

## 자기공명영상의 해부구조를 익히기 위한 3차원영상과 소프트웨어

이용숙, 정민석, 황성배, 왕지남<sup>1</sup>, 유승현<sup>2</sup>

아주대학교 의과대학 해부학교실,

<sup>1</sup>아주대학교 산업정보시스템공학부, <sup>2</sup>아주대학교 기계공학부

**간추림** : 의과대학 학생과 의사는 자기공명영상에서 정상 해부구조가 어떻게 보이는지 익혀야 한다. 이를 위하여 건강한 사람 온몸의 수평, 관상, 시상자기공명영상과 이 자기공명영상에 들어맞는 구역화영상을 마련하였다. 그런데 이것만 갖고는 관심 있는 자기공명영상과 구역화영상을 골라서 보기 불편하고, 자기공명영상에서 입체 해부구조가 어떻게 보이는지 깨닫기 어렵다. 따라서 이 연구에서는 수평, 관상, 시상자기공명영상 중에서 관심 있는 자기공명영상을 손쉽게 골라서 볼 수 있고, 들어맞는 구역화영상을 함께 볼 수 있는 소프트웨어를 만들었다. 또한 구역화영상을 쌓고 수동으로 surface rendering해서 각 해부구조의 3차원영상을 재구성한 다음에 관심 있는 3차원영상을 손쉽게 골라서 보고 돌려서 볼 수 있는 소프트웨어를 만들었다. 이 연구에서 만든 소프트웨어를 쓰면 자기공명영상에서 입체 해부구조가 어떻게 보이는지 깨닫는 데 도움이 되며, 나아가 환자의 자기공명영상에서 병리구조를 판독하는 데 도움이 될 것이다.

**찾아보기 낱말** : 자기공명영상, 해부구조, 구역화영상, Surface rendering, 3차원영상, 소프트웨어

### 서 론

자기공명영상에서 병리구조가 어떻게 보이는지 알기 위해서는 먼저 자기공명영상에서 정상 해부구조가 어떻게 보이는지 익혀야 한다. 이를 위하여 한국사람 온몸의 수평, 관상, 시상 자기공명영상을 마련하였고, 이 자기공명영상에 들어맞는 구역화영상을 마련하였다(김진용 등 2003). 그런데 이것만 갖고는 관심 있는 자기공명영상을 골라서 보기 불편하고, 자기공명영상에 들어맞는 구역화영상을 함께 보기 불편하고, 자기공명영상에서 입체 해부구조가 어떻게 보이는지 깨닫기 어렵다. 따라서 이 연구에서는 들어맞는 자기공명영상과 구역화영상을 손쉽게 골라서 볼 수 있고, 자기공명영상에서 입체 해부

구조가 어떻게 보이는지 쉽게 깨달을 수 있는 교육 자료를 마련하기로 하였다. 이를 위하여 이미 마련한 수평, 관상, 시상 자기공명영상과 구역화영상을 손쉽게 골라서 보는 소프트웨어를 만들었고, 구역화영상을 쌓고 수동으로 surface rendering해서 각 해부구조의 3차원영상을 재구성한 다음에 이 3차원영상을 손쉽게 골라서 보는 소프트웨어를 만들었다.

### 재료 및 방법

#### 1. 수평, 관상, 시상 자기공명영상과 구역화영상 만들기

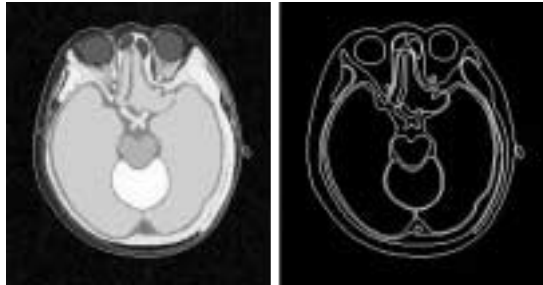
대상자를 뽑아서 온몸의 자기공명영상을 찍었다. 표준 체형을 갖고 있고 건강한 한국 남성(19세, 키 1,760 mm, 몸무게 67 kg)을 대상으로 뽑았다. 온몸의 자기공명영상 613개 (간격 3 mm, 해상도 512×274,

\* 이 연구는 과학기술부 특정연구개발사업(2001년, 2002년)의 지원으로 이루어졌음.

교신저자: 정민석(아주대학교 의과대학 해부학교실)  
전자우편: dissect@ajou.ac.kr

**Table 1.** Features of horizontal, coronal, sagittal MRIs and segmented images

Dataset of images (File names)	Intervals	Resolution	Bit depth	One file size	Total file size
Horizontal MRIs (hm001.tif-hm613.tif)	3 mm	512 × 274	8 bits gray	145 Kbytes	87 Mbytes
Horizontal segmented images (hs001.tif-hs613.tif)	3 mm	512 × 274	24 bits color	412 Kbytes	249 Mbytes
Coronal MRIs (cm001.tif-cm274.tif)	1 mm	512 × 1,839	8 bits gray	928 Kbytes	249 Mbytes
Coronal segmented images (cs001.tif-cs274.tif)	1 mm	512 × 1,839	24 bits color	2,759 Kbytes	738 Mbytes
Sagittal MRIs (sm001.tif-sm512.tif)	1 mm	274 × 1,839	8 bits gray	499 Kbytes	249 Mbytes
Sagittal segmented images (ss001.tif-ss512.tif)	1 mm	274 × 1,839	24 bits color	1,477 Kbytes	738 Mbytes
Total					2,310 Mbytes



**Fig. 1.** Segmented image (left) and outline image (right).

빛깔 8 bits gray, T1 강조, NEX 2, interleave 방법)를 찍은 다음에 개인용컴퓨터로 옮겨서 TIFF 파일 (hm001.tif - hm613.tif)로 저장하였다. 이 때 자기공명영상을 아래에서 보게 하였고, 첫째 자기공명영상 (hm001.tif)을 대상자의 위 끝으로 결정하였다 (Table 1).

자기공명영상을 바탕으로 임신구역화영상을 만들었고, 임신구역화영상을 바탕으로 구역화영상을 만들었다. 자기공명영상 613개를 CorelDRAW 소프트웨어 (10.0판, Corel Corporation<sup>TM</sup>)로 옮겼다. 자기공명영상 위에 해부구조 59개 (Table 2)의 테두리를 수동으로 그려서 임신구역화영상 613개를 만든 다음에 CDR 파일 (001.cdr - 613.cdr)로 저장하였다. 임신구역화영상에서 해부구조의 테두리 속에 빛깔을 채워서 구역화영상(빛깔 24 bits color) 613개를 만든 다음에 TIFF 파일 (hs001.tif - hs613.tif)로 저장하였

**Table 2.** Fifty-nine anatomical structures, which are segmented

Systems	Anatomical structures
Skeletal	Skull (except mandible), Mandible, Rib, Sternum*, Vertebra, Clavicle*, Scapula*, Humerus*, Radius*, Ulna*, Carpal bone, Metacarpal bone, Phalanx of hand, Hip bone*, Femur*, Patella, Tibia*, Fibula*, Tarsal bone, Metatarsal bone, Phalanx of foot
Muscular	Muscle
Digestive	Oral cavity, Palate, Parotid gland, Submandibular gland, Esophagus, Stomach, Small intestine, Large intestine, Liver, Gallbladder, Pancreas
Respiratory	Nasal cavity, Nasopharynx, Oropharynx, Larynx, Thyroid cartilage, Trachea and bronchus, Lung
Urinary	Kidney, Ureter, Urinary bladder
Genital	Testis, Prostate
Endocrine	Thyroid gland
Cardiovascular	Heart, Blood vessel
Lymphatic	Thymus, Spleen
Nervous	Cerebrum, Cerebellum, Brain stem, Spinal cord, Meninx, Optic nerve
Sensory	Eye ball, Skin

\* Spongy bones of these bones are also segmented.

다 (Fig. 1, Table 1).

모든 자기공명영상과 구역화영상의 행을 쌓아서 관상자기공명영상 (cm001.tif - cm274.tif)과 관상구역화영상 (cs001.tif - cs274.tif) 274쌍 (해상도 512 × 1,839)을 만들었다. 이 때 관상영상을 앞에서 보게 하였고, 첫째 관상영상 (cm001.tif, cs001.tif)을 대상

자의 앞 끝으로 결정하였다. 모든 자기공명영상과 구역화영상의 열을 쌓아서 시상자기공명영상(sm001.tif - sm512.tif)과 시상구역화영상(ss001.tif - ss512.tif) 512쌍(해상도 274 × 1,839)을 만들었다. 이 때 시상영상을 왼쪽에서 보게 하였고, 첫째 시상영상(sm001.tif, ss001.tif)을 대상자의 오른쪽 끝으로 결정하였다(Table 1).

## 2. 각 해부구조의 3차원영상 만들기

임시구역화영상에서 자기공명영상을 지우고 해부구조의 테두리만 추려서 테두리영상 613개를 만든 다음에 AI (Adobe Illustrator) 파일 (001.ai - 613.ai)로 저장하였다(Fig. 1).

3ds max 소프트웨어 (5.0판, Discreet™)에서 테두리영상 613개를 모눈선(grid) 613개에 놓아 테두리3차원영상을 만들었다. 모눈선 사이의 간격을 적당하게 바꾸었다. 이를 위하여 대상자의 몸 크기와 자기

공명영상의 간격(3 mm)을 고려해서 모눈선 사이의 간격을 결정하였고, 이 간격을 맞추기(customize)의 모눈선과잡기를정하기(grid and snap settings)의 집 모눈선(home grid)의 모눈선간격(grid spacing)에 입력하였다(Fig. 2). 테두리영상 613개를 모눈선 613개에 하나씩 놓았다. 이를 위하여 첫째 테두리영상(001.ai)을 불러서 첫째 모눈선에 놓았는데, 이 때 앞보기(front view) 창에서 첫째 테두리영상을 첫째 모눈선의 언저리에 놓으면 첫째 테두리영상이 자동으로 첫째 모눈선에 들어맞았다. 마찬가지로 방법으로 나머지 테두리영상 612개를 하나씩 불러서 각각의 모눈선에 놓았다. 테두리영상을 모두 놓아서 만든 테두리3차원영상을 MAX 파일로 저장하였다.

해부구조 47개의 테두리3차원영상을 만들었다. 해부구조 59개 중에서 대체로 뚜렷하지 않은 해부구조 12개(손목뼈, 근육, 입안, 입천장, 식도, 큰창자, 쓸개, 후두, 혈관, 대뇌, 소뇌, 해면뼈)를 빼 나머지 해부

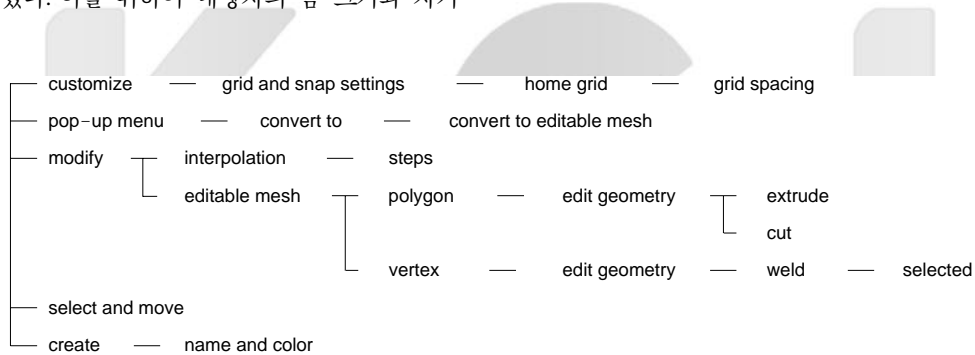


Fig. 2. Menus of the polygon method for manual surface rendering on the 3ds max software.

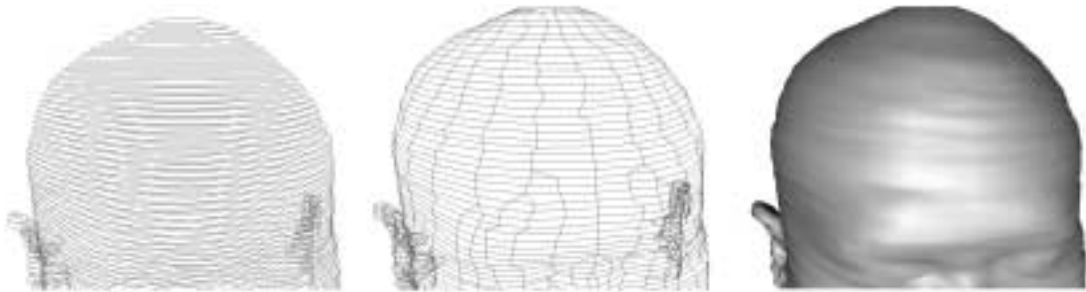


Fig. 3. 3D image of the skin outlines (left), 3D image of the skin before coloring, which is reconstructed by the polygon method for manual surface rendering (center), and 3D image of the skin after coloring (right).

구조 47개의 테두리3차원영상을 만들기로 하였다 (Table 3). 테두리3차원영상에서 각 해부구조의 테두

**Table 3.** Forty-seven anatomical structures, 3D images of which are reconstructed

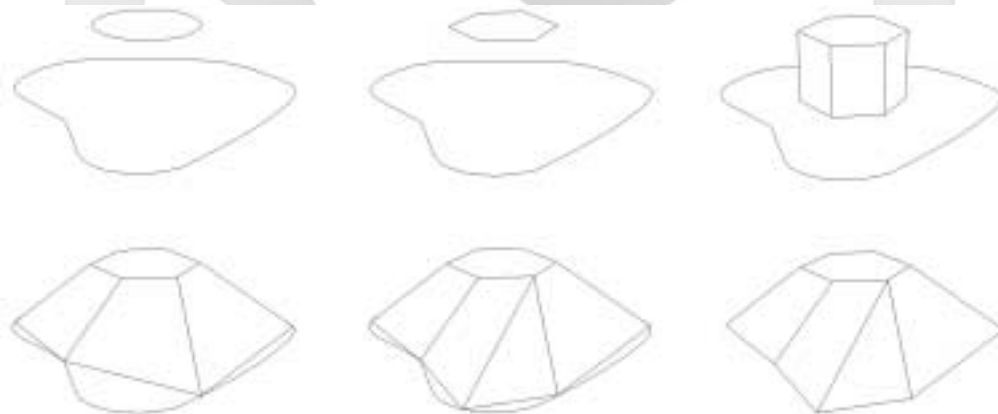
Systems	Anatomical structures (File size of 3D images, Mbytes)
Skeletal	Skull (3.3), Mandible (3.2), Rib (3.0), Sternum (2.7), Vertebra (4.0), Clavicle (6.0), Scapula (6.0), Humerus (6.3), Radius (5.0), Ulna (5.0), Metacarpal bone (4.9), Phalanx of hand (5.0), Hip bone (2.8), Femur (5.2), Patella (4.7), Tibia (5.2), Fibula (5.6), Tarsal bone (4.8), Metatarsal bone (4.8), Phalanx of foot (4.7)
Digestive	Parotid gland (3.1), Submandibular gland (3.0), Stomach (2.7), Small intestine (2.6), Liver (3.9), Pancreas (2.6)
Respiratory	Nasal cavity (3.1), Nasopharynx (3.2), Oropharynx (3.2), Thyroid cartilage (3.0), Trachea and bronchus (3.1), Lung (5.6)
Urinary	Kidney (3.3), Ureter (2.4), Urinary bladder (2.6)
Genital	Testis (2.4), Prostate (2.4)
Endocrine	Thyroid gland (3.6)
Cardiovascular	Heart (3.9)
Lymphatic	Thymus (2.7), Spleen (2.9)
Nervous	Brain stem (3.3), Spinal cord (3.2), Meninx (3.4), Optic nerve (3.1)
Sensory	Eye ball (6.1), Skin (4.4)
Total	(181.0)

리를 추려서 각 해부구조의 테두리3차원영상을 만든 다음에 역시 MAX 파일로 저장하였다. 보기를 들어 피부의 테두리를 추려서 피부의 테두리3차원영상을 만들었다 (Fig. 3). 이 때 각 해부구조의 테두리 3차원영상을 원래 모눈선에 놓았다.

테두리3차원영상을 살핀 다음에 틀린 임시구역화영상을 고쳤다. 3ds max 소프트웨어에서 각 해부구조의 테두리3차원영상을 돌려 보아서 테두리3차원영상이 위아래로 매끈한지 살폈다 (Fig. 3). 테두리3차원영상이 위아래로 매끈하지 않았으면, 즉 임시구역화영상이 틀렸으면, 임시구역화영상에서 해부구조의 테두리를 수동으로 고쳤다. 테두리영상과 테두리 3차원영상을 다시 만든 다음에 테두리3차원영상을 다시 살폈다.

3ds max 소프트웨어에서 각 해부구조의 테두리3차원영상을 polygon 방법으로 surface rendering해서 각 해부구조의 3차원영상을 수동으로 재구성하였다 (Fig. 3). 이를 위하여 어느 해부구조의 테두리3차원영상에서 첫째 테두리와 둘째 테두리 사이에 사각형과 삼각형을 다음처럼 채웠다.

첫째로 어느 해부구조의 테두리3차원영상을 부른 다음에 닫힌곡선인 첫째 테두리를 다각형으로 바꾸었으며, 이 결과로 다각형이 첫째 테두리의 노릇을



**Fig. 4.** Polygon method for manual surface rendering between the first and second outlines. The first original outline is converted into a polygon (top left to top center). The first outline is extruded to the level of the second outline to make vertical rectangles (top center to top right). Lower vertices of the vertical rectangles are moved to the second outline to make quadrangles between the first and second outlines (top right to bottom left). A triangle is made for fitting the long and complex second outline (bottom left to bottom center). The second original outline is deleted (bottom center to bottom right).

하게 되었다(Fig. 4). 이를 위하여 첫째 테두리를 고른 다음에 불쑥차림표(pop-up menu)의 바꾸기(convert to)의 고칠그물로바꾸기(convert to editable mesh)를 눌렀다(Fig. 2). 이 때 첫째 테두리의 본디 모습을 거의 간직하는 범위에서 다각형의 꼭지점을 되도록 적게 만들었다(Fig. 4). 이를 위하여 첫째 테두리를 다각형으로 바꾸기에 앞서 고치기(modify)의 보간(interpolation)의 단계값(steps, 1 - 100)에 적당한 값을 입력하였다(Fig. 2). 둘째 테두리 중에서 첫째 테두리와 이어지지 않는 테두리가 있는 경우에는 이 둘째 테두리도 마찬가지로 방법으로 다각형으로 바꾸었다.

둘째로 첫째 테두리를 둘째 모눈선까지 밀어서 여러 직사각형으로 이루어진 수직직사각형을 만들었다(Fig. 4). 이를 위하여 첫째 테두리를 고른 다음에 모눈선 사이의 간격을 고치기의 고칠그물(editable mesh)의 다각형(polygon)의 외형고치기(edit geometry)의 밀기(extrude)의 값에 입력한 다음에 밀기를 눌렀다(Fig. 2).

셋째로 수직직사각형의 아래 꼭지점을 둘째 테두리로 끌어서 첫째 테두리와 둘째 테두리 사이에 사각형을 채웠다. 수직직사각형의 아래 꼭지점을 둘째 테두리로 끌었으며, 이 결과로 수직직사각형을 이루는 직사각형이 사각형으로 바뀌었고, 이 사각형은 모서리가 직선이었고 면이 꼭면이었다(Fig. 4). 이를 위하여 고치기의 고칠그물의 꼭지점(vertex)을 누르고, 골라서움직이기(select and move)를 누른 다음에 수동으로 수직직사각형의 아래 꼭지점을 둘째 테두리로 끌었다(Fig. 2). 이 때 아래 꼭지점을 클릭하면 x축, y축, z축이 나타났는데, x축과 y축은 둘째 모눈선에 놓인 축이었고, z축은 둘째 모눈선에 수직인 축이었다. 따라서 아래 꼭지점을 x축과 y축으로만 끌어서 아래 꼭지점이 둘째 모눈선에서 벗어나지 않게 하였다.

넷째로 첫째 테두리와 둘째 테두리가 많이 다른 첫째 테두리와 둘째 테두리 사이에 삼각형을 채웠다. 둘째 테두리가 첫째 테두리보다 훨씬 더 길거나 복잡하면 둘째 테두리에 더 많은 꼭지점을 놓아서 삼각형을 만들었으며, 이 결과로 만들어진 삼각형은 모서리가 직선이었고 면이 직면이었다(Fig. 4).

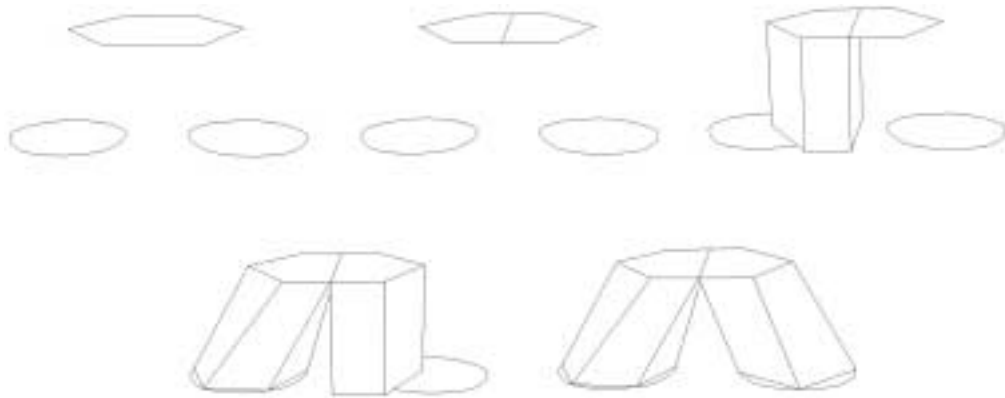


Fig. 5. Part of the polygon method for manual surface rendering. A triangle is made for fitting the short and simple second outline (left to right).

이를 위하여 고치기의 고칠그물의 다각형의 외형고치기의 자르기(cut)를 누른 다음에 사각형의 아래 꼭지점 사이의 모서리를 클릭해서 새 꼭지점을 만들었고, 사각형의 위 꼭지점을 클릭해서 새 꼭지점과 위 꼭지점을 잇는 새 모서리를 만들었고, 새 꼭지점을 둘째 테두리로 끌었다(Fig. 2). 둘째 테두리가 첫째 테두리보다 훨씬 더 짧거나 단순하면 둘째 테두리에 더 적은 꼭지점을 놓아서 삼각형을 만들었다(Fig. 5). 이를 위하여 사각형의 아래 꼭지점 2개를 고른 다음에 고치기의 고칠그물의 꼭지점의 외형고치기의 합치기(weld)의 골라짐(selected)을 눌러서 꼭지점 2개를 1개로 합쳤고, 이 꼭지점을 둘째 테두리로 끌었다(Fig. 2).

다섯째로 닫힌곡선인 둘째 테두리를 지웠으며, 이 결과로 사각형과 삼각형의 아래 모서리로 이루어진 다각형이 둘째 테두리의 노릇을 하게 되었다(Fig. 4). 이를 위하여 닫힌곡선인 둘째 테두리를 고르고 자판의 지우기(delete) 단추를 눌렀다.

첫째 테두리가 1개이고, 둘째 테두리가 2개(왼테두리와 오른테두리)인 경우에는 사각형을 다음처럼 채웠다. 첫째로 첫째 테두리를 왼테두리와 오른테두리로 나누었다(Fig. 6). 이를 위하여 고치기의 고칠그물의 다각형의 외형고치기의 자르기를 누른 다음에 첫째 테두리의 꼭지점 사이의 모서리를 클릭해서 새 꼭지점을 만들었고, 다른 꼭지점 사이의 모서리를 클릭해서 새 꼭지점을 또 만들었으며, 이 결과로 새 꼭지점 2개를 잇는 모서리를 경계로, 첫째 테두리가 왼테두리와 오른테두리로 나뉘었다(Fig. 2). 둘째로 첫째 왼테두리를 둘째 모눈선까지 밀어서 원수직직사각형을 만들었다(Fig. 6). 셋째로 원수직직사각형의 아래 꼭지점을 둘째 왼테두리로 끌어서 첫째 왼테두리와 둘째 왼테두리 사이에 사각형을



**Fig. 6.** Part of the polygon method for manual surface rendering between the first outline and second two outlines. The first outline is divided into the first left and first right outlines (top left to top center). The first left outline is extruded to make left vertical rectangles (top center to top right). Lower vertices of the left vertical rectangles are moved while the first right outline is extruded (top right to bottom left). Lower vertices of the right vertical rectangles are moved (bottom left to bottom right).

채웠다. 아울러 첫째 오른테두리를 둘째 모눈선까지 밀어서 오른수직직사각형을 만들었다 (Fig. 6). 넷째로 오른수직직사각형의 아래 꼭지점을 둘째 오른테두리로 끌어서 첫째 오른테두리와 둘째 오른테두리 사이에 사각형을 채웠다 (Fig. 6).

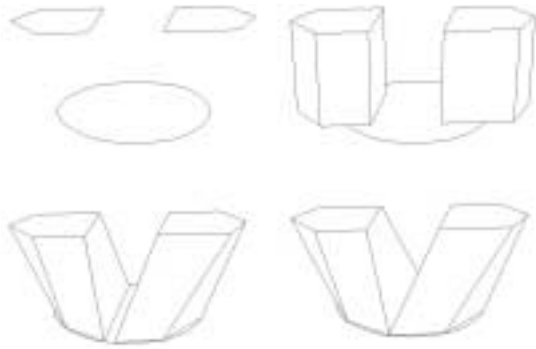
첫째 테두리가 2개 (왼테두리와 오른테두리)이고 둘째 테두리가 1개인 경우에는 사각형을 다음처럼 채웠다. 첫째로 첫째 왼테두리와 오른테두리를 둘째 모눈선까지 밀어서 왼수직직사각형과 오른수직직사각형을 만들었다 (Fig. 7). 둘째로 왼수직직사각형과 오른수직직사각형의 아래 꼭지점을 둘째 테두리로 끌었으며, 이 때 왼수직직사각형의 아래 모서리 1개와 오른수직직사각형의 아래 모서리 1개가 거의 겹치게 하였다 (Fig. 7). 셋째로 거의 겹친 아래 모서리 2개를 합쳤으며, 이 결과로 다각형 1개가 둘째 테두리의 노릇을 하게 되었다 (Fig. 7). 이를 위하여 거의 겹친 꼭지점 2개를 고른 다음에, 고치기의 고칠그물의 꼭지점의 외형고치기의 합치기의 골라짐을 늘려서 꼭지점 2개를 1개로 합쳤으며, 이 일을 한 번 더 되풀이하였다 (Fig. 2).

첫째 테두리와 둘째 테두리 사이에 사각형과 삼각형을 채우는 방법을 나머지 모든 테두리 사이에서 되풀이하여, 즉 수동으로 surface rendering하여,

이 해부구조의 3차원영상을 재구성하였다 (Fig. 3). 3차원영상의 모든 사각형과 삼각형에 특정한 빛깔을 칠했고 (Fig. 3), 3차원영상을 MAX 파일로 저장하였다. 이를 위하여 3차원영상을 통째로 고른 다음에 만들기 (create)의 이름과빛깔 (name and color)을 늘려서 특정한 빛깔을 칠했으며 (Fig. 2), 이 때 첫째 테두리와 마지막 테두리의 속에도 같은 빛깔이 칠해졌다. 자판의 F3 단추를 누르면 빛깔을 칠한 3차원영상, 즉 사각형과 삼각형이 보이지 않는 3차원영상이 나타났고, 다시 자판의 F3 단추를 누르면 빛깔을 칠하지 않은 3차원영상, 즉 사각형과 삼각형이 보이는 3차원영상이 나타났다 (Fig. 3). 이처럼 빛깔을 칠한 3차원영상을 MAX 파일로 저장하였다. 같은 방법을 되풀이해서 해부구조 47개의 3차원영상을 재구성하였다 (skull-3d.max, mandible-3d.max, ...) (Table 3). 이 때 3차원영상의 원래 위치를 간직하였으며, 이 결과로 여러 3차원영상을 함께 보면 해부구조의 위치 관계를 볼 수 있었다 (Fig. 8).

### 3. 자기공명영상, 구역화영상, 3차원영상을 손쉽게 보는 소프트웨어 만들기

자기공명영상, 구역화영상, 3차원영상을 손쉽게 보는 소프트웨어를 만들었다. Visual C++ 프로그래밍



**Fig. 7.** Part of the polygon method for manual surface rendering between the first two outlines and second outlines. Both the first left and right outlines are extruded to make left and right vertical rectangles (top left to top right). Lower vertices of the left and right vertical rectangles are moved (top right to bottom left). Neighboring lower vertices of the left and right quadrangles are united (bottom left to bottom right).



**Fig. 8.** 3D image of the skin (left) and that of all anatomical structures except the skin (right).

언어 (6.0판, Microsoft™)로 개인용컴퓨터에서 쓸 수 있는 소프트웨어를 만들어서 자기공명영상, 구역화 영상, 3차원영상을 다음처럼 보게 하였다. 세 묶음 (수평, 관상, 시상)의 자기공명영상 중에서 한 묶음의 자기공명영상을 고른 다음에 관심 있는 자기공명영상 1개를 골라서 보게 하였다. 자기공명영상을 확대해서 보고 옮겨서 보게 하였으며, 언제나 자기공명영상에 들어맞는 구역화영상을 함께 보게 하였다. 해부구조 47개의 3차원영상 중에서 관심 있는 3차원영상을 겹쳐서 고른 다음에 3차원영상을 불투명하게 또는 반투명하게 만들어서 보게 하였고, 확대해서 보고 옮겨서 보게 하였고, 돌려서 보게 하였다. 이 때 3차원영상을 원근방법 (perspective)으로 처리해서 보게 하였고, 3차원영상에 그림자를 넣어서 보게 하였다. 구역화영상과 3차원영상에서 나타난 해부구조의 이름을 보게 하였다. 모든 영상을 손쉬운 사용자사이드 (user interface)을 써서 보게 하였다.

## 결 과

해부구조 47개의 3차원영상을 마련하였다. 자기공명영상에서 대체로 뚜렷한 해부구조 47개의 3차원영상을 마련하였는데, 이 3차원영상은 파일형식이 MAX 파일이었고, 전체 파일크기가 181 Mbytes였다 (Table 3).

3차원영상이 대체로 올라왔다. 3ds max 소프트웨어에서 각 해부구조의 3차원영상을 돌려서 본 결과, 3차원영상의 생김새가 해부학 지식에 들어맞았다 (Fig. 9). 각 해부구조의 3차원영상을 반투명한 피부, 뼈의 3차원영상과 함께 돌려서 본 결과, 3차원영상의 위치가 올라왔다. 이웃한 해부구조의 3차원영상을 함께 돌려서 본 결과, 3차원영상의 위치 관계가 올라왔다 (Fig. 8). 피부의 3차원영상을 돌려서 본 결과, 3차원영상의 생김새가 대상자의 사진에 들어맞았고 (Fig. 10), 생체계측값의 비율에 들어맞았다.

자기공명영상, 구역화영상, 3차원영상을 손쉽게 보는 소프트웨어를 마련하였다. 이 소프트웨어는 개인용컴퓨터에서 쓸 수 있었다. 이 소프트웨어는 3차원영상이 없어도 쓸 수 있었으나, 자기공명영상과 구

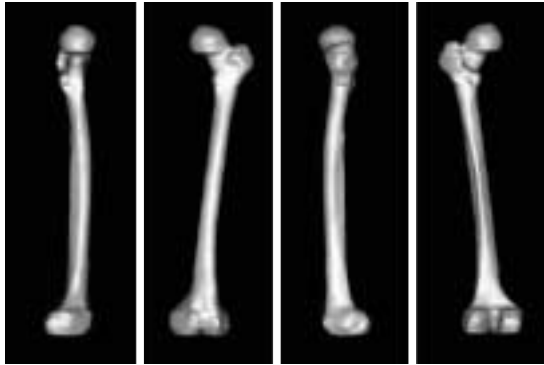


Fig. 9. Rotated 3D image of the femur.



Fig. 10. 3D image of the skin (left) and photograph of the volunteer (right).

역화영상이 없으면 쓸 수 없었다. 따라서 이 소프트웨어(파일크기 41 Mbytes)를 쓰기 위해서는 자기공명영상과 구역화영상(파일크기 2,310 Mbytes)을 함께 개인용컴퓨터의 하드 디스크에 복사해야 했다.

자기공명영상, 구역화영상을 손쉽게 보는 소프트웨어의 기능은 다음과 같았다.

초기 화면에서 1번 - 613번 수평영상의 가운데인 306번 수평영상을 볼 수 있었으며, 이 수평영상의 번호(306번)도 함께 볼 수 있었다. 수평영상의 번호를 입력해서 다른 수평영상을 골라서 볼 수 있었다. ‘앞 번호로 가기’ 또는 ‘뒤 번호로 가기’ 단추를 눌러서 이웃한 수평영상을 넘겨서 볼 수 있었다. ‘앞 번호로 잇달아 가기’ 또는 ‘뒤 번호로 잇달아 가기’ 단추를 눌러서 수평영상을 잇달아 넘겨서 볼 수 있었고, 잇달아 넘겨서 보다가 멈추어서 볼 수 있었다. 수평영상을 골라서 보고 넘겨서 볼 때마다 서로 들어맞는 자기공명영상과 구역화영상을 함께 볼 수 있었다(Fig. 11).

‘확대하기’ 단추를 누를 때마다 수평영상을 2배씩 확대해서 볼 수 있었고, 16배까지 확대해서 볼 수 있었다. 두루마리막대(scroll bar)를 써서 확대한 수평영상을 옮겨서 볼 수 있었다. 거꾸로 ‘축소하기’ 단추를 눌러서 수평영상을 본디 배율(1배)로 축소해서 볼 수 있었다. 수평영상을 확대해서 보고 옮겨서 보고 축소해서 볼 때마다 서로 들어맞는 자기공

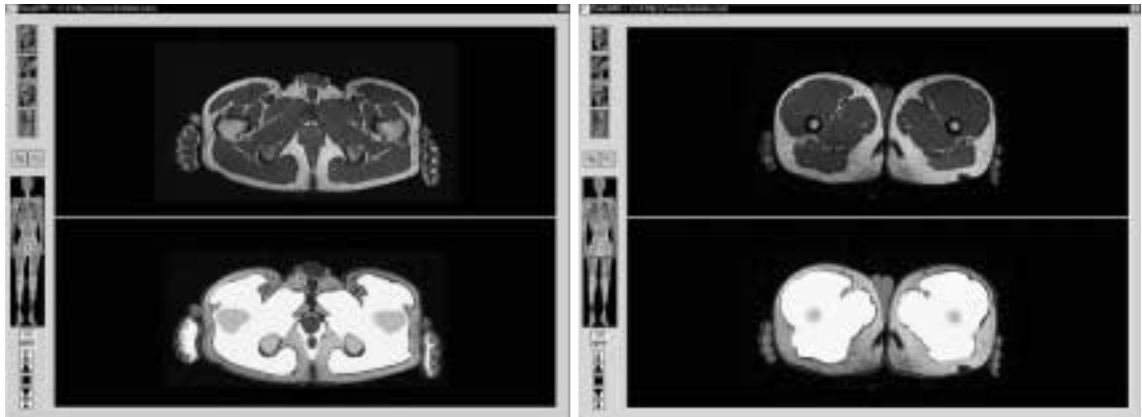


Fig. 11. Corresponding horizontal MRI (hm306.tif) and segmented image (hs306.tif) on the initial feature of the software (left) and other corresponding horizontal MRI (hm326.tif) and segmented image (hs326.tif) (right).

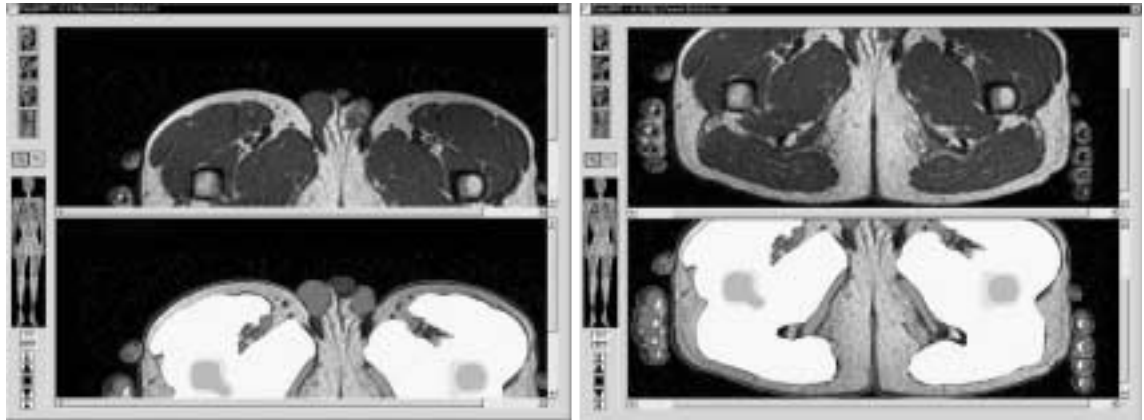


Fig. 12. Zoomed-in horizontal images (left) and moved horizontal images (right) on the software.

명영상과 구역화영상을 함께 볼 수 있었다(Fig. 12).  
 구역화영상의 해부구조에 마우스 포인터를 놓으면 이 해부구조의 이름을 풍선도움말에서 볼 수 있었다(Fig. 13).

수평영상을 관상영상으로 바꾸어서 1번 - 274번 관상영상의 가운데인 137번 관상영상을 볼 수 있었고, 수평영상을 시상영상으로 바꾸어서 1번 - 512번 시상영상의 가운데인 256번 시상영상, 즉 정중영상을 볼 수 있었다. 수평영상과 마찬가지로 관상영상과 시상영상을 골라서 볼 수 있었고 넘겨서 볼 수 있었고 확대해서 볼 수 있었고 옮겨서 볼 수 있었고 해부구조의 이름을 볼 수 있었다. 역시 관상영상과 시상영상을 볼 때마다 서로 들어맞는 자기공명영상과 구역화영상을 함께 볼 수 있었다(Fig. 14).

3차원영상을 손쉽게 보는 소프트웨어의 기능은 다음과 같았다.

초기화면에서 피부의 3차원영상을 앞에서 볼 수 있었다. 3차원영상에 그림자를 넣었기 때문에 입체감을 느낄 수 있었다(Fig. 15).

해부구조 목록에서 여러 해부구조의 3차원영상을 겹쳐서 골라 볼 수 있었고, 한 계통에 속한 모든 해부구조의 3차원영상을 한꺼번에 골라서 볼 수 있었다. 또한 일부 3차원영상을 반투명하게 만들어서 볼 수 있었다. 보기를 들어 피부의 3차원영상을 반투명하게 만들고, 다른 해부구조의 3차원영상을 불투명

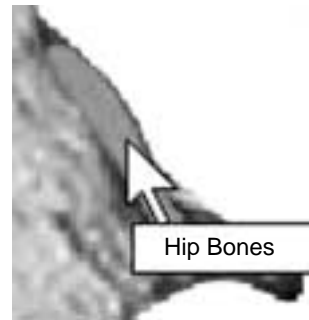


Fig. 13. Name of an anatomical structure displayed beside the mouse pointer on the software.

하게 만들어서 볼 수 있었다(Fig. 16).

마우스의 바퀴를 써서 3차원영상을 끝없이 (endlessly) 확대해서 볼 수 있었다. 마우스의 원단추와 오른단추를 함께 써서 확대한 3차원영상을 옮겨서 볼 수 있었다(Fig. 17). 거꾸로 3차원영상을 본디 배율로 축소해서 볼 수 있었다.

마우스의 원단추를 써서 해부구조의 3차원영상을 원오른쪽으로 또는 위아래로 끊임없이 (continuously) 돌려서 볼 수 있었다. 3차원영상을 원근방법으로 처리하였기 때문에 가까운 해부구조를 크게 보고, 먼 해부구조를 작게 볼 수 있었다(Fig. 18).

해부구조의 3차원영상에 마우스 포인터를 놓으면

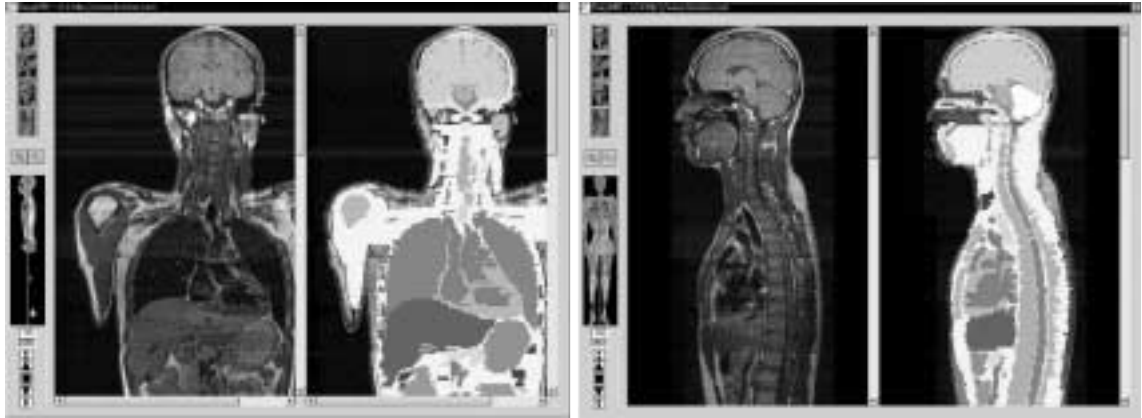


Fig. 14. Corresponding coronal MRI and segmented image (left) and corresponding sagittal MRI and segmented image (right) on the software.

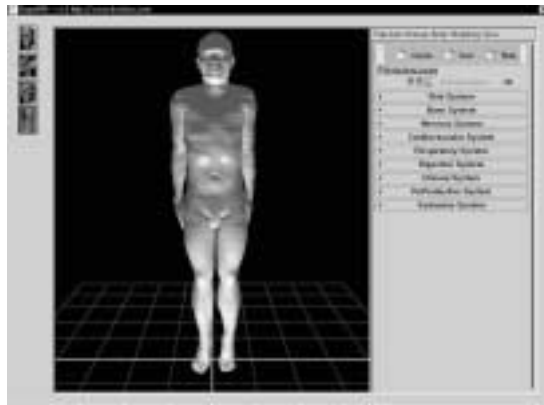


Fig. 15. 3D image of the skin on the initial feature of the software.

이 해부구조의 이름을 풍선도움말에서 볼 수 있었다. 반투명한 3차원영상과 속에 있는 불투명한 3차원영상이 겹친 곳에 마우스 포인터를 놓으면 불투명한 3차원영상의 해부구조 이름을 볼 수 있었다 (Fig. 19).

## 고 찰

자기공명영상의 해부구조를 익히는 데 도움이 되



Fig. 16. 3D image of the selected bones (left) and that of the semitransparent skin accompanied by the opaque liver, lungs, and meninges (right) on the software.

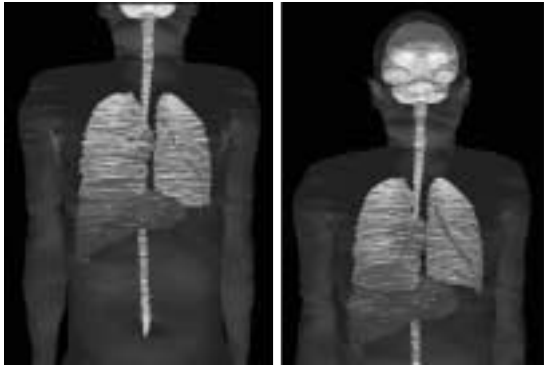


Fig. 17. Zoomed-in 3D image (left) and moved 3D image (right) on the software.

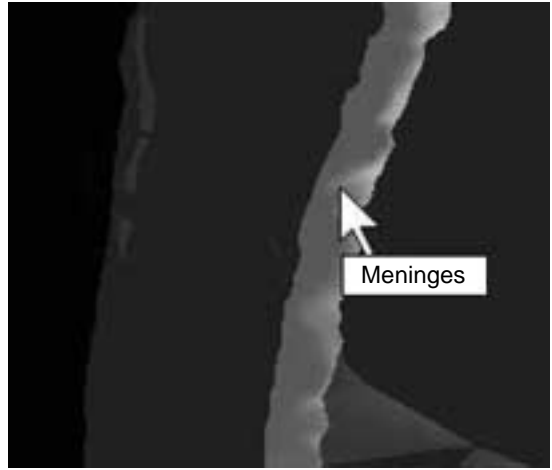


Fig. 19. Name of an anatomical structure displayed beside the mouse pointer on the software.

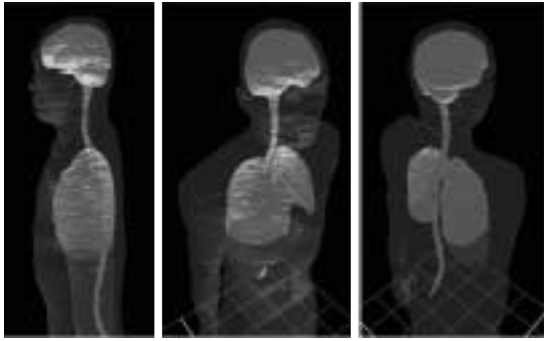


Fig. 18. Rotated 3D image at free angles on the software.

기 위해서는 다음과 같은 자기공명영상과 구역화영상을 마련해야 한다.

첫째로 수평, 관상, 시상 자기공명영상과 구역화영상의 방향과 번호를 짜임새 있게 결정해야 한다. 수평영상, 관상영상, 시상영상의 방향과 번호가 짜임새 있지 않으면 영상을 보면서 갈피를 잡기 어렵다. 따라서 이 연구에서는 영상의 방향과 번호를 다음처럼 결정하였다. 수평영상을 대상자의 아래에서 보게 하였고, 첫째 수평영상을 대상자의 위 끝으로 결정하였는데, 이것은 임상에서 쓰는 수평영상의 방향, 번호와 같았다(Fig. 11). 또한 관상영상을 앞에서 보게 하였고, 첫째 관상영상을 대상자의 앞 끝으로 결정하였으며, 시상영상을 왼쪽에서 보게 하였고, 첫째

시상영상을 대상자의 오른쪽 끝으로 결정하였다(Fig. 14). 이 결과로 수평영상의 가장 위 행이 첫째 관상영상이었으며, 가장 왼쪽 열이 첫째 시상영상이었다. 마찬가지로 관상영상의 가장 위 행이 첫째 수평영상이었으며, 가장 왼쪽 열이 첫째 시상영상이었으며, 시상영상의 가장 위 행이 첫째 수평영상이었으며, 가장 왼쪽 열이 첫째 관상영상이었다(Figs. 11, 14).

둘째로 구역화영상이 올바르고 자기공명영상에 들어맞아야 한다. 구역화영상이 올바르지 않거나 자기공명영상에 들어맞지 않으면 자기공명영상의 해부구조를 깨닫는 데 별로 도움이 되지 않는다. 따라서 이 연구에서는 올바른 구역화영상을 마련하기 위하여 의학전문가가 해부구조의 테두리를 수동으로 그려서 임시구역화영상을 만들었고, 관상구역화영상, 시상구역화영상, 테두리3차원영상을 살핀 다음에 틀린 임시구역화영상을 고쳤다(김진용 등 2003). 또한 자기공명영상에 들어맞는 구역화영상을 마련하기 위하여 자기공명영상 위에 해부구조의 테두리를 그려서 구역화영상을 만들었고(Fig. 1), 구역화영상의 해상도와 여백을 자기공명영상과 같게 하였고, 관상자기공명영상과 시상자기공명영상을 만드는 방법대로 관상구역화영상과 시상구역화영상을 만들었다(Table 1)(김진용 등 2003). 이 연구에서는 수동으

로 구역화하였는데, 앞으로는 반자동으로 구역화해서 구역화하는 시간과 노력을 줄이고, 더 객관적인 구역화영상을 만들 필요가 있다. 또한 이 연구에서는 대체로 중요한 해부구조 59개를 구역화하였는데 (Table 2), 앞으로는 관심 있는 해부구조를 더 낱낱이 구역화할 필요가 있다. 보기를 들어 뇌에 관심이 있으면 뇌를 회색질, 백색질, 뇌실 등으로 나누어서 구역화할 필요가 있다.

셋째로 자기공명영상과 구역화영상의 파일이 작아야 한다. 자기공명영상과 구역화영상의 파일이 크면 널리 퍼뜨리기 어렵고, 소프트웨어에서 자기공명영상과 구역화영상을 보는 데 시간이 오래 걸린다. 따라서 이 연구에서는 수평자기공명영상의 쓸모 없는 여백을 잘라내어서 해상도 (512×512)를 512×274로 줄였다. 그리고 수평, 관상, 시상 자기공명영상과 구역화영상을 대체로 파일이 작은 TIFF 파일로 저장하였다. 이 결과로 모든 자기공명영상과 구역화영상의 파일크기가 2,310 Mbytes였다 (Table 1) (김진용 등 2003). 이 연구에서는 구역화영상을 24 bits color로 저장하였는데, 구역화영상을 8 bits color로 저장하면 24 bits color로 저장하는 것에 비해서 파일크기가 1/3 즈음이다. 8 bits color의 빛깔 개수 (256개)는 구역화한 해부구조의 개수 (59개)보다 많기 때문에 8 bits color로 저장하면 구역화한 해부구조를 나타내는 데 문제가 없으나, 구역화한 해부구조의 사이에 있는 자기공명영상을 나타내는 데에는 문제가 있다. 따라서 이 연구에서는 구역화영상을 24 bits color로 저장하였다. 그러나 구역화한 해부구조의 사이를 검은 빛깔로 나타내는 경우에는 구역화영상을 8 bits color로 저장하는 것이 바람직하다.

자기공명영상의 해부구조를 익히는 데 도움이 되기 위해서는 다음과 같은 3차원영상을 마련해야 한다.

첫째로 자기공명영상에 들어맞는 3차원영상을 재구성해야 한다. 3차원영상이 자기공명영상에 들어맞지 않으면 자기공명영상의 해부구조를 깨닫는 데 별로 도움이 되지 않는다. 따라서 이 연구에서는 자기공명영상을 바탕으로 임시구역화영상을 만들었고, 이 임시구역화영상을 바탕으로 각 해부구조의 3차원영상을 재구성하였다. 이 때 3차원영상의 원래 위

치를 간직해서 여러 3차원영상을 함께 보면 해부구조의 위치 관계를 볼 수 있게 하였다 (Figs. 8, 16-18).

둘째로 surface rendering해서 3차원영상을 재구성해야 한다. Volume rendering해서 3차원영상을 재구성하면 3차원영상을 마음대로 잘라서 절단면을 볼 수 있다. 그러나 volume rendering하면 3차원영상의 파일이 매우 커서 3차원영상을 널리 퍼뜨리기 어렵고, 3차원영상을 실시간에 고르고 반투명하게 만들고 확대하고 옮기고 돌려서 볼 수 없다. 따라서 이 연구에서는 surface rendering해서 3차원영상을 재구성하였으며 (Fig. 3), 이 결과로 모든 3차원영상의 파일크기가 181 Mbytes뿐이었다 (Table 3). 이 연구에서는 표준 체형의 3차원영상을 마련하였는데, 이 3차원영상을 다른 체형의 3차원영상으로 변형할 값어치가 있다. 보기를 들어 한국사람의 체질인류학 자료를 바탕으로 뚱뚱한 체형의 3차원영상으로 변형할 값어치가 있다. 이 연구에서 마련한 3차원영상을 쉽게 변형할 수 있는데, 이것은 surface rendering해서 재구성한 3차원영상이기 때문이다. 그러나 앞으로는 3차원영상을 마음대로 잘라서 보기 위하여 자기공명영상 또는 구역화영상을 쌓은 다음에 volume rendering해서 3차원영상을 재구성할 필요도 있다.

셋째로 수동으로 surface rendering해서 3차원영상을 재구성하는 것이 바람직하다. 자동으로 surface rendering하면 3차원영상을 재구성하는 데 시간이 적게 걸리지만, 이를 위해서는 컴퓨터 전문가의 도움을 받아서 소프트웨어를 만들어야 한다. 따라서 이 연구에서는 3ds max 소프트웨어에서 수동으로 surface rendering하는 방법을 개발하였다 (Figs. 4-7). 이 방법을 쓰면 다른 연구자도 컴퓨터 전문가의 도움을 받지 않고 surface rendering해서 3차원영상을 재구성할 수 있다. 그러나 앞으로는 컴퓨터 전문가의 도움을 받아서 반자동으로 surface rendering하는 것이 바람직하다. 보기를 들어 자동으로 surface rendering하되, 해부구조가 갈라지거나 합쳐지는 곳에서는 수동으로 surface rendering하는 것이다. 이처럼 반자동으로 surface rendering을 하면 시간과 노력을 줄이는 데 도움이 되고, 구역화영상을 고치면

서 3차원영상을 재구성할 수 있기 때문에 올바른 3차원영상을 마련하는 데에도 도움이 된다.

넷째로 polygon 방법으로 surface rendering해서 3차원영상을 재구성하는 것이 바람직하다. 3ds max 소프트웨어에서 수동으로 surface rendering하는 방법에는 nurbs 방법, patch 방법, polygon 방법이 있다. Nurbs 방법 또는 patch 방법을 써서 surface rendering하면 부드러운 곡면을 만들어서 해부구조의 표면을 잘 나타낼 수 있지만, 알고리즘이 복잡해서 3차원영상의 파일이 비교적 크다(박정현 등 2000, 서승욱 2001). 따라서 이 연구에서는 polygon 방법으로 surface rendering하였다(Figs. 4-7).

다섯째로 등고선이 있는 3차원영상을 재구성해야 한다. 보기를 들어 어느 해부구조의 표면을 3차원스캐너로 컴퓨터에 입력하면 등고선이 없는 3차원영상을 마련할 수 있으며, 이 3차원영상은 사각형 또는 삼각형의 개수가 적어도 해부구조의 표면을 잘 나타낸다. 그러나 이 연구에서는 테두리3차원영상을 바탕으로 surface rendering하였기 때문에 테두리가 남아 있는, 즉 등고선이 있는 3차원영상을 마련하였다(Fig. 3). 필요하면 이 연구에서 만든 3차원영상을 등고선이 없는 3차원영상으로 바꿀 수 있다.

여섯째로 테두리 사이에 사각형과 삼각형을 채운 3차원영상을 재구성해야 한다. 테두리 사이에 사각형만 채우면 모든 테두리는 꼭지점의 개수가 한결 같은 다각형으로 바뀐다. 그런데 길고 복잡한 테두리도, 짧고 단순한 테두리도 꼭지점의 개수가 한결 같은 다각형으로 바뀐다는 것은 모순이다. 따라서 이 연구에서는 테두리 사이에 사각형을 채운 다음에 테두리가 크게 다르면 테두리 사이에 삼각형을 채워서 다각형의 꼭지점을 늘리거나 줄였다(Figs. 4, 5).

일곱째로 사각형과 삼각형의 개수가 알맞은 3차원영상을 재구성해야 한다. 사각형과 삼각형이 너무 적으면 해부구조의 본디 모습을 간직하지 못하는 문제가 있고, 너무 많으면 3차원영상을 재구성하고 고치는 데 시간이 오래 걸리고, 3차원영상의 파일이 커지는 문제가 있다. 따라서 이 연구에서는 닫힌곡선인 첫째 테두리를 다각형으로 바꿀 때 테두리의 본디 모습을 거의 간직하는 범위에서 다각형의 꼭

지점을 되도록 적게 만들었다(Fig. 4). 또한 테두리 사이에 사각형과 삼각형을 채울 때에도 테두리의 본디 모습을 거의 간직하는 범위에서 사각형과 삼각형을 되도록 적게 만들었다(Figs. 4-7). 이 결과로 머리덜개의 피부처럼 단순한 해부구조에는 사각형과 삼각형이 적게 만들어졌고, 컷바퀴의 피부처럼 복잡한 해부구조에는 사각형과 삼각형이 많이 만들어졌다(Fig. 3).

여덟째로 되도록 손쉽게 3차원영상을 재구성해야 한다. 3차원영상을 재구성하는 것이 손쉽지 않으면 수동으로 하는 것이기 때문에 시간이 오래 걸린다. 따라서 이 연구에서는 다음처럼 손쉽게 3차원영상을 재구성하였다. 모눈선 사이의 간격을 한꺼번에 바꾸었다. 테두리영상을 모눈선의 언저리에 놓으면 자동으로 모눈선에 들어맞게 하였다. 사각형과 삼각형에 빗깔을 칠하지 않은 채로, 즉 사각형과 삼각형의 면을 투명하게 만들고 꼭지점과 모서리만 보이게 한 채로, 일하였다. 테두리를 다음 모눈선까지 밀 때 모눈선 사이의 간격을 입력하였다. 수직적사각형의 아래 꼭지점을 테두리로 끌 때 x축과 y축으로만 끌어서 아래 꼭지점이 모눈선에서 벗어나지 않게 하였다(Figs. 4, 6, 7). 한 해부구조의 3차원영상을 이루는 모든 사각형과 삼각형에 특정한 빗깔을 한꺼번에 칠했다(Fig. 3).

아홉째로 재구성한 3차원영상을 MAX 파일로 저장해야 한다. 3차원영상을 MAX 파일로 저장하지 않으면 누구나 쉽게 열어 볼 수 없다. 따라서 이 연구에서는 3차원영상을 MAX 파일로 저장하였다. MAX 파일로 저장한 3차원영상을 널리 퍼뜨리면 누구나 쉽게 3ds max 소프트웨어에서 열어 볼 수 있고 가공할 수 있다. 앞으로는 반자동으로 surface rendering한 경우에도 3차원영상을 MAX 파일로 저장할 필요가 있다.

열째로 3차원영상을 원근방법으로 처리하는 것이 바람직하다. 이 연구에서는 3차원영상을 원근방법으로 처리해서 가까운 해부구조를 크게 나타냈고, 먼 해부구조를 작게 나타냈으며, 이 결과로 3차원영상을 돌려 보면 실감났다(Fig. 18). 그러나 3차원영상을 원근방법으로 처리하지 않는 것이 좋은 경우도 있다. 보기를 들어 3차원영상에 포함되어 있는 자기공명

영상을 볼 경우에는 가까운 3차원영상에 쏠려 있는 자기공명영상의 크기와 먼 3차원영상에 쏠려 있는 자기공명영상의 크기가 같은 것이 좋다(Fig. 20).

열번째로 3차원영상에 그림자를 넣어야 한다. 3차원영상에 그림자를 넣지 않으면 입체감이 없기 때문에 3차원영상을 실감나게 볼 수 없다. 따라서 이 연구에서는 광원이 알맞은 곳에서 빛을 비추는 것으로 가정하고 3차원영상에 그림자를 넣었다(Figs. 8, 10, 15-19).

자기공명영상, 구역화영상, 3차원영상을 손쉽게 보는 소프트웨어를 마련해야 한다. 소프트웨어를 마련하지 않으면 자기공명영상과 구역화영상을 ACDSSee 소프트웨어(5.0판, ACD Systems™)와 같은 영상 소프트웨어에서 보아야 하고, 3차원영상을 3ds max 소프트웨어에서 보아야 한다. 그런데 ACDSSee 소프트웨어에서 자기공명영상과 구역화영상을 함께 고르고 넘기고 확대하고 옮겨서 보는 것(Figs. 11, 12, 14)이 매우 불편하고, 3ds max 소프트웨어에서 3차원영상을 고르고 반투명하게 만들고 확대하고 옮기고 돌려서 보는 것(Figs. 16-18)도 매우 불편하다. 또한 ACDSSee 소프트웨어와 3ds max 소프트웨어에서 새로운 기능, 보기를 들면 해부구조의 이름을 보는 기능(Figs. 13, 19)을 덧붙일 수 없다. 따라서 이 연구에서는 영상을 손쉽게 보는 소프트웨어를 마련하였다.

자기공명영상의 해부구조를 익히는 데 도움이 되기 위해서는 다음과 같은 소프트웨어를 마련해야 한다.

첫째로 자기공명영상을 손쉽게 골라야 한다. 자기공명영상을 손쉽게 고르지 못하면 관심 있는 자기공명영상을 보는 데 시간이 오래 걸린다. 따라서 이 연구에서는 다음처럼 하였다. 소프트웨어의 초기화면에서 가장 많이 쓰는 수평영상이 나타나게 하였고, 수평영상을 관상영상 또는 시상영상으로 바꾸게 하였다. 이 때 각 영상의 가운데인 306번 수평영상, 137번 관상영상, 256번 시상영상이 나타나게 하였다(Figs. 11, 14). 영상의 번호를 입력해서 영상을 고르게 하였고, 아울러 이웃한 영상으로 넘기면서 영상을 고르게 하였다(Fig. 11). 앞으로는 수평영상의 어느 한 곳에 마우스 포인터를 놓고 눌러서 관상영상

과 시상영상을 고르게 할 필요가 있으며, 마찬가지로 관상영상에서 수평영상과 시상영상을 고르게 하고, 시상영상에서 수평영상과 관상영상을 고르게 할 필요가 있다. 또한 해부구조 목록에서 영상을 고르게 할 필요가 있는데, 보기를 들어 해부구조 목록에서 심장을 고르면 심장을 담은 128번 - 172번 수평영상을 잇달아 나타나게 할 필요가 있다.

둘째로 자기공명영상을 잇달아 넘겨야 한다. 자기공명영상을 잇달아 넘기지 못하면 자기공명영상에서 입체해부구조가 어떻게 보이는지 깨닫기 어렵다. 따라서 이 연구에서는 자기공명영상을 잇달아 넘겨서 영화처럼 보게 하였고, 잇달아 넘기는 것과 멈추는 것을 마음대로 되풀이하게 하였다(Fig. 11).

셋째로 자기공명영상을 확대해야 한다. 자기공명영상을 확대하지 못하면 관심 있는 해부구조를 낱알이 볼 수 없다. 또한 관상자기공명영상과 시상자기공명영상을 확대하지 못하면 제대로 볼 수 없는데, 이것은 관상자기공명영상, 시상자기공명영상의 세로 해상도(1,839)가 보통 모니터의 세로 해상도(768)보다 훨씬 크기 때문이다(Table 1). 따라서 이 연구에서는 자기공명영상을 16배까지 확대하게 하였고, 두루마리막대를 써서 확대한 자기공명영상을 옮기게 하였다(Fig. 12). 앞으로는 확대한 자기공명영상을 마우스 포인터로 잡아 끌어서 옮기게 할 필요가 있다.

넷째로 들어맞는 자기공명영상과 구역화영상을 함께 봐야 한다. 구역화영상을 함께 보지 못하면 자기공명영상의 해부구조를 익히기 어렵다(정민석 등 2000, Chung과 Kim 2000). 따라서 이 연구에서는 자기공명영상을 고르고 넘기고 확대하고 옮겨서 볼 때마다 들어맞는 구역화영상을 함께 보게 하였다(Figs. 11, 12, 14). 보기를 들어 자기공명영상을 2배 확대하면 구역화영상도 자동으로 2배 확대되었다. 이 연구에서는 자기공명영상과 구역화영상을 나란히 늘어놓고 보게 하였는데, 앞으로는 모니터의 넓이를 아끼기 위하여 자기공명영상만 보게 하다가 어느 단추를 누르면 자기공명영상을 구역화영상으로 바꾸어서 보게 할 필요도 있다.

다섯째로 구역화영상에서 보이는 해부구조가 무엇인지 쉽게 알아야 한다. 구역화영상에서 보이는

해부구조가 무엇인지 알 수 없으면 이 구역화영상은 도움이 되지 않는다. 따라서 이 연구에서는 구역화영상을 만들 때 각 해부구조의 테두리 속에 특정한 빛깔을 채웠고, 소프트웨어를 만들 때 구역화영상의 해부구조에 마우스 포인터를 놓으면 이 해부구조의 이름이 나타나게 하였다(Fig. 13). 앞으로는 자기공명영상에 마우스 포인터를 놓아도 해부구조의 이름이 나타나게 할 필요가 있다. 또한 앞으로는 자기공명영상 하나를 보면 이 자기공명영상에 담긴 모든 해부구조가 해부구조 목록에 나타나게 하고, 자기공명영상을 넘기면 해부구조 목록이 자동으로 바뀌게 할 필요가 있다.

여섯째로 3차원영상을 겹쳐서 골라야 한다. 여러 해부구조의 3차원영상을 겹쳐서 고르지 못하면 관심 있는 해부구조의 위치 관계를 깨달을 수 없다. 따라서 이 연구에서는 해부구조 목록에서 여러 해부구조의 3차원영상을 겹쳐서 고르게 하였다(Fig. 16). 보기를 들어 뼈와 콩팥의 3차원영상을 겹쳐서 고르면 콩팥의 위치를 쉽게 깨달을 수 있었다. 또한 한 계통에 속한 모든 해부구조의 3차원영상을 한꺼번에 고르게 하였다. 보기를 들어 해부구조 목록에서 비뇨계통을 고르면 콩팥, 요관, 방광의 3차원영상이 한꺼번에 나타났다.

일곱째로 3차원영상을 반투명하게 만들어야 한다. 3차원영상을 반투명하게 만들지 못하면 속에 있거나 뒤에 있는 다른 3차원영상을 볼 수 없기 때문에 해부구조의 위치 관계를 깨닫기 어렵다. 따라서 이 연구에서는 일부 3차원영상을 반투명하게 만들게 하였다. 보기를 들어 피부의 3차원영상을 반투명하게 만들고, 허파의 3차원영상을 불투명하게 만들면 허파의 위치를 쉽게 깨달을 수 있었다(Figs. 16-18).

여덟째로 3차원영상을 확대하고 돌려야 한다. 3차원영상을 확대하지 못하면 관심 있는 해부구조를 낱알이 볼 수 없다. 따라서 이 연구에서는 3차원영상을 끝없이 확대하게 하였고, 확대한 3차원영상을 옮기게 하였다(Fig. 17). 한편 3차원영상을 돌리지 못하면 해부구조를 여러 방향에서 볼 수 없다. 따라서 이 연구에서는 3차원영상을 원오른쪽으로 또는 위아래로 끊임없이 돌리게 하였으며, 이 결과로 3차

원영상을 모든 방향에서 볼 수 있었다(Fig. 18). 이 연구에서는 마우스의 바퀴와 단추를 써서 3차원영상을 확대하고 돌리게 하였는데, 앞으로는 차림포에서 확대율과 돌리는 각도를 입력해서 확대하고 돌리게 할 필요도 있다.

아홉째로 각 3차원영상이 무엇인지 쉽게 알아야 한다. 각 3차원영상이 어느 해부구조인지 알지 못하면 이 3차원영상은 도움이 되지 않는다. 따라서 이 연구에서는 다음처럼 하였다. 해부구조 목록에서 해부구조를 클릭하면 3차원영상이 나타나거나 나타나지 않게 하였다. 보기를 들어서 해부구조 목록에서 허파를 잇달아 클릭하면 허파의 3차원영상이 나타나고 나타나지 않는 것을 되풀이하게 하였다. 또한 3차원영상에 마우스 포인터를 놓으면 해부구조의 이름이 나타나게 하였다. 이 때 반투명한 3차원영상과 속에 있는 불투명한 3차원영상이 겹친 곳에 마우스 포인터를 놓으면 불투명한 3차원영상의 해부구조 이름이 나타나게 하였는데, 이렇게 한 것은 대개 불투명한 3차원영상에 관심 있기 때문이었다(Fig. 19). 앞으로는 구역화영상의 빛깔과 3차원영상의 빛깔을 같게 만들어서 각 해부구조가 구역화영상과 3차원영상에서 어떻게 보이는지 쉽게 깨닫게 할 필요가 있다.

열째로 자기공명영상과 3차원영상을 함께 봐야 한다. 자기공명영상과 3차원영상을 따로 보면 자기공명영상에서 입체 해부구조가 어떻게 보이는지 깨닫기 어렵다. 따라서 이 연구에서는 자기공명영상, 구역화영상을 보는 소프트웨어와 3차원영상을 보는 소프트웨어를 함께 실행해서 두 모니터(dual monitor)에서 보는 것을 권하였다. 그런데 두 소프트웨어를 함께 실행하는 것은 불편하기 때문에 앞으로는 통합 소프트웨어를 만들 필요가 있다. 보기를 들어 해부구조 목록에서 심장을 클릭하면 심장을 담은 자기공명영상과 심장의 3차원영상이 함께 나타나는 소프트웨어를 만들 필요가 있다. 나아가 자기공명영상이 심장의 3차원영상에 꼭혀 있는 모습으로 나타나는 통합 소프트웨어를 만들 필요가 있다. 자기공명영상이 3차원영상에 꼭혀 있게 하려면 자기공명영상과 3차원영상의 정합(registration)을 간직해야 하고, 비트맵(bitmap) 영상인 자기공명영상과 벡터

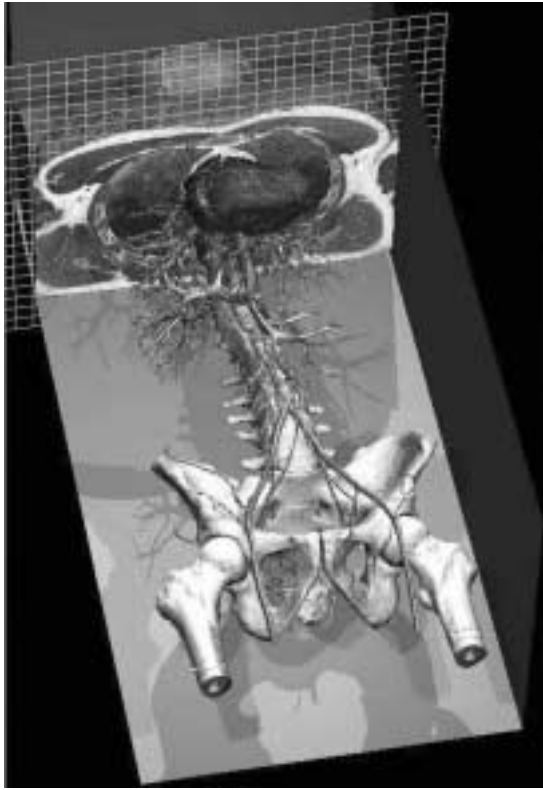


Fig. 20. Combined three-dimensional and two-dimensional images on the VOXEL-MAN software.

(vector) 영상인 3차원영상을 엉겨 붙게 하는 기술이 필요하다 (Fig. 20) (Pommert 등 2001, Schiemann 등 2002).

열한째로 편리한 사용자사이드를 써서 영상을 봐야 한다. 편리한 사용자사이드를 쓰지 못하면 관심 있는 영상을 골라서 보는 데 시간이 오래 걸린다. 따라서 이 연구에서는 편리한 사용자사이드를 다음처럼 만들었다. 그림이 담긴 단추와 체계적인 해부구조 목록을 쓰게 하였고, 마우스의 왼단추, 오른단추, 바퀴를 알맞게 쓰게 하였다. 앞으로 더 손쉽고 한결같고 예쁜 사용자사이드를 개발할 필요가 있다. 보기를 들어 마우스의 오른단추를 클릭하면 알맞은 불쑥차림표가 나타나게 하고, 사용자사이드의 단추에 마우스 포인터를 놓으면 이 단추의 풀이가 풍선

다음말에 나타나게 하는 것이다.

이 연구에서 마련한 영상과 소프트웨어를 널리 퍼뜨려야 한다. 이 연구에서 마련한 영상과 소프트웨어를 널리 퍼뜨리지 않으면 의과대학 학생과 의사가 교육 자료로 쓸 수 없고, 다른 연구자가 더 좋은 구역화영상, 3차원영상, 소프트웨어를 만들 수 없다. 따라서 이 연구에서 만든 영상과 소프트웨어를 오프라인으로 또는 온라인으로 널리 퍼뜨릴 계획이다. 모든 자기공명영상과 구역화영상의 파일크기가 2,310 Mbytes (압축파일크기 447 Mbytes)였고 (Table 1), 3차원영상의 파일크기가 181 Mbytes (압축파일크기 12 Mbytes)였고 (Table 3), 소프트웨어의 파일크기가 41 Mbytes (압축파일크기 2 Mbytes)였기 때문에 모든 영상과 소프트웨어의 파일크기는 2,351 Mbytes (압축파일크기 393 Mbytes)였다. 따라서 압축파일을 CD title 1개에 담아서 오프라인으로 쉽게 퍼뜨릴 수 있고, 인터넷을 통해서 온라인으로 쉽게 퍼뜨릴 수 있다. 다른 연구자가 바라는 경우에는 중간 결과물인 임시구역화영상, 테두리영상, 테두리3차원영상도 퍼뜨릴 필요가 있다.

이 연구에서 마련한 소프트웨어가 있으면 서로 들어맞는 자기공명영상과 구역화영상을 손쉽게 볼 수 있고, 자기공명영상에 담긴 해부구조의 3차원영상을 손쉽게 볼 수 있기 때문에 자기공명영상에서 입체 해부구조가 어떻게 보이는지 깨닫는 데 도움이 되며, 나아가 환자의 자기공명영상에서 병리구조를 판독하는 데 도움이 될 것이다. 따라서 이 연구에서 마련한 소프트웨어는 의과대학 학생과 의사가 자기공명영상의 해부구조를 익히는 데 쓸 것으로 기대된다. 또한 이 연구에서 개발한 수동 surface rendering 방법은 다른 연구자가 컴퓨터 전문가의 도움 없이 3차원영상을 재구성하는 데 쓸 것으로 기대된다.

## 참고문헌

- 김진용, 정민석, 이용숙, 박진서, 조재현, 최우영 : 한국사람 온몸의 수평, 관상, 시상 자기공명영상과 구역화영상. 대한체질인류학회지 16 : 1-13, 2003.
- 박정현, 황혜현 : 3ds max 4.x & animation 고급활용. 정보문화사, pp. 372-413, 2000.

서승욱 : 3dsMAX 4.2 & Hint. 성안당, p. 405, 2001.

정민석, 조우정, 김이석, 조재현, 최우영 : 여러 방향으로 촬영한 머리, 목의 자기공명영상을 익힐 수 있는 컴퓨터 소프트웨어. 대한체질인류학회지 13 : 263-270, 2000.

Chung MS, Kim SY: Three-dimensional image and virtual dissection program of the brain made of Korean cadaver. Yonsei Med J 41 : 299-303, 2000.

Pommert A, Hohne KH, Pflesser B, Richter E, Riemer M, Schiemann T, Schubert R, Schumacher U, Tiede U :

Creating a high-resolution spatial/symbolic model of the inner organs based on the Visible Human. Med Image Anal 5 : 221-228, 2001.

Schiemann T, Freudenberg J, Pflesser B, Pommert A, Priesmeyer K, Riemer M, Schubert R, Tiede U, Hohne KH: Exploring the Visible Human using the VOXEL-MAN framework. Comput Med Imaging Graph 24 : 127-132, 2002.

K C I

**Abstract**

## **Three-Dimensional Images and Software for Studying Anatomical Structures in MRIs**

**Yong Sook Lee, Min Suk Chung, Sung Bae Hwang,  
Gi-Nam Wang<sup>1</sup>, Seung-Hyun Yoo<sup>2</sup>**

*Department of Anatomy, Ajou University School of Medicine, Suwon, Korea*

<sup>1</sup>*Department of Industrial & Information System Engineering, Ajou University, Suwon, Korea*

<sup>2</sup>*Department of Mechanical Engineering, Ajou University, Suwon, Korea*

Medical students and doctors should study the appearance of normal anatomical structures in the magnetic resonance images (MRIs). For this purpose, horizontal, coronal, sagittal MRIs of a healthy entire body and corresponding segmented images were prepared. However, it is inconvenient to select interesting MRIs and segmented images; and it is difficult to understand how stereoscopic anatomical structures appear in the MRIs. Therefore, in this research, a software, on which interesting MRIs can be conveniently displayed among the horizontal, coronal, sagittal MRIs and corresponding segmented images can be displayed together, was composed. And after stacking the segmented images, three-dimensional (3D) image of each anatomical structure was manually reconstructed by surface rendering. Then another software, on which interesting 3D images of anatomical structures can be displayed and rotated, was composed too. These softwares are expected to help medical students and doctors understand normal anatomical structures in the MRIs and read pathological findings in the MRIs of patients.

**Key words** : MRIs, Anatomical structures, Segmented images, Surface rendering, 3D images, Software