



# MRI Data 가시화용 후처리 프로그램 개발

명현국,<sup>\*1</sup> 최훈희<sup>2</sup>

## DEVELOPMENT OF A POST-PROCESSING PROGRAM FOR VISUALIZATION OF MRI DATA

H.K. Myong<sup>\*1</sup> and H.H. Choi<sup>2</sup>

*A post-processing program based on the OOP(Object-Oriented Programming) concept has been developed for visualization of MRI. User-friendly GUI(Graphic User Interface) has been built on the base of MFC(Microsoft Foundation Class). The program is organized as modules by classes based on VTK-library, and these classes are made to function through inheritance and cooperation which are an important and valuable concept of object-oriented programming. The major functions of this post-processor program are introduced and demonstrated, which include contour plot, surface plots, cut plot and clip plot as well as view manipulation (translation, rotation, scaling etc).*

**Key Words:** Post-Processing(후처리), Visualization(가시화), OOP(Object-Oriented Programming: 객체지향프로그래밍), VTK(Visualization ToolKit), Class(클래스), GUI(Graphic User Interface; 그래픽 사용자 인터페이스), MRI(Magnetic Resonance Image)

### 1. 서 론

1895년 뢰트겐이 X-선을 발견하여 의학의 새로운 시대가 열리고 CT(Computer Tomography) 와 1973년 개발된 MRI(Magnetic Resonance Image)는 현대 물리학으로 현대 의학의 새로운 지평을 열었다. MRI 영상은 자기 공명영상기법을 이용하여 신체 부위의 다량의 단면 사진들을 병합하여 입체적으로 나타내는 것으로, 이전의 CT는 횡단면만 촬영하는 것이 가능했지만 MRI는 종, 횡단면 모두 촬영이 가능하기 때문에 뼈, 근육, 연골, 인대, 혈관처럼 수분이 많은 곳이나 미세한 부분도 선명하게 찍어 낼 수 있다. 이러한 영상들을 가시적으로 확인할 수 있는 입체영상과 여러 기능을 표현하기 위해서는 컴퓨터 그래픽으로 처리해주는 후처리(post-processing) 프로그램이 필요하다. 이 후처리 프로그램은 영상판독, 분석이 용이하도록 선명하고 깨끗한 화질 뿐만 아니라 사용자의 필요에 의한 커팅(cutting)과 슬라이싱

(slicing) 등 각종 기능들이 포함되어 있어야 한다.

그러나 이러한 MRI 영상처리를 하는 후처리 프로그램은 국내에서 상대적으로 연구가 많이 진행되지 않은 실정으로, 주로 고가의 외국 상용 소프트웨어를 구입하여 사용하고 있다. 그러나 기존의 외국 상용 소프트웨어는 고가의 도입비용과 유지비용이 불가피하여 예산문제를 가중시키고 많은 설정과 기능들로 사용법을 익히기도 어려운 문제를 가지고 있다.

저자는 이러한 문제를 해결하고자 최근 자체 개발한 유동 해석용 코드(PowerCFD)의 후처리 장치[1]에 사용된 모듈(module)에 새로운 모듈을 추가하여 의료분야에서 널리 사용되는 MRI 데이터를 3차원 가시화하는 후처리 프로그램(PowerCFD/PostPv1.0)을 개발하였다. 이 후처리 프로그램은 C++ 언어로 짜여 있으며, 객체지향프로그래밍(OOP; Objected-Oriented Programming) 기법으로 구현된 3차원 그래픽 라이브러리인 VTK(Visualization Toolkit)[2]를 사용함으로써, 보다 강력하고 편리한 기능구현이 가능하고 모듈화를 이용한 단순한 구조의 프로그램을 통해 변경이나 확장에 유연하다. 또한, 3차원 가시화에 있어서 입력된 데이터(Siemens Medical Systems, Inc에서 제공한 109-slice MRI data)를 이미지로 표현하는 단순 그래픽 기능뿐만 아니라, 입력된 데이터를 필터

1 정회원, 국민대학교 기계자동차공학부

2 학부생, 국민대학교 기계자동차공학부

\* Corresponding author, E-mail: myong@kookmin.ac.kr

개념의 독립된 모듈로 구성함으로써 보다 자유로운 가시화 작업을 가능하게 하였다. 현재까지 개발된 주요 기능(특징)으로는 Cut Plot, Clip Plot, Contour Plot, Slice Bar 기능 등이 있다.

본 논문에서는 OOP 개념에 기초하여 개발된 후처리 프로그램(PowerCFD/PostP v1.0)의 MRI 처리 모듈의 구조, 주요 기능 및 적용 예를 소개한다.

### 2. VTK 소개

일반적으로 3차원 그래픽을 컴퓨터로 구현하는 것은 용이한 작업이 아니다. 즉, 많은 사람들이 OpenGL 등을 공부하고 이를 이용해 3차원 그래픽을 구현하지만, OpenGL은 3차원 데이터를 모니터와 같은 하드웨어에 그려주는 기본적인 그래픽 라이브러리이기 때문에 거의 모든 것을 프로그래머가 작성해야 한다. 예를 들어 자동차의 충돌 시뮬레이션을 수행하고, 그 결과를 등고선 형태로 표현한다고 할 경우, 프로그래머는 등고선으로 출력하기 위해서 자동차를 구성하는 데이터에서 등고선으로 출력하는 알고리즘을 알아야 하고, 이것을 직접 프로그램으로 작성해야 한다. 또한 화면에 3차원 물체를 출력했는데 이를 확대하고 이동시키는 등의 연산을 하고 싶다면, 이것 역시 프로그래머가 직접 구현해야 한다. 즉, OpenGL을 직접 사용하는 것은 Windows 프로그래밍을 MFC(Microsoft Foundation Class)를 사용하지 않고 SDK를 이용하여 프로그램을 작성하는 것과 비슷하다.

본 연구에서 사용한 VTK[2]는 변수와 함수를 결합한 클래스(class)와 클래스간의 파생과 상속(inheritance) 등의 개념을 제공하므로, 효율적이고 강력한 후처리 프로그램 개발이 가능한 객체지향 기법으로 설계·구현된 3차원 그래픽 라이브러리이다. 또한 복잡한 후처리 기능들에 관한 대부분의 알고리즘은 이미 구현되어 모듈로 만들어져 있기 때문에, 유연한 확장성과 함께 비교적 쉽게 사용할 수 있는 장점이 있다. 즉, 컴퓨터 그래픽과 가시화(visualization)에 대한 기본개념을 알고 있으면, 누구나 비교적 용이하게 프로그래밍이 가능하다.

### 3. 프로그램의 구조

#### 3.1 개발 목표

본 프로그램의 개발 목표는 다음과 같다.

첫째, 특화된 MRI 가시화용 프로그램 개발로, 프로그램 사용방법을 익히는데 들어가는 시간을 최소한으로 줄여주기 위해 외국 상용 소프트웨어들의 가시화 기능들 중에서 많이 사용되면서도 중요한 기능을 구현하여 데이터 가시화 및 분석용으로 활용하는 것이다.

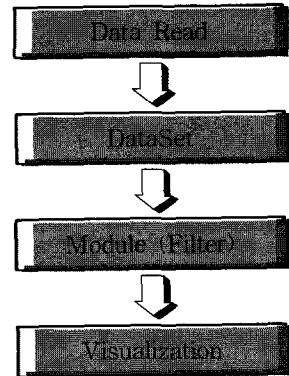


Fig. 1 Flow chart of visualization program

둘째, 사용자에게 친숙한 환경을 제공해줄 수 있는 편리한 GUI(Graphic User Interface) 환경을 구축하는 것으로, 이것은 사용자들을 확보하고 프로그램을 성장시킬 수 있는 가장 기본적인면서 중요한 사항이다.

셋째, 확장성 및 관리의 용이성을 위해 OOP 개념에 기초하여 프로그램 구조를 설계하는 것이다.

#### 3.2 프로그램의 순서도

가시화 프로그램이 실행되는 기본절차를 간단하게 표현하면 Fig. 1과 같다. 즉, 데이터를 읽은 후 메모리를 동적으로 생성하여 데이터를 저장할 수 있는 클래스 객체에 데이터를 저장하고, 저장된 데이터는 필터라는 독립된 모듈에 의해 가공되어 화면에 출력되는 시스템이다.

#### 3.3 클래스

Fig. 2는 클래스 협력관계를 도표를 나타낸 것으로, 매우 많은 수의 VTK 클래스가 사용되므로 vtkDataSet을 제외하고는 그림에 나타내지 않았다. 여기서 각각의 박스들은 클래스를 의미하며, 실선은 상속관계를, 점선은 협력(collaboration)관계를 각각 나타내고 있다.

#### 3.4 Graphics Model

본 프로그램에서 사용하는 Graphics Model은 VTK에서 제공하는 클래스를 이용하고 있다. 이것은 OpenGL의 렌더링 라이브러리 개념보다 높은 레벨로, 이해하고 사용하기 쉬운 장점이 있다. Windows와 같은 GUI 환경에 기반을 둔 8개의 기본 객체(basic objects)는 다음과 같다.

1. Render Window : 장치에 보여 지는 윈도우를 관리한다. 물체를 표현하기 위해 하나의 윈도우에 여러 개의 렌더러가 적용되기도 한다.
2. Renderer : 렌더링되는 Lights, Cameras, Actors를 표현한다.

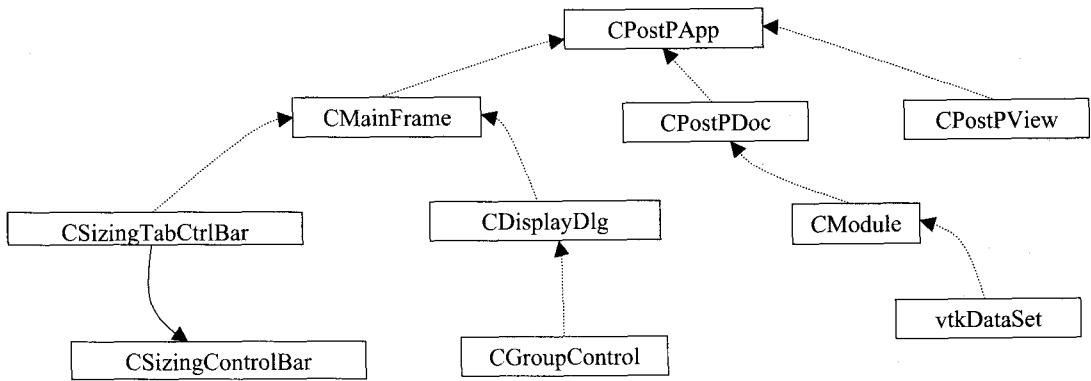


Fig. 2 Class inheritance and collaboration diagram

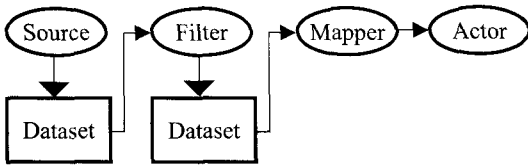


Fig. 3 Imaging model pipeline

3. Light : Actor가 보이도록 빛을 받는다.
4. Camera : 시점, 초점, 카메라의 특징 등을 계산한다.
5. Actor : Renderer에 의해 구현되는 모든 것.
6. Property : 색상, 빛, 질감, 스타일처럼 렌더링시 Actor로부터 표현되는 속성.
7. Mapper : Lookup Table을 통해 매핑(mapping)된 물체를 표현한다. 하나 이상의 물체가 같은 Mapper를 사용할 수 있다.
8. Transform : Actors, Cameras, Lights의 위치와 방향 값을 가진다.

### 3.5 Imaging Model

Reader로부터 얻어진 데이터는 구조화된 Dataset으로 생성되어 특수한 기능의 필터를 거쳐 하나의 기능을 수행하는 Module로 생성된다. Dataset을 대상으로 하는 처리 과정은 항상 input과 output의 Dataset 객체를 갖는데 필터의 결과물인 Output Dataset은 Mapper를 거쳐 Actor로 가시화된다. 참고로 Fig. 3은 Imaging model pipeline을 나타낸 것이다.

## 4. 프로그램의 구조

### 4.1 Dataset

Dataset은 기하학(geometry)과 위상(topology) 정의를 거쳐 점(point), 셀 속성데이터(cell attribute data) 생성과 연결로 이

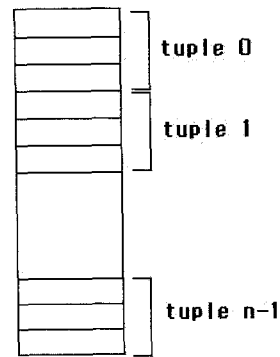


Fig. 4 Tuple, data array structure

루어진다. 처리하는 Dataset의 종류에 따라 기하학과 위상 정의는 다르지만, 속성 데이터의 지정법은 모두 동일하다. 즉, 속성 데이터는 상대적으로 간단한 자료구조인 배열을 사용하는데 Dataset을 구현할 때 스칼라(scalar), 벡터(vector), 텐서(tensor) 등의 속성들과 각각의 필드(field) 값을 하나로 묶어 주기 위하여 Fig. 4와 같은 튜플(배열의 배열)을 이용한다.

또한 속성이 연결될 기하학과 위상은 다수의 이미지를 표현하게 되므로 이미지 데이터(image data)로 처리하게 되는데, 본 코드에서 채용한 이미지 데이터는 가장 간단한 형태의 Dataset으로 x, y, z축과 평행한 등간격으로 이루어진 격자로 표현된다. 즉, 기하학이 간격과 원점을 설정하고, 위상이 각각의 축으로의 간격(spacing)을 설정하여 궁극적으로 셀과 점을 생성시켜 지정한다.

### 4.2 Filter

#### Contour

Surface Contour를 구현하기 위해 사용된 알고리즘은 Contour를 그리기 위한 방법으로 많이 사용되고 있는 Marching Square 알고리즘이다[2]. 이 알고리즘의 원리는 사

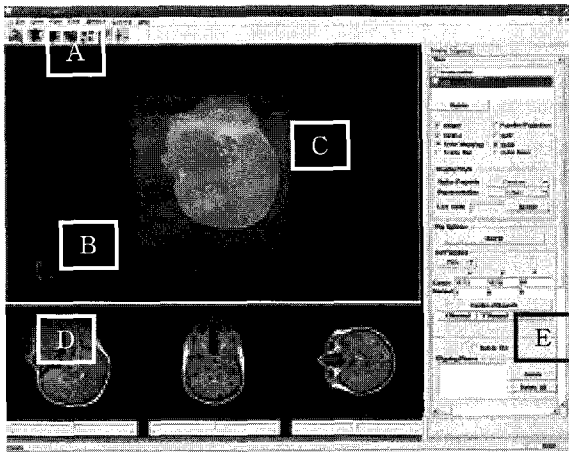


Fig. 5 The structure of GUI

변형 형태의 모든 셀에 대하여 각 셀의 변에서 동일한 값을 꼭짓점으로부터 보간하여 찾아내고 선으로 이어 주는 것이다. 하나의 셀에서 Contour Line을 만들 수 있는 경우는 16가지이지만 대칭성을 고려하면 네 가지로 줄일 수 있다[2].

Iso-Surface를 구현하기 위한 알고리즘은 널리 사용되고 있는 Marching Cubes 방법이다[2]. 이 방법은 삼각형 조각을 이어 붙여 면을 만드는 방법으로써 육면체의 꼭짓점의 선택여부에 따라 256가지의 경우의 수가 생길 수 있지만, Contour Line에서와 비슷하게 대칭성을 고려하면 14가지 경우로 축약시킬 수 있다[2].

### Cutting

Dataset을 임의의 평면으로 자르고 각 데이터 값을 보간하여 평면상의 값을 표현하는 것을 Cutting 또는 Data Cutting이라 한다. Cutting된 평면상의 값은 Color Mapping되어 화면상에 표현된다.

### Color Mapping

스칼라 알고리즘(scalar algorithm)에 속하는 Color Mapping은 Mapper에 전달된 데이터의 속성 데이터 중 스칼라 값을 Mapper 내부 변수인 Color Lookup Table을 이용하여 특정 색으로 변환하여 화면에 표시한다. 이들 색은 RGB 색상 값을 직접 입력하거나 HSV 색상 모델을 이용하여 RGB 색상 값을 생성하는 방법들이 사용되고 있다.

### 4.3 속성 변환 버튼 및 컨트롤

CDisplayDlg 클래스는 CModule 클래스와 협력관계에 있는 클래스로 CModule 클래스의 속성 즉 필터의 종류에 맞게 버튼 및 여러 가지 컨트롤 등을 바꿔주는 기능을 한다. 참고로

본 프로그램의 큰 특징 중에 하나는 각각의 기능을 구현하기 위한 메뉴가 별도의 공간을 필요로 하지 않는다는 것으로, 이 기능을 수행하는 것이 CDisplayDlg 클래스이다. 예를 들어 Clip 필터를 수행시켜 Clip 모듈을 생성해내면 Main View 오른쪽에 위치한 창에는 Clipping 기능과 관련된 여러 가지 컨트롤이 나타나지만, Cut 모듈을 생성해내면 이번에는 Cutting 기능과 관련된 컨트롤이 나타나. 여러 가지 가지화 작업을 수행할 수 있다. 이는 기능마다 각각의 대화상자(dialog)가 생성되어 Graphic Viewport를 가려서 대화상자를 매번 이동해야하는 번거로움을 덜어주는 편리한 기능이다.

## 5. 구성 및 적용 예

### 5.1 GUI 구성

본 연구를 통하여 개발된 프로그램의 화면구성은 Fig. 5와 같이 이루어져 있으며 각 영역의 기능은 다음과 같다.

- A) ToolBar : ToolBar는 자주 사용하는 기능들을 간단한 그림을 사용해 표시한 버튼형식으로 구성되어있다. 사용을 위한 별도의 교육이 없더라도 ToolBar에 그려진 기능을 나타낸 그림과 Tooltip을 보면 누구나 손쉽게 기능을 구현할 수 있다.
- B) Axes : 격자의 좌표축을 나타내며 마우스로 끌기를 하면 위치를 이동할 수 있다.
- C) Graphic Viewport : 3차원 그래픽 처리 영역.
- D) Slicing Window : 불러들인 볼륨을 x, y, z축 방향으로 잘라내어 위치를 조정하며 단면을 볼 수 있다.
- E) Property Sheet : 각 필터들의 세부 설정을 조정하는 컨트롤들을 생성한다.

### 5.2 Slicing

Slicing 기능은 MRI Dataset에서 읽어 들인 3차원 데이터와 이미지를 x, y축 방향으로 회전시켜 임의의 평면으로 자르고 각각의 절단면을 메인윈도우 하단의 서브윈도우의 화면상에 정면, 측면 및 상면을 표현한다(Fig. 6). 또한 이들 세 단면들은 하단의 슬라이드 바를 이용해 사용자가 보고 싶은 임의의 단면으로 이동시켜 볼 수 있게 하였다.

### 5.3 Contour Plot

Contour Plot 기능은 3차원 데이터에 대해서는 Iso-surface를 구현하고, 2차원 데이터에 대해서는 Contour-line을 구현한다(Fig. 7). 본 프로그램은 Contour 값의 최대값을 적색으로 최소값을 청색으로 표현하고 있다. 또 Contour 개수를 지정해서 자동으로 생성하거나 하나씩 수동으로 생성할 수 있으며, 메뉴창의 Contour Values List에서 Contour 값을 추가하거

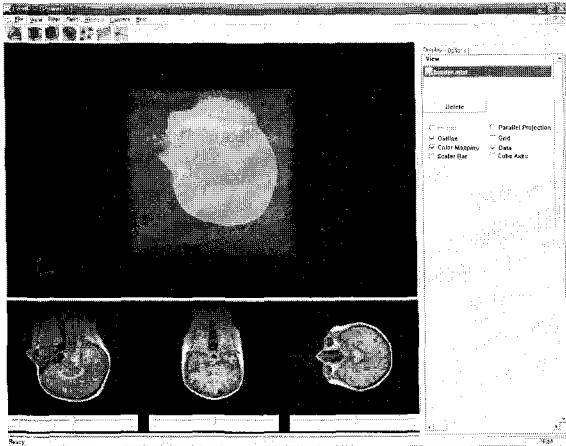


Fig. 6 Slicing Plot

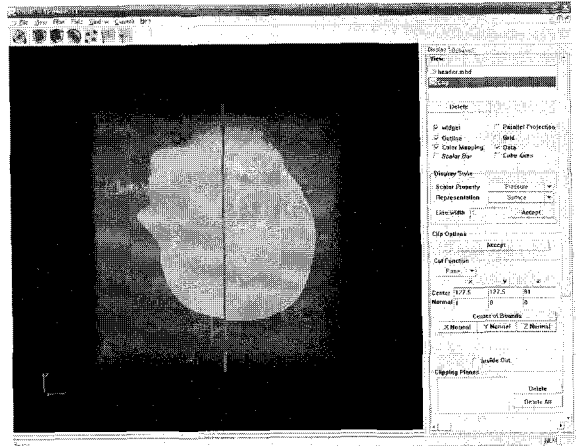


Fig. 8 Clip Plot

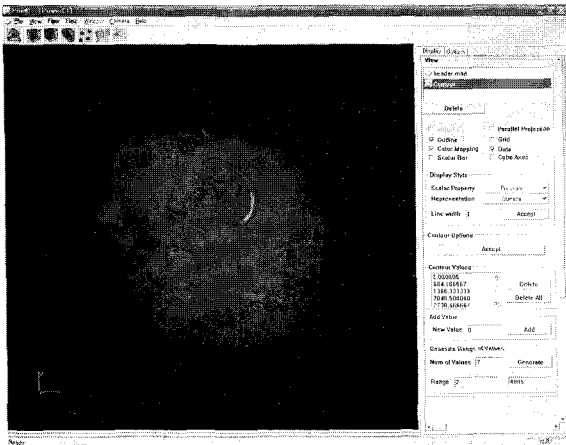


Fig. 7 Contour Plot

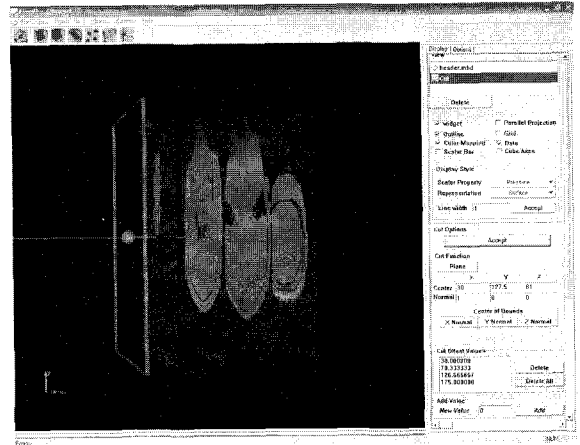


Fig. 9 Cut Plot

나 삭제할 수 있다. 만일 현재 Contour Plot으로 표현된 변수를 변경할 경우 자동으로 변경된 변수의 최대값과 최소값을 Contour 개수만큼 등간격으로 나누어 화면에 출력하므로 사용자는 자유롭게 변수를 변경하여 볼 수 있다.

#### 5.4 Clip Plot

ContourClip Plot은 Fig. 8에 나타낸 것과 같이 임의의 평면으로 절단된 나머지 데이터만을 화면에 나타낸다. Inside Out 기능이 있어 반대편의 데이터만을 볼 수도 있으며, 임의의 평면은 Plane Widget이라고 하는 도구에 의해서 자유롭게 설정할 수 있다. 또한 마우스로 화살표의 머리나 꼬리를 잡아 돌리면 평면도 따라서 회전하고, 평면을 이루고 있는 외각의 굵은 튜브를 마우스로 잡아끌면 평행이동을 할 수도 있다.

#### 5.5 Cut Plot

Cut Plot은 Fig. 6와 같이 3차원으로 표현된 데이터를 마우스를 사용해 임의로 설정 가능한 평면을 사용하여 자르고 단면을 볼 수 있는 기능이다. Offset 값을 입력함으로써 동시에 여러 개의 Cutting 단면을 생성할 수 있다(Fig. 9).

### 결 론

본 논문에서는 최근 자체 개발한 유동해석용 코드 (PowerCFD)의 후처리 장치[1]에 사용된 모듈(module)에 새로운 모듈을 추가하여 의료분야에서 널리 사용되는 MRI Data를 3차원 가시화하는 후처리 프로그램(PowerCFD/PostPv1.0)의 구조, 주요 기능 및 적용 예를 소개하였다. 이 후처리 프로그램은 C++ 언어로 짜여 있으며, VTK(Visualization

Toolkit[2]라는 객체지향기법(OOP)으로 구현된 3차원 그래픽 라이브러리를 사용함으로써 보다 강력하고 편리한 기능구현이 가능하고 모듈화를 이용한 단순한 구조의 프로그램을 통해 변경이나 확장에 유연하다. 또한, 3차원 가시화에 있어서 입력된 데이터(Siemens Medical Systems, Inc에서 제공한 109-slice MRI data)를 이미지로 표현하는 단순 그래픽 기능뿐만 아니라, 입력된 데이터를 필터 개념의 독립된 모듈로 구성함으로써 보다 자유로운 가시화 작업을 가능하게 하였다.

향후, 이 후처리 프로그램이 MRI 영상처리용 후처리기에 대한 국내 연구자들의 관심을 고취시켜 국내에서 경쟁력 있는 후처리기가 개발될 수 있도록, 다양한 실험적 기법들을 채용하고 테스트할 예정이다.

## 후 기

본 연구는 서울시 산학연 협력사업(2005년도 신기술연구개발 지원사업)의 연구비를 지원받아 수행된 연구이다.

## 참고문헌

- [1] 2006, Myong, H.K., and Ahn, J. Ki., "Development of a Post-Processing Program for Flow Analysis," *Proc. of 4th National Congress on Fluid Engineering, Kyungju, Korea*, pp.901-904, Aug.23-25.
- [2] <http://kitware.com>