Smart In-situ Visualization for Large-scale Numerical Simulations Aiming at Efficient Knowledge Acquisition

Naohisa Sakamoto

naohisa.sakamoto@people.kobe-u.ac.jp

Kobe University / RIKEN R-CCS
Kobe, Japan

The 4th China-Japan Joint Visualization Workshop
July 24, Wuhan, China (Online)

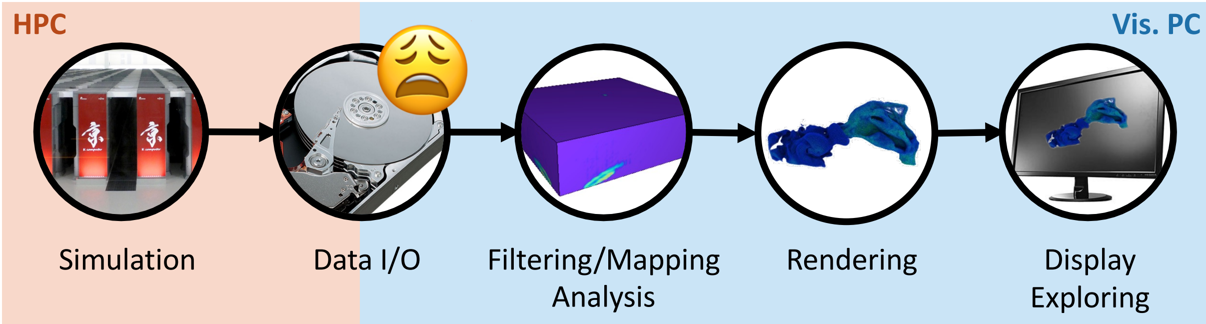
High Performance Computing Systems

- **Supercomputer Fugaku**
  - 158,976 processing nodes
  - 537 PFLOPS
  - Awarded
    - Top500, HPCG, HPL-AI, Graph500

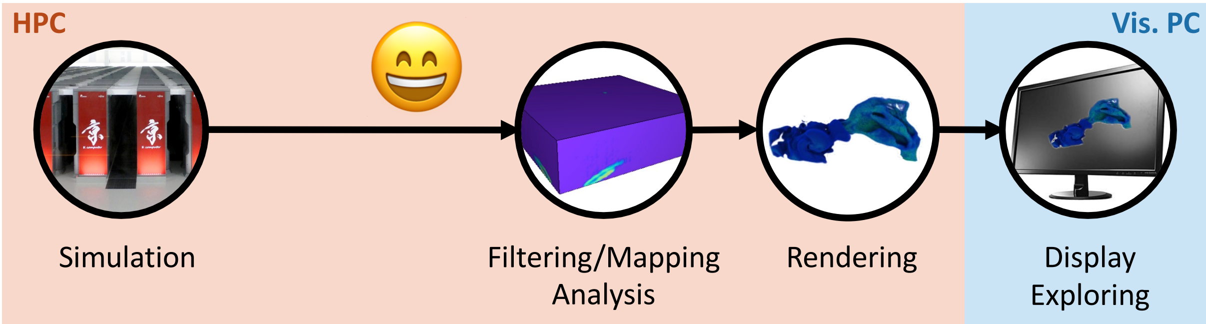
➡ **Visualization and analysis tasks even more challenging**

In-situ Visualization

- **Post-hoc vis.**
  - Transfer data to vis. PC
  - Rendering on vis. PC (with GPUs)
  - Data I/O for sim. results is expensive



- **In-situ vis.**
  - Avoiding data I/O
  - Rendering directly from sim. results on HPC

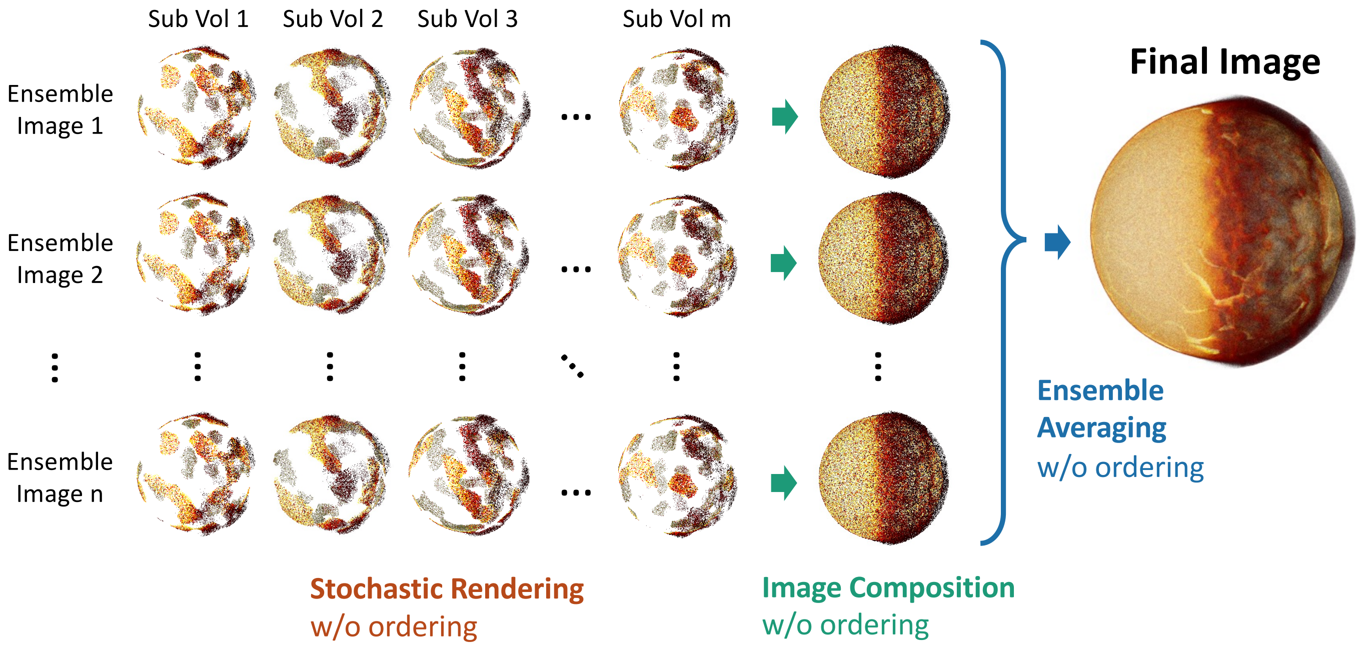


Parallel Rendering

- **Order-independent technique for volume rendering**

CFD simulation: Magnus force acting on a rotation ball placed in a uniform flow
- 19M prism elements
- 256 sub-volumes

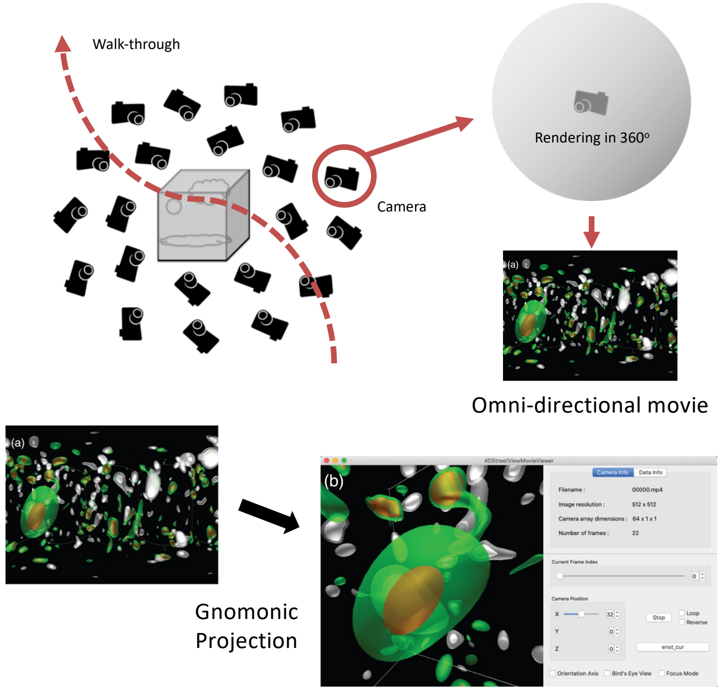
# of ensemble images

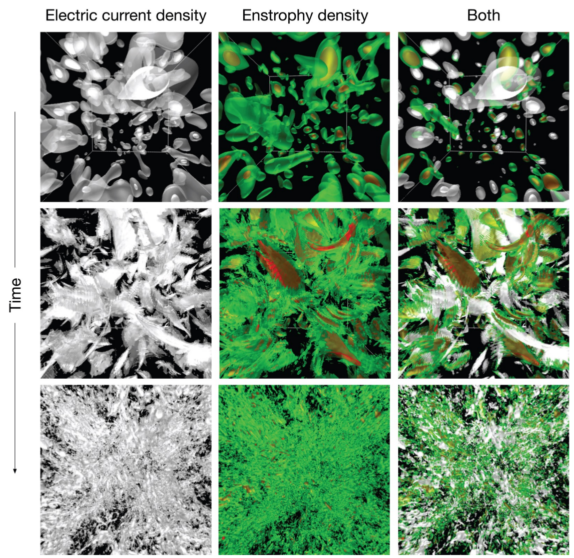


Y. Yamaoka, K. Hayashi, N. Sakamoto, J. Nonaka, A Memory Efficient Image Composition-based Parallel Particle Based Volume Rendering, JASSE, Vol.6, Issue.1, pp.1-10, 2019.3

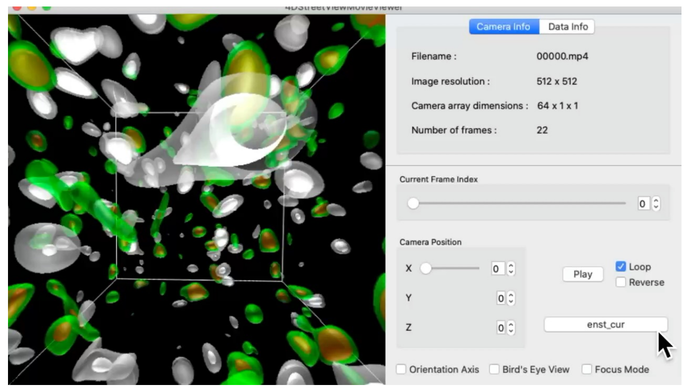
Interactive Exploration for In-situ Vis.

- **Multi-viewpoint omni-directional rendering**





In-situ visualization of the hall HMD simulations: Semi-transparent isosurfaces for two different physical quantities.

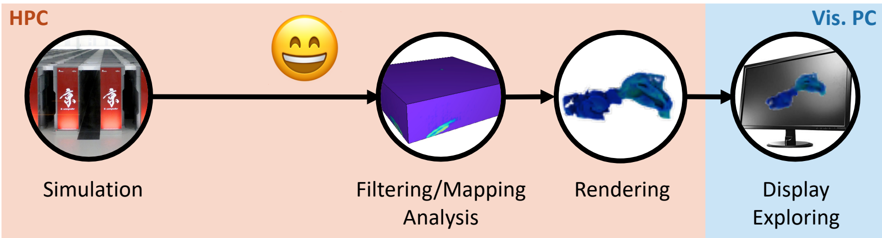


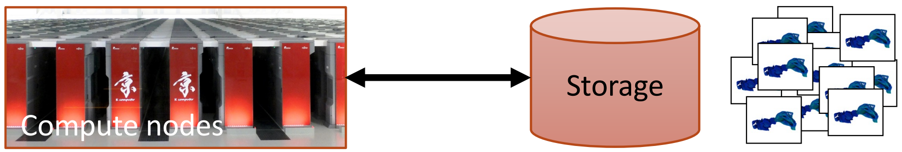
Post-hoc data exploration for in-situ visualization results based on mutli-viewpoint omnidirectional movies.

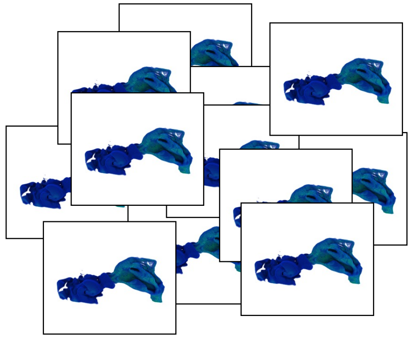
Akira Kageyama, Naohisa Sakamoto, 4D Street View: A Video-based Visualization Method, PeerJ Computer Science, 6:e305, DOI:10.7717/peerj-cs.305, 2020.9.

In-situ Visualization on HPC

- **Blind execution**
  - Pre-defined visualization parameters
    - File staging-based processing
    - Rendering for time-varying datasets
    - Large number of output images



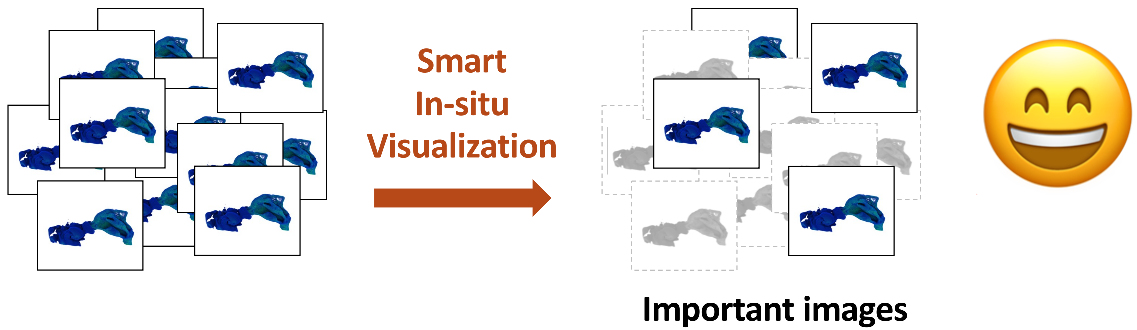




Difficult to gain insight into the data and facilitate its understanding

Smart In-situ Visualization

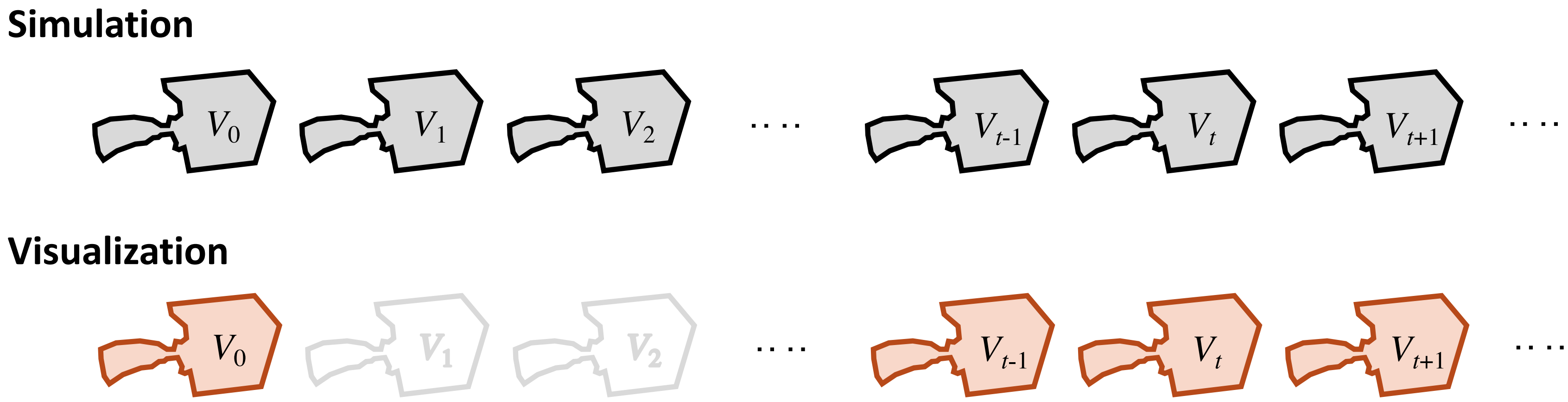
- **Time to discover**
  - Reducing the time required to obtain scientific knowledge from the numerical simulation results
  - Identify the spatio-temporal region to be visualized by automatically evaluating the state changes for the simulation dataset to be calculated
- **Adaptive sampling for smart in-situ vis.**
  - Time
  - Space



Japan Society for the Promotion of Science (JSPS), Grant-in-Aid for Scientific Research (B) 20H04194, Smart in-situ visualization for large-scale numerical simulations aiming at efficient knowledge acquisition

In-situ Timestep Selection

- 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

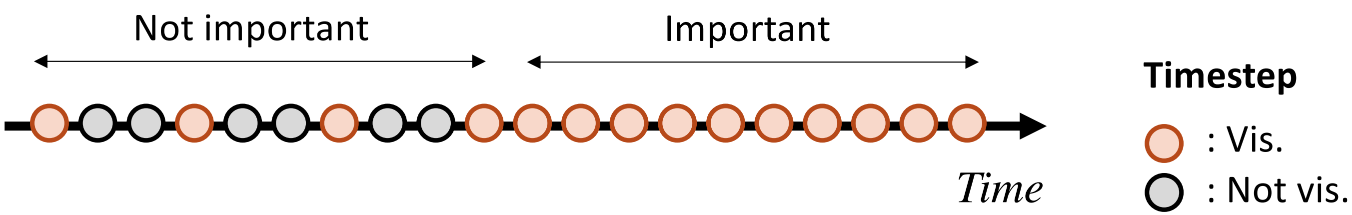


Y. Yamaoka, K. Hayashi, N. Sakamoto, J. Nonaka, In-Situ Adaptive Timestep Control and Visualization based on the Spatio-Temporal Variations of the Simulation Results, ISAV2019, pp.12-16, 2019.11

In-situ Timestep Selection

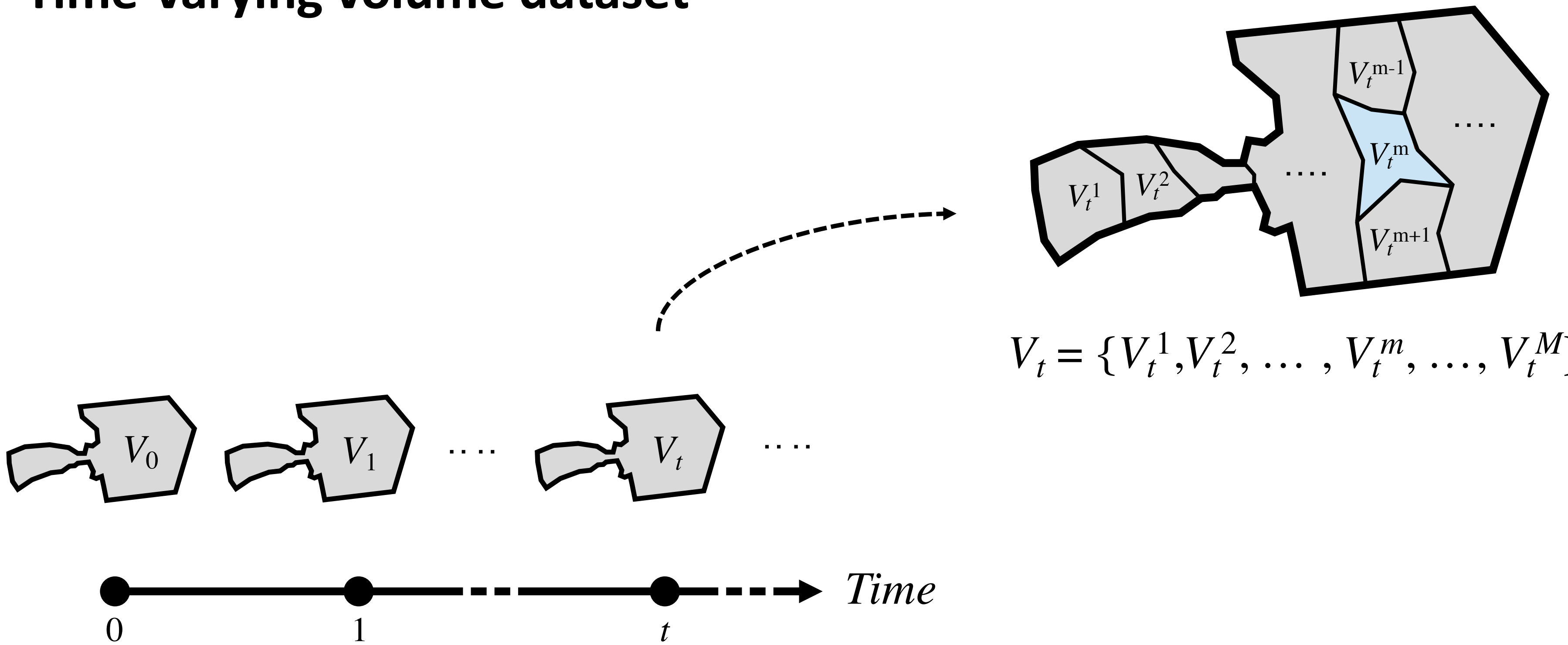
1. **Spatio-temporal variations between volumes**

2. **Adaptive timestep sampling**



Spatio-temporal Variations

- **Time-varying volume dataset**



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

Spatio-temporal Variations

- **Probability density function P of each sub-volume**
  - Kernel Density Estimation (KDE)

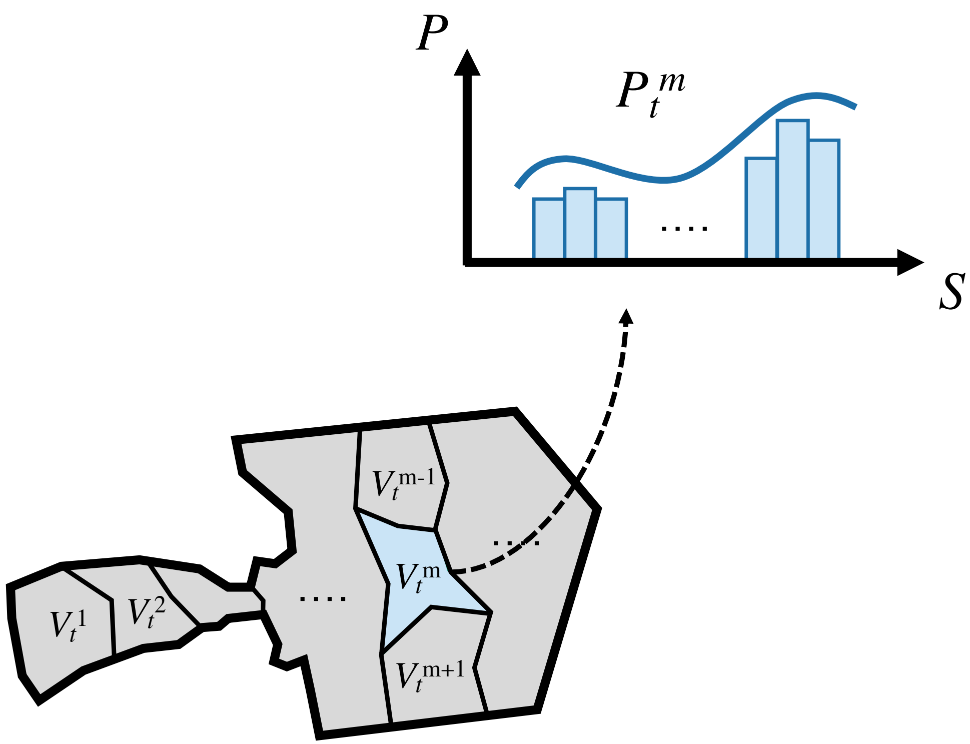
$$P_t^m(s) = KDE(V_t^m, s)$$

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

x : position
s : scalar data
n : number of scalar data
h : bandwidth $h = n^{-1/5}$

K: Gaussian basis function

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



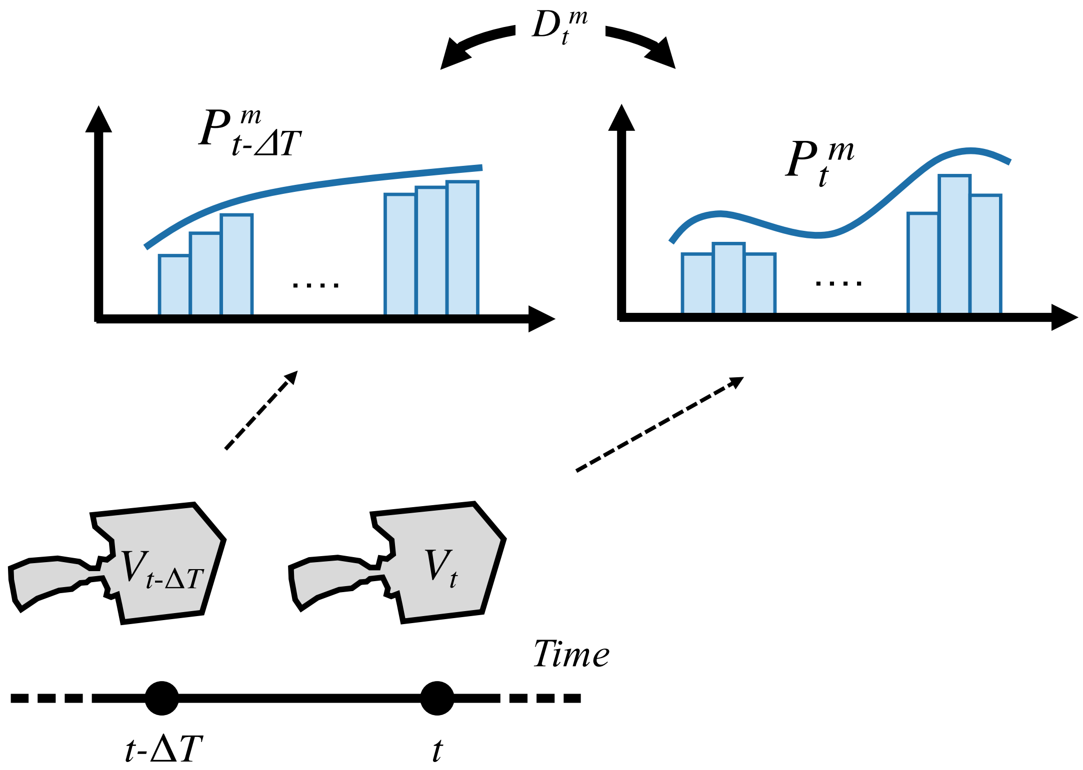
Spatio-temporal Variations

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

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

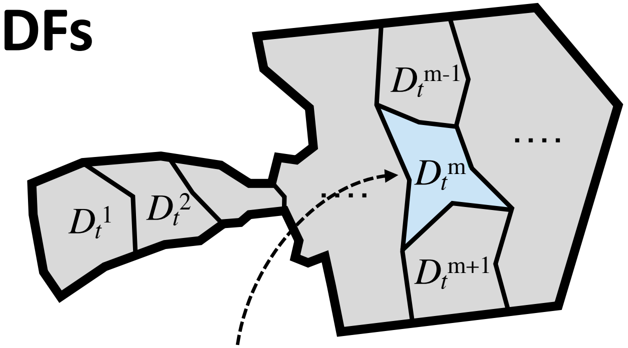
$$= \sum_{s=s_{min}}^{s_{max}} P_t^m(s) \log \frac{P_t^m(s)}{P_{t-\Delta T}^m(s)}$$

ΔT: time interval used for validating the divergence

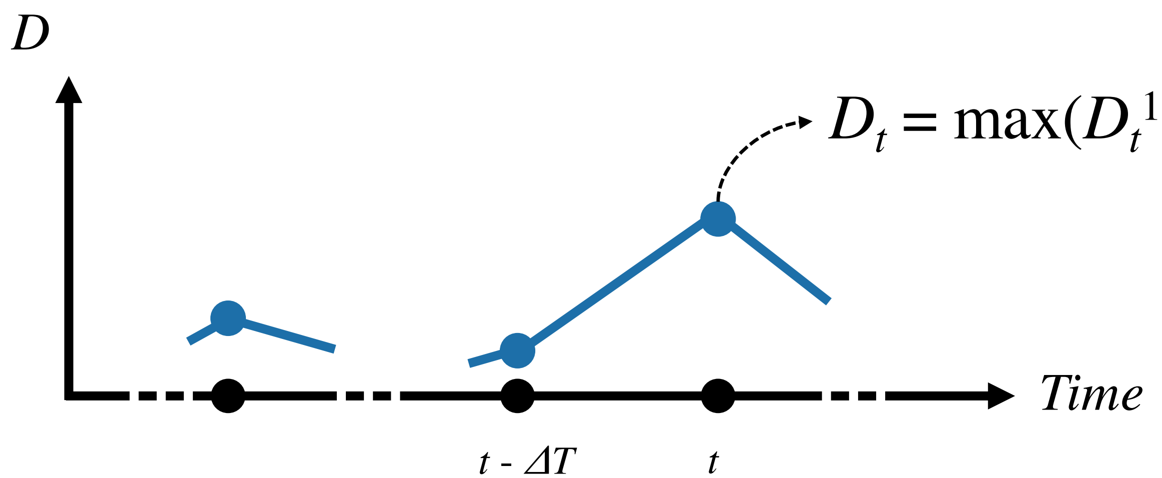


Spatio-temporal Variations

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



$$D_t = \max(D_t^1, D_t^2, \ldots, D_t^m, \ldots, D_t^M)$$



Adaptive Time Sampling

- **Variation patterns and time intervals**

Times and Intervals

Simulation time

$$t_{i+1} = t_i + \Delta t$$

Visualization time (candidate)

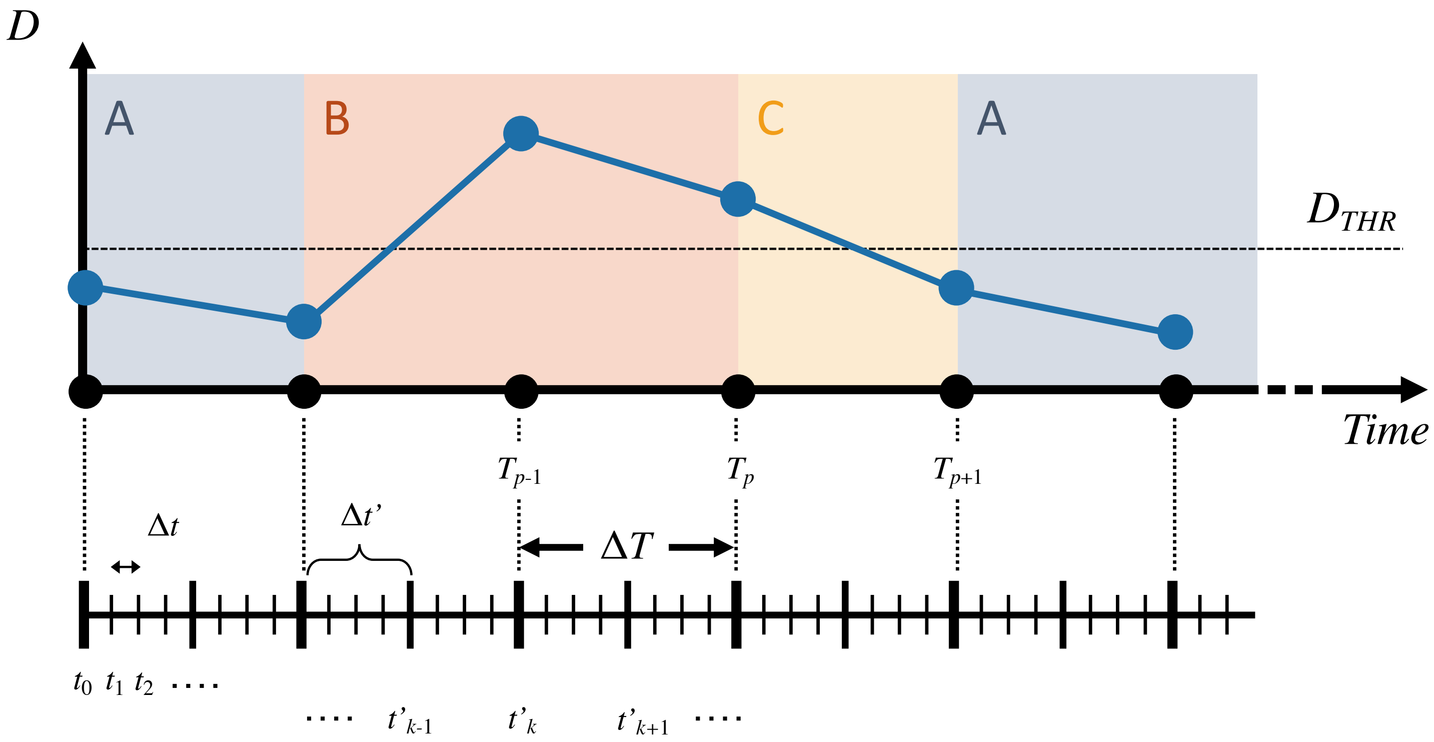
$$t'_{i+1} = t'_i + \Delta t'$$

$$\Delta t' = l \cdot \Delta t$$

Validation time for divergence

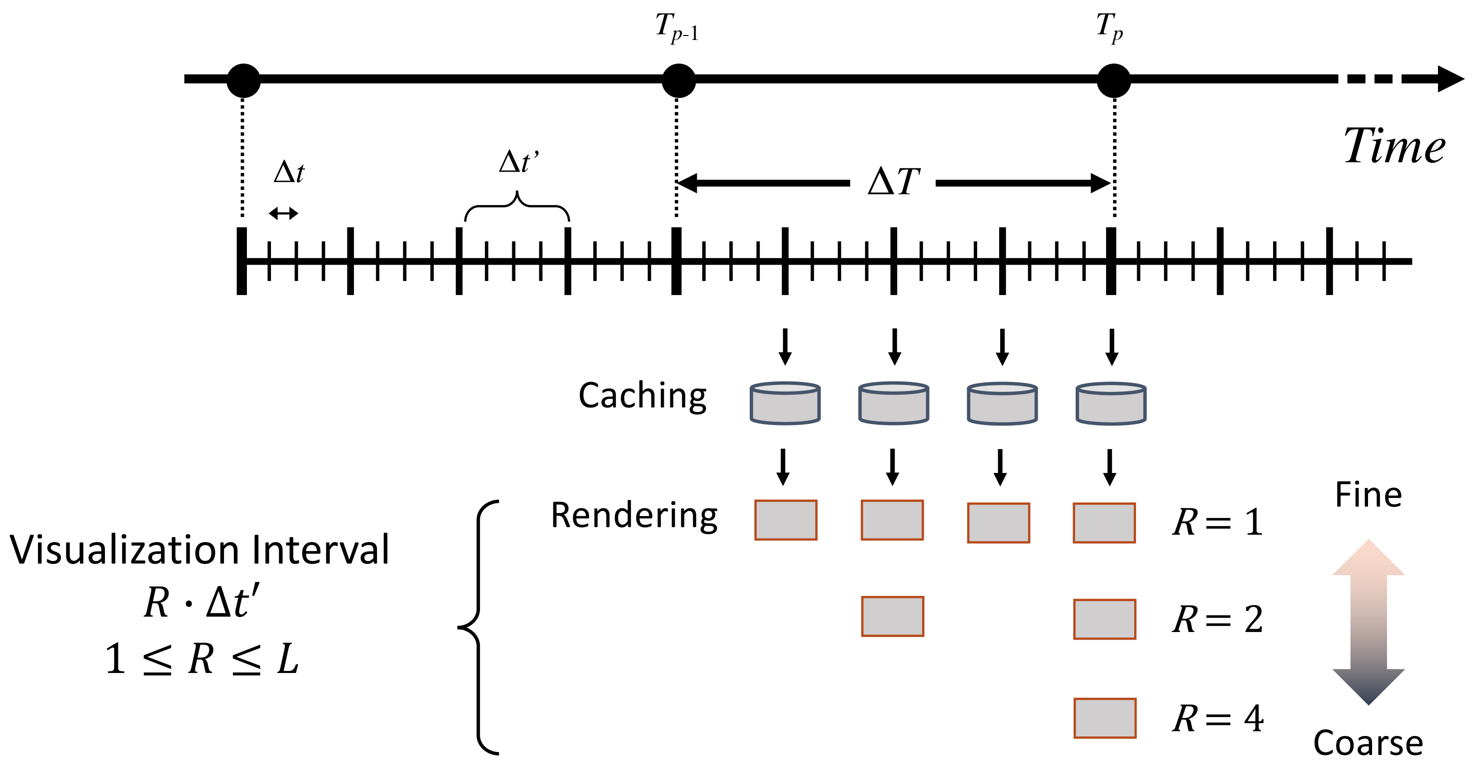
$$T_{i+1} = T_i + \Delta T$$

$$\Delta T = L \cdot \Delta t'$$



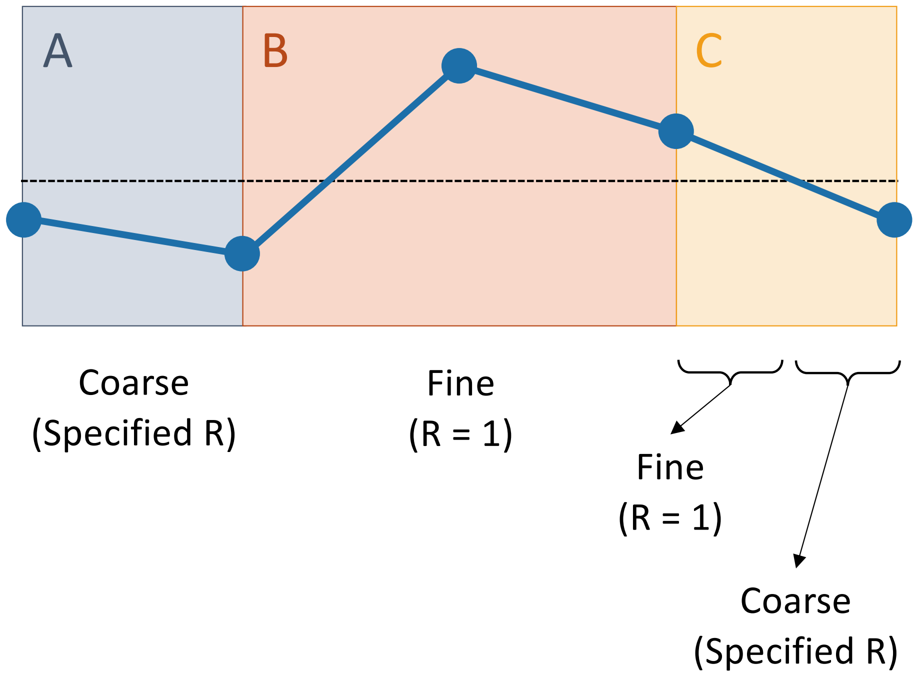
Visualization Interval

- **Data caching and granularity R for low variation region**



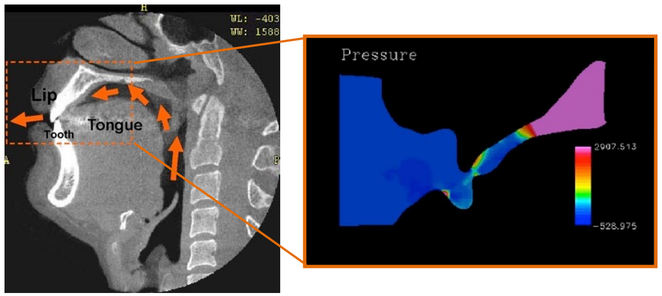
Visualization Interval

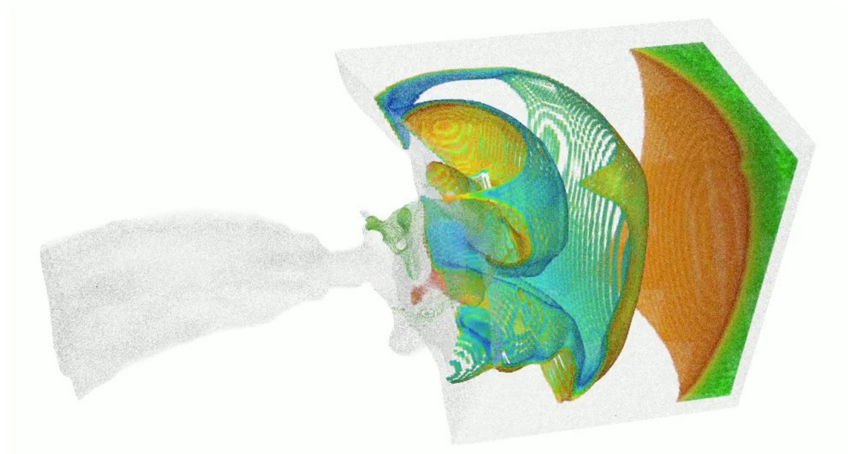
- **Granularity for each pattern**
  - Pattern A
    - Coarse sampling
    - Using specified R
  - Pattern B
    - Fine sampling
    - Using R = 1
  - Pattern C
    - Combined sampling
    - Using R = 1 (fine) in the first half and specified R in the other half



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

Tsukasa Yoshinaga, Kazunori Nozaki, and Shigeo Wada. Experimental and numerical investigation of the sound generation mechanisms of sibilant fricatives using a simplified vocal tract model. Physics of Fluids 30, 3, 2018).

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: KVS[1]
  - Software rasterizer: OSMesa (LLVMpipe)
  - Rendering method:
    - Parallel particle-based rendering[1]
    - 234 image composition[2]

- **Simple framework for in-situ vis.[3]**
  - ✓ Adaptive timestep selection
  - ✓ Multi-viewpoint rendering
  - ✓ Omni-directional rendering
  - ✓ Parallel irregular volume rendering

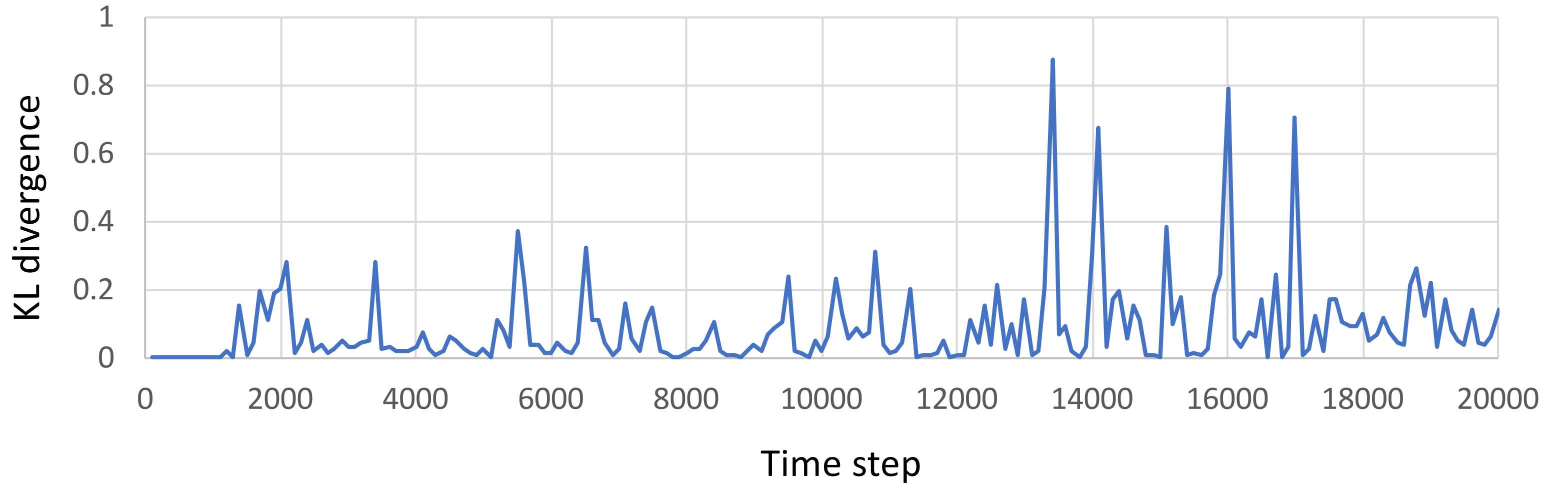
```
InSituVis::TimestepControlledAdaptor vis( MPI_COMM_WORLD );
vis.setImageSize( 1024, 1024 );
vis.setAnalysisInterval( l );
vis.setValidationInterval( L );
vis.setGranularity( R );
vis.setDivergenceThreshold( D_thr );
vis.setViewpoint( { {p} } );
vis.setPipeline( [](Screen& s, Object& o) { ... } );
...
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)

Conclusion

- **Large data visualization on HPC**
  - Order-independent technique for parallel rendering
  - Multi-viewpoint omni-directional rendering for in-situ visualization
- **Smart In-situ visualization**
  - Focusing on “time-to-discover” not only time-to-solution/rendering
  - Adaptive timestep selection for smart in-situ visualization
    - Spatio-temporal variations between the simulation results based on the Kullback-Leibler divergence
    - Adaptive timestep control in consideration of the variation changing patterns
    - Utilizing adaptive timestep sampling mechanism for reducing the amount of data for the subsequent processing (e.g., data storing for the post-hoc visualization and analytics, in-situ/in-transit visualization)

Future Works

- Optimal parameter estimation for the input time intervals
- Tuning and evaluation on Fugaku and other HPC systems
- Viewpoint selection without missing significant physical phenomenon

Acknowledgements

- This work was partially supported by Japan Society for the Promotion of Science (JSPS), Grant-in-Aid for Scientific Research (KAKENHI)
  - 20H04194
  - 19H03976
- Collaborators
  - Jorji Nonaka (RIKEN R-CCS)
  - Takashi Shimizu, Kengo Hayashi, Yoshiaki Yamaoka (Former Kobe U student)
  - Tsukasa Yoshinaga (Toyohashi University of Technology)
  - Kazunori Nozaki (Osaka University)