



ParaView-VR: VR 환경에서의 과학 데이터 가시화

(ParaView-VR: Scientific Visualization on VR Environments)

허 영 주 (popea@kisti.re.kr)

한국과학기술정보연구원
Korea Institute of Science & Technology Information

목차

1. 개요	1
2. 관련 연구	1
3. ParaView-VR의 구조	4
4. VTK-VR	5
5. ParaView-VR에서 제공하는 사용자 인터페이스	6
6. ParaView-VR에서 제공하는 기능	7
7. 결론	10
8. 참고문헌	10

그림 차례

[그림 3-1] ParaView-VR의 구조	4
[그림 5-1] ParaView-VR의 메뉴 인터페이스	6
[그림 6-1] Wireframe 모드	8
[그림 6-2] Clip	9
[그림 6-3] Probe	9

1. 개요

과학 데이터는 대부분 과학 시뮬레이션의 결과로 얻어지며, 사람이 직관적으로 이해하기 어려운 숫자의 나열인 경우가 많다. 이런 수치 데이터를 가시화하면 사람이 직관적으로 쉽게 이해하고 분석할 수 있는 형태가 갖춰진다. 이 때, 가시화 환경을 VR 환경으로 옮기면 사용자와의 상호작용을 통해 사용자가 원하는 형태로 데이터를 가시화할 수 있을 뿐만 아니라 VR이 제공하는 몰입형 환경을 통해 데이터에 대한 이해도를 높일 수 있다.

이런 데이터는 일반적으로 VR 환경에서 가시화하기에 앞서 다양한 필터나 가시화 알고리즘을 적용해서 가시화에 적합한 형태로 바꿔야 한다. 이런 복잡한 알고리즘은 VTK(Visualization ToolKit)에서 다양한 형태로 제공한다.

VTK는 여러 다양한 기능을 제공하며 여러 기반 애플리케이션 개발에 활용되고 있다. ParaView도 VTK를 기반으로 한 애플리케이션 중 하나로, 주로 다양한 데이터셋을 가시화할 용도로 제작된 것이다.

본 문서에서는 이런 ParaView에서 1차로 다양한 필터를 적용한 데이터를 VR 환경에서 가시화하는 프레임워크를 제안한다. 이 프레임워크에서는 ParaView에서 가시화한 데이터를 그대로 VR 환경으로 옮겨서 볼 수 있다. 원칙적으로 VR 환경을 지원하지 않는 VTK와 VR 환경과의 인터페이스로는 VtkCave를 활용했으며, ParaView에서 가공한 데이터를 사용자가 VR 환경에서 다시 가공해서 볼 수 있도록 VTK를 이용한 몇몇 기능을 제공한다.

이 프레임워크에서는 기본적으로 다중 뷰(multi-view)기능 제공함으로써, 여러 가지 가시화된 결과를 서로 비교해서 볼 수 있게 했다. 또, 데이터를 가공해서 볼 수 있는 여러 다양한 기능을 제공한다.

2. 관련 연구

VTK는 방대한 가시화 알고리즘을 제공하는 공개소스 소프트웨어 라이브러리로, 주로 과학 데이터를 가시화하는데 널리 이용되고 있다.

특히 VTK는 3차원 컴퓨터 그래픽스, 이미지 처리 및 가시화에 주로 사용되고 있으며, 매우 다양한 기능을 제공한다. VTK는 스칼라, 벡터, 텐서, 텍스처 및 볼륨 데이터를 표현하는 자료구조 뿐만 아니라 데이터를 필터링 하는 필터, 그리고 다양한 고급 모델링/렌더링 기법도 제공하고 있으며, 현재 데이터 가시화에 매우 널리 사용되고 있는 툴이다. 이렇게 다양한 VTK의 기능을 VR 환경에 적용할 수 있다면, VR 환경에서 사용자에게 다양한 가시화 기능을 제공하고, 사용자와의 상호 작용을 통해 다양한 방식으로 데이터를 표현하고 이해하는 것이 가능하다.

이런 VTK의 다양한 방식을 이용한 애플리케이션으로는 ParaView를 들 수 있다. ParaView는 Kitware사와 Los Alamos National Laboratory의 공동 프로젝트를 통해 개발된 애플리케이션으로, 다양한 데이터셋을 가시화할 용도로 설계, 제작됐다. 이 애플리케이션은 데이터 처리와 렌더링에 VTK를 이용하며, 병렬 가시화 기능을 제공한다.

그러나 VTK나 ParaView는 VR과 관련된 기능을 제공하지 않는다. VTK에서는 일부 입체 가시화 기능을 제공하기는 하지만, VR의 핵심 기능이라 할 수 있는 트래킹이나 사용자와의 상호 작용과 관련된 기능을 지원하지 않기 때문이다. 이에, VTK의 다양한 알고리즘을 VR 환경에서 활용할 수 있게 하기 위한 시도는 지속적으로 이뤄져 왔다.

VTK에 VR 기능을 접목하는 방법은 크게 2가지로 나뉜다. 우선, VTK의 액터(actor)를 VR 환경에서 주로 사용하는 씬 그래프(Scene Graph)로 변환하는 방법이 있다.

VTK는 기하 정보를 액터라는 형태의 데이터 구조로 생성해서 렌더링하는데, 이런 액터를 VR 환경에서 많이 사용하는 씬 그래프로 변환하는 것이다. 이 방식은 복잡한 필터링 알고리즘과 렌더링이 분리되기 때문에, 렌더링이 필터링 알고리즘에 의해 방해받지 않는다는 장점이 있다. 그러나, VTK 파이프라인과의 직접적인 상호작용이 불가능하기 때문에, VR 환경에서 직접 사용자가 원하는 형태로 데이터를 가공해서 가시화하는 것은 불가능하다. 이 방식을 채택한 툴로는 vtkActorToPF와 VTK2CAVE를 들 수 있다.

vtkActorToPF는 VTK의 액터를 Performer의 씬 그래프 노드인 PfGeode 형태로 변환한다. 즉, vtkActorToPF의 사용자는 VTK를 이용해

서 기하 정보를 추출하고 Iris Performer를 이용해서 그 정보를 렌더링하게 되는 것이다. 이 때, 썬 그래프는 VTK와는 독립적으로 렌더링되며, 이렇게 함으로써 VR 환경을 위한 다중 채널 렌더링이 가능해지는 것이다.

반면, VTK2CAVE는 변환 프로세스가 진행되는 중, VTK 구조 말단에서 동작하는 OpenGL 명령어를 CAVE 라이브러리의 특정 프레임워크에 알맞은 형태로 재생성한다. VTK2CAVE는 매우 안정적이고 사용하기 쉬우며, 성능이 좋다는 장점이 있다.

VTK에 VR 기능을 접목하는 또다른 방법은 VTK의 렌더러를 VR 환경에 알맞게 변경하는 것이다. 이 방식의 핵심은 VTK에서 VR기능을 지원하지 않는 `vtkRenderer` 클래스와 `vtkRenderWindow` 클래스 및, `vtkRenderWindowInteractor` 클래스와 관련된 클래스에 VR 지원 기능을 추가하는 것이다. 이 방식은 VTK의 거의 모든 기능을 그대로 사용할 수 있을 뿐만 아니라 비교적 구현하는데도 어려움이 없다는 장점을 가진다. 이 방식을 채택한 툴로는 `VtkCave`와 `VR-VTK`, `ViSTA` 등을 들 수 있다.

`VtkCave`는 VTK와 `CaveLib`을 사용하는 라이브러리로, VTK 모델링 스타일의 모델링 기법과 3차원 가상 메뉴, 그리고 `CaveLib` 스타일의 상호작용을 제공한다. `VtkCave`는 VTK 라이브러리에 추가 가능한 한 요소로 취급할 수 있으며, 사용자가 별도로 OS나 디스플레이 시스템, 디바이스-레벨의 프로그래밍 기법에 신경을 쓰지 않아도 애플리케이션에 VR 기능을 추가할 수 있다.

`VR-VTK`는 VTK에 multimodal VR 인터페이스를 덧붙여서 손으로는 위젯과 픽커를 이용한 조작을, 발로는 클러칭을, 머리로는 카메라 조작을, 그리고 말로 시스템을 조절하는 것도 가능하다. `VR-VTK`에는 몇가지 새로운 VTK 클래스를 제공하는데, 이때 새로 추가된 클래스는 상호작용과 렌더링에 관련된 기능만 제공하며, 2차원 및 3차원으로 구성된 다양한 사용자 인터페이스와 상호 작용을 제공한다.

`ViSTA`는 수치 시뮬레이션, 가상 프로토타입, 구조, 약학 및 심리학 분야를 주 타겟으로 하는 VR 가상화용 라이브러리로, `VtkCave`를 기반으로 하는 VR 기능을 제공한다.

본 문서에서는 `ParaView`에서 1차적으로 다양한 필터를 적용해서 가

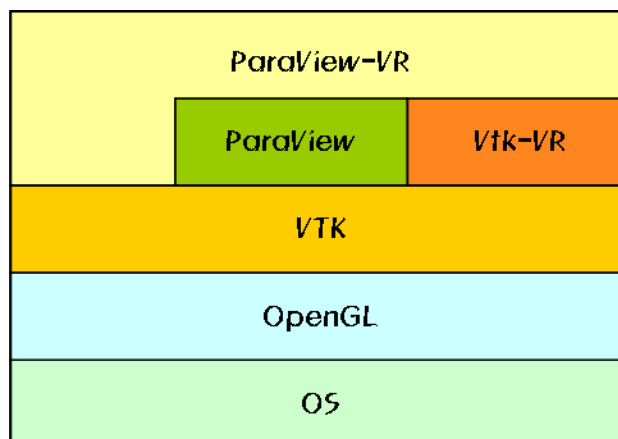
시화한 데이터를 VR 환경으로 옮겨서 다양한 상호작용을 제공하는 ParaView-VR 애플리케이션을 구현한 내용을 설명할 것이다. 이 애플리케이션에서는 ParaView에서 가시화한 데이터를 그대로 VR 환경으로 옮겨서 볼 수 있는데, 기존의 가시화 애플리케이션과 VR 애플리케이션을 이런 식으로 연계한 사례는 COVISE와 OpenCOVER에서 찾아볼 수 있다.

ParaView-VR에서는 ParaView에서 가시화한 결과를 VR 환경에서 탐색할 수 있을 뿐만 아니라, VR 인터페이스에서 제공하는 여러 필터 기능을 통해 데이터를 직접 조작해서 가시화한 결과를 비교해 보는 것도 가능하다.

3. ParaView-VR의 구조

ParaView-VR은 ParaView에서 각종 필터를 적용한 결과를 VR 환경으로 가져와서 사용자 인터랙션을 포함한 각종 동작을 가능케 한다. 이 때, ParaView와 VTK에서는 지원하지 않는 VR 기능을 지원하기 위해 VTK-VR을 사용한다. VTK-VR은 VtkCave를 기반으로 트래킹 및 인터랙션과 관련된 기능을 제공하는 라이브러리로, 이 라이브러리를 이용하면 대부분의 VTK 기능을 VR 환경에 적용할 수 있다.

ParaView-VR은 ParaView와 VTK-VR에서 제공하는 기능을 통해 VR 환경에서 사용하는 기능을 제공하면서, 동시에 VTK에서 제공하는 필터를 이용해서 VR 환경에서 가시화한 데이터를 사용자와의 상호작용에 따라 가공하는 역할도 수행한다. 이런 ParaView-VR의 구조를



[그림 3-1] ParaView-VR의 구조

그림으로 표현하면 [그림 3-1]과 같다.

ParaView-VR에서 VTK를 이용해서 제공하는 기능은 데이터를 Wire frame, Surface, Point의 다양한 형태로 표현하는 기능을 포함해서 clip, slice, probe, Magnifying 등이다.

4. VTK-VR

ParaView-VR은 VTK-VR 인터페이스를 이용, VTK를 기반으로 만든 ParaView 데이터를 VR 환경에 디스플레이한다. VTK-VR은 VtkCave를 기반으로 하는 라이브러리로, VTK와 VR 환경 사이에서 트래킹이나 상호작용과 관련된 기능을 제공한다.

기존의 VtkCave는 유연성이 높고 사용하기 쉬운 라이브러이긴 하지만, VTK의 이전 버전을 대상으로 구현됐기 때문에 현재 널리 사용되고 있는 VTK 버전과는 호환되지 않는다는 단점이 있다. 게다가 현재 개발이 중단된 상태로, 향후 개선의 여지도 거의 없다고 볼 수 있다. 이에, VtkCave를 VTK의 현 버전과의 호환성을 고려해서 수정하고 시스템의 특성에 따라 여러 기능을 수정하거나 추가, 제거함으로써 현재 VTK 버전과 가상화 시스템에서 활용 가능한 VTK-VR을 구현했다.

VtkCave에서 VTK의 현 버전과 주로 문제를 일으키는 부분은 함수의 리턴 데이터 타입과 콜백 함수를 호출하는 부분이다. VTK의 버전이 바뀌면서 가장 변화가 많이 생긴 부분이기 때문이다. 따라서 이 부분을 VTK 현버전에 맞게 수정함으로써 VtkCave에서 발생하는 컴파일 문제를 해결하고, VTK의 현버전에서 사라져버린 함수를 추가하는 작업도 수행했다. 또, 시스템과 충돌 문제를 일으키는 vtkCaveCamera 클래스를 삭제함으로써 프로그램 실행시 발생하는 에러 문제를 해결했다. 이외에도 VTK-VR의 핵심이라 할 수 있는 vtkCaveActor 클래스를 시스템에 맞게 수정하고, 사용자 인터페이스를 제공하는 vtkCaveMenu 클래스와 vtkCaveMenuButton 클래스에 적절한 기능을 추가, 수정함으로써 메뉴 구성에 좀더 변화를 줄 수 있게 했다. 또, 사용자와의 상호작용을 담당하는 vtkCaveInteractor 클래스와 vtkCaveRenderWindowInteractor에도 여러 기능을 추가, 수정함으로써 시스템에서 3차원 VR 입력장치인 wand를 사용할 수 있게 했을뿐만 아니라, 보다

다양한 방식의 상호작용을 제공할 수 있게 했다.

5. ParaView-VR에서 제공하는 사용자 인터페이스

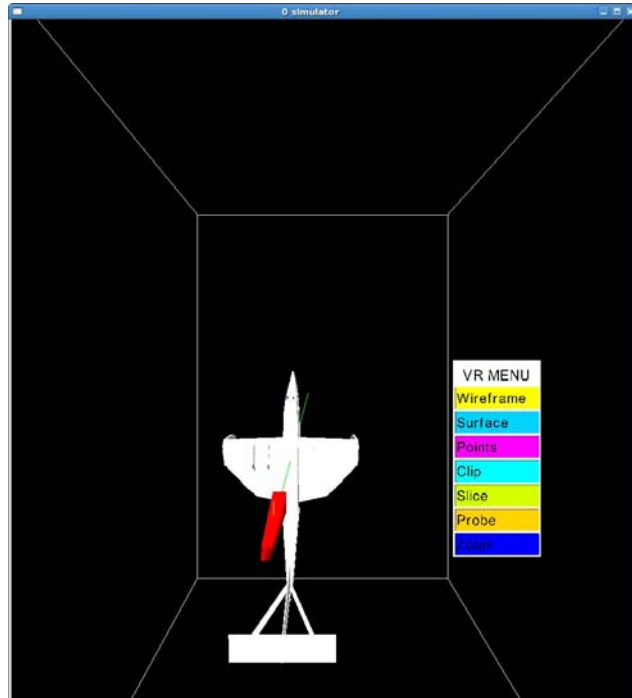
ParaView-VR 은 트랙커 중심의 인터페이스를 제공하며, 사용자 입력은 주로 wand에서 제공하는 버튼과 조이스틱을 통해 받는다.

오브젝트의 선택은 wand를 통해 이뤄지며, 사용자는 wand의 중앙에서 나오는 레이를 통해 wand가 현재 가리키는 위치를 파악할 수 있다. wand의 첫 번째 버튼으로는 오브젝트에 대한 동작 모드를 설정할 수 있고, wand의 두 번째 버튼으로는 오브젝트를 선택하고 그 오브젝트의 동작을 제어하거나 메뉴 버튼을 선택할 수 있다.

wand의 레이로 오브젝트를 향하게 한 뒤, wand의 버튼을 누르면 오브젝트에 대해 Pick 이벤트가 발생하면서 해당 오브젝트가 선택된다. 만약 2번째 버튼을 누른 채로 wand를 움직이면 현재 모드에 따라 오브젝트를 이동(translation), 회전(rotation), 혹은 크기 조절(scaling) 하는 것이 가능하다.

이동 모드에서는 오브젝트의 움직임이 wand의 움직임에 대해 반응한다. 회전 모드에서는 wand의 회전각(orientation)에 따라 오브젝트가 회전하며, 크기 조절 모드에서는 wand의 Z-방향 움직임에 따라 오브젝트의 크기가 달라진다.

레이를 오브젝트를 향하게 한 상태에서 wand의 첫 버튼을 누르면 오브젝트의 움직임에 대한 모드를 변경할 수 있다. 오브젝트의 움직임에 대한 모드는 이동, 회전, 크기 조절 순으로 바뀌게 되며,



[그림 5-1] ParaView-VR의 메뉴 인터페이스

향후에는 모드가 바뀔 때마다 커서가 바뀌는 기능도 추가될 예정이다. 또, 회전 모드에서는 화면 한편에 오브젝트에 대한 회전축을 추가함으로써 회전 각도를 보다 명확하게 구분할 수 있게 하는 인터페이스를 추가할 예정이다.

ParaView-VR에서 제공하는 또다른 인터페이스로는 [그림 5-1]과 같은 메뉴 인터페이스를 들 수 있다.

메뉴 인터페이스에서는 마지막으로 선택한 오브젝트에 다양한 가시화 효과를 적용할 수 있게 한다. 애플리케이션을 실행하면 메뉴 인터페이스는 화면 한쪽에 초기화된 상태로 나타나며, 메뉴 타이틀을 선택하면 오브젝트를 옮기듯이 메뉴의 위치를 옮기는 것도 가능하다. 물론 메뉴를 선택하는 방식은 오브젝트를 선택하는 방식과 동일하다. 즉, wand의 레이를 원하는 메뉴 위치에 놓고 2번째 버튼을 누르면 Pick 이벤트가 발생해 원하는 동작을 수행할 수 있다. 만약 메뉴 타이틀에서 이 이벤트가 발생했다면 메뉴의 위치를 옮길 수 있고, 메뉴 버튼에서 이 이벤트가 발생했다면 해당 기능이 오브젝트에 적용된다.

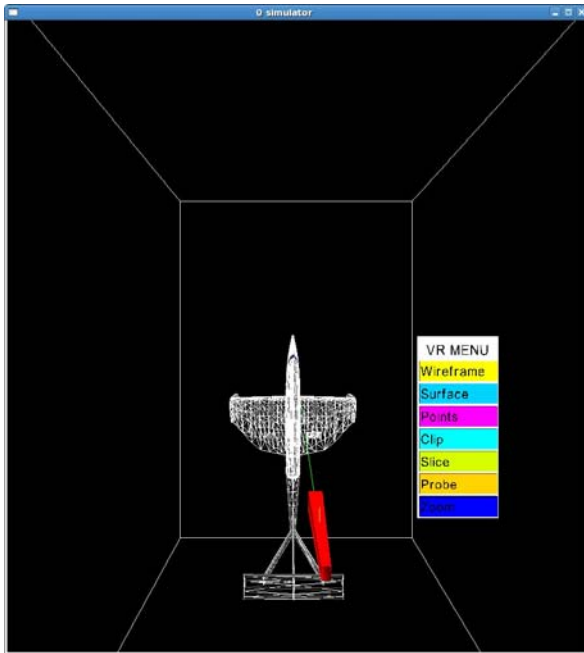
ParaView-VR에서는 오브젝트 및 메뉴를 이동하는 기능뿐만 아니라 조이스틱을 이용한 Navigation 기능도 제공한다. 이 기능은 wand와 조이스틱의 방향에 따라 상, 하, 좌, 우 마음대로 움직이며 VR 환경을 탐색하는 기능이다.

메뉴와 wand를 이용한 인터페이스 외에도 ParaView-VR에서는 멀티뷰를 선택하는 버튼이나 조명, 아웃라인 등을 제어하는 버튼을 VR환경의 한 영역에서 제공한다. 향후에는 더욱 다양한 인터페이스를 제공하게 될 것이다.

6. ParaView-VR에서 제공하는 기능

ParaView-VR은 VR 환경으로 로딩된 데이터를 보다 다양한 형태로 분석하는 기능을 제공한다. 이런 기능은 대부분 VTK를 이용해서 구현했다.

폴리곤 데이터는 wireframe(그림 6-1), surface, point의 형태로 분석할 수 있으며, 특히 wireframe과 point 형태의 경우에는 폴리곤의 surface가 다른 데이터를 분석하는데 방해가 되는 경우에 유용하게 활



[그림 6-1] Wireframe 모드

용할 수 있다.

데이터 분석과 관련된 기능으로는 clip(그림 6-2), slice 및 probe(그림 6-3) 기능을 제공한다.

clip, slice 혹은 probe를 메뉴에서 선택하면 오브젝트를 둘러싼 아웃라인이 나타나면서, 해당 기능과 연동된 위젯이 나타난다. clip과 slice는 평면 형태의 plane 위젯을 사용하고, probe는 point 위젯을 사용한다.

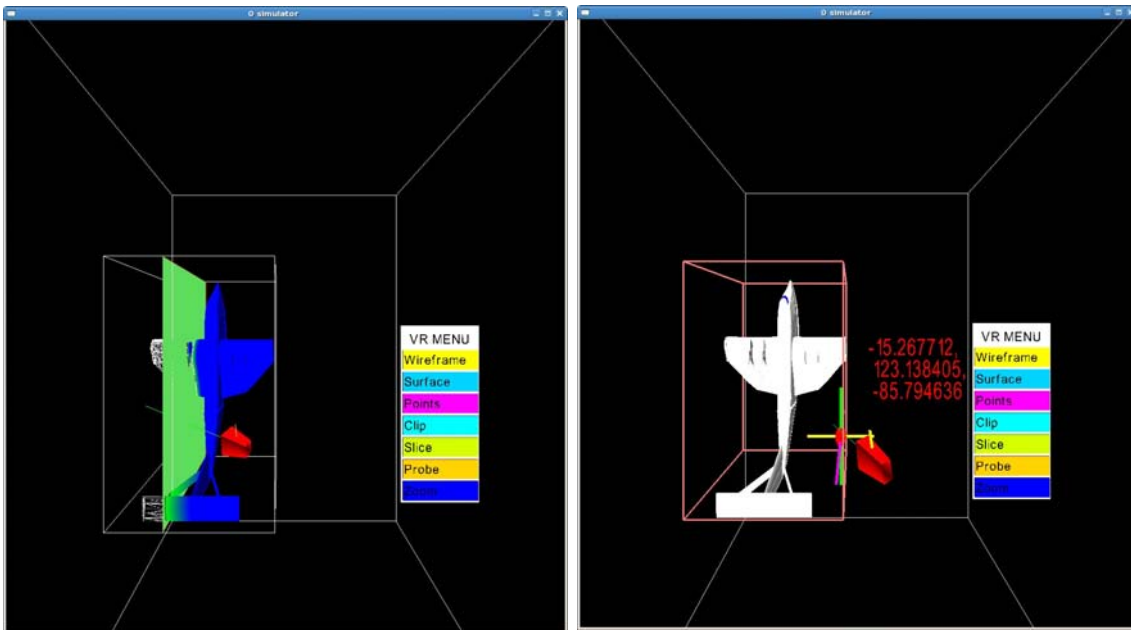
ParaView-VR에서는 이외에도 box 형태의 위젯도 제공하며, 이런 위젯

은 ParaView-VR을 이용하는 애플리케이션이라면 모두 이용할 수 있다. 각 위젯은 wand로 선택해서 움직일 수 있으며, 선택된 경우에는 하이라이트 색상으로 바뀌게 된다.

clip 기능은 데이터의 일부를 cut plane으로 잘라서 절단된 오브젝트를 보여주는 기능으로, cut plane은 오브젝트와 같은 방식으로 마음대로 이동하거나 회전시킬 수 있다. 단, 이 cut plane은 오브젝트의 아웃라인 내에서만 움직인다. cut plane에 대한 이런 기능은 기본적으로 plane 위젯이 제공한다. 메뉴에서 clip을 선택하면 오브젝트 주위로 아웃라인이 만들어지고, 아웃라인의 중앙에는 cut plane이 나타난다. 사용자는 이 cut plane을 움직이면서 절단된 오브젝트를 관찰할 수 있다.

slice 기능은 clip 기능과 거의 유사한데, clip 기능이 오브젝트를 절단한 후 절단된 물체를 보여주는 기능인 반면, slice 기능은 오브젝트의 절단면을 보여주는 기능이다. 이 기능 역시 clip과 마찬가지로 plane 위젯으로 구성된 cut plane을 사용하며, 오브젝트의 단면은 데이터의 값과 종류에 따라 다양한 형태로 표현된다.

Probe는 사전적으로는 ‘탐침’, ‘탐사기’라는 의미를 가지며, 입력 데이터셋을 일련의 포인트로 샘플링함으로써 데이터셋의 속성을 파악한다는 의미를 가진다. ParaView-VR에서는 데이터 값의 세부 내역을 제



[그림 6-2] Clip

[그림 6-3] Probe

공하는 기능을 나타내며, 사용자가 지정한 데이터 내의 한 지점에 대한 벡터 혹은 스칼라 값을 나타낸다. 이렇게 데이터의 값을 나타내는 방법에는 여러 가지가 있는데, 직접 숫자로 나타내는 방법도 있고, Glyph 필터같은 것을 이용해서 벡터의 방향을 직접 시각적으로 보여주는 방법도 있다. 혹은 평면을 이용, 해당 평면에 스칼라 값을 투사하는 것도 가능하다. 현재는 메뉴에서 Probe를 선택하면 데이터 내에 나타나는 포인트 위젯의 위치 좌표를 텍스트로 보여주는 기능을 제공한다.

Zoom은 줌-인 기능으로, 이 메뉴를 선택하면 오브젝트는 화면 한쪽에 작게 표시되고, 오브젝트 내에는 박스 위젯이 나타난다. 이 박스 위젯을 움직이면 메인 화면에 박스 위젯이 가리키는 위치에 해당하는 데이터가 확대되어서 나타난다. 즉, 오브젝트에 위치한 박스 위젯의 내부 데이터를 자세히 볼 수 있는 기능이다.

또, 멀티뷰(Multi-View) 인터페이스를 이용하면 한 데이터에 대해 다양한 필터 기능을 적용한 결과를 VR 환경에서 비교 분석하는 것이 가능하다. ParaView-VR의 멀티-뷰 기능은 여러 개의 데이터에 대한 가시화 결과를 ParaView에서 저장, 로딩하면 사용 가능하다. 여러 개의 가시화 결과가 저장돼 있는 파일을 로딩하면 화면 한쪽에 각 데이터에 대한 아이콘이 나타나는데, 이 아이콘을 선택함으로써 메인 화면에 가시화 결과를 로딩할 수 있다.

현재 이런 기능은 구현 단계에 있다. 향후에는 현재 구현 단계에 있는 기능의 완성도를 높이고 보다 다양한 기능을 제공하게 될 것이다.

7. 결론

ParaView-VR은 ParaView와 VTK의 기능에 VTK-VR의 VR 기능을 덧붙인 몰입형 가시화 환경용 애플리케이션으로, ParaView에서 가시화한 결과를 그대로 VR 환경으로 옮겨놓을 수 있는 애플리케이션이다. ParaView-VR에서는 사용자와의 직접적인 상호작용을 통해 데이터를 원하는 형태로 가공해서 가시화하는 것도 가능하다. 이렇게 하기 위해 ParaView-VR은 Clip, Slice, Probe, Zoom같은 필터를 제공하며, 이렇게 필터를 적용한 데이터를 서로 비교해 볼 수 있는 멀티-뷰 기능도 제공한다.

향후에는 보다 다양한 VR 인터페이스를 제공하고, 완성도를 한층 높은 다양한 기능을 제공할 것이다.

8. 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.Adom

eit, 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, Jörg 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>