

ヘッドマウントディスプレイ向け 統合可視化基盤システムの開発



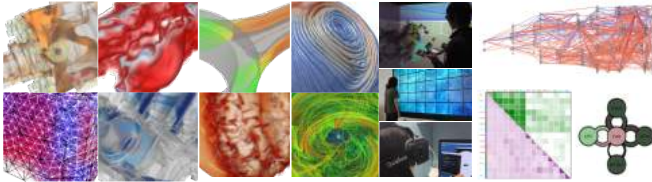
長濱 愛珠咲 (神戸大学工学部情報知能工学科)

坂本 尚久 (神戸大学大学院システム情報学研究所)

はじめに

様々な可視化技術・システム

- GPUを使用したリアルタイム可視化技術
- 大規模数値計算向け可視化技術 (In-situ可視化)
- CAVEやタイル型表示装置を使ったVR可視化システム
- グラフ可視化技術などを使った視覚的分析システム



- ヘッドマウントディスプレイ (HMD) を使った没入型分析技術が注目されているがシステム移植に手間

目的

In-situ可視化から対話的なVR可視化までをサポートするHMD向けの統合可視化基盤システムを開発する。

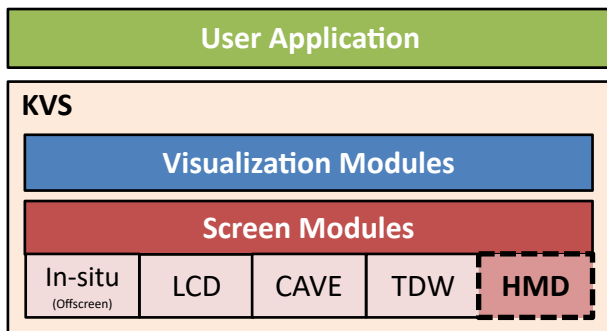
方法

特徴

- 可視化フレームワークKVSに対してHMD向け表示機能を実装することで既存コードと高い互換性を実現
- 対話操作機能の実装基盤を提供し独自のインタラクション機能を実装可能

可視化フレームワークKVS

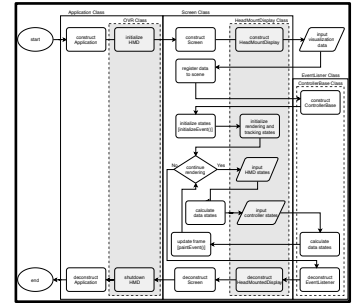
- ・システム構成



<https://github.com/naohisas/KVS>

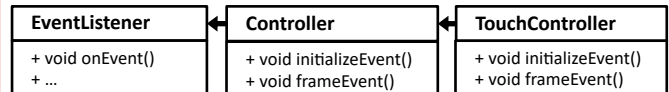
HMD向け表示機能の実装

- Applicationクラス
 - HMDの接続・起動の管理を行う。
- Screenクラス
 - HMDの情報の管理を行う。
 - HMD用の座標変換や描画処理を制御する。



対話操作機能の実装

- Controllerクラス
 - KVSのEventListenerクラスを継承しイベント処理を行う。
 - デバイスの姿勢情報を参照し独自操作機能を実装する。



結果

実験環境

- ・ HMD: Oculus Rift / Touch
- ・ PC: Intel Core i7, GeForce GTX 1080Ti

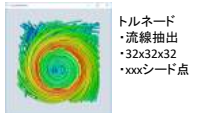


実装例

- ・ 可視化コード例

```

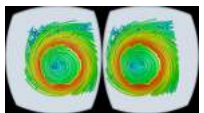
kvs::glut::Application app( argc, argv );
kvs::glut::Screen screen( &app );
screen.registerObject( Streamlines() );
screen.show();
app.run();
  
```



- ・ HMD向けコード例

```

kvs::oculus::Application app( argc, argv );
kvs::oculus::Screen screen( &app );
kvs::oculus::TouchController cntl( &screen );
screen.registerObject( Streamlines() );
screen.show();
app.run();
  
```



対話操作機能

- ・ 平行移動 (左手)
- ・ 回転 (右手)
- ・ 拡大縮小 (両手)



<https://github.com/vizlab-kobe/KVS.oculus>

まとめ

- VRに関する知識がなくてもHMD向け可視化システムの基本機能を実装できる可視化基盤を開発した。
- 目的に応じた対話操作機能の実装が必要である。