Interpolation of Scientific Image Databases

Eric Georg Kinner
Technische Universität Kaiserslautern, Germany

Jonas Lukasczyk
Arizona State University, Tempe, AZ, US

David Honegger Rogers
Los Alamos Research Laboratory, NM, US

Ross Maciejewski
Arizona State University, Tempe, AZ, US

Christoph Garth
Technische Universität Kaiserslautern, Germany

Abstract

This paper explores how recent convolutional neural network (CNN)-based techniques can be used to interpolate images inside scientific image databases. These databases are frequently used for the interactive visualization of large-scale simulations, where images correspond to samples of the parameter space (e.g., timesteps, isovalues, thresholds, etc.) and the visualization space (e.g., camera locations, clipping planes, etc.). These databases can be browsed post hoc along the sampling axis to emulate real-time interaction with large-scale datasets. However, the resulting databases are limited to their contained images, i.e., the sampling points. In this paper, we explore how efficiently and accurately CNN-based techniques can derive new images by interpolating database elements. We demonstrate on several real-world examples that the size of databases can be further reduced by dropping samples that can be interpolated post hoc with an acceptable error, which we measure qualitatively and quantitatively.

2012 ACM Subject Classification Computing methodologies → Image processing

Keywords and phrases Image Interpolation, Image Database, Cinema Database

Digital Object Identifier 10.4230/OASIcs.iPMVM.2020.19

Supplementary Material *Software*: `https://github.com/EricKinner/InterpScImgDB`
archived at `swh:1:dir:c86bc70a8f4399d4f3f8779b21e4bae2da16791b`

1 Introduction

Today, almost any domain of research and engineering is using scientific simulations. The complexity and scale increased over time due to innovations and continuous improvements in both hardware and algorithmic solutions. With this, the results also grew in size. Now, the management of data is increasingly becoming a major concern for large-scale applications. It has been examined that persistently saving or archiving data produces over-proportionally high costs (Kunkel et al. [13]), which strain institutional resources. Approaching exa-scale computing capabilities, it becomes apparent that storage constraints limit the scope of more and more simulations. Researchers are forced to decrease data output frequencies, shorten the simulation time, or shrink the region and ensemble sizes, which in return hinders the analysis and scientific workflow in general.

Scientific simulations are a valuable tool for researchers of many domains to derive and verify hypothesis, which are otherwise unobservable. Therefore, there were many techniques proposed to mitigate this problem [6][14][7][31][10][12] (further discussed in related work). One of them by Ahrens et al. [2], who proposed the creation of a image databases. These contain in situ rendered visualizations of the simulation for e.g. each timestep. Those images are then directly used to visually analyze the results afterwards. This has several benefits

2nd International Conference of the DFG International Research Training Group 2057 – Physical Modeling for Virtual Manufacturing (iPMVM 2020).
Editors: Christoph Garth, Jan C. Aurich, Barbara Linke, Ralf Müller, Bahram Ravani, Gunther Weber, and Benjamin Kirsch; Article No. 19; pp. 19:1–19:17

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

over the storage of raw simulation data. Adhinarayanan et al. [1] confirmed through a theoretical model the overall lower power consumption, hardware- and storage requirements in comparison to conventional methods.

But the creation of image databases also introduces a new trade-off between the sampling-rate and the required storage space. The denser the simulation is sampled, the more intuitive and detailed is the post hoc analysis. But excessive sampling over a large number of parameters (e.g. isovalues, thresholds, timesteps, camera angles) requires potentially the same or more disk space.

In this paper we explored how image interpolation can resolve or mitigate this trade-off. The reconstruction of intermediate frames through interpolation could be used as compression to reduce the size of these image databases. This allows in return to effectively save more samples. The creation of new frames is also possible, which allows the arbitrary change of the sampling rate. More frames usually mean smoother transitions and improved user experience. In general, image interpolation could be used for compression and data augmentation on these specific or any other image databases. It may be employed as a compression or pre/post processing step for an analysis or visualization.

However, we are not aware of any image interpolation technique that was developed for scientific visualizations. Also, to our knowledge, general purpose image interpolation has not yet been applied on scientific image datasets. Common images and scientific visualizations differ substantially in the depicted objects and the way they were rendered. Hence, we examined the feasibility and properties along minor improvements in the following. In particular we present the following contributions:

- Feasibility of image interpolation on scientific image databases
- Comparison between state-of-the-art approaches on scientific visualizations
- Publicly available docker images for reproducible results
- Case study on background substitution for algorithmic approaches
- Case study on fine-tuning CNNs for scientific visualizations

2 Related Work

The increasing demand of disk space for large-scale simulations has naturally gained the attention of many researchers. Several mitigation techniques have been developed to reduce the memory consumption of high-performance systems. Burtscher et al. [6] presented a floating point compressor, which can also be used for inter-process communication. Sequences of floating point numbers are packed and compressed together to reduce size. Lossy techniques, like the SZ compression by Di et al. [7], provide even greater data reduction. Sophisticated approaches like the "In situ Sort-And-B-spline Error-bounded Lossy Abatement" (ISABELA), developed by Lakshminarasimhan et al. [14], or the work of Tao et al. [31] reveal correlations of seemingly random data and provide very high compression for general data assemblies. Approaches that are specialized on a certain domain can often exploit similarities in the data more and therefore reach better results. For example, Jeruzalski et al. [10] introduced a novel compression for contact-dominated rigid body simulations. Since it can operate on a subset of the data, the resulting memory overhead is minimal. Kumar et al. [12] on the other hand proposed a lossy compression, that utilizes discrete cosine transform and principal component analysis on molecular dynamics simulation data.

This work is motivated by the approach of Ahrens et al. [2], who rendered the simulation data in situ and created image databases. Instead of storing raw data (e.g. position/velocity of every particle in a simulation), everything is captured through images, which are sampled along the parameter space of interest (e.g. timesteps, camera locations, isovalues, thresholds,

etc.). Adhinarayanan et al. [1] confirmed the lower storage and power requirements of in situ rendered simulation analysis in comparison to other methods. However, the denser the simulation is sampled, the more precise is the analysis. This imposes a strong trade-off between accuracy and the required storage space. Storing more images eventually becomes inefficient.

Our approach, to use general purpose image interpolation on these scientific image databases, mitigates the described predicament. The most similar techniques to our method include the work of Lukasczyk et al. [20]. They proposed a view approximation approach to reduce the spacial sampling rate. Through depth images, the scene is reconstructed. This enables the free exploration of the data with arbitrary viewpoint. But the generation of depth images is infeasible for e.g. volumetric datasets, which limits the applicability of this approach. Methods specialized on volumes, like from Fernandes et al. [8], enable post hoc camera changes through i.e. space-time-coherent volumetric depth images. But this solution is limited to volumes and can only be applied in situ. The compression of image databases through video codecs was evaluated by Berres et al. [5], which resulted in great data reduction rates. The video compression of image databases works also post hoc, but does not improve on user experience. Our image interpolation approach enables post hoc changes in frame rate, which augments the data and increases the expressiveness of data. Similar techniques are used on medical data from CT or MRT scans to improve the hardware restricted resolution of scans. A specialized technique for this was developed by Leng et al. [15], which focuses on the transformations of these scans.

The application of image interpolation on scientific datasets can compress its size and improve user experience. It is also applicable regardless of the underlying data type. It can be applied in situ as well as on existing datasets. Also, advancements in interpolation techniques can directly be incorporated due to the high modularity.

Image interpolation is an ongoing challenge in image processing and finds its application commonly in slow motion generation or the resampling of movies. The active research community continuously publishes ever improving solutions to this problem. Inspirational papers include the work of Mahajan et al. [23], which presented a algorithmic path-based technique and most recently the RRIN model of Li et al. [16].

Among numerous other techniques the following image interpolation techniques were chosen to test their applicability on scientific visualizations. Their source code is publicly accessible, which allows our results to be reproduced and extended.

- **Inter$_{\text{sepconv}}$: Video Frame Interpolation via Adaptive Separable Convolution** by Niklaus et al. (2017) [27] is a convoluational neural network for image interpolation. They use separable kernels to reduce the memory complexity to $O(2n)$. With this, larger motions can be detected. The approach is set to interpolate a single frame in the middle between two reference frames. Benchmarks show a performance of 0.9 seconds for a 1920x1080 frame on a Nvidia Titan X (Pascal). In this paper, we will refer to this approach by "Inter$_{\text{sepconv}}$" for convenience.
- **Inter$_{\text{phase}}$: Phase-based frame interpolation for video** of Meyer et al. (2015)[25] solves the interpolation of images algorithmically. The core idea is to interpret the color of images as functions and motion as the phase shift of these functions. Interpolation of the images is then solved by interpolating the phase. Without global optimization, peak performance of 1 seconds for a image of 720p can be reached on a Nvidia GeForce GTX 770. The approach allows for arbitrarily many equally spaced interpolated frames between two reference frames. In this paper, we will refer to this approach by "Inter$_{\text{phase}}$" for convenience.

- **Inter$_{\text{voxel}}$: Video Frame Synthesis using Deep Voxel Flow** is the initial work of Liu et al. (2017) [19] and involves a convolutional neural network for image interpolation and extrapolation. It is implicitly trained end-to-end on voxel flow by images directly. Although theoretically arbitrary interpolation is possible, the open source implementation only supports one intermediate frame. The authors did not make statements about the performance of the model aside from quality evaluations. In this paper, we will refer to this approach by "Inter$_{\text{voxel}}$" for convenience.
- **Inter$_{\text{cyclic}}$: Deep Video Frame Interpolation using Cyclic Frame Generation** is the later work of Liu et al. (2019) [18], which introduced a forward-backward consistency as a mean for regularization. The model is by design restricted to a single interpolation between two reference frames. The authors did not make any statements about the performance of the model. In this paper, we will refer to this approach by "Inter$_{\text{cyclic}}$" for convenience.
- **Inter$_{\text{dain}}$: Depth-Aware Video Frame Interpolation** by Boa et al. (2019) [4] is a composed image interpolation approach. It uses four existing models to estimate depth, optical flow, context, and the kernels. The approach allows for arbitrarily many equally spaced interpolated images between two reference frames. A peak performance of 0.125 seconds for a 640x480 image can be achieved on a Nvidia Titan X (Pascal). In this paper, we will refer to this approach by "Inter$_{\text{dain}}$" for convenience.
- **Inter$_{\text{nearest}}$: Nearest Neighbor Interpolation** is the most basic interpolation technique. The intermediate image is constructed by copying the temporally closest reference frame. It is used as a baseline method to compare the other approaches against it. In this paper, we will refer to this approach by "Inter$_{\text{nearest}}$" for convenience.

The publicly available open source implementations of the aforementioned techniques have been modified to provide compatibility and a uniform user interface. The encapsulation into docker images assures consistent behavior across systems without complex dependencies. They are available on: `https://github.com/EricKinner/InterpScImgDB`

3 Method

Since the domain of scientific visualizations considerably differs from real world images, it has to be evaluated if image interpolation techniques work in this context. For that, the comparison of the aforementioned approaches is most naturally done by comparing the interpolated results to ground truth images.

The similarity of two images is measured through metrics. A multitude of metrics have been developed and each of them has its own characteristics, which highlight certain properties (e.g. sharpness) and hide others. To prevent a bias caused by one of these properties, multiple metrics are used instead. They ideally cancel each others shortcomings out and provide comparability to other studies.

- **SSIM** - Structural Similarity Index [35]. A perception-based metric widely used in image processing. (range [0-1], where higher values are good; one is perfect response)
- **MS-SSIM** - Multiscale Structural Similarity Index [33]. A development of SSIM with multiple stages of subsampling. (range [0-1], where higher values are good; one is perfect response)
- **VIFP** - Visual Information Fidelity in Pixel Domain [30]. Part of the commercial FVQA (Fusion-based Video Quality Assesment) [17] by Netflix for video quality monitoring. (range [0-1], where higher values are good; one is perfect response)

- **MSE** - Mean Squared Error. Basic absolute image error metric used in many applications. (range [0-$255^2 \times width \times height$], where lower values are good; zero is perfect response)
- **PSNR** - Peak Signal to Noise Ratio. Used by image compression algorithms and derived from MSE ($10 \times log_{10}(255^2 \div MSE)$) (range (0-inf], where higher values are good)
- **UQI** - Universal Quality Index [34]. Predecessor of SSIM. (range [0-1], where higher values are good; one is perfect response)

It is to be expected that a given interpolation approach achieves different quality for different types of visualizations. For that, a variety of datasets were chosen so that common classes of scientific visualizations are represented. The testing size is of course not exhaustive, but it is large enough to evaluate the feasibility in this domain and deduce tendencies.

- **Asteroid** - Volume rendering of asteroid impact with changing turbulences. Published by Patchett et al. [28].
- **Middlebury** - Standard dataset for optical flow estimation and image interpolation by Baker et al. [3].
- **MPAS** - Model for Prediction Across Scales [26]. Static mesh with maps of atmospheric and oceanic currents.
- **Nyx** - Cosmological simulation data of contracting mass into galaxy structures, provided by Lukic [22].
- **Viscous Fingers** - Isosurface rendered example dataset of ttk library [32] [21], originally generated for the 2016 IEEE visualization contest [9].

The comparison of image interpolation techniques and any other complex system involves a multitude of variables and variations. The analysis can be performed over several dimensions. The presented results are therefore only a summary of the measurements that were conducted. The full set of the results is available in the repository (`https://github.com/EricKinner/InterpScImgDB`). For the creation of new data, generalized docker images provide comfortable interfaces and allow the validation and continuation of this work.

Feasibility. The first step towards the deployment of image interpolation techniques on scientific visualizations is the examination of its feasibility. For this, it has to be compared against suitable alternatives. Scientific image databases are usually not resampled. Leaving the frame rate untouched is equivalent to the nearest neighbor interpolation ($\text{Inter}_{\text{nearest}}$). The normalization with this baseline approach gives insight on the amount of improvement and ensures the interpolation is not worse than actually doing nothing. Note that normalized results are only comparable in the amount of improvement over the baseline approach and not over their absolute error values.

Background Substitution. The background of scientific visualizations is usually a plain color. This is a distinct difference to common images, which might negatively influence the quality of the interpolated frames. The color of the background usually does not hold any information, that is of interest to researchers. Therefore, if more favorable configurations lead to better interpolation results, it could simply be exchanged. Background substitution can be a low-cost improvement for the adaptation of image interpolation techniques on this domain of visualization.

To explore this, a set of different background images were generated synthetically. The background is substituted by the color value, which leaves translucent pixels with the original background.

- **none** - Original image background.
- **green** - Color, which is unique in the image.
- **colorGrad** - Gradient between two unique image exclusive colors.
- **black** - Uniform black background.
- **white** - Uniform white background.
- **grad** - Gradient between black and white.
- **grid** - Original image background with rectangular white grid.
- **photo** - Photograph of real world scene with house and trees.

Fine-tuning. The majority of image interpolation techniques is based on convolutional neural networks as seen in the ranking of the middlebury dataset [3]. Fine-tuning is an established method to adapt neural networks to similar, but different domains of operation. Therefore, we examined how well fine-tuning can be used to adapt neural-network based image interpolation approaches to scientific visualizations. For this, the $\text{Inter}_{\text{dain}}$ method was trained on synthetic simulation-like data. Because quality and importantly, quantity is critical for successful machine learning, a collection of public and custom datasets were used. To extend the dataset with as many samples a possible, variations (e.g. volume, isosurface, etc.) of the underlying data were generated.

- **blender scene 1** - Custom blender scene of moving light sources and primitives. The intent with this dataset is to train the model on hard edges of solid meshes.
- **blender scene 2** - Custom blender scene of primitives, which emerge from each other and change in size over time. The intended effect of this dataset on the model is to introduce changes in mass and volume to the network. This is uncommon for real images, but a common occurrence in scientific simulations of e.g. isosurfaces or turbulences.
- **mrBrain** - Sliced scan of brain tissue, provided by [24]. Static frame with changing texture is meant to increase stability for semi-static scenes.
- **bonsai** - CT-Scan of a bonsai tree, provided by[11]. Rendered as volume with moving camera and changing isosurface.
- **lobster** - Resin embedded CT-Scan of a lobster, provided by [11]. Variations include moving volume rendering and changing isosurface.
- **csafe heptane** - Simulation step of combusting jet of heptane gas [11]. Training samples include volume renderings, sliced layers, and changing isosurfaces.

4 Results

Feasibility. The quality of a given image interpolation approach on scientific visualizations varies heavily. A positive example is displayed in Figure [1], which displays the $\text{Inter}_{\text{nearest}}$ normalized MSE error of $\text{Inter}_{\text{cylcic}}$ for all datasets. The scaling of the graph is below one. This indicates for the given normalized error metric, that the produced results are consistently better than the baseline approach. The least error is accumulated for the asteroid and nyx datasets. They even outperform the middlebury datset, which is the de-facto standard dataset for the comparison of image interpolation techniques. This confirms the general applicability of image interpolation on scientific visualizations. On some of the tested datasets an improvement of up to ten times could be achieved.

Yet for other approaches results like the ones displayed in Figure [2] can be observed. The normalized PSNR metric displays the lack of quality of $\text{Inter}_{\text{dain}}$ on all datasets. For this metric, values below one indicate worse quality in in comparison to the baseline ($\text{Inter}_{\text{nearest}}$). It can be seen that the interpolation of some datasets yield results worse than $\text{Inter}_{\text{nearest}}$. This technique is, therefore, not suited for the application on scientific visualizations.

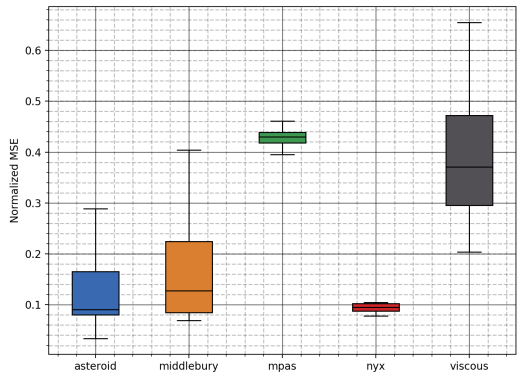

Figure 1 Evaluation of MSE error on single frame skipped interpolation by $\text{Inter}_{\text{cyclic}}$. Lower values are good (zero is perfect response).

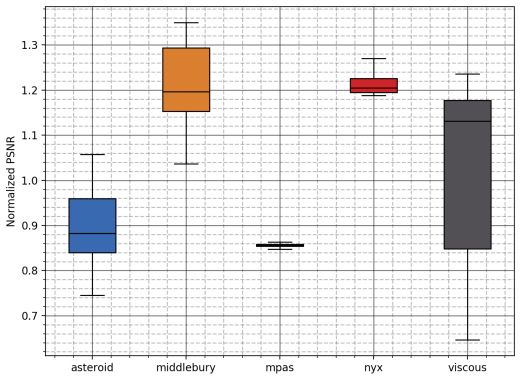

Figure 2 Evaluation of PSNR quality metric on single frame skipped interpolation by $\text{Inter}_{\text{dain}}$. Higher values are good.

In conclusion, it can be seen that the feasibility of general purpose image interpolation on scientific visualizations is heavily dependent on the dataset and method. Results range from much worse to up to ten times better than the baseline alternative ($\text{Inter}_{\text{nearest}}$). In the following we present suggestions for the selection of interpolation techniques. These aim to reduce the time to find an optimal configuration of data and approach.

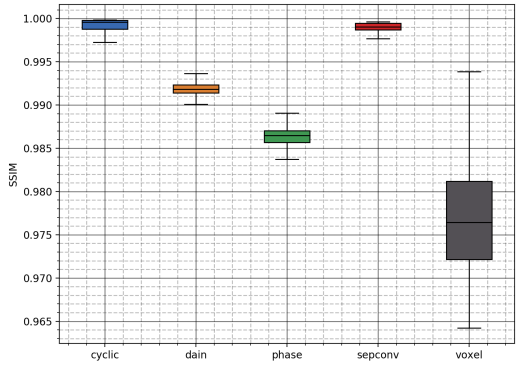

Figure 3 Evaluation of SSIM quality metric on the asteroid dataset with single frame skipped. Higher values are good. (one is perfect response).

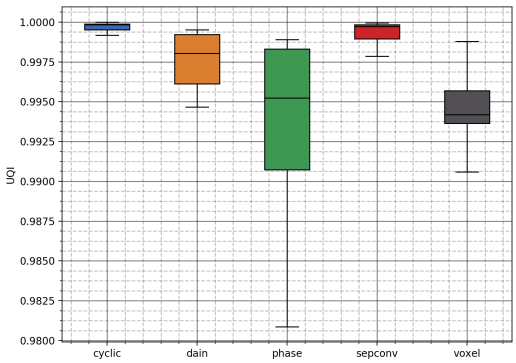

Figure 4 Evaluation of UQI quality metric on the asteroid dataset with single frame skipped. Higher values are good. (one is perfect response).

Limitations. The application of general purpose image interpolation techniques on scientific visualizations is in general feasible, but their results range a lot in quality. A closer look on their limitations and artifacts lead to a deeper understanding on favorable configurations.

For that, Figure [3] displays the quality of the interpolation approaches on the asteroid dataset in form of the SSIM metric. The asteroid dataset displays a volume rendered asteroid impact with subsequent atmospheric turbulences. A low quality and high variance of $\text{Inter}_{\text{voxel}}$ reveals its poor performance on volume-rendered images. Strict pixel- and voxel-flow based approaches enforce consistencies, which are not fulfilled in scientific simulations. There, colors are calculated through density functions of e.g. turbulences rather than being the product of previous motions.

The same configuration of dataset, approach, and frames skipped is displayed in Figure [4]. Here the the UQI quality metric is used. The $\text{Inter}_{\text{phase}}$ approach stands out with a high variance. A side-by-side comparison of the reconstructed frame with the original

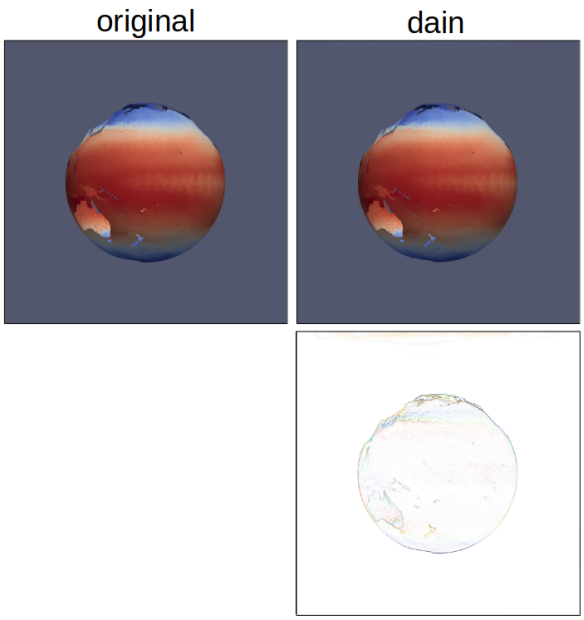

Figure 5 Difference from interpolated frame to original of the MPAS dataset with $\text{Inter}_{\text{dain}}$. Note: Enhanced contrasts for visibility.

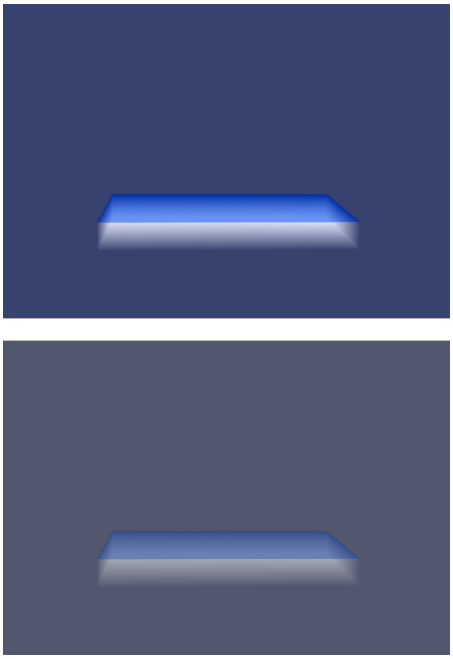

Figure 6 Interpolated frame (top) of the asteroid dataset with $\text{Inter}_{\text{phase}}$. Original frame (bottom) shows color difference.

one is shown in Figure [6]. It reveals that the colors of the interpolated frame are more saturated in comparison to the original. After the simulation progressed and a bigger portion of the screen is occupied by objects, the color difference becomes less distinct. Compression artifacts cannot be the reason for this anomaly, since lossless PNG compression [29] was used throughout the study. This suggests that the uniform colored background, which is most prominent in the first frames of the simulation, is the cause for this artifact.

A static mesh of the earth with changing textures of atmospheric and oceanic currents is displayed in the MPAS dataset. The $\text{Inter}_{\text{phase}}$ approach generates frames with slightly different color for this dataset as well. A different artifact is produced by $\text{Inter}_{\text{dain}}$. Because the general motion of the texture is directed towards a single direction, motion is falsely detected and corrected for the worse.. This causes the mesh to slightly change its position as a consequence. Figure [5] shows an example frame of the difference to ground truth. The fake motion causes a high error at the edges. Complex compound systems like $\text{Inter}_{\text{dain}}$ are capable of perceiving scenes very well. This generally increases the interpolation quality but may also lead to artifacts specific to scientific visualizations as the consistency of the scene is not necessarily preserved.

The Nyx dataset displays changing cosmological formations through sliced textures. The $\text{Inter}_{\text{voxel}}$ approach shows for this dataset a particular low quality. Figure [7] displays a interpolated frame of this dataset. It can be seen that the edge of the simulation region is fragmented and merged with the background. The temporal perception of the dataset is most probably related to this phenomenon. It displays the formation of cosmological structures. Through the accumulation of mass into stellar structures the texture seems to contract. The perceived contraction then causes $\text{Inter}_{\text{voxel}}$ to 'suck' pixels from neighboring regions (i.e. the background) into the simulation region. This did not occur for the motion of the MPAS

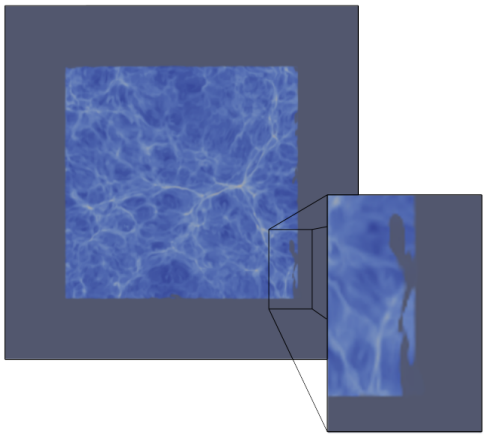

Figure 7 Interpolated frame of the nyx dataset, generated by $\text{Inter}_{\text{voxel}}$ approach. Areas on the edges are fragmented and merged with background.

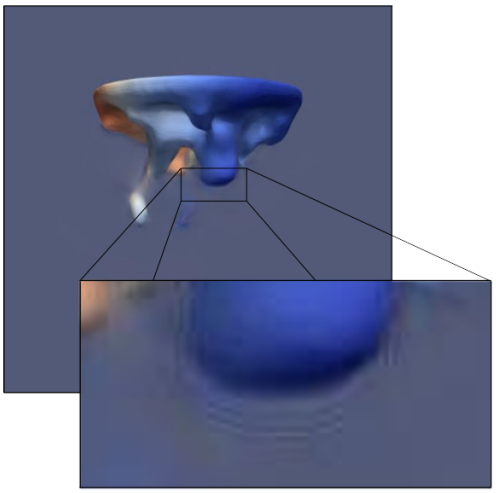

Figure 8 Interpolated frame of the viscous finger dataset, generated by $\text{Inter}_{\text{phase}}$ approach. Expanding fingers produce high frequency noise around the edges.

dataset. The reason might be the scale of the motion, which can only be perceived scene-wide and cannot be detected locally. This shows very clearly that strict flow-based approaches like $\text{Inter}_{\text{voxel}}$ are unable to deal with differences between optical flow and relative motion, which is not uncommon in scientific visualizations.

The most difficult dataset for all approaches is the viscous finger dataset. The isosurface rendering of mixing fluids (oil and water) shows expanding, contracting, and vanishing fingers, which appeared to be a challenge for most approaches. Each approach, however, reacted with different symptoms. The intermediate frames of $\text{Inter}_{\text{phase}}$ showed high frequency artifacts caused by the big ambiguity between consecutive frames like seen in Figure [8]. Other approaches, like $\text{Inter}_{\text{sepconv}}$ and $\text{Inter}_{\text{cyclic}}$, produced ghosting artifacts similar to linear blending. The $\text{Inter}_{\text{voxel}}$ interpolation shows the same artifacts as the ones created on the nyx dataset. The artifacts are clearly visible in Figure [9]. It shows well how the approach predicts paths from one image to another. It tries to fulfill the requirement of consistent voxel transitions. The fluctuations of mass is unlike any real world scenario though. This prevents matches between the frames, which are interpreted by most strict flow-based approaches as occlusion. And occluded areas are filled with a prediction of the background. The best reconstruction is from $\text{Inter}_{\text{dain}}$, which is capable of interpolating fingers with the least amount of blur. But it, too, is unstable on uniform backgrounds and produces fog-like artifacts in the beginning of the dataset as displayed in Figure [10]. Since this is the only dataset, where $\text{Inter}_{\text{dain}}$ showed these artifacts, it may be possible to adapt for it by fine-tuning, as seen later in the respective section.

Because every interpolation approach is different, it is not possible to make absolute statements about their applicability on scientific visualizations. But through domain knowledge and the conducted measurements, tendencies become apparent. With these recommendations simulation teams can exclude certain approaches and may save some time in the selection of suitable interpolation approaches.

Volume rendered images can be interpolated best by CNN based approaches, which are trained on optical flow, like $\text{Inter}_{\text{sepconv}}$ and $\text{Inter}_{\text{cyclic}}$. These methods are comparatively basic, but produce solid results for the nyx and mpas datasets as well. Algorithmic solutions,

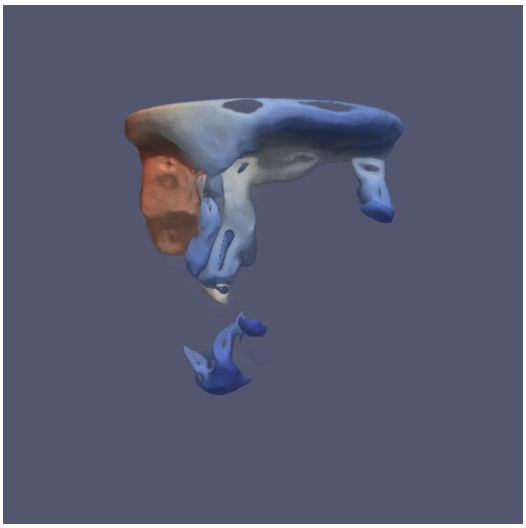

Figure 9 Frame interpolated by $\text{Inter}_{\text{voxel}}$ from viscous finger dataset.

Figure 10 Interpolated frame of the viscous finger dataset, generated by $\text{Inter}_{\text{dain}}$.

like $\text{Inter}_{\text{phase}}$, should be avoided, since they often rely on constraints and properties common in images but rare in scientific visualizations. Shown through multiple examples, it is also apparent that strict voxel or pixel flow based approaches produce heavy artifacts for changing masses and turbulences. Complex multi-component architectures can deal with these situations better since scene information is processed as well. But these systems are less predictable due to the high complexity and require careful fine-tuning to avoid a wrong perception of the scene.

The accuracy with respect to the desired features should be considered as well. If scientists want to derive other information from the interpolated images, stability with respect to e.g. classification, edge detection etc. has to be maintained. High-frequency noise like in Figure [8] or ghosting artifacts may invalidate the detection of edges, while it may be acceptable for a segmentation. This is also true for annotated data. Spatially coherent pixel-based information (like depth) can, in theory, also be interpolated and benefit from this approach. However, most available interpolation techniques are not designed for floating point images.

Iterative Application. The level of compression and maximum frame rate is directly linked to the number of intermediate frames. While some approaches allow for arbitrary interpolation between two reference frames, it is most common to interpolate a single frame in the middle of two reference frames. These approaches ($\text{Inter}_{\text{cyclic}}$, $\text{Inter}_{\text{voxel}}$, $\text{Inter}_{\text{sepconv}}$) have to be applied multiple times consecutively to achieve the same benefits. The number of intermediate frames through iterative application grows in $2^n - 1$. Approaches with variable number of intermediate frames ($\text{Inter}_{\text{dain}}$, $\text{Inter}_{\text{phase}}$) are more flexible in the selection of error thresholds for adaptive compression. In comparison, no definite qualitative difference between variable and iterative interpolation could be observed.

The maximum number of reconstructed frames is dependent on the initial sampling rate of the dataset. For very sparsely sampled data, image interpolation might not be a viable compression technique. But these datasets tend to benefit even more from the augmentation of their data and increased frame rates. The datasets, which were examined in this paper,

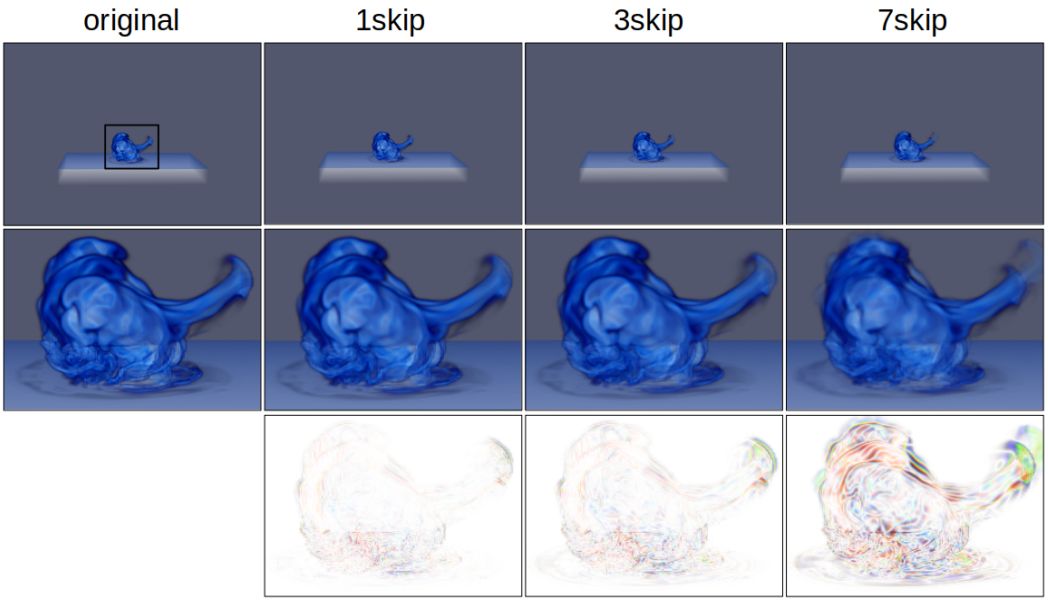

Figure 11 Comparison of example frame of asteroid dataset on $\text{Inter}_{\text{cyclic}}$ with iterative application and increasing interval of skipped frames.

had an average initial sampling rate. For them, we found empirically that 1-3 skipped frames result in acceptable quality. Only for densely sampled datasets 7 frames could be skipped (87% compression). Larger temporal steps between frames then tend to introduce aliasing artifacts. Figure [11] displays the results of iterative application of $\text{Inter}_{\text{cyclic}}$ on example frames of the asteroid dataset.

Table 1 Average performance of the interpolation approaches on a 512x512 frame inside the docker container. T_{init} is the time to load the model and all associated resources. T_{interp} is the time to interpolate a single frame. The measurements were conducted on Ubuntu 20.04.1 LTS, 16 GB RAM, i7-7700HQ @ 2.8GHz * 8, GeForce GTX 1050 Ti Mobile (4GB), Docker version 19.03.13.

approach	$Inter_{cylcic}$	$\text{Inter}_{\text{dain}}$	$\text{Inter}_{\text{phase}}$	$\text{Inter}_{\text{sepconv}}$	$\text{Inter}_{\text{voxel}}$
T_{init} [sec]	6.305	3.328	1.473	3.142	5.912
T_{interp} [sec]	2.248	23.841	5.352	0.220	3.520

Performance. The speed at which frames can be interpolated determines how the interpolation is used. Fast interpolations may be employed for online applications, while slower approaches would only be used for archiving data. Some of the approaches ($\text{Inter}_{\text{dain}}$, $\text{Inter}_{\text{cyclic}}$) use large models, which need to be loaded on initialization. Not only do they require sufficient GPU memory but also require significant time to load.

Table [1] displays the average performance on an 512x512 frame. T_{init} is the time needed for loading the model and other resources. T_{interp} is the time needed to interpolate a single frame. Note that the measurements were conducted inside a docker container. The virtualization overhead among other factors might influence the actual runtime. It is also important to note that the execution (T_{interp}) of $\text{Inter}_{\text{dain}}$ is most likely much lower. During the fine-tuning of $\text{Inter}_{\text{dain}}$ a processing speed of ~1.22 seconds were observed. The training was conducted on an Nvidia GeForce GTX 2080 (8GB), which provided enough memory to

load all of the sub-models simultaneously.

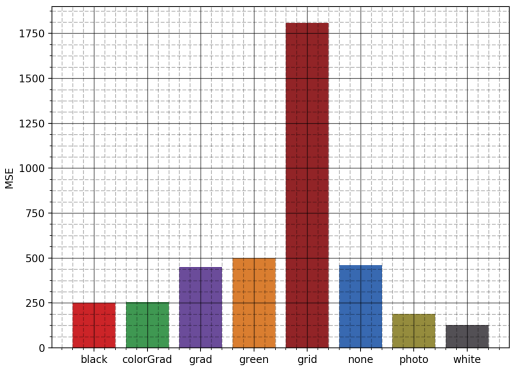

Figure 12 Evaluation of MSE error metric on a single frame of the asteroid dataset. The dataset was substituted with different backgrounds and interpolated by $\text{Inter}_{\text{phase}}$. Lower values are good. (zero is perfect response).

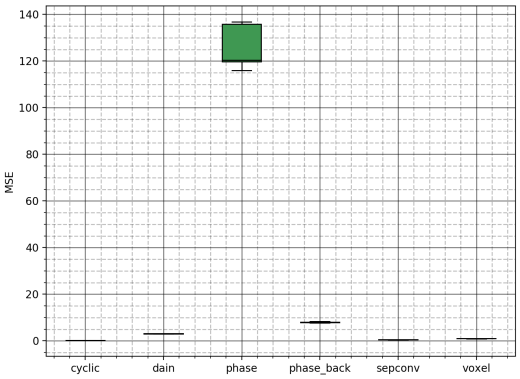

Figure 13 Evaluation of MSE error metric on the mpas dataset with a single frame skipped. Comparison of interpolation approaches including background substitution for $\text{Inter}_{\text{phase}}$. Lower values are good. (zero is perfect response).

Background Substitution. The background of scientific simulations can have a negative effect on the quality of the interpolation result. (see Figure [10] and [6]). The investigation and substitution can be of special benefit for algorithmic solutions ($\text{Inter}_{\text{phase}}$), since they generally lack other means of optimization. In the following, the effect of background substitution is examined on $\text{Inter}_{\text{phase}}$, which produced intermediate frames with slightly different color and contrast to the original dataset.

An example frame of the asteroid dataset is tested to select an optimal background for the $\text{Inter}_{\text{phase}}$ approach. Figure [12] displays the MSE error metric of the different substitutions. Contrary to expectations are the worst results produced by the grid background, which provides extra reference points. The uniform white background produced the best results for this approach. It also outperforms the substitution by the real world photo. This was not expected, since the methods were developed for real images. But this shows that well chosen synthetic images can actually be used to increase the quality of image processing methods. And since generally no extra information is encoded in background pixels, a substitution or initial selection of a favorable background can be a low-cost improvement for the later analysis framework.

The effect on the total quality of the reconstruction can be seen in Figure [13], which displays the results of all interpolation approaches aside the white substituted background. The former high error due to a change in colors is completely corrected and competitive quality can be achieved. A side-by-side comparison of frames from the MPAS dataset with the background substitution can be seen in Figure [17]. There, the improvement is not restricted to the correction of artifacts, but also reduces the error of the object. Background substitution can be applied to any approach regardless of their type of architecture. It reduces artifacts an improves the interpolation of the objects in the scene. It is a powerful tools for maximizing the quality of the reconstruction and is easily integrated into post hoc analysis frameworks.

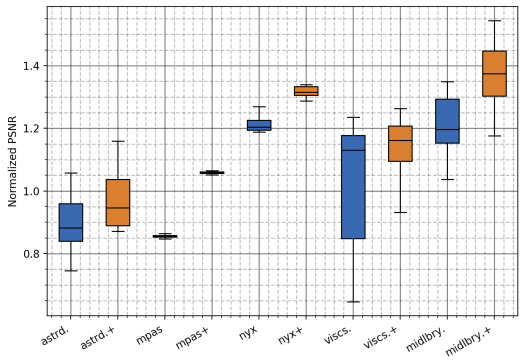

Figure 14 Comparison of fine-tuned $\text{Inter}_{\text{dain}}$ (indicated by "+")(orange) against the unmodified version (blue) on all datasets with PSNR quality metric. Higher values are good.

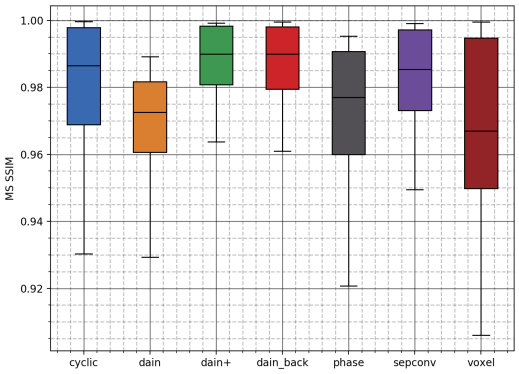

Figure 15 Evaluation of MS SSIM quality metric on the viscous dataset with 1 frame skipped and comparison to fine-tuned $\text{Inter}_{\text{dain}}$. Higher values are good. (one is perfect response).

Fine-tuning. The adaptation of machine learning approaches to new domains is commonly done via fine-tuning. In the following, $\text{Inter}_{\text{dain}}$ is fine-tuned. The new results give information about the magnitude of improvement, that can be expected from fine-tuning approaches or whether the architecture of the network is more important. The training was conducted on the provided model with default settings (apart from: num-epochs=32, lr=0.0005, batch-size=1) and on a Nvidia RTX 2080 with a total runtime of $\sim$ 6 hours.

The performance of the fine-tuned model (indicated by "+") in comparison to the unmodified approach is displayed in Figure [14] with the PSNR quality metric. The graph is normalized by $\text{Inter}_{\text{nearest}}$ to be able to display all datasets side by side. Higher values indicate better quality. The graphs shows that the quality increased for the fine-tuned version in comparison to the original model on all tested datasets. The extent of the improvement is also, as seen by the scale of the graph, of significant magnitude. The median increased, while variances decreased. Even the results for the middlebury dataset could be improved. Note that the metrics used in this evaluation are different from the metric used to determine the ranking of the middlebury dataset. But this shows that synthetic data can positively impact the training of conventional interpolation techniques. Artificially constructed edge cases like changing masses and deformations can deepen the understanding of the scene for the model and improve results. A frame-based side-by-side comparison of the fine-tuned version is shown in Figure [16].

The resulting performance of the fine-tuned model in comparison to all the other interpolation approaches, as well as a background substituted version, is shown in Figure [15]. The MS SSIM quality metric shows a significant increase in quality of the fine-tuned model. This changes its ranking among the other methods. $\text{Inter}_{\text{dain+}}$ is now the best choice. The rise in quality can mostly be attributed to the lack of artifacts, like seen in Figure [10], which is also achieved by suitable background substitution. Yet also edges and some fingers of the dataset are interpolated better, which could otherwise not been achieved. A side-by-side comparison is displayed in Figure [18], which illustrates the new robustness against moving textures, which previously caused errors like in Figure [5].

Although fine-tuning is able to overcome shortcomings on most datasets, it does not universally become the best option for every approach. The previous statements about the general performance of architectures on different datasets still hold. The correct choice of

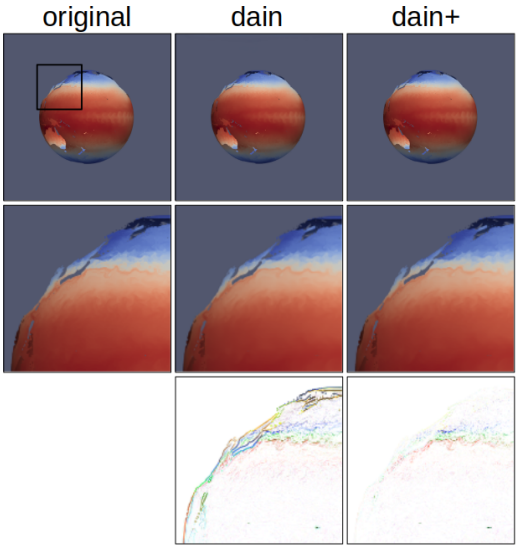

Figure 16 Comparison of example frame of asteroid dataset on $\text{Inter}_{\text{dain}}$ with fine-tuned version.

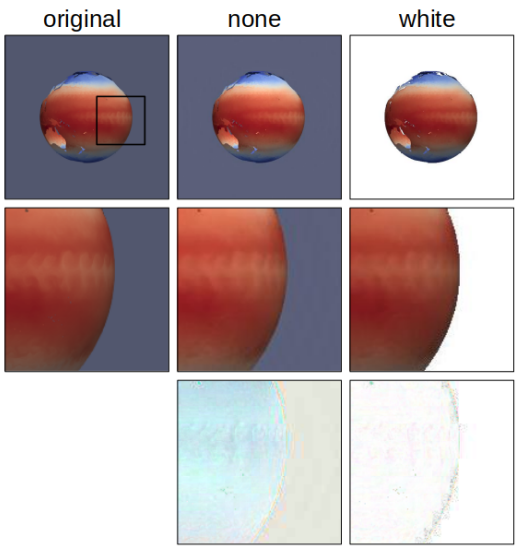

Figure 17 Comparison of example frame of mpas dataset on $\text{Inter}_{\text{phase}}$ with white background substitution.

architecture, visualization setting (i.e. background), and data type has a greater effect on the resulting quality. The application of image interpolation on scientific datasets without fine-tuning would a waste of potential. It is worthwhile and strongly recommended to adapt the interpolation approach in some way to the target dataset.

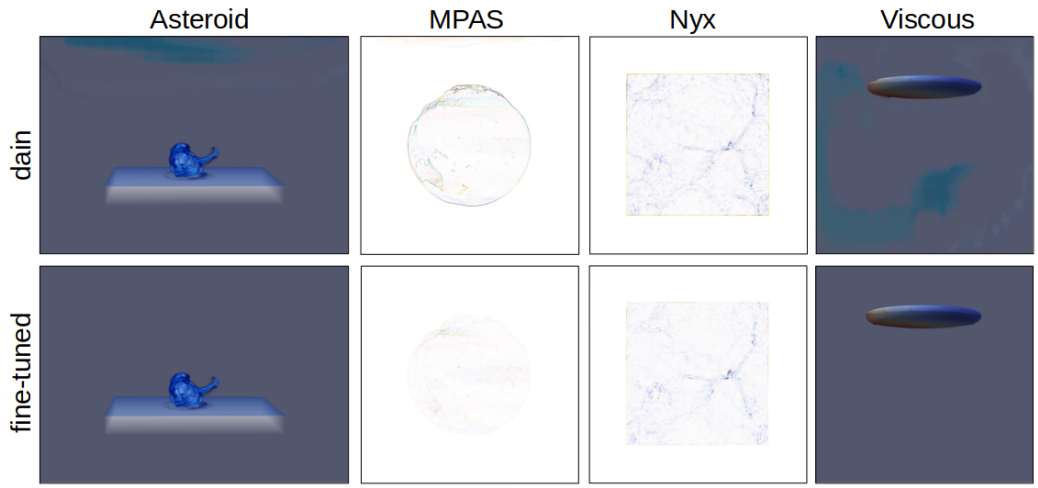

Figure 18 Example images of single frame skipped interpolation of all datasets between original and fine-tuned $\text{Inter}_{\text{dain}}$. MPAS and nyx dataset are displayed as enhanced difference images to increase the visibility of differences.

5 Conclusion

In this paper, we explored the feasibility of general purpose image interpolation techniques on scientific image databases. A survey over state-of-the-art methods with extensive evaluation over multiple metrics and datasets demonstrated the successful application on scientific visualizations. Tendencies toward favorable configurations could be observed, which give concrete recommendations for the selection of suitable interpolation approaches. A measurement of the runtime classifies the approaches for online application or post hoc data reduction. Invariably, all utilized approaches are publicly available through unified docker images. This enables this work to be verified or extended without setting up external dependencies. The limitations of e.g. iterative application have been examined and clarify how far and when the application of interpolation techniques is useful. Case studies for low-cost improvements, such as background substitution and fine-tuning, have shown to adjust systems for this domain of visualization.

In summary, image interpolation is a viable solution for improved user experience and storage compression of scientific image databases. Further research may include the development of expert systems for the interpolation of scientific interpolations as well as a composition of tools like view interpolation [20] and video compression [5] into a unified framework. The visualization of uncertainty for reconstructed frames would improve the acceptability and open its usage to more critical domains. Lastly, it may also be of interest to investigate the applicability of general purpose image interpolation onto volumetric datasets, which often are already sliced (e.g. MRT, CT) and could potentially be enhanced or compressed.

References

1 V. Adhinarayanan, W. c. Feng, D. Rogers, J. Ahrens, and S. Pakin. Characterizing and Modeling Power and Energy for Extreme-Scale In-Situ Visualization. In *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 978–987, May 2017. doi:10.1109/IPDPS.2017.113.

2 James Ahrens, Sébastien Jourdain, Patrick O'Leary, John Patchett, David H. Rogers, and Mark Petersen. An Image-based Approach to Extreme Scale in Situ Visualization and Analysis. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '14, pages 424–434, Piscataway, NJ, USA, 2014. IEEE Press. doi:10.1109/SC.2014.40.

3 Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, March 2011. https://vision.middlebury.edu/flow/eval/. doi:10.1007/s11263-010-0390-2.

4 Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-Aware Video Frame Interpolation. *CoRR*, abs/1904.00830, 2019. Implementation from https://github.com/baowenbo/DAIN. arXiv:1904.00830.

5 Anne S. Berres, Terece L. Turton, Mark Petersen, David H. Rogers, and James P. Ahrens. Video Compression for Ocean Simulation Image Databases. In Karsten Rink, Ariane Middel, Dirk Zeckzer, and Roxana Bujack, editors, *Workshop on Visualisation in Environmental Sciences (EnvirVis)*. The Eurographics Association, 2017. doi:10.2312/envirvis.20171104.

6 Martin Burtscher and Paruj Ratanaworabhan. FPC: A High-Speed Compressor for Double-Precision Floating-Point Data. *Computers, IEEE Transactions on*, 58:18–31, February 2009. doi:10.1109/TC.2008.131.

7 S. Di and F. Cappello. Fast Error-Bounded Lossy HPC Data Compression with SZ. In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 730–739, 2016.

8 O. Fernandes, S. Frey, F. Sadlo, and T. Ertl. Space-time volumetric depth images for in-situ visualization. In *2014 IEEE 4th Symposium on Large Data Analysis and Visualization (LDAV)*, pages 59–65, 2014.

9 IEEEVIS. Scientific Visualization Contest. http://www.uni-kl.de/sciviscontest/, 2016.

10 Timothy Jeruzalski, John Kanji, Alec Jacobson, and David I.W. Levin. Collision-Aware and Online Compression of Rigid Body Simulations via Integrated Error Minimization. *Computer Graphics Forum (Proc. SCA)*, 2018.

11 Pavol Klacansky. Open SciVis Datasets, December 2017. https://klacansky .com/open-scivis-datasets/. URL: `https://klacansky.com/open-scivis-datasets/`.

12 Anand Kumar, Xingquan Zhu, Yi-Cheng Tu, and Sagar Pandit. Compression in molecular simulation datasets. In Changyin Sun, Fang Fang, Zhi-Hua Zhou, Wankou Yang, and Zhi-Yong Liu, editors, *Intelligence Science and Big Data Engineering*, pages 22–29, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

13 Julian Kunkel, Michael Kuhn, and Thomas Ludwig. Exascale Storage Systems – An Analytical Study of Expenses. *Supercomputing Frontiers and Innovations*, 1(1), 2014. URL: `https://superfri.org/superfri/article/view/20`.

14 Sriram Lakshminarasimhan, Neil Shah, Stephane Ethier, Seung-Hoe Ku, C. S. Chang, Scott Klasky, Rob Latham, Rob Ross, and Nagiza F. Samatova. ISABELA for effective in situ compression of scientific data. *Concurrency and Computation: Practice and Experience*, 25(4):524–540, 2013. `doi:10.1002/cpe.2887`.

15 Juelin Leng, Guoliang Xu, and Yongjie Zhang. Medical image interpolation based on multi-resolution registration. *Computers + Mathematics with Applications*, 66(1):1–18, 2013. `doi:10.1016/j.camwa.2013.04.026`.

16 H. Li, Y. Yuan, and Q. Wang. Video Frame Interpolation Via Residue Refinement. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2613–2617, 2020.

17 J. Y. Lin, T. Liu, E. C. Wu, and C. . J. Kuo. A fusion-based video quality assessment (fvqa) index. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, pages 1–5, 2014.

18 Yu-Lun Liu, Yi-Tung Liao, Yen-Yu Lin, and Yung-Yu Chuang. Deep Video Frame Interpolation using Cyclic Frame Generation. In *Proceedings of the 33rd Conference on Artificial Intelligence (AAAI)*, 2019. Implementation from https://github.com/alex04072000/CyclicGen.

19 Ziwei Liu, Raymond Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video Frame Synthesis using Deep Voxel Flow. In *Proceedings of International Conference on Computer Vision (ICCV)*, October 2017. Implementation from https://github.com/liuziwei7/voxel-flow and https://github.com/lxx1991/pytorch-voxel-flow.

20 J. Lukasczyk, E. Kinner, J. Ahrens, H. Leitte, and C. Garth. VOIDGA: A View-Approximation Oriented Image Database Generation Approach. In *2018 IEEE 8th Symposium on Large Data Analysis and Visualization (LDAV)*, pages 12–22, 2018.

21 Jonas Lukasczyk, Gunther Weber, Ross Maciejewski, Christoph Garth, and Heike Leitte. Nested Tracking Graphs. *Comput. Graph. Forum*, 36(3):12–22, 2017. `doi:10.1111/cgf.13164`.

22 Zarija Lukic. Nyx Cosmological Simulation Data. http://dx.doi.org /10.21227/k8gb-vq78, 2019. `doi:10.21227/k8gb-vq78`.

23 Dhruv Mahajan, Fu-Chung Huang, Wojciech Matusik, Ravi Ramamoorthi, and Peter Belhumeur. Moving Gradients: A Path-Based Method for Plausible Image Interpolation. *ACM Trans. Graph.*, 28(3), July 2009. `doi:10.1145/1531326.1531348`.

24 Bill Lorensen Marc Levoy, Graham Gash. University of North Carolina Volume Rendering Test Data Set, 1987. URL: `https://graphics.stanford.edu/data/voldata/`.

25 S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. Sorkine-Hornung. Phase-based frame interpolation for video. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1410–1418, June 2015. Implementation from https://github.com/justanhduc/Phase-based-Frame-Interpolation. `doi:10.1109/CVPR.2015.7298747`.

26 MPAS. Model for Prediction Across Scales. https://mpas-dev.github.io/, 2019.

27 Simon Niklaus, Long Mai, and Feng Liu. Video Frame Interpolation via Adaptive Separable Convolution. *CoRR*, abs/1708.01692, 2017. Implementation from https://github.com/sniklaus/sepconv-slomo. `arXiv:1708.01692`.

28 John Patchett and Galen Gisler. Deep Water Impact Ensemble Data Set. Technical report, LANL, 2017. LA-UR-17-21595. URL: `https://datascience.dsscale.org/wp-content/uploads/2017/03/DeepWaterImpactEnsembleDataSet.pdf`.

29 PNG. Portable Network Graphics Specification. https://tools.ietf.org/html /rfc2083, March 1997.

30 H. R. Sheikh and A. C. Bovik. Image information and visual quality. *IEEE Transactions on Image Processing*, 15(2):430–444, 2006.

31 D. Tao, S. Di, Z. Chen, and F. Cappello. Significantly Improving Lossy Compression for Scientific Data Sets Based on Multidimensional Prediction and Error-Controlled Quantization. In *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 1129–1139, 2017.

32 Julien Tierny, Guillaume Favelier, Joshua A. Levine, Charles Gueunet, and Michael Michaux. The Topology ToolKit. *CoRR*, abs/1805.09110, 2018. `arXiv:1805.09110`.

33 Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, volume 2, pages 1398–1402 Vol.2, 2003.

34 Zhou Wang and A. C. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, 2002.

35 Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.