

# 의료영상처리를 위한 시각 프로그래밍 환경

성 증 원<sup>†</sup> · 김 진 호<sup>††</sup> · 김 지 인<sup>†††</sup>

## 요 약

의료 영상 처리는 특수한 전문 분야로서 새롭게 개발되는 기술을 임상에 적용하고 그 결과를 전문 의사들이 분석하여 기술을 보완하면서 발전시켜 나간다. 그러므로, 새로운 의료 영상 처리 알고리즘을 임상에 쉽게 적용하고 그 결과를 분석할 수 있는 도구의 필요성은 매우 크다고 볼 수 있다. 본 논문에서는 시각 프로그래밍 기법을 도입하여 프로그래밍 전문 기술이 없는 일반 의사들도 의료 영상 처리 프로그램을 쉽게 작성할 수 있는 환경을 설계하고 구현하여 의료 영상 알고리즘의 적용을 편리하게 하려고 한다. 개발된 시각 프로그래밍 환경에는 의료영상처리에 필요한 다양한 함수들을 코딩하여 아이콘(Icon)으로 정의한 다음, 함수 라이브러리로 만들어 놓았다. 사용자는 영상 처리 응용 프로그램을 개발할 때에, 명령어나 변수를 타이핑하여 프로그램을 작성하는 대신, 아이콘으로 정의되어 제공되는 함수들 중에서 필요한 함수를 선택하고, 이들 사이의 자료흐름에 따라 아이콘들을 연결시켜서 프로그램의 논리를 자료흐름도(Dataflow Diagram)로 표현하여 작성하면 된다. "VPI(Visual Programming Interface)"라고 불리는 그래픽 편집기는 자료 흐름도를 그려내는 창과 준비된 함수 라이브러리를 시각화하여 보여주는 창으로 구성되어 있다. 함수 라이브러리에 포함된 함수들을 용도에 따라서 분류하고, 사용자가 필요한 함수를 찾기 쉽고 사용하기 편리하도록 시각화하여 하이퍼볼릭 트리(Hyperbolic Tree)라는 새로운 형태로 표현하였다. 하이퍼볼릭 트리를 사용하면 함수들의 개별적 특성과 함수들을 분류한 전체 구조를 한꺼번에 잘 볼 수 있다. 개발된 프로그램의 사용자 인터페이스를 쉽게 구현할 수 있는 "GUI Builder"라는 도구를 설계하고 구현하였다. 개발된 프로그래밍 환경을 사용하면 프로그래밍 전문 지식이 없는 의사도 쉽게 영상 처리 응용 프로그램을 작성하여 최신 의료영상 처리 기법을 쉽게 임상에 적용하고 실험할 수 있는 장점이 있다.

## A Visual Programming Environment for Medical Image Processing

Chong-Won Sung<sup>†</sup> · Jin-Ho Kim<sup>††</sup> · Jee-In Kim<sup>†††</sup>

## ABSTRACT

In medical image processing, if new technologies are developed, they are applied to real clinical cases. The results are to be analyzed by doctors to improve the new technologies. So, it is important for doctors to have a tool that helps the doctors in applying the new technologies to clinical cases and analyzing the clinical results. In this paper, we design and implement a visual programming environment where non-programming experts, such as medical doctors, can easily compose a medical image processing application program. A set of image processing functions are implemented and represented as icons. The user selects functions by clicking corresponding icons. The users can easily find necessary functions from the visualized library. A user selects a function from the visualized library and put the function node into a canvas of Visual Programming Interface. The user connects nodes to compose a dataflow diagram. The connected dataflow diagram shows the flow of the program. Hyperbolic Tree is helpful in visualizing a set of function icons in a single screen because it provides both the whole structure of the function library and the details of the focused functions at the same time. We also developed a GUI builder where the user interfaces of the medical image processing applications are composed. Therefore, non-programming experts such as physicians can apply new medical image processing algorithms to clinical cases without performing complex computer programming procedures.

\* 본 연구는 한국과학재단 핵심전문 연구비(961 0910 055 2) 지원으로 수행되었음.

† 준 회 원 : 인버스(주) 주임연구원

†† 준 회 원 : 액세서컴퓨터(주) 연구원

††† 정 회 원 : 건국대학교 컴퓨터공학과 교수

논문접수 : 1999년 9월 21일, 심사완료 : 2000년 7월 28일

## 1. 서 론

CT(Computerized Tomography), MRI(Magnetic Resonance Imaging), PET(Positron Emission Tomography), SPECT(Single Photon Emission Computed Tomography) 등과 같은 의료영상 장비가 발달하면서, 인체의 내부를 단층으로 촬영한 의료영상을 판독하여 신속하고 정확하게 환자를 진단할 수 있게 되었다. 이러한 의료 영상들은 의료기기의 종류나 촬영 방법, 환자의 질병 종류에 따라, 의사들이 쉽고 정확하게 영상을 판독할 수 있도록 여러 가지의 형태로 처리되거나 변환되어 사용된다. 예를 들어, 이들 단층 촬영된 의료 영상을 3차원으로 재구성하여 나타내면 의료영상 판독 훈련을 받지 않은 의사나 일반인들도 의료영상을 이해하기 쉬울 뿐만 아니라 모의수술, 원격진료, 의료교육 등의 분야에서 유용하게 사용될 수 있다[1].

의료 영상들을 다양한 형태로 처리하기 위한 영상 처리 알고리즘은 다양하게 개발되어 있다. 문제는, 다양한 의료기기에서 촬영한 다양한 포맷의 의료 영상들을 실제로 임상에 적용하는데 있어서 많은 시간이 걸리고, 전문 프로그래밍 교육을 받지 않은 일반 의사들의 입장에서 새로운 알고리즘을 자신의 컴퓨터에 구현하여 실제로 사용하는 것은 매우 어렵다는 것이다. 따라서 최신의 의료영상 처리 기법이 임상에서 신속하게 활용되어 환자들에게 직접적인 도움을 주지 못하고 있다. 본 논문에서는 시각 프로그래밍(Visual Programming) 기법을 이용하여 전문 프로그래밍 교육을 받지 않은 사람도 편리하게 의료영상처리를 위한 응용 프로그램을 작성할 수 있도록 도와주기 위한 프로그래밍 환경을 제안한다. 시각 프로그래밍 기법은 사용자 위주의 새로운 프로그래밍 방식이다. 즉, 함수 라이브러리(Library)에 정의되어 있는 의료영상처리를 위한 함수들을 아이콘으로 표현한 다음, 사용자가 필요로 하는 함수를 아이콘을 클릭하여 선택하고, 그들 사이의 자료흐름(Dataflow)에 따라 아이콘들을 연결하여, 응용 프로그램의 논리를 나타내는 자료흐름도(Dataflow Diagram)를 정의함으로써 프로그램을 작성하는 방법이다.

본 논문에서 제안하는 프로그래밍 환경을 사용하여 프로그램을 개발할 때, 사용자는 "VPI(Visual Programming Interface)"라는 그래픽 편집기를 이용하여 자료흐름도를 작성함으로써 필요로 하는 의료 영상

응용 프로그램의 논리를 정의한다. 또한, 응용 프로그램에서 사용하는 사용자 인터페이스를 쉽고 편하게 개발하기 위하여, 제안하는 프로그래밍 환경에서 제공되는 "GUI(Graphic User Interface) Builder"라고 불리는 도구를 사용한다. VPI와 GUI Builder, 두 가지 도구를 이용하면 쉽고 편리하게 의료영상처리 응용 프로그램을 개발할 수 있다. VPI에서 영상 처리 프로그램을 작성하는데 있어서, 함수 아이콘들을 서로 연결하는 경우, 함수들 사이의 자료흐름 경로를 점검하여, 함수들 사이를 흘러가는 자료들의 타입(Type)이 서로 맞게 설정되어 있는지, 등 점검을 하도록 한다. 이렇게 하여 사용자가 프로그램을 작성하면서 지지를 수 있는 형식상의 실수를 미리 알아내어 고칠 수 있도록 해 줌으로써, 사용자가 프로그래밍을 정확하고 편하게 할 수 있게 도와준다. GUI Builder에는 의료 영상 프로그램에서 일반적으로 필요로 하는 사용자 인터페이스를 쉽게 구현할 수 있도록 인터페이스를 구성하는 컴포넌트(Component)들이 정의되어 아이콘으로 표현되어 있다. 사용자는 필요로 하는 인터페이스 컴포넌트를 아이콘 창에서 골라 사용자 인터페이스를 쉽게 구성할 수 있다.

제한한 시각 프로그래밍 환경에서는 의료 영상 처리에 필요한 여러가지 함수들을 라이브러리로 만들어서 각 함수들을 특징별로 분류한 다음 아이콘으로 나타낸다. 아이콘으로 표현되는 함수들은 VPI의 "라이브러리 창"에 나열된다. 라이브러리에 등록되는 함수의 수가 많아지면, 라이브러리 창에 나타나는 아이콘의 수가 많아져서 한 화면에 모든 함수를 표시하기가 어렵고, 지금까지 창 옆에 붙은 스크롤바(Scroll Bar)를 사용한 방법으로는 사용자가 원하는 함수를 찾아보기가 쉽지 않다. 그래서, 제한된 크기의 "창"에 많은 아이콘을 표현하고, 쉽게 필요한 아이콘을 찾을 수 있도록 하는 작업이 필요하게 되었다. 본 논문에서는 라이브러리의 함수들을 분류하여 계층 구조를 갖도록 정리하였고, 그 전체 구조와 개별적인 함수들을 모두 한 화면에 나타내기 편리한 하이퍼볼릭 트리(Hyperbolic Tree)라는 새로운 사용자 인터페이스를 이용하여 라이브러리 창을 구현하였다.

본 논문의 구성은 다음과 같다. 제2장에서는 관련연구에 관하여 기술하고, 제3장에서는 본 논문에서 제안하는 의료영상을 위한 시각 프로그래밍 환경인 VPE(Visual Programming Environment)에 관하여 상세하

게 설명한다. 즉, 프로그래밍 도구인 시각 프로그래밍 인터페이스(VPI)와 사용자 인터페이스를 만드는 도구인 그래픽 사용자 인터페이스 빌더(GUI Builder)의 설계 및 구성에 대하여 설명한다. 제4장에서는 제안된 프로그래밍 환경을 구현하는데 있어서 필요한 요소와 구현 과정을 예제를 통하여 설명하고, 5장에서 제안하는 시각 프로그래밍 환경이 의료 영상 처리 분야에서 특히 유용하게 사용되는 이유에 관하여 토론한다. 마지막으로 6장에서는 결론 및 향후과제에 대하여 기술한다.

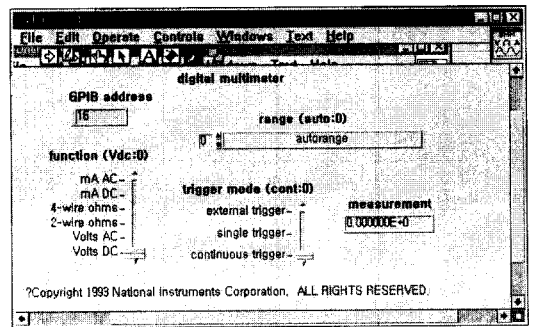
## 2. 관련연구

### 2.1 시각 프로그래밍

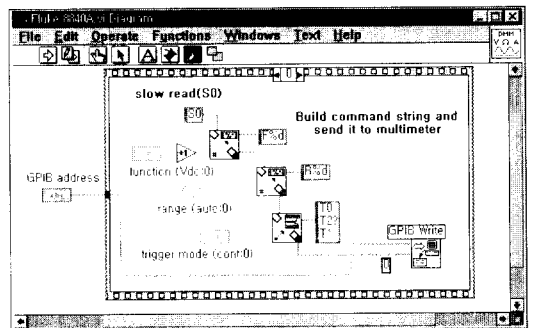
전통적인 텍스트 중심의 프로그래밍 작성 방법에서 벗어나서 프로그램을 아이콘(Icon), 플로우차트(Flow chart), 자료흐름도, 등을 이용하여 시각적으로 표현하고 작성하는 시각 프로그래밍 기법은 많이 연구되어 왔다. 현재 Visual Basic, Visual C++와 같이 사용자 인터페이스를 시각적으로 구축할 수 있도록 하는 방식은 상용화되어 널리 사용되고 있고, 더욱 발전하여 알고리즘 자체를 시각적으로 표현할 수 있는 시각 프로그래밍 기법이 발전하여 프로그래밍 전문 지식이 많지 않은 사람들도 프로그램을 할 수 있게 도와 주고 있다. 지금까지 개발된 시각 프로그래밍 도구들로서는 National Instruments의 LabVIEW[2], International AVS Center의 AVS[3], SGI의 Iris Explorer[4], New Mexico대학의 Khoros[5] 등이 있다. LabVIEW는 National Instruments에서 만든 시각 프로그래밍 도구로서 가상 계측기(VI, Virtual Instrument)를 PC에서 시각적으로 구현할 수 있도록 하는 소프트웨어이다. 가상 계측기를 개발하는데, VI의 사용자 인터페이스(VI의 "Front Panel"이라고 부른다)를 만드는 다음, 그 작동 원리를 별도의 창에 블록 다이어그램("Block Diagram")으로 시각적으로 작성하여 정의하도록 설계되어 있다.

예를 들면, (그림 1)의 왼쪽에 보이는 창에 가상 계측기(VI)의 사용자 인터페이스인 Front Panel 을 만들어 놓는다고 하자. 이 과정은 마치 Visual Basic이나 Visual C++에서 사용자 인터페이스를 정의하는 과정과 비슷하여, "Text Entry", "Combo Box", "Sliding Switch" 등 LabVIEW에서 제공하는 사용자 인터페이스

소 컴포넌트를 끌어서 VI의 Front Panel을 구성하게 된다. 다음 단계는 (그림 1)의 (a)에서 구성한 VI의 Front Panel의 입력 단자나 출력 단자가 작동되는 원리를 정의하는 것이다. VI의 Front Panel에 정의된 입출력 단자에는 각각 대응되는 아이콘이 (그림 1)의 (b)에 보이는 블록 다이어그램에 나타난다. 사용자는 VI의 Front Panel에 정의된 입력 단자를 통하여 들어오는 자료를 어떻게 처리하여 출력 단자로 내보내는지를 자료흐름도로 작성하여 그려 주어야 한다. LabVIEW에는 Numeric Display, Meters, Gauges, Thermometers, LEDs, Graphs 등 다양한 사용자 인터페이스와 함수 모듈이 개발되어 제공된다. 계측기를 이용한 신호 처리가 주된 응용 분야이므로 PC와 센서를 연결하는 인터페이스가 개발되어 있고, 공학용 신호처리 응용 프로그램을 만드는 데 주로 사용된다[2].



(a) VI의 사용자 인터페이스

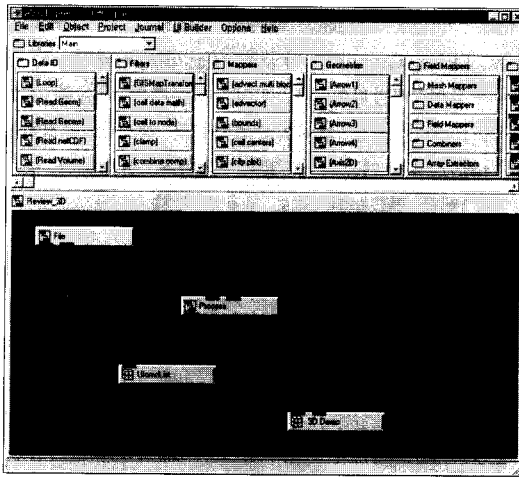


(b) VI의 동작 원리가 프로그램된 자료흐름도

(그림 1) LabVIEW 주요 창(Window)들

AVS(Advanced Visual Systems)는 International AVS Center에서 개발한 객체 지향 시각 컴포넌트 도구(Object Oriented Visual Component Tool)로서 시각

프로그래밍 기법을 이용하여 의료영상, 인공위성, 컴퓨터 시뮬레이션 분야 등 큰 규모의 데이터를 시각화하는 분야에 주로 적용된다. AVS를 이용한 시각 프로그래밍 환경은 거의 모든 과학기술 데이터 처리를 하는데 있어서, 전통적인 문자 명령어 방식의 프로그래밍이 아닌, 아이콘 조작과 흐름도 작성을 통한 프로그래밍 방식으로 수행해 낼 수 있도록 개발된 소프트웨어이다.



(그림 2) AVS의 주요 창들

AVS는 두 가지 창을 사용자에게 제공하고 있다. 첫 번째 창은 (그림 2)의 위 부분에 보이는 함수 라이브러리 창이고, 두 번째 창은 (그림 2)의 아래 부분에 보이는 자료 흐름도 창이다. “함수 라이브러리 창”에는 사용자가 필요로 하는 함수들을 미리 개발하여 종류별로 분류한 다음 직사각형의 아이콘으로 나열하여 놓았다. 예를 들면, “Field Mapper”라는 함수 그룹에는 “Mesh Mapper”, “Data Mapper”, 등의 함수들이 개발되어 아이콘 형식으로 정의되어 있다. 자료 흐름도 창은 사용자가 원하는 응용 프로그램을 개발하는 공간이다. 함수 라이브러리 창에서 필요로 하는 함수 아이콘을 선택하여 자료 흐름도 창으로 끌어 온다. 그 다음 선택된 함수들 사이의 자료 흐름을 화살표로 표현하여 아이콘들을 연결함으로써 프로그램을 개발한다. 프로그래밍 방식이 자세한 알고리즘의 구현 보다는 필요한 함수들을 선택하여 그들 사이의 자료의 흐름을 화살표로 연결하여 자료흐름도를 그리는 “Very High Level Programming” 방식이므로 컴퓨터 프로그래밍 전문가

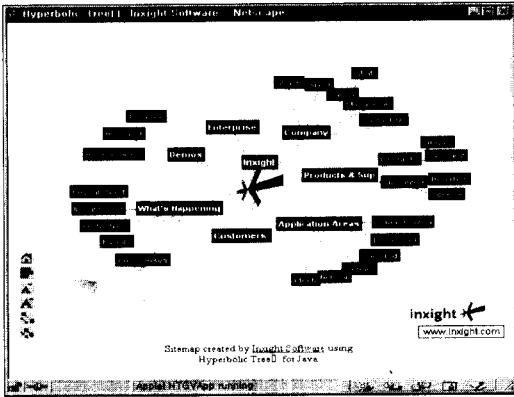
식이 없는 사용자들도 쉽게 프로그램을 개발할 수 있다는 장점이 있다. 또한, 개발된 프로그램을 빨리 이해할 수 있다는 것이 특징이다. 이러한 방식을 이용하면 컴퓨터 프로그래밍에 기울이는 시간과 노력을 현저하게 줄일 수 있기 때문에, 사용자는 데이터 자체에 대한 이해와 분석에 더욱 많은 시간을 할애할 수 있다.

## 2.2 계층정보의 시각화

지금까지 연구 개발된 시각 프로그래밍 기법에서 중요한 개념 중의 하나는 “필요한 함수들을 미리 개발하고 함수 라이브러리로 만들어서 아이콘으로 표현한 다음, 사용자가 원하는 함수를 선택하여, 다이어그램을 작성하여 프로그램을 개발한다”는 것이다. LabVIEW나 AVS와 같은 경우에도, 라이브러리에 수 많은 함수들을 개발해 놓고, 사용자들에게 보여 주고 선택하여 사용하게 한다. 그러나, 제한된 공간을 이용하여 많은 함수를 분류한 다음, 그 구조와 각각의 함수를, 계층 구조의 아이콘으로 표현하는 것은 매우 어려워져서, AVS 같은 경우, 하나의 창에 5개 정도의 함수 그룹을 보여 주며 각각의 그룹은 5개 정도의 함수를 보여 주게 된다. 만약 사용자가 원하는 함수가 현재 창에 보이지 않으면 사용자는 창을 스크롤(Scroll)하여 필요한 함수를 찾아가야 한다. 함수의 개수가 적을 때에는 이러한 작업이 큰 문제가 되지 않는다. 그러나 수백개 이상의 함수들을 포함하는 라이브러리를 나타내는 경우에는 기존의 윈도우를 이용한 스크롤 방식으로는 효율적인 검색이 어려워진다. 함수 라이브러리를 여러 개의 그룹으로 분류하고 각각의 그룹에 속한 많은 함수들을 효과적으로 하나의 창에 나타내는 방법을 없을까?

하이퍼볼릭 트리[6]는 계층적으로 구성되는 수 많은 정보를 하나의 창에 효과적으로 나타낼 수 있는 방법 중의 하나이다. 제한된 공간의 화면에, 나타내고자 하는 정보의 전체적인 구조와 각 정보의 상세한 내용을 한번에 나타내기 위하여 기존의 Euclidean좌표계 대신에 Hyperbolic좌표계를 사용하게 된다. 화면에 나타나는 Tree의 노드 크기를 볼록 거울에 반사되는 것처럼 표현하여, 현재 사용자가 관심이 있는 가운데 부분은 확대하여 더 많은 표현공간으로 표현하고 그 밖의 부분은 상대적으로 더 작은 공간에 표현하여 크기를 축소하여 나타내는 기술이다. 이러한 방식을 “Focus+Context”라고 부르는데, 이 기술을 사용하면 많은 노드를 하나의 창에 나타내어 전체 구조와 세부 사항을 동시에 보여줄

으로써, 기존의 평면적인 정보 표현방식이 갖고 있는 단점을 보완하는 장점이 있다[7]. 즉, 전체 함수 라이브러리 구조를 한 눈에 볼 수 있어서 어떠한 그룹이 있는지 쉽게 찾을 수 있고, 각각의 개별적인 함수를 확대하여 자세히 볼 수 있게 되어 정확히 원하는 함수를 선택할 수 있게 된다.



(그림 3) 하이퍼볼릭 트리

### 3. 의료 영상 시각 프로그래밍 환경

본 논문에서는 시각 프로그래밍 방식의 장점을 최대한 활용하여 의료영상 프로그램 개발을 신속하고 편리하게 할 수 있도록 도와 주는 프로그래밍 개발 환경을 구현하려고 한다. 이 목적을 달성하기 위하여 다음과 같이 연구 방향을 설정하였다.

#### ① 의료영상 처리 분야에 알맞은 시각 프로그래밍 도구 개발

의사가 시각 프로그래밍 도구를 이용하면 특정한 작업을 의사에 맞게 고쳐서 사용할 수 있다. 그러나 기존 시각 프로그래밍 도구들은 범용으로 제작된 것이 대부분이어서 의사들이 사용하기에는 어려움이 있다. 의료영상 처리를 위한 특수 목적의 시각 프로그래밍 도구가 필요하다.

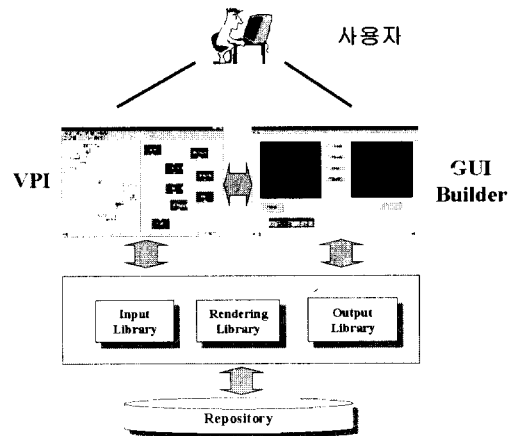
#### ② 사용자 인터페이스 개발 도구 적용

시각 프로그래밍 기법은 프로그래밍 방식을 시각화 하는데 초점을 맞추고 있다. 시각 프로그래밍 기법을 이용하여 쉽게 알고리즘을 구현한다 하더라도, 프로그램에 필요한 사용자 인터페이스는 따로 만들어야 하는

불편함이 있다. 실제로 사용 가능한 프로그램 도구를 만들려면 사용자 인터페이스 개발을 쉽게 해주는 도구가 필요하고, 알고리즘을 구현하는 도구들과 유기적으로 연결되어야 한다. 그래서, 사용자 인터페이스와 알고리즘을 하나의 환경 속에서 함께 개발할 수 있어야 한다.

#### ③ 함수 라이브러리의 시각화

소프트웨어 공학의 발전으로 여러 가지 유용한 함수들로 구성된 라이브러리를 사용하게 된다. 시각 프로그래밍에서 사용되는 함수들도 기본적으로는 라이브러리 형태로 제공되는 것이 일반적이다. 그러나 일반 사용자들은 어떤 종류의 함수가 존재하는지 쉽게 알아내기 어렵다. 라이브러리를 구성하고 있는 함수들의 개수가 늘어나면 사용자들의 어려움도 더욱 커지게 마련이다. 라이브러리에 속한 함수들을 효과적으로 표현하여 사용자가 쉽게 필요한 함수를 선택할 수 있어야 한다.



(그림 4) 의료영상 시각 프로그래밍 환경

본 논문에서 제안하는 시각 프로그래밍 환경에는 개발하고자 하는 응용 프로그램의 자료흐름도를 그리는 “VPI”와 개발되는 응용 프로그램의 사용자 인터페이스를 개발하는데 필요한 도구인 “GUI Builder”, 두 가지의 도구가 제공된다. 이 두 가지 도구는 서로 유기적으로 연결되어있다. 즉, VPI에서 정의하여 놓은 응용 프로그램의 입출력 함수들에 대응하여, GUI Builder에서는 필요한 사용자 인터페이스의 버튼이나 출력 창을 시각적으로 정의한다. (그림 4)에 예시된 것처럼, 사용자는 VPI를 사용하여 알고리즘을 시각적으로

로 정의하고, GUI Builder를 사용하여 응용 프로그램에 필요한 사용자 인터페이스를 구현한다. 현재 개발된 시가 프로그래밍 환경에는 입력 함수들을 모아놓은 "Input Library"와 출력 함수들이 구현되어 있는 "Output Library", 렌더링 함수들을 포함하는 "Rendering Library"가 준비되어 있다. 사용자는 VPI를 통하여 필요한 함수를 시각적으로 선택하고 연결하여 응용 프로그램을 개발하게 된다.

3.1 VPI(Visual Programming Interface)

VPI는 의료영상 처리 프로그램의 자료흐름도를 작성하는 도구이다. VPI는 함수들을 아이콘으로 표현한 "라이브러리 창"과 이 함수 아이콘을 이용하여 자료흐름도를 그릴 수 있는 "프로그램 창"으로 구성된다.

3.1.1 라이브러리 창

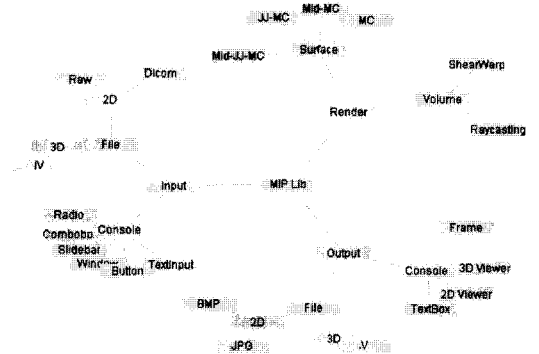
라이브러리 창에는 함수들이 아래 (그림 5)와 같이 계층적인 구조를 가지고 아이콘 형태로 표현되어 있다. 현재의 라이브러리는 입력, 출력을 담당하는 부분과 2차원이미지를 3차원으로 재구성하는 부분으로 구성된다[9]. 한 화면에 여럿개의 함수들과 그 구조를 알기 쉽게 표현하기 위하여 하이퍼블릭 트리로 표현하였다. 사용자는 원하는 함수를 찾기 위하여 마우스로 하이퍼블릭 트리를 움직이면서 필요한 부분을 검사한다. 그리고, 노드에 적힌 함수 이름을 보고 원하는 함수를 찾는다. 마우스 트리를 움직일 때 전체 노드의 구조가 바뀌어야 한다. 즉, 노드들을 매년 새로운 위치로 움직여준다. 이를 위한 수식은 다음과 같으며, 이는 z에 위치한 노드를 각도로 P만큼 움직인 위치를 구하는 식이다.

$$z' = \frac{\theta \times z + P}{1 + P \times \theta \times z}$$

z,  $\theta$ , P는 복소수,  
 $\bar{P}$ 는 P의 켈레복소수,  $|\theta| = 1$ ,  $|P| < 1$ .

(그림 5)에 나타난 하이퍼블릭 트리의 루트(Root) 노드는 "MIP Lib"라는 이름의 의료 영상 처리 함수들의 라이브러리이다. "Input"이라는 노드는 입력 함수 라이브러리를 가지고 있으며 다양한 의료장비에서 얻은 의료영상 데이터를 다양한 형식으로 읽어 들일 수 있는 모듈이 필요하다. 크게 보아 "File" 입력 함수들과 모니터에서 직접 자료를 읽어 들이는 함수들은 "Console"이라는 이름 아래 저장되어 있다. 의료영상의 표준 포맷인 "DICOM Format"을 읽는 함수들이 있고,

단순한 그림 형식을 읽는 "Raw Data Format" 입력 함수들도 제공한다. "Output"에는 출력 함수들이 정리되어 있다. 3차원으로 재구성한 3차원 영상을 파일로 저장할 수 있도록 지원하는 함수들 중에는 Open Inventor에서 사용하는 파일 형식("iv")과, VRML 2.0 파일 형식("vrl"), JPEG 압축 형식("jpg") 등을 지원하는 함수들이 존재한다. "Render" 노드 아래에는 2차원 볼륨 데이터를 3차원 영상으로 재구성하는 알고리즘 중에서 표면기반 렌더링 방법(Surface-based Rendering Method)들이 "Surface" 노드 아래 저장되어 있다. 볼륨기반 렌더링 방법(Volume-based Rendering Method)들을 기반으로 하는 함수들은 "Volume" 노드에 포함되어 있다. 표면기반 렌더링 방법은 볼륨데이터로부터 3차원으로 시각화 할 표면의 밀도(density) 값을 추출하여 삼각형 매쉬(Triangle Meshes)를 생성하는 방법이다. 현재, 마칭 큐브(Marching Cubes)[8], Dividing Cubes[9], 데시메이션(Decimation of Triangle Meshes)[10], 제이 제이(Jasper Jensen[14]) 등의 알고리즘이 구현되어 있다. 볼륨기반 렌더링 방법은 볼륨데이터로부터 중간단계 없이 직접 이미지를 생성하는 알고리즘으로 표면기반 렌더링 방법보다 정보의 손실이 없다는 장점이 있지만 처리시간이 오래 걸리는 단점이 있다. 현재, Ray-tracing 알고리즘[11], Shear-Warp 알고리즘[12]이 구현되어 있다.

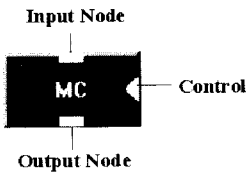


(그림 5) 의료 영상 처리용 함수의 계층 구조를 시각화하여 하이퍼블릭 트리로 표현

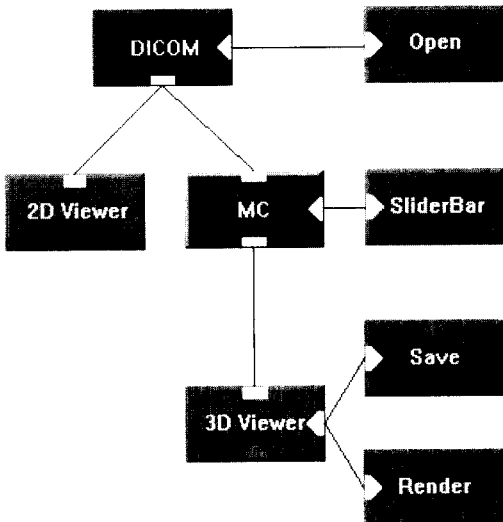
3.1.2 프로그램 창

응용 프로그램을 개발하기 위하여 함수 아이콘들을 선택하고 연결하여 자료흐름도를 그리는 창이다. 사용자는 드래그앤드롭("Drag and Drop") 방식으로 라이브

러리 창에서 필요한 함수를 선택하여 프로그램 창으로 오려 놓는다. 그러면 프로그램 창에 (그림 6)에 보이는 모양을 한 노드가 생성된다. 노드는 실행이 가능한 "함수"를 나타내는데, 작성하는 자료흐름도의 노드이다. 각각의 노드에는 3개의 단자가 정의되어 있는데, "Input Node", "Output Node", "Control" 단자들이다. 노드에 입력되는 자료는 입력 단자를 통해 들어오고, 함수 실행 결과에 따라 출력되는 자료는 출력 단자를 통한다. 입력 단자는 노드의 위쪽에 사각형으로 표현되고, 출력 단자는 노드의 아래 쪽에 사각형으로 표현된다. 함수 실행에 필요한 제어 신호나 값은 제어 단자를 통하여 함수 노드에 입력된다. 제어단자는 노드의 측면에 삼각형으로 표현된다. 자료나 신호의 흐름은 자료흐름도의 링크로 나타내며 사용자는 노드들의 입출력 및 제어 단자를 연결하여 노드들 사이의 자료흐름을 정의하게 된다[13].



(그림 6) 노드 구성



(그림 7) 자료 흐름도

(그림 7)에는 의료 영상들을 처리하여 3차원 입체

영상으로 만들어 표현하는 프로그램을 개발하기 위하여 작성된 자료흐름도가 예시되어 있다. 사용자는 (그림 5)에 나타난 함수 라이브러리 중에서 필요한 함수를 골라서 (그림 6)에서 보여지는 노드의 형태로 프로그램 창에 생성시킨 다음 그들 사이의 자료흐름을 링크로 연결하여 표현한다. 각각 노드의 속성을 정의하고 변경할 수 있다. 위의 예제를 보면, "DICOM" 형식으로 작성된 입력 영상 자료를 읽어서 "MC(marching cube)" 알고리즘을 수행시키고, 그 결과로 만들어진 3차원 영상을 "3D Viewer"에 표현해내도록 프로그램한 것이다. "2D Viewer"는 입력되는 2차원 의료 영상을 보기 위한 것이고 "3D Viewer"는 "MC"에서 만들어진 3차원 영상을 보여주는 기능을 가지고 있다. 그래서 "DICOM" 함수를 실행한 결과가 "MC"의 입력으로 사용되도록 자료흐름도의 두 노드들 사이에 링크가 연결되어 있고, "3D Viewer"에는 "MC"에서 출력된 3차원 영상이 입력되어 화면에 보여지도록 링크가 연결되어 있다. "Open"은 "DICOM" 함수에서 읽어 들일 파일을 지정하는데 사용되는 기능인데, 사용자 인터페이스에 입력 명령 단추로 표현된다. "Slide Bar"는 "MC" 알고리즘에서 필요로 하는 입력 값을 정하는데 사용되는 입력 도구이므로 "Control" 단자들을 서로 연결하였다. "Save"와 "Render"는 "3D Viewer"에서 사용하는 입력 단추들이고 "Control" 단자를 통하여 연결된다. 사용자는 이렇게 자료흐름도를 작성하여 프로그램을 개발하게 된다.

### 3.2 GUI Builder

GUI Builder는 Visual C++의 리소스 매니저, Visual Basic의 도구상자를 이용한 폼 설계 도구와 같이 사용자 인터페이스 개발을 위해 사용하는 도구이다. 이 도구는 VPI를 이용하여 개발한 프로그램을 직접 조작할 수 있는 사용자 인터페이스를 구축하는데 사용된다. VPI의 라이브러리 창에 정의된 입출력 함수들에 대응되어 사용자 인터페이스 단자들이 GUI Builder 창에 나타난다. VPI와 GUI Builder는 상호간에 연결성을 가지고 있어서 사용자가 VPI에서 입력 함수나 출력 함수를 생성하면, GUI Builder에는 생성된 함수의 기능을 수행하는 인터페이스 구성요소가 생성된다. 반대로, GUI Builder 자체에서 직접 인터페이스를 설계하여 인터페이스 구성요소를 만들면 VPI에서 자동적으로 이 인터페이스 요소와 관련된 입출력함수 노드가 생성된

다. 예를 들어, VPI에 "Radio Button"이라는 함수 노드가 정의되면, GUI Builder 창에 라디오 버튼이 실제로 구현되어 나타난다. 반대로, (그림 8)의 메뉴에 있는 도구상자에서 가져다가 GUI Builder작업 창에 라디오 버튼을 정의하면, VPI에 "Radio Button"이라는 함수 노드가 정의된다. 이렇게 VPI와 GUI Builder 창은 그 내용이 서로 일치하여야 한다.



(그림 8) GUI Builder의 사용자 인터페이스

GUI Builder로 사용자 인터페이스를 만들 때, 실행 모드와 편집모드, 두 가지의 모드가 있다. 편집모드에서는 사용자 인터페이스를 설계하고, 실행모드에서는 최종 사용자가 이용할 수 있는 상태로 변환된다. (그림 8)에 예시된 것은 GUI Builder에서 사용자 인터페이스를 정의하는데 사용되는 단축 아이콘들이다. 이들의 기능을 간단히 살펴보면 다음과 같다.

**열기버튼, 저장버튼** 등을 사용하여 사용자 인터페이스를 개발한 결과를 파일에서 읽어들이거나 저장한다. **입력단자버튼**과 **출력단자버튼**을 사용하여 입출력을 위한 단자를 생성하게 된다. 예를 들면, 2차원 뷰어(Viewer)를 가지고 2차원 이미지를 보여주는 창을 생성하거나 3차원 뷰어를 가지고 3차원 이미지를 보여주는 창을 생성할 수 있게 된다. **링크버튼**은 자료흐름도에 정의된 노드들을 연결하여 자료흐름을 표현하는 링크를 나타내는데 쓰인다. 입출력 단추들을 그룹으로 묶어서 정의하는 경우, **프레임버튼**을 가지고 그룹의 경계를 정의한다. **포인트버튼**은 노드나 링크와 같은 자료흐름도의 객체를 선택하는데 필요하여 사용되며, **채우기버튼**은 선택된 노드들의 색을 칠하는 도구이다. **텍스트버튼**은 화면의 아이콘에 문자를 쓸 수 있도록 하는 도구이다.

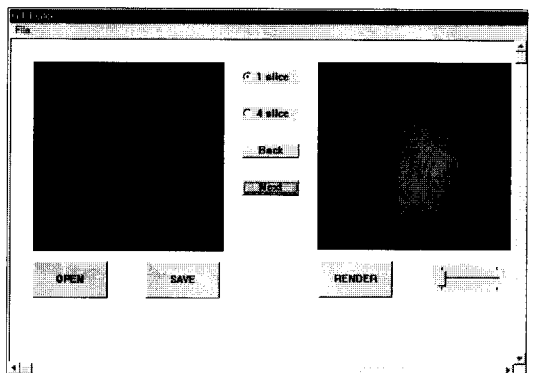
#### 4. 프로그래밍 예

일반적으로 CT, MRI와 같은 단층 촬영기를 사용하여 얻어낸 영상들을 처리하여 3차원 영상으로 만들면 이해도가 훨씬 증가하게 된다. 단층 촬영된 의료 영상을 3차원으로 재구성하는 알고리즘은 이미 많이 개발되어 있다. 이러한 알고리즘을 이용하여 환자의 의료

영상을 3차원으로 표현하는 프로그램을 개발한다고 가정하자. 보통은C, C++와 같은 프로그래밍 언어를 사용하여 함수를 개발해야 한다. 그러나, 전문 프로그래밍 교육을 받지 않은 의사가 직접 개발하는 경우, C나 C++를 사용하여 프로그램하기는 매우 어렵다. 만약, 시각 프로그래밍 환경을 이용하여 프로그램을 작성하게 되면 그 어려움이 훨씬 덜해질 것이다. 시각 프로그래밍 환경을 이용하여 3차원 영상 재구성 및 조작 프로그램을 개발하는 과정을 예를 들어서 살펴보자. 의사들은 사용자 인터페이스를 먼저 정의하고, 이 인터페이스를 작동시키는데 필요한 프로그램의 자료흐름도를 작성한다. 함수들의 아이콘을 선택하고 그들을 연결하여 자료흐름도를 시각적으로 정의하여 프로그램을 개발하게 된다.

##### 4.1 사용자 인터페이스 만들기(그림 9) 참조)

GUI Builder를 이용하여 프로그램 수행에 필요한 사용자 인터페이스를 (그림 9)에 보여지는 것처럼 만든다. 먼저, GUI Builder를 "편집모드"로 설정하고 이들 구성요소를 나타내는 아이콘들을 선택하여 사용자 인터페이스 화면에 가져와서 각각의 요소를 정의한다. **2D Viewer**는 (그림 9)의 사용자 인터페이스의 왼쪽에 정의되어 있는, 단층 촬영된 의료 영상을 출력하는 화면이다. 이 화면에 1장의 영상을 표현할 것인지("1 slice") 또는 4부분으로 나누어 4장의 영상을 출력할 것인지를("4 slice") 정의하는 라디오 버튼이 정의된다. 그리고, 현재 영상 보다 하나 이전의 영상으로 이동하는 "Back" 버튼과 다음 장을 보여주는 "Next" 버튼이 있다.



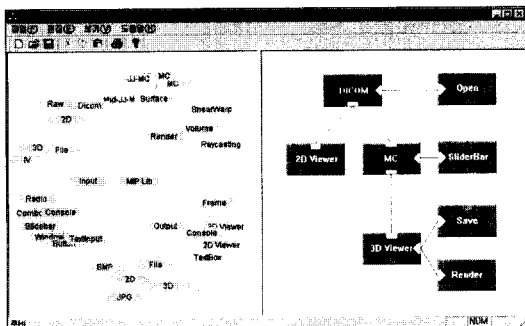
(그림 9) GUI Builder를 사용하여 작성한 결과물



**3D Viewer**는 사용자 인터페이스의 오른쪽에 정의되어 있는, 3차원 영상을 출력하는 화면이다. **Open Button**은 단층 촬영된 영상 파일을 불러들이는 단추이다. 3차원 영상을 지정된 파일로 저장하게 하는 단추는 **Save Button**이다. **Slide Bar**는 3차원 재구성 알고리즘인 마칭 큐브 알고리즘에서 영상 처리에 필요한 입계값을 설정해주는 인터페이스로서 바를 움직여서 값을 변경시키고, 마칭 큐브 알고리즘을 원하는 대로 수행시킨다. 마칭 큐브 알고리즘이 재구성한 3차원 영상을 3D Viewer에 렌더링하게 하는 명령은 **Render** 단추를 사용하여 수행한다.

4.2 자료흐름도 작성하기(그림 10 참조)

프로그램의 자료흐름도는 VPI를 이용하여 작성한다. 앞에서, GUI Builder를 이용하여 사용자 인터페이스를 작성하였다. 그 과정에서 인터페이스 구성 요소와 대응되는 함수들이 자동으로 VPI의 프로그램 창에 생성된다. 예를 들어, GUI Builder에 “2D Viewer”를 정의하면, 그것에 대응하여 VPI의 프로그램 창에는 “2D Viewer” 기능을 수행하는 함수의 아이콘이 생겨난다. 이렇게 생성된 사용자 인터페이스 관련 함수들 이외에 필요한 영상 처리 함수를 정의한다. 제공되는 함수 라이브러리에서 프로그램 작성에 필요한 함수들을 찾아서 더블클릭하면 프로그램 창에 함수 노드들이 생성된다. 예를 들면, 마칭 큐브 알고리즘을 수행하기 위하여 라이브러리를 검색하여 해당 아이콘을 더블클릭하면 프로그램 창에 “MC”라는 아이콘이 생겨난다.



(그림 10) VPI로 모듈을 구성한 예

다음, 프로그램 창에 생성되어 있는 함수 노드들의 아이콘을 연결하여 프로그램의 자료 흐름을 정의한다. 예를 들어, “DICOM” 함수는 의료 영상 표준 형식인

DICOM 양식을 갖는 단층 촬영 영상 파일을 읽고, 그 출력은 “MC” 즉 마칭 큐브 알고리즘의 입력으로 전달된다. 그러므로 사용자는 “DICOM” 노드의 출력단자와 “MC” 노드의 입력단자를 선으로 연결한다. 또한, “DICOM” 함수가 있어야 할 입력 파일 이름은 “Open” 단추를 눌러 입력되므로, “Open” 노드로부터 “DICOM” 노드로 제어선이 연결되어야 한다.

이제 사용자 인터페이스와 이를 작동시키는 프로그램의 자료흐름도가 작성되었다. 개발자는 이제 이들을 컴파일하게 된다. 의사나 사용자는 개발된 응용 프로그램의 입출력 단자들을 선택하여 입력 파일을 지정하고 자료를 입력하며, 영상 처리에 필요한 파라미터를 입력하여 프로그램을 실행하게 된다. 그 결과는 지정된 뷰어를 통하여 표현된다.

5. 성능 평가

앞에서 본 논문에서 제안한 시각 프로그래밍 환경을 사용하여 3차원 영상 재구성 프로그램을 작성하는 과정을 예를 들어 살펴보았다. 과연 시각 프로그래밍이 사용하기 편한가? 프로그래밍 기술이 없는 초보자들도 쉽게 사용할 수 있는가? 이 기법은 의료 영상 분야에 얼마나 유용하게 사용될 수 있는가? 등의 문제를 생각하면서 본 논문에서 제안한 시각 프로그래밍 환경의 성능을 평가하려고 한다.

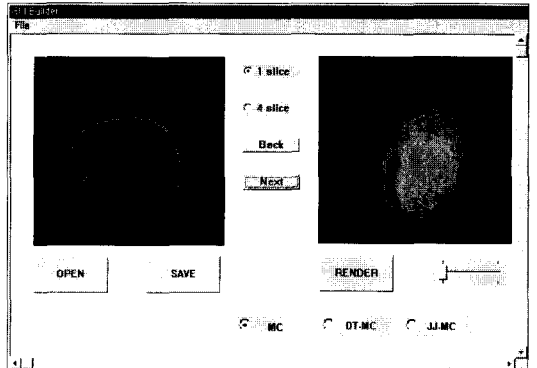
앞에서 살펴 보았듯이, 본 시각 프로그래밍 환경에서는 C나 C++와 같은 프로그래밍 언어를 사용하지 않아도 3차원 의료 영상을 다루는 응용 프로그램을 쉽게 작성할 수 있었다. 다양한 영상 처리 알고리즘이 구현되어 라이브러리와 되어 있고, 아이콘 형태로 제시되어 있으므로 의사들은 사용자 인터페이스를 정의하고 이를 작동시키는 프로그램을 시각적으로 정의할 수 있었다. 즉, 사용자 인터페이스는 표준화된 입출력 단자들을 사용하여 쉽고 편하게 작성하고, 필요한 함수를 마우스 클릭하여 자료흐름도를 그려서 원하는 응용 프로그램을 쉽게 개발하였다. 의료 영상 처리는, 분야의 특성상 많은 임상 적용이 필요하므로, 의사들이 적극적으로 개입하여 의견을 제시하고 결과에 대한 전문적인 판단을 제시하여 응용 프로그램의 기능을 계속하여 향상시켜가는 작업이 요구된다. 따라서, 의사들에게 다양한 실험을 할 수 있고, 프로그램을 여러 가지 형태로 바꾸어 볼 수 있으며, 사용하기 쉽고 편리한 프로

그래밍 도구를 제공해주는 것이 상당히 중요하다. 그런 면에서 시각 프로그래밍 환경의 역할이 중요하며 의료 영상 처리 기술의 발전과 이를 임상에 적극적으로 적용하는 일에 유용하게 사용될 것으로 생각된다.

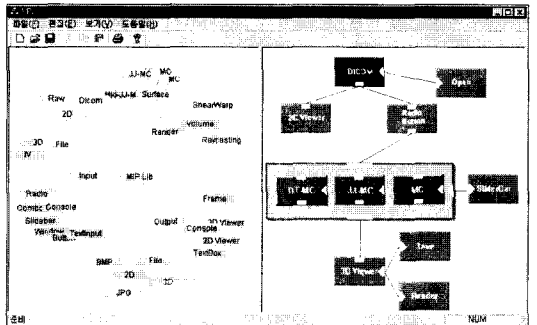
시각 프로그래밍 환경을 이용하여 의료 영상 프로그램의 기능을 확장하는 예를 보면서 시각 프로그래밍 환경이 얼마나 사용하기에 편한지 살펴보기로 한다. 예를 들어, 앞에서 (그림 9)와 (그림 10)에 예시된 것과 같은 프로그램을 가지고 의사들이 3차원 영상을 만들어서 결과를 분석하였다고 하자. 의사들이 보기에 영상은 사용하기에 좋은데, 마칭 큐브 알고리즘을 가지고 만들어낸 3차원 영상의 크기가 커서 디스크에 저장하기에도 문제가 있고, 네트워크를 통하여 전달하는데도 부담스럽다. 그래서, 영상 파일의 크기를 줄이기 위한 알고리즘들이 개발되었는데, 만약 의사들이 새로운 파일 축소 알고리즘을 개발하거나 구현하여 실험을 하려면, 상당한 프로그래밍 기술과 시간을 투자하여야 할 것이다. 그러나, 본 시각 프로그래밍 환경을 사용하면 불과 10분 이내에 파일 축소 알고리즘을 적용하여 실험을 해 볼 수 있다. 3.1.1에서 기술한 바와 같이, 현재 영상 파일 축소 알고리즘인 DT-MC(테시메이션 [10])와 JJ-MC(제이 제이[14])가 구현되어 함수 라이브러리에 저장되어 있다. 개발자는 GUI Builder를 사용하여 (그림 11)에 보이는 것처럼 (그림 9)에 보여지는 사용자 인터페이스에다가 3차원 영상 파일 축소 알고리즘을 선택할 수 있는 라디오 버튼을 추가한다. 그래서, MC 단추를 고르면 마칭 큐브 알고리즘을 실행하고, DT-MC 단추를 누르면 테시메이션 알고리즘을, JJ-MC를 선택하면 제이 제이 알고리즘을 실행하여 3차원 영상을 만들어 볼 수 있도록 한다. 자료흐름도 역시 VPI를 사용하여 (그림 10)의 자료흐름도를 약간만 바꾸면 된다. 즉, (그림 12)에 보이는 대로, "Radio Button" 선택 노드가 삽입되고, 거기에 복합 노드가 정의되어 "MC", "DT-MC", "JJ-MC", 세 개의 노드를 포함하게 하면 된다. 그렇게 되면, 라디오 버튼 단추에서 선택되는 대로 해당되는 알고리즘이 실행될 것이다.

지금까지 살펴본 바와 같이, 본 논문에서 제안한 시각 프로그래밍 도구는 의료 영상 처리 프로그램을 개발하고, 임상에 적용하여 그 결과를 토대로 새로운 알고리즘을 적용해 보면서 프로그램을 개발하는데 있어서 매우 필요한 도구이다. 특히, 전문 프로그래밍 기술을 가지고 있지 않은 의사들이 사용하기에 쉽고 편하

므로, 의료 영상 처리 분야에서 쓸모가 많은 도구가 될 것이다.



(그림 11) 다양한 3차원 영상 처리 알고리즘을 사용하는 사용자 인터페이스



(그림 12) 다양한 3D 영상 처리 알고리즘을 구비한 (그림 11)의 사용자 인터페이스의 기능을 구현하기 위한 자료흐름도가 작성된 VPI 화면

## 6. 결론 및 향후과제

본 논문에서 제안한 의료영상 처리를 위한 시각 프로그래밍 환경은, C 언어나 C++ 언어와 같은 프로그래밍 언어를 알지 못해도, 사용자가 High-Level에서 개발하고자 하는 프로그램의 자료흐름도를 시각적으로 정의하여 의료영상상을 위한 응용 프로그램을 쉽고 편하게 개발할 수 있도록 도와 주는 도구이다. 시각 프로그래밍 환경에서는 의료 영상 처리에 필요한 함수들이나 사용자 인터페이스 요소들이 미리 만들어져서 아이콘 형태로 표현되어 제공되므로, 사용자는 필요한 함수나 기능을 선택하여 노드로 만들고, 그들의 자료흐름이나 제어흐름에 따라 링크로서 연결하여 자료흐름

도를 만들어 응용 프로그램으로 완성하면 된다. 개발하고자 하는 응용프로그램의 사용자 인터페이스는 사용자들이 직접 WYSIWYG(What You See Is What You Get) 방식으로 제작할 수 있고, 대응되는 프로그램의 자료흐름도를 함께 검토하여 실행 가능한 코드를 생성해낸다. 이러한 시각 프로그래밍 도구는, 의료 영상 처리 작업을 하여 임상에 적용하고 연구를 해야 하는 의사들에게는 매우 유용한 도구가 될 것이다.

본 논문에서 제안하는 시각 프로그래밍 환경에는 의료 영상 처리에 필요한 많은 함수들이 제공이 되는데, 사용자가 많은 함수들 중에서 필요한 함수를 빨리 찾아내는 것은 매우 어려운 일이다. 우리는 하이퍼볼릭 트리라는 새로운 방식의 표현 방법을 이용하여, 계층 구조의 정보를 쉽고 자세하게 표현하고 검색할 수 있도록 하였다. 함수 라이브러리에 저장된 함수들의 전체 구성도를 한 화면에 볼 수 있고, 필요한 부분은 선택하여 좀더 자세히 볼 수 있도록 블록기울 형태로 표현한다. 그래서, 사용자가 쉽고 편하게 필요한 함수를 찾아서 원하는 자료흐름도를 작성할 수 있도록 하였다.

앞으로 진행하여야 할 연구과제로는 (1) 함수 라이브러리를 보강할 수 있는 도구를 개발하는 것이다. 현재 구축된 의료 영상용 함수 라이브러리에 새로운 함수 코드를 추가하여 라이브러리의 기능을 확장할 것이다. (2) VPI의 프로그램 창에서 자료흐름도를 작성할 때, 온라인 파싱 방법을 사용하여 Syntax 및 Semantics, Type Checking 등을 온라인으로 점검하고 사용자가 입력한 자료흐름도에 문제가 있는 경우 자동으로 "Error Recovery" 할 수 있는 기능을 향상시키는 것이다. (3) 3차원 의료영상은 그 크기가 방대하므로, 사용자가 온라인으로 이미지를 처리하고 조작하기 위해서는 막대한 계산량이 요구된다. 따라서, 대규모의 의료 영상 데이터를 처리를 위한 병렬 처리 기법을 연구하고, 최적화된 3D 및 영상, 동영상 표현 기술을 개발하여야 할 것이다.

### 참 고 문 헌

- [1] 김진호, 김지인, "다중 매체 방식의 3차원 의료 영상 표현을 위한 VRML 시스템 개발", 한국정보과학회 가을학술발표논문집, Vol.23, No.2, pp.417-420, 1996.
- [2] G. Johnson, 'LabVIEW Graphical Programming : Practical Applications in Instrumentation and Control', Second Edition, McGraw Hill, 1997.
- [3] S. Punzi, 'AVS/Express Provides Powerful Visualization Techniques for AISS S.P.A.', Published in DM Review, 1999.
- [4] R. Iles, A. van Maanen, S. Turner and J. Walton, "IRIS Explorer Module Writer's Guide(NT)," The Numerical Algorithms Group Ltd., Oxford, UK, 1997.
- [5] M. Young, D. Argiro, and S. Kubica, 'Cantata : The Visual Programming Environment for the Khoros System', Khoros Research, Inc, 2000.
- [6] 김기준, 류수연, 이상호, 김지인, "Hyperbolic Tree를 이용한 Bookmark의 시각화", 한국정보과학회 봄학술발표논문집, 제26권 제1호, pp.422-424, 1999.
- [7] J. Lamping, R. Rao, and P. Pirolli, "A focus + context technique based on hyperbolic geometry for visualizing large hierarchies," In Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, ACM, 1995.
- [8] W. Lorensen, and H. Cline, "Marching Cubes : A High Resolution 3D Surface Construction Algorithm," IEEE Computer Graphics, Vol.21, No.4, pp. 163-169, July 1987.
- [9] H. Cline, W. Lorensen, S. Ludke, C. Crawford, and B. Teeter, "Two Algorithms for the Three-Dimensional Construction of Tomograms," Medical Physics, Vol.15, No.3, pp.320-327, June 1988.
- [10] W. Schoroder, J. Zarge, and W. Lorensen, "Decimation of Triangle Meshes," IEEE Computer Graphics and Application, Vol.26, No.3, July 1992.
- [11] M. Levoy, "Display of Surfaces from Volume Data," IEEE Computer Graphics and Application, Vol.8, No.5, pp.29-37, May 1988.
- [12] P. Lacroute and M. Levoy, "Fast volume rendering using a shear-warp factorization of the viewing transformation," IEEE Computer Graphics Proceedings, Annual Conference Series(SIGGRAPH '94), Orlando, pp.451-458.
- [13] M. Burnett, A. Goldberg, and T. Lewis, 'Visual Object-Oriented Programming', Manning Publications Co, pp.10-19, 1995.
- [14] J. Jensen, '3D Visualization', Denmark Tekniske University Press, 1995.



**성 종 원**

e-mail : cwsung@cse.konkuk.ac.kr  
1996년 건국대학교 전자공학과  
졸업(학사)  
1999년 건국대학교 컴퓨터공학과  
졸업(석사)  
1999년~현재 인버스(주) 주임  
연구원

관심분야 : 시각 프로그래밍, 컴퓨터 그래픽스, 의료  
영상처리, 등



**김 진 호**

e-mail : jhkim@cse.konkuk.ac.kr  
1995년 호서대학교 컴퓨터공학과  
졸업(학사)  
1997년 건국대학교 컴퓨터공학과  
졸업(석사)  
1997년~현재 건국대학교 컴퓨터  
공학과 박사과정

1997년~현재 앤써커뮤니티(주) 연구원  
관심분야 : 컴퓨터 그래픽스, 의료영상처리, 가상현실, 등



**김 지 인**

e-mail : jnkm@kkucc.konkuk.ac.kr  
1982년 서울대학교 컴퓨터공학과  
졸업(학사)  
1984년 한국과학기술원 전산학과  
졸업(석사)  
1993년 University of Pennsylva-  
nia 전산학과 졸업(박사)

1982년~1987년 금성통신(주) 주임 연구원  
1987년~1993년 University of Pennsylvania 연구조교  
1993년~1995년 미국 CCCC 전산과학자  
1995년~현재 건국대학교 컴퓨터공학과 부교수  
관심분야 : 시각 프로그래밍, 의료영상처리, 컴퓨터  
그래픽스, 가상현실, 등