메쉬.
Fig. 3. Vertex, Polygon, Mesh.

그림 4. 2차원 디스패리티 이미지의 그리드 분할.
Fig. 4. Grid-based 2D disparity image.

그림 5. 하나의 블록과 4개의 정점 후보.
Fig. 5. Four vertex candidates per block.

표 3. 3개의 정점 후보.
Table 3. Three vertex candidates.

폴리곤	정점 후보(시계방향)
1	0, 1, 2
2	2, 1, 3

부터 획득한 2차원 디스패리티 이미지를 그리드(Grid) 기반으로 분할한 후 선택한다. 그림 4는 10 x 10으로 분할된 2차원 디스패리티 이미지를 보여준다.

그림 5와 같이 그리드로 분할된 2차원 디스패리티 이미지를 구성하는 하나의 블록으로부터 2개의 폴리곤 생성을 위한 4개의 정점 후보를 선택할 수 있으며 하나의 폴리곤은 3개의 정점 후보를 갖는다. 표 3은 하나의 폴리곤 생성을 위

그림 6. 폴리곤 생성에 적합하지 않은 정점 후보(0).
Fig. 6. Unsuitable vertex candidate '0' for a polygon.

그림 7. 가까운 시점에서 본 점군 데이터(a), 체크보드 위에 놓인 박스(b), 점군 데이터(c).
Fig. 7. Point cloud data from close view(a), Actual object of point cloud data(b), Point cloud data(c).

한 3개의 정점 후보를 나타낸다.

하나의 폴리곤을 이루는 3개의 정점 후보들은 깊이 보정과 IR 카메라 보정을 통하여 점군 데이터를 이루는 정점으로 변환된다. 이 때 변환된 3개의 정점 후보들을 연결하여 하나의 폴리곤을 생성할 수 있으나 각 후보들이 폴리곤을 생성하기에 적합한지 확인해야 하므로 정점 후보들에 대한 거리 테스트를 수행하였다.

2차원 디스패리티 이미지는 디스패리티 정보의 크기에 따라 색을 지정하여 시각적으로 확인할 수 있도록 한 이미지이다. 그림 4에서 밝게 표현된 부분은 디스패리티가 작은 값이며 가까운 거리를 의미하고 어둡게 표현될수록 멀리 떨어진 거리를 의미한다. 그림 5의 하나의 블록으로부터 폴리곤 2를 생성하기 위한 3개의 정점 후보 2, 1, 3은 그 색이 유사하므로 이를 정점으로 변환하면 이웃한 3개의 정점이 되고 하나의 폴리곤을 생성할 수 있다. 만약 하나의 정점이 다른 두 개의 정점보다 멀리 떨어져 있다면 서로 이웃하지 않아 폴리곤을 생성하는데 적합하지 않은 정점이 된다.

그림 6은 그림 5의 블록을 확대한 그림을 보여준다. 폴리곤 1을 생성하기 위한 정점 후보 0, 1, 2에서 정점 후보 0은 정점 후보 1, 2 보다 어둡게 표현되며 정점으로 변환하면 정점 0은 정점 1, 2보다 멀리 떨어져 서로 이웃하지 않은 정점이 되므로 폴리곤 1을 생성하는데 적합하지 않은 것으로 판

그림 8. 정점에 대한 거리 테스트 적용(a), 정점에 대한 거리 테스트 미적용(b).

Fig. 8. Applying depth test(a), Not applying depth test(b).

정한다.

그림 7(a)는 그림 7(c)의 붉은색 사각형 영역을 위에서 바라본 모습이다. 그림 7(a)의 노란색 사각형 영역은 그림 7(b)의 체크보드 위에 놓인 박스의 윗부분에 대한 노란색 영역에 해당되며, 그림 7(a)의 붉은색 사각형 박스 영역은 그림 7(b)의 붉은색 영역(박스와 체크보드가 만나는 지점)에 해당된다.

그림 7(a)의 검은색으로 표현된 영역은 점군 데이터가 없는 것을 의미하며 노란색과 붉은색 영역은 서로 떨어져 있으므로 폴리곤을 생성하지 말아야 한다. 그림 8(a)는 정점에 대한 거리 테스트를 적용한 결과를 보여주며 파란색 사각 영역에 폴리곤이 생성되지 않은 것을 확인할 수 있다. 그림 8(b)는 정점에 대한 거리 테스트를 적용하지 않은 결과이며 동일한 영역에 폴리곤이 생성된 것을 확인할 수 있으며 이러한 폴리곤은 비어있어야 할 공간을 채워 정확하지 않은 환경 정보를 제공하게 된다.

본 논문에서는 2차원 디스패리티 이미지를 320 x 240의 그리드로 분할하여 76,800개의 블록을 생성하였다. 각 블록들에 대하여 2개의 폴리곤 생성을 위한 4개의 정점 후보를 선택한 후 각각의 폴리곤에 대한 정점 후보 3개를 정점으로 변환하여 거리 테스트를 수행하였다. 그림 9는 이미지 그리드 기반 실시간 폴리곤 생성 알고리즘에 대한 의사코드를 보여준다.

3. 2차원 컬러 이미지 맵핑(Mapping)

거리 테스트를 통하여 선택된 3개의 정점으로 생성한 폴리곤에 2차원 컬러 이미지를 추가하여 폴리곤으로 재구성된 원격 작업지 환경에 질감을 주고 원격 작업자에게 향상된 입장감을 제공할 수 있다.

그림 9. 이미지 그리드 기반 실시간 폴리곤 생성 알고리즘(의사코드).

Fig. 9. Grid-based real-time polygon generation algorithm (pseudo-code).

```
function POLYGONGENERATION
  for all vertex candidates do
    current disparity coordinate ← visited
```

```
    block size = 2 x 2
    FIND_BLOCK(current disparity coordinate,
    block size, current four vertex candidates per
    block)
    four vertex candidates ← current four vertex
    candidates per block
    FIND_POLYGON_CANDIDATE(four vertex
    candidates, current polygon candidate lists)
    polygon candidate lists ← current polygon
    candidate lists
    CREATE_POLYGON(polygon candidate lists,
    polygon)
    polygon lists ← polygon
  end for
end function

function FIND_BLOCK(current disparity coordinate, block
size, four vertex candidates per block)
  for current disparity coordinate do
    vertex candidate ← current disparity coordinate
    + block size
    four vertex candidates per block ← vertex
    candidate
  end for
end function

function FIND_POLYGON_CANDIDATE(four vertex
candidates, current polygon candidate lists)
  for all four vertex candidates do
    current vertex candidate ← visited
    CONVERT_TO_POINTCLOUD(current
    vertex candidate, current vertex)
    current polygon candidate lists ← current
    vertex
    CONVERT_TO_UV(current vertex candidate,
    current UV coordinate)
    current polygon candidate lists ← current UV
    coordinate
  end for
end function

function CREATE_POLYGON(polygon candidate lists,
polygon)
  for polygon candidate lists do
    current polygon candidate ← visited
    for current polygon candidate do
      DEPTH_TEST(current polygon
      candidate, bool)
      if bool = true then
        polygon ← current polygon
        candidate
      else
        continue
      end if
    end for
  end for
end function

function DEPTH_TEST(current polygon candidate, bool)
  for all z of vertex of current polygon candidate do
    if abs(z_0 - z_1) <= ε and abs(z_0 - z_2) <= ε
    and abs(z_2 - z_1) <= ε then
      bool ← true
    else
      bool ← false
    end if
  end for
end function
```


폴리곤을 구성하는 3개의 정점은 3차원 직교 좌표이며 2차원 컬러 이미지는 2차원 픽셀좌표이다. 2차원 컬러 이미지를 3차원 직교 좌표에 맵핑하기 위하여 RGB 카메라와 IR 카메라의 스테레오 보정[11]을 수행한 후 2차원 디스페리티 이미지에 2차원 컬러 이미지를 맵핑하여 보정된 2차원 컬러 이미지를 획득한다. 보정된 2차원 컬러 이미지의 픽셀좌표는 2차원 디스페리티 이미지의 픽셀좌표와 같으므로 정점 후보와 같은 픽셀좌표를 갖는다. 따라서 폴리곤 생성을 위한 3개의 정점 후보를 정점으로 변환한 후 해당 정점에 보정된 2차원 컬러 이미지의 픽셀 좌표를 1:1 대응하여 맵핑할 수 있다.

2차원 컬러 이미지를 맵핑한 폴리곤으로 재구성된 원격 작업지 환경을 출력하기 위해서 Direct3D 또는 OpenGL과 같은 3D API를 사용해야 하며 본 논문에서는 Direct3D를 사용하여 출력하였다[14]. 폴리곤 생성을 위해 변환된 정점에 보정된 2차원 컬러 이미지의 픽셀 좌표를 맵핑하기 위해 해당 픽셀 좌표를 Direct3D에서 사용하는 텍스처(Texture)의 UV 좌표로 변환해야 하며 이 과정은 폴리곤 생성 알고리즘 내부에서 함께 수행된다.

보정된 2차원 컬러 이미지는 Direct3D의 텍스처로 변환되며 변환 시 0.25 [ms]의 연산 시간이 소요되므로 4.12 [ms]의 연산 시간에 생성되는 메쉬에 실시간으로 텍스처를 추가하여 출력할 수 있다. 그림 10(a)는 제안하는 알고리즘으로 생성한 폴리곤들이 모인 메쉬를 보여준다.

그림 10(a)의 메쉬는 바닥과 물체의 식별이 어렵고 무엇에 대한 메쉬인지 판별이 어렵지만 보정된 2차원 컬러 이미지를 맵핑한 그림 10(b)에서는 바닥에 놓여있는 체크보드와 그 위에 놓여진 박스, KITECH 로고를 확인할 수 있다.

그림 11은 그림 10의 메쉬를 가까운 시점에서 바라본 것이다.

그림 10. 메쉬(a), 보정된 2D 컬러 이미지 맵핑(b).

Fig. 10. Mesh(a), Texture mapping with calibrated 2D color image(b).

그림 11. 메쉬(가까운 시점)(a), 보정된 2D 컬러 이미지 맵핑(가까운 시점)(b).

Fig. 11. Mesh(close view)(a), Texture mapping with calibrated 3D color image(close view)(b).

그림 11(a)는 그림 10(a)와 동일하게 바닥과 그 위의 물체의 식별이 어렵지만 그림 11(b)의 경우 체크보드 위에 놓여진 박스와 KITECH 로고 및 로고 밑의 글자를 확인할 수 있다.

III. 실험 결과

본 논문에서는 제안하는 알고리즘과 컴퓨터 그래픽스 알고리즘의 연산 시간 및 폴리곤 생성 품질에 대한 비교를 수행하였으며 재구성된 환경을 점군 데이터로만 출력하였을 경우와 폴리곤을 생성하고 텍스처를 맵핑하여 출력한 경우에 대한 비교를 수행하였다. 그 결과는 아래와 같다.

1. 연산 시간 비교

동일한 환경에 대한 307,200개의 점군 데이터를 획득하여 제안하는 알고리즘과 컴퓨터 그래픽스의 Alpha Shape [5], Ball Pivoting [6] 알고리즘의 연산 시간을 비교하였다. 실험에 사용한 시스템은 Intel Core i7-2600K(CPU), 16GB(RAM)이며 단일 스레드를 기준으로 수행하였다. 그림 12는 실험에 사용한 점군 데이터와 2차원 컬러 이미지이다.

각 알고리즘에 대한 연산시간을 30회씩 측정하였으며 그 결과는 표 4와 같다.

Alpha Shape와 Ball Pivoting 알고리즘은 각각 10,801 [ms]와 733,655 [ms]의 연산 시간을 보여준다. 특히 Ball Pivoting 알고리즘의 경우 약 12분의 연산 시간을 보여주며 두 알고리즘 모두 실시간 연산에 적합하지 않은 것을 확인할 수 있다. 제안하는 알고리즘은 4.12 [ms]의 연산 시간을 보여주므로 33 [ms]마다 갱신되는 깊이 감지 카메라의 데이터를 실시간으로 처리할 수 있다.

2. 폴리곤 생성 품질 비교

컴퓨터 그래픽스의 알고리즘은 연산 시간이 길지만 생성한 폴리곤의 품질은 제안하는 알고리즘 보다 나은 결과를 보여준다. 그림 13은 각 알고리즘들에 대한 폴리곤 생성 결과를 와이어프레임(Wireframe)으로 보여주고 있으며 그림 13(a)~(c)는 그림 13(d)의 붉은색 사각형 영역을 왼쪽에서 바라본 모습이다.

그림 13(a)는 Alpha Shape 알고리즘의 결과이며 그림 13(b)는 Ball Pivoting 알고리즘의 결과를 보여준다. 두 알고리즘 모두

그림 12. 점군 데이터(a)와 2차원 컬러 이미지(b).

Fig. 12. Point cloud data(a) and 2D color image(b).

표 4. 알고리즘의 연산 시간.

Table 4. Running time of the algorithms per loop.

	제안하는 알고리즘	Alpha Shape	Ball Pivoting
연산시간[ms]	4.12	10,801	733,655
생성한 폴리곤 수 [개]	154,236	244,623	568,774

그림 13. Alpha Shape(a), Ball Pivoting(b), 제안하는 알고리즘(c), 점군 데이터(d).

Fig. 13. Alpha Shape(a), Ball Pivoting(b), Grid-based algorithm(c), Point cloud data(d).

제안하는 알고리즘(c)보다 모서리 부분의 폴리곤이 부드럽게 생성된 것을 볼 수 있으며 생성된 폴리곤들도 더 작게 생성된 것을 볼 수 있으나 모서리를 제외한 나머지 부분은 큰 차이를 보이지 않는다. 3가지 알고리즘 모두 생성한 폴리곤을 통해 전체적인 물체의 모습을 확인할 수 있으며 그림 8(a)에서와 같이 폴리곤을 생성하지 말아야 하는 영역에 대하여 폴리곤이 생성되지 않은 것을 볼 수 있다.

3. 점군 데이터와 폴리곤으로 이루어진 데이터와의 비교

그림 14(a)는 점군 데이터를 이루는 정점들에 2차원 컬러 이미지의 RGB값을 맵핑한 것이며 그림 14(b)는 점군 데이터로부터 생성한 메쉬에 텍스처를 맵핑한 것을 보여준다. 그림 14(a), (b) 모두 체크보드와 그 위에 놓인 물체를 확인할 수 있

그림 14. RGB 정보가 추가된 점군 데이터(a), 텍스처가 추가된 메쉬(b), RGB 정보가 추가된 점군 데이터(가까운 시점)(c), 텍스처가 추가된 메쉬(가까운 시점)(d).

Fig. 14. Colored point cloud data(a), Texture mapped mesh(b), Colored point cloud data with close view(c), Texture mapped mesh with close view(d).

으나 그림 14(a)는 점군 데이터로만 이루어져 있으므로 점군 데이터 사이의 빈 공간은 검은색 줄무늬처럼 출력된다. 그림 14(b)는 메쉬에 텍스처가 맵핑된 것이므로 점들 사이의 빈 공간이 존재하지 않으며 그림 14(a)보다 선명하게 출력된 것을 확인할 수 있다.

그림 14(c), (d)는 그림 14(a), (b)를 가까운 시점에서 바라본 것이다. 그림 14(c), (d)의 붉은색 사각 영역은 서로 동일한 위치를 나타내지만 그림 14(c)는 점들 사이의 빈 공간으로 인해 체크보드 위의 물체와 물체 뒤로 보이는 체크보드 무늬가 중첩되어 출력되므로 체크보드와 물체의 식별이 용이하지 않다. 하지만 그림 14(d)는 물체 뒤의 체크보드 무늬가 보이지 않으며 체크보드와 물체의 식별이 용이하고 물체에 적합한 KINECT [7]의 글자와 그림을 모두 확인할 수 있고 앞부분에 위치한 KITECH로고와 글자 모두 식별이 가능하다.

4. 원격 작업과 이미지 그리드 기반 실시간 폴리곤 생성 알고리즘

그림 15는 원격 작업지에 위치한 이동 로봇으로부터 원격 작업자의 워크스테이션까지 데이터가 전송되는 과정을 보여준다.

이동 로봇에서 획득한 디스패리티 정보와 보정된 RGB 이미지는 TCP Socket을 통하여 원격 작업자의 워크스테이션까지 전송되며 전송된 디스패리티 정보는 깊이 보정, IR 카메라 보정 및 핀홀 카메라 모델을 통하여 점군 데이터로 변환된다. 그림 16은 원격 작업자의 워크스테이션에서 점군 데이터가 메쉬로 변환되는 과정을 보여주며, 이러한 메쉬에 보정된 RGB 이미지로부터 생성한 텍스처를 추가한 후 3차원으로 시각화한다. 원격 작업자는 시점 제어 명령을 통해 다양한 시점에서 재구성된 3차원 환경을 관찰한다.

그림 17은 이동 로봇의 원격 제어에 적용한 이미지 그리드 기반 실시간 폴리곤 생성 알고리즘을 보여주며 3차원으로 시각화 한 메쉬정보를 원격 작업자에게 제공함으로써 원격 작업자는 메쉬로 재구성된 환경을 다양한 시점에서 관찰하며 원격 작업을 수행할 수 있다.

그림 15. 데이터 전송 다이어그램.

Fig. 15. Data transfer diagram.

그림 16. 원격 워크스테이션에서 텍스처가 추가된 메쉬의 생성.

Fig. 16. Texture mapped mesh generation on remote workstation.

그림 17. 원격 작업에 적용한 이미지 그리드 기반 실시간 폴리곤 생성 알고리즘.

Fig. 17. Real-time polygon generation algorithm implemented in tele-operation.

IV. 결론

본 논문에서는 원격 작업자의 원활한 원격 작업을 위하여 3차원 점군 데이터에 대한 실시간 폴리곤 생성 및 텍스처 맵핑 알고리즘을 제안하였다. 3차원 점군 데이터를 시각화하여 가까운 곳으로 시점을 이동하게 되면 점군 사이의 빈 공간으로 인해 원격 작업자의 환경 인지가 어려워지므로 3차원 공간 센서로부터 획득한 3차원 점군 데이터의 실시간 시각화에 있어 폴리곤 생성 및 텍스처 맵핑은 매우 중요하다. 그러나 기존의 컴퓨터 그래픽스에서 연구된 알고리즘들은 많은 연산 시간이 소요되고 높은 하드웨어 성능을 필요로 하여 실시간성과 하드웨어 구성에 제약이 있는 원격 작업에는 적합하지 않음을 확인하였다. 따라서 3차원 공간 센서로부터 획득한 환경 정보를 실시간으로 3차원 시각화하여 원격 작업에 적용하기 위해 본 논문에서는 알고리즘의 실시간 성능에 중점을 둔 이미지 그리드 기반 폴리곤 생성 알고리즘과 폴리곤 정보를 이용한 텍스처 맵핑 알고리즘을 제안하였으며 그 결과를 제시하였다.

제안하는 알고리즘을 통하여 307,200개로 이루어진 점군 데이터로부터 154,236개의 폴리곤을 생성하였으며 연산 시간은 4.12 [ms]이므로 33 [ms]마다 갱신되는 점군 데이터를 실시간으로 처리할 수 있었다. 또한 폴리곤을 이루는 정점들에 대한 거리 테스트를 수행하여 환경의 재구성성에 적합하지 않은 폴리곤을 생성하지 않음으로써 보다 정확하게 원격 작업지 환경을 재구성할 수 있었으며 생성한 폴리곤에 환경의 2차원 컬러 이미지 정보를 추가하여 폴리곤으로 재구성된 원격 작업지 환경에 질감을 주어 원격 작업자에게 향상된 입장감을 제공할 수 있었다. 실시간으로 이루어지는 원격 작업에 제안하는 알고리즘을 적용하면 원활한 원격 작업을 가능케 할 것으로 기대된다.

REFERENCES

- [1] ROS, "PR2," <http://www.ros.org/>.
- [2] PCL, "Point Cloud Library," <http://pointclouds.org/>.
- [3] G R. Jang, Y. D. Shin, J. S. Yoon, J. H. Bae, J. H. Park, and M. H. Baeg, "3D-based visual assistance system for teleoperation,"

ICCAS 2012, pp. 1998-2000, Oct. 2012.

- [4] G R. Jang, Y. D. Shin, J. S. Yoon, J. H. Park, and M. H. Baeg, "Grid-based real-time polygon generation algorithm of a 3D point cloud for a tele-operation," *Proc. of 2013 28th ICROS Annual Conference (in Korean)*, pp. 387-388, May 2013.
- [5] H. Edelsbrunner, "Three-dimensional alpha shapes," *ACM Trans. on Graphics*, vol. 13, no. 1, pp. 43-72, Jan. 1994.
- [6] F. Bernardini, "The ball-pivoting algorithm for surface reconstruction," *IEEE Trans. on Visualization and Computer Graphics*, vol. 5, no.4, pp. 349-359, Dec 1999.
- [7] Microsoft, "KINECT for windows," <http://www.microsoft.com/en-us/kinectforwindows/>.
- [8] Microsoft, "Direct3D11," <http://msdn.microsoft.com/enus/library/windows/desktop/ff476342.aspx/>.
- [9] Y. D. Shin, G R. Jang, J. H. Park, and M. H. Baeg, "Three-dimensional point clouds data acquisition from a depth sensing camera," *Proc. of KSPE 2011 Spring Conference (in Korean)*, pp. 99-100, Jun. 2011.
- [10] F. Menna, F. Remondino, R. Battisti, and E. Nocerino, "Geometric investigation of a gaming active device," *Proc. of SPIE*, Prague, Czech Republic, pp. 19-22, Sep. 2011.
- [11] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," *Proc. of the IEEE Int. Conf. on Computer Vision*, vol. 1, pp. 666-673, Sep. 1999.
- [12] Camera Calibration Toolbox for Matlab. Available online: http://www.vision.caltech.edu/bouguetj/calib_doc/.
- [13] J. H. Park, Y. D. Shin, J. H. Bae, and M. H. Baeg, "Spatial uncertainty model for visual features using a Kinect™ sensor," *Sensors*, vol. 12, no. 7, pp. 8640-8662, Jun. 2012.
- [14] G R. Jang, Y. D. Shin, J. H. Park, and M. H. Baeg, "Efficient 3D point cloud rendering for tele-operator : real-time and large scale data," *Proc. of 2012 27th ICROS Annual Conference (in Korean)*, pp. 382-383, Apr. 2012.

장 가 람

2008년 가천대학교 물리학과 학사. 2011년~현재 한국생산기술연구원 로봇융합연구그룹 연구원. 관심분야는 3차원 정보 처리 및 3차원 시각화 기술.

신 용 득

2007년 부산대학교 전자공학과 학사. 2009년 과학기술연합대학원대학교 지능형로봇공학과 석사. 2009년~현재 한국생산기술연구원 로봇융합연구그룹 연구원. 관심분야는 3차원 환경 인식 및 3차원 맵핑.

윤 재 식

2012년 서울과학기술대학교 자동차공학과 학사. 2012년~현재 과학기술연합대학원대학교 지능형로봇공학과 석사 과정. 관심분야는 3차원 정보 처리 및 동작 계획.

박 재 한

1998년 동아대학교 전자공학과 학사.
2000년 부산대학교 전자공학과 석사.
2010년 고려대학교 전기전자전파공학과
박사 수료. 2004년~현재 한국생산기술연
구원 로봇융합연구그룹 선임연구원. 관
심분야는 3차원 정보 처리, 인식기반
로봇 작업.

배 지 훈

1999년 명지대학교 전기전자공학과 학
사. 2001년 동 대학원 석사. 2004년
Ritsumeikan 대학교 로보틱스학과 박사.
2009년~현재 한국생산기술연구원 로봇
융합연구그룹 선임연구원. 관심분야는
양팔 로봇 작업, 로봇 손 제어.

이 영 수

서울대학교 산업공학과 학사. 한국과학
기술원 석사. University of Wisconsin -
Madison 기계공학 박사. 한국생산기술
연구원 수석연구원. 관심분야는 극한환
경로봇, 원격작업로봇.

백 문 홍

1982년 서울대학교 제어계측공학과 학
사. 1984년 동 대학원 석사. 1995년 동경
대학교 전기 전공 박사. 1996년~현재 한
국생산기술연구원 로봇융합연구그룹 수
석연구원. 관심분야는 로봇 비전, 3차원
공간 인지.