

데이터 가시화를 위한 VR Juggler와 OpenSceneGraph를 이용한 통합 사용자 인터페이스

허영주
한국과학기술정보연구원(KISTI)
e-mail:popea@kisti.re.kr

An Integrated User Interface for Visualizing User Data Using VR Juggler and OpenSceneGraph

YoungJu Hur
Korea Institute of Science and Technology Information(KISTI)

요 약

컴퓨터에서 생성된 시뮬레이션 결과는 가시화(Visualization)라는 과정을 통해 인간이 직관적으로 해석하기 쉬운 형태로 가공된다. 사람이 직관적으로 이해하기 어려운 수치의 나열로 구성돼 있던 데이터가 컴퓨터 그래픽스 기술의 적용을 통해 보다 쉽게 분석할 수 있는 형태로 변형되는 것이다. 최근에는 이런 시뮬레이션 데이터의 크기와 복잡도가 점점 증가하고 있는 추세로, 이런 추세에 따라 고해상도 디스플레이 장치나 몰입형 가상현실 장치의 필요성은 점점 증가하고 있다. 이런 이유로 클러스터 시스템을 이용한 고해상도의 디스플레이 장치에서 해당 시스템에 대한 전문적인 지식을 갖추지 않은 일반 연구자가 상호작용할 수 있는 사용자 인터페이스에 대한 필요성도 높아지고 있다.

본 논문에서는 고해상도 디스플레이 장치에서 데이터를 가시화하는 프레임워크와 그 프레임워크에서 사용하는 통합 환경 인터페이스를 소개하기로 한다. 이 인터페이스는 VR Juggler와 OSG(OpenSceneGraph)를 기반으로 하며, 다양한 형태의 디스플레이 장치와 몰입형 가상현실 환경에서 시뮬레이션 데이터와 실시간 상호작용을 통한 해석을 수행하는데 필요한 기반 환경을 제공한다.

1. 서론

컴퓨터 시뮬레이션은 분자화학, 고에너지물리(High Energy Physics), 천체물리, 기계공학, 항공 우주공학 등의 다양한 응용 분야에 적용되고 있으며, 최근에는 고성능 컴퓨터(HPC, High Performance Computer)의 급격한 성능 향상에 힘입어 보다 정밀하고 규모가 큰 시뮬레이션 실험이 수행되고 있다. 이로 인해 대규모 시뮬레이션의 결과로 만들어지는 데이터의 크기가 방대해지고 구조도 매우 복잡해지는 추세다. 따라서 이런 데이터를 분석하고 해석하는 데도 점차 고성능 컴퓨터를 이용하는 경우가 늘어나고 있다.

데이터를 해석하는 방법은 여러 가지가 있을 수 있는데, 그중 시뮬레이션 데이터에 컴퓨터 그래픽스 기술을 적용한 과학 데이터 가시화(Scientific Visualization) 방식은 직관적으로 데이터를 해석하는데 매우 유용한 방식으로 꼽히고 있다. 앞서 설명대로 데이터의 크기와 복잡도가 증가함에 따라 데이터 가시화 기법뿐만 아니라 가시화된 결과물 역시 매우 복잡해지고 있는데, 이렇게 복잡해진 가시화 데이터를 분석하는데 있어 기존 장비는 적합치 않으며, 데이터 수준에 상응하는 고해상도 디스플레이 장치나 몰입형 가상현실 장치의 사용은 필연적이다. 그러나 이런 장

치를 제어하는 데는 해당 시스템에 대한 전문적인 지식이 요구되며, 이런 이유로 응용 과학자가 자신의 대용량 데이터를 스스로 가시화할 수 있는 경우는 극히 드물다. 따라서 이런 장비에 대한 전문적인 지식을 갖추지 않은 일반 연구자가 상호작용을 통해 자신의 데이터를 가공하고 가시화할 수 있는 프레임워크에 대한 필요성이 대두되고 있다.

본 논문에서는 이런 고해상도 디스플레이 장치에서 사용자 데이터를 가공하고 가시화할 수 있게 해주는 프레임워크에 대해 소개한다. 이 프레임워크는 고해상도 디스플레이 장치를 포함하는 통합 환경 내에서 데이터를 정성적, 정량적으로 분석할 수 있게 해주며, 다양한 형태의 디스플레이 장치, 혹은 몰입형 가상현실 환경에 이르기까지의 다양한 형태의 가시화 장비를 지원한다. 또, 기본 프레임 위에 각 응용분야에 특화시킨 인터페이스를 제공할 수 있으므로 다양한 응용분야로의 확장이 가능하다. 이 프레임워크에서 사용하는 인터페이스는 VR Juggler와 OSG(OpenSceneGraph)를 기반으로 하며, 다양한 가시화 환경에서 데이터를 실시간으로 상호작용을 통해 분석하는데 필요한 기반환경을 제공한다.

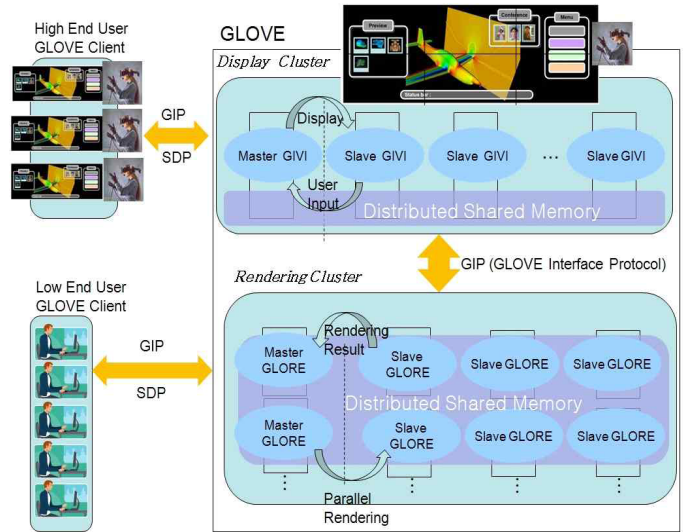
2. 관련 연구

다양한 응용분야의 시뮬레이션 결과를 가시화할 수 있는 응용 프로그램은 매우 다양하다. 이런 응용프로그램들 중 대표적으로 많이 사용되는 것으로 COVISE([7])나 ParaView([12])를 들 수 있다. COVISE는 병렬처리를 통해 대용량 데이터를 가시화할 수 있으며 가상현실을 포함한 다양한 기능을 지원하며, 현재 자동차 설계 분야에서 널리 사용되고 있는 응용 프로그램이다. ParaView는 대용량 데이터를 처리할 수 있는 병렬 처리 능력과 범용성을 가장 큰 특징으로 들 수 있다. ViSTA[10]는 수치 시뮬레이션, 구조, 약학 등의 분야를 주 타겟으로 하며, 역시 병렬처리 기능을 제공한다. VisIT([8],[14])은 테라급 이상의 대용량 데이터를 가시화할 수 있으며, 다양한 가시화 기능을 제공한다. 이런 프로그램들은 대용량 데이터 처리를 포함하는 다양한 기능을 제공하지만, 특정 분야에 특화된 기능을 제공하기 때문에 응용 프로그램이 지원하지 않는 다른 분야의 연구자들이 직관적으로 사용하기에는 어려움이 따를 수 있으며, 일반 사용자들이 대용량 데이터를 가시화하는데 있어 병렬처리 등에 어려움을 느낄 수 있다.

시뮬레이션 데이터를 가시화할 수 있는 또다른 방법은 가시화 알고리즘을 지원하는 소프트웨어를 이용해서 응용 용도에 맞게 프로그램을 제작하는 것이다. VTK([16])는 과학 데이터 가시화에 많이 사용되는 공개 소프트웨어 라이브러리로, 다양한 분야의 데이터를 가시화하는데 널리 사용되고 있다. 앞서 언급한 ParaView나 ViSTA, VisIT은 VTK를 사용해서 제작됐다. 그러나 VTK는 고해상도 디스플레이 장치나 가상현실 환경 등은 지원하지 않으므로 VTK가 이런 환경을 지원할 수 있게 해주는 미들웨어들이 개발돼 왔다. 이런 미들웨어로는 CaveLib([6])이나 VR Juggler([5], [15]) 등을 들 수 있다. [13]에서는 VTK를 이용한 3차원 가상현실 인터페이스인 VR-VTK를 소개하고 있다. VR-VTK는 손, 발, 머리, 음성 등의 다양한 인터페이스를 통해 3차원 가상현실 인터페이스에 대한 다양한 구현을 시도했다. vjVTK([9])는 VTK의 다양한 가시화 알고리즘을 그대로 사용할 수 있으면서 출력부분을 VR Juggler 와 연계해서 다양한 디바이스를 지원할 수 있게 해준다. 그러나 vjVTK의 경우에는 VR Juggler에서 VTK를 지원하기 위한 자체 렌더링을 사용함으로써 대용량 데이터에 대한 렌더링 성능이 저하되는 경향이 있다.

OSG(OpenSceneGraph, [11])는 씬 그래프(scene graph)를 기반 구조로 하는 그래픽스 툴킷으로 뛰어난 성능의 다양한 그래픽스 알고리즘을 제공하므로 널리 사용되고 있으며, VR Juggler와의 연계도 가능하다.

본 논문에서는 VR Juggler와 OSG를 기반으로 하는 인터페이스 프레임워크를 소개한다. 이 프레임워크는 VR Juggler를 미들웨어로 사용함으로써 다양한 형태의 디스플레이 장치와 다양한 디바이스를 지원할 수 있고, OSG의 뛰어난 렌더링 성능을 통해 vjVTK의 단점을 극복하면서 가시화된 데이터에 적절한 인터페이스를 제공한다.



(그림 1) GLOVE 프레임워크

3. GLOVE 프레임워크

GLOVE(GLOBAL Virtual Environment for research)는 통합환경 내에서 과학 데이터를 정성적, 정량적으로 분석할 수 있게 해주는 가시화 프레임워크로 단일 모니터에서 타일형 디스플레이 장치, 혹은 몰입형 가상현실 환경에 이르기까지 다양한 형태의 가시화 장비를 지원하며, 병렬 처리를 통한 대용량 데이터의 가시화가 가능하다. GLOVE는 가시화 알고리즘을 수행하는 GLORE와 통합 인터페이스를 제공하고 가시화 데이터를 렌더링해서 사용자에게 보여주는 GIVI 부분으로 나눌 수 있다. GIVI는 사용자 인터페이스를 총괄하는 부분으로 가상현실 입/출력 장치를 포함한 다양한 입/출력장치로부터 사용자 입력을 받아서 GLORE에 전달함으로써 GLORE가 가시화 작업을 수행하도록 한다. GLORE는 대용량 데이터를 가공하는 엔진으로 GIVI의 요청을 받아서 사용자가 원하는 형태로 데이터를 가공한 뒤, 그 결과를 GIVI로 전송한다. 이 때, GLORE는 VTK의 다양한 가시화 알고리즘을 사용하여 가시화 데이터를 생성하며, 이렇게 생성된 VTK 오브젝트를 GIVI로 전송한다. GIVI는 VTK 오브젝트들을 받아서 OSG 오브젝트로 변환한 뒤 렌더링 작업을 수행, 그 결과를 출력 디바이스로 내보내게 된다. 이렇게 동작하는 GLOVE의 전체 프레임워크는 (그림 1)과 같다. GLORE와 GIVI간의 데이터 통신은 GIP(GLOVE Interface Protocol)을 통해 수행되는데, 이 프로토콜은 비동기적으로 통신을 수행한다. GLOVE를 이용하면 사용자는 비교적 손쉽게 복잡한 병렬 처리 가시화 작업을 수행할 수 있으며, 이에 대한 인터페이스는 GIVI가 담당하게 된다.

GIVI는 VR Juggler의 동작 매커니즘에 따라 마스터 프로세스와, 마스터와 동일하게 동작하는 슬레이브 프로세스로 구성된다. 여러 대의 컴퓨터와 디스플레이 장치로 구성된 대형 가시화 시스템에서 동작할 경우에는 서로 다른



(그림 2) OSG 렌더링 엔진 구조의 인터페이스 프레임워크

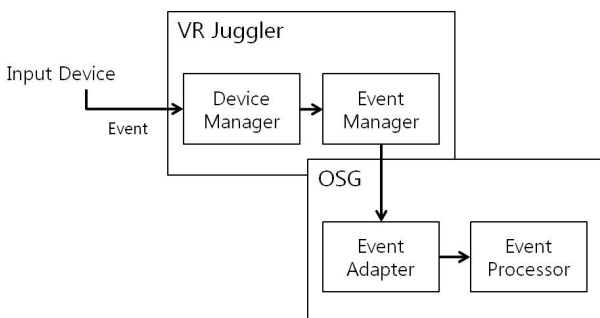
GIVI 슬레이브 프로세스가 각 디스플레이 장치에 대한 렌더링을 담당하게 된다.

3.1. 인터페이스 프레임워크 구조

OSG 렌더링 엔진 구조와 VR Juggler를 연계한 GIVI 인터페이스 프레임워크는 (그림 2)와 같다.

GIVI는 고해상도 디스플레이나 가상현실 환경 등, 다양한 형태의 입/출력 디바이스를 지원하기 위해 VR Juggler를 미들웨어로 사용하며, 실제적인 사용자 인터페이스와 데이터 가시화를 OSG를 통해 수행한다. OSG 내 부적으로는 사용자와의 상호 작용을 오브젝트에 대한 콜백 메커니즘을 통해 제공하며, 이런 콜백은 키보드 혹은 마우스에 대한 동작을 통해 구현할 수 있게 되었다. 즉, 특정 오브젝트에 대해 마우스나 키보드 동작에 대해 이벤트를 발생시키고, 그 오브젝트의 해당 이벤트에 대한 콜백 함수를 호출함으로써 사용자와 상호작용이 가능한 것이다. 반면 VR Juggler는 다양한 입력 디바이스에 대한 입력값을 제공한다. 이런 입력값은 wand나 head tracker 등과 같은 가상현실 디바이스에 대한 좌표값이나 해당 디바이스에 대해 사용자가 발생시킨 이벤트에 대한 디지털 입력값의 형태로 전달된다. 따라서 이런 VR Juggler상의 디바이스 입력값을 OSG상의 이벤트로 전환하는 작업이 수행되어야 한다. 즉, 가상현실 디바이스의 입력 디바이스에 대한 데이터를 받아들여서 이를 디바이스상의 이벤트로 인지한 뒤, OSG가 인지할 수 있는 이벤트로 분류, 처리함으로써 OSG의 이벤트 처리 루틴으로 전환하는 과정이 필요한데, 이를 수행하는 과정은 (그림 3)에서 볼 수 있다.

Device Manager는 VR Juggler와 물리적인 입력 디바이스를 연결하는 부분으로, Device Manager를 통해 입력



(그림 3) GIVI의 이벤트 처리 루틴

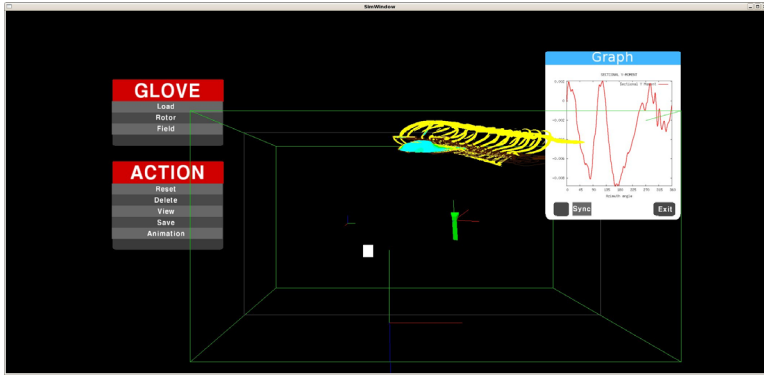
디바이스의 값과 이벤트를 받아들인다. 이렇게 받아들여진 입력값은 VR UI 부분의 Event Manager를 통해 OSG의 Event Adapter로 전달되며, Event Adapter로 전달된 이벤트는 Event Processor에 의해 이벤트에 해당하는 동작을 수행하게 된다. 이에 대한 상세 과정은 다음과 같다..

- VR UI 부분의 Event Manager는 VR Juggler의 Device Manager로부터 입력 디바이스에 대한 데이터를 받아들인다. 이 값은 VR Juggler 환경 내에서의 디바이스의 위치에 대한 좌표값과 디바이스에 대해 사용자가 발생시킨 이벤트에 대한 입력값을 모두 포함한다.
- 입력 디바이스의 위치와 방향으로부터 디바이스가 가리키는 영역에 대한 정보를 계산할 수 있다. 디바이스가 가리키는 영역은 다양한 형태로 계산할 수 있는데, 일반적으로는 디바이스의 위치로부터 사용자가 바라보는 방향으로 뻗어나가는 직선의 형태를 많이 사용한다. 이 때 이 영역에 대한 계산은 OSG 좌표계에서 이뤄지게 된다.
- 디바이스가 가리키는 영역 내에 OSG 오브젝트가 존재하는지 여부를 조사한다.
- 디바이스 영역 내에 OSG 오브젝트가 존재할 경우, 디바이스 내에서 이벤트가 발생했는지 여부를 체크한다. 만약 이벤트가 발생했을 경우에는 해당 이벤트에 대해 발생시키고 싶은 OSG의 Event Adapter를 대응시킴으로써 OSG의 이벤트 루틴을 호출할 수 있다. 즉, 입력 디바이스로부터 입력되는 디지털 값의 상태 변화에 따라 감지되는 이벤트를 OSG의 EventAdapter상의 이벤트로 매핑시킴으로써 OSG 루틴으로 이벤트를 전달할 수 있는 것이다.
- 이렇게 이벤트가 발생한 오브젝트와 OSG상의 이벤트 종류를 동시에 전달함으로써 OSG 오브젝트상에서 콜백 루틴을 수행할 수 있다. VR Juggler는 preframe 단계에서 이벤트가 발생했는지 여부를 체크해서 수행하며, update 단계에서 이벤트에 대한 콜백(즉, event processor)을 수행한다.

3.2. GIVI 인터페이스 환경

GIVI는 3.1에서 설명한 인터페이스 프레임워크와 이벤트 처리 루틴을 이용한 통합 인터페이스 환경을 제공한다. 이 인터페이스 환경은 다양한 이벤트를 구성할 수 있는 기반 환경을 제공하는데, 각종 사용자 요구사항을 처리하는데 필요한 인터페이스와 사용자가 데이터의 기하 정보를 이용, 데이터상에서 직접 조작하는데 필요한 인터페이스를 모두 포함한다.

GIVI의 VRUI(그림 2)는 과학 데이터 가시화에 필요한 다양한 툴과 위젯(widget)을 제공한다. 이런 위젯 중에는 사용자가 발생시킨 이벤트를 콜백 함수와 연계시키는 위젯도 있고, 메뉴를 구성하는데 필요한 위젯들과 그래프를 보여주는 그래프 패널, 값의 범위 등을 입력하는데 사용되는 숫자 패널 등의 위젯도 있으며, 3차원적으로 가시화된



(그림 4) OSG와 VR Juggler를 이용한 통합 인터페이스 환경

데이터 오브젝트와 직접 상호작용할 수 있게 해주는 wand, plane, point 등의 위젯도 지원한다.

VR Juggler와 OSG를 이용한 GIVI의 통합 인터페이스 환경은 (그림 4)에서 볼 수 있다. (그림 4)는 로터 동역학 데이터에 대한 가시화 환경을 보여주며, 데이터에 대한 isosurface와 정량적 분석값인 그래프를 가시화한 결과를 보여준다. 사용자는 다양한 입력 디바이스를 이용해서 각종 파라미터를 조절해서 가시화 수위를 조절할 수도 있으며, 오브젝트에 대해 transform, rotation 등의 상호작용을 수행함으로써 원하는 위치에서 오브젝트를 해석할 수 있다.

4. 결론 및 향후 계획

고해상도 디스플레이 장치나 몰입형 가상현실 환경 등의 다양한 첨단장비를 이용해서 데이터를 가시화하면 복잡한 데이터를 보다 직관적으로 이해하고 분석하는 것이 가능해진다.

GLOVE의 통합 인터페이스 환경은 VR Juggler와 OSG를 기반으로 다양한 가시화 환경에서 데이터와 실시간으로 상호작용하며 분석하는데 필요한 기반환경을 제공하며, 병렬처리를 통한 대용량 데이터 가시화를 지원한다. GLOVE는 이런 기반 환경을 제공하기 위해 다양한 인터페이스 프레임워크를 지원하며, 물리적으로 다양한 입력 디바이스로부터 받은 입력을 VR Juggler와 VR UI 레이어를 통해 OSG에 대한 입력으로 전환함으로써 사용자와의 물리적인 상호작용을 지원한다. 또, 다양한 VR UI를 제공함으로써 사용자가 데이터를 해석하는데 필요한 인터페이스 환경을 제공하고 있으며, 해석 데이터와 통계 데이터를 한눈에 비교, 분석하는 기능도 지원한다. 향후에는 보다 다양한 형태의 인터페이스를 제공함으로써 사용자에게 다양한 방법으로 데이터를 탐색할 수 있는 길을 열어 줄 것이며, 여러 분야의 데이터를 보다 효율적으로 탐색할 수 있는 고성능의 인터페이스의 개발을 계속 진행할 것이다.

참고문헌

[1] S.Byron, "Virtual Reality in Scientific Visualization",

Communications of the ACM, 39(5):62-71, 1996.

[2] A.van Dam, A.S.Forsberg, D.H.Laidlaw, J.J.Laviola, R.M.Simpson, "Immersive VR for Scientific Visualization: A Progress Report", IEEE Computer Graphics and Applications, 2000.

[3] Andrew Forsberg, Prabhat, Graff Haley, Andrew Bragdon, Joseph Levy, Caleb I.Fassett, David Shean, James W.Head III, Sarah Milkovich, Mark Duchaineau, "Adviser: Immersive Field Work for Planetary Geoscientists", IEEE Computer Graphics and Applications, July 2006.

[4] Arjan J.F.Kok & Robert van Liere, "A Multimodal Virtual Reality Interface for 3D Interaction with VTK", Knowledge and Information Systems, vol 11(3), 2007.

[5] A.Bierbaum, C.Just, P.Hartling, K.Meinert, A.Baker, C.Cruz-Neira, "VR Juggler: A Virtual Platform for Virtual Reality Application Development", Proceedings of IEEE Virtual Reality, 2001.

[6] CaveLib, http://www.mechdyne.com/integrated_solutions/software/products/CAVELib/CAVELib.html

[7] COVISE, <http://www.hlrs.de/organization/av/vis/covise>

[8] H.R.Childs, E.S.Brugger, K.S.Bonnel, J.S.Meredith, M.C.Miller, B.J.Whitlock, N.L.Max, "A Contract-Based System for Large Data Visualization", Proceedings of IEEE Visualization, 2005.

[9] K.J.Blom, "vjVTK: A toolkit for interactive visualization in virtual reality", Eurographics Symposium on Virtual Environment, 2006.

[10] M.Schirki, A.Germdt, T.van Reimersdahl, T.Kuhlen, P.Adorneit, O.Lang, S.Pischinger, C.Bischof, "Vista FlowLib - A Framework for Interactive Visualization and Exploration of Unsteady Flows in Virtual Environments", Eurographics Workshop on Virtual Environments, 2003.

[11] OpenSceneGraph, <http://www.osg.org>

[12] ParaView, <http://paraview.org/New/index.html>

[13] Thomas van Reimersdahl, Torsten Kuhlen, Andreas Gerndt, Jorg Henrichs, & Christian Bischof, "ViSTA: A Multimodal, Platform-independent VR-Toolkit based on WTK, VTK, and MPI", Proceedings of 4th International Immersive Projection Technology Workshop, 2000.

[14] VisIt, <http://wci.linl.gov/codes/visit>

[15] VR Juggler, <http://www.vrjuggler.org>

[16] VTK, <http://www.vtk.org>