

効率的な知見獲得を目指した 大規模数値計算向けスマートIn-situ可視化

坂本 尚久

神戸大学

プラズマシミュレータシンポジウム2021
2021年09月16日 - 17日（オンライン）

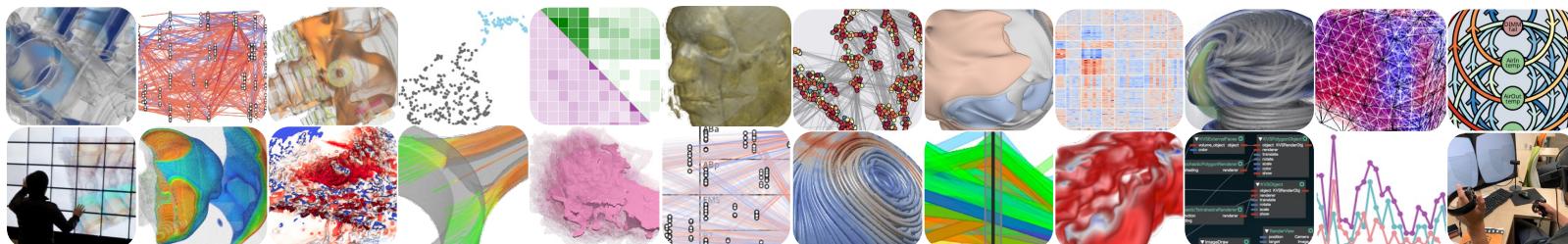
自己紹介

・坂本 尚久（さかもと なおひさ）

- ・所属：神戸大学・システム情報学研究科・計算科学専攻（准教授）
理化学研究所・計算科学研究センター（客員研究員）
 - ・学会：IEEE Computer Society, Association for Computing Machinery,
可視化情報学会（理事・和文誌編集委員）,
日本シミュレーション学会（理事・英文誌編集委員）, etc.

• 研究

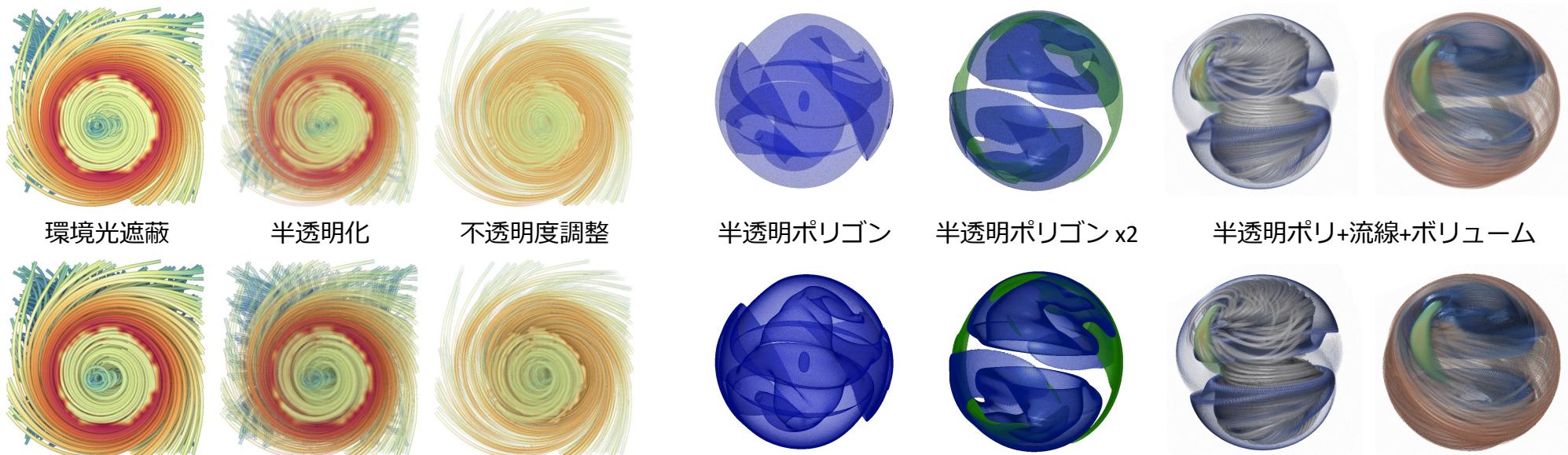
- #### • データ可視化と視覚的データ解析



研究紹介 (1/4)

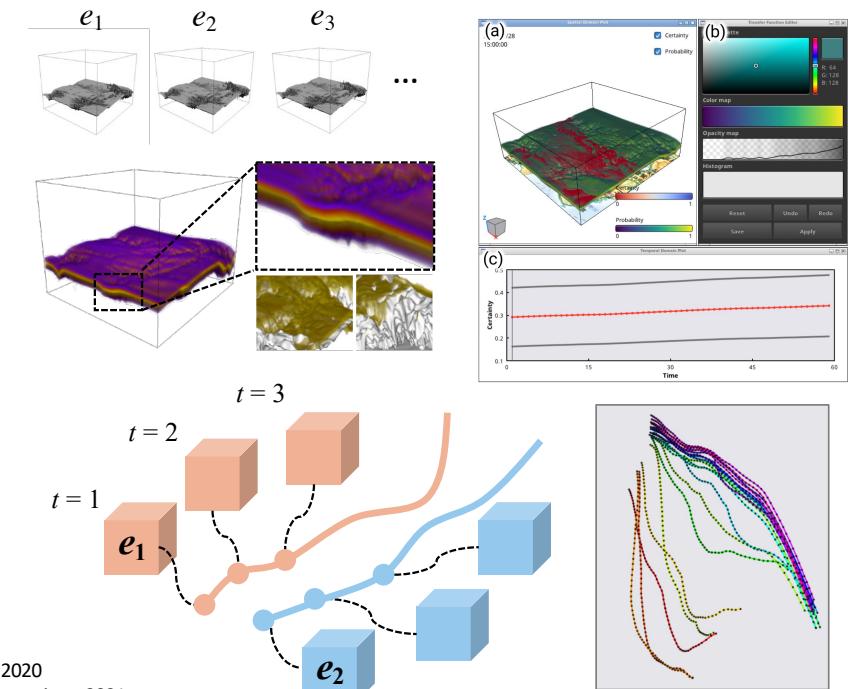
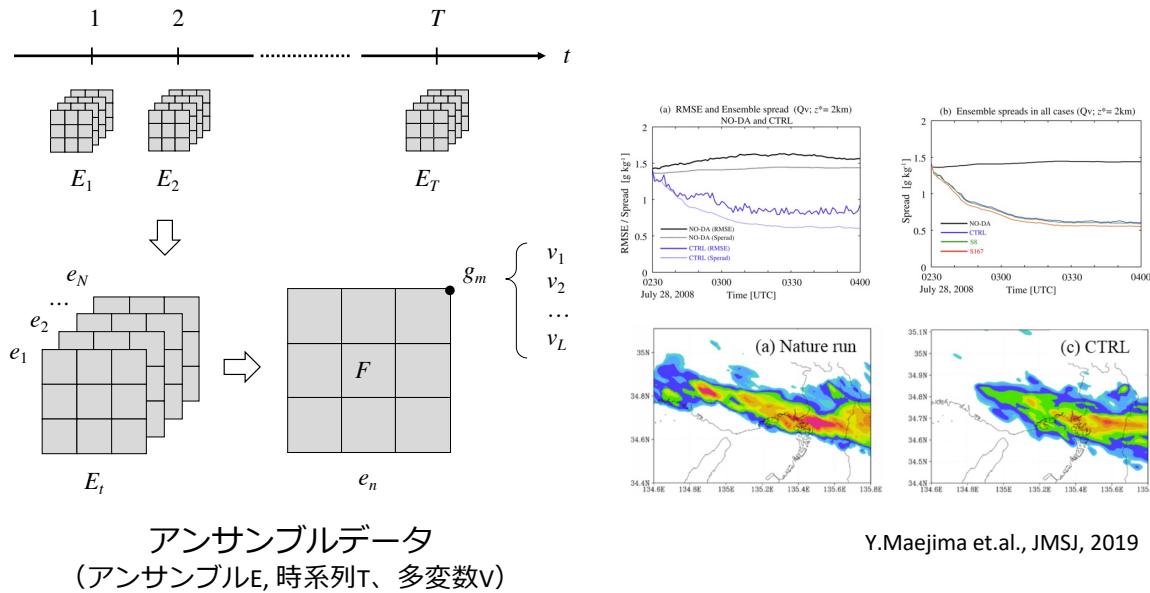
- 重要な変化を見逃さない可視化
 - 複雑データ向けの高精細半透明レンダリング

データ提供：神戸大陰山教授



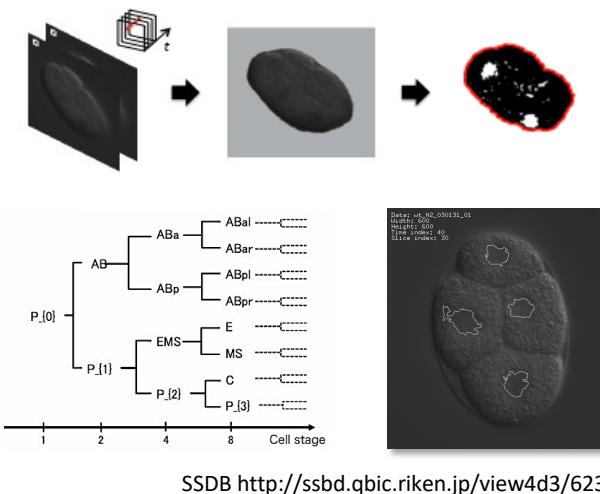
研究紹介 (2/4)

- 重要な変化を見逃さない可視化
 - アンサンブルシミュレーション向け視覚的データ解析

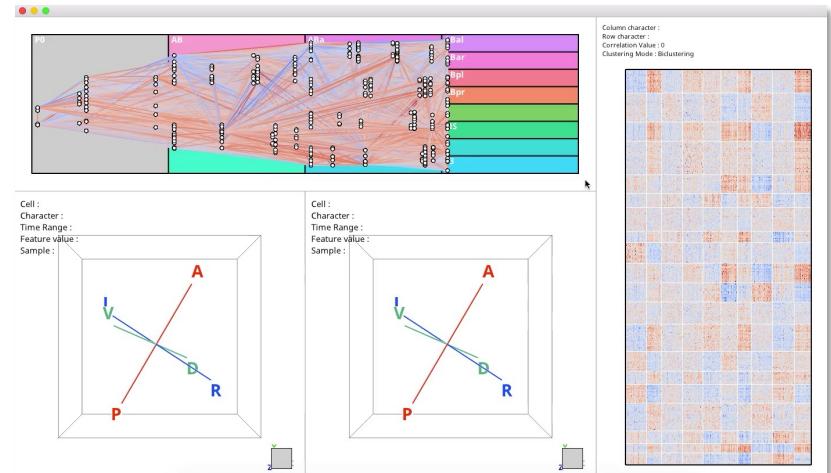
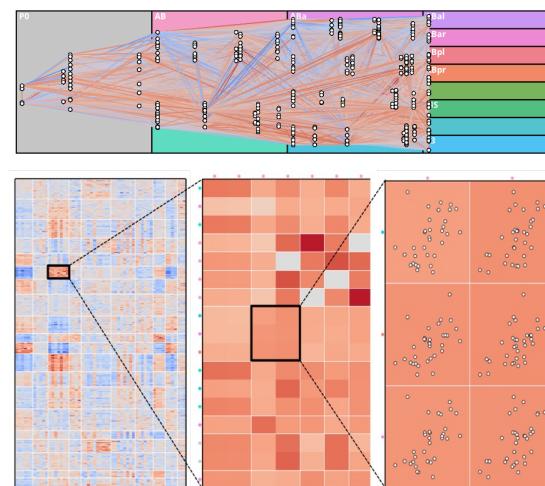


研究紹介 (3/4)

- 対話的に試行錯誤ができる視覚的解析
 - 生物細胞の発生動態解析

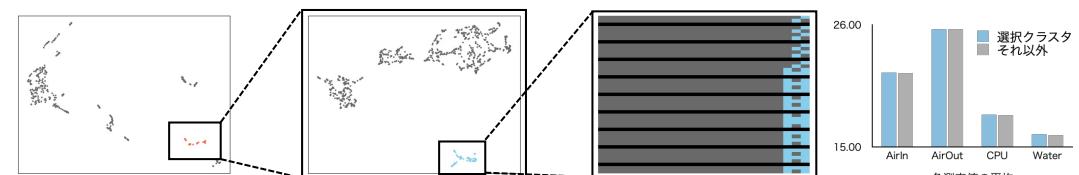
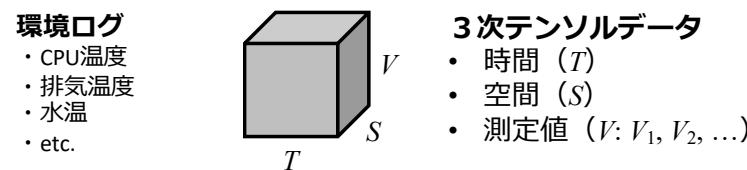
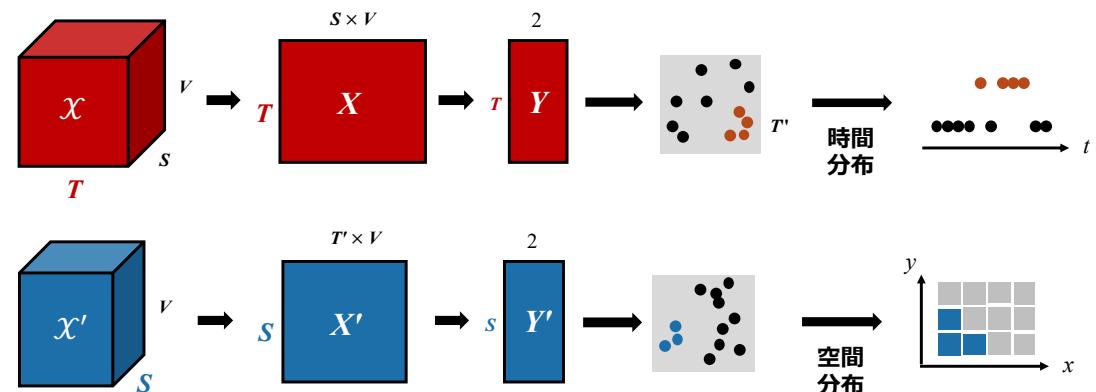
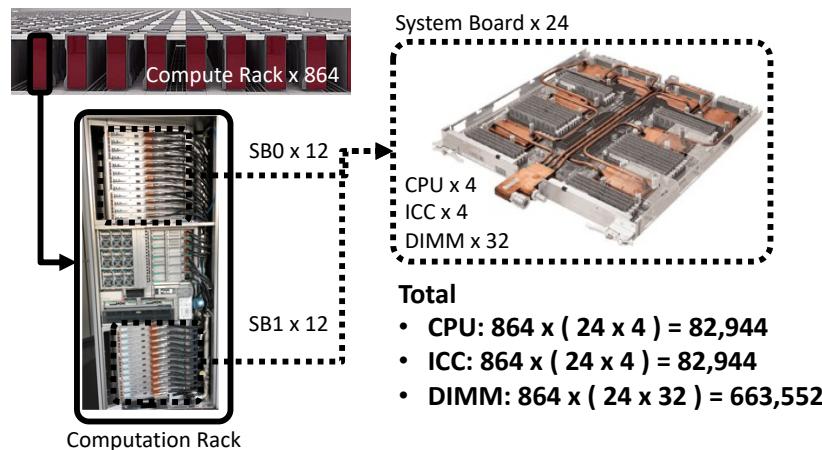


データ提供：理研大浪教授



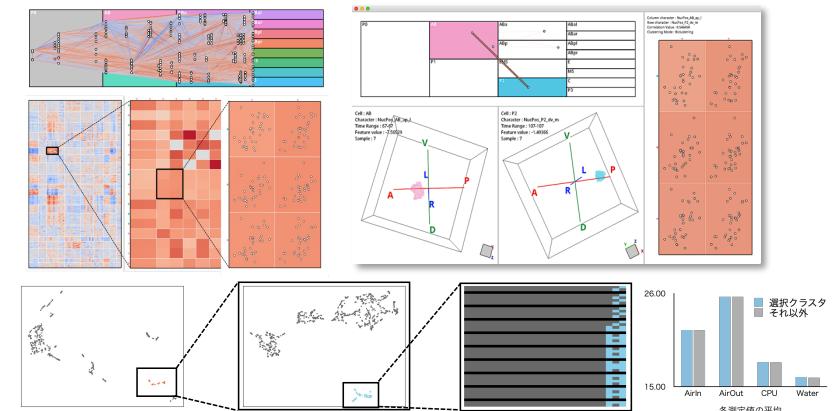
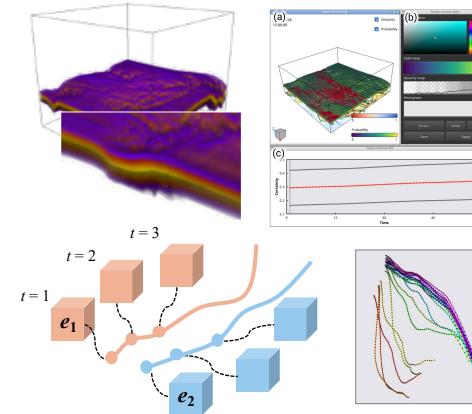
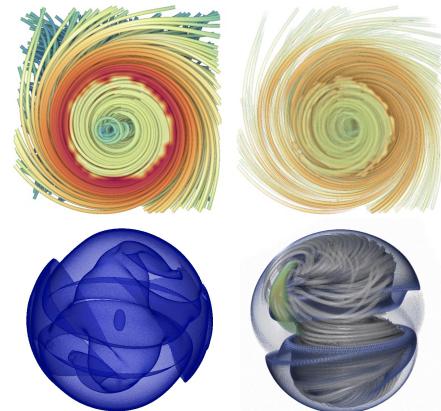
研究紹介 (4/4)

- 対話的に試行錯誤ができる視覚的解析
 - スパコンログデータを使った状態解析



背景

・データ可視化と視覚的データ解析



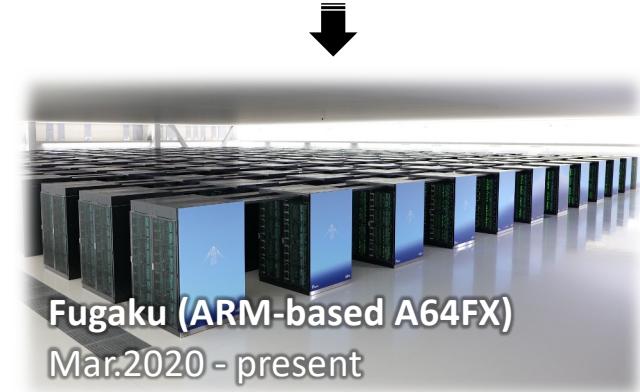
原点回帰
スパコンと可視化

背景

- ・**スーパーコンピュータと可視化**
 - ・スパコンの高機能・高性能化
 - ・「京」から「富岳」へ
 - ・世界スパコン性能ランキング1位
 - Top500, Graph500, HPCG, HPL-AI
 - ・分野横断的なデータの利活用
 - データが示唆する情報を多角的に理解
 - データの大規模・複雑化
 - 大規模データからの新たな知見獲得が困難な状況が発生
- ・**大規模データ向けの可視化***
 - 可視化技術の更なる高度化
 - 知識獲得に至る**プロセスの簡略化**



K computer (SPARK64)
Sep.2012 to Aug.2019

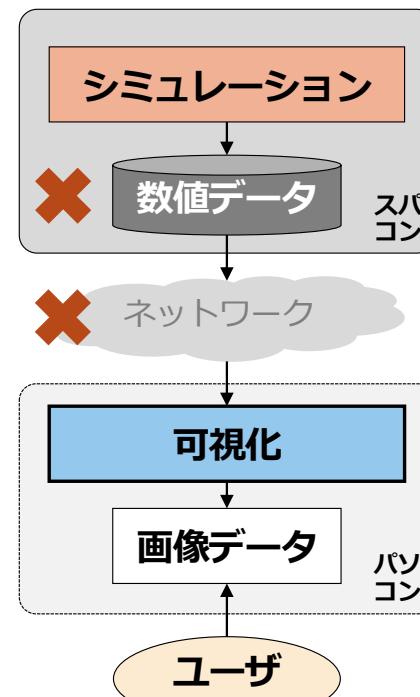


Fugaku (ARM-based A64FX)
Mar.2020 - present

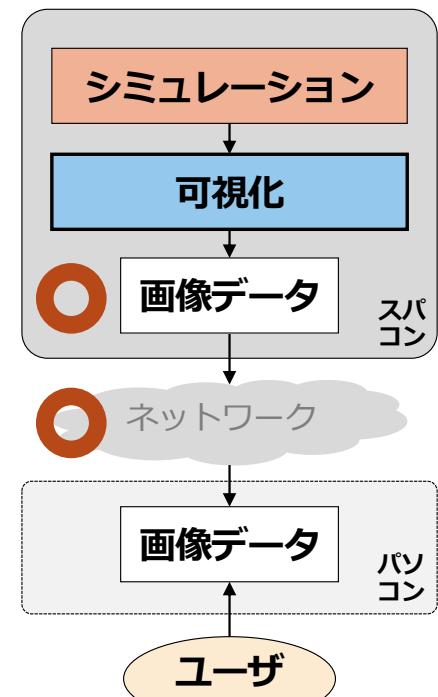
* H.Cilds, J.Bennett, C.Garth, and B.Hentschel. In situ visualization for computational science. IEEE CGA, Vol. 39, No. 06, pp. 76–85, 2019.

In-situ可視化

- Post-hoc可視化（事後可視化）
 - 可視化端末へのデータ転送
 - GPUを使った高速レンダリング
 - データI/O問題
 - 計算能力とI/Oコストのギャップ
 - データ保存頻度の低下
- In-situ可視化（その場可視化）
 - スパコン上で直接可視化
 - データI/Oコスト削減
 - 画像データ << 数値データ
 - 高時間分解能での可視化



Post-hoc可視化



In-situ可視化

先行研究

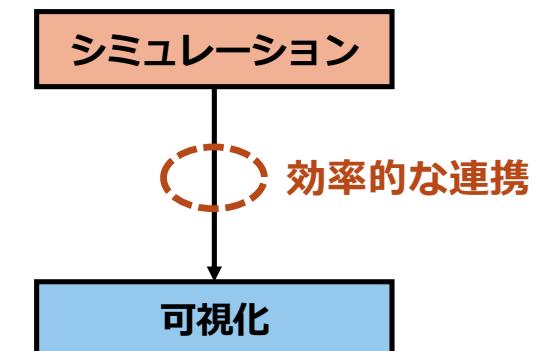
- **In-situ可視化基盤**

- 汎用的可視化ソフトウェアVTKをベースにした基盤

- ParaView + ParaView Catalyst [U. Ayachit et al., 2015]
- VisIt + Libsim [B. Whitlock et al., 2011]

- シミュレーション処理と可視化処理の連携を支援

- SENSEI [U. Ayachit et al., 2016]
 - In-situ可視化インターフェースの定義
- ADIOS2 [W. F. Godoy et al., 2020]
 - データI/O処理の簡便化
- Ascent [M. Larsen et al., 2017]
 - 簡単なシミュレーションと可視化機能を提供



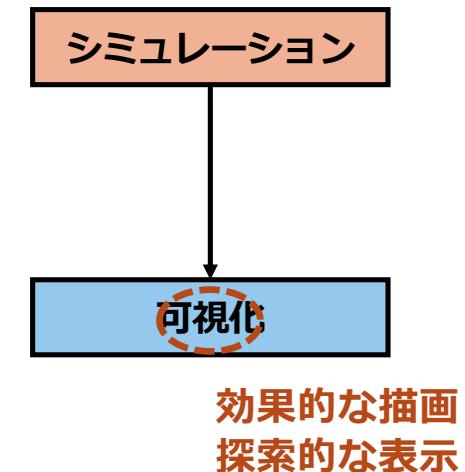
先行研究

- **高品質ソフトウェアレンダリング基盤**

- 並列レンダリング
 - 画像重畳
 - 並列ボリュームレンダリング
- ソフトウェアラスタライザ
 - OSPRay [I. Wald et al., 2016]
 - Embree [I. Wald et al., 2014]

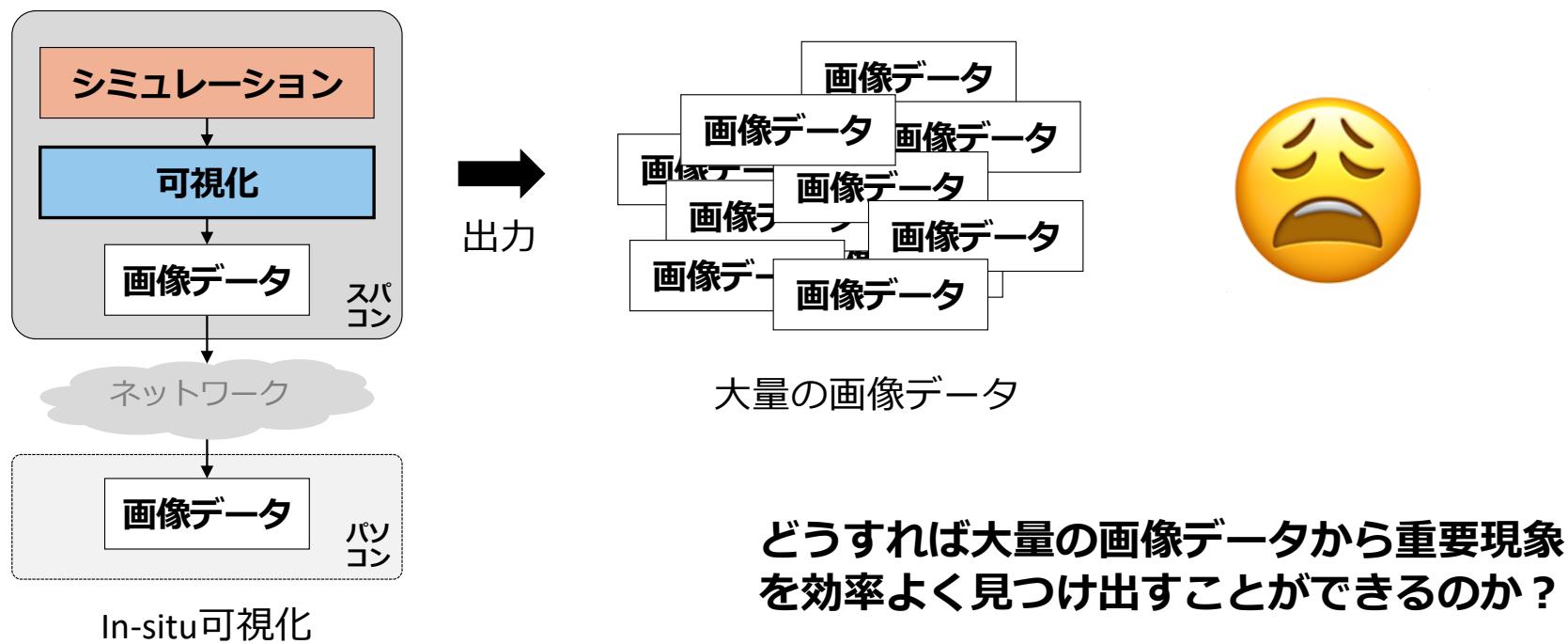
- **探索的In-situ可視化**

- 対話的なデータ探索実現のための技術
 - Explorable images [A. Tikhonova et al., 2010]
 - ParaView Cinema [J. Ahrens et al., 2015]
 - 4D street view [A. Kageyama et al., 2020]



In-situ可視化の問題

- In-situ可視化技術の発展により大量の画像が出力



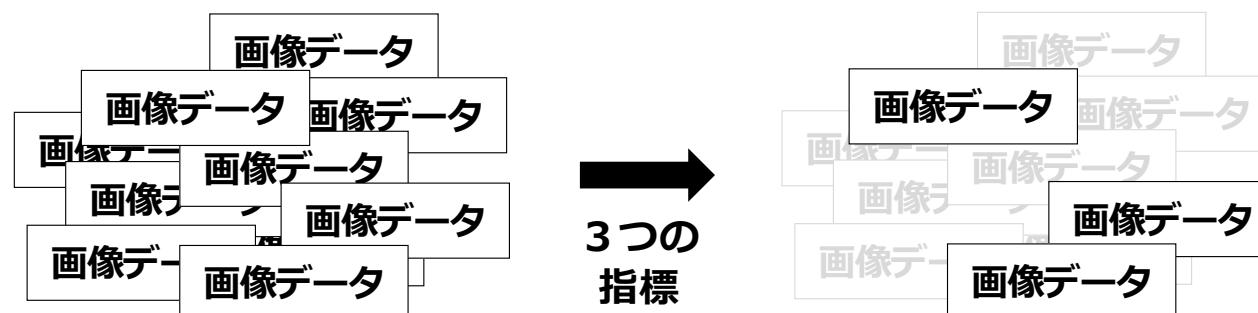
スマートIn-situ可視化

- 効率的な知見獲得を目指したIn-situ可視化

- シミュレーション中の重要な変化を見逃さない可視化
- 3つの指標

- 時間：重要な変化が発生する時間領域を推定
- 空間：重要な変化を撮影できる空間（視点）を推定
- 変数：状態を表す変数間の影響を考慮した重要度の推定

適応的時間選択
Adaptive Timestep Selection



Adaptive Timestep Selection for In-situ vis.

- Based on the amount of changes between the simulation timesteps
- Helpful for the data I/O and visualization cost savings
- Easy to find the important features related to the correct understanding of the underlying physical phenomena

Simulation



Visualization

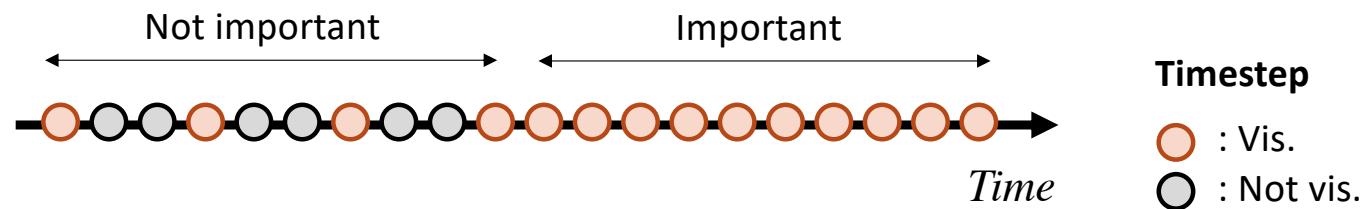


Adaptive Timestep Selection for In-situ vis.

1. Spatio-temporal variations between volumes

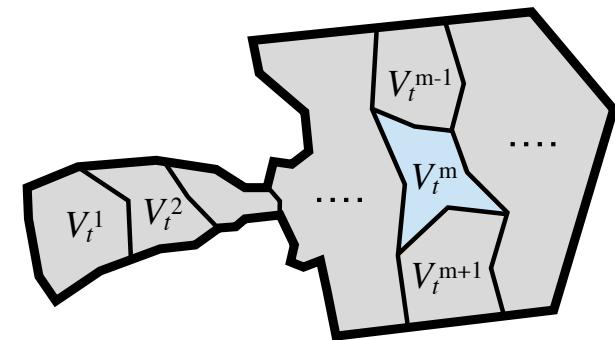
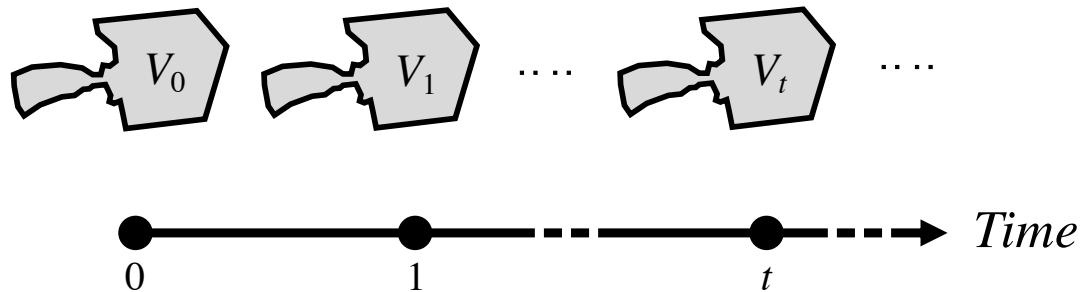


2. Adaptive timestep sampling



Spatio-temporal Variations

- Time-varying volume dataset



$$V_t = \{V_t^1, V_t^2, \dots, V_t^m, \dots, V_t^M\}$$

Spatio-temporal Variations

- Probability density function P of each sub-volume

- Kernel Density Estimation (KDE)

$$P_t^m(s) = \text{KDE}(V_t^m, s)$$

$$= \frac{1}{nh} \sum_x K\left(\frac{s - V_t^m(x)}{h}\right)$$

x : position

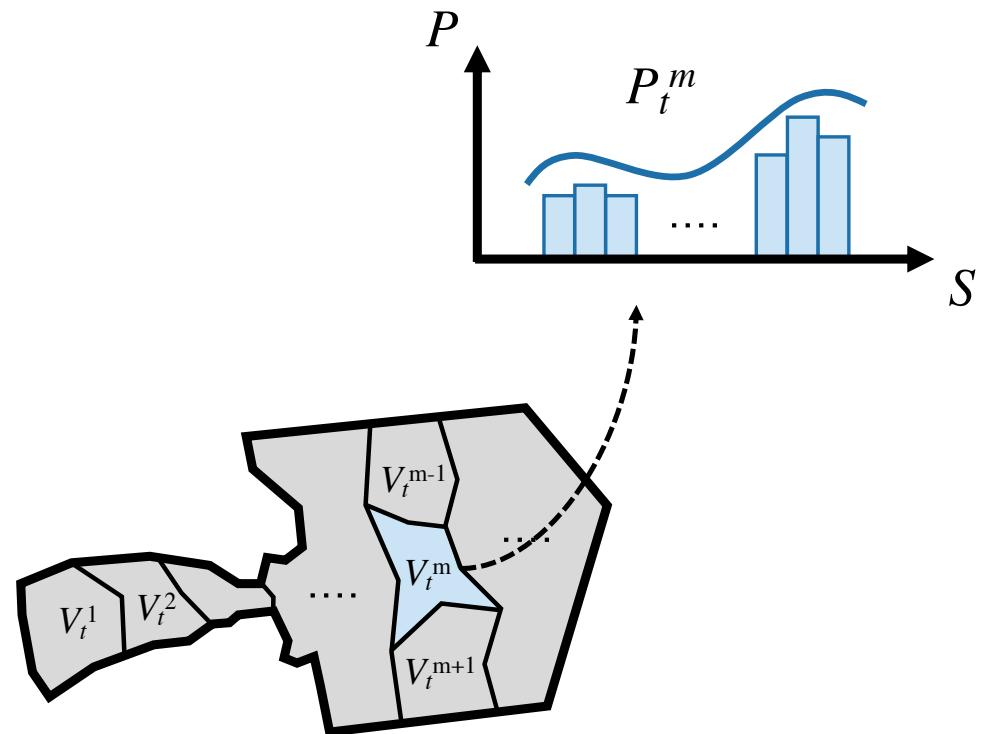
s : scalar data

n : number of scalar data

h : bandwidth $h = n^{-1/5}$

K : Gaussian basis function

$$K(a) = \frac{1}{\sqrt{2\pi}} e^{-a^2/2}$$



Spatio-temporal Variations

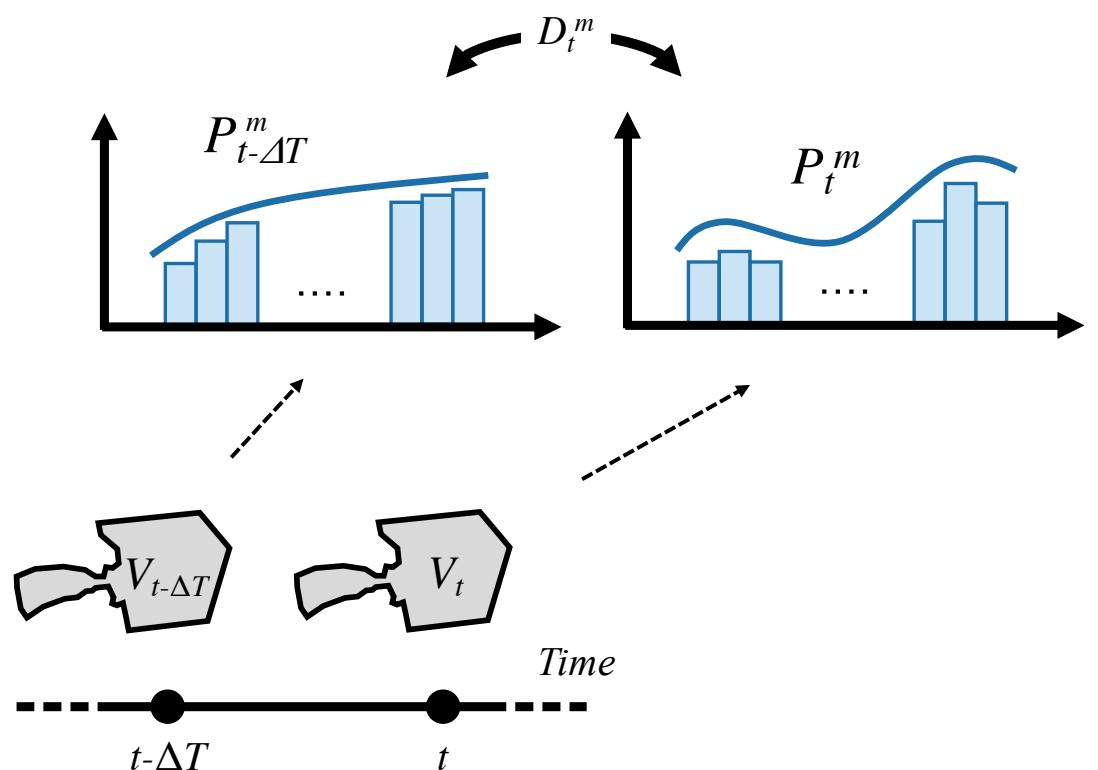
- Relative information divergence D between PDFs

- Kullback-Leibler (KL) divergence

$$D_t^m = KL(P_{t-\Delta T}^m, P_t^m)$$

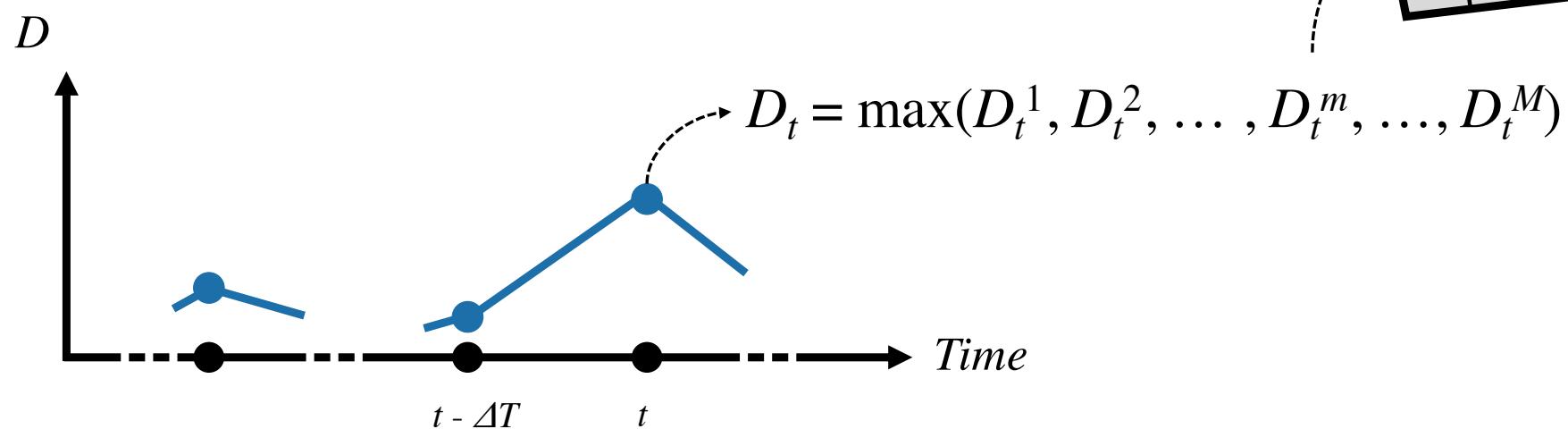
$$= \sum_{s=s_{min}}^{s_{max}} P_t^m(s) \log \frac{P_t^m(s)}{P_{t-\Delta T}^m(s)}$$

ΔT : time interval used for validating the divergence



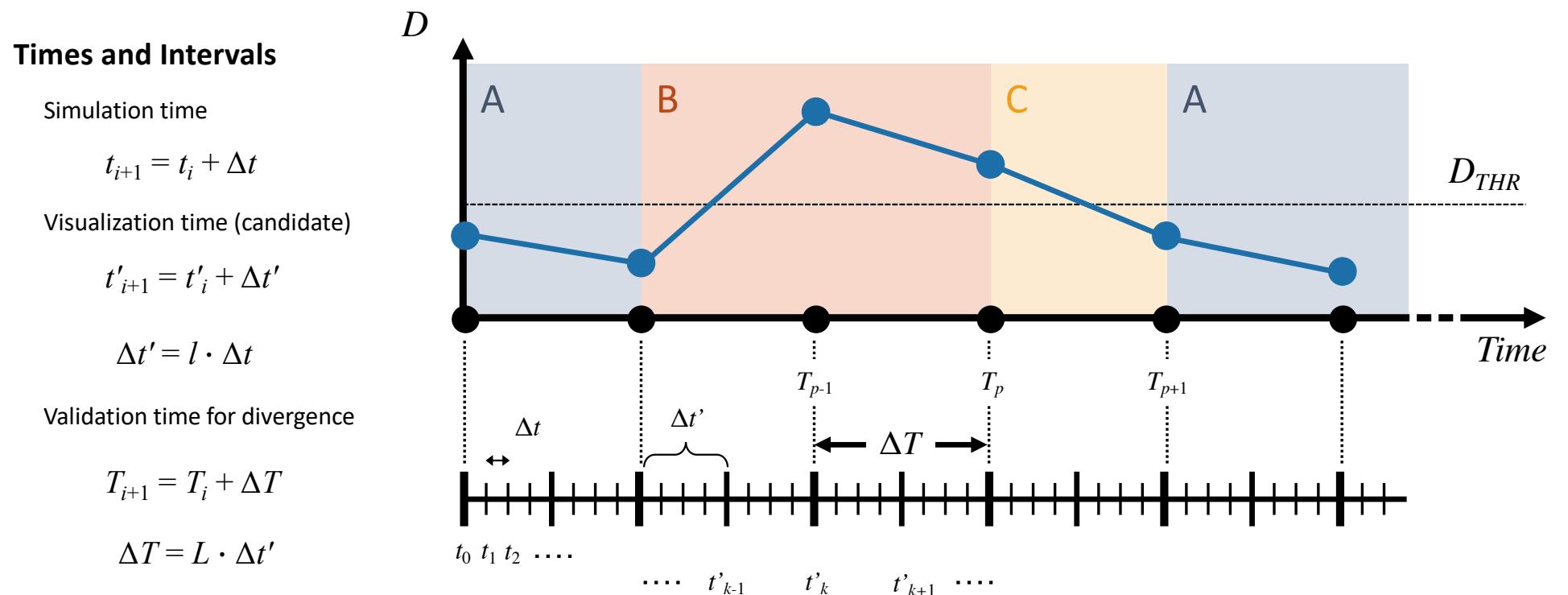
Spatio-temporal Variations

- Relative information divergence D between PDFs
 - Kullback-Leibler (KL) divergence



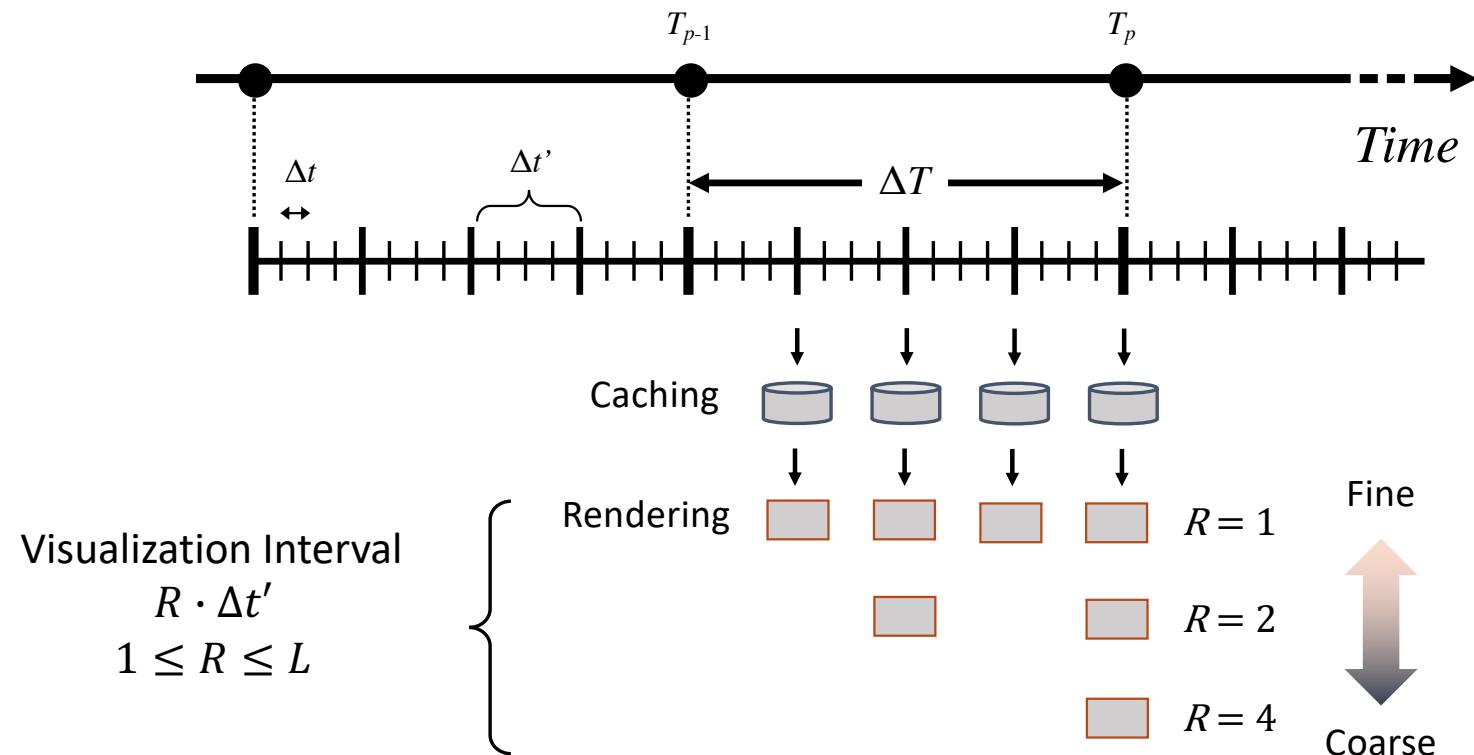
Adaptive Time Sampling

- Variation patterns and time intervals



Visualization Interval

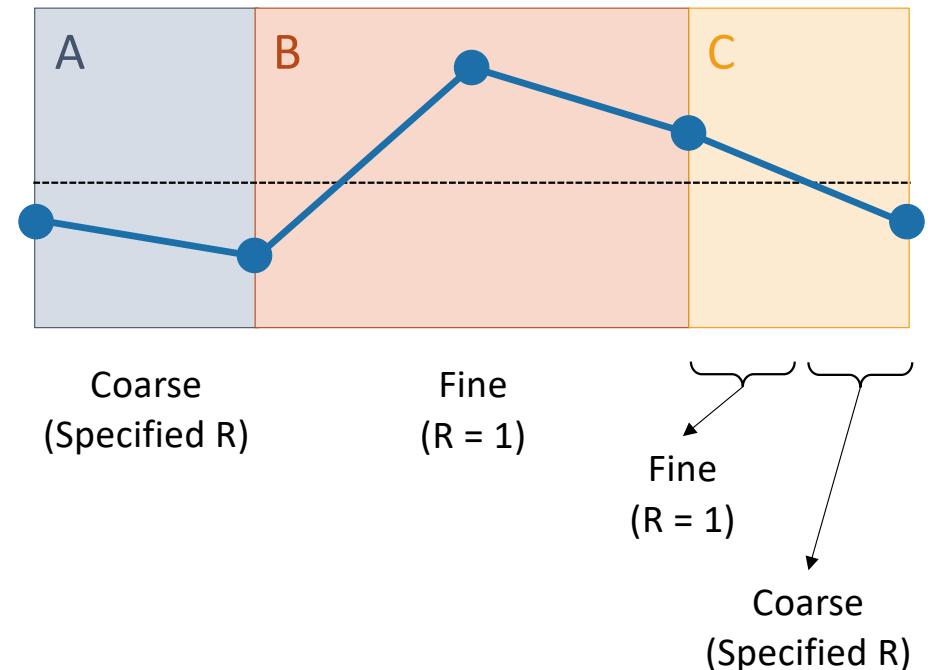
- Data caching and granularity R for low variation region



Visualization Interval

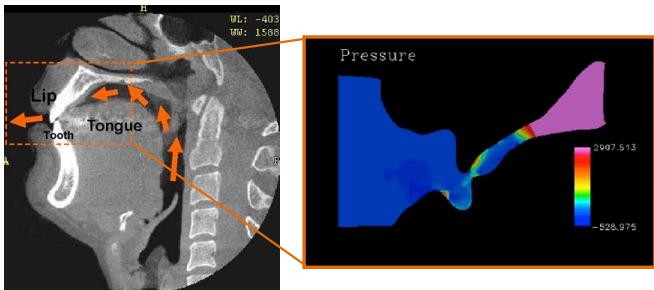
- **Granularity for each pattern**

- Pattern A
 - Coarse sampling
 - Using specified R
- Pattern B
 - Fine sampling
 - Using $R = 1$
- Pattern C
 - Combined sampling
 - Using $R = 1$ (fine) in the first half and specified R in the other half

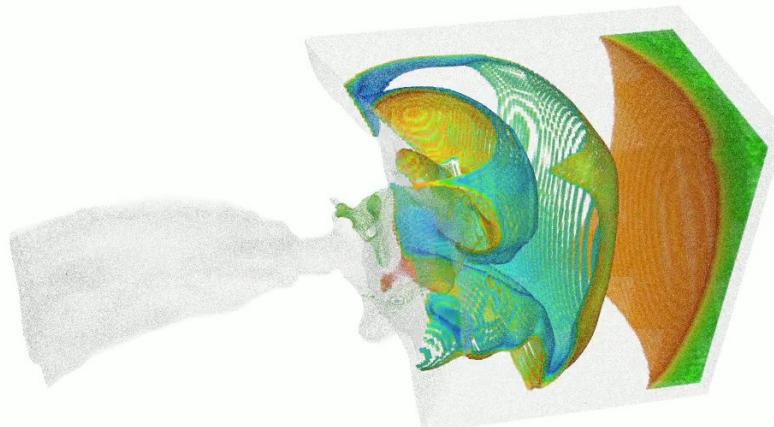


Experiment

- Numerical simulation of sibilant fricatives
 - Implicit compressible flow solver based on OpenFOAM



- Unstructured volumes
 - 3,197,279 hexahedrons
 - 48 sub-volumes
 - 20,000 timesteps



Experiment

- **Environment**
 - K pre-post cloud system
 - X86 based on-premise could system
 - Dual Intel Xeon Platinum 8168 (Skylake)
24cores/2.7GHz
 - 48 vCPUs on the VM for the evaluation
- **Software**
 - Simulation: OpenFOAM
 - Visualization framework: KVS¹
 - Software rasterizer: OS Mesa (LLVMpipe)
 - Rendering method:
 - Parallel particle-based rendering¹
 - 234 image composition²
- **Simple framework for in-situ vis.**³
 - ✓ Adaptive timestep selection
 - ✓ Multi-viewpoint rendering
 - ✓ Omni-directional rendering
 - ✓ Parallel irregular volume rendering

```
InSituVis::TimestepControlledAdaptor vis( MPI_COMM_WORLD );
vis.setImageSize( 1024, 1024 );
vis.setAnalysisInterval( 1 );
vis.setValidationInterval( L );
vis.setGranularity( R );
vis.setDivergenceThreshold( D_thr );
vis.setViewpoint( { {p} } );
vis.setPipeline( [] (Screen& s, Object& o) { ... } );
...
while ( runTime.run() )
{
    auto field = sim.exec(t);
    ...
    InSituVis::Converter conv( vis.world() );
    vis.put( conv.exec( field ) );
    vis.exec( {runTime.value(), runTime.timeIndex()} );
    ...
}
```

Example code

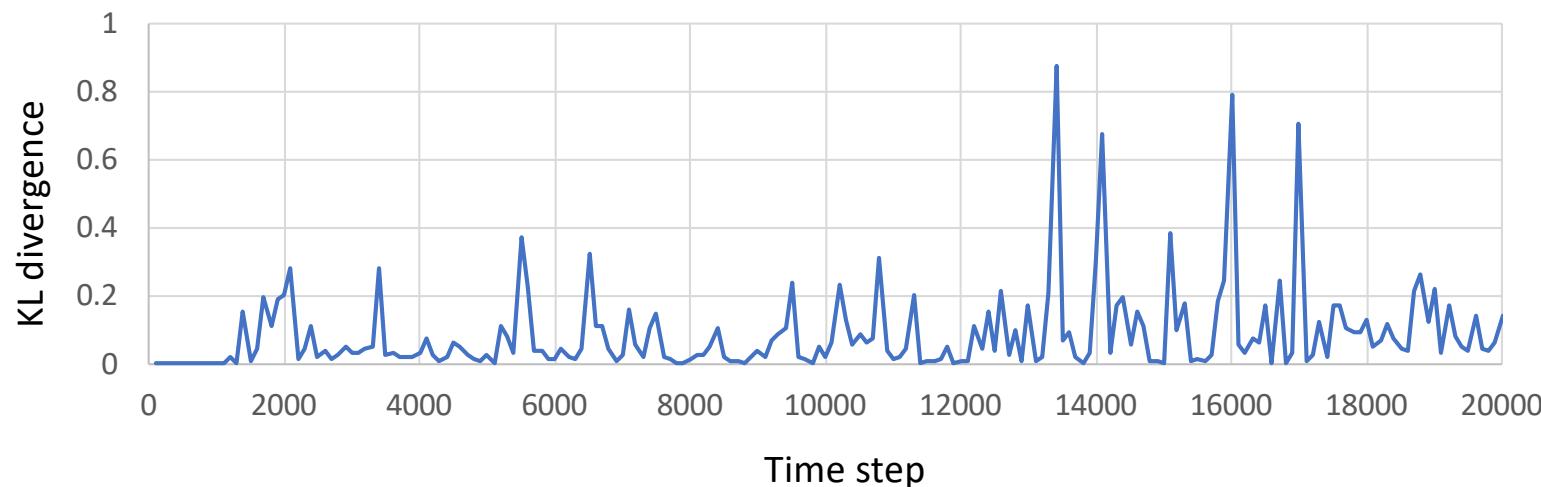
1) <https://github.com/naohisas/KVS>

2) <https://github.com/avr-aics-riken/234Compositor>

3) <https://github.com/vizlab-kobe/InSituVis> (private)

Result

- **Changes of KL divergence (D)**
 - Simulation timesteps: 20,000
 - Validation time interval for KL divergence ($\Delta T / \Delta t$) : 100
 - Visualization timestep intervals (l) : 5
 - Validation timestep intervals (L) : 20



Result

- **Number of vis. images and reduction rate**

- Total number of stacked data: 4,000

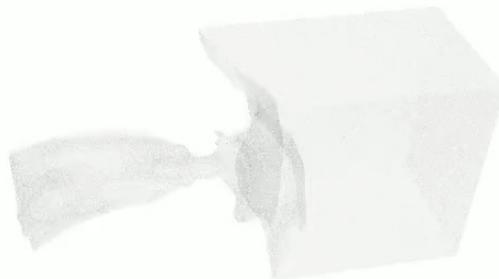
D _{THR}	R=5 (fine sampling)	R=10 (coarse sampling)
0	4,000 (0%)	4,000 (0%)
0.01	3,452 (13.7%)	3,384 (15.4%)
0.02	3,100 (22.5%)	2,896 (21.6%)
0.03	2,856 (28.6%)	2,616 (34.6%)
0.04	2,496 (37.6%)	2,288 (42.8%)
0.05	2,064 (48.4%)	1,892 (52.7%)

D_{THR}: Threshold for the KL divergence (D)

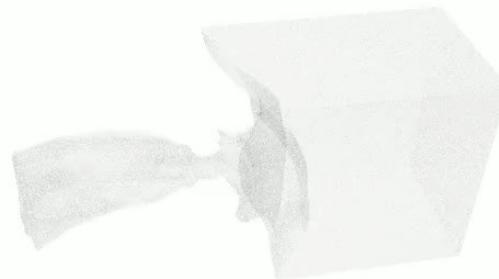
R: Granularity for the time sampling for A and C interval patterns

Result

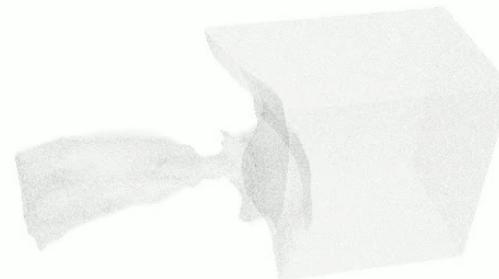
- In-situ visualization results w/ and w/o our method
 - $D_{THR} = 0.05$



Original
(4,000 images)



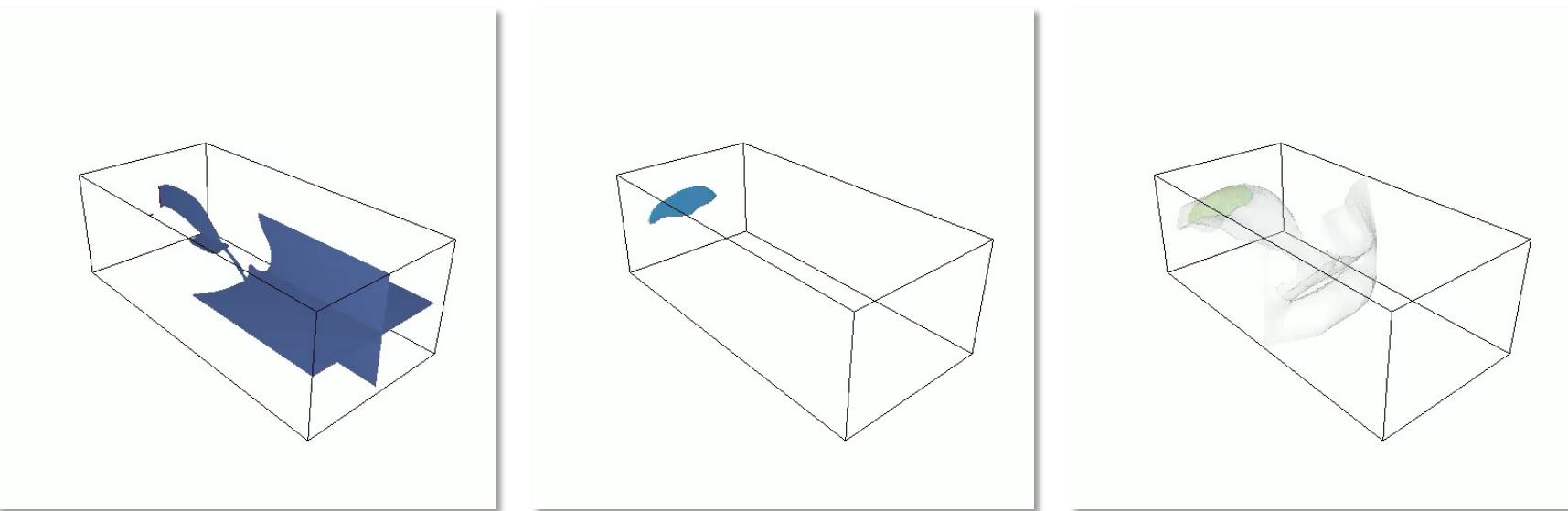
Our method with R=5
(2,064 images)



Our method with R=10
(1,892 images)

参考) 可視化例

- **単視点**



スライス断面（2平面）

等値面（3閾値）

ボリュームレンダリング（参考）

参考) 処理時間

• 計測環境

✓ CPU: Intel Xeon 6238R (2.20GHz, 28core) x 2, RAM: 32GBx12 (384GB)

• 結果

✓ 処理時間 (秒)

8 proc.	Pipeline			Rendering			Composition			Visualization
	Min	Max	Ave	Min	Max	Ave	Min	Max	Ave	Total (max)
Ortho slice	0.003	0.005	0.004	0.035	0.342	0.191	0.085	0.405	0.242	0.752
Isosurface	0.002	0.004	0.003	0.201	0.493	0.332	0.079	0.373	0.240	0.870
Volume rendering	0.620	0.883	0.746	0.755	2.748	1.513	5.553	8.039	7.053	11.670

48 proc.	Pipeline			Rendering			Composition			Visualization
	Min	Max	Ave	Min	Max	Ave	Min	Max	Ave	Total (max)
Ortho slice	0.003	0.008	0.004	0.013	0.746	0.177	0.005	1.337	0.845	2.090
Isosurface	0.002	0.006	0.003	0.049	0.925	0.311	0.003	1.500	0.891	2.431
Volume rendering	0.891	3.380	2.642	1.060	3.835	1.923	51.922	56.194	54.145	63.409

参考) 処理時間

• 計測環境

✓ 富岳 (並列 : 8, 12, 48, 4096)

12 proc.	Pipeline			Rendering			Composition			Visualization
	Min	Max	Ave	Min	Max	Ave	Min	Max	Ave	Total (max)
Ortho slice	0.031	0.078	0.052	0.157	1.726	0.889	0.005	1.618	0.772	3.422
Isosurface	0.034	0.050	0.039	0.174	4.989	0.890	0.007	4.853	3.156	9.892
Volume rendering	3.046	3.518	3.306	0.452	1.352	0.657	0.244	1.292	0.801	6.162

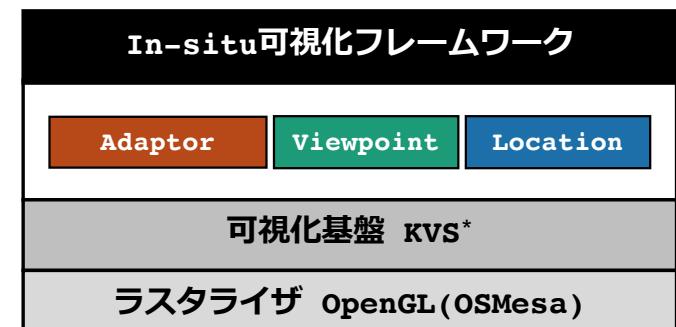
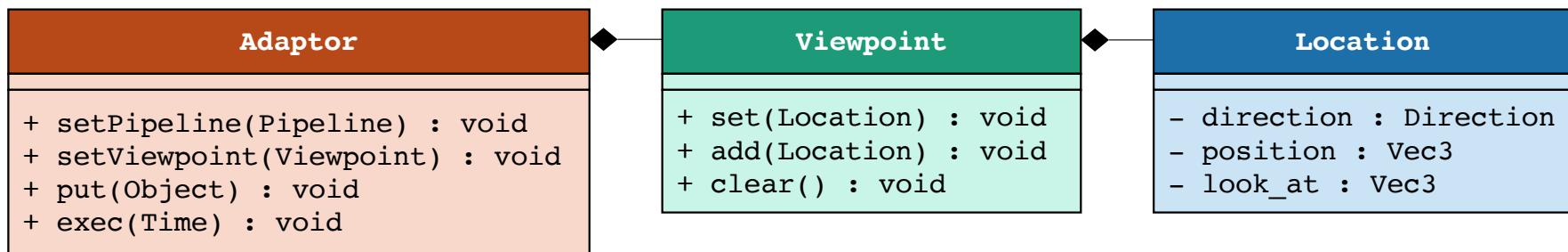
8 proc.	Pipeline			Rendering			Composition			Visualization
	Min	Max	Ave	Min	Max	Ave	Min	Max	Ave	Total (max)
Ortho slice	0.003	0.005	0.004	0.035	0.342	0.191	0.085	0.405	0.242	0.752
Isosurface	0.002	0.004	0.003	0.201	0.493	0.332	0.079	0.373	0.240	0.870
Volume rendering	0.620	0.883	0.746	0.755	2.748	1.513	5.553	8.039	7.053	11.670

48 proc.	Pipeline			Rendering			Composition			Visualization
	Min	Max	Ave	Min	Max	Ave	Min	Max	Ave	Total (max)
Ortho slice	0.003	0.008	0.004	0.013	0.746	0.177	0.005	1.337	0.845	2.090
Isosurface	0.002	0.006	0.003	0.049	0.925	0.311	0.003	1.500	0.891	2.431
Volume rendering	0.891	3.380	2.642	1.060	3.835	1.923	51.922	56.194	54.145	63.409

参考) In-situ可視化フレームワーク

• C++クラスライブラリ

- **Adaptor** クラス
 - シミュレーション処理と可視化処理の接続
 - 可視化パイプラインの設定と並列実行の制御
- **Viewpoint** クラス
 - 視点の管理
- **Location** クラス
 - 視点・視線の設定



* N.Sakamoto and K.Koyamada. KVS: A simple and effective framework for scientific visualization. Journal of Advanced Simulation in Science and Engineering, Vol.2, No.1, pp.76-95, 2015.

参考) 実装例

- 等値面可視化
 - 単視点

```
// 可視化パイプラインの設定
auto pipeline = [](Screen& screen, Object& object )
{
    auto volume = VolumeObject::DownCast( object );
    auto polygon = Isosurface( volume );
    auto renderer = PolygonRenderer();
    screen.add( polygon, renderer );
}

// 視点（単方向）の設定
Location location( {0,0,12}, UniDir );
Viewport viewport( location );

// シミュレーションコードとの接続アダプタの設定
Adaptor adaptor( MPI_COMM_WORLD );
adaptor.setImageSize( 512, 512 );
adaptor.setOutputDirectory( "Output" );
adaptor.setAnalysisInterval( 5 );
adaptor.setPipeline( pipeline );
adaptor.setViewport( viewport );
```

```
// シミュレーションコードとの接続
adaptor.initialize();
while ( t < T )
{
    auto data = Simulation( t );
    auto object = Conversion( data );

    adaptor.put( object );
    adaptor.exec( t );
    ++t;
}
adaptor.finalize();
```

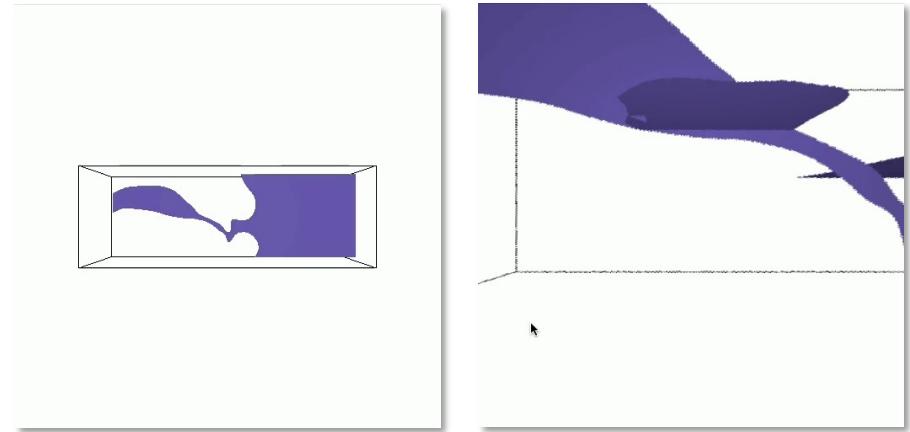
参考) 実装例

- 等値面可視化
 - 多視点

```
// 可視化パイプラインの設定
auto pipeline = [](Screen& screen, Object& object )
{
    auto volume = VolumeObject::DownCast( object );
    auto polygon = Isosurface( volume );
    auto renderer = PolygonRenderer();
    screen.add( polygon, renderer );
}

// 視点（単方向）の設定
Location location( {0,0,12}, UniDir );
Viewport viewport( location );

// シミュレーションコードとの接続アダプタの設定
Adaptor adaptor( MPI_COMM_WORLD );
adaptor.setImageSize( 512, 512 );
adaptor.setOutputDirectory( "Output" );
adaptor.setAnalysisInterval( 5 );
adaptor.setPipeline( pipeline );
adaptor.setViewport( viewport );
```



// 多視点（等間隔配置）の設定

```
CubicViewport viewport;
viewport.setDims( {3,3,3} );
viewport.create( UniDir ); // UniDir or OmniDir
```

// 多視点（球面配置）の設定

```
SphericalViewport viewport;
viewport.setDims( {3,3,3} );
viewport.create( UniDir ); // UniDir or OmniDir
```

参考) 実装例

- 等値面可視化
 - 適応的時間選択アダプタ

```
// 可視化パイプラインの設定
auto pipeline = [](Screen& screen, Object& object )
{
    auto volume = VolumeObject::DownCast( object );
    auto polygon = Isosurface( volume );
    auto renderer = PolygonRenderer();
    screen.add( polygon, renderer );
}

// 視点（単方向）の設定
Location location( {0,0,12}, UniDir );
Viewport viewport( location );

// シミュレーションコードとの接続アダプタの設定
Adaptor adaptor( MPI_COMM_WORLD );
adaptor.setImageSize( 512, 512 );
adaptor.setOutputDirectory( "Output" );
adaptor.setAnalysisInterval( 5 );
adaptor.setPipeline( pipeline );
adaptor.setViewport( viewport );
```



```
// 適応的時間選択の設定
TimestepControlledAdaptor adaptor( MPI_COMM_WORLD );
adaptor.setImageSize( 512, 512 );
adaptor.setOutputDirectory( "Output" );
adaptor.setAnalysisInterval( 5 );
adaptor.setPipeline( pipeline );
adaptor.setViewport( viewport );
adaptor.setValidationInterval( 10 );
```

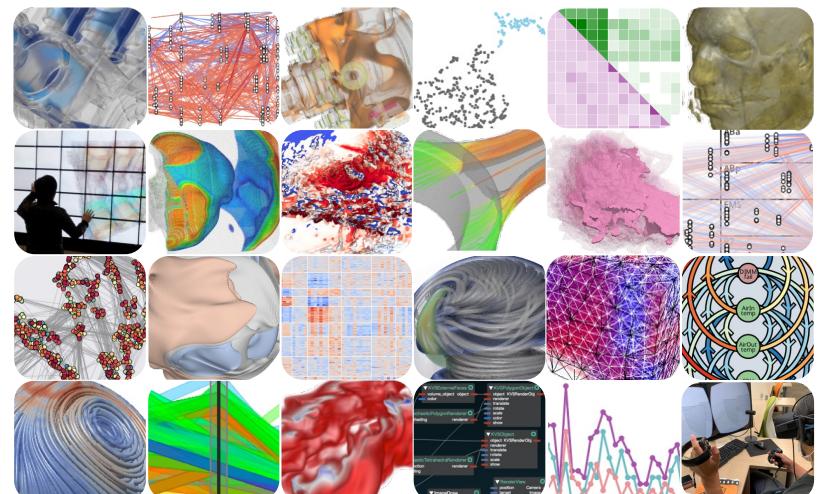
まとめ

・スマートIn-situ可視化

- ・効率的な知見獲得を目指したIn-situ可視化
- ・3つの指標：時間・空間・変数
- ・適応的時間選択

今後の予定

- ・スマートIn-situ可視化フレームワーク
- ・適応的空间選択（最適視点推定）
- ・In-situ可視化後の探索性能向上



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

<https://github.com/vizlab-kobe/InSituVis>

謝辞

- **科研費**

- 20H04194 : 効率的な知見獲得を目指した大規模数値計算向けスマート In-situ可視化 (基B・代表)
- 19H03976 : 構音シミュレーターと医療情報を用いた構音障害の物理的病理の推定 (基B・分担)

- **協力**

- 吉永司助教 (豊橋技術科学大学)
- 野崎一徳准教授 (大阪大学歯学部附属病院)