

Introduction of 3-dimensional image

문 주 영
삼성종합기술원

1. 3D Graphics Fundamentals

..... How the eyes perceive three dimensions

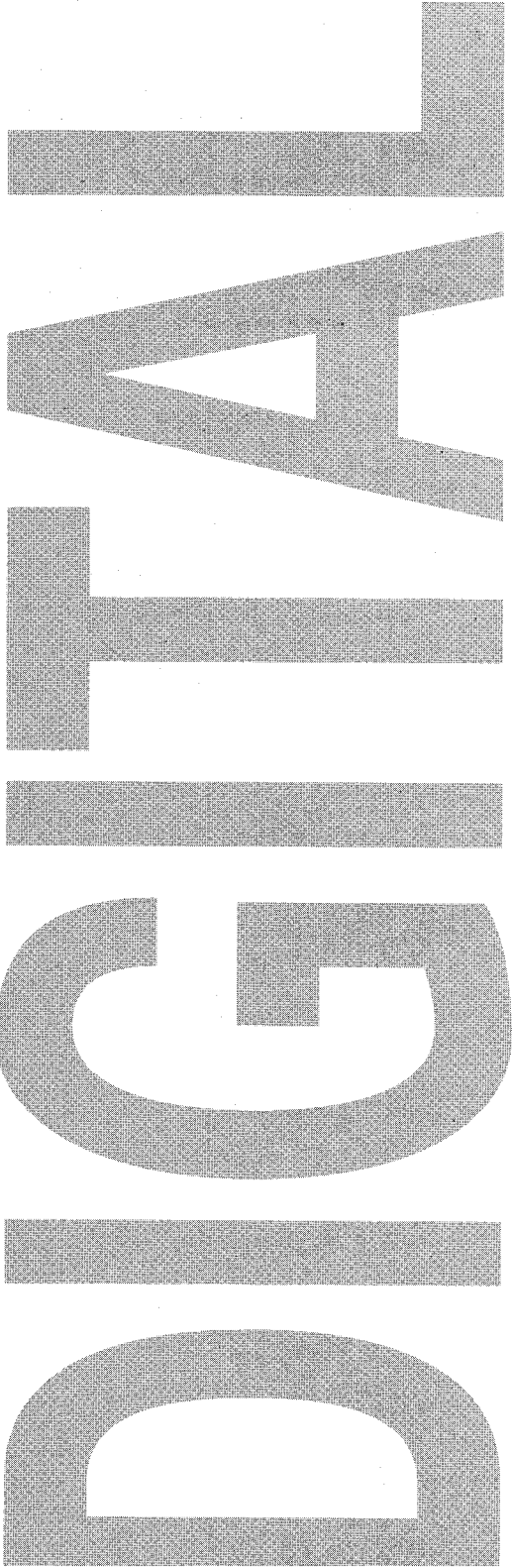
How a 2D image can have the appearance of 3D

How Cartesian coordinate specify objects positions

How to work orthographic and perspective projections

1.1. 3-D Perception

“3차원 컴퓨터 그래픽”은 대개 평평한 화면에 표현되는 2차원의 이미지로서, 눈의 착각을 이용해 3차원으로 인식케 된다. 한편, 실제세계(real World)에서 3차원 물체를 인식하게 되는 것은 그림 1-1로 설명될 수 있다. 두 눈은 망막에 비친 임시 사진과 같은 2차원의 이미지를 각각 받는다. 두 눈은 각기 다른 각도에서 보게 되기 때문에 이들 이미지들은 조금씩 다르다. 그림 1-1에서 보는 바와 같이 두뇌는 조금씩 이러한 다른 이미지들을 조합하여 머리속에 간단하고 복잡한 3차원의 그림을 만들어 낸다. 그 예를 Viewmaster와 3차원 영화에서 볼 수 있다. 한편, 3차원 물체가 멀어져 감에 따라



이미지의 θ 각이 작아지며, 두 이미지 사이의 θ 각이 클수록 3차원의 효과도 커진다.

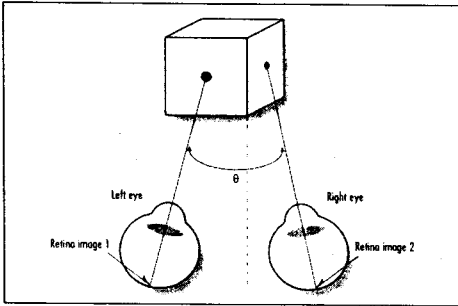


그림 1-1. How the eyes "see" three dimensions

1) 2D + Perspective = 3D

2차원 세계에도 3차원의 효과들이 나타난다. 그 중 요한 예로 가까운 물체가 먼곳의 것보다 더욱 커 보인다는 것이다. 이 효과를 Perspective(투시)라고 한다. 또한 색의 변화(color change), 재질(texture), 조명(lighting), 음영(shading) 또는 각종 농도(빛에 의한)들이 3차원의 효과들을 높여준다.

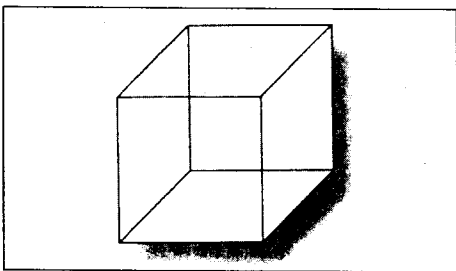


그림 1-2. This simple wireframe cube demonstrates perspective

2) Hidden Line Removal (숨겨진 선 없애기)

그림 1-2의 경우는 그림속에 면이 없는 것으로 보여 두뇌에 혼동을 갖게 하여, 앞뒷면의 구별이 곤란하다. 그림 1-3과 같이 뒤부분이 앞부분에 가리워지도록 하면 앞뒷면의 구분이 가능하게 되어 3차원의 효과가 보다 증대된다.

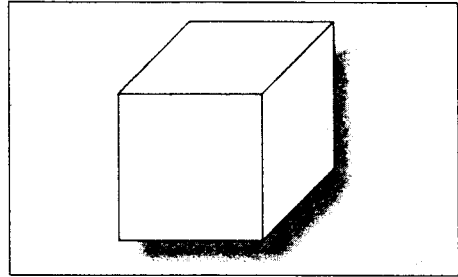


그림 1-3. The cube after hidden lines are removed

3) Color and Shading(색과 음영)

그림 1-3에서는 입방체 표면이 배경과 색이 똑같고 객체의 앞 모서리 밖에 볼 수 없다. 실제 객체는 어떤 색이나 재질이 있으므로 각 표면에 색을 넣되(그림 1-4 참조) 다른 색으로 칠하던지 아니면 각 표면에 음영을 넣어 빛의 효과를 낼 필요가 있다. (그림 1-5)

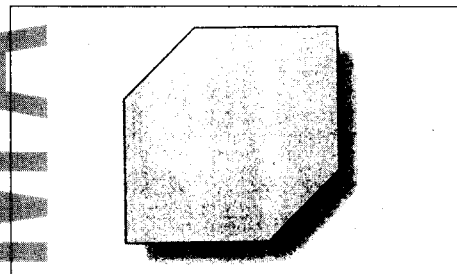


그림 1-4. The cube with color, but no shading

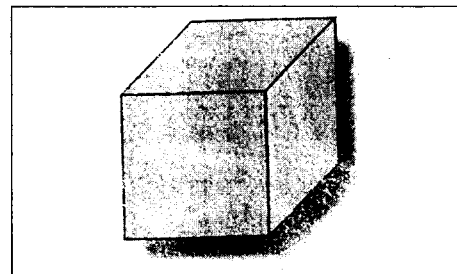


그림 1-5. The cube with its visible faces in three different shades

4) Lights and Shadows (빛과 그림자)

3차원 물체의 표현에 있어서 중요한 것중의 하나가 빛의 효과이다. 3차원에서 빛은 두가지 효과를 갖는다. 음영의 효과와 그림자의 효과가 그것이다. (그림 1-6)

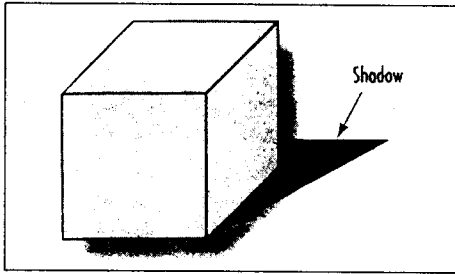


그림 1-6. A solid cube illuminated by a single light

1.2. Projection, The Essence of 3D

3차원의 물체를 출력하는 장치의 평평한 화면에 표현하는 것을 투사(Projection) 이라고 한다. (그림 1-7)

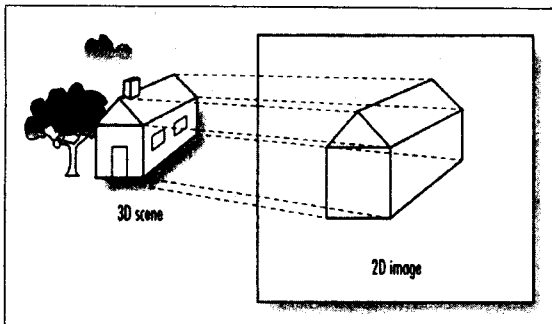


그림 1-7. A 3D image projected onto a 2D space

Projection에는 두 가지 방법이 있는데, 물체의 모든 점들을 평행선을 따라 2D 평면에 투사하는 경우와 모든 점들을 투사중심점(center of projection)이라고 불리는 지점과 연결되는 선을 따라 투사하는 경우이다.

이 두가지 방법을 각각 평행투사법(parallel projection), 원근투사법(perspective projection)이라고 한다. (그림 1-8)

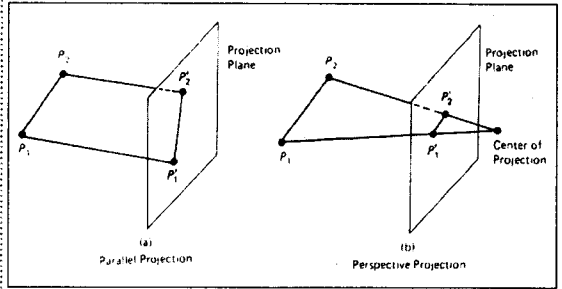


그림 1-8. 시계표면에 선분을 투사하는 두가지 방법

1) 평행투사법(Parallel Projection)

평행투사법에서는 물체의 상대거리를 그대로 유지하고 있어서, 3D 물체의 실물 확대, 축소 작업에 유용하게 쓰이는 기법이다. 이 평행투사법에 의하여 보이는 것은, 투사면과 투사각도에 따라서 달라지게 된다. 투사각이 투사면과 직각이 될 때에 이것을 직각투사법(orthographic projection) 이라고 하며, 직각이 아닌 경우에는 사각투사법(oblique projection)이라고 한다. (그림 1-9)

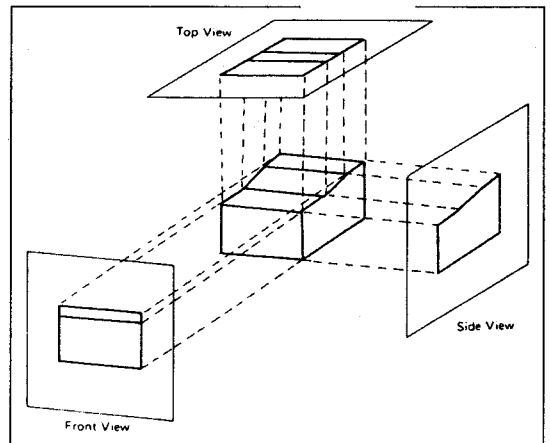


그림 1-9. 한 물체를 직각투사법에 의하여 세측면에서 보고 있다.

2) 원근투사법(Perspective Projection)

3차원 물체를 원근투사법에 의하여 투시도를 만들기 위해서는, 투사중심점에서 만나는 투사선을 따라서 점을 투사하여야 한다. 원근투사법에서는 먼 거리에 있는 선분은 작게 보이고 가까이 있는 선분은 보다 커 보인다. (그림 1-10)

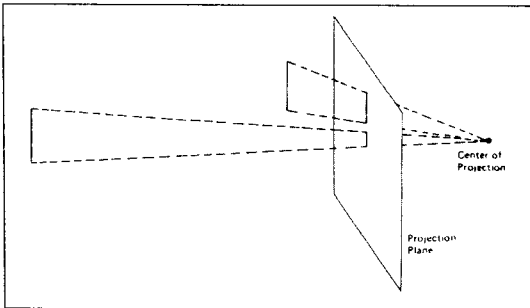


그림 1-10. 원근투사법

2. 3D Transformations(Three-Dimensional Transformations)

2.1. 3D위치전환(Translation)

3D 좌표의 표현에서 하나의 점(x, y, z) 지점에서(x', y', z') 지점으로 위치 이동하는 것은 다음 행렬연산에 의하여 수행된다. (그림 2-1)

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix} \quad (2-1)$$

(단, T_x, T_y, T_z 는 위치변환에 따르는 상대거리, 실수 값)

식(2-1)에서의 행렬식은 결국 다음과 같은 식과 동일하다.

$$x' = x + T_x, \ y' = y + T_y, \ z' = z + T_z \quad (2-2)$$

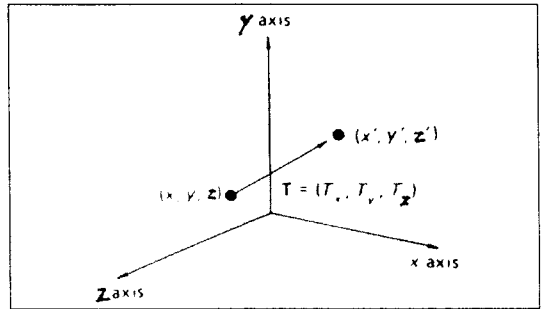


그림 2-1. 점에 대한 위치의 변환

다각형으로 정의된 3차원 물체에 대한 위치전환의 경우, 각 면의 꼭지점에 대하여 위치전환 작업을 수행한다.(그림 2-2)

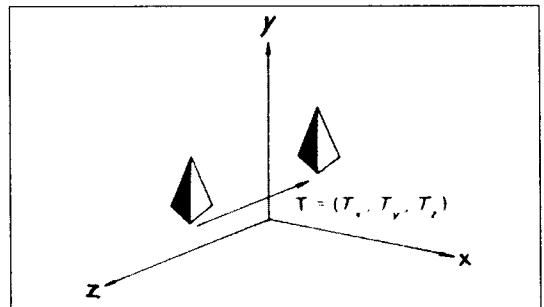


그림 2-2 물체에 대한 위치의 변환

2.2. 3D 확대, 축소(Scaling)

3차원 물체의 확대, 축소 작업을 위한 행렬식은 다음과 같다.

$$[x' y' z' 1] = [xyz 1] \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-3)$$

(단, S_x, S_y, S_z 는 임의의 양수)

즉, $x' = x \cdot S_x, y' = y \cdot S_y, z' = z \cdot S_z$ 이다. (2-4)

3차원의 물체에 적용될 때, 배울값이 모두 같지 않은 경우에는 물체의 모양이 변하게 된다. 이때 배울값이 모두 같아야만(즉 $S_x = S_y = S_z$) 본래의 모습을 그대로 갖는다. (그림 2-3)

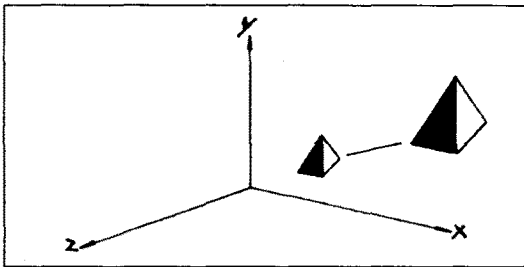


그림 2-3. 물체의 크기를 2배로 확대하면, 원점에서의 거리도 2배로 늘어난다.

또한, 원점의 고정점을 기준으로 확대, 축소하는 경우는 그림 2-4 와 같은 과정을 행해야 한다.

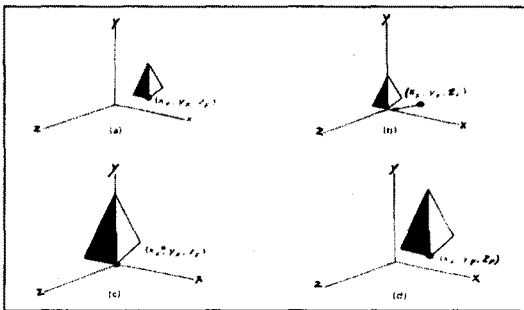


그림 2-4. 고정점을 기준으로 한 확대작업은 위와 같은 작업순서에 의한다.

2.3. 3D 회전(Rotation)

한 물체를 회전시킬 때, 반드시 물체가 어떤 축을 중심으로 회전하므로, 중심이 되는 회전축과 회전각도가 명시되어야 한다. 2D 회전에서는 회전축이 항상 xy 평면에 수직이었지만, 3D 회전의 경우, 회전축은 3차원 공간에서 임의의 방향이 될 수 있다. 여기서는 회전축을 쉽게 정의하기 위하여, 회전축이 좌표축과 항상 평행하게 하는 것이다. 임의의 회전축에 따라 회전시키기 위해서는 3개의 좌표축에 따라 적당히 회전시킴으로써 가능하다.

좌표상의 원점을 기준으로 양수값을 갖는 축의 방향에서 바라볼 때, 시계반대 방향으로 회전하는 각도를 양의 각도를 갖고 회전하는 것으로 정의한다. (그림 2-5)

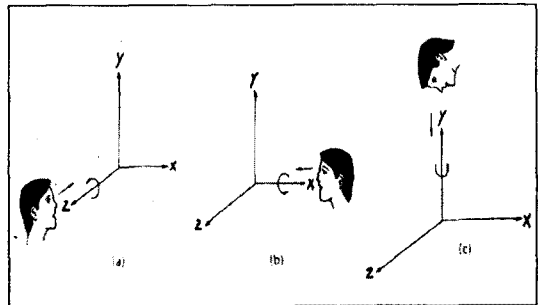


그림 2-5. 좌표축을 기준으로 양의 방향의 회전은 시계반대 방향이다.

z 축을 회전축으로 하는 회전의 경우, 2D의 xy 평면에서의 회전이 그대로 적용된다. 즉

$$\begin{aligned} z' &= z \cos \theta - x \sin \theta \\ x' &= z \sin \theta + x \cos \theta \\ y' &= y \end{aligned} \quad (2-5)$$

(단, θ 는 회전각도이다.)

이를 행렬로 표시하면,

$$[x' y' z' 1] = [x y z 1] \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-6)$$

그림 2-6은 물체를 z 축을 회전축으로 하여 회전시키는 것을 보여주고 있다.

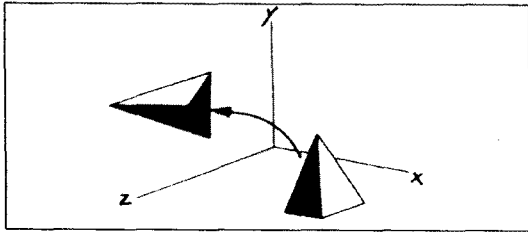


그림 2-6. z 축을 기준으로 할때의 회전작업

x 축 및 y 축에 대한 회전 변환식을 식 2-5에 좌표변수를 차례로 바꾸어 주면 된다. 즉, x 축을 기준으로 한 3D 회전식은

$$\begin{aligned} y' &= y \cos \theta - z \sin \theta \\ z' &= y \sin \theta + z \cos \theta \\ x' &= x \end{aligned} \quad (2-7)$$

이며 그림 2-7이 그 예이다.

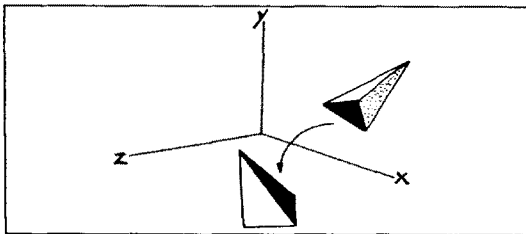


그림 2-7. x 축을 기준으로 할때의 회전작업

또한, y 축을 기준으로 한 3D 회전식은

$$\begin{aligned} z' &= z \cos \theta - x \sin \theta \\ x' &= z \sin \theta + x \cos \theta \\ y' &= y \end{aligned} \quad (2-8)$$

이며 그림 2-8이 그 예이다.

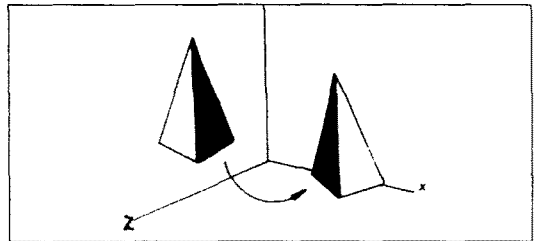


그림 2-8. y 축을 기준으로 할때의 회전작업

마지막으로, x 축과 평행인 임의의 축을 기준으로 한 회전작업의 처리순서를 그림 2-9에 표시하였다.

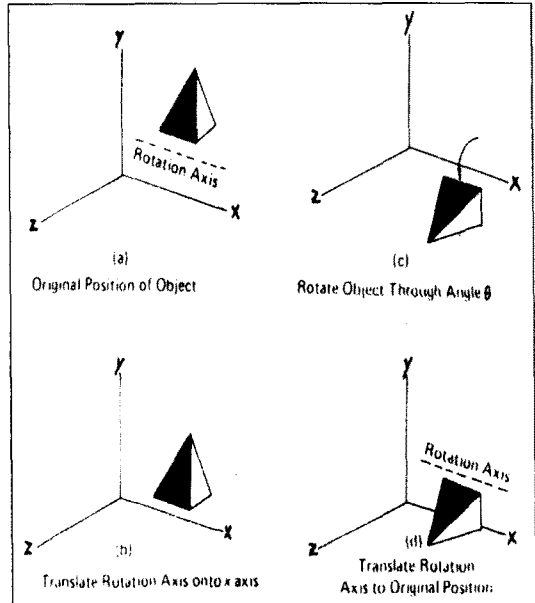


그림 2-9. x 축과 평행인 임의의 축을 기준으로 한 회전작업의 처리순서

2.4. 기타 전환(Other Transformations)

3D 응용에는 몇 가지 다른 전환 방법이 있다. 특정 평면에 의하여 반사되는 3D 반사(3D Reflection), 주어진 점의 3개의 좌표값중에서 2개의 값을 바꾸어 주는 변형(3D shears), 한 좌표계에서 다른 좌표계로 바꾸는 좌표축 변경(Transformation of Coordinate Systems) 등이 있다.

3. Volume Rendering & Surface Rendering

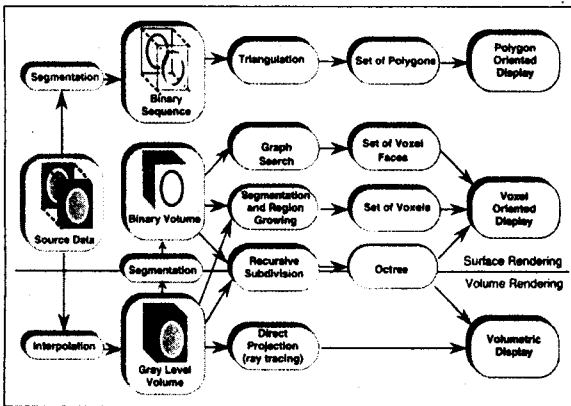


그림 3-1. Overview of surface and volume rendering methods.

3.1. Surface Rendering

1) Polygon-Oriented Technique

가장 최초로 개발된 방법으로서, parallel slices의 해부학적 구조를 3D로 표현할 수 있다. 이 고전적인 알고리즘은 각 slice에 대해 source data를 이진화시켜 윤곽(contour)을 구하고 서로 연결하고 있는 slice끼리 적당히 연결시켜 3D로 표현하는 방법이다. (예: 그림 3-2)

process 1) Contour Extraction

process 2) Curvature Sampling(The Generation of Control Points)

process 3) Shading Processing

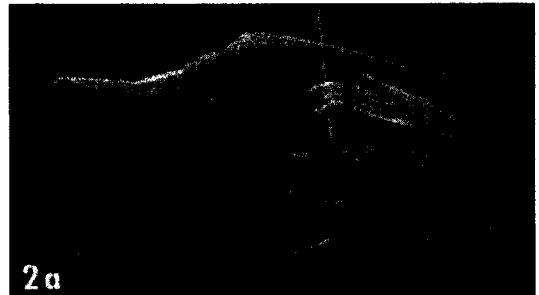


그림 3-2. Surface Rendering of a brain model(brain stems, ventricles, etc.) by using polygons

2) Voxel-Oriented Technique

3D Representation에서는 보다 발달된 방법으로서, 정확도와 신뢰성이 높은 새로운 알고리즘이다. 이 방법으로는 binary data 뿐만 아니라 gray-level data도 처리 가능하다. 여기서는 우선 3개의 좌표축으로 향하는 voxel의 homogeneous matrix를 구성하기 위하여 interpolation 작업을 필요로 한다. 그 다음으로는 Polygon-Oriented Technique의 Contour Extraction 처리를 3차원으로 확장시키는 개념으로 이웃하는 voxel



그림 3-3. Voxel-based surface rendered image of a human head

을 비교하여 surface 를 추적(tracing)해 나간다.(예 : 그림 3-3)

3.2. Volume Rendering

Volume Rendering의 주요점은 gray scale volume으로부터 직접 표현한다는 점이다. 그 대표적인 방법으로는 Ray Tracing Technique과 Octree Encoding Technique를 꼽을 수 있다.

1) Ray Tracing Technique

Octree Encoding Algorithm에 비하여 Octree를 준비하는 선처리과정이 불필요하다. 또한 1980년대 초부터 발달되어 온 기법으로서 의학영상에서 주로 사용되고 있다.

Ray란?

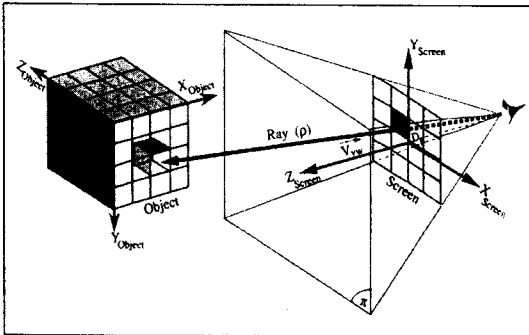


그림 3-4. Camera model for ray-tracing with perspective view

Process 1) Geometric Transformation

Process 2) Spatial Segmentation

Process 3) Gray Level Segmentation(예 : 그림 3-5)



그림 3-5. Volume Rendered image of a human brain by using a ray-tracing algorithm

2) Octree Encoding Technique

배면 및 배선제거에 용이하게 쓰이는 방법이며, 표면(surface)의 명암을 표현하기에 알맞은 알고리즘이다.

Octree란?

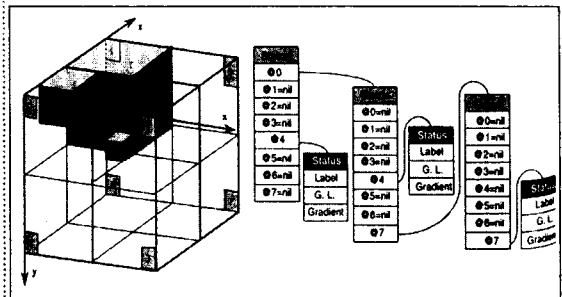


그림 3-6. Octree encoding with a computer representation by using a point structure



그림 3-7. Volume Rendered image of a human head by means of octree encoding

4. Example of 3D Medical Image Processing

4.1. 혈관의 3차원적인 구조분석을 이용한 예

〈폐암 자동진단 시스템에 기본적 연구〉

1) 개요

종양과 혈관의 CT 값이 매우 유사하여 양자의 차이를 CT 값으로 판별하는 것은 어렵다. 더욱이 굵은 혈관이 밀집되어 있는 폐문부의 경우는 더욱 곤란하며 또한 단일 슬라이스로부터 혈관과 종양을 구별하는 것도 쉽지 않다.

혈관은 차례차례 가지(Branch)로 나뉘어져 있는 3차원 Tree 구조를 하고 있고, 혈관의 굵기는 폐문부에서 멀어질수록 즉 폐의 외곽에 가까울수록 얇아진다. 이것을 이용하여 본 연구에서는 복잡한 폐문부부의 혈관을 3차원 패턴인식 관점에서 분석하였다. 다음으로, 혈관으로부터 떨어져 있는 영역과, 그리고 앞에서 기술한 의학적인 지식에 근거하여 혈관의 굵기에 이상이 있는 영역을 검출하여 그것을 종양일 가능성이 있는 영역으로 인식하는 방법을 채택한다.

안정적으로 혈관영상을 추출하기 위한 첫번째 처리단계로서 먼저 각 슬라이스의 영상에 대해 Smoothing 처리를 한다. 그 다음으로, Binary Image로 변환시켜 각 슬라이스로부터 혈관영상을 추출하지만 혈관과 종양이 불확실하게 혼재된 혈관영상이 나타난다. 따라서, 혈관 횡단면의 상하 접속관계를 조사한 후 혈관 Tree 구조를 구성하여 3차원적 분석을 한다. 이렇게 함으로써 혈관 Tree 구조에 포함되지 않은 혈관의 영역을 종양 후보로서 검출한다. 또한, 각 혈관의 가지(Branch) 굵기를 측정하여 해부학적으로 부자연스러운 혈관도 종양 후보로 검출한다. (그림 4-1)

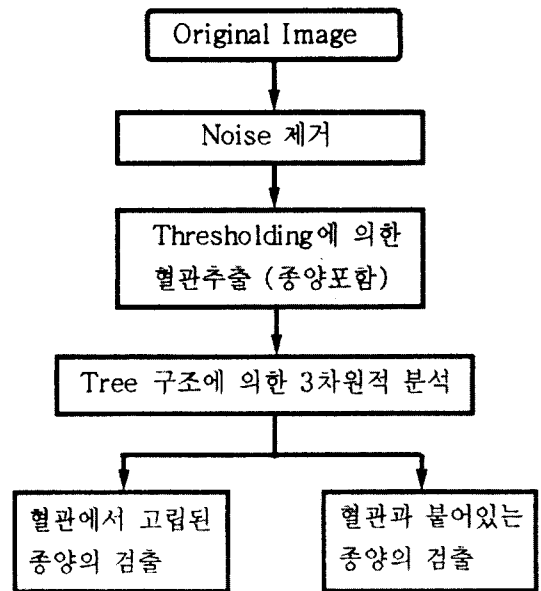


그림 4-1. 시스템의 개요도

2) 혈관영상의 3D Structure 구성

상하 slice의 비교를 통하여 혈관의 가능성이 있는 영역을 추출한다.

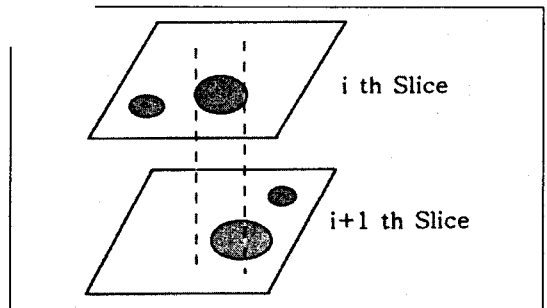


그림 4-2. 상하 slice에 대한 혈관 Tree의 Node 검색

3) 혈관 branches의 tracing을 통한 혈관굵기의 이상 추출

혈관 Tree로 구성된 각 가지(Branch)에 대하여 혈관

의 굵기를 구하고, Parent Node의 영역보다 Child Node의 영역이 넓은 경우(그림 4-3에서의 Node "B"에 해당), 즉 어떤 혈관 횡단면에 대해 심장(心腸) 쪽보다 말단(末端) 쪽의 굵기가 큰 경우에는 이상이 있는 국소로 판단하여 혈관에 부착되어 있는 종양 후보로 검출한다. 한편, 혈관의 굵기 측정은 각 혈관의 영역에 대해 거리변환을 계산하여 최대거리값을 혈관의 굵기(혈관의 반경 -Node 영역의 넓이)로 결정하는 방법을 채택한다.

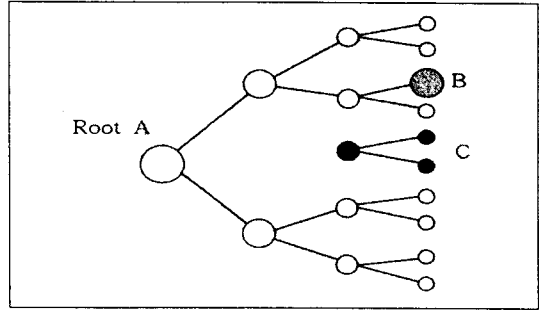


그림 4-3. 혈관 Tree구조의 구상도