



맞춤형 사용자 환경을 제공하는 가시화 시스템 설계

(Customized Next-generation Visualization System Design)

김 민 아 (petimina@kisti.re.kr)

한국과학기술정보연구원
Korea Institute of Science & Technology Information

목차

1. 개요	1
2. GLOVE 기능 요구 사항	2
가. 응용 독립적 기능 요구 사항	2
1) 데이터 로딩	2
2) 데이터 관리	2
3) VR 장비와 tiled-display의 지원	3
4) 병렬 렌더링	3
5) Drawing Object의 관리	3
6) 협업	3
나. 로터 시뮬레이션 데이터 가시화를 위한 사용자 요구사항	4
1) 블레이드 표면 데이터 가시화	4
2) 필드 데이터를 위한 요구사항	5
3) 데이터 로딩	2
3. GLOVE HW Architecture	8
4. GLOVE SW Architecture	10
가. GLOVE Framework 와 응용 특화된 기능을 위한 계층구조 ..	10
나. GLOVE Components	11
1) GIVI	12
2) UI Widget Manager	13
3) Data Mgr	13

4) DOTree	13
5) GIVI Renderer	13
6) GIP Mgr	13
7) Collaboration Mgr	13
8) GLORE	14
9) GLORE Renderer	14
5. GLOVE 클래스 설계	14
가. Collaboration Diagram	14
1) GIVI	15
2) vrj::VTKApp	15
3) dotree::Group	15
4) glvRotorDataSet	15
5) glvRotorRenderer	15
6) glvDataMgr	15
7) vtkVJPicker	16
8) glvFieldDataSet	16
9) glvWallDataSet	16
10) glvDataSet	16
11) glvData	16
나. Sequence Diagram	16
6. 결론	18
7. 참고문헌	19

그림 차례

[그림 1-1] 블레이드 표면 압력 분포 가시화	4
[그림 1-2] 블레이드 표면 데이터 그래프 메뉴	5
[그림 1-3] 필드 데이터 압력 분포 가시화	6
[그림 1-4] Vorticity Iso-surface 기능	7
[그림 1-1] Vorticity Contour Line 기능	7
[그림 1-1] Velocity Stream Line 기능	8
[그림 1-1] 블레이드 표면 압력 분포 가시화	4
[그림 2-1] GLOVE HW Architecture	9
[그림 3-1] GLOVE Basic Framework and Data Management Hierarchy	11
[그림 3-2] GLOVE Components	12
[그림 4-1] GIVI Collaboration Diagram	14
[그림 4-2] Pressure Distribution Visualization Sequence	17

1. 개요

컴퓨팅 자원의 급격한 성능향상은 지구과학, 고에너지물리(High Energy Physics, HEP), 천체물리, 항공우주공학 등 다양한 e-Science 응용 분야들이 복잡한 과학 현상들을 더 정밀하고 규모가 큰 실험(또는 시뮬레이션)으로 가능하게 하였다. 이에 따라 고성능 컴퓨터를 이용한 시뮬레이션이 만들어내는 데이터의 크기도 페타 바이트 수준으로 기하급수적으로 증가하고 있지만, 대용량/고해상도 데이터를 분석하기 위한 가시화 도구의 성능은 아직 데이터양의 증가속도에 미치지 못하고 있다.

가시화 도구의 성능에 대한 요구와 수요로 인해, 과학적 가시화 관련 기술도 병렬 데이터 처리, 고해상도 렌더링, 복잡한 데이터를 알아보기 쉽게 보여주기 위한 표현 방법을 지원하도록 발전해 왔다. 이러한 기술적 발전과 요구를 수용하기 위해 개발된 가시화 도구들은 각 응용분야 연구자들의 기호에 따라 사용되고 있다. Tecplot, ParaView, COVISE 등이 국내외 연구자에게 알려진 대표적인 예다.

Tecplot은 데이터 가공이나 렌더링 과정에서의 병렬처리 능력이 상대적으로 취약하고, tiled display와 같은 고해상도 가시화 시스템을 지원하지 않기 때문에 대용량, 고해상도 데이터를 수용하기 힘든 단점을 갖고 있다. ParaView의 경우, 높은 범용성과 뛰어난 병렬처리 능력을 갖추고 있지만 데이터의 정량적인 분석 도구가 미비하며, 입체 영상을 지원하지 않기 때문에 복잡한 구조를 파악하는 데에는 어려움이 따른다. 마지막으로 COVISE는 대용량 데이터를 위한 병렬처리 및 협업 환경, VR 환경 등을 지원하지만 다양한 응용들을 수용할 목적으로 설계되었기 때문에 응용연구자들이 사용하기에 다소 어렵다. 또한, 정량적인 데이터 분석 도구도 지원하지 않는다.

본 문서에서는 고성능 컴퓨팅 환경에서 대용량 시뮬레이션 데이터를 효과적으로 분석하고 가시화하여 공유할 수 있는 프레임워크인 GLOVE에 대해 기술한다. GLOVE는 가상현실 기반의 가시화/협업 프레임워크로, 아래의 특징을 갖는다.

- VR 기반 통합 환경 내에서 데이터의 정성적, 정량적 분석

-
- 병렬 처리를 통한 대용량 데이터의 효율적 분석 및 가시화
 - 단일 모니터에서 가상현실 장비에 이르기까지 다양한 형태의 가시화 장비 지원 원격지 사용자와 가시화 결과를 실시간으로 공유할 수 있는 협업 환경
 - 원격지 사용자와 가시화 결과를 실시간으로 공유할 수 있는 협업 환경

GLOVE는 이러한 기본 프레임 위에 사용자 인터페이스를 각 응용분야에 특화시켜서 해당분야 연구자들이 접근하기 용이한 통합 환경을 제공한다. 본 문서에서는 로터 시뮬레이션 데이터 해석을 위한 맞춤형 가시화 도구의 개발을 통해, 기본적인 GLOVE의 프레임워크를 설계한다.

2. GLOVE 기능 요구 사항

GLOVE의 요구 사항은 기능적 관점에서 볼 때 크게 응용에 독립적으로 GLOVE의 기본 프레임을 형성하는 기능들과 응용에 의존적인 기능들이다. 가시화 도구의 개발에 앞서 이런 기능들은 분류되어야 하며, 이러한 분류작업을 거쳐 GLOVE는 계층적으로 이들을 구성하여 상호 의존성을 최소화 하여야 한다. 즉, 응용이 바뀌더라도 수정을 최소화 하도록 설계하여야 한다. 이를 위해 본 문서에서는 로터 시뮬레이션을 위한 요구 사항을 도출하여 그 기능들을 정의함으로써, 기본 프레임을 설계한다.

가. 응용 독립적 기능 요구사항

1) 데이터 로딩

대용량 데이터를 다루는 가시화 도구에 있어 데이터 입출력을 효율적으로 빠르게 수행하는 것은 매우 중요하다. 이는 하드웨어적 지원과 소프트웨어적 지원이 모두 이루어져야 가능하다. 이를 위해 가시화 도구는 기본적으로 데이터를 읽어 들일 때, 클러스터 시스템과 병렬 입출력, 병렬 렌더링이 함께 고려되어야 한다. 또한, 대용량 데이터의 경우 시스템 캐쉬가 무용하므로, 캐쉬를 보완할 다른 메커니즘도 고려해야 한다.

2) 데이터 관리

맞춤형 사용자 인터페이스를 제공하는 가시화 도구는 범용 프레임워크 위에 응

용별 요구 사항을 반영하는 사용자 인터페이스를 올려야 한다. 이를 위해 가시화를 위한 데이터 구조는 가시화를 위해 범용 데이터 구조로부터 응용을 위한 데이터 구조로의 계층적 관리가 필요하다.

3) VR 장비와 tiled-display의 지원

대용량 데이터는 가시화할 경우 그 크기로 인해 고해상도를 필요로 한다. 그러나, 단일 모니터 내에서 이러한 고해상도를 얻기 어려우므로 tiled-display 기술들이 등장하게 되었다. tiled-display 는 여러 대의 모니터로 하나의 데이터를 영역을 나누어 가시화 하므로, 이들 모니터 간의 동기화는 물론, 협업을 지원할 경우, 이를 위한 프로토콜과 네트워크의 지원까지 필요로 한다. GLOVE 는 대용량 데이터의 가시화를 목표로 하므로 이러한 tiled-display 장비뿐 아니라 장비를 지원하기 위한 기술적 지원을 필요로 한다.

4) 병렬 렌더링

병렬렌더링은 대용량의 데이터를 가시화 하기 위해 필수적으로 요구되는 기능이다. 사용자가 참을 수 있는 대기 시간 내에 데이터를 처리하기 위해, 여러 개의 프로세서가 동시에 렌더링을 할 수 있어야 하며, 병렬 렌더링을 위해 적절한 데이터 분배와 결과의 통합이 이루어 져야 한다.

5) Drawing Object 의 관리

어떤 응용을 위한 가시화 도구를 만든다 할지라도, 화면에 여러 데이터를 가시화 하고, 메뉴를 관리하기 위해서는 화면에 그려진 object들을 관리할 필요가 있다. GLOVE 는 화면에 그려진 object들을 적절하게 관리하여 기능요구 사항들이 원활하게 수행될 수 있도록 해야 한다. 기존의 Scene Graph 형태이든 GLOVE를 위한 특별한 구조이든 Object의 관리 방안이 있어야 한다.

6) 협업

기본적으로 협업은 화상회의 기능과 데이터 공유 기능 크게 두 가지로 나누어 볼 수 있다. 화상회의 기능에 대한 기술들은 이미 많은 상용 제품이 나와 있을 정도이나, Tiled-display의 VR 환경에서의 화상회의는 또 다른 이슈들을 포함한다. 화상과 소리와 동기화 뿐만 아니라 좌우 두 이미지의 동기화, 그에 더하여 여러 개의 tiled-display에 대한 동기화가 모두 함께 이루어 져야 한다. 이들 각각은 동기화 메커니즘을 가지고 있지만, 모두를 통합하여 동기화 하는 기능은 새로이 설계되어야 한다. 이에 더하여 공유하는 데이터의 동기화까지 포함하여야

한다.

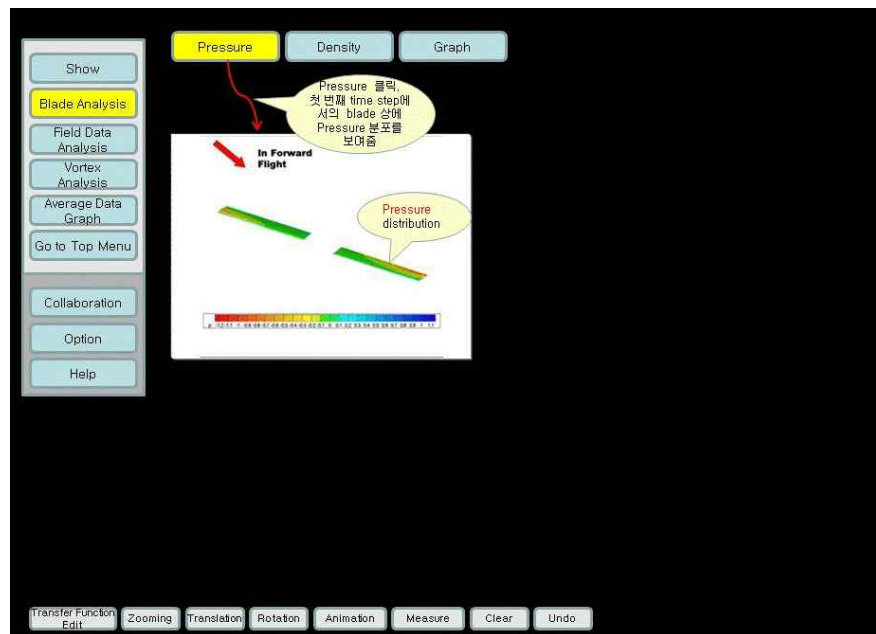
나. Rotor 시뮬레이션 데이터 가시화를 위한 사용자 요구사항

1) 블레이드 표면 데이터의 가시화

블레이드 표면 데이터란 로터 시뮬레이션 시에 블레이드에 나타나는 압력, 밀도, 속도 등의 변화 값을 의미한다. GLOVE 는 이들 데이터를 적절하게 가시화 하여 표현할 수 있어야 한다.

가) 블레이드 표면의 압력 분포 가시화

[그림 1-1]과 같이 블레이드 표면에 특정 범위의 압력에 해당하는 데이터영역들은 동일한 색깔로 표현함으로써, 압력 분포를 보여줄 수 있어야 한다. 로터 시뮬레이션에 있어 표면의 압력 변화는 주의해서 관찰되어야 할 요소이다.



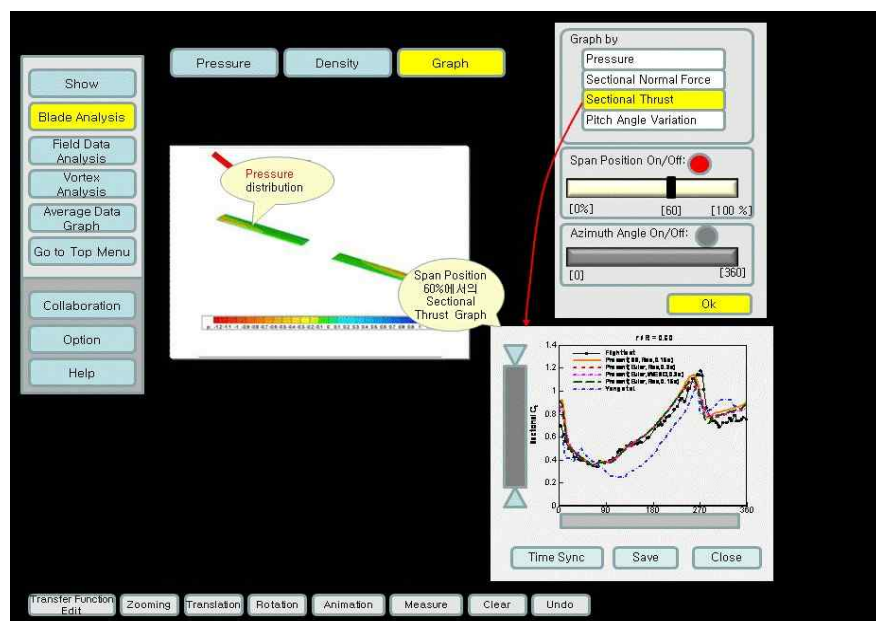
[그림 1-1 블레이드 표면 압력 분포 가시화]

나) 데이터의 정량적 분석

[그림 1-2]와 같이 사용자는 데이터의 정량적 분석을 위해 원하는 그래프의 형태와 범위를 지정할 수 있다. 블레이드 데이터 분석을 위해 GLOVE 가 제공해야 하는 그래프는 다음과 같다.

- Pressure distribution : 주어진 span position에서 회전각에 따른 압력의 변화를 정량적으로 보여준다.

- Pitch angle variation : 특정 span position과 0~360°의 회전각에 대한 pitch angle의 변화를 보여준다.
- Sectional force : 주어진 span position과 회전각에 대한 sectional normal force, sectional chord force, sectional span force 정보를 보여준다.
- Sectional moment : span position과 회전각에 대한 sectional x-moment, sectional y-moment, sectional z-moment 그래프를 커브와 디스크 형태로 제공한다.



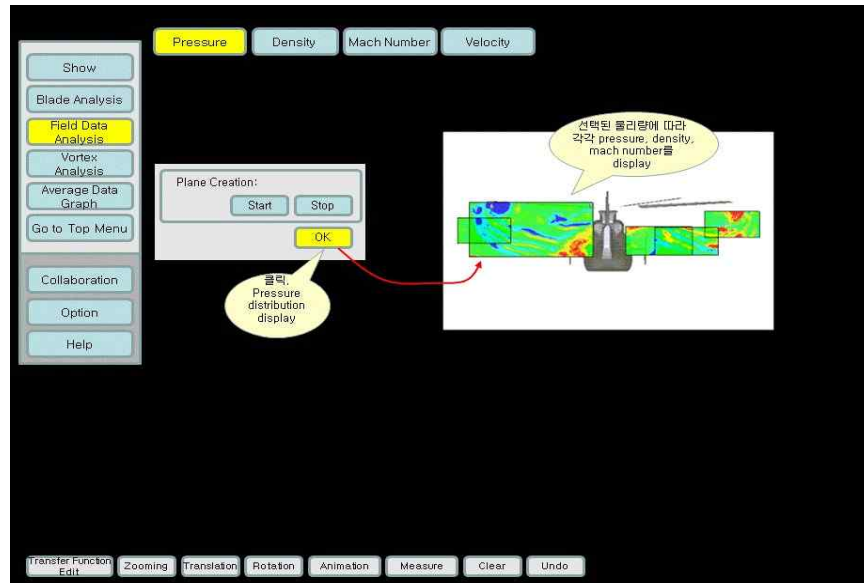
[그림 1-2 블레이드 표면 데이터 그래프 메뉴]

다) Animation

시간의 흐름에 따른 블레이드 표면의 압력 분포 변화를 그래프의 타임스텝과 동기화해서 애니메이션으로 보여줄 수 있어야 한다.

2) 필드 데이터를 위한 요구사항

GLOVE는 VR 환경의 3차원 가시화를 지원하므로, 이들 데이터를 가시화함으로써, 블레이드와 필드 데이터 사이의 상호작용을 명확하게 시각적으로 확인할 수 있다. 이를 위해 GLOVE는 다음의 기능을 제공할 수 있어야 한다. 필드 데이터란 로터 시뮬레이션 후에 생성되는 데이터로 블레이드 주변의 압력, vorticity, velocity 등의 데이터를 말한다. 블레이드의 표면 위에 분포하는 유의미한 데이터와 구분하기 위해 필드 데이터로 총칭한다.



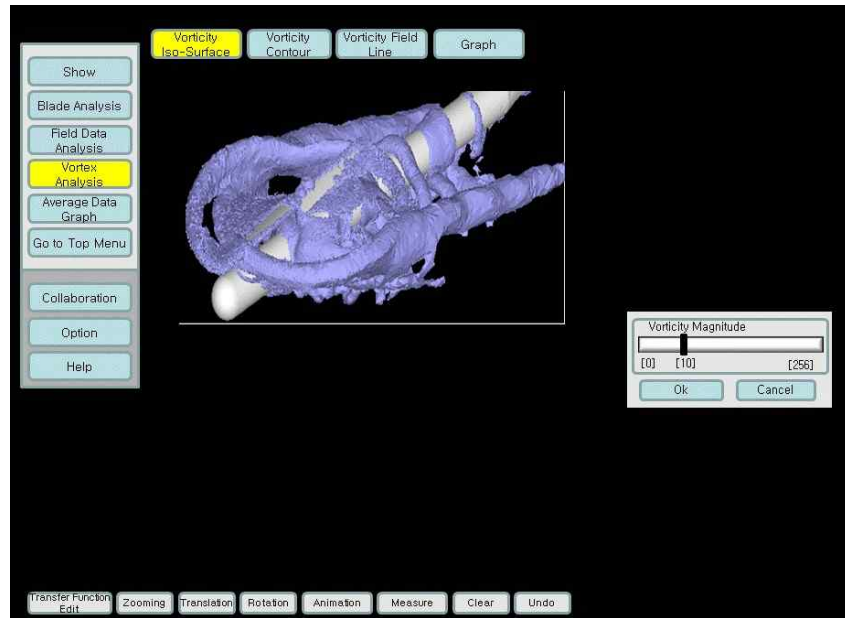
[그림 1-3 필드 데이터 압력 분포 가시화]

가) 필드 데이터의 부분적 가시화

로터 시뮬레이션 후 후처리에 의해 생성되는 필드 데이터의 기본은 압력분포 데이터 들이다. 필드 데이터의 압력 분포는 블레이드 표면 데이터와 달리, 전 영역에 걸쳐 있는 매우 방대한 분량의 데이터들이다. 따라서 이중 의미가 있는 영역을 추출해서 보여줄 필요가 있다. GLOVE 는 로터 시뮬레이션의 분석을 위해, 이들 필드 데이터에서 블레이드 주변, 동체 주변 등 특정 부분과 인접하여 유의미하고 관심 있는 영역을 추출하여 가시화 할 수 있는 기능을 제공하여야 한다. [그림 1-3]은 그 예를 보여 준다.

나) Vorticity iso-surface

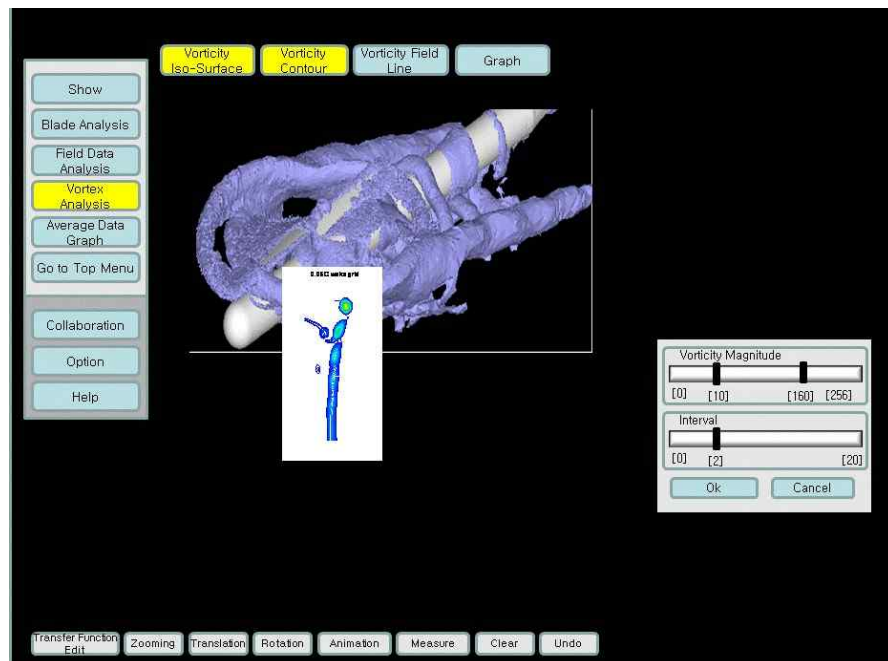
주어진 vorticity 값을 갖는 iso-surface를 보여준다. 보고자 하는 vorticity의 크기 값을 입력 받아 그럴 수 있어야 한다.



[그림 1-4 Vorticity Iso-surface 기능]

다) Q-Criteria

Vortex 영역을 가시화 하는 것은 vorticity iso-surface 와 같으나 영역의 추출 기법을 Q-criteria 기법을 사용한다. 이러한 기능을 제공함으로써, 두 가지 중 더 나은 방법으로 vortex 영역을 보여 줄 수 있어야 한다.



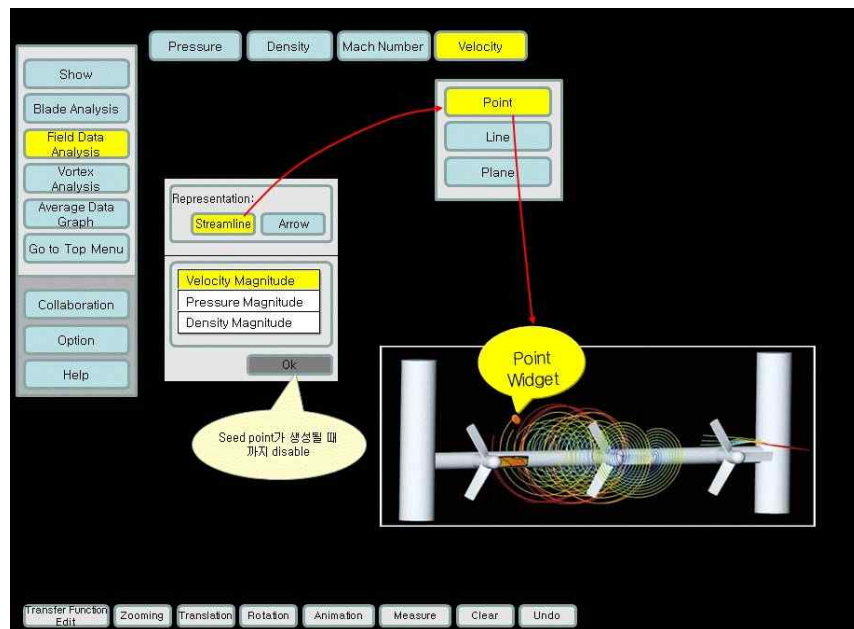
[그림 1-5 Vorticity Contour Line 기능]

라) Vorticity Contour line

동일한 vorticity 값을 갖는 점들을 이은 라인을 보여 줌으로써, vortex 영역의 단면을 가시화할 수 있다. vortex 영역의 특정 단면을 자를 수 있어야 하며, 그 단면 영역에 vorticity contour line을 그릴 수 있어야 한다.

마) Velocity streamline

관심 있는 영역을 지정하거나 vorticity 값이 특정 값 이상인 부분에서 velocity streamline을 보여 줄 수 있어야 한다. [그림 1-6]은 그 예를 보여준다.



[그림 1-6 Velocity Stream Line 기능]

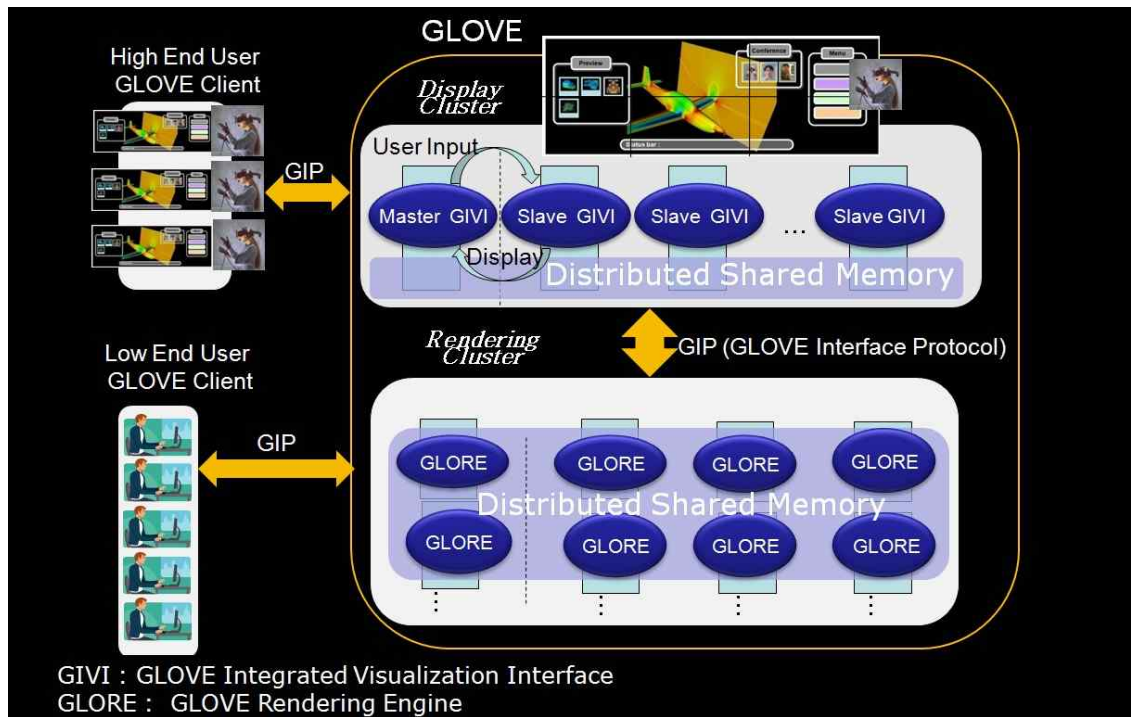
바) Velocity iso-surface

사용자가 지정한 velocity 값을 갖는 iso-surface를 보여준다. 사용자가 그리고자 하는 velocity 크기 값을 지정할 수 있다.

사) Animation

블레이드 데이터와 vorticity, velocity 에 대한 데이터를 가시화하여 블레이드 회전 시 vorticity와 velocity 변화를 관찰할 수 있다.

3. GLOVE H/W Architecture



[그림 2-1 GLOVE HW Architecture]

[그림 2-1]은 GLOVE의 기본 구조를 보여 준다. GLOVE는 사용자 인터페이스인 GIVI(GLOVE Integrated Visualization Interface)와 데이터 가공 및 렌더링을 담당하는 GLORE(GLOVE Rendering Engine)의 두 부분으로 나뉜다. GIVI는 가상현실 입/출력 장치를 총괄하며, 사용자로부터 입력을 받아서 GLORE에 전달하고 GLORE의 실행 결과를 전달 받아서 화면에 출력한다. GLORE는 대용량 데이터의 가공 및 렌더링을 위한 GIVI의 서브시스템이다.

GIVI는 VR 환경을 제공하는 프레임워크인 VRJuggler로 구현 하여 키보드와 마우스뿐만 아니라 IS-900, CAVE, tiled display와 같이 특수한 형태의 가상현실 입출력 장치를 지원한다. 따라서 사용자는 책상 위의 PC부터 CAVE나 대형 tiled display에 이르는 다양한 가시화 시스템에서 GLOVE를 동일한 형태로 실행할 수 있다. GIVI는 VRJuggler 어플리케이션의 동작 매커니즘에 따라 마스터 프로세스와, 마스터와 동일하게 동작하는 슬레이브 프로세스로 구성된다. 여러 대의 컴퓨터와 디스플레이로 구성된 대형 가시화 시스템에서 동작할 경우, 서로 다른 GIVI 슬레이브 프로세스가 각 디스플레이의 렌더링을 담당한다.

일반적으로 GLOVE는 GLORE와 GIVI가 서로 독립적으로 운영되는 클러스터에

서 실행된다. 이러한 구조는 GLOVE의 인터페이스(GIVI)와 핵심 렌더링 기능(GLORE)을 물리적으로 분리할 수 있기 때문에, 충분한 그래픽 처리능력을 갖추지 못한 시스템을 가지고 있는 사용자가 자신이 보유하고 있는 시스템에서는 GIVI만 실행하고, 원격지의 고성능 컴퓨터에서 실행되는 GLORE를 이용해 대용량 데이터를 렌더링 할 수 있다. 이렇게 분산된 시스템 구조에서는 데이터의 공유를 위한 효율적인 전송이 반드시 필요하다. GLOVE는 GIVI와 GLORE 사이의 효율적인 데이터 전송을 위해 초고속 네트워크 기반의 분산 캐시 메커니즘을 이용한다. 분산 캐시 메커니즘은 디스크 입출력 횟수를 줄이고, 서로 독립적으로 운영되는 클러스터 사이의 병렬 데이터 전송을 가능하게 한다.

가시화 기능 외에도, GLOVE는 렌더링 결과를 원격지 사용자와 공유하면서 원격 회의를 진행할 수 있는 협업 환경을 지원한다. 특히 HD 급(1920×1080) 이상의 해상도를 갖는 다수의 비압축 영상과 대용량 데이터의 실시간 전송을 위해 독자적 프로토콜인 GIP (GLOVE Interface Protocol)으로 원격지 연구자 간의 화상 회의와 렌더링 이미지 교환, 데이터 공유를 할 수 있다.

4. GLOVE S/W Architecture

가. GLOVE Framework 과 응용 특화된 기능을 위한 계층 구조

2절의 GLOVE 요구사항에서 언급한 바와 같이 GLOVE 는 크게 응용 독립적으로 수행해야 할 기능과 응용에 맞게 설계 되어야 할 부분으로 나뉜다. [그림 3-1]은 응용에 특화된 기능과 기본적으로 대용량 데이터를 다루는 가시화 도구가 제공해야 할 기능에 대한 기능적 분류에 이에 대한 데이터의 계층적 구조를 보여준다.

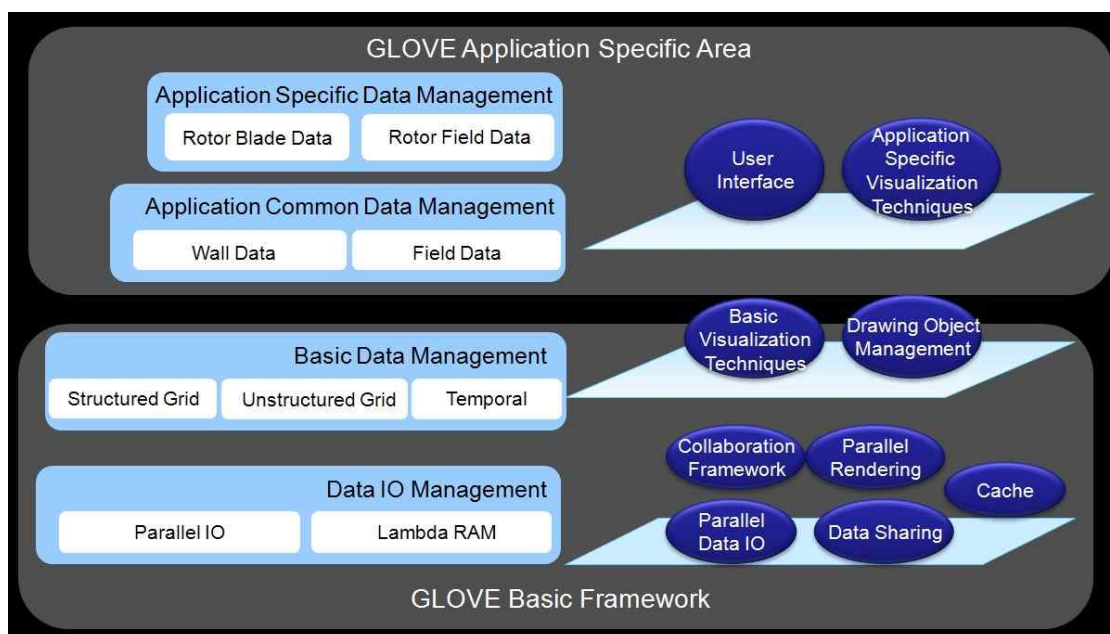
응용 독립적 기능들은 여러 노드에서 동일한 파일에 대한 접근을 효율적으로 만들어 주는 Parallel IO 기능과 입출력 속도를 개선하는 Cache 기능, 이들 읽어들인 데이터를 병렬 렌더링 하는 부분, 병렬 렌더링을 위해 데이터를 공유하는 부분, 협업을 위한 기능 등이 있다. 이러한 기본 기능 위에, 기본적인 가시화 기술, 가시화 기술로 표현되어 화면에 그려진 오브젝트를 관리하는 Drawing Object Management들이 응용에 독립적인 GLOVE Framework 라 한다면, [그림 3-1]의 왼쪽에 그려진 데이터는 이를 데이터적인 관점에서 계층화한 그림이다.

데이터 계층 구조의 가장 아랫부분에 위치한 Data I/O management 계층에서는 데이터의 병렬 입/출력 및 공유 캐시(cache) 등의 메커니즘을 관리한다.

Basic data management 계층에서는 가시화를 위한 structured grid, unstructured grid 등 기본적인 메시 구조와 다중 블록(multi-block) 데이터 및 시변환(time-variant) 데이터를 위한 자료구조를 관리한다.

Application common data management 계층에서는 GLOVE가 지원하는 응용분야에서 공통으로 활용하는 자료구조를 다룬다. CFD에 있어서는 벽면 데이터와 유동장 데이터를 이 계층에서 정의할 수 있다. 의료영상과 같은 다른 분야의 응용을 추가할 경우에는 이 계층부터 새로운 자료구조를 정의한다.

Application specific data management 계층에서는 특정 응용분야에서 사용하는 자료구조를 관리한다. 로터 시뮬레이션 데이터 해석을 위한 가시화의 경우 로터 블레이드 데이터, 로터 필드 데이터 등을 이 계층에서 정의한다.

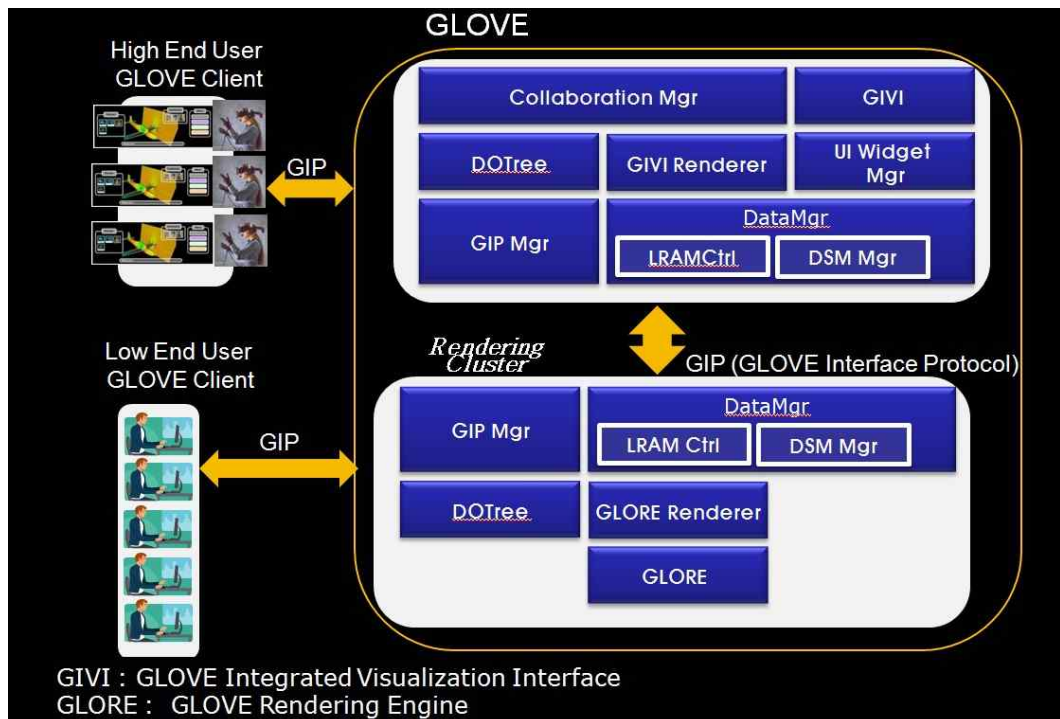


[그림 3-1 GLOVE Basic Framework and Data Management Hierarchy]

나. GLOVE Components

GLOVE의 기능과 계층적 Data Management를 고려한 GLOVE 의 SW Architecture 는 [그림 3-2]과 같다. [그림 3-2]은 [그림 3-1]의 HW Architecture를 구현하기 위한 SW Architecture이다. GLOVE 는 앞서 설명한 바와 같이 VR 인터페이스를 지원하는 VRJuggler와 기본적인 가시화 도구의 라이브러리를 제공하는 VTK, VRJuggler와 VTK를 이어 주는 vjVTK 세 가지 툴킷으로 구현된다.

GIVI 는 크게 사용자 인터페이스를 구성하는 UI Widget Manager 와 GIP을 제어하는 GIP Manager 그리고, 위에서 언급한 계층적 데이터 구조를 관리하는 Data Manager 와 GIVI 에서의 Rendering 기능을 수행하는 GIVI Renderer, Collaboration을 주관하는 Collaboration Mgr과 마지막으로 전체를 제어하는 GIVI, 7개의 컴포넌트로 구성된다. GLORE 내의 GIP Mgr과 Data Mgr은 렌더링 명령의 전달과 데이터 공유를 위해 display cluster의 GIVI 와 rendering cluster 의 GLORE 내에 모두 존재한다. 또한 DOTree는 렌더링 메커니즘을 돕기 위해 Drawing Context를 공유할 목적으로 GIVI와 GLORE 내에 모두 존재한다.



[그림 3-2 GLOVE Components]

1) GIVI

VRJuggler 프레임워크 내에서 사용자 입력에 따른 인터랙티브한 가시화 기능을

수행한다. 모든 컴포넌트들을 제어한다.

2) UI Widget Manager

UI Widget Manager 는 개발 도구인 VRJuggler, VTK, vjVTK 와 밀접하게 연관되어 사용자 인터페이스를 구성하는 widget을 관리하는 컴포넌트이다. 즉 vjWidget 클래스들을 관리한다. 이러한 widget 들은 VR 인터페이스 도구인 wand 와 메뉴, 메뉴를 제어하는 callback 구조 등을 포함한다.

3) Data Mgr

Data Mgr은 LambdaRAM Controller 와 분산 메모리 관리자인 DSM Mgr을 포함하는 데이터 관리 기능을 수행한다. GIVI Renderer 가 필요로 하는 데이터를 메모리나 디스크에서 읽어 들여 원하는 형태로 가공해 준다.

4) DOTree

DOTree는 화면에 그려진 오브젝트들을 관리하는 자료구조로 오브젝트의 삭제 그려진 순서, 이동 시 동일한 transform 함수가 반영되는 오브젝트들의 그룹핑 등을 책임진다.

5) GIVI Renderer

GIVI 에서의 렌더링 기능을 담당하는 컴포넌트로 GLORE 와 구분하여 GIVI Renderer 라 명명한다. GIVI에서 렌더링 명령을 구분하여, 병렬 렌더링을 필요로 할 경우, GLORE 로 GIP Mgr에 명령을 전달하고 그 결과를 받는다. 또한 각 응용에 적합한 renderer를 모두 포함하며, 응용별 적절한 renderer를 제공해 준다.

6) GIP Mgr

GLOVE의 모든 대외 인터페이스를 담당하는 프로토콜인 GIP의 기능을 제공한다. GIP은 협업을 수행하는 연구자들에게 가시화 결과를 전달하고, GIVI와 GLORE 간의 통신을 담당한다.

7) Collaboration Mgr

Collaboration Mgr은 말 그대로, 협업을 위한 기능을 수행하는 컴포넌트이다. Collaboration Mgr의 가장 주요한 기능은 화상회의를 위한 화상과 소리의 전송이다. 이를 위해 3D 이미지를 생성하기 위해 좌우 카메라에서 찍은 이미지와 소리

를 전송하고, 이를 수신하여 서로 동기화 하여 출력한다. 또한, 가시화 데이터 공유를 위한 가시화 데이터 전송 제어 기능도 수행한다. 더하여, 협업을 위한 세션의 생성 및 관리, 만들어진 세션에 사용자를 추가하고, 삭제하는 사용자 관리 등의 기능을 제공한다.

8) GLORE

GLORE 는 전체 GLORE 의 기능을 제어한다.

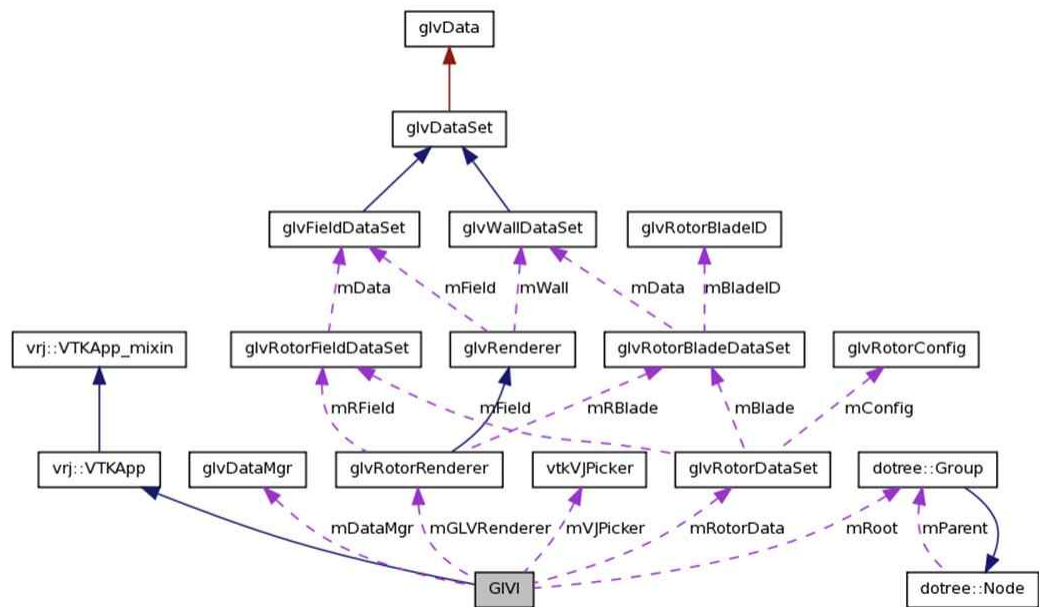
9) GLORE Renderer

GLORE 본연의 기능인 렌더링 기능을 수행한다. 대용량 데이터의 렌더링을 위해 설계되었으므로 병렬 렌더링의 데이터 분배 알고리즘과 결과의 병합 등 병렬 렌더링 매커니즘을 담당한다.

5. GLOVE 클래스 설계

4절까지 GLOVE 전체 시스템의 HW Architecture, SW Architecture 및 Component 구성까지 살펴보았다. 본 절에서는 이러한 기본 시스템 구조 하에 클래스 단계에서의 설계를 살펴본다.

가. Collaboration Diagram



[그림 4-1 GIVI Collaboration Diagram]

[그림 4-1]은 GLOVE Integrated Visualization Interface 인 GIVI의 Collaboration Diagram을 보여준다. Collaboration Diagram은 이들 클래스들의 구체적인 관계를 도식화 한 것이다. Collaboration diagram에 그려진 클래스들의 관계와 4절에서 언급한 기능단위의 SW Architecture와 상관관계는 다음과 같다.

1) GIVI

VRJuggler의 프레임워크 내에서 동작하는 GIVI는 vrj::VTKApp 클래스를 상속 받는다. glvDataMgr, glvRotorRenderer, glvRotorDataSet, dotree::Group의 클래스는 멤버 클래스들로 SW Architecture의 각 component 들과 대응된다.

2) vrj::VTKApp

vrj::VTKApp 클래스는 VRJuggler 의 프레임워크를 제공해 준다. VR환경의 사용자 입력 및 tiled-display 화면 출력 등을 자신의 프레임워크 내에서 관장한다.

3) dotree::Group

dotree 는 현재 화면 context를 관리하는 자료구조이다.

4) glvRotorDataSet

glvRotorDataSet은 유동장 데이터인 glvRotorFieldDataSet 과 glvRotorBladeDataSet, 그리고, 이들 데이터의 형태와 읽어들이 파일의 개수, 그 외 응용 특화된 패러미터들을 기술한 config 파일을 관리하는 glvRotorConfig 클래스를 멤버로 가진다. 즉 glvRotorDataSet은 로터 시뮬레이션에 특화된 Application Specific Data Management를 담당한다.

5) glvRotorRenderer

glvRotorRenderer는 GIVI에서의 렌더링을 담당한다. 렌더링을 가시화의 주기능이며, 대용량 데이터를 가시화하기 위해서는 여러 가지 기술들이 필요하다. glvRotorRenderer는 이러한 기술이 구현되어 있는 클래스이며, 병렬 렌더링이 필요할 경우 GLORE 와의 통신으로 병렬 렌더링을 제어하는 기능까지 수행한다. glvRotorRenderer는 GIVI Renderer component의 일부이며, glvRotorDataSet을 입력으로 가시화 결과를 출력하는 method들을 지원한다.

6) glvDataMgr

glvDataMgr은 데이터 로딩에 관련된 모든 제어를 제공하는 클래스이다. Lambda

RAM 제어를 비롯해 병렬 입출력, 분산 메모리 제어 등을 수행한다. 이를 수행함으로써, glvRotorRenderer가 렌더링을 위해 필요로 하는 형태의 데이터를 만들어 준다.

7) vtkVJPicker

vtkVJPicker는 vjVTK가 제공하는 클래스로 입력 도구가 화면에 보여진 특정 오브젝트를 선택하였을 때, 선택한 오브젝트가 무엇인지에 대한 정보를 파악하여 유지하는 클래스이다.

8) glvFieldDataSet

glvFieldDataSet은 CFD(Computational Fluid Dynamics)를 수행하는 응용들을 위한 유동장 데이터의 기본적인 데이터 구조를 제공한다. Application Common Data Management 영역에 해당하는 데이터 구조이다.

9) glvWallDataSet

glvFieldDataSet은 CFD(Computational Fluid Dynamics)를 수행하는 응용들을 위한 벽면 데이터의 기본적인 데이터 구조를 제공한다. Application Common Data Management 영역에 해당하는 데이터 구조이다.

10) glvDataSet

glvDataSet은 Structured Grid, Unstructured Grid 등의 가시화를 위한 기본적인 데이터와 시변환 데이터를 관리하기 위한 자료구조 및 method 들을 제공한다. 이는 Basic Data Management 영역에 해당한다.

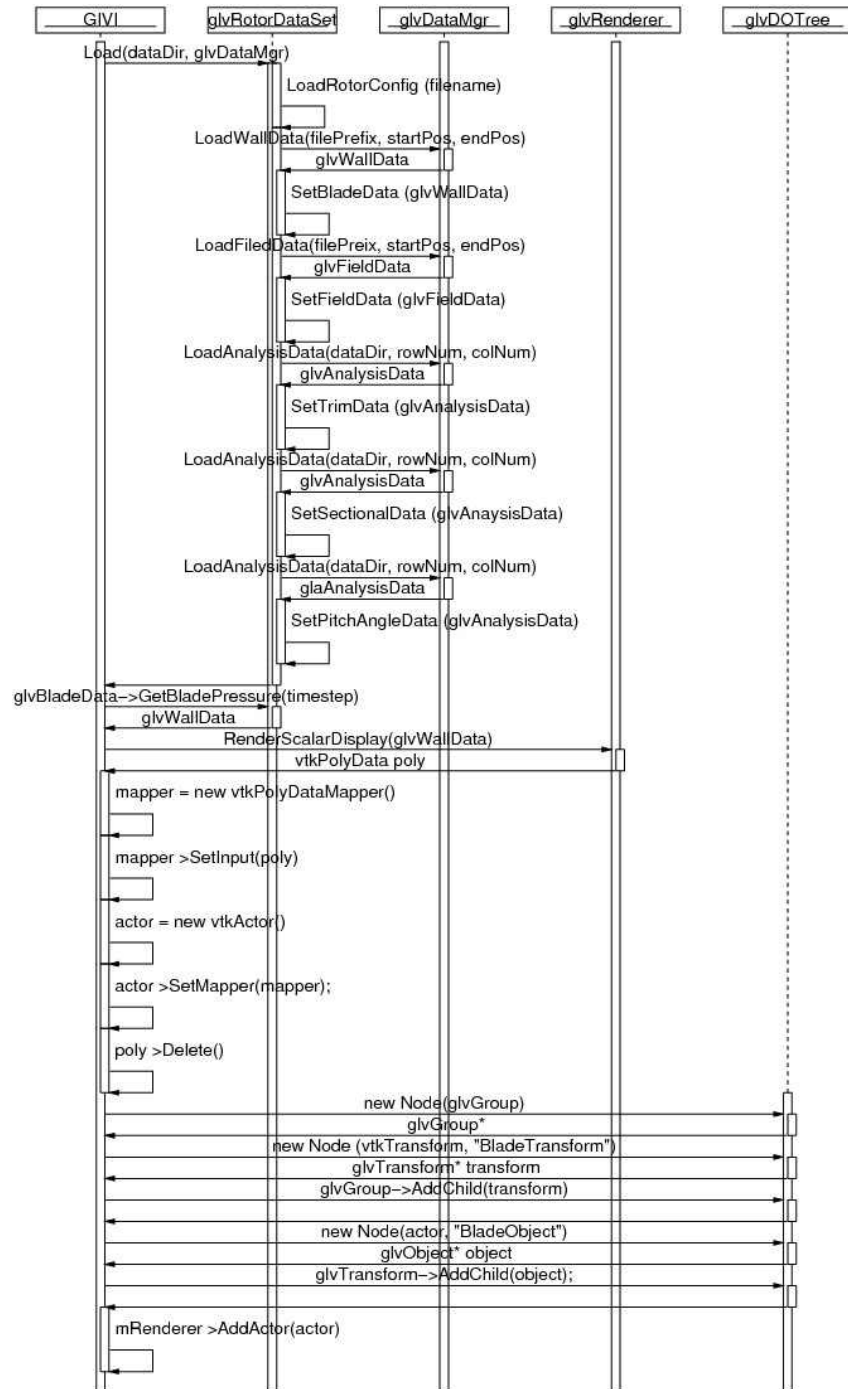
11) glvData

glvData 는 모든 데이터의 구조의 root로 시스템의 데이터 관리부분을 담당한다. 예를 들어, little endian 시스템에서 만들어진 데이터를 big endian 으로 변환하는 등의 method를 제공한다.

나. Sequence Diagram

[그림 4-2]는 각 클래스들의 동작 sequence를 보여 준다. Collaboration diagram에서 기술된 관계는 sequence diagram 에 의해 시간 순서로 명확하게 기술된

다. 이 sequence diagram은 VRJuggler의 프레임워크 내에서 동작한다. VRJuggler의 프레임워크는 [3]을 참조한다.



[그림 4-2] Pressure Distribution Visualization Sequence

GIVI 가 `glvRotorDataSet`을 Load 하면 `glvRotorDataSet`은 `glvDataMgr`에 `LoadRotorConfig`, `LoadFieldData`, `LoadWallData`, `LoadAnalysisData`함수를 호출 함으로써, 원하는 데이터를 얻는다. GIVI가 사용자 입력으로 로터 블레이드의 압력 분포를 가시화 하라는 명령을 입력 받으면, GIVI는 이미 로드된 데이터 중 `GetBladePressure` 함수를 호출하여 블레이드 데이터를 얻은 다음, 이를 `glvRender`에 요청해 렌더링된 `vtkPolyData`를 얻는다. GIVI 는 `vtkPolyData`를 가공하여 가시화 하기 위한 `vtkActor`를 만들고 이 `vtkActor`를 현재 context를 관리하는 `glvDOTree`에 추가 한 다음 `vjVTK`의 render 인 `vjRender`에 `vtkActor`를 추가한다. `VTKApp`를 상속받은 GIVI의 `draw()` function이 `VRJuggler`의 framework 내에서 호출되는 순간 `vjRenderer`인 `mRenderer` 내의 `vtkActor`들은 화면에 나타난다.

2절에서 정리한 로터 블레이드 시뮬레이션의 응용 특화된 기능들은 [그림 4-2]의 sequence diagram내에서, 데이터와 rendering 요청 함수만 바뀌어 모두 수행 가능하다.

6. 결론

GLOVE는 가상현실 환경 내에서 복잡한 대용량 데이터를 실시간으로 가시화하고, 그 결과를 원격지 사용자와 공유할 수 있는 통합 환경을 제공한다. 이와 유사한 목적으로 개발된 여러 가시화 도구들이 범용 사용자 환경을 제공하는 것과 달리 응용 맞춤형 사용자 환경을 제공한다. 이를 위해 GLOVE 는 어떤 응용이라도 가시화를 위해 적용할 수 있는 계층적 데이터 관리 구조를 가지며, 대용량 데이터를 효율적으로 처리하는 구조를 가진다. 또한 고해상도의 tiled display 와 VR 환경을 제공함으로써 응용 데이터의 해석을 용이하게 할 수 있다.

본 문서에서는 이를 위한 첫 번째 응용으로 로터 동역학 시뮬레이션 데이터의 가시화를 위한 GLOVE 프레임워크를 설계했다. 현재 GLOVE는 1단계의 구현을 마친 상태며, 향후 대용량 데이터의 처리를 위한 병렬 입출력과 분산 데이터 관리, 병렬 렌더링 기능을 추가로 구현할 예정이다. 궁극적으로는 슈퍼컴퓨터와 연계한 시뮬레이션의 실시간 제어 기능을 포함하는, 비주얼 슈퍼컴퓨팅(Visual Super Computing)을 가능하게 하는 통합 환경으로 확장해 나갈 예정이다.

7. 참고문헌

- [1] Venkatram Vishwanath, Robert Burns, Jason Leigh, Michael Seablom
"Accelerating tropical cyclone analysis using LambdaRAM, a distributed
cache over wide-area ultra-fast networks," Future Generation Computer
Systems, 2008
- [2] J. Jeong and F. Hussain, "On the Identification of a Vortex", Journal of
Fluid Mechanics, 285, pp.69-94, 1995
- [3] VRJuggler, <http://www.vrjuggler.org>
- [4] ParaView, <http://www.paraview.org>
- [5] COVISE, <http://www.hlr.de/covise>