

ParaView-VR: 가상현실(VR) 환경에서의 과학 데이터 가시화를 위한 프레임워크

히영주, 금복희, 구기범, 이증연
한국과학기술정보연구원(KISTI)
e-mail:popea@kisti.re.kr

A Framework for Visualization and Exploration of Scientific Data in Virtual Environments

Youngju Hur, Bokhee Keum, Geebum Koo, Joongyoum Lee
Korea Institute of Science and Technology Information

요 약

과학 데이터는 대부분 과학 시뮬레이션의 결과로 얻게 되며, 사람이 직관적으로 이해하기 어려운 숫자의 나열인 경우가 많다. 이런 수치 데이터를 가시화하면 보다 쉽게 이해해서 분석할 수 있는 형태가 갖춰지게 되며, 가시화 환경을 VR 환경으로 옮기면 사용자와 상호작용하면서 사용자가 원하는 형태로 데이터를 가시화하는 것이 가능할 뿐만 아니라 몰입형 환경 덕분에 데이터에 대한 이해도도 높일 수 있다. 그러나 VR 환경의 시스템적 한계와 다양한 가시화 알고리즘에 대한 필요성 때문에 VR 환경에서 과학 데이터를 가시화하는 데는 한계가 있다.

본 논문에서는 VR 환경에서 과학 데이터를 가시화하는 프레임워크를 제안한다. 이 프레임워크는 VTK에 기반을 둔 애플리케이션인 ParaView를 활용함으로써 과학 데이터를 가시화하는 알고리즘을 제공하며, VTK를 이용해서 보다 다양한 방식으로 과학 데이터를 분석하는 툴을 제공한다. VR과의 인터페이스로는 VTK와 CAVELib을 기반으로 하는 VtkCave를 활용해서 시스템에 맞는 VR 인터페이스를 제공한다. 향후에는 이 프레임워크에 보다 다양한 인터페이스와 가시화 기법을 더해서 풍부한 서비스를 제공할 수 있도록 할 예정이다.

1. 서론

과학 시뮬레이션의 결과로 얻어지는 수치 데이터를 가시화하는 방법은 매우 다양하다. 이렇게 수치 데이터를 가시화하면 사람이 직관적으로 쉽게 이해하고 분석할 수 있는 형태가 갖춰지는데, VR 환경에서 가시화하면 사용자와 상호작용하면서 사용자가 원하는 형태로 데이터를 가시화하는 것이 가능하다.

이런 과학 데이터를 가시화하는 방법은 매우 다양하며, 때로는 복잡한 가시화 알고리즘을 요구하기도 한다. 이런 복잡한 가시화 알고리즘은 VTK(Visualization Toolkit)을 이용해서 해결할 수 있다.

VTK는 3차원 컴퓨터 그래픽스, 이미지 처리 및 가시화에 주로 사용되는 공개 소프트웨어 라이브러리로 매우 다양한 기능을 제공한다. VTK는 스칼라, 벡터, 텐서, 텍스처 및 볼륨 데이터를 표현할 수 있는 자료구조뿐만 아니라 다양한 고급 모델링/렌더링 기법을 제공함으로써 과학 데이터 가시화에 매우 널리 사용되고 있다([7],[12]). 이런 VTK를 VR 환경에 접목시킬 수 있다면 VR 환경에서 사용자와 상호 작용을 통해 다양한 방식으로 데이터를 표현하고 이해하는 것이 가능해질 것이다.

본 논문에서는 VR 환경에서 과학 데이터를 가시화하

는 프레임워크를 제안한다. 이 프레임워크는 VTK에 기반을 둔 애플리케이션인 ParaView를 활용함으로써 과학 데이터를 가시화하는 알고리즘을 제공하며, VTK를 이용해서 보다 다양한 방식으로 과학 데이터를 분석하는 툴을 제공한다. 원칙적으로 VR 환경을 지원하지 않는 VTK와 VR 환경의 인터페이스로는 VtkCave를 활용해서 Vtk-VR 라이브러리를 구현해서 사용했다.

이 프레임워크에서는 기본적으로 다중 뷰(multi-view) 기능과 폴리곤 데이터를 wireframe, point 로 표현하는 기능, Clip, Slice, Probe 및 Zoom 기능을 제공한다.

2. 관련 연구

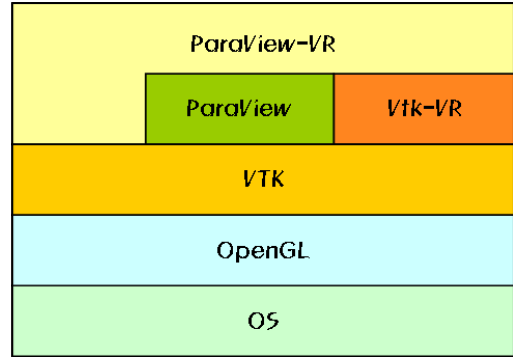
VTK는 방대한 가시화 알고리즘을 제공하는 공개소스 소프트웨어 라이브러리로, 주로 과학 데이터를 가시화하는데 널리 이용되고 있다([7],[12]). 그러나 VTK는 현재 입체 가시화 기능만 지원할 뿐, VR의 핵심이라 할 수 있는 트래킹이나 3차원 인터랙션과 관련된 기능을 지원하지 않는다. 이에, VTK의 다양한 알고리즘을 VR 환경에서 이용하려는 시도는 지속적으로 이뤄져 왔다.

VTK에 VR을 접목하는 방법은 크게 2가지로 나뉘볼 수 있다[6]. 우선 VTK의 액터(actor)를 VR 환경에서 주

로 사용하는 씬 그래프(scene graph)로 변환하는 방식이 있다. VTK는 기하정보를 액터라는 형태의 데이터구조로 생성해서 렌더링하는데, 이런 액터를 씬 그래프로 변환하는 것이다. 이 방식은 복잡한 필터링 알고리즘과 렌더링 알고리즘이 분리되기 때문에, 렌더링이 필터링 알고리즘에 의해 방해받지 않을 수 있다는 장점을 가진다. 그러나 VTK 파이프라인과의 직접적인 인터랙션이 불가능하다는 한계가 있다. 이 방식을 사용하는 대표적인 툴로는 vtkActorToPF([9])와 VTK2CAVE를 들 수 있다. vtkActorToPF는 VTK의 액터를 Performer의 씬 그래프의 한 노드인 pfGeode 형태로 변환하는 역할을 수행한다. 즉, vtkActorToPF의 사용자는 VTK를 이용해서 기하 정보를 추출하고 Iris Performer를 이용해서 그 정보를 렌더링하는 것이다. 이 때, 씬 그래프는 VTK와는 독립적으로 렌더링되며, 여기에서 VR 환경을 위한 다중 채널 렌더링의 가능성이 생긴다. 반면, VTK2CAVE는 변환 프로세스의 말단에서 순수 OpenGL 명령어를 CAVE 라이브러리의 특정 프레임워크에 맞는 형태로 다시 생성한다. VTK2CAVE의 장점은 매우 안정적이고 사용하기 쉽다는 것이다([6]).

또다른 방식은 VTK의 렌더러(renderer)를 VR환경에 알맞게 변경하는 것이다. 이 방식은 VTK에서 VR 기능이 지원되지 않는 vtkRenderer 클래스와 vtkRenderWindow 클래스를 VR 기능을 지원하는 클래스로 교체하는 것으로, VTK의 거의 모든 기능을 그대로 사용할 수 있을 뿐만 아니라 구현하기도 그다지 어렵지 않다는 장점을 가진다. VtkCave([8])나 VR-VTK([1]), ViSTA([5],[6]) 등은 이 방식을 사용한다. VtkCave는 VTK와 CaveLib을 사용하는데, VTK 모델링 스타일의 모델링 기법, 3차원 가상 메뉴와 CAVE 스타일의 상호작용을 제공한다. VtkCave는 VTK 라이브러리에 추가 가능한 한 요소로 취급되기 때문에 사용자는 별도로 OS 나 디스플레이 시스템, 디바이스-레벨의 프로그래밍 기법에 신경쓰지 않고 애플리케이션에 VR 기능을 덧붙일 수 있으며, 비교적 구현이 간단하고 유연성이 높은 라이브러리다. VR-VTK는 VTK에 multimodal VR 인터페이스를 덧붙인 것으로, 손으로는 위젯과 픽터를 이용한 3차원 조작을, 발로는 클러칭을, 머리로는 카메라 조작을, 그리고 말로 시스템을 조절할 수 있게 해주는 인터페이스다. ViSTA는 수치 시뮬레이션, 가상 프로토타이핑, 구조, 약학 및 심리학 분야에 대한 가시화를 주 타겟으로 하는 라이브러리로, VtkCave를 기반으로 VR 기능을 제공한다.

VTK는 다양한 알고리즘을 제공하는 유용한 툴이긴 하지만, 실제적으로 이용하는 사용자들이 프로그래밍 기법을 익혀야 한다는 단점이 있다. ParaView는 Kitware사와 Los Alamos National Laboratory는 공동 프로젝트를 통해 개발된 애플리케이션으로, VTK를 이용한 병렬 가시화 애플리케이션이다. ParaView는 다양한 데이터 셋을 가시화하기 위해 설계된 애플리케이션으로, 데이터 처리와



(그림 1) ParaView-VR의 구조

렌더링에 VTK를 이용하며, 병렬 기능을 제공한다[11].

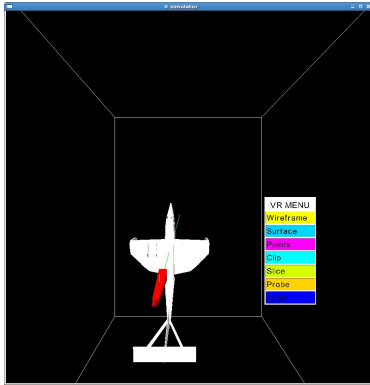
본 논문에서는 VTK와 ParaView를 이용해서 VR 환경에서 과학 데이터를 가시화하는 프레임워크를 제안한다. VR 환경과의 인터페이스를 위해서는 VtkCave를 기반으로 Vtk-VR이라는 라이브러리를 구현했으며, VTK의 다양한 필터 기능과 가시화 기법을 VR 환경에서 사용할 수 있게 했다. 이 프레임워크는 ParaView에 VR 인터페이스를 제공함으로써 ParaView에서 가시화한 결과를 그대로 VR 환경으로 옮겨서 탐색할 수 있게 할 뿐만 아니라, VR 인터페이스 자체에서 제공하는 여러 필터 기능을 통해 데이터를 직접 조작해서 가시화한 결과를 서로 비교하는 것도 가능하다.

3. ParaView-VR

ParaView-VR의 구조는 (그림 1)과 같다. ParaView-VR은 ParaView에서 각종 필터를 적용한 결과를 그대로 VR 환경으로 가져와서 VtkCave를 기반으로 트래킹 및 인터랙션과 관련된 VR 기능을 제공하는 Vtk-VR을 사용, VR 환경에서 과학 데이터를 가시화하는 프레임워크를 제공한다. VR 환경에서는 데이터를 탐색하는 기능 외에도 폴리곤 데이터를 Wireframe, Surface, Point의 다양한 형태로 보는 기능과 Clip, Slice, Probe, Zoom과 같은 기능을 제공하는데, 이런 기능은 VTK의 필터 기능을 이용해서 제공한다. 즉, 기본적인 필터링 작업을 ParaView에서 수행한 뒤, 그 수행 결과를 ParaView 데이터 형식으로 저장한 다음, ParaView-VR로 VR 환경에서 가시화하는 것이다. 그런 다음에는 ParaView-VR의 기능을 이용, 가시화된 데이터를 탐색하거나 원하는 형태로 변경, 비교분석 하는 것이 가능하다.

3.1. VTK-VR

ParaView-VR은 Vtk-VR이라는 인터페이스를 이용해서 VTK 기반으로 이뤄진 ParaView 데이터를 VR 환경에 디스플레이한다. Vtk-VR은 VtkCave를 기반으로 VR과 관련된 트래킹이나 상호작용을 제공한다. 그리고, VTK와의

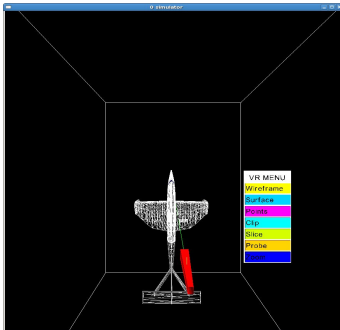


(그림 2) ParaView-VR의 메뉴 인터페이스

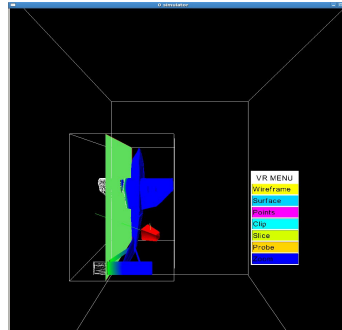
호환성을 염두에 두고 구현했기 때문에 대부분의 VTK 기능을 VR 환경에 적용할 수 있다.

VtkCave는 유연성이 높은 유용한 라이브러리가긴 하지만, VTK의 이전 버전을 대상으로 구현했기 때문에 ParaView가 기반을 두고 있는 VTK의 현 버전과는 잘 호환되지 않는다는 단점이 있다. 게다가 현재는 개발이 중단된 상태여서 향후 개발을 통한 개선의 여지도 거의 없다고 볼 수 있다. 이에, VtkCave를 VTK 현 버전과의 호환성을 고려해서 여러 기능을 수정하거나 추가, 제거함으로써 현재 VTK 버전과 가상화시스템에서 활용할 수 있는 Vtk-VR을 구현했다.

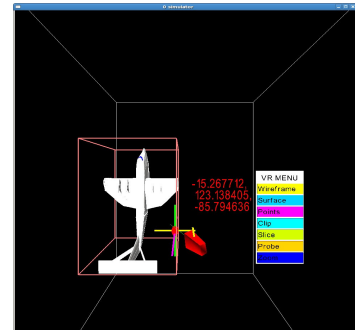
VtkCave에서 VTK의 현 버전과 주로 문제가 되는 부분은 데이터 타입과 콜백 함수의 호출 부분이다. 따라서 이 부분을 수정하면 일단 VTK 현 버전과의 호환 문제 중 가장 큰 부분을 해결할 수 있다. 거기에 본 연구소에서 보유한 시스템과 문제를 일으키는 vtkCaveCamera 클래스와 렌더링의 핵심이라고 볼 수 있는 vtkCaveActor 클래스를 시스템에 맞게 수정하고 vtkCaveMenu 및 vtkCaveMenuButton 클래스를 수정한 뒤, 메뉴에 좀 더 변화를 줄 수 있는 기능을 추가했다. 또, 사용자와의 상호작용을 담당하는 vtkCaveInteractor 클래스와 vtkCaveRenderWindowInteractor에도 여러 기능을 추가함



(그림 4) Wireframe 모드



(그림 3) Clip



(그림 5) Probe

으로써 시스템에서 wand를 사용할 수 있게 했을 뿐만 아니라 보다 다양한 방식의 상호작용이 가능케 했다.

3.2. 사용자 인터페이스

ParView-VR은 VR 환경에 필수적인 트랙커 중심의 인터페이스를 제공하며, 사용자 입력은 wand를 통해 받는다. 이 때, 사용자는 wand의 중심에서 나오는 레이를 통해 wand가 현재 가리키는 위치를 파악할 수 있다.

오브젝트의 선택은 wand를 통해 이뤄지는데, wand의 레이를 오브젝트로 향한 채로 wand의 2번째 버튼을 누르면 오브젝트가 선택된다. 선택된 오브젝트로써는 현재 모드에 따라 이동(translation), 회전(rotation), 크기조절(scaling)의 동작을 수행할 수 있다. wand의 레이를 오브젝트에 향한 채로 버튼을 누르면 Vtk-VR에서 오브젝트에 대한 Pick 이벤트가 발생한다. 이 이벤트가 발생하면 wand의 2번째 버튼을 누르는 동안에는 원하는대로 오브젝트를 이동하거나 회전, 혹은 크기조절이 가능하다. 이 때, 2번째 버튼이 아니라 첫번째 버튼을 누르면 오브젝트의 움직임에 대한 모드를 변경할 수 있다. 오브젝트의 이동 모드는 이동, 회전, 크기조절 순으로 바뀌게 되며, 향후에는 모드가 바뀔 때마다 커서가 바뀌게 하는 기능도 추가할 예정이다. 또, 회전 모드로 전환했을 경우에는 오브젝트에 회전축을 추가해서 회전 상태를 보다 명확하게 구현하는 기능을 추가하는 것도 고려중이다.

ParaView-VR은 오브젝트에 대한 움직임을 통해 오브젝트를 제어하는 것 외에도 메뉴 인터페이스(그림 2)를 제공한다. 이 메뉴 인터페이스를 이용하면 보다 다양한 가상화 효과를 오브젝트에 적용할 수 있다. 메뉴 인터페이스는 화면 한쪽에 초기화된 상태로 나타나며, 메뉴 타이틀을 선택하면 메뉴의 위치를 옮기는 것도 가능하다. 메뉴를 선택하는 것은 기본적으로 오브젝트를 선택하는 것과 동일하며, wand의 레이를 선택하고자 하는 메뉴버튼에 향하게 한 뒤에, 2번째 버튼을 누르면 Pick 이벤트가 발생해 해당 기능이 동작하기 시작한다. 향후에는 3차원 메뉴 등 보다 다양한 메뉴 인터페이스를 통해 다양한 기능을 지원하게 할 예정이다.

ParaView-VR에서는 오브젝트 및 메뉴를 이동할 수 있을 뿐만 아니라 Navigation 모드로 이동하는 것도 가능하다. Navigation 모드는 wand의 조이스틱으로 조작할 수

있으며, 이 기능을 이용하면 wand와 조이스틱의 방향에 따라 상, 하, 좌, 우로 마음대로 VR 세계를 탐색할 수 있다.

ParaView-VR에서는 멀티뷰(Multi-View) 인터페이스를 제공함으로써 한 데이터에 대해 다양한 필터 기능을 적용, 비교 분석하는 기능을 제공한다. ParaView-VR의 멀티 뷰 기능은 여러 개의 데이터에 대한 가시화 결과를 ParaView에서 한 파일에 저장하면 사용 가능하다. 여러개의 데이터가 저장된 파일을 로딩하면 화면 한쪽에는 각 데이터에 대한 아이콘이 나타나고, 이 아이콘을 선택하면 해당 아이콘에 대한 데이터 화면을 메인 화면에서 볼 수 있다. 향후에는 여러 개의 데이터를 한꺼번에 메인 화면에서 볼 수 있는 기능을 제공할 예정이다.

3.3. 기능

ParaView-VR에서는 VR 환경으로 로딩한 데이터에 필터를 적용해서 보다 다양한 형태로 데이터를 관찰하며 분석하는 것이 가능하다. 이런 필터 기능은 VTK의 필터에서 제공하는 기능을 사용했다.

폴리곤 데이터는 wireframe(그림 3), surface, point의 형태로 변경해서 관찰할 수 있으므로, 폴리곤 데이터의 형태가 다른 데이터를 분석하는데 방해가 되는 경우에 유용하게 활용할 수 있다.

또, 데이터에 대해 Clip(그림 4), Slice, probe(그림 5), Zoom 기능을 제공한다.

Clip과 Slice는 데이터의 한 부분을 내/외부를 cut plane으로 잘라서 그 단면을 볼 수 있는 기능이다.

Probe는 데이터 값을 자세히 보는 기능으로, 사용자가 지정한 데이터 내의 한 지점에 대한 벡터 또는 스칼라 값을 나타내는 것이다. 벡터 또는 스칼라 값을 나타내는 방법으로는 여러 가지가 있는데, 직접 숫자로 나타내는 방법도 있고, Glyph 필터같은 것을 이용, 벡터의 방향을 구체적으로 보여주는 방법도 있다. 혹은 평면을 이용, 해당 평면에 스칼라 값을 투사하는 방법도 가능하다. 향후에는 이런 기능을 제공할 예정이고, 현재는 probe 필터의 위치를 나타내는 point widget의 위치 좌표를 텍스트로 나타낸다.

Zoom은 줌-인 기능으로, 데이터 내에 위치한 박스 위젯 내부 데이터를 자세히 볼 수 있는 기능이다.

현재 기능은 구현 단계이며, 향후에는 보다 다양한 기능을 제공하고 기능의 완성도를 높임으로써 VR 환경에서 불편없이 데이터를 탐색할 수 있는 기능을 제공할 것이다.

4. 결론 및 향후 계획

ParaView-VR은 ParaView의 다양한 필터링 기능을 이용해 한 단계 걸러진 데이터를 VR 환경에서 가시화하는 애플리케이션이다. 이 애플리케이션은 VR 환경과의 인터페이스, 즉 트래킹과 사용자와의 상호작용을 제어하는 기능을 제공한다. VR 환경의 장점은 직접적인 사용자와의 상호작용을 통해 바로 원하는 형태로 데이터를 가시화할

수 있다는 것이다. ParaView-VR은 VR 환경의 이런 장점을 살려 사용자와의 상호 작용을 통해 데이터를 VR 환경에서 직접 원하는 형태로 가공해서 가시화하는 것도 가능하다. 현재는 이런 상호작용을 위해 Clip, Slice, Probe, Zoom과 같은 기능을 메뉴에서 제공하고 있다.

향후에는 메뉴 이외의 가능한 선택 인터페이스와 multimodal 인터페이스 등, 보다 다양한 VR 인터페이스를 구현해서 제공할 예정이다. 또, 기능 면에서도 완성도와 다양성을 높여서 사용자가 필요로 하는 데이터 탐색 기능을 제공할 것이다.

참고문헌

- [1] Arjan J.F.Kokl, Robert van Liere. "A Multimodal Virtual Reality Interface for 3D Interaction with VTK". Knowledge and Information Systems, vol 11(3), 2007.
- [2] Andries van Dam, Andrew S.Forsberg, David H.Laidlaw, Joseph H.Laviola, Rosemary M.Simpson. "Immersive VR for Scientific Visualization: A Progress Report". IEEE Computer Graphics and Applications, 20(6):26-52, 2000.
- [3] D.Rantzau, U.Lang. "A scalable virtual environment for large scale scientific data analysis". Future Generation Computer Systems, 14(3-4), 1998.
- [4] Joseph J.Laviola JR. "MSVT: A Virtual Reality-based Multimodal Scientific Visualization Tool". Proceedings of Second IASTED International Conference on Computer Graphics and Imaging, pp.221-225, 1999.
- [5] M. Schirski, A.Germdt, T. van Reimersdahl, T.Kuhlen, P.Adomeit, 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.
- [6] 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 the 4th International Immersive Projection Technology Workshop, 2000.
- [7] W. Schroeder, K.Martin, B.Lorensen. "The Visualization Toolkit, an Object-Oriented Approach to 3D Graphics, 4th Edition". Kitware, 2006.
- [8] D.P.Shamonin. VtkCave, <http://staff.science.uva.nl/~dshamoni/myprojects/VtkCave.html>
- [9] <http://brinhton.ncsa.uiuc.edu/prajlich/vtkActorToPF>
- [10] <http://www.hlrs.de/organization/vis/covise>
- [11] <http://www.paraview.org>
- [12] <http://www.vtk.org>