DBpia

집적영상 시스템에서 룩업테이블을 사용한 요소영상 배열의 효과적인 생성

정회원 권 영 만*, 김 승 철**, 종신회원 김 은 수**

Generating the Array of Elemental Image efficiently by using Look-Up Table in Integral Imaging System

Young-Man Kwon*, Seung-Chul Kim** *Regular Members*, Eun-Soo Kim** *Lifelong Member*

요 약

본 논문에서는 컴퓨터적으로 생성하는 집적영상 시스템에서 룩업테이블(Look-up Table)을 사용해서 요소영상 배열을 효과적으로 생성하는 알고리즘을 제안한다. 이 알고리즘은 물체의 점이 픽업 면에 투영되는 위치를 찾기 위해서 x와 y의 투영 점에 대해서 서로 독립적인 룩업테이블을 작성하고 이를 사용해서 요소영상 배열을 생성하는 계산 속도를 개선하였다. 제안된 알고리즘과 기존 알고리즘들의 계산 시간을 실험을 통해서 비교한 결과 제안된 알고리즘이 더 효율적임을 증명하였다.

Key Words : Elemental image array, Depth information, Lens array, Integral imaging system, 3D image

ABSTRACT

In this paper, we propose the algorithm for generating the array of elemental image by using look-up table (LUT) in a computer generated integral imaging system. It makes the LUT independently for the projection point of x and y. The algorithm using LUT to the existing ones needs less computing time to generate the array of elemental image. By comparing the computing time of proposed algorithm with that of the existing algorithms experimently, we proved the efficiency of proposed algorithm.

Ⅰ. 서 론

3차원 입체영상 기술은 향후 기존의 2차원 디지털 방송, 통신을 대신할 수 있는 차세대 실감 영상미디어로 기대되고 있다. 또한, 원격의료, 가상현실, 영화, 게임 산업 등에서도 큰 영역을 차지할 것으로 예상되고 있다. 그동안 다양한 방법의 3차원 영상 디스플레이 방식이 제안되거나 실제로 여러 응용에서 활용되고 있다[1]. 특히, 1908년에 Lippmann에 의해서 처음 제안된 집적영상 방식은 백색광을 사용하면서도 자연스러운 3차원 영상을 저장하고 재생할 수 있다는 장점으로 활발한 연구가 진행되고 있다[1-8].

한편 3차원 방송에서 3차원 정보를 어떻게 전송할 것인가는 큰 이슈이다. 즉, 각 가정에서 사용하는 3차원 TV의 특성이 모두 같을 것인가 하는 것이다. 예를 들어 어느 가정은 스테레오 방식의 3차원 TV를 사용하고, 다른 곳은 집적 영상 방식의 3차원 TV, 또 다른 곳은 홀로그램 방식의 3차원 TV를 사용할 수 있을 것이다. 즉 3차원 방송 서비스에서 3차원 정보를 전송할 때 어떤 방법으로 전송할 것인가를 고

※ 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 육성·지원사업 (IITA-2008-C1090-0801-0018) 지원에 의하여 수행되었습니다.

* 을지대학교 의료산업학부 시스템 연구실(ymkwon@eulji.ac.kr), ** 광운대학교 차세대 3D 디스플레이 연구센터(sckim@kw.ac.kr)

논문번호 : kics2008-08-368, 접수일자 : 2008년 8월 25일, 최종논문접수일자 : 2008년 10월 28일

www.dbpia.co.kr

려해야 한다. 이를 고려한 것 중의 하나가 2002년부터 2004년까지 유럽에서 수행된 ATTEST(Advanced Three-Dimensional Television System Technologies)과제이다. 이 과제에서는 3DTV 방송을 위하여 실제 방송환경에서 2D 비디오 영상과 이에 동기화된 깊이정보를 획득하여 이를 전송하고 이로부터 각 가정의 TV 방식에 따라 스테레오 영상, 다시점 영상, 홀로그램 영상 등을 생성하여 3차원 방송을 보는 것이다[2]. 즉, 스테레오 영상, 다시점 영상, 집적 영상, 홀로그램 영상 등을 전송하기에는 많은 대역폭이 필요하고, 각각의 3DTV의 특성이 다르기 때문에 적은 데이터 양으로 전송을 할 수 있고, 이로부터 다양한 방법의 3차원 영상을 만들어 낼 수 있는 2차원 영상정보와 이에 대응하는 깊이정보로 이루어진 3차원 정보를 전송하고, 각 가정에서 각각의 TV에 맞게 3차원 정보를 생성하여 시청하는 것이다. 따라서 이와 같은 시스템에서 3차원 디스플레이 방식의 하나인 집적영상 기술을 사용하는 경우에도 깊이정보로부터 컴퓨터적으로 요소영상 배열을 생성하는 기법은 실질적인 관점에서 매우 중요한 요소라고 할 수 있다.

요소영상 배열을 컴퓨터적으로 생성하는 기법은 컴퓨터 그래픽에서 가시면(visible surface)을 검출하는 원리와 같다. 따라서 물체의 정의 내역을 다루는 것인가 혹은 투영된 상을 다루는 것인가에 따라 크게 물체 공간(object space) 기법과 이미지 공간(image space) 기법으로 구분된다. 이중 물체 공간 기법에 근거하여 요소영상 배열을 생성하는 알고리즘으로 Direct 알고리즘이 제안되었고[3], 이미지 공간 기법에 근거하여 요소영상 배열을 생성하는 알고리즘으로 Efficient 알고리즘이 제안되었다[4].

Direct 알고리즘은 요소영상 배열을 생성하기 위해서 물체를 구성하고 있는 모든 점들을 요소 렌즈 배열을 통해서 영상 면에 투영한다. 따라서 객체를 구성하고 있는 픽셀의 수가 N개이면 계산시간이 O(N)이 된다. 더욱이 객체를 구성하고 있는 여러 개의 픽셀들이 영상 면의 같은 위치에 투영되는 경우가 있어서 중복 계산이 되는 비효율적인 측면이 있다. 이러한 비효율적인 측면을 개선하기 위해 Efficient 알고리즘이 제안되었다[4]. Efficient 알고리즘은 영상 면의 각 픽셀에 대해서 공간에 있는 물체의 대응점을 찾아서 영상 면에 기록함으로써 중복 계산을 없애주어 계산 속도를 향상시킬 수 있는 알고리즘이다.

특히 이미지 공간 기법을 사용한 Efficient 알고리즘은 물체를 구성하고 있는 면이 평면영상으로 요소 렌즈에 수직하고 몇 개의 면으로 구성된 특별한 경우에만 계산 속도를 향상시킬 수 있다. 따라서 깊이정보로 표현된 3차원 영상정보를 전송하는 실시간 방송 시스템에 이미지 공간 기법인 Efficient 알고리즘을 적용하면 알고리즘의 효율성이 달라질 수 있다. 따라서 본 논문에서는 물체 공간 기법인 Direct 방식에 LUT를 사용한 알고리즘을 새로이 제안하고, 기존의 알고리즘들과 계산 속도를 실험을 통해서 비교한 결과 제안된 알고리즘이 더 효율적임을 확인하였다.

Ⅱ. 집적영상 시스템

집적영상 시스템은 그림 1과 같이 크게 픽업(Pickup) 과정과 재생 과정으로 나눌 수 있다. 집적영상의 픽업 과정에서는 3차원 물체의 정보를 렌즈배열과 CCD와 같은 2차원 영상 감지기를 이용하여 2차원 요소영상 배열(the array of elemental image)을 픽업한다. 그리고 재생 과정에서는 픽업과정에서 얻어진 2차원 요소영상 배열을 LCD와 같은 디스플레이 패널에 표현하고 이들을 다시 렌즈 배열을 통과시켜 3차원 영상으로 재생한다.

그러나 위와 같은 방법을 그대로 방송 시스템이나 비디오 디스플레이 시스템에 적용하게 되면 문제가 생기게 된다. 즉, 요소영상 배열은 높은 해상도의 영상이기 때문에 데이터의 양이 많아지게 된다는 문제가 생긴다. 또한 픽업 과정과 재생 과정의 렌즈 특성이 다른 경우가 생길 수도 있게 되어 전송받은 요소영상 배열을 디스플레이 할 수 없게 될 수도 있다. 따라서 실제로 집적영상 기술을 3차원 방송 시스템에 적용하기 위해서는 그림 2와 같은 구조의 시스템으로 변형하여야 한다. 즉, 3차원 영상정보를 얻을 수 있는 3차원 영상정보 획득부, 획득된 3차원 정보를 전송하는 전송부, 그리고 전송된 3차원 정보로부터 요소영상 배열을 생성하는 요소영상 생성부, 그리고 마지막으로 생성된 요소영상을 디스플레이 하는 재생부로 구성된다.

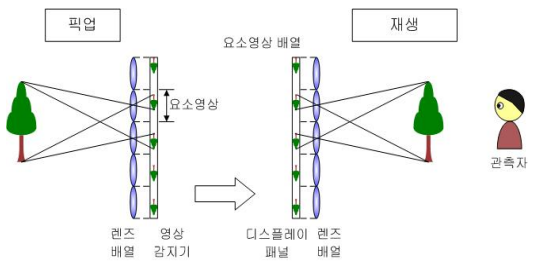


그림 1. 집적영상 시스템

www.dbpia.co.kr

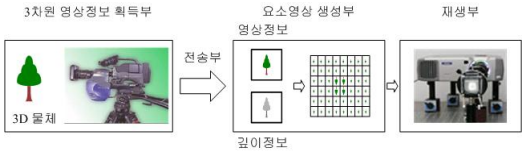


그림 2. 3차원 영상정보 획득부, 전송부, 요소영상 생성부 및 재생부

그림 2의 시스템에서 획득된 3차원 정보는 2차원 영상정보와 이에 동기화된 2차원 깊이정보로 표현된다고 가정한다. 따라서 재생부에서 디스플레이를 하기 전에 전송된 2차원 영상정보와 깊이정보를 사용해서 집적 시스템에 맞는 적합한 요소영상 배열을 컴퓨터적으로 생성한 후에 이를 사용해서 디스플레이를 해야 한다. 최근에 컴퓨터 프로세서의 발달로 요소영상 배열을 빠른 시간에 계산할 수 있지만, 방송과 같은 실시간 응용 분야에서는 요소영상 배열을 실시간으로 빠르게 계산하는 방법은 매우 중요하다.

Ⅲ. 기존의 요소영상 배열 생성 기법

본 장에서는 3차원 정보에 대해 영상정보와 깊이정보가 주어졌을 때 이로부터 컴퓨터적으로 요소영상 배열을 생성하는 방법에 대해 고려한다. 깊이정보는 물체의 거리 정보를 가지고 있으며, 이를 표현하기 위하여 일반적으로 명암도(gray scale) 영상 파일로 나타낸다. 이 명암도 영상에서 밝을수록 가까이 있는 것을 뜻하고 어두울수록 멀리 있다는 것을 뜻한다. 깊이정보로부터 요소영상 배열을 생성하는 개념적인 구조는 그림 3과 같다. 즉, 깊이정보로부터 요소영상 배열을 생성하기 위해서 먼저 z=0 위치에 핀홀 (pinhole) 배열을 위치시키고 z축의 음의 방향 g 위치에 영상 면을 둔다. 그리고 핀홀 배열에서 적당한 거리를 두고 깊이정보의 밝은 부분을 가까이 두고 어두운 부분을 먼 쪽에 둔다. 그런 후에 광선추적 (ray-tracing) 방식을 사용해서 요소영상 배열을 생성하게 된다.

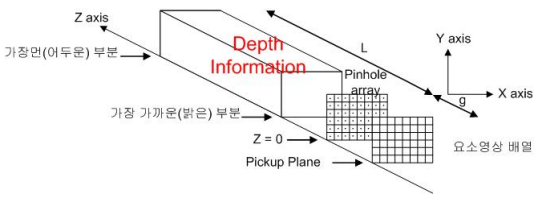


그림 3. 깊이정보로부터 요소영상 배열을 생성하는 개념적인 구조

```
g = distance_between_lens_and_pickup_device
For every_object : from_the_farthest
  L = distance_between_object_and_lens
  For every_pixel_in_the_object
    (x, y) = physical_location_of_a_pixel
    For every_elemental_lens
      (cX, xY) = center_of_the_elemental_lens
      sX = cX +(cX-x)*g/L
      sY = cY +(cY-y)*g/L
      If |sX-cX| < half_lens_pitch &&
          |sX - cY| < half_lens_pitch
        (iX, iY) =
            index_of_(sX, sY)_in_the_elemental_image
        save_the_color_of_the_pixel_at_(iX, iY)
      End If
    End For
  End For
End For
```

그림 4. Direct 알고리즘

3.1 Direct 알고리즘

Direct 알고리즘은 물체 공간 기법에 근거하여 요소영상 배열을 생성하는 알고리즘이다. 즉, 이 알고리즘에서는 먼저 물체내의 모든 복셀에 대한 위치정보를 구한다. 그리고 모든 요소렌즈(elemental lens)의 중심 좌표를 사용해서 객체의 각 복셀에서 발생한 광선이 요소렌즈의 중심을 통과하여 요소영상 배열에 투영되는 지점을 찾는다. 다음 단계는 투영되는 지점이 각 요소렌즈의 영상범위 안에 있는 지를 확인하고 마지막으로 투영되는 지점이 요소렌즈의 범위 안에 있으면 물체 점이 투영된 위치를 요소영상 배열의 픽셀 인덱스로 변환하고 그 위치에 물체 점의 복셀 컬러 값을 저장한다. 이 알고리즘의 pseudo-code를 그림 4에 나타낸다[4].

3.2 Efficient 알고리즘

Efficient 알고리즘은 이미지 공간 기법에 근거하여 요소영상 배열을 생성하는 알고리즘이다. 즉, 먼저 요소영상 배열 중 한 픽셀 위치를 선택한다. 두 번째로 이 픽셀을 포함하고 있는 요소렌즈의 중심 위치를 찾는다. 세 번째로 광선추적 방식을 사용해서 점으로부터 선택된 요소렌즈의 중심을 관통하는 광선이 물체와 접하는 지를 체크한다. 이 체크 과정은 가장 가까이에 있는 물체로부터 먼 물체순서로 체크하며 만일 대응하는 점이 존재하면 그 과정을 정지할 수 있다. 마지막으로 물체에서 대응하는 점을 찾으면 물체의 위치를 픽셀 인덱스로 바꾸어 물체의 영상값을 찾아서 요소영상 배열에 기록한다. 이 과정을 요소영상 배열의 모든 픽셀에 대하여 수행하면 최종

www.dbpia.co.kr

```
g = distance_between_lens_and_pickup_device
centerX = x_coordinate_of_first_elemental_lens
boundX = centerX + half_lens_pitch
For every_x_in_the_elemental_image : from_left_to_right
  If x>boundX
    centerX = centerX + lens_pitch
    boundX = boundX + lens_pitch
  End If
  centerY = y_coordinate_of_first_elemental_lens
  boundY = centerY + half_lens_pitch
  For every_y_in_the_elemental_image :
      from_bottom_to_top
    If y>boundY
      centerY = centerY + lens_pitch
      boundY = boundY + lens_pitch
    End If
    For every_object : from_the_nearest
      L = distance_between_object_and_lens
      sX = centerX +(centerX - x)*L/g % sampling point
      sY = centerY +(centerY - y)*L/g
      If (sX, sY)_is_inside_the_object
        sample_the_color_of_(sX, sY)_in_the_object
        break_innermost_for_loop
      End If
    End For
  End For every_y_in_the_elemental_imag
End For every_x_in_the_elemental_image
```

그림 5. Eficient 알고리즘

적으로 요소영상 배열을 생성할 수 있다. 이 알고리즘의 pseudo-code를 그림 5에 나타낸다[4].

Ⅳ. LUT를 사용한 요소영상 배열 생성

기존의 제안된 알고리즘들은 물체 공간과 이미지 공간 기법으로 구분할 수 있고, 제안된 알고리즘들은 단순히 기본 원리를 설명하고 있다. 그림 2에 나타낸 것과 같이 3차원 방송과 같은 실시간 시스템에서는 빠른 요소영상 배열을 생성하는 기법이 필요하다. 따라서 3차원의 영상정보와 깊이정보가 주어진 경우, LUT를 미리 작성하고 이를 이용하여 요소영상 배열을 효과적으로 생성할 수 있는 새로운 알고리즘을 제안한다. 이를 위해서는 먼저 물체의 복셀이 픽업 면에 투영되는 위치를 찾는 방법을 알아야 하고 이를 이용해서 LUT를 작성해야 한다.

4.1 LUT의 구성

먼저 요소영상 배열 생성에서 사용할 LUT(Look-Up Table)을 작성하는 방법을 생각해 보자. 그림 6은 3차원 공간에 있는 객체의 임의의 한 점 (Ox, Oy, Oz)에서 발생한 광선이 핀홀 배열을 통과하여 픽업 면에 투영되는 방법을 나타내었다. 즉 광선추적 기법을 나타내었다.

그림 6을 보면 요소영상 배열을 생성하는 것은 임의의 복셀 (Ox, Oy, Oz)가 주어졌을 때 광선추적 기법을 사용해서 픽업 면에 투영되는 점 (Px, Py)를 구하는 문제이다. 그림 6에서 알 수 있듯이 복셀 (Ox, Oy, Oz)가 픽업 면에 투영하는 점 (Px, Py)는 (Ox, Oz) 값에 따라 Px 값이, (Oy, Oz)값에 따라 Py 값이 각각 독립적으로 결정되는 것을 알 수 있다. 즉 2차원 픽업 면에 투영되는 점 (Px, Py)를 구하는 문제는 1차원 Px와 1차원 Py를 구하는 문제로 변형할 수 있으며, 이를 더 간략화 하여 1차원 도면으로 변경하여 그림 7에 나타낸다.

그림 7에서 투영되는 점 Px = Cx + (Cx - Ox)*g/L 식에 의해서 구할 수 있다. 여기서 Cx는 핀홀 배열에 있는 각각의 핀홀 중심에 대한 x 좌표이고, g는 핀홀 배열과 픽업 면과의 z축 방향의 수직 거리이고, L은 핀홀 배열과 물체의 복셀 간의 z축 방향의 수직 거리이다. 그림 7을 보면 광선 (a)와 (b)의 경우는 픽업 면에서 해당 요소 렌즈의 영역을 벗어나게 된다. 즉, 투영되는 점의 위치가 핀홀 중심에서 거리가 lens_ pitch/2 보다 크기 때문이다. 따라서 이와 같은 경우 해당 요소영상의 픽업 면에 기록되지 않는다.

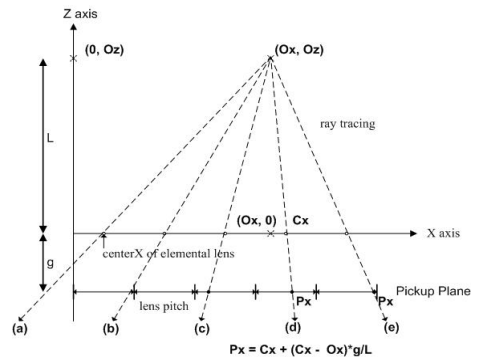


그림 6. (Ox, Oz) 값에 의한 투영 점을 구하는 방법

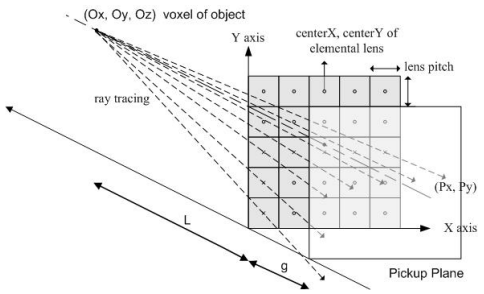


그림 7. 임의의 복셀에 대한 광선추적 기법

www.dbpia.co.kr

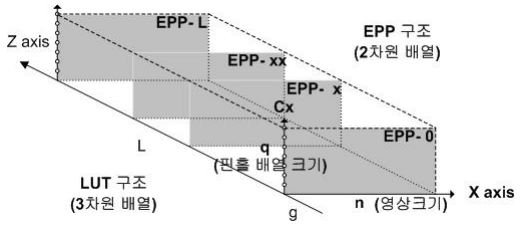


그림 8. (Ox, Oz) 값에 대한 룩업테이블 구조

광선 (c), (d), (e)의 경우에는 투영되는 점의 위치가 핀홀 중심에서 거리가 lens_pitch/2보다 작기 때문에 투영되는 점이 생기고 결과적으로 해당 요소영상의 픽업 면에 기록되게 된다. 이 결과로부터 알 수 있는 것은 점들이 z 방향으로 멀리 있는 경우에는 많은 요소영상 배열에 투영되나 가까이 있는 경우에는 약간의 요소영상 배열에 투영되는 것을 알 수 있다. 이런 특성으로 인하여 실험 데이터의 종류와 구성에 따라 계산속도가 달라진다.

먼저 핀홀 배열이 p개의 행과 q개의 열, 즉 pq 배열로 구성되어 있고 영상의 크기가 m개의 행과 n개의 열, 즉 mn행렬이라고 가정하고 룩업테이블을 만드는 방법을 생각해 보자. 이는 투영되는 점 Px = Cx + (Cx - Ox)*g/L에 의해서 결정된다. 따라서 투영되는 점에 대한 룩업테이블은 z 방향으로의 거리인 g와 L의 값이 고정되면, 모든 x 값에 대해서 x 방향으로 핀홀 배열의 개수인 q 크기의 투영되는 점들을 구하게 된다. 이를 요소 투영 패턴(EPP, Elemental Projection Pattern)이라고 하고, 크기는 qn이 된다. 그런 후에 고정된 g 값에 대해서 모든 z(L)의 값에 따라 EPP들을 구하면 x의 투영 점에 대한 룩업테이블이 완성된다. 이를 개념적으로 나타내면 그림 8과 같다. 물론 y 값에 대해서도 EPP를 만들고 이를 모든 z의 값에 따라 EPP들을 구하면 y의 투영 점에 대한 룩업테이블이 된다. 또한 핀홀 배열이 x와 y 방향으로 같은 크기이고 중심이 같으면 하나의 룩업테이블을 작성해서 사용해도 된다. x의 투영 점에 대한 룩업테이블을 작성하는 알고리즘의 pseudo-code를 그림 9에 나타낸다.

4.2 룩업테이블을 사용한 요소영상 배열 생성

룩업테이블을 사용한 요소영상 배열 생성 알고리즘은 Direct 알고리즘과 같이 물체 공간 기법의 알고리즘으로 실행 순서는 다음과 같다. 첫 번째로 거리가 가장 먼 곳으로부터 임의의 한 점을 선택한다. 선택된 점으로부터 깊이 정보인 거리 L과 (Ox, Oy)값을 알 수 있다. 두 번째 단계는 요소영상 배열의 각각 요소 영상에 대해서 복셀 (Ox, Oy, L)에 대한 투영 점의 좌표 (Px, Py)를 룩업테이블을 사용해서 구한다. 세 번째 단계는 (Px, Py)의 값이 모두 양이면 물체의 영상 정보를 픽업 면에 저장한다. 그런 후에 거리가 먼 곳부터 가까운 곳으로 모든 픽셀에 대해서 위의 과정을 수행하면 요소영상 배열을 생성할 수 있다. 이 알고리즘의 pseudo-code를 그림 10에 나타낸다.

룩업테이블을 사용하면 계산 속도가 빨라지나 룩업테이블 방식의 Px = eiXLUT(Ox, L, a_x) 식에서 3차원 배열 eiXLUT에서 값을 찾기 위해서는 주어진 인덱스 값을 사용해서 Ox + L*(임의의 수) + a_x*(임의의 수) 식을 계산해야 한다는 것을 기억해야 한다.

```
% make the LUT for the (x, z) value
For L=1:z_Max % z variable
  % for making each EPP(elemental Projection Pattern)
  For col=1:obj_xdim % x variable
    % for the array of elemental image
    For a_x=1:a_xdim
      % x center of current ei array
      cX_ei = (a_x * ei_xpitch) - (ei_xpitch/2);
      pX = cX_ei + (cX_ei - col)*g/L;
      If ( abs(pX - cX_ei) < (ei_xpitch/2) )
        iX = round(pX);
        eiXLUT(col, L, a_x) = iX;
      else
        eiXLUT(col, L, a_x) = 0;
      End If
    End For
  End For
End For
```

그림 9. (Ox, Oz) 값에 대한 룩업테이블 작성 알고리즘

```
extract the points from the depth information
sort every_pixel_in_the_object_from_the_farthest
For every_pixel_in_the_object : from_the_farthest
  L = distance_between_pixel_and_lens    % z coordinate
  (Ox, Oy) = the coordinate of voxel in the space
  % for the array of elemental image
  For a_y=1:a_ydim
    % find the y projection coordinate by LUT
    Py = eiYLUT(Oy, L, a_y);
    If (pY> 0)
      For a_x=1:a_xdim
        % find the x projection coordinate by LUT
        Px = eiXLUT(Ox, L, a_x);
        If ( Px>0 )
          save_the_color_of_the_voxel_at_(pX, pY)
        End If
      End For % for a_x
    End If % if
  End For % for a_y
End For
```

그림 10. 룩업테이블을 사용한 알고리즘

www.dbpia.co.kr

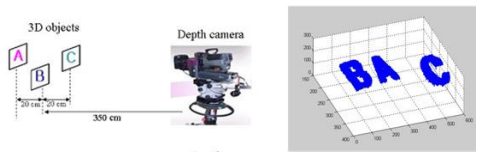
V. 실험 및 결과 분석

본 논문에서 제안한 룩업테이블을 사용한 알고리즘의 효율성을 검증하기 위해서 Depth 카메라로 추출한 깊이정보와 3D MAX로 추출한 깊이정보를 사용해서 기존의 알고리즘들과 제안된 알고리즘의 계산 속도를 측정하는 실험을 수행하였다.

5.1 Depth 카메라로 추출한 깊이정보 사용

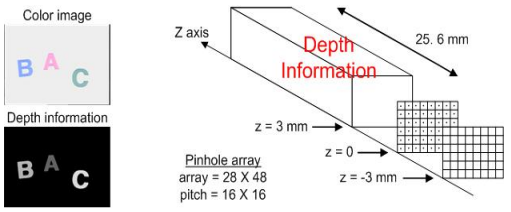
Depth 카메라를 사용하여 깊이정보를 얻기 위해서 그림 11의 (a)와 같이 'A','B' 그리고 'C'의 세 글자로 이루어진 3차원 물체를 사용하였다. 이들 글자의 실제크기는 약 8.5 cm × 8.5 cm이고, 깊이 카메라로부터 350 cm 위치에 놓여있다. 실험을 통해서 추출된 깊이정보는 그림 11의 (b)에 나타내었으며 크기는 740×468 픽셀이다. 이 데이터는 거의 3개의 면으로 구성된 간단한 형태의 깊이정보이다.

그림 11의 깊이정보로부터 요소영상 배열을 생성하기 위한 실험구성은 그림 12의 (a)와 같고, 같은 물체 공간 기법을 사용하는 Direct 알고리즘과 제안한 알고리즘은 모두 같은 요소영상 배열을 생성하였으며 그림 12의 (b)와 같다.

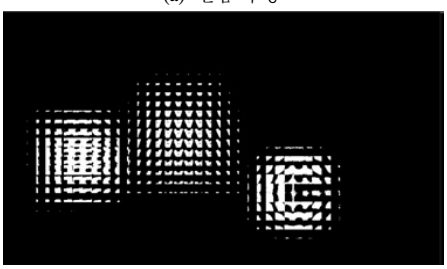


(a) 실험 구성 (b) 추출된 깊이정보

그림 11. 깊이정보를 추출하기 위한 실험구성과 깊이정보



(a) 실험 구성



(b) 획득된 요소영상 배열

그림 12. 그림 11의 깊이정보에 대한 실험 구성과 요소영상 배열

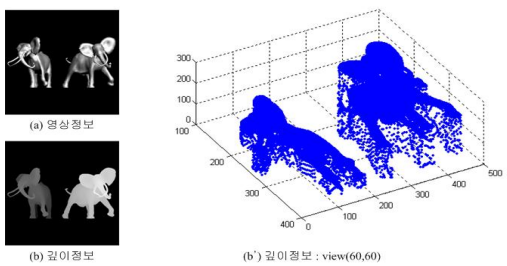
실험에 사용된 ABC 데이터의 깊이정보에 있는 점(복셀)들의 개수는 총 19759 개이었다. Matlab 7.0에서 Efficient 알고리즘의 수행시간은 27.7656 sec, Direct 알고리즘의 수행시간은 3.2969 sec, LUT을 사용한 알고리즘은 0.2031 sec가 되었다. 따라서 더 빠른 Direct 알고리즘에 대해서도 LUT를 사용한 알고리즘이 상대적으로 약 16배 빠른 것으로 나타났다. 또한 위의 실험 구성에서 깊이정보를 3mm 간격을 두지 않고 실험한 결과 Direct 알고리즘의 수행시간은 3.1406 sec, LUT를 사용한 알고리즘은 0.1563 sec로 약 20배 빠른 것으로 나타났다. 이는 LUT를 사용한 알고리즘은 복셀들이 렌즈 배열에 가까이 있을수록 속도가 많이 빨라지는 특성을 가지고 있기 때문이다.

5.2 3D MAX로부터 추출한 깊이정보 사용

3D MAX 프로그램으로 코끼리 두 마리를 그림 13의 (a)와 같이 설계하고 이 3차원 물체를 사용하여 추출한 깊이정보는 그림 13의 (b)와 같고, 크기는 500x500 픽셀이었다. 이 데이터는 일반적인 형태의 깊이정보이다.

그림 13의 깊이정보로부터 요소영상 배열을 생성하기 위한 실험구성은 그림 14의 (a)와 같고, 같은 물체 공간 기법을 사용하는 Direct 알고리즘과 제안한 알고리즘은 모두 같은 요소영상 배열을 생성하였으며 그림 14의 (b)와 같다.

실험에 사용된 코끼리 두 마리 데이터의 깊이정보에 있는 점(복셀)들의 개수는 총 53421 개이었다. Matlab 7.0에서 Efficient 알고리즘의 수행시간은 19.5313 sec, Direct 알고리즘의 수행시간은 6.9375 sec, LUT을 사용한 알고리즘은 0.4531 sec가 되었다. 따라서 더 빠른 Direct 알고리즘에 대해서도 상



(a) 영상정보 (b) 깊이정보 (b') 깊이정보 : view(60,60)

그림 13. 3D MAX의 영상정보와 추출한 깊이정보

www.dbpia.co.kr

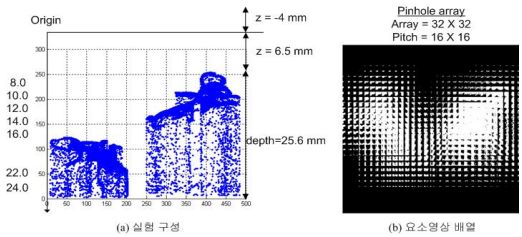


그림 14. 그림 13의 깊이정보에 대한 실험구성과 요소영상 배열

대적으로 LUT를 사용한 알고리즘이 약 15배 빠른 것으로 나타났다.

Ⅵ. 결 론

결론적으로, 본 논문에서는 3차원 집적 영상 기술에서 깊이정보로부터 요소영상 배열을 생성하는 새로운 LUT를 사용한 알고리즘 제안하고 이에 대해서 분석하였다. Depth 카메라와 3D MAX로부터 획득한 깊이 정보에 대해서 제안한 알고리즘과 기존의 Direct 알고리즘과 Efficient 알고리즘을 사용해서 요소영상 배열을 생성하는 실험을 수행하고 계산시간을 측정하였다. 이를 통하여 제안한 알고리즘의 계산 속도가 상대적으로 효율적임을 확인하였다.

참 고 문 헌

〔1〕 김은수, 이승현 공역, *3차원 영상의 기초*, 기다리, 1998

〔2〕 Andre Redert, Mark Op de Breek, Christoph Fehn, Wijnand IJsselsteijn, Marc Pollefeys, Luc Van Gool, Eyal Ofea, Ian Sexton, Philip Surman, "ATTEST: Advanced Three-dimensional Television System Technologies," *Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission(3DPVT02) 2002*, IEEE

〔3〕 S. Jung, S.-W. Min, J.-H. Park, and B. Lee, "Study of Three-dimensional display system based on Computer-generated Integral Photography," *Journal of Optical Society of Korea*, Vol.5, No.3, pp.117-122, September 2001.

〔4〕 S. Oh, J. Hong, J.-H. Park, and B. Lee, "Efficient Algorithm to Generate Elemental Images in Integral Imaging," *Journal of Optical Society of Korea*, Vol.8, No.3, pp.115-121, September 2004.

〔5〕 F. Okano, H. Hoshino, J. Arai, and I. Yuyama, "Three-dimensional video system based on integral photography," *Opt. Eng.*, 38, pp.1072-1077, 1999.

〔6〕 H. Arimoto and B. Javidi, "Integral three-dimensional imaging with digital reconstruction," *Opt. Lett.*, Vol.26, No.3, pp.157-159, 2001.

〔7〕 S.-H. Hong, J.-S.Jang, and B. Javidi, "Three-dimensional volumetric object reconstruction using computational integral imaging," *Opt. Express 12*, 483-491, 2004.

〔8〕 D.-H. Shin, M. Cho, K.-C. Park and E.-S. Kim, "Computational technique of volumetric object reconstruction in integral imaging by use of real and virtual image fields," *ETRI Journal*, Vol.27, No.6, pp.208-212, 2005.

권 영 만 (Young-Man Kwon) 정회원

1983년 2월 광운공과대학 전자공학과 졸업
1985년 2월 한국과학기술원 전기및전자공학과 석사
1998년 8월 한국과학기술원 정보및통신공학과 박사수료
2007년 2월 광운대학교 전자공학과 박사
1993년 3월~현재 을지대학교 의료산업학부 교수
<관심분야> 영상처리, 머신비전, 운영체제

김 승 철 (Seung-Chul Kim) 정회원

2002년 2월 광운대학교 전자공학과 졸업
2004년 2월 광운대학교 전자공학과 석사
2007년 2월 광운대학교 전자공학과 박사
2007년 3월~현재 광운대학교 전자공학과 차세대3D디스플레이 연구센터 연구교수
<관심분야> 3차원 디스플레이, 홀로그래픽 디스플레이, 광공학

김 은 수 (Eun-Soo Kim) 종신회원
한국통신학회지 Vol.24, No.9A참조

www.dbpia.co.kr