Real-time Acquisition and Reconstruction of Dynamic Volumes with Neural Structured Illumination

Yixin Zeng Zoubin Bi Mingrui Yin Xiang Feng Kun Zhou Hongzhi Wu*
State Key Lab of CAD&CG, Zhejiang University

Abstract

We propose a novel framework for real-time acquisition and reconstruction of temporally-varying 3D phenomena with high quality. The core of our framework is a deep neural network, with an encoder that directly maps to the structured illumination during acquisition, a decoder that predicts a 1D density distribution from single-pixel measurements under the optimized lighting, and an aggregation module that combines the predicted densities for each camera into a single volume. It enables the automatic and joint optimization of physical acquisition and computational reconstruction, and is flexible to adapt to different hardware configurations. The effectiveness of our framework is demonstrated on a lightweight setup with an off-the-shelf projector and one or multiple cameras, achieving a performance of 40 *volumes per second at a spatial resolution of* 128^3*. We compare favorably with state-of-the-art techniques in real and synthetic experiments, and evaluate the impact of various factors over our pipeline.*

1. Introduction

High-quality volumetric reconstruction of dynamic phenomena is an essential task in scientific research, with a wide variety of important applications, including aircraft design [8], vehicle manufacturing [7], weather forecasting [6], and even modern microscopy [29]. Expressed as a temporally-varying 3D density volume, the captured results help scientists better understand/validate different physical properties (e.g., aerodynamics) of the complex underlying phenomena.

However, it is difficult to acquire and reconstruct dynamic volumes from the physical world with high fidelity. First, the samples from common 2D imaging sensor(s) are usually not the direct measurements of a 3D volume. Furthermore, the dynamic nature limits the sampling budget for each volume. The fundamental challenge here is the information gap between the samples and the final temporally-changing 3D volumes.

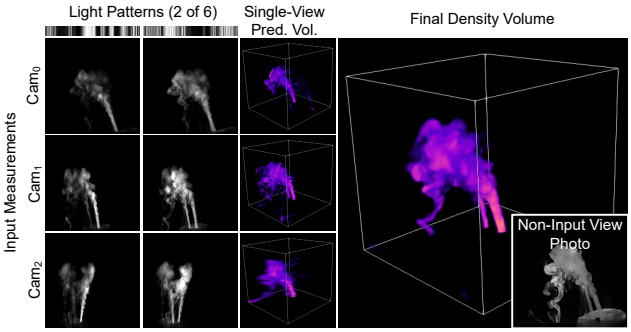

Figure 1. Using as few as 6 pre-optimized structured light patterns, we capture and reconstruct high-quality, dynamic 3D volumes from corresponding image measurements at different views, with a lightweight projector-camera setup. We achieve a performance of 40 volumes per second for both acquisition and reconstruction. Pred. = predicted, and vol. = volume.

Considerable research efforts have been made over the past decades. Approaches with uncontrolled illumination fill in the information gap with prior knowledge, based on heuristics [18, 20], physical rules [13, 14], or learning from existing data [16, 33]. The reconstruction quality and computation time are usually not satisfactory, especially with a sparse sampling, as the information missing from the measurements is effectively hallucinated. On the other hand, controlled-lighting-based methods pack more physical information into the measurements to reduce the gap [19, 21]. But existing work often suffers from long acquisition time or low Signal-to-Noise Ratio (SNR). The *sampling efficiency* is still insufficient for high-quality acquisition of dynamic volumes.

To tackle the above challenge, we propose a novel framework for high-quality, real-time acquisition and reconstruction of each dynamic 3D volume independently, for a projector-camera setup. We map both the projected structured light patterns and computational reconstruction to a neural network, which enables the automatic and joint optimization of hardware and software, towards optimal sam-

*Y. Zeng and Z. Bi contributed equally. K. Zhou and H. Wu are the corresponding authors. E-mail: {kunzhou,hwu}@acm.org

pling efficiency. Moreover, we propose a generalizable and reusable 1D decoder, which only takes as input *single-pixel* measurements along with the related local incident lighting during acquisition, and outputs the 1D density distribution over the corresponding camera ray. After applying the decoder to each valid camera ray at each view, we aggregate the results to produce the final 3D density volume. Simulated fluid data are used to train the entire pipeline.

The effectiveness of our framework is demonstrated on a lightweight setup with one off-the-shelf projector and one or multiple synchronized cameras. Using as few as 6 optimized, structured light patterns, we capture and reconstruct 3D volumes with a spatial resolution of 128^3 at a speed of 40 volumes per second, on a number of dynamic physical scenes. The framework is validated with synthetic data as well as volumes captured with light slices [21]. Finally, we compare with state-of-the-art techniques both qualitatively and quantitatively, and evaluate the impact of different factors over our pipeline.

2. Related Work

Below we categorize related work based on whether the lighting is controlled or not during acquisition. Note that our focus is on *markerless* capture of *dynamic* volumes. For a broader view of the topic, please refer to the excellent survey of [25].

2.1. Uncontrolled Illumination

Existing work in this category does not require specialized light sources, and therefore is less strict on the acquisition conditions. The downside is the difficulty to distinguish 3D structures from the projection onto a single 2D image sensor. One class of methods employ a large number of views to alleviate this issue, resulting in expensive hardware, such as a dense set of cameras [28] or multiple lightfield cameras [23]. On the other hand, another class of approaches perform a more practical sparse view sampling, and rely on *priors* to fill in the information gap.

Heuristic Priors. Traditionally, researchers hand-derive priors to effectively constrain the solution space when only limited input is available. Various priors are proposed, including subspaces spanned by different basis functions [2, 4, 5, 24], spatial compactness [20], appearance consistency across views [30] and reprojection consistency [40].

Physics Priors. The knowledge of physics laws can be exploited, to essentially propagate the information about the phenomenon across the temporal domain by, e.g., computing velocity/force fields. Image-based reconstruction can be performed in conjunction with physical simulation [13, 14, 37]. Recently, Chu et al. [10] leverage Navier-Stokes equations in an end-to-end optimization of a neural representation to reconstruct dynamic fluid phenomena.

Learned Priors. Priors learned from a large amount of data are generally more robust for reconstruction than handcrafted ones. In [15], a learned 2D discriminator constrains observations from unseen angles. Qiu et al. [33] utilize a differentiable advection layer and a velocity estimation network to facilitate an end-to-end optimization. An adversarial loss is trained to restrict the density volume to a plausible appearance in [16].

The above reconstruction methods essentially *"hallucinate"* the information missing from the sparse input, which may lead to inaccurate results if the priors are not applicable. Moreover, their computational overhead is too expensive to support real-time reconstruction. In comparison, we aim to increase the information useful for reconstruction in each sample via a joint optimization, at the cost of a more controlled setup.

2.2. Controlled Lighting

This category of work programs the illumination to more actively probe the physical domain and less rely on priors, resulting in a higher reconstruction quality. Related work can be further divided, depending on whether one or multiple lights are used at the same time.

Light Scanning. While conceptually simple, it is challenging for scanning-based approaches to achieve a sufficiently high performance for dynamic capture. One may either reach a high scanning speed by employing expensive hardware [21], or sacrifice the completeness of the result to reduce the scanning burden (e.g., limited angle [3, 22] or sparse view [9]). In the latter case, priors are also needed to fill in the information gap, similar to Sec. 2.1. Note that the scanning idea is widely adopted in different fields of scientific imaging, including laser induced fluorescence [12], light-sheet microscopy [31], and particle image velocity [32].

Illumination Multiplexing. This class of methods considerably improve the acquisition efficiency by programming multiple sources simultaneously. A density volume is reconstructed/interpolated from a single-view image with multiple laser lines at the same time [17]. Zhang et al. [41] adopt a Hadamard multiplexing scheme to decrease the exposure time. The total number of images, however, is not reduced. Furthermore, compressive sensing is applied to volumetric acquisition in [19]: only 24 input images under random patterns are needed to compute a 3D volume, at the cost of an involved optimization procedure. Recently, Kang et al. [26] build a visible-light tomography prototype using 1,920 interleaved sources and detectors with a complete 360° coverage. Their reconstruction network is highly coupled with the device. It is not clear how to extend to other common setups, typically with more samples in the spatial domain and less in the angular domain.

Our approach is closest to this class. Compared with the

majority of existing work, we perform a joint optimization of physical acquisition (i.e., neural structured lighting) and computational reconstruction for enhanced sampling efficiency. In comparison with [26], we obtain substantially superior results with a simpler, more device-independent network, as our decoder only predicts a 1D density distribution directly from measurements under different 1D incident lighting.

3. Preliminaries

Throughout this paper, we denote the number of light patterns as #p, and the number of cameras/views as #v. The cameras are named as cam_0, cam_1, ... We represent a density volume with a spatial resolution of 128×128×128, denoted as ρ. No color information is considered in our pipeline.

Assumptions. Similar to existing work [17, 19, 26], we assume (1) no attenuation, reflection, refraction, or multiple scattering in light propagation; (2) a constant phase function for all light received by one camera; (3) that the physical phenomenon remains static in the duration for capturing #p images/projecting #p light patterns. Note that the validity of this imaging model is demonstrated in the above work on reconstructing *optically thin* phenomena (e.g., smoke/vapor). We do not exploit any temporal coherence: each volume is reconstructed *independently*.

Imaging model. Under the above assumptions, an image measurement $\mathbf{I}$ can be modeled as the integral of the product between the local incident lighting L_{local} and the 1D density distribution along a particular camera ray:

$$\mathbf{I} = \int L_{\text{local}}(r)\rho(\mathbf{p}(r))dr. \tag{1}$$

Here r is the distance from the camera center to a 3D point $\mathbf{p}(r)$ along the camera ray. $L_{\text{local}}(r)$ is computed as follows: connect $\mathbf{p}(r)$ to the projector center; find the intersection with the projector plane; and fetch the intensity at the intersection as the result. In addition, $\rho(\mathbf{p}(r))$ can be viewed as a 1D density distribution over a camera ray (which varies with r). A graphical illustration is shown in Fig. 3-b & c.

Illumination Multiplexing. Similar to related work [1, 19, 42], our light patterns consist of vertical strips (see Fig. 12 for examples). We represent each pattern $\mathbf{L}$ as a 1D vector, which contains 128 intensities that correspond to different strips. Due to the linearity of light transport, the image measurement $\mathbf{I}$ under a pattern $\mathbf{L}$ can be modeled as a linear combination of single-strip-lit measurements:

$$\mathbf{I} = \Sigma_{i=0}^{127}\mathbf{L}_i\mathbf{I}_i, \tag{2}$$

where $\mathbf{L}_i$ is the intensity of the i-th strip, and $\mathbf{I}_i$ is the measurement with only the i-th strip on and set to an intensity of 100%, similar to [21].

4. Acquisition Setup

Our lightweight setup consists of a single consumer-grade projector and one or multiple vision cameras, all pointing towards a **valid volume** of 96mm×96mm×96mm, as illustrated in Fig. 2. The projector, BenQ X3000, has a spatial resolution of 1920×1080 and a projection speed of 240 fps. The cameras, Basler acA1440-220umQGR, capture *gray-scale* images of 1440×1080, and are synchronized with the projector via time-varying tags (Sec. 10.3). In the setup, cam_0 is perpendicular to the projection direction, while cam_1/cam_2 are $\pm 30^\circ$ apart, as determined by our experiment in Fig. 10/Sec. 7.2.

A projection window of 512×640 roughly covers the valid volume, with each strip of 4×640 corresponding to an intensity in a light pattern. The strip size is determined after balancing reconstruction resolution and SNR during acquisition.

We carefully calibrate parameters of the system, including the intrinsic/extrinsic/gamma curve of each camera/projector, via classic methods followed by a joint, differentiable calibration of the entire system. Please refer to the supplementary material for details.

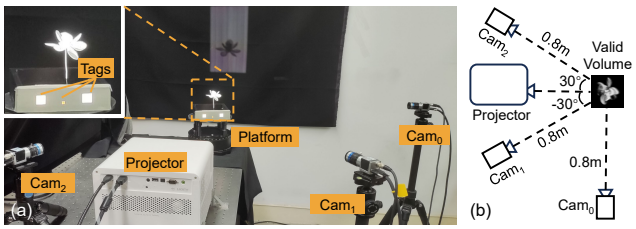

Figure 2. Our acquisition setup. A photograph of the setup is shown in (a), with 1 projector and 3 cameras. Multiple light patterns are projected to the physical scene on the platform for acquisition. The time-varying tags are used for synchronization. A top-view layout of the setup, as well as the valid volume, is in (b).

5. Overview

To scan a dynamic phenomenon, we loop over the *same, fixed* set of pre-optimized light patterns (whose total number is #p) with the projector, and take corresponding photographs with one or multiple cameras. Next, for each camera view, a group of #p consecutive images are processed to produce a 3D density volume. All volumes from different views are then aggregated to obtain the final result, when multiple cameras are employed. We repeat this process to reconstruct a sequence of dynamic volumes. Please refer to Fig. 3 for a graphical illustration.

6. Our Network

For training, the network input is a synthetic 3D density volume (Sec. 6.6). First, this volume is encoded by the light

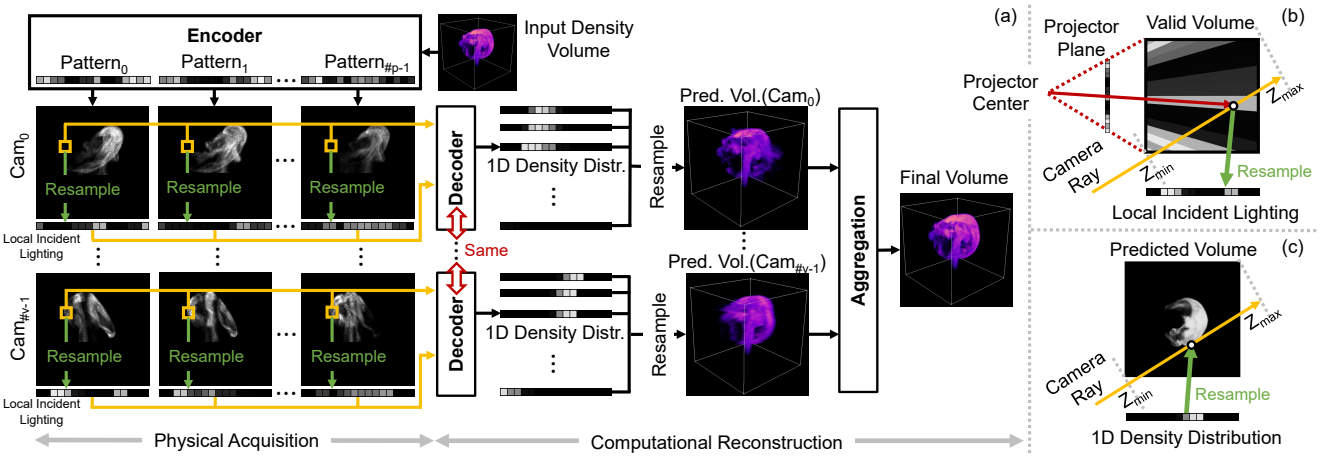

Figure 3. Our pipeline (a), and the resampling process from a light pattern to 1D local incident lighting (b), and from a predicted 1D density distribution to a density volume (c). Starting from a synthetic/physical 3D density volume, we first project the pre-optimized light patterns (i.e., weights in the encoder) to the volume. For each valid pixel at each camera view, we send all its measurements along with the resampled local illumination conditions to a decoder, to predict a 1D density distribution over the corresponding camera ray. All density distributions for one camera are then collected and resampled into a single 3D volume. In the multi-camera case, the predicted volumes for each camera are fused to obtain the final result. Pattern = light pattern, distr. = distribution, pred. = predicted, and vol. = volume.

patterns as image measurements for each camera (Sec. 6.2), to simulate the physical measurement process. Next, for each valid pixel at each camera view, we send all its #p measurements along with the related local illumination conditions (Fig. 3-b) to a decoder (Sec. 6.3), to predict a 1D density distribution over the corresponding camera ray. All density distributions for one camera are then collected and resampled (Sec. 6.1 & Fig. 3-c) into a single 3D volume. In the multi-camera case, the predicted volumes for each camera are fused to obtain the final result (Sec. 6.4). Please refer to Fig. 3 for a graphical illustration of the network, as well as Fig. 4 for detailed architectures.

At runtime, we project #p pre-trained light patterns (i.e., the weights in the encoder) for acquisition, and feed the corresponding physical measurements for each camera as input to the decoder. The results of the decoder will then be resampled and aggregated to predict the final 3D density volume.

6.1. Resampling

Before describing individual componets of the network, we first introduce our resampling process, which is used in the encoder and after the decoder. Specifically, we uniformly sample #s points along the current camera ray, whose depth is in the range of $[z_{\min}, z_{\max}]$. $z_{\min}$ and $z_{\max}$ are computed from the intersections of all camera rays of the current view with the valid volume, as illustrated in Fig. 3-b & c. In our experiments, #s is set to $\lceil 128\sqrt{3} \rceil$ to prevent undersampling.

Note that the coefficients involved in the resampling from light patterns to local incident lighting (i.e., for linearly combining the intensities in a light pattern to compute an intensity of the local lighting), as well as from predicted 1D density distributions to a density volume, only depend on geometric relationships between the projector, the camera(s) and the valid volume. Therefore, we precompute all the coefficients for runtime efficiency.

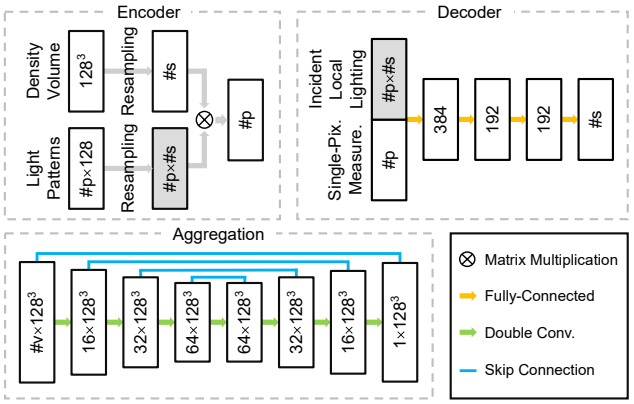

Figure 4. Architectures of 3 components of our network: the encoder, the decoder and the aggregation module. Measure. = measurements, and pix. = pixel.

6.2. Encoder

The encoder simulates the measurement process, linking the light patterns, the input density volume and the output image measurements in a differentiable manner. For each pixel location at each camera view, local incident lighting are resampled from the light patterns (Fig. 3-b); we also trace into

the density volume to resample a 1D density distribution along the current camera ray (Fig. 3-c); finally, an image measurement is computed based on the above two factors, according to Eq. (1). Please refer to Fig. 4 for the architecture.

The encoder models the #p light patterns as optimizable weights. For physical plausibility, each such weight goes through a sigmoid function to ensure that as a light intensity, it is within the range of [0, 1].

Note that our light patterns are independent of the physical sample, as they are trained to be efficient *in expectation*. Moreover, at a single pixel location, the fundamental ambiguity is that there could be multiple 1D density distributions corresponding to the measurements. Our jointly-trained lighting helps physically transform the 1D density distribution to as unambiguous measurements as possible.

6.3. Decoder

Our decoder consists of 4 fc layers and works on a per-pixel basis. It takes as input the measurements at the same pixel location and the corresponding local incident lighting (resampled from #p light patterns according to Sec. 6.2), and outputs a 1D density distribution along the corresponding camera ray. Once the decoding for all pixels of a camera is done, we collect and resample all results into a single density volume.

Compared with a straightforward network that takes as input the light patterns, pixel location and camera parameters, our design is more elegant and efficient. The decoder focuses on the key reconstruction task only: how to predict the densities along a camera ray, from pixel measurements with different local incident light? Other complex tasks, such as geometric transformations of input/output information, are delegated to manually coded resampling procedures.

6.4. Aggregation

In the multi-camera case, we take as input the 3D volumes predicted by the decoder for each camera, and fuse the multi-view information to output a high-quality volume with a 3D UNet [11]. It consists of 8 convolution layers as well as 4 skip connections (Fig. 4). While a 3D UNet works well in all experiments, our pipeline is not married to this way of aggregation; other architectures (e.g., MLP) can also be plugged in (see Fig. 8). Note that if only one camera is used in the setup, this aggregation step may be skipped.

6.5. Loss Functions

In the single-camera case, the loss function is the simple per-voxel Root-Mean-Square Error (RMSE):

$$\mathcal{L}_{\text{single}} = \sqrt{\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_2}{n}}, \tag{3}$$

where $\mathbf{x}$ is the density volume predicted by the decoder, $\tilde{\mathbf{x}}$ is the ground-truth, and $n = 128^3$ is the total number of voxels of a volume.

In the multi-camera case, the loss function is defined as:

$$\mathcal{L}_{\text{multi}} = \sqrt{\frac{\|\mathbf{x}_{\text{aggre}} - \tilde{\mathbf{x}}\|_2}{n}} + \lambda \sum_{i=0}^{\#v-1} \sqrt{\frac{\|\mathbf{x}_i - \tilde{\mathbf{x}}\|_2}{n}}. \tag{4}$$

Here $\mathbf{x}_{\text{aggre}}$ is the final volume after aggregation, $\mathbf{x}_i$ is the volume predicted by the decoder for the i-th camera, and λ is set to 0.5 in all experiments. While the ultimate goal is to minimize the first term, we find that adding the second term helps accelerate the convergence in training. Also, we intentionally avoid additional regularization terms for generalization of our network.

6.6. Training

Similar to existing work [26, 38], our training data are density volumes from randomly generated sequences of fluid motions with Mantaflow [36]. Fig. 5 shows some examples. The initial values of our light patterns are drawn i.i.d. from a normal distribution ($\mu = 0, \sigma = 1$).

To increase the robustness in physical experiments, we simulate measurement noise with an absolute Poisson noise ($\lambda = 0.02$) and a relative Gaussian noise ($\mu = 0, \sigma = 6\%$) during training, following the model in [27]. The noise model as well as its parameters are determined via extensive experimental comparisons between simulated and physical measurements under different light patterns.

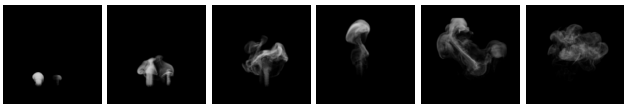

Figure 5. Examples of synthetic training data, each of which is visualized from the view of cam_0.

7. Results

All computation experiments are conducted on a server with dual AMD EPYC 7763 CPUs, 768GB DDR4 memory and 8 NVIDIA GeForce RTX 4090 GPUs. Our network is implemented in PyTorch, and trained using the Adam optimizer, with a learning rate of 5×10^{-3} and a batch size of 5. The total training time is 48 hours. All volumetric results are rendered with Pytorch3D [34] or visualized with Tomviz [35]. In either case, a density value lower than a threshold of 1×10^{-3} is considered noise and set to 0.

Two settings are used in this paper: a single camera with 12 light patterns, and 3 cameras with 6 patterns. Unless noted, the latter is the default setting for our experiments. It corresponds to an acquisition speed of 40 volumes per

second with our projector (Sec. 4). And it takes 9.2ms to reconstruct a volume from captured images under 6 light patterns, with our unoptimized code. Therefore, we achieve real-time performance for *both* acquisition and reconstruction.

Fig. 6 shows images from 4 sequences of the sublimation of dry ice, captured and reconstructed with our approach. We add dry ice to a bottle with liquid water and control its sublimation into the air with a valve, which is further directed to the valid volume as a source using a rubber tube, similar to [14]. The number of sources used in all 4 sequences are 1, 1, 3, and 2, respectively. For qualitative validations, we also capture the scene using a smartphone from a view direction close to the diagonal of the valid volume, and render our results approximately to this view. Readers are encouraged to watch the supplementary video for animated results of all sequences.

7.1. Comparisons

We first compare our method with other competitors, including a single-view approach [19] and multi-view ones [10, 15], on reconstructing a real static object in Fig. 7. For fairness, the same pattern number #p=12 is used in multi-pattern-based methods. All approaches are compared with a baseline method that essentially performs "optical sectioning" of the object, by projecting one of 128 light slices at a time, similar to [21]. None of the techniques can produce a result whose quality is higher than ours. Note that even though our network is trained on fluid simulations, it generalizes well to the flower-like shape in this experiment, due to the design of our single-pixel decoder.

Next, we compare against existing work on a synthetic sequence of smoke in Fig. 9. Various reconstruction errors are reported. Please also refer to the supplementary video for a comparison of the animated sequences. Our results are superior to existing techniques [10, 15] qualitatively and quantitatively. Moreover, our millisecond-level reconstruction time per volume is several orders magnitudes of less than [10] or [15], which requires about 10 to 40 minutes to perform involved computations for reconstruction a 3D volume.

7.2. Evaluations

We evaluate the impact of various factors over our pipeline. Fig. 10 visualizes the impact of view angles of cam_1 and cam_2. We keep the view direction of cam_0 perpendicular to the projection direction, as this minimizes the ambiguity along the depth dimension. For cam_1 and cam_2, a number of different combinations of view angles are tested, by training and computing the reconstruction error at a volumetric resolution of 32^3. It is interesting to note that the lowest error is achieved with all view angles equally split up π, which justifies our multi-cam configuration in Sec. 4.

Fig. 11 plots the validation losses of the networks with different numbers of light patterns/cameras (also cf. Fig. 7 to compare the reconstruction quality with different numbers of cameras on a real object). The loss decreases with the increase of the number of light patterns or cameras, demonstrating the flexibility/scalability of our framework to exploit different sampling capabilities.

In Fig. 12, we evaluate the impact of different sets of light patterns over reconstruction quality with 1 camera. We train variants of our network in conjunction with random binary patterns or a subset of Hadamard ones. These alternative patterns are fixed during training. With the same $\#p$, our result is more accurate. This is because our light patterns (i.e., physical sampling) are jointly optimized with the reconstruction network towards optimal quality, which better harnesses the capability of the setup.

Finally, Fig. 8 tests different design choices of our network on a synthetic sample. We compare with 4 alternatives: (1) one-hot encoding of the camera index and the screen-space coordinates as input to the decoder (to replace the 1D local incident lighting); (2) directly aggregating camera-space volumes (instead of volumes defined in a common global space); (3) aggregation with a 4-layer MLP (instead of a 3D UNet); and (4) an end-to-end network with 9 convolution layers, which directly predict a final volume from input measurements. For (1), such a common encoding of input information is more straightforward, as the camera index and the screen-space coordinates are sufficient to determine a camera ray, from which the local incident lighting can be derived. However, this encoding is not as simple and elegant as our input, as now the network would have the additional burden to learn to effectively resample the light patterns. For (2), that network would need to implicitly align all volumes anyway, making it less efficient. For (3), it shows that our pipeline can switch to different ways of aggregations with ease. For (4), an end-to-end network does not fully exploit the structure of the reconstruction problem, leading to inferior accuracy.

8. Limitations and Future Work

Similar to any work based on sequential patterns, the physical phenomenon of interest is assumed to remain static for the duration of multi-pattern projection. To lift this restriction, one may employ an advanced light source (e.g., the lightweight lightfield projector in [39]), or consider additional physical dimensions (e.g., wavelength/polarization), to encode sufficient information in a single image for high-quality volumetric reconstruction. In addition, it will be interesting to extend the imaging model to account for more complex optical phenomena, such as reflection, refraction and multiple scattering. Moreover, combining our approach with neural representations seems to be a promising research direction. Finally, we hope that our lightweight sys-

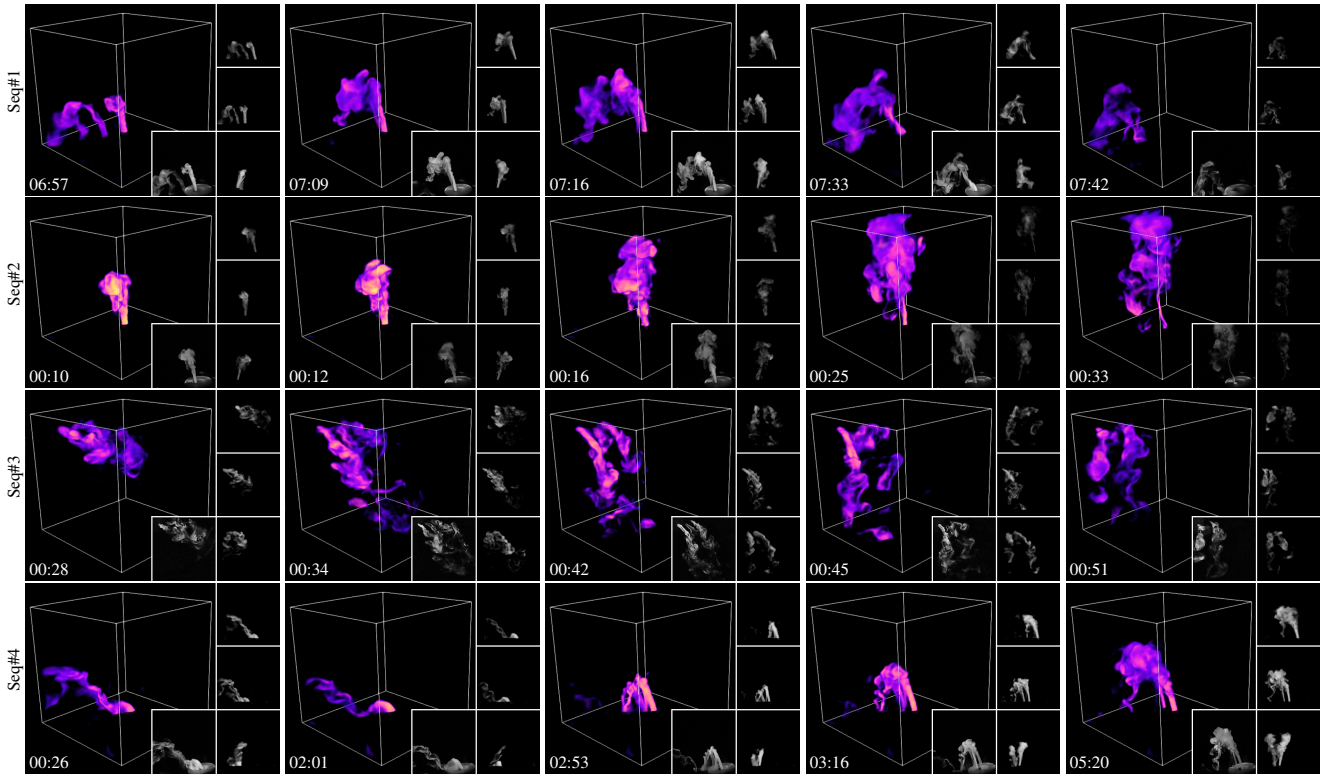

Figure 6. Reconstructions of different dynamic scenes. We visualize a subset of the reconstructed results from the sequences of 4 dynamic scenes. For each image, in addition to the visualization of the reconstructed volume, we show the photograph of the scene at a non-input view as the inset next to the bottom-right one. The column of 3 rightmost insets are the rendering results of the volume at 3 input camera views. A time stamp is displayed on the bottom-left corner. Please also refer to the supplementary video for animated results.

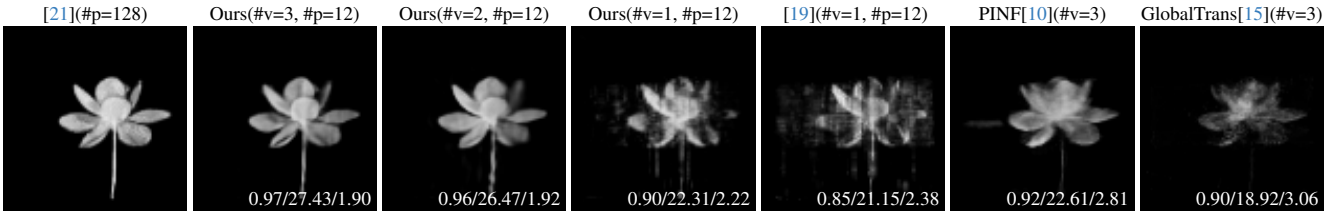

Figure 7. Comparisons of different techniques on a real static object. The leftmost image is the reconstruction result from a baseline method that essentially performs optical sectioning [21]. For all other results, quantitative errors with respect to the baseline result are reported in SSIM/PSNR/RMSE($\times 10^{-2}$) at the bottom-right corner of corresponding images. All volumes are rendered with a non-input view.

tem can enable novel applications in the future, where both real-time acquisition and reconstruction are desired.

Acknowledgements. We would like to thank Xiaohe Ma, Kaizhang Kang, Xianmin Xu and Chong Zeng for their generous help. This work is partially supported by NSF China (62332015 & 62227806), the Fundamental Research Funds for the Central Universities (226-2023-00145), the XPLORER PRIZE and Information Technology Center and State Key Lab of CAD&CG, Zhejiang University.

References

[1] Andres Aguirre-Pablo, Abdulrahman Aljedaani, Jinhui Xiong, Ramzi Idoughi, W. Heidrich, and Sigurdur Thoroddsen. Single-camera 3d ptv using particle intensities and structured light. *Experiments in Fluids*, 60, 2019. 3

[2] L. Ahrenberg, I. Ihrke, and M. Magnor. Volumetric reconstruction, compression and rendering of natural phenomena from multi-video data. In *Fourth International Workshop on Volume Graphics, 2005.*, pages 83–230, 2005. 2

[3] Rushil Anirudh, Hyojin Kim, Jayaraman J. Thiagarajan, K. Aditya Mohan, Kyle Champley, and Timo Bremer. Lose

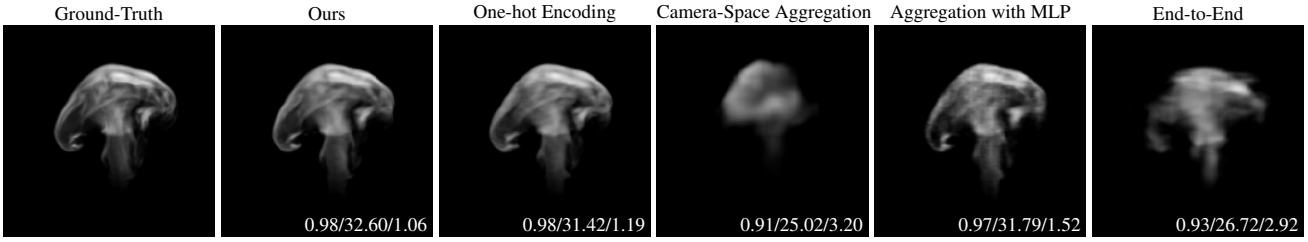

Figure 8. Impact of various factors over reconstruction quality on a synthetic smoke example. We use #p=6 and #v = 3 in all cases. From the left to right: the ground-truth, our network, our network with one-hot encoding of the camera index and screen-space coordinates, our network that aggregates on camera-space volumes, our network with an MLP-based aggregation module, an end-to-end convolution network. Quantitative errors in SSIM/PSNR/RMSE($\times 10^{-2}$) are reported at the bottom-right of corresponding images. Please refer to Sec. 7.2 for details.

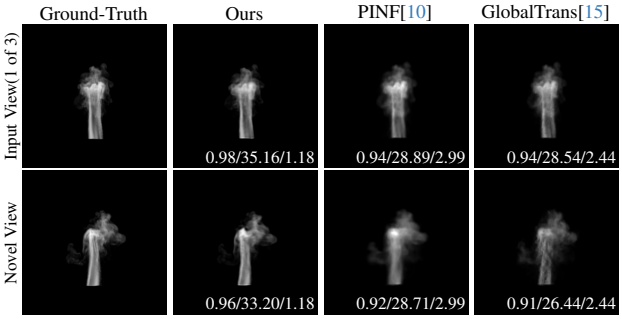

Figure 9. Comparisons of different methods on a synthetic smoke sequence. The rendering results of an input view and a novel view are shown in the 1st and 2nd row, respectively. Quantitative errors in SSIM/PSNR/RMSE($\times 10^{-2}$) are reported at the bottom-right corner of corresponding images. Please also refer to the supplementary video for animated results. Note that the reconstruction errors for each volume are displayed in the video, and the average errors in supplementary material.

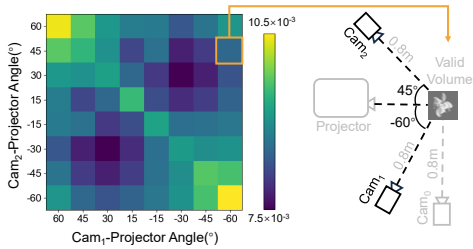

Figure 10. Impact of the angles between cam_1/cam_2 and the projector over the reconstruction loss. cam_0 is fixed to be perpendicular to the projector. An example with cam_1-projector angle=-60° and cam_2-projector angle=45° is illustrated on the right.

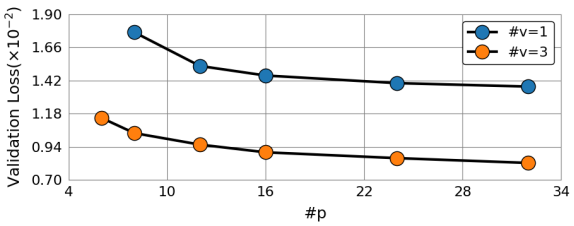

Figure 11. Impact of the light pattern number (#p) and the camera number (#v) over the reconstruction quality. The loss decreases with the increase in either #p or #v, as more information can be sampled.

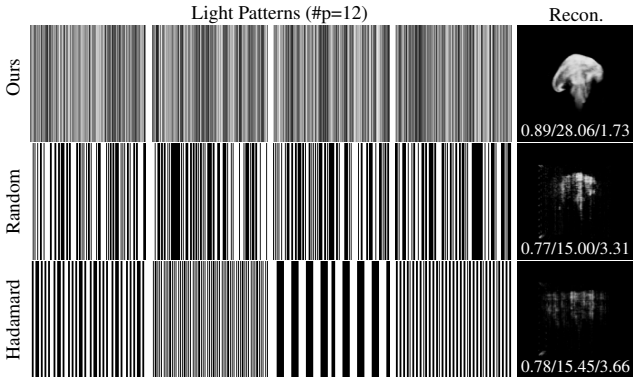

Figure 12. Impact of different light patterns on reconstructing a synthetic smoke example. For each row, the left 4 images are a subset of all 12 light patterns, and the rightmost image is the reconstructed volume. From the top row to bottom, our patterns, random binary patterns, and 1D Hadamard ones. Quantitative errors in SSIM/PSNR/RMSE($\times 10^{-2}$) are reported at the bottom-right corner of corresponding images.

the views: Limited angle ct reconstruction via implicit sinogram completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[4] Bradley Atcheson, Ivo Ihrke, Wolfgang Heidrich, Art Tevs, Derek Bradley, Marcus Magnor, and Hans-Peter Seidel. Time-resolved 3d capture of non-stationary gas flows. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 27(5): 132, 2008. 2

[5] Bradley Atcheson, Wolfgang Heidrich, and Ivo Ihrke. An evaluation of optical flow algorithms for background oriented schlieren imaging. *Experiments in Fluids*, 2009. 2

[6] Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):

47–55, 2015. 1

[7] D. M. Bushnell and K. J. Moore. Drag reduction in nature. *Annual Review of Fluid Mechanics*, 23(1):65–79, 1991. 1

[8] Louis Cattafesta, Chris Bahr, and Jose Mathew. Fundamentals of wind-tunnel design. *Encyclopedia of aerospace engineering*, pages 1–10, 2010. 1

[9] Guang-Hong Chen, Jie Tang, and Shuai Leng. Prior image constrained compressed sensing (piccs): a method to accurately reconstruct dynamic ct images from highly undersampled projection data sets. *Medical physics*, 35:660–3, 2008. 2

[10] Mengyu Chu, Lingjie Liu, Quan Zheng, Erik Franz, Hans-Peter Seidel, Christian Theobalt, and Rhaleb Zayer. Physics informed neural fields for smoke reconstruction with sparse data. *ACM Transactions on Graphics*, 41(4):1–14, 2022. 2, 6, 7, 8, 1

[11] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference*, pages 424–432. Springer, 2016. 5

[12] S Deusch and T Dracos. Time resolved 3d passive scalar concentration-field imaging by laser induced fluorescence (lif) in moving liquids. *Measurement Science and Technology*, 12:188, 2001. 2

[13] M.-L Eckert, W. Heidrich, and N. Thuerey. Coupled fluid density and motion from single views. *Computer Graphics Forum*, 37:47–58, 2018. 1, 2

[14] Marie-Lena Eckert, Kiwon Um, and Nils Thuerey. ScalarFlow: a large-scale volumetric data set of real-world scalar transport flows for computer animation and machine learning. *ACM Transactions on Graphics*, 38(6):1–16, 2019. 1, 2, 6

[15] Erik Franz, Barbara Solenthaler, and Nils Thuerey. Global transport for fluid reconstruction with learned self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1632–1642, 2021. 2, 6, 7, 8, 1

[16] Erik Franz, Barbara Solenthaler, and Nils Thuerey. Learning to estimate single-view volumetric flow motions without 3d supervision. In *International Conference on Learning Representations*, 2023. 1, 2

[17] Christian Fuchs, Tongbo Chen, Michael Goesele, Holger Theisel, and Hans-Peter Seidel. Density estimation for dynamic volumes. *Computers & Graphics*, 31(2):205–211, 2007. 2, 3

[18] James Gregson, Michael Krimerman, Matthias Hullin, and Wolfgang Heidrich. Stochastic tomography and its applications in 3d imaging of mixing fluids. *ACM Transactions on Graphics - TOG*, 31:1–10, 2012. 1

[19] Jinwei Gu, Shree K. Nayar, Eitan Grinspun, Peter N. Belhumeur, and Ravi Ramamoorthi. Compressive structured light for recovering inhomogeneous participating media. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):1–1, 2013. 1, 2, 3, 6, 7

[20] Samuel W. Hasinoff and Kiriakos N. Kutulakos. Photo-consistent reconstruction of semitransparent scenes by density-sheet decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):870–885, 2007. 1, 2

[21] Tim Hawkins, Per Einarsson, and Paul Debevec. Acquisition of time-varying participating media. *ACM Transactions on Graphics (ToG)*, 24(3):812–815, 2005. 1, 2, 3, 6, 7

[22] Yixing Huang, Xiaolin Huang, Oliver Taubmann, Yan Xia, Viktor Haase, Joachim Hornegger, Guenter Lauritsch, and Andreas Maier. Restoration of missing data in limited angle tomography based on helgason-ludwig consistency conditions. *Biomedical Physics Engineering Express*, 3, 2017. 2

[23] Yuta Ideguchi, Yuki Uranishi, Shunsuke Yoshimoto, Yoshihiro Kuroda, Masataka Imura, and Osamu Oshiro. Reconstruction of smoke based on light field consistency. *IEEJ Transactions on Sensors and Micromachines*, 136(12):522–531, 2016. 2

[24] Ivo Ihrke and Marcus Magnor. Image-based tomographic reconstruction of flames. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 365–373, 2004. 2

[25] Ivo Ihrke, Kiriakos N Kutulakos, Hendrik PA Lensch, Marcus Magnor, and Wolfgang Heidrich. Transparent and specular object reconstruction. In *Computer Graphics Forum*, pages 2400–2426. Wiley Online Library, 2010. 2

[26] Kang Kaizhang, Bi Zoubin, Feng Xiang, Dong Yican, Zhou Kun, and Wu. Hongzhi. Differentiable dynamic visible-light tomography. *SIGGRAPH Asia*, 2023. 2, 3, 5

[27] Mikhail Konnik and James Welsh. High-level numerical simulations of noise in ccd and cmos photosensors: review and tutorial. 2014. 5

[28] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics*, 38(4):1–14, 2019. 2

[29] Douglas Murphy and Michael Davidson. *Fundamentals of Light Microscopy and Electronic Imaging*. 2012. 1

[30] Makoto Okabe, Yoshinori Dobashi, Ken Anjyo, and Rikio Onai. Fluid volume modeling from sparse multi-view images by appearance transfer. *ACM Transactions on Graphics (TOG)*, 34(4):1–10, 2015. 2

[31] Omar E. Olarte, Jordi Andilla, Emilio J. Gualda, and Pablo Loza-Alvarez. Light-sheet microscopy: a tutorial. *Adv. Opt. Photon.*, 10(1):111–179, 2018. 2

[32] JL Partridge, Adrien Lefauve, and Stuart B Dalziel. A versatile scanning method for volumetric measurements of velocity and density fields. *Measurement Science and Technology*, 30(5):055203, 2019. 2

[33] Sheng Qiu, Chen Li, Changbo Wang, and Hong Qin. A rapid, end-to-end, generative model for gaseous phenomena from limited views. In *Computer Graphics Forum*, pages 242–257. Wiley Online Library, 2021. 1, 2

[34] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 5

[35] Joanthan Schwartz, Chris Harris, Jacob Pietryga, Huihuo Zheng, Prashant Kumar, Anastasia Visheratina, Nicholas Kotov, Brianna Major, Patrick Avery, Peter Ercius, Utkarsh Ayachit, Berk Geveci, David Muller, Alessandro Genova, Yi Jiang, Marcus Hanwell, and Robert Hovden. Real-time 3d analysis during electron tomography using tomviz. *Nature Communications*, 13, 2022. 5

[36] Nils Thuerey and Tobias Pfaff. Mantaflow. 2018. 5

[37] Huamin Wang, Miao Liao, Qing Zhang, Ruigang Yang, and Greg Turk. Physically guided liquid surface modeling from videos. *ACM Trans. Graph.*, 28(3), 2009. 2

[38] You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Transactions on Graphics (TOG)*, 37(4):95, 2018. 5

[39] Xianmin Xu, Yuxin Lin, Haoyang Zhou, Chong Zeng, Yaxin Yu, Kun Zhou, and Hongzhi Wu. A unified spatial-angular structured light for single-view acquisition of shape and reflectance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2023. 6, 1

[40] Guangming Zang, Ramzi Idoughi, Congli Wang, Anthony Bennett, Jianguo Du, Scott Skeen, William L. Roberts, Peter Wonka, and Wolfgang Heidrich. Tomofluid: Reconstructing dynamic fluid from sparse view videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[41] J Zhang, R Peng, S Chang, JP Lu, and O Zhou. Imaging quality assessment of multiplexing x-ray radiography based on multi-beam x-ray source technology. In *Medical Imaging 2010: Physics of Medical Imaging*, pages 1450–1457. SPIE, 2010. 2

[42] Alessandro Zunino, Francesco Garzella, Alberta Trianni, Peter Saggau, Paolo Bianchini, Alberto Diaspro, and Martí Duocastella. Multiplane encoded light-sheet microscopy for enhanced 3d imaging. *ACS Photonics*, 8(11):3385–3393, 2021. 3

Real-time Acquisition and Reconstruction of Dynamic Volumes with Neural Structured Illumination

Supplementary Material

9. Supplementary Video

The video mainly consists of two parts: reconstruction results of 4 captured sequences, and comparisons with different methods on the reconstruction of a synthetic sequence.

For the first part of the video, the layout is as follows. We show the captured input images from each camera on the left. The captured image at a non-input view is in the center, with a time code displayed in the lower-left corner. And the 3D reconstruction at a view close to the center non-input view is visualized on the right.

For the second part, we compare with PINF [10] and GlobalTrans [15] on a synthetic smoke sequence with 190 frames. The same 3 input cameras are used for all methods. In the video, we compare the reconstruction results, rendered at one input view and a novel view. Quantitative errors in SSIM/PSNR/RMSE are reported at the bottom-right corner of each rendered volume. Note that SSIM and PSNR measure the 2D error of the rendered volume at a view, while RMSE measures the error over the entire 3D volume. In addition, the errors averaged over all frames are reported in Tab. 1 and 2. In all cases, our approach outperforms competing approaches in terms of result quality.

We also compare the computation time of different methods on reconstructing the synthetic sequence. For a fair comparison, we conduct all profiling experiments on a single GeForce RTX 3090 for back-compatibility with GlobalTrans, whose code cannot be executed on RTX 4090 as in our main paper. The results are 13 seconds, 13 hours and 84 hours for our approach, PINF [10] and GlobalTrans [15], respectively.

View	Ours	PINF[10]	GlobalTrans[15]
Input(1/3)	0.98/34.36	0.96/29.04	0.96/28.66
Novel	0.97/33.15	0.95/29.83	0.94/27.14

Table 1. Comparison with different methods on reconstruction quality (SSIM/PSNR) of a synthetic sequence. We list the reconstruction errors averaged over all frames shown in the final part of the supplementary video. The second row shows the reconstruction errors for one of the three input views (i.e., cam_0). The situation with other input views is similar. The third row is the reconstruction errors for a novel non-input view.

Ours	PINF[10]	GlobalTrans[15]
1.20×10^{-2}	2.72×10^{-2}	2.50×10^{-2}

Table 2. Comparison with different methods on reconstruction quality (RMSE) of a synthetic sequence. The RMSE is computed as the error averaged over each reconstructed 3D volume.

10. Calibrations

10.1. Geometric Calibration

We calibrate the intrinsic and extrinsic parameters of the projector and the cameras in the following 4 steps.

(1) We pre-calibrate the intrinsic parameters of all cameras with a chessboard pattern.

(2) We pre-calibrate the intrinsic parameters of the projector using a calibration board with printed ARTags and one of the cameras. Please refer to Fig. 13-a for an illustration. We cast vertical and horizontal lines from the projector to the board (Fig. 13-c), and take pictures with the camera. In each captured image, the screen-space coordinates of each intersection can be estimated with sub-pixel accuracy, and the extrinsic parameters of the board can be computed from the ARTags. With the additional help of the camera intrinsic parameters from the previous step, we calculate the camera-space 3D positions of each intersection. We repeat this process for different combinations of rotated board/camera. The 3D positions of the intersections along with their 2D counterparts on the projector plane are used to compute the intrinsic parameters of the projector.

(3) We pre-calibrate the extrinsic parameters of the projector and all cameras with the calibration board. The board is rotated to different angles, one at a time (Fig. 13-b). Just like in step (2), we cast vertical and horizontal lines to the board. With the intrinsic parameters of each camera, we calculate the camera-space 3D positions of each intersection. The 3D positions of all intersections at each camera view are then used to compute the extrinsic parameters of the corresponding camera with respect to the projector.

(4) Similar to existing work [39], all pre-calibrated parameters are jointly fine-tuned in an end-to-end fashion with differential optimization, by minimizing the reprojection error of each intersection at each camera view. We fine-tune the intrinsic and extrinsic parameters for 20,000 epochs with a learning rate of 10^{-3}.

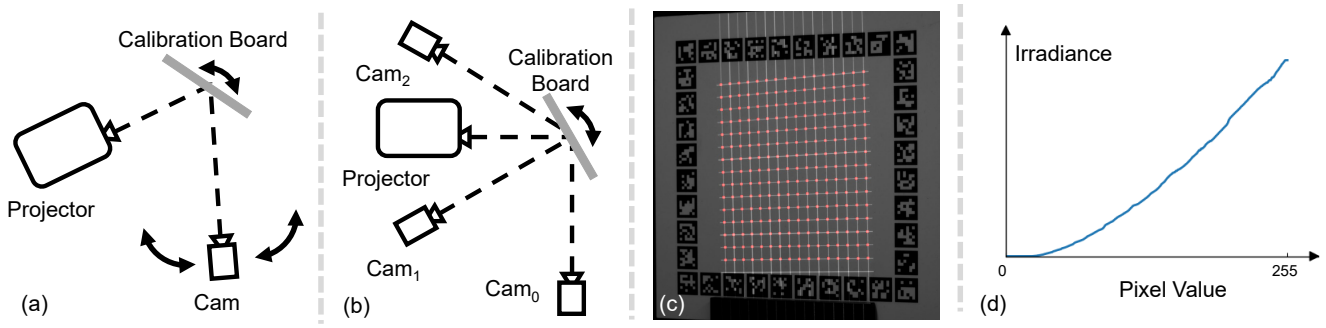

Figure 13. Geometric calibration (a-c) and the projector response curve (d). (a) Pre-calibration of the intrinsic parameters of the projector. (b) Pre-calibration of the extrinsic parameters of the projector and all cameras. (c) A photograph of the calibration board with projected horizontal and vertical lines. The reprojected intersections points are marked in red. (d) The projector response curve.

10.2. Radiometric Calibration

Our machine vision cameras can be set up to employ a linear response curve. For the projector, we directly capture its response curve as follows. We cast uniform patterns onto the calibration board, with the projector pixel intensity changes from 0 to 255. For each such pattern, we record the pixel intensity averaged over a square region observed by one calibrated camera. The collection of all pairs of projector/camera pixel intensity is the response curve, as plotted in Fig. 13-d. To linearize the projector, we apply the standard approach of inverting a 1D cumulative distribution function computed from the response curve.

10.3. Synchronization

All cameras are synchronized via a hardware trigger. In addition, we project 3 special tags along with each light pattern to facilitate projector-camera synchronization, as our projector does not support external triggers. Please refer to the inset of Fig. 2-a for tag examples.

Specifically, each tag is a white box. The center tag only appears with the first light pattern, to mark the start of our group of patterns. The left tag is projected with each odd-numbered pattern, while the right with each even-numbered pattern. An ideal synchronization will result in either the left or right tag in a captured image. If this is the case, the synchronization is finished. Otherwise, both boxes of different intensities can be observed. We then estimate the offset to the starting time of one exposure, by dividing the observed intensities by pre-calibrated intensities of the white boxes. Finally, we add this offset as a feedback to a proportional–integral–derivative (PID) algorithm, to adjust the start time of the exposure. Once the algorithm converges, the synchronization is done and we can start to capture the physical phenomenon.