

최적 4진 트리와 깊이 영상을 이용한 3차원 물체 표현 방법에 관한 연구

* 송유진, 홍민석, 최윤호, 박상희

연세대학교 전기공학과

Construction of Octree for Depth Images using Optimal Quadtree

Yu-Jin Song, Min-Sug Hong, Yun-Ho Choi, Sang-Hui Park

Dept. of Electrical Engineering, Yonsei University

ABSTRACT

This study presents new algorithm for generating octree by volume intersection with pseudo-octrees which are constructed for each orthogonal face views in order to represent 3D objects exactly.

In this algorithm, optimal quadtree is constructed for depth images in which distance information is represented in gray level, and then pseudo-octrees are constructed by using optimal quadtree in each orthogonal face view.

It is shown that 3D object with concave parts can be reconstructed exactly by our suggested algorithm, and that memory space and tree construction time are also reduced compared with other algorithms.

1. 서론

최근 공학 및 컴퓨터 과학이 급속도로 발달함에 따라 컴퓨터를 이용한 3차원 물체의 표현 및 이용은 입체 모델링(solid modeling), 컴퓨터 그래픽, 컴퓨터 비전, CAD, 영상 처리 및 로보트 공학등 여러 분야에서 그 중요성을 더해 가고 있다.

3차원 물체의 표현 방법에는 여러 가지가 있는데, 반복적 부분 분할(recursive subdivision) 원리를 이용하여 물체를 표현하는 것을 8진트리(octree)라 한다.

8진트리는 2차원적 표현인 4진트리(quadtree)를 3차원적으로 확장한 계층적 데이터 구조(hierarchical data structure)이며 트리 탐색, 물체사이의 교차 검사, 공간에서 점이나 불력의 위치 결정, 음면 제거 조작, 연결된 성분의 라벨링, 이웃 영역 발견 등에 아주 편리하다.

Chien 과 Aggawal[2]은 3차원 물체의 3개의 직교면에 대한 실루엣(silhouettes) 영상을 이용하여 8진트리를 구성하는 알고리즘을 제시하였는데 이 알고리즘은 이진 영상(binary image) 즉 실루엣 영상을 사용하기 때문에 복잡한 물체는 정확히 표현하지 못하는 단점이 있다. 그후에 Ahuja와 Veenstra[3]는 3개의 면, 6개의 모서리 및 4개의 꼭지점을 관측점으로 해서 얻은 실루엣 영상을 이용하여 8진트리를 구성하는 알고리즘을 제시하였는데 이는 물체를 좀더 정확히 표현하는 장점이 있으나 관측점이 너무 많아서 8진트리를 구성하는데 있어 구성 시간이나 메모리 운용면에서 비효율적이다.

따라서 본 연구에서는 이와 같은 단점을 개선하기 위하여 6개의 각 직교면에 대한 실루엣 영상을 거리 정보를 갖는 깊이 영상(depth image)로 변환하여 이에 대하여 트리 구성시간과 메모리 운용면에서 효율적인 최적 4진트리(optimal quadtree) 알고리즘을 이용하여 4진트리를 구성한 다음 의자 8진트리로 확장하여 체적 교차법에 의해 3차원 물체를 정확하게 표현하는 완전한 8진트리(complete octree)를 구성하는 알고리즘을 제시하고자 한다.

2. 최적 4진트리 구성

Samet 가 제안한 래스터 영상(raster image)을 포인터 구조를 이용하여 4진 트리를 구성하는 알고리즘[4]과 여타의 다른 선형 4진 트리(linear quadtree) 알고리즘[5]이 구성 시간과 메모리 사용량에 있어서 입력 영상의 화소수에 비례하는 것과 달리 최적 4진 트리 구성(optimal quadtree construction) 알고리즘[5]은 입력 영상의 불력 수에 비례한다.

2.1 계층적 데이터 구조

포인터(pointer)를 이용한 트리 구성 알고리즘은 트리의 조작에 있어서 효율성을 갖고 있으며 트리 탐색에도 용이하다.

Samet가 4진트리 구성 알고리즘에서 제안한 포인터 구조 즉 데이터 구조(data structure)는 6개의 필드로 구성되는데, 첫번째 필드는 이웃 노드(neighbor node) 탐색에 용이한 부 노드(father node)에 대한 포인터이고 다음 4개의 필드는 4개의 자노드(son node)에 대한 포인터이며 여섯 번째 필드는 노드 표현자이다. 여기서 노드 표현자는 해당 노드가 지엽 노드가 아닌 경우 4개의 자 노드의 평균색 즉 회색(gray)으로 표시된다.

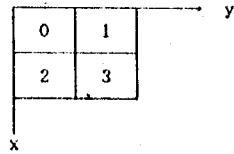


그림 2.1 4진소 라벨

한편 그림 2.1은 본 연구에서 사용한 4진소 라벨을 나타낸다. 포인터를 이용한 트리 구성은 위에서 언급한 여러가지 장점에도 불구하고 노드에 대한 포인터가 메모리 공간의 대부분을 차지하기 때문에 노드가 많이 발생하는 경우 메모리 운용면에서 비효율적이다.

따라서 본 연구에서는 메모리 공간의 대부분을 차지하는 포인터중 부 노드에 대한 포인터를 없애고 5개의 필드로 구성되는 데이터 구조를 사용하였으며 부 노드 정보를 위한 버퍼를 구성하여 부 노드에 대한 포인터를 대신하였다. 그림 2.2는 본 연구에서 사용한 데이터 구조이고 그림 2.3은 8진트리에 사용될 그림 2.2의 확장된 데이터 구조이다.

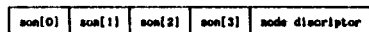


그림 2.2 4진트리의 데이터 구조

SONE(0)	SONE(1)	SONE(2)	SONE(3)	SONE(4)	SONE(5)	SONE(6)	SONE(7)	node discriminator
---------	---------	---------	---------	---------	---------	---------	---------	-----------------------

그림 2.3 확장된 데이터 구조

2.2 최대 노드의 삽입

최적 4진 트리를 구성하는데 있어서 최대 노드 삽입 (largest node insertion) 알고리즘이 사용되는데 구체적인 과정은 다음과 같다.

입의 노드 E의 여러개의 작은 지엽 노드(F,G,H,I)에 대응되는 지엽 노드 L이 노드 E에 삽입되면 작은 지엽 노드들은 L로 대체된다(그림 2.4b의 A노드). 만약 L이 지엽 노드 F와 크기가 같다면 L의 색으로 대체되며 그결과 병합된다(그림 2.4c의 E노드).

L이 L과 같은 색을 가지는 지엽 노드의 한 부분일 경우 그 지엽노드에는 아무런 변화가 일어나지 않는다(그림 2.4d).

삽입된 지엽 노드 L가 다른색의 다른 지엽 노드 B의 한 부분일 경우 그노드 B는 L과 같은 크기의 지엽 노드가 생길때까지 부 4진소(subquadrant)로 부분 분할된 다음 L의 색으로 대체된다(그림 2.4e).

따라서 한번의 삽입은 4진 트리의 많은 노드를 제거하거나 첨가한다. 이와 같은 삽입 방법을 이용하여 입의 화소에 대하여 노드를 삽입할때 그 화소를 첫번째 화소로하는 가장 큰 노드를 삽입하는 것을 최대 노드 삽입이라 한다.

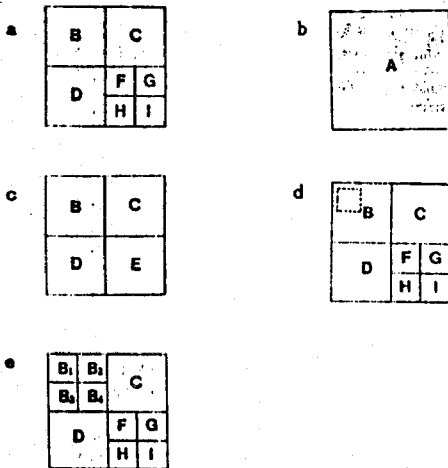


그림 2.4 노드의 삽입 효과.

2.3 이웃 노드의 탐색과 발견

영상의 특정 블록에 해당하는 노드가 주어졌을때 수평 혹은 수직 방향의 크기가 다른 이웃 노드를 발견하는 과정은 다음과 같은 두가지 과정으로 구성된다.

과정1 : 두 노드의 공통 부모노드(common father node)를 결정한다.

과정2 : 두 노드에 해당하는 블록 사이의 공통 경계선에 의해 형성되는 축에 대해 반사영상(mirror image)이 움직이는 경로로 다시 탐색해 내려간다.

한편 Samet가 4진 트리를 구성하기 위해 사용한 이웃 노드 발견 알고리즘[5]은 종열(column)이 바뀔 때는 이웃 노드를 제대로 찾아 가지만 행열(row)이 바뀔때는 이웃 노드를 찾아 가지 못하는 단점이 있다.

본 연구에서는 이러한 단점을 보완하는 알고리즘을 제

시하고자 한다. 개선된 이웃 노드 발견 알고리즘은 알고리즘1과 같다. 표2.1 및 표2.2는 이 알고리즘에서 사용되는 함수에 대한 정의이다.

```

procedure NEIGHBOR(P D Q L)
begin
  value node P ;
  value direction D ;
  reference node Q ;
  reference integer L ;
  D ← E ;
  L ← L + 1 ;
  if not NULL(FATHER(P)) and ADJ(D,SONTYPE(P))
  then
    NEIGHBOR(FATHER(P),D,Q,L) ;
  else Q ← FATHER(P) ;
  if(col=2N-1) then D ← S ;
  else D ← E ;
  if not NULL(Q) and GRAY(Q) then
  begin
    if(row=2N-1) then
      REFLECT(D,SONTYPE(P)) ← SW ;
      Q ← SON(Q,REFLECT(D,SONTYPE(P))) ;
      L ← L - 1 ;
    end ;
  end ;
end ;
    
```

알고리즘1 개선된 알고리즘

표 2.1 함수 ADJ(S,Q)

표 2.2 함수 REFLECT(S,Q)

ADJ		REFLECT	
S	Q	S	Q
	NW NE SW SE		NW NE SW SE
N	T T F F	N	SW SE NW NE
E	F T F T	E	NE NW SW SE
S	F F T T	S	SW SE NW NE
W	T F T F	W	NE NW SE SW

위의 알고리즘을 사용하면 행열이 바뀔 때 이웃 노드를 제대로 찾아 가고 있음을 알 수 있다. 따라서 본 연구에서 제안한 알고리즘은 이웃 노드를 탐색할 때 매번 먼저 루트 노드를 찾아가서 다시 이웃 노드를 향해 탐색해 내려오는 이전의 알고리즘 보다 시간 면에서 훨씬 효율적이다.

2.4 포인터를 이용한 최적 4진 트리 구성

Samet가 제안한 이전의 4진 트리 구성 알고리즘[4]은 이웃 노드 추가(add neighboring)루틴을 이용하여 입력 영상의 각각의 화소에 대하여 먼저 지엽 노드를 생성한 다음 4개의 4진소의 색이 같으면 병합하는 과정을 반복해서 마지막으로 전체 영상을 나타내는 루트 노드를 생성한다.

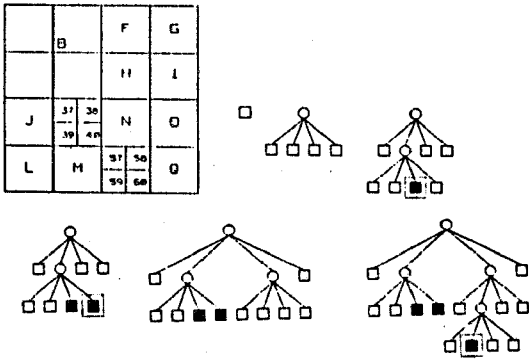
그러나 앞절에서 기술한 최대 노드 삽입 알고리즘과 이웃 노드 탐색 알고리즘을 이용하면 트리상의 각 노드가 입력 영상의 블록에 대해 발생하기 때문에 병합 과정이 필요없게 된다.

구체적인 최적 4진 트리 형성 과정은 다음과 같다. 레스터 영상의 각 화소들은 첫번째 행열의 첫번째 화소부터 처리된다.색이 C인 첫번째 화소가 처리된 후 4진 트리는 루트 노드(전체 영상)에 해당하는 C색의 지엽 노드로 표현된다. 다음 화소가 처리될때 그화소의 색이 C이면 처리되지 않고 다음 화소로 넘어가고 다른색 D를 가지면 현재의 노드는 회색 노드(gray node)가 되고 C색을 가진 4개의 자노드가 발생한다.이때 현재의 노드는 현재의 화소에 대응되는 자 노드이다.

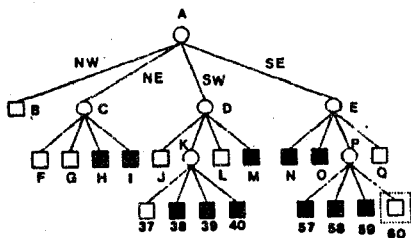
현재의 화소가 현재 노드에 해당하는 영상의 블록의 첫번째(왼쪽 상단) 화소이면 노드의 색은 D로 변하고 그렇지 않은 경우 현재의 화소가 현재의 노드에 해당하는 블록의 첫번째 화소가 될때까지 부분 분할을 반복한다. 그리고 다음에 처리될 화소가 현재의 노드에 해당하는 블록의 동쪽 경계선 외부에 있거나 행열이 바뀔 때는 이웃 노드 탐색 알고리즘을 이용하여 이웃 노드를 발견함

으로써 위치한다.

그림2.5의 (a)는 최적 4진트리가 구성되는 과정을 그림으로 나타낸 것이고 (b)는 구성된 최적 4진 트리를 나타낸다.



(a) 최적 4진 트리 구성 과정



(b) 구성된 최적 4진 트리

그림2.5 최적 4진 트리

3. 깊이 영상으로부터의 8진트리 구성

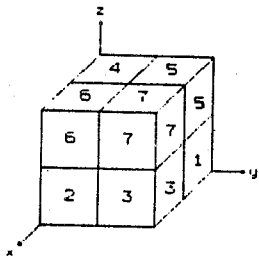


그림3.1 8진소 라벨

3차원 공간에서 물체의 정점에 대한 좌표는 스테레오 영상(stereo image)이나 센서 등을 통해 얻을 수 있다.

본 연구에서는 복잡한 3차원 물체를 8진트리로 정확히 표현하기 위해 먼저 6면(x, y, z, -x, -y, -z)의 면 관측에 의한 실루엣 영상을 거리 정보를 가진 깊이 영상(depth image)으로 변환하여 이를 8진트리를 구성하는 알고리즘을 제시하고자 한다.

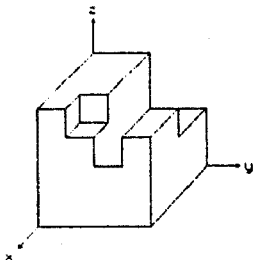
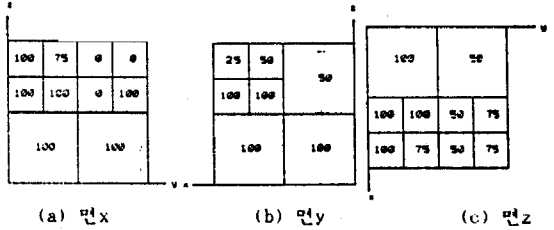


그림3.2 3차원 물체

먼저 그림3.2의 3차원 물체의 정점에 대한 좌표를 이용하여 관측자로 부터의 거리에 따라 영역 구분된 실루엣 영상에 대하여 그 거리에 반비례하여 계조도 값을 0(%)에서 100(%)까지 차례로 주어서 거리 정보를 가진 깊이 영상(depth image)으로 변환한다. 변환된 x, y, z 면의 깊이 영상은 그림3.3와 같다



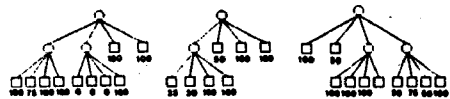
(a) 면x (b) 면y (c) 면z

그림3.3 계조도 값에 의해 구분된 깊이 영상

이와 같은 방법에 의해 얻어진 깊이 영상은 단순히 3차원 물체의 공간 점유 영역에 대한 정보뿐 아니라 그 영역의 깊이에 대한 정보도 포함된 영상이 된다.

그림3.3의 깊이 영상을 이용하여 3차원 물체에 대한 8진트리를 구성하기 위해선 우선 깊이 영상에 대한 4진트리로 구성해야 한다.

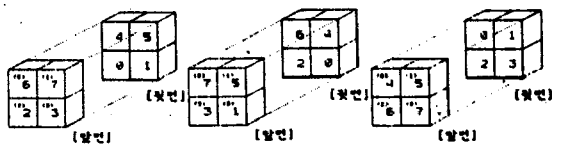
본 연구에서는 4진트리를 구성하는데 있어 구성 시간과 메모리 운용의 효율성을 위하여 3장에서 서술한 최적 4진트리 구성 알고리즘(optimal quad tree construction algorithm)을 이용하여 깊이 영상의 각 영역이 같은 계조도 값을 가질 때까지 반복적으로 부분 분할함으로써 4진트리를 구성하였다. 구성된 4진트리는 그림3.4와 같다.



(a) 면x (b) 면y (c) 면z

그림3.4 각 면의 깊이 영상에 대하여 구성된 4진트리

이와 같이 구성된 4진트리를 이용하여 각 면에 대한 의사 8진트리(pseudo octree)를 구성하기 위해선 4진소와 8진소의 사상 관계가 필요한데 그림3.5은 앞면과 뒷면에 따른 사상 관계를 나타낸다.



(a) 면x (b) 면y (c) 면z

그림3.5 4진소와 8진소의 사상 관계

또한 각 면에 대한 의사 8진트리를 구성하는데 있어 8진트리 공간의 임의의 8진소가 흑색 노드이나 백색 노드 이냐는 부 8진소(father octant)내에서 앞면에 위치하느냐 혹은 뒷면에 위치하느냐와 8진트리에서의 깊이에 따라 결정되는 8진트리 공간상의 8진소의 계조도 값에 의해 결정되는데 그 계조도 값을 결정하는 알고리즘은 다음과 같다. 아래 알고리즘2에서 rs1과 gry의 초기치 즉 루트 노드(깊이 0)에서의 값은 전체 계조도 값의 100(%)이고 octant_resolution() 루틴이 한번 실행될 때마다 브리상의 깊이는 '1'씩 증가된다.

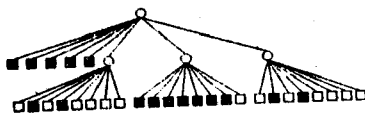
```

procedure octant_resolution(i,rsl,L,gry)
/* 임의의 8진소의 계조도값 결정 */
begin
  value positioni: /* 각8진소의앞면의
                  원점으로부터의거리*/
  value size L ;
  reference resolution rsl : /* L:3차원 물체의
  value integer gry ; /* rsl:임의의 8진소의
                        길이 */
                        /* rsl:임의의 8진소의
                        계조도 값 */
  gry ← gry/2 ;
  L ← L/2 ;
  /* 8진소가 부8진소(father node)의 뒷면에 위치
  하는 경우 */
  if(i<L) then rsl = rsl - gry ;
  if(i!=L) then octant_resolution(i,rsl,L,gry)
end ;
    
```

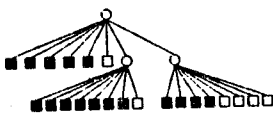
알고리즘2 8진소의 계조도 값 결정

길이 '1'에서 8진소가 가지는 계조도값은 '0', '50', '100' 3종류이고 길이 '2'에서는 '0', '25', '50', '75', '100' 5종류의 값을 가진다. 즉 길이를 d라 할때 2^{d+1} 개의 계조도 값이 나타날 수 있다. 따라서 8진트리 구성에 있어서 임의의 길이에 존재하는 8진소에 사상되는 4진소가 그 길이에 나타날 수 있는 8진소의 계조도 값들과 다른 값을 가지는 경우와 회색 노드(gray node)인 경우 그 8진소는 회색 노드가 되며 그 이외의 경우는 지엽 노드(leaf node)가 된다.

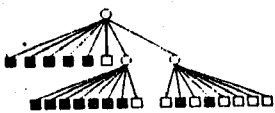
또한 임의의 8진소가 지엽 노드(leaf node)인 경우 흑색노드(black node)이나 백색 노드(white node)이나 하는 것은 알고리즘2에서 얻어진 임의의 8진소의 계조도 값이 4진트리상에서 사상되는 4진소의 계조도 값보다 작거나 같으면 흑색 노드(black node)가 되고 크면 백색 노드(white node)가 된다. 이와 같이 8진소와 4진소의 사상 관계와 계조도 값의 비교에 의해 각 면에 대해 구성된 의사 8진트리는 그림3.6과 같다.



(a)면x



(b)면y



(c)면z

그림3.6 각 면에 대한 의사 8진트리

위와 같이 구성된 각 면에 대한 의사 8진트리를 간단한 체적 교차법(volume intersection)[2]을 이용해서 완전한 8진트리(complete octree)로 구성하였으며 구성된 완전한 8진트리는 그림3.7과 같다.

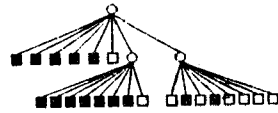


그림3.7 완전한 8진트리

4. 실험 및 결과 고찰

4.1 실험

본 연구에서 사용한 전체 실험 시스템의 구성도는 그림4.1과 같다.

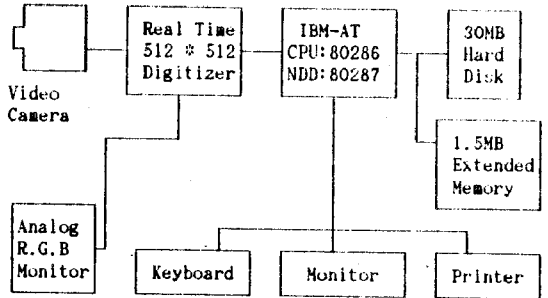
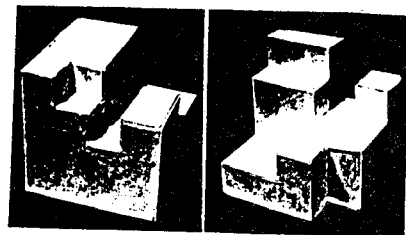


그림4.1 전체 실험 시스템의 구성

본 연구의 실험에 사용한 3차원 물체는 그림4.2의 물체이다. 그림4.2의 3차원 물체에 대한 관측 영상은 128×128 의 해상도를 가지며 입력 LUT(look-up table)을 통해 실험 영상으로 변환하였다.

한 물체에 대하여 가능한 관측은 6개의 면 즉 면x, 면y, 면z, 면-x, 면-y, 면-z 이며 본 논문에서는 물체 구성이 가능한 수 만큼 관측 영상을 얻었다.

오목한 부분과 볼록한 부분을 동시에 가지고 있는 그림4.2의 물체1은 면x, 면y, 면z, 에 대해 그리고 물체2도 마찬가지로 면x, 면y, 면z 에 대해 각각 관측 영상을 얻어 실험 영상으로 변환한 다음 이를 3차원 물체의 정점에 대한 좌표를 이용하여 깊이영상으로 변환하였다(그림4.3).



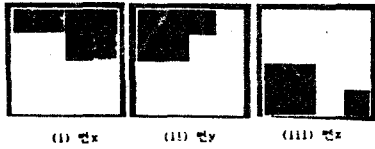
(a)물체1 (b)물체2

그림4.2 실험에 사용한 3차원 물체



(i)면x (ii)면y (iii)면z

(a) 물체1



(a) 전x (b) 옆x (c) 뒷x

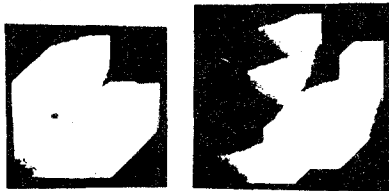
(b)물체2

그림4.3 3차원 물체의 깊이 영상

그림4.3의 각 면에 대한 깊이 영상을 본 논문에서 제시한 최적 4진트리 구성 알고리즘을 이용하여 4진트리로 구성한 다음 의사 8진트리로 확장하여 체적 교차법에 의해 완전한 8진트리로 구성하였다.

4.2 결과 고찰

그림4.4은 그림4.3의 3차원 물체의 각 면에 대한 깊이 영상을 사용하여 8진트리를 구성한 다음 이를 디스플레이한 것이다. 실루엣 영상을 사용해서는 구성하기 어려운 오목한 부분을 가지는 복잡한 물체도 본 연구에서 제시한 방법의해 정확하게 구현하고 있음을 알 수 있다. 또한 8진트리를 구성하는 과정에서 발생한 노드수와 트리 구성 시간은 기존의 4진트리 구성 알고리즘을 이용해 8진트리를 구성할 때 보다 현저하게 감소하였다. 표 4.1은 물체1,2에 대해서 기존의 8진트리 구성 알고리즘과 본 연구에서 제시한 알고리즘을 이용했을 때 발생한 노드수와 트리 구성 시간에 대한 비교이다.



(a)물체1 (b)물체2

그림4.6 깊이 영상으로부터 구성한 8진트리의 디스플레이

표4.1 노드 수 와 구성 시간 비교

[CPU CLOCK:8 MHz]

	물체 1		물체 2	
	노드 수 [개]	구성 시간 [sec]	노드 수 [개]	구성 시간 [sec]
기존 알고리즘	65,643	81	65,691	84
제시한 알고리즘	143	6	207	9

5 결 론

본 연구에서 제시한 알고리즘을 이용해 8진트리를 구성한 결과는 다음과 같다.

- 1) 기존의 트리 구성 알고리즘에서 사용한 데이터 구조를 구성하는 포인터 중 부 노드(father node)에 대한 포인터를 제거함으로써 포인터들이 차지하는 메모리 공간을 줄였으며, 4진트리 구성을 위해 사용한 이웃 노드 발견 알고리즘이 입력 영상의 형질이 바뀔 때 이웃 노드를 제대로 찾아가지 못하는 단점을 제거하였다.
- 2) 각 면의 깊이 영상을 최적 4진트리 구성 알고리즘을 이용하여 4진트리를 구성함으로써 8진트리를 구성할때 발생하는 노드 수 와 구성 시간이 기존의 알고리즘에 대하여 각각 물체1은 1/459, 1/14, 물체2는 1/317, 1/9 로 감소하였다.

- 3) 실루엣 영상을 이용하여 8진트리를 구성하는 기존의 알고리즘이 복잡한 물체에 대하여 8진트리를 정확하게 구성하지 못하는 반면에 본 연구에서 사용한 거리 정보를 가진 깊이 영상 및 제시된 계조도 결정 방법을 이용함으로써 오목한 부분을 가지는 복잡한 물체에 대해서도 정확하게 8진 트리를 구성할 수 있었다.

참 고 문 헌

- 1) C.H.Chien, Y.B.Sim, and J.K.Aggarwal, "Generation of Volume /surface octree from range data",PROC.,IEEE COMPUTER CONF.ON COMPUTER VISION,VOL.36,pp100-113, 1988
- 2) C.H.Chien and J.K.Aggarwal,"Volume/surface octrees for the representation of three-dimensional objects ", CVGIP,VOL.36,pp100 - 113, 1986
- 3) N.Ahja and J.Veenstra, " Generating from object silhouettes in orthographic views", IEEE Trance on PAMI,VOL.11,NO.2,1989
- 4) H.Samet," An algorithm for converting raster to quadtrees ", IEEE Trance on PAMI,VOL.3,pp93-95, 1981.
- 5) C.A. Shaffer and H.Samet ,"Optimal quadtree construction",CVGIP,VOL.37,pp402-419, 1987
- 6) H. Samet,"Neighbor finding techniques for images represented quadtree",CVGIP,VOL.18,pp37-57,1982.