Variable Imaging Projection Cloud Scattering Tomography

Roi Ronen, Vadim Holodovsky and Yoav Y. Schechner

Abstract—Scattering-based computed tomography (CT) recovers a heterogeneous volumetric scattering medium using images taken from multiple directions. It is a nonlinear problem. Prior art mainly approached it by explicit physics-based optimization of image-fitting, being slow and difficult to scale. Scale is particularly important when the objects constitute large cloud fields, where volumetric recovery is important for climate studies. Besides speed, imaging and recovery need to be flexible, to efficiently handle variable viewing geometries and resolutions. These can be caused by perturbation in camera poses or fusion of data from different types of observational sensors. There is a need for fast variable imaging projection scattering tomography of clouds (VIP-CT). We develop a learning-based solution, using a deep-neural network (DNN) which trains on a large physics-based labeled volumetric dataset. The DNN parameters are oblivious to the domain scale, hence the DNN can work with arbitrarily large domains. VIP-CT offers much better quality than the state of the art. The inference speed and flexibility of VIP-CT make it effectively real-time in the context of spaceborne observations. The paper is the first to demonstrate CT of a real cloud using empirical data directly in a DNN. VIP-CT may offer a model for a learning-based solution to nonlinear CT problems in other scientific domains. Our code is available online.

Index Terms—Inverse rendering, Physics-based Vision, Scientific imaging, Participating media

1 Introduction

Computational photography tasks such as image analysis and optical system engineering are traditionally formulated as physics-based optimization. There is a model for imaging, and this model is inverted by optimizing a cost as a function of sought variables. These variables can be those of the scene, the optical system, or both - in end-to-end optimization. This traditional approach is often slow. Thus computational photography significantly benefits from the development of deep neural networks (DNNs). There, DNNs prove useful in outcome quality and processing speed. Moreover, thanks to their speed and adaptability, DNNs that analyze image data can be optimized jointly, end-to-end, with the optical system that acquires the data.

DNNs show significant utility when the imaging model is linear. This includes tasks as denoising, deblurring [1], linear computed tomography (CT) [2], [3], [4] and localization of microscopic emitters [5], [6]. We believe that DNNs may have even a stronger advantage in solving inverse problems where the imaging model is complex, as such models are harder and slower to use by traditional means. Indeed, DNNs are introduced to nonlinear imaging models, including phase retrieval [7], [8] and ptychography [9], [10]. There, the nonlinear model is simple and easily differentiable.

Even more complex models are those that include multiple-scattering in a translucent volumetric[1] medium [16]. There, the image formation model is recursive and computationally complex, making practical differentiation a non-trivial approximation [17], [18]. A DNN is helpful there: Che et al. [19] used a trained DNN to recover the bulk properties of a homogeneous medium. However, in this paper, we take on the bigger challenge of recovering a three dimensional (3D) heterogeneous volumetric object, whose extinction coefficient per voxel is unknown. Thus, the imaging model has high complexity due to both recursive nonlinearity and large scale.

Volumetric reconstruction of scattering media is important for a range of scientific domains, including biological microscopy [20], [21], [22], tissue imaging [23], [24], [25], [26] and atmospheric science [27], [28]. The latter involves critical climate questions that are linked to the unknown volumetric structure of warm clouds. Biases there stem from a traditional approximation that the atmosphere (including clouds) is made of infinite layers, having only vertical variations [29]. 3D volumetric recovery of clouds is achievable by scattering-based CT (a highly nonlinear, recursive problem). Scattering-based CT uses two dimensional (2D) images (view projections) of the scene from multiple directions, using incoherent radiation. However, most techniques proposed so far use a traditional physics-based optimization [30], [31], [32], [33], [34], which is slow and unscalable.

Recently, various works utilized machine learning to assess atmospheric radiative properties, using single-view image data [35], [36], [37]. Wertheimer et al. [38] suggested a DNN for recovering a heterogeneous scattering volume having a few degrees of freedom, using coherent lighting. The closest work to ours is that of Sde-Chen et al. [39], whose DNN-based system (3DeepCT) performs CT of clouds, having tens of thousands of unknown voxels' values, using scattering of incoherent light. It is very fast. However, it is only trained for a fixed geometry. It has no flexibility for variability of projections or resolution.

We present variable imaging projection cloud scattering tomography (VIP-CT), to handle the mentioned issues. It solves a highly complex, nonlinear problem in computational photography: scattering-based CT, at large scales; It

1. Some 3D shape estimation [11] methods represent geometry of *opaque* objects as volumetric, for rendering and reconstruction [12], [13], [14], [15]. Our paper focuses on non-opaque objects.

does so fast, thanks to a DNN; It has flexibility for imaging using variable projections and resolutions. Moreover, VIP-CT yields much lower errors than prior art (traditional physics-based optimization, 3DeepCT). VIP-CT has a model that trains on thousands of labeled scenes derived by rigorous physics-based simulations. The model learns to infer the extinction coefficient per voxel. The inputs are the voxel location, imaging geometry and learned image features in pixels that correspond to the voxel.

2 Background

Multi-view images serve as data to CT recovery of a volumetric distribution of object properties. Most established CT problems are linear (or can be linearized) in the object properties [40], [41], [42]. Then, algebraic methods are useful [43]. When scattering is a significant portion of the signal, image formation is based on 3D radiative transfer, which is nonlinear in the object variables. Thus, scattering CT has been formulated as a nonlinear optimization problem, solved using gradient-based methods. Recently, learning-based methods have been proposed, both for approximating image formation and for solving the inverse (tomography) problem. This section reviews this background.

2.1 3D Radiative Transfer

Denote 3D location by $\mathbf{X}$ and 3D propagation direction of radiation by unit vector $\boldsymbol{\omega}$. A scattering event changes the propagation direction from $\boldsymbol{\omega}'$ to $\boldsymbol{\omega}$. The normalized distribution of scattered radiation is set by a dimensionless scattering phase function $\mathrm{p}(\mathbf{X}, \boldsymbol{\omega} \cdot \boldsymbol{\omega}')$. Scattering constitutes a portion of the radiance incident at $\mathbf{X}$. This portion depends on two variables that characterize matter at $\mathbf{X}$. One is the extinction coefficient $\beta(\mathbf{X}) \geq 0$. The extinction coefficient expresses how optically thick the matter is.[2] The other variable is the single scattering albedo $0 \leq \varpi(\mathbf{X}) \leq 1$. For example, in vacuum $\beta(\mathbf{X}) = 0$. In a totally absorbing black voxel, $\varpi = 0$. On the other hand, in notable situations $\varpi \sim 1$, i.e., absorption is negligible. These include interaction of visible light with air molecules and cloud water droplets, and interaction of near-infrared light with tissue.

Radiance $I(\mathbf{X}_0, \boldsymbol{\omega})$ is incident at the medium in direction $\boldsymbol{\omega}$ at boundary point $\mathbf{X}_0$. This radiance then undergoes multiple scattering in many paths of interaction which include absorption and/or scattering events, ultimately yielding the scene's radiance field $I(\mathbf{X}, \boldsymbol{\omega})$ at each location and direction. This process is described by the coupled recursive 3D radiate transfer equations [44],

$$I(\mathbf{X}, \boldsymbol{\omega}) = I(\mathbf{X}_0, \boldsymbol{\omega}) T(\mathbf{X}_0, \mathbf{X}) + \exp\left[- \int_{\mathbf{X}_0}^{\mathbf{X}} J(\mathbf{X}', \boldsymbol{\omega}) \beta(\mathbf{X}') T(\mathbf{X}', \mathbf{X}) d\mathbf{X}' \right] , \quad (1)$$

$$J(\mathbf{X}, \boldsymbol{\omega}) = \frac{\varpi(\mathbf{X})}{4\pi} \int_{4\pi} \mathrm{p}(\mathbf{X}, \boldsymbol{\omega} \cdot \boldsymbol{\omega}') I(\mathbf{X}, \boldsymbol{\omega}') d\boldsymbol{\omega}' , \quad (2)$$

2. Often, an object voxel contains a mixture of particle types, each with its own properties (extinction coefficient, phase function, single scattering albedo). In a cloud, a voxel can contain water droplets, air molecules and aerosols. The generalization of the model of this section to such a case is explained in Levis et al. [31].

where

$$T(\mathbf{X}_1, \mathbf{X}_2) = \exp\left[- \int_{\mathbf{X}_1}^{\mathbf{X}_2} \beta(\mathbf{X}) d\mathbf{X} \right] \quad (3)$$

is the transmittance between any two points $\mathbf{X}_1$, $\mathbf{X}_2$. The field $J(\mathbf{X}, \boldsymbol{\omega})$ is termed the *source function* in the literature.

As seen in Eqs. (1,2,3), the radiance field I integrates factors of β and T, the latter being exponential in β. Hence, I is nonlinear in $\beta(\mathbf{X})$. In linear CT, typically used in medical imaging, Eq. (1) can be linearized by assuming a non-scattering medium, that is, $J = 0$. Another linearization approach is done in *diffusion optical tomography* (DOT). DOT models the light propagation, usually in biological tissue, by an assumption that radiance scatters nearly isotropically in an optically thick medium, after sufficient scattering events. Then, under this approximation, Eq. (1) leads to a linear relation between I and β. The diffusion approximation significantly increases the speed of radiative transfer simulations of optically thick media [45]. Recently, [46] suggested using the diffusion approximation inside the core of optically thick clouds. However, more detailed 3D radiative transfer in the outer shell of the cloud is still needed. In this paper, we simulate the scene's radiance field using a physics-based 3D radiative transfer solver (SHDOM) [47], without resorting to the diffusion limit.

This paper deals with cases where the radiance incident on the object $I(\mathbf{X}_0, \boldsymbol{\omega})$ is known. Moreover, we focus on cases where $\mathrm{p}(\mathbf{X}, \boldsymbol{\omega} \cdot \boldsymbol{\omega}')$ and ϖ are approximately known. Thus the object variables constitute the extinction field $\beta(\mathbf{X})$. Let us sample $\beta(\mathbf{X})$ in a voxel grid, yielding a vector $\boldsymbol{\beta}$ that approximately represents the scene's variables across the domain. The scene radiance relates to this approximate representation using 3D radiative transfer (Eqs. 1,2,3), which we denote here in brief as an operator $\mathcal{R}$:

$$I(\mathbf{X}, \boldsymbol{\omega}) \approx \mathcal{R}(\boldsymbol{\beta}) . \quad (4)$$

2.2 Imaging by projection and sampling

Imaging projects the 3D spatial domain to a 2D image domain, then samples the radiance field by discrete pixels. The relation is both geometric and radiometric. See Fig. 1. Geometrically, camera $c \in [1 \dots N^{\mathrm{cam}}]$ has an array of pixels, indexed p. The 2D location of pixel p is $\mathbf{x}_p$. The camera has a 3D center of projection at $\mathbf{X}_{c,p}$. In a perspective camera, the center of projection is independent of p, i.e., $\mathbf{X}_{c,p} = \mathbf{X}_c$. In remote sensing, a *pushbroom camera* is common: it advances along a *track*. There, $\mathbf{X}_{c,p}$ is common to all pixels and voxels across a track, but varies along track [48].

At camera c, an image pixel $\mathbf{x}_p$ and the center of projection $\mathbf{X}_{c,p}$ uniquely define a line of sight (LOS), having direction $\boldsymbol{\omega}_{c,p}$. This LOS projects a 3D coordinate $\mathbf{X}$ in the object domain to the image plane. The projection operation at camera c is expressed by an operator

$$\mathbf{x} = \pi_c(\mathbf{X}) . \quad (5)$$

Note that $(\mathbf{X}_{c,p} - \mathbf{X})$ is parallel to the unit vector $\boldsymbol{\omega}_{c,p}$ and defines the LOS. Moreover, the pixel p in image c is uniquely defined by $\mathbf{X}$ and the center of projection. Hence, stating $\boldsymbol{\omega}_{c,p}$ is redundant. We thus sometimes express the LOS geometry using $\mathbf{X}_c|\mathbf{X}$, instead of $(\mathbf{X}_{c,p}, \boldsymbol{\omega}_{c,p})$.

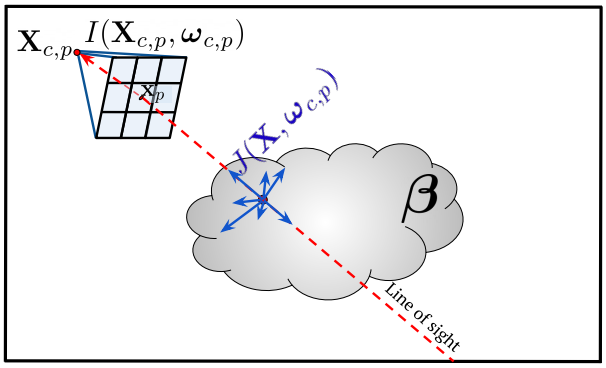


Fig. 1. Light scatters in a medium having a spatially varying extinction field $\beta(\mathbf{X})$, where $\mathbf{X}$ is a 3D location. Scattering generally occurs multiple times. Radiative transfer creates a source function J (Eq. 2) and a radiance field I (Eq. 1). In camera c, a line of sight passing though $\mathbf{X}$ projects to pixel p, through a center of projection $\mathbf{X}_{c,p}$, along direction $\boldsymbol{\omega}_{c,p}$. In a perspective camera, there is a unique center of projection, $\mathbf{X}_{c,p} = \mathbf{X}_c$. Imaging involves projection and sampling of the radiance field, measuring $I(\mathbf{X}_{c,p}, \boldsymbol{\omega}_{c,p})$.

Radiometrically, the LOS samples the radiance field (given in Eq. 1) at $\mathbf{X}_{c,p}$ and in direction $\boldsymbol{\omega}_{c,p}$. This yields

$$y_c(\mathbf{x}_p) = I(\mathbf{X}_{c,p}, \boldsymbol{\omega}_{c,p}) \quad . \tag{6}$$

Accounting for all such sampling operations in all viewpoints and all pixels, imaging transforms the field $I(\mathbf{X}, \boldsymbol{\omega})$ into a data array $\mathbf{y}$ by a *measurement operator* [31]

$$\mathbf{y} = \mathcal{M}[I(\mathbf{X}, \boldsymbol{\omega})] \quad . \tag{7}$$

Compounding Eqs. (4,7), a *forward model* operator $\mathcal{F}(\boldsymbol{\beta})$ relates the object to the measurements, accounting for 3D radiative transfer, projections and sampling:

$$\mathbf{y} \approx \mathcal{M}[\mathcal{R}(\boldsymbol{\beta})] \equiv \mathcal{F}(\boldsymbol{\beta}) \quad . \tag{8}$$

Note that the radiative transfer operator $\mathcal{R}$, radiance field I and source function J are independent of the imaging process and the recovery method. Furthermore, $\mathcal{R}$ (radiative transfer) is a law of nature. However, the measurement operator $\mathcal{M}$, thus $\mathcal{F}$, depends on the viewing geometry.

2.3 Optimization-based scattering tomography

Prior art on scattering CT [32], [49], [50], mainly relies on iterated gradient-based optimization, using explicitly the physics-based model. A cost function is defined

$$\mathcal{E}\left[\mathbf{y}, \mathcal{F}(\boldsymbol{\beta})\right] = \frac{1}{2}\|\mathbf{y} - \mathcal{F}(\boldsymbol{\beta})\|_2^2 \, . \tag{9}$$

Then, tomography is formulated by

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \, \mathcal{E}\left[\mathbf{y}, \mathcal{F}(\boldsymbol{\beta})\right] \quad . \tag{10}$$

To solve Eq. (10), prior art used various methods to approximate the Jacobian $\partial\mathcal{F}/\partial\boldsymbol{\beta}$ in each optimization iteration.

This approach has several drawbacks[3]. The solution depends on an initialization, because $\mathcal{F}(\boldsymbol{\beta})$ is nonlinear, making $\mathcal{E}$ multi-modal. Furthermore, computations of $\mathcal{F}(\boldsymbol{\beta})$ and its Jacobian turn out to be complex and slow. Due to these reasons, the approach has so far been difficult to scale [51], [52].

3. These drawbacks are further discussed in Sec. 2.1 of Sde-Chen et al. [39], especially the nonlinearity of the problem. Ref. [39] shows specifically in its supplementary material that a linear CT solution is incompatible with our problem.

2.4 3DeepCT

To bypass drawbacks of optimization-based solutions to scattering CT, an approach based on DNNs is explored. This is the motivation of this paper. A highly related work has recently been published. Thus, it is our main reference for comparison, benefiting from its public domain code and data [53]. That work is on the 3DeepCT system [39]. It is specifically designed for passive tomography of clouds.

A learning-based system as 3DeepCT can yield a fast stand-alone learning-based solution. Alternatively, Sde-Chen et al. [39] suggested a *hybrid system*, where a learning-based system provides an initialization for a physics-based optimizer [31], as in Sec. 2.3. This is contrary to a default initialization for a physics-based optimizer [31] by a constant $\boldsymbol{\beta}^{\text{initial}}$. Ref. [39] also suggested a *quick hybrid system*, which is similar to the hybrid system, but uses only 10 physics-based optimization iterations. We test these approaches in our work as well.

3DeepCT is simple. All multi-view images become input channels, forming a 3D array. The output is an estimated 3D array $\hat{\boldsymbol{\beta}}$. The input images have the *same* dimensions as the horizontal dimensions of the estimated 3D array, and so are the lateral dimensions of all intermediate layers in the DNN. Training is obtained by the following steps:
(a) There is a database of physics-based simulated clouds.
(b) For each training cloud, the extinction $\boldsymbol{\beta}^{\text{true}}$ is known.
(c) For each training scene, the forward model $\mathcal{F}$ is run, including 3D radiative transfer, projection *only according to a perspective model* and sampling. This yields a set of multi-view labeled data images.
(d) Using these images as input to the system and $\boldsymbol{\beta}^{\text{true}}$ as the labelled output, the DNN is trained to minimize the mean square error of the sought variables, $\|\boldsymbol{\beta}^{\text{true}} - \hat{\boldsymbol{\beta}}\|_2^2$.

Thanks to its simplicity, 3DeepCT executes inference very fast: it is about 5 orders of magnitude faster than optimization-based methods. However, it suffers from several drawbacks, which our approach overcomes:
(i) Due to the equal array size of the input and output, 3DeepCT strictly couples the image size and resolution to the horizontal size and resolution of the scene. This lack of flexibility is a major limitation for CT. Our proposed approach (VIP-CT) alleviates the strict dimension constraints.
(ii) 3DeepCT is an expert system for a particular viewing geometry, having fixed perspective viewpoints. However, often in reality, viewpoints vary. For example, in remote sensing, images are taken from aircraft or satellites at locations that vary during flight. To handle this variability, 3DeepCT would need to re-train from scratch for any possible variation of viewpoints, which is impractical. Moreover, 3DeepCT does not accept data which is not perspective. Our proposed approach, on the other hand, has very significant flexibility to the viewpoint geometry, as we show, and is not tied to perspective projection.
(iii) Let the image width be W. The number of layers in 3DeepCT increases linearly with W. This, in conjunction to the array-dimension coupling, means that for any change of resolution, a completely new system has to be trained.

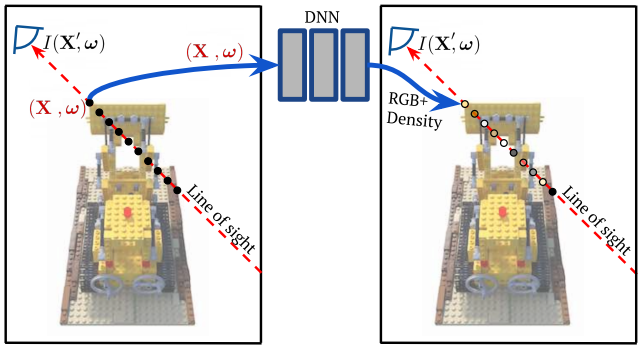


Fig. 2. NeRF [54] associates the coordinates $(\mathbf{X},\boldsymbol{\omega})$ to an *R-G-B-Density* vector, using a fully-connected DNN. Both training and inference (novel view synthesis) sample the radiance field by cameras, and associate a pixel value by an integral along $\boldsymbol{\omega}$ across $\mathbf{X}$.

2.5 Neural Radiance Fields

Neural radiance fields (NeRF) [54] is a representation of the radiance field at sampled viewpoints, using a DNN. This representation provides impressive view synthesis, after training on a specific object or a narrow class of objects [54], [55], [56]. This training can be long (hours per scene). Our image formation model has some partial analogies to NeRF, hence we considered using it. Eventually, we use some ideas developed for NeRF, though we differ in others.

Consider Fig. 2. A 3D point and a direction $(\mathbf{X},\boldsymbol{\omega})$ are mapped by NeRF to an *R-G-B-Density* vector. Considering Figs. 1 and 2, there are analogies. The *Density* component has loose analogy to our β. The *R-G-B* components of the vector have a loose analogy to our source function J. NeRF converts the *R-G-B-Density* values per voxel and viewing direction, to model samples of radiance field at specific camera locations and viewpoints, as defined in Eq. (6). This conversion does *not* use full 3D radiative transfer. Instead, it uses a line integral on a LOS, in analogy to single-scattering or emission models of image formation.

Though conversion of *R-G-B-Density* to radiance samples (multiview images) is by a model which is unrelated to correct physics of radiative transfer, it is often *not* a concern in NeRF. The reason is that the main goal of NeRF is view synthesis, rather than a physical recovery of a true volumetric multiply-scattering object. Hence, the mapping $(\mathbf{X},\boldsymbol{\omega})$ to *R-G-B-Density* is learned by penalizing image inconsistency, rather than errors of the 3D object. There is, however, evidence [57] that ignoring multiple scattering in a volumetric object can bias view synthesis.

In contrast, we fundamentally seek recovery of an actual physical field, $\boldsymbol{\beta}$. Moreover, the field $\boldsymbol{\beta}$ physically governs the radiance field and sampled images using full 3D radiative transfer (Eqs. 1,2,3) including multiple scattering. We thus do not use NeRF as a model. Moreover, we need a *scalable* system to analyze fast highly diverse and large scenes, without long training per object or a narrow class. Clouds are very diverse: they are formed by chaotic air flows. Thus, each cloud is very different than other clouds, i.e., it is a very broad class. In VIP-CT, inference of each cloud takes sub-second time. Nevertheless, NeRF research spawns ideas which we find useful for our approach.

3 VIP-CT

We propose the VIP-CT network, to learn and infer scattering CT. We have made the code public domain.[4] Much of the architecture is generic and suitable to learning-based scattering CTs. We focus on clouds, so we set some hyperparameters to this class and use clouds for training and testing. Between training samples and inference, the sizes of the input and output domains, as well as the camera poses may vary. The entire system is trained in an end-to-end supervised fashion using the loss

$$\mathrm{Loss}(\boldsymbol{\beta}^{\mathrm{true}},\hat{\boldsymbol{\beta}}) = \frac{\|\boldsymbol{\beta}^{\mathrm{true}}-\hat{\boldsymbol{\beta}}\|_2^2}{\|\boldsymbol{\beta}^{\mathrm{true}}\|_2^2} \ . \tag{11}$$

Inspired by NeRF, the architecture of VIP-CT maps radiance samples at arbitrary sensor poses $\{(\mathbf{X}_{c,p},\boldsymbol{\omega}_{c,p})\}_{c=1}^{N^{\mathrm{cam}}}$ to content in a single voxel (β). The main parts of VIP-CT are illustrated in Fig. 3.

The core of VIP-CT is a decoder, which assigns each 3D voxel location $\mathbf{X}$ an estimated value of the sought unknown $\hat{\beta}(\mathbf{X})$. The decoder has two inputs. One is a vector of image features $\mathbf{v}(\mathbf{X})$ associated with this voxel. It is the output of an *image-feature extractor*, described in Sec. 3.1. The other is a set of vectors that express 3D geometry relating to the voxel location and the set of viewpoints. They are respectively denoted $\mathbf{g}^{\mathrm{domain}}(\mathbf{X})$ and $\{\mathbf{g}^{\mathrm{cam}}(\mathbf{X}_c|\mathbf{X})\}_{c=1}^{N^{\mathrm{cam}}}$, and are the output of a *geometric encoder*, described in Sec. 3.2.

Thus, the decoder executes a function

$$\hat{\beta}(\mathbf{X}) = f_{\boldsymbol{\Theta}}\left[\mathbf{v}(\mathbf{X}),\mathbf{g}^{\mathrm{domain}}(\mathbf{X}),\{\mathbf{g}^{\mathrm{cam}}(\mathbf{X}_c|\mathbf{X})\}_{c=1}^{N^{\mathrm{cam}}}\right], \tag{12}$$

where $f_{\boldsymbol{\Theta}}$ is a function set by a vector of parameters $\boldsymbol{\Theta}$. These parameters are learned.

3.1 Image-Feature Extractor

In pixelNeRF [58], image features augment spatial and angular coordinates as inputs for view-synthesis. Training an image feature extractor across multiple scenes allows [58] learning a scene prior. Inspired by [58], the inputs to the decoder of VIP-CT include *image features* that correspond to $\mathbf{X}$, enabling a fast scattering-CT in a feed-forward manner.

Here we describe image feature extraction in VIP-CT. Irrespective of the resolution or size of the image or object domains, feature extraction is done image-wise. It follows these actions:

(1) *Pre-process* each image globally in the image domain, irrespective of the 3D volumetric element $\mathbf{X}$.
(2) *Query* $\mathbf{X}$: Using Eq. (5), project voxel $\mathbf{X}$ to all cameras c (Fig. 4). This yields a set of continuous-valued image locations $\{\mathbf{x}(c)\}_{\forall c}$, which correspond to $\mathbf{X}$.
(3) *Extract image features* per $\mathbf{x}(c)$, yielding a vector $\mathbf{u}_c(\mathbf{X})$.
(4) *Concatenate* corresponding features to a single vector:

$$\mathbf{v}(\mathbf{X}) = [\mathbf{u}_1(\mathbf{X}),\mathbf{u}_2(\mathbf{X}),\ldots,\mathbf{u}_{N^{\mathrm{cam}}}(\mathbf{X})] \ . \tag{13}$$

3.1.1 Pre-processing on a discrete grid

Pre-processing is done independently on each input image. This alleviates coupling between the dimensions of the tomographic results and the number of input images, contrary to [39]. Across the whole discrete image domain, features

4. Our code is available at https://github.com/ronenroi/VIPCT

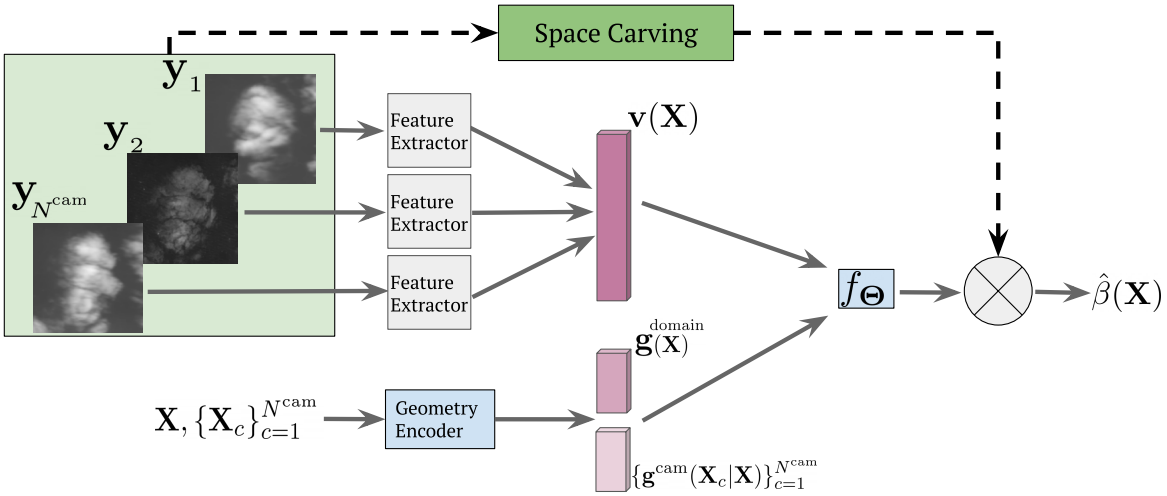


Fig. 3. The VIP-CT scheme for scattering-tomography having variable projections. A voxel location $\mathbf{X}$ is encoded to a vector $\mathbf{g}^{\text{domain}}(\mathbf{X})$. In camera c, this voxel is observed from a center of projection at a location denoted $\mathbf{X}_c|\mathbf{X}$. This location is encoded to a vector $\mathbf{g}^{\text{cam}}(\mathbf{X}_c|\mathbf{X})$. There are N^{cam} cameras observing the 3D domain from multiple viewpoints. In all images, features are extracted using the same convolutional neural network (CNN). This leads to a vector $\mathbf{v}(\mathbf{X})$ of features from all images, at image pixels that are geometric projections of $\mathbf{X}$. These vectors are passed to a decoder that infers the extinction coefficient $\hat{\beta}(\mathbf{X})$. A space-carving mask nulls voxels outside the object of interest.

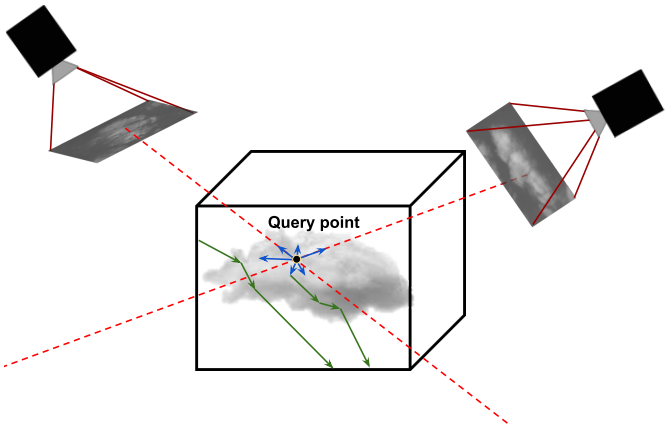


Fig. 4. Light scatters multiple times, then observed from multiple viewpoints. A 3D location $\mathbf{X}$ is queried by numerical projection of $\mathbf{X}$ to the viewpoints, pointing to specific pixels in 2D image domains.

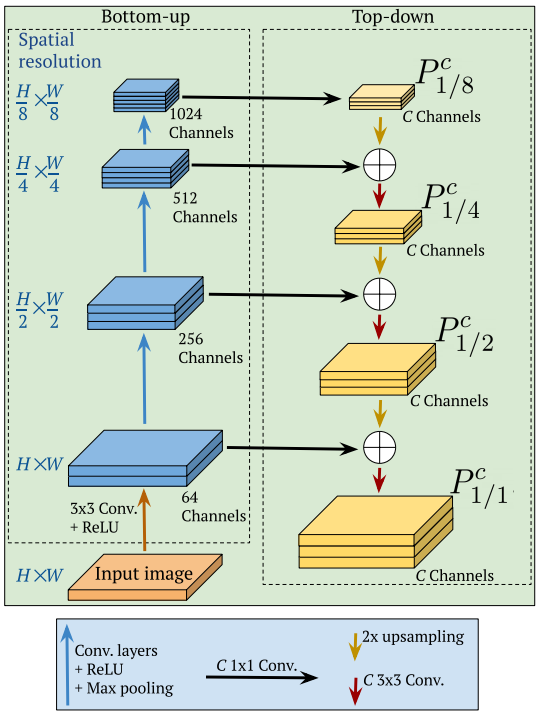


Fig. 5. A feature pyramid network (FPN) extracts image features. In the bottom-up pathway (blue arrays), the resolution of each pyramid level decreases by a factor of two in each spatial axis. The base level has full spatial resolution. This pathway relies on ResNet 50 [60]. The top-down pathway contains four levels (yellow arrays), each having C channels. Bottom-up and top-down layers of the same spatial size are merged by lateral connections.

are derived efficiently using a convolutional neural network (CNN). Moreover, the CNN is the *same* for all images. It is parallelized for all images, by stacking the image set at the batch dimension of the CNN. The CNN is learned.

As mentioned by [39], a 2D-domain CNN is a natural architecture for cloud-fields, because the statistics of cloud fields tend to be stationary. Moreover, a CNN is very efficient, making it suitable to process wide fields. Part of the features maintain the image spatial dimensions without any dimensionality reduction. This conserves the resolution and degrees of freedom of complex objects. Specifically, clouds are created by chaotic flows, and have random eddies at a huge range of scales - which matter to science. For the same reason, all convolutional layer kernels are of spatial size 3×3 or 1×1. Moreover, to obtain physical context, for any image input pixel, the corresponding feature vector has elements having a receptive field up to 244 pixels wide.

To handle objects of multiscale nature (as in clouds and cloud fields), we adapt an off-the-shelf feature pyramid network (FPN) [59] to generate features (See Fig. 5). The pyramid involves a bottom-up pathway, a top-down pathway and lateral connections. The bottom-up path relies on the ResNet 50 [60]. There are four pyramid levels. The basic level ($L = 1$) has full spatial resolution. Then, each pyramid stage spatially downsizes the domain by a factor

of two in each spatial axis.[5] Each pyramid layer has multiple channels, created by different convolution kernels.

The top-down pathway and lateral connections are as in [59]. Here, each spatially coarse feature map (higher in the pyramid) is expanded by a factor of two in each spatial axis. As a pyramid layer expands, it merges with the corresponding layer of the bottom-up pathway, creating C channels at that layer. For an input image $\mathbf{y}_c$ taken by camera c, the resulting merged layer at level L is a feature array denoted $P^{(c)}_{1/L}$. Correspondingly, all four levels of feature arrays constitute the feature set $\mathcal{P}_c = \{P^c_{1/1}, P^c_{1/2}, P^c_{1/4}, P^c_{1/8}\}$.

3.1.2 Extracting features at continuous location $\mathbf{x}(c)$

The features derived in Sec. 3.1.1 are on discrete grids, which relate to a discrete image-pixel grid. However, recall action (2) above: a *Query* yields a set of *continuous* image coordinates $\{\mathbf{x}(c)\}_{\forall c}$, corresponding to 3D voxel location $\mathbf{X}$. There is thus a need the extract features that correspond to intermediate locations between image pixels. This is done using interpolation of the feature maps $\{P^c_{1/1}, P^c_{1/2}, P^c_{1/4}, P^c_{1/8}\}$. This process alleviates the strict coupling of input image resolution to the output voxel grid. The result of this interpolation is the feature vector $\mathbf{u}_c(\mathbf{X})$, which is used in Eq. (13). The details of the interpolation process are not critical, and are thus provided in the supplementary material.

3.2 Geometry Encoder

Most objects have some spatial trends. For example, cloud voxels tend to be denser (have higher β), the higher the voxel is above a cloud base [51], yet atmospheric density generally decreases with altitude above sea level.

Moreover, radiance in the scene also tends to have spatial and directional trends. Directional trends are affected by the phase function and illumination direction. Images appear differently in back-light (forward scatter) than in back-scatter (illumination from behind the camera). For example, in clouds, the side facing towards the sun tends to be brighter than the side facing away from the sun, and clouds appear very differently if viewed from above or below.

For learning and inference to express such trends, it is required that the input of the system includes the 3D location $\mathbf{X}$ and the viewing directions. Viewing directions are set by $\{\mathbf{X}_c|\mathbf{X}\}_{c=1}^{N^{\rm cam}}$, as described in Sec. 2.2. By having these vectors as inputs to a learning system, the system learns to react to changes in the viewing geometry.

As in NeRF, spatial coordinates are not inserted raw. Rather, they are embedded in high-dimensional representations [54]. We found that this process improves performance (see supplementary material). The respective representations are in high-dimension vectors $\mathbf{g}^{\rm domain}(\mathbf{X})$ and $\mathbf{g}^{\rm cam}(\mathbf{X}_c|\mathbf{X})$.

3.3 Complexity

3.3.1 Inference

Inference is done by running Fig. 3. Inference is done independently on each voxel. Then, each voxel projects to each image independently. These processes can be parallelized, to the extent of the GPU hardware memory. If the volume domain or images are too large, then they can be divided to batches: each batch is processed in parallel, and different batches run sequentially.

A scene has $N^{\rm voxels}$ voxels. It is observed from $N^{\rm cam}$ directions. Each direction yields an image of size $H \times W$. The feature vector $\mathbf{v}(\mathbf{X})$ has length $N_{\rm v} = 4N^{\rm cam}C$. The length of $\mathbf{g}^{\rm domain}(\mathbf{X})$ is $l_{\rm domain}$. The length of $\mathbf{g}^{\rm cam}(\mathbf{X}_c|\mathbf{X})$ is $l_{\rm cam}$. So, if a whole scene is analyzed using all pixels and voxels in parallel, the memory required is $\mathcal{O}[N^{\rm voxels}(1 + l_{\rm domain}) + 4N^{\rm cam}CHW + N^{\rm cam}l_{\rm cam}]$.

3.3.2 Training

Let us list the data size. There are $N^{\rm scenes}$ training volumetric objects. Each has $N^{\rm voxels}$ with ground-truth β. Each scene yields observation data of size $N^{\rm cam}HW$. So,

$$\text{Training data size} = N^{\rm scenes}(N^{\rm voxels} + N^{\rm cam}HW). \quad (14)$$

The number of parameters that VIP-CT learns is *independent* of the domain sizes HW and $N^{\rm voxels}$. Because complexity does not increase with the domain sizes, it means that training scenes of sufficient number and domain size (Eq. 14) can potentially constrain the system sufficiently for good generalization.

• Feature extraction on a grid. Complexity is dominated[6] by the bottom-up path (Fig. 5): it uses ResNet 50, for which

$$\#\text{ feature extraction parameters} \approx 24\text{ Million}. \quad (15)$$

• Decoding. The decoder input is a vector whose number of elements is $[4N^{\rm cam}C + N^{\rm cam}l_{\rm cam} + l_{\rm domain}]$. This vector then enters a fully connected DNN. The first layer decreases the vector dimension to 2048, and subsequent layers[7] gradually decrease dimensionality to 1 (a scalar β). The number of parameters is dominated by the first layer. So,

$$\#\text{ decoder parameters} \approx 2048[4N^{\rm cam}C + N^{\rm cam}l_{\rm cam} + l_{\rm domain}]. \quad (16)$$

• Geometry encoding. Vector $\mathbf{X}$ has three elements (spatial coordinates). This vector enters a DNN having four fully-connected ReLU layers,[8] each having $l_{\rm domain}$ neurons, to yield $\mathbf{g}^{\rm domain}(\mathbf{X})$. A similar DNN converts a viewpoint $\mathbf{X}_c|\mathbf{X}$ to $\mathbf{g}^{\rm cam}(\mathbf{X}_c|\mathbf{X})$. So, geometric encoding requires

$$\#\text{ encoder parameters} \approx 3(l_{\rm domain}+l_{\rm cam})+3(l^2_{\rm domain}+l^2_{\rm cam}). \quad (17)$$

• Feature sampling at a continuous location. This interpolation has 81 parameters per camera. The overall number of parameters is $81N^{\rm cam}$, which is negligible.

We use $l_{\rm domain} = l_{\rm cam} = 64$. Sec. 4 describes data-sets and imaging settings, which clarify the balance between training data size and the number of parameters.

5. In the original ResNet 50, the basic layer downsizes by a factor of 4 each spatial axis. We modified this layer to maintain the spatial dimensions as the input image.

6. A lateral connection between layers at a corresponding level requires 256 parameters. Per level, merging involves a convolution using 9 elements, per each of C channels. These operations have $4 \times 256 + 3 \times 9C$ parameters. This number is negligible relative to the $24e^6$ parameters of ResNet 50.

7. There are nine fully-connected ReLU layers, with a skip connection from the second to the sixth layer.

8. There is a skip connection that concatenates the input to the second layer's activation, following [12].

4 Simulations

4.1 Datasets

Obtaining real-world databases of 3D cloud extinction is infeasible. Thus, we follow [39], to train using physical simulations. The simulations solve the coupled equations of a turbulent atmosphere using measured real-world boundary conditions. This yields our ground-truth cloud examples. We demonstrate our method on two cloud datasets that were presented in [39]. In both datasets [61], [62], the domains are aligned with the North, East, Up coordinate system and divided to voxels 50 m wide and 40 m thick:

BOMEX has an atmospheric domain 1.6×1.6 km wide, 1.2 km thick. Having 32 voxels along each axis, there are $N^{\text{voxels}} = 32,768$ variable extinction coefficients. Training used $N^{\text{scenes}} = 6000$ scenes. Testing used 566 scenes.

CASS has an atmospheric domain 3.2×3.2 km wide, 1.2 km thick. There are 64 voxels along each horizontal axis and 32 voxels along the vertical axis. Hence, there are $N^{\text{voxels}} = 131,072$ variable extinction coefficients. Training used $N^{\text{scenes}} = 10,908$ scenes. Testing used 1000 scenes.

4.2 Imaging settings

We follow [39] and render images of ground-truth scenes using an online open-source physics-based radiative transfer solver (SHDOM) [47]. The images are rendered based on parameters of the solar illumination direction, intrinsic and extrinsic camera parameters and noise specifications of the sensor. To compare with 3DeepCT [39], we use noise specifications derived from the CMV4000 sensor. The maximum image-pixel value is set to correspond to 90% of the sensor full well, which is 13,500 photo-electrons. Thus, sampled radiance is converted to a Poissonian distributed photo-electron count. There are 13 photo-electrons per graylevel. The readout noise has 13 electrons standard deviation (STD).

VIP-CT has significant qualitative advantages regarding flexibility to variable geometry. However, we also want to compare VIP-CT quantitatively to 3DeepCT [39]. Hence, we made simulations using the same geometric settings that are in [39] (specifically for comparisons), as well as in other geometries.

Geometries to compare with 3DeepCT:

32 Viewpoints. This is a northbound string-of-pearls [63] formation of 32 satellites, that orbit at altitude of 600km. Nearest neighboring satellites are 100km apart. The formation spans an angular viewing range $\sim \pm 75^\circ$ off-nadir. The images are perspective, have ground resolution of 50 m, with $H = W = 32$ pixels for BOMEX and $H = W = 64$ pixels for CASS. In this geometry, we use $C = 256$. Using these dimensions in Eq. (15,16,17), overall VIP-CT has ≈ 100 million parameters to train. On the other hand, from Eq. (14), BOMEX and CASS respectively provide data of sizes ≈ 400 million and ≈ 2900 million. So, the data size is very high, relative to the number of VIP-CT parameters.

In this resolution and geometry, a *Subset of Seven Clouds* drawn from the BOMEX test set is pointed out and used in [39]. This subset thus serves as a basis for some comparisons with [39].

10 Viewpoints. This geometry is motivated by the CloudCT space mission [64], [65], [66]. Here, only 10 viewpoints are used, spanning an angular viewing range $\sim \pm 40^\circ$ off-nadir, in the same orbit and inter-satellite distance of the 32-viewpoint geometry. In this geometry, we use $C = 512$. Using these dimensions in Eq. (15,16,17), overall VIP-CT has ≈ 68 million parameters to train. On the other hand, from Eq. (14), BOMEX and CASS respectively yield data of sizes ≈ 300 million and ≈ 1900 million elements.

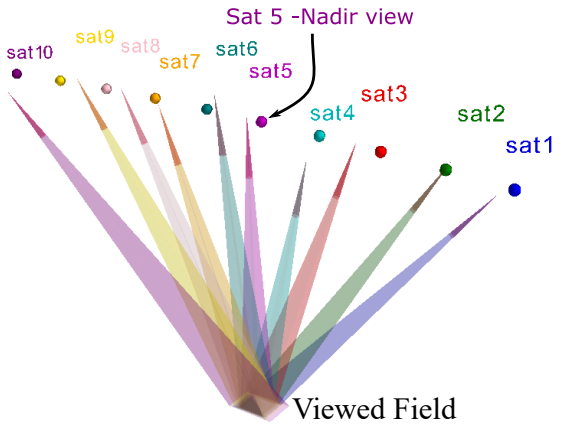


Fig. 6. Geometry of 10 perturbed viewpoints. A northbound string-of-pearls formation of observation satellites orbits at altitude of 600km. The unperturbed locations are depicted by dots, having corresponding colors to viewing angles of poses that are randomly perturbed.

TABLE 1
We randomly perturb each of the 10 perspective viewpoints, in each spatial coordinate. Here X is along the formation (string of pearls) axis, Y is across the formation axis, and Z is the altitude. The perturbation is sampled uniformly in a $\Delta X \times \Delta Y \times \Delta Z \text{km}^3$ volume.

	ΔX km	ΔY km	ΔZ km
Small (S)	± 5	± 5	± 5
Medium (M)	± 25	± 25	± 25
Large (L)	± 50	± 50	± 50
Extra-large (XL)	± 50	± 100	± 50

Novel imaging settings:

We demonstrate use of 10 randomly perturbed viewpoints. In the **10 Perturbed Viewpoints** geometry, each of the 10 viewing satellites has a pose which is translated randomly in a uniform distribution in each of the 3D spatial coordinates. The randomly perturbed camera poses are known. VIP-CT makes inferences for such perturbed views, which are not present during training. An illustration is presented in Fig. 6. In different tests, the maximum amplitude of the random translation distribution has a different magnitude, as detailed in Table 1.

3DeepCT [39] operates on a fixed horizontal grid and limited to images having 50 m ground resolution. We go beyond and render images having 20 m ground resolution. This yields images with 116×116 and 236×236 pixels for the BOMEX and CASS datasets, respectively.

4.3 Implementation details

Each input image is normalized by the mean and STD of the training data. Training and inference query voxels that reside in a space-carving [67] mask of each scene. We choose a liberal space-carving threshold, to increase the likelihood of including potential cloud voxels. Training to optimize

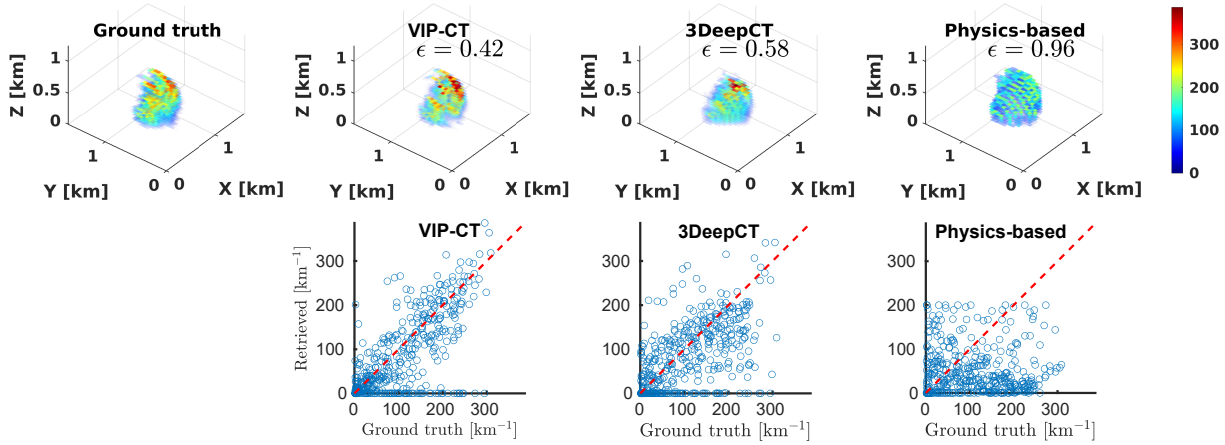


Fig. 7. [Up] 3D reconstructions of the extinction coefficient, and the relative error ϵ. [Bottom] Scatter plots of the estimated $\hat{\beta}$ across all voxels. The correlation coefficients of the scatter plot resulting from our approach (VIP-CT), 3DeepCT and physics-based are 0.73, 0.55 and 0.02, respectively.

Eq. (11) is done by $\approx 100,000$ iterations of stochastic gradient descent, using an Adam optimizer having a learning rate of 5e-5 and a weight decay of 1e-5. An iteration uses 1000 randomly sampled query voxels. The system runs on an NVIDIA GeForce RTX 3090 GPU.

We compare four approaches mentioned in Sec. 2.4. These are: a physics-based optimizer [31] initialized by a default constant value for $\boldsymbol{\beta}^{\text{initial}} = 1\text{km}^{-1}$; a stand-alone learning-based solution (VIP-CT, 3DeepCT); a *hybrid* and a *quick hybrid* system [39]. For training, as is common, we use L2-based norm (11). To quantify inference quality numerically, we follow [31], [39] and use per scene

$$\epsilon = \frac{\|\boldsymbol{\beta}^{\text{true}} - \hat{\boldsymbol{\beta}}\|_1}{\|\boldsymbol{\beta}^{\text{true}}\|_1} \ . \tag{18}$$

We calculate the mean and STD of ϵ over the test sets. The supplementary material provides further statistics using Eq. (11).

4.4 Results

4.4.1 Fixed geometry

We first show results in which the viewing geometry is fixed. We have 1566 test scenes. One example is shown in Fig. 7. Additional examples, statistics and ablation studies are given in the supplementary material. An overview of statistics of ϵ (Eq. 18) for each database and viewing geometry is shown in Fig. 8. Consistently, VIP-CT surpassed 3DeepCT [39] significantly by a large margin.

Similarly to [39], ϵ slightly decreases when N^{cam} increases from 10 to 32, but this change has just marginal significance, considering the STD of ϵ in each geometry. On the hand, VIP-CT yields significant improvement if images are taken at 20 m resolution (the voxel grid is maintained). We did not run this resolution in 3DeepCT [39], as the available 3DeepCT system is not suited to these dimensions.

It can be beneficial that a single system would train on data that is diverse, to increase the generalizability of inference. However, 3DeepCT is designed for a single output size. Hence multiple databases having different domain dimensions cannot be used simultaneously on a single 3DeepCT model. On the other hand, a single unified VIP-CT system can train on all examples of both datasets. We apply this on a 10-Viewpoint geometry and 20 m image resolution. When testing this single system, ϵ increases by 1% when tests are drawn from the CASS set, and 12% when tests are drawn from the BOMEX set, compared to VIP-CT systems that train exclusively on each.

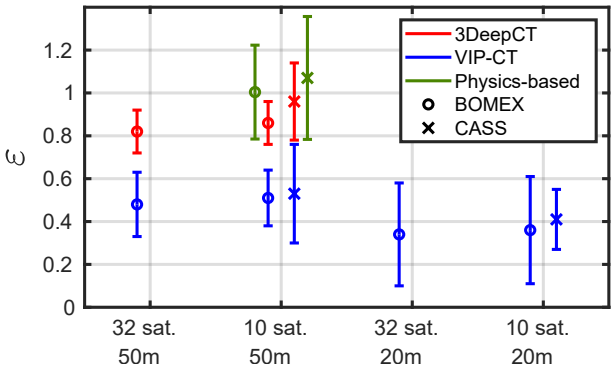


Fig. 8. Results for a fixed geometry, comparing VIP-CT to 3DeepCT and a physics-based solution [31]. Bars represent STD. In terms of ϵ, VIP-CT outperforms 3DeepCT [39] across test data-sets and geometries.

An overview of performance is provided in Fig. 9. These results are mainly based on a *Subset of Seven Clouds* described in Sec. 4.2. Physics-based optimization, based on a default initialization of a cloud having a uniform extinction coefficient yields the worse performance: it runs for 1000-2000 seconds, yielding $\epsilon \sim 0.8$. On the other extreme, using 3DeepCT as a stand-alone solution is very fast, running at several milliseconds, yielding $\epsilon \sim 0.65$. Based on initialization by 3DeepCT, minor improvement of ϵ is obtained in the *hybrid* and *quick hybrid* methods, at significantly longer runtimes, dominated by physics-based optimization iterations.

The situation is markedly better, and with small uncertainty, using our VIP-CT. On the *Subset of Seven Clouds*, using the learning-based VIP-CT as a stand-alone solution (without physics-based optimization iterations) yields quality which is far better than any prior art, at about $\epsilon \sim 0.45$. Moreover, $\epsilon \sim 0.38$ when assessed over the full test set of hundreds of scenes (not just a subset), using just 10 viewpoints, if imaging is done at 20 meter resolution. Runtime to recover a scene using the learning-based VIP-CT is

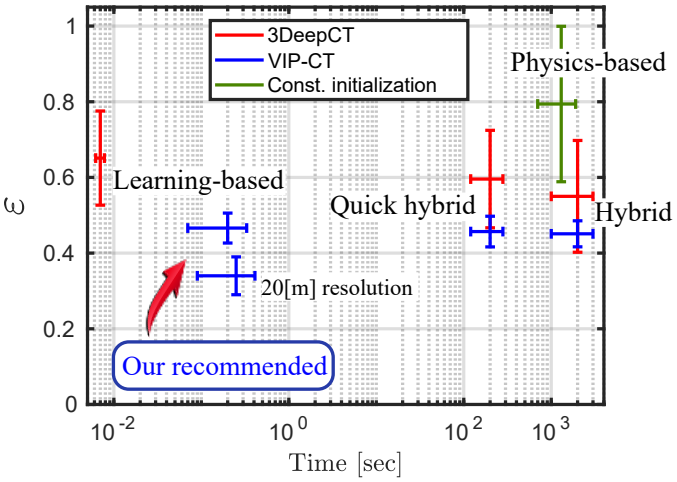


Fig. 9. A *Subset of Seven Clouds* from BOMEX tested recovery performance using several methods: physics-based optimizer (Green) initialized by a constant β, learning-based 3DeepCT (red), learning-based VIP-CT (blue), and hybrid and quick-hybrid methods, where a physics-based optimizer is initialized by the respective learning-based results. Bars express STD of ϵ and the run-time. The case of VIP-CT (our method) using 20 m image resolution is also shown. Recovery time for physics-based optimization and hybrid solutions is in the order of 1000 sec. Quick-hybrid is in the order of 100 sec. The mean inference times of VIP-CT and 3DeepCT are 0.25 sec and 0.007 sec, respectively.

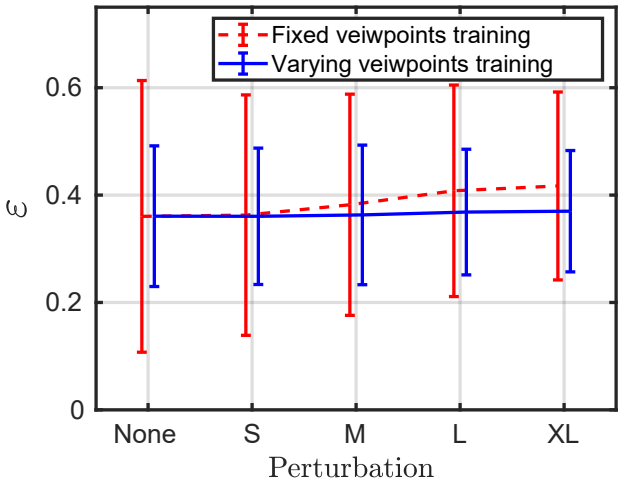


Fig. 10. VIP-CT tested on perturbed imaging geometries. The four perturbation magnitudes are described in Sec. 4.1 and Table 1. When a model is trained on fixed viewpoints, tests that include pose perturbations have some increased (though contained) degradation of results (red). A model that was trained on random perturbed viewpoints maintains an approximately constant error (blue).

≈ 0.25 seconds. This is three to four orders of magnitude faster than recovery methods that rely on physics-based optimization steps. Note that VIP-CT has more involved processes (inc. projection, sampling, decoder), lengthening its run-time in comparison to 3DeepCT. The training time for VIP-CT is similar to that of 3DeepCT: about 15 hours.

4.4.2 *Varying geometry*

We test how VIP-CT handles viewpoints having random variations (only 10 viewpoints are used). As described in Sec. 4.2 and Fig. 6, the viewpoints are perturbed. The results are presented in Fig. 10, for two models. In one model, VIP-CT trained only on fixed viewpoints, while testing has perturbed viewpoints. The system tolerates this discrepancy, attesting the flexibility and robustness of VIP-CT. However, ϵ has a significant STD. There is also a degradation of the ϵ-mean as the perturbation amplitude increases.

In a second model, VIP-CT trained on perturbed viewpoints, then also tested on variable geometry. Here, each scene in the training set had been imaged with random viewpoints having L magnitude (See Table 1). The results in Fig. 10 show that on average, ϵ is insensitive to the perturbation. Moreover, the STD of ϵ is lower, including in test scenes that have no geometry perturbations at all.

5 AirMSPI Real World Empirical Data

Levis et al. [31] and consequent works [68] on scattering tomography demonstrated methods experimentally on real-world data acquired by AirMSPI. We follow the same approach, which enables comparison to the prior art. AirMSPI is a remote sensing instrument designed, built and operated by the Jet Propulsion Laboratory (JPL). It is flown on board the ER-2 aircraft of NASA at altitude of 20 km. From this altitude, the spatial resolution of the observed domain is 10m. The AirMSPI camera has a pushbroom configuration. It can take multi-angular observations over a $\pm 67^\circ$ angular span along-track, as it flies. It has a step-and-stare mode, which sequentially acquires $N^{\text{cam}} = 9$ images of the same observation domain from that angular range. Each angular setting is indexed by $c = 1 \dots N^{\text{cam}}$.

Despite being described as pushbroom, in practice image projection does not follow this model, because the aircraft velocity vector and angular pointing are perturbed during flight. Consequently, the projection model is more general. It is known, however, because AirMSPI is supplemented by extensive geometric and radiometric calibration. Thus, for every voxel $\mathbf{X}$ imaged at a time by a camera at angular setting c, we know the location of the camera $\mathbf{X}_c$.

To train a VIP-CT model, we followed [39], to use the BOMEX-Aux data-set.[9] The liquid water content of each BOMEX cloud is multiplied by 1/10 there. A voxel in BOMEX-Aux is 50 m wide and 40 m thick. Each scene then undergoes 3D radiative transfer (SHDOM), followed by image rendering at 10 m resolution. Fig. 10 indicates that training VIP-CT on a variety of perturbed viewpoints leads to a better generalization and a higher recovery accuracy. Thus, we render the training cloud scenes using five different AirMSPI flight experiments, generating a variety of realistic perturbed viewpoints. This data then trains VIP-CT, using $C = 64$. The remaining parameters in the system are as in Sec. 4.3.

To test VIP-CT on real-world data, we use the 660 nm data channel from a flight at 20:27GMT on February 6, 2010 over a cloudy ocean scene at 32N 123W. This flight path was not included training. Because imaging is sequential, the scene shifts by global wind during the time it takes ER-2 to fly over the domain. So, we follow [68] to assess this

9. 3DeepCT is not used here, because 3DeepCT was neither designed nor trained for images taken by pushbroom or more general projection models: only on perspective inputs. In Sde-Chen et al. [39], AirMSPI data had first been analyzed by physics-based optimization to yield $\hat{\boldsymbol{\beta}}$. That estimated 3D distribution was then considered ground-truth, from which perspective images were rendered for use in 3DeepCT.

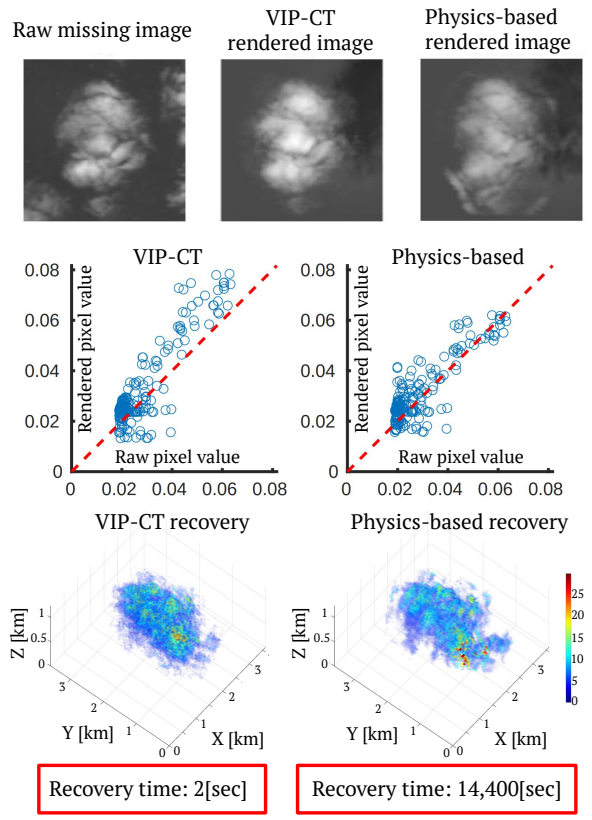


Fig. 11. Real-world experiment. [Top] An AirMSPI missing view and corresponding rendered views of a cloud that is estimated either by our VIP-CT approach or a physics-based optimizer [31]. Both use data that excludes this view. [Middle] Scatter plots of the pixel values in these views. The correlation coefficient of the scatter plot resulting from our approach (VIP-CT) is 0.8. For physics-based optimization, it is 0.84. [Bottom] Visualization of the recovered cloud by the respective methods.

global wind and back-shift the images, to reduce the effect of this drift.

The domain of our empirical scene requires a grid of $72 \times 72 \times 32$ voxels, that is, having 165,888 unknowns. We used the trained model to infer the empirical scene described above, and recover the cloud. However, there is no ground truth information on this data (nor other relevant empirical data that we sought). So, as in prior work, we check for consistency using cross-validation[10] [31], [39], [68]. For this, we excluded the $+47^\circ$ view, trained VIP-CT and performed inference of $\hat{\boldsymbol{\beta}}$ using only eight viewpoints. Then we used $\mathcal{F}(\hat{\boldsymbol{\beta}})$ to render the missing view, using Eq. (8). The results are shown in Fig. 11.

The results appear mutually comparable, however, recovery took just *two seconds* using VIP-CT. In contrast, a physics-based optimizer took *four hours*. We stress that in this experiment, the physics-based solver exploits an advantage that $\boldsymbol{\beta}^{\text{true}}$ is $\approx \times 10$ smaller than in the simulated datasets. This optical-thinness aids convergence of a physics-based solver. In the supplementary material, we show cross-validation results of a second viewpoint and additional results of a different real-data cloud.

10. The cross-validation measure has drawbacks. It requires approximating additional scene unknowns, like the cloud phase function, the cloud single-scattering albedo, and the ground albedo. Hence, higher cross-validation performance does not necessarily guarantee a better $\boldsymbol{\beta}$ recovery.

6 Further Outlook

We gain broad conclusions, in the context of computational photography (technology) and in relation to its use for science, specifically remote sensing of the atmosphere. Deep learning already excels in analysis of 2D manifolds, such as 2D images and range maps to opaque surfaces. A major reason is the availability of massive datasets of *labeled* 2D images and maps, which enable supervised training of DNNs. Sufficiently large datasets of ground-truth labeled volumetric realistic 3D translucent (scattering) heterogeneous objects have largely been missing. This has hampered progress of deep learning of such objects. We find that the datasets in [53] can start to provide a breakthrough for learning-based computational photography, dealing with heterogeneous translucent 3D volumetric objects. This is irrespective of the application. These large datasets are driven by physics (fluid dynamics, thermodynamics, condensation etc.). They may thus be used to develop DNN technologies unrelated to atmospheric sciences, possibly affecting research on computer graphics and biomedical imaging.

Large datasets have created benchmarks that propelled computer vision, enabling groups to compete in developing technologies. We hope it may happen here. Even in the context of cloud tomography, we can expect future DNN-based systems that will outperform VIP-CT, as well as 3DeepCT, using such large datasets.

For remote sensing of the atmosphere, the results can be readily pivotal. From Sec. 4.4, VIP-CT takes ≈ 0.25 seconds to tomographically recover clouds in high resolution ($\sim 20\text{m}$) and quality, in an area of $1.6 \times 1.6\text{km}^2$. Simply using the same hardware and sequentially analyzing such area patches, VIP-CT would need only 3 minutes to tomographically recover a cloud field $43 \times 43\text{km}^2$ wide.

Three minutes is essentially real-time in spaceborne observations. Typically, downlink from space is a bottleneck, often requiring longer times to receive that much data. Actually, it can take longer to acquire the data, before downlink: the MISR spaceborne instrument of NASA takes multi-angular images of the atmosphere [69], requiring seven minutes to fly above the observed region and scan it. VIP-CT shows that with existing technology, 3D tomography of the atmosphere can be achieved as soon as data is obtained, and tomography computation is *not* a time bottleneck. Very large scales and high quality are now reachable.

Moreover, VIP-CT naturally accommodates other aspects of remote sensing, including variable viewing geometries. These are created by perturbation of the platform pose, maneuvers, and fusion of data obtained by multiple types of orbiting instruments. We thus believe that thanks to a flexible learning-based approach as VIP-CT, 3D volumetric analysis is practical for operations.

Extensions of this work should include the following: adding the illumination direction as input to the decoder; recovery of a vector of parameters per voxel (single scattering albedo, particle sizes, density of particles); recovery of the albedo of the ground or ocean (boundary of the medium); and use of multimodal data, such as polarized imaging channels. Further extensions can be in the time domain, for analyzing dynamic objects and/or temporal light transport.

ACKNOWLEDGMENTS

We thank Yael Sde-Chen, Eshkol Eytan and Ilan Koren for the advice, and Ina Talmon and Daniel Yagodin for technical support. Yoav Schechner is the Mark and Diane Seiden Chair in Science at the Technion. He is a Landau Fellow supported by the Taub Foundation. His work was conducted in the Ollendorff Minerva Center. Minvera is funded through the BMBF. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (CloudCT, grant agreement No. 810370).

REFERENCES

[1] A. Chakrabarti, "A neural approach to blind motion deblurring," in *Proc. European Conference on Computer Vision*. Springer, 2016, pp. 221–235.

[2] E. D. Zhong, T. Bepler, B. Berger, and J. H. Davis, "CryoDRGN: reconstruction of heterogeneous Cryo-EM structures using neural networks," *Nature Methods*, vol. 18, no. 2, pp. 176–185, 2021.

[3] D. Wu, K. Kim, and Q. Li, "Computationally efficient deep neural network for computed tomography image reconstruction," *Medical Physics*, vol. 46, no. 11, pp. 4763–4776, 2019.

[4] A. Levis, P. P. Srinivasan, A. A. Chael, R. Ng, and K. L. Bouman, "Gravitationally lensed black hole emission tomography," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 841–19 850.

[5] E. Nehme, D. Freedman, R. Gordon, B. Ferdman, L. E. Weiss, O. Alalouf, T. Naor, R. Orange, T. Michaeli, and Y. Shechtman, "DeepSTORM3D: dense 3D localization microscopy and PSF design by deep learning," *Nature Methods*, vol. 17, pp. 734–740, 2020.

[6] E. Nehme, B. Ferdman, L. E. Weiss, T. Naor, D. Freedman, T. Michaeli, and Y. Shechtman, "Learning optimal wavefront shaping for multi-channel imaging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 7, pp. 2179–2192, 2021.

[7] P. Hand, O. Leong, and V. Voroninski, "Phase retrieval under a generative prior," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[8] C. Metzler, P. Schniter, A. Veeraraghavan *et al.*, "prDeep: robust phase retrieval with a flexible deep network," in *Proc. International Conference on Machine Learning*. PMLR, 2018, pp. 3501–3510.

[9] X. Dai, P. C. Konda, S. Xu, and R. W. Horstmeyer, "Towards a vectorial treatment of Fourier Ptychographic microscopy," in *Computational Optical Sensing and Imaging*. Optical Society of America, 2020, pp. CF2C–3.

[10] L. Tian, X. Li, K. Ramchandran, and L. Waller, "Multiplexed coded illumination for Fourier Ptychography with an LED array microscope," *Biomedical optics express*, vol. 5, pp. 2376–2389, 2014.

[11] M. Zanfir, A. Zanfir, E. G. Bazavan, W. T. Freeman, R. Sukthankar, and C. Sminchisescu, "THUNDR: Transformer-based 3D human reconstruction with markers," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 971–12 980.

[12] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 165–174.

[13] E. Grant, P. Kohli, and M. v. Gerven, "Deep disentangled representations for volumetric reconstruction," in *Proc. European Conference on Computer Vision*. Springer, 2016, pp. 266–279.

[14] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4460–4470.

[15] G. Riegler, A. Osman Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3577–3586.

[16] G. Satat, B. Heshmat, D. Raviv, and R. Raskar, "All photons imaging through volumetric scattering," *Scientific reports*, vol. 6, no. 1, pp. 1–8, 2016.

[17] I. Gkioulekas, A. Levin, and T. Zickler, "An evaluation of computational imaging techniques for heterogeneous inverse scattering," in *Proc. European Conference on Computer Vision*. Springer, 2016, pp. 685–701.

[18] M. Chen, D. Ren, H.-Y. Liu, S. Chowdhury, and L. Waller, "Multi-layer Born multiple-scattering model for 3D phase microscopy," *Optica*, vol. 7, no. 5, pp. 394–403, 2020.

[19] C. Che, F. Luan, S. Zhao, K. Bala, and I. Gkioulekas, "Towards learning-based inverse subsurface scattering," in *Proc. IEEE International Conference on Computational Photography*, 2020, pp. 1–12.

[20] A. Matlock, A. Sentenac, P. C. Chaumet, J. Yi, and L. Tian, "Inverse scattering for reflection intensity phase microscopy," *Biomedical optics express*, vol. 11, no. 2, pp. 911–926, 2020.

[21] B. Li, S. Tan, J. Dong, X. Lian, Y. Zhang, X. Ji, and A. Veeraraghavan, "Deep-3D microscope: 3D volumetric microscopy of thick scattering samples using a wide-field microscope and machine learning," *Biomedical Optics Express*, vol. 13, pp. 284–299, 2022.

[22] K. Kim, P. C. Konda, C. L. Cooke, R. Appel, and R. Horstmeyer, "Multi-element microscope optimization by a learned sensing network with composite physical layers," *Optics Letters*, vol. 45, no. 20, pp. 5684–5687, 2020.

[23] M. Shimano, Y. Asano, S. Ishihara, R. Bise, and I. Sato, "Imaging scattering characteristics of tissue in transmitted microscopy," in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2020, pp. 236–245.

[24] D. Ren, C. Ophus, M. Chen, and L. Waller, "A multiple scattering algorithm for three dimensional phase contrast atomic electron tomography," *Ultramicroscopy*, vol. 208, p. 112860, 2020.

[25] A. K. Singh, D. N. Naik, G. Pedrini, M. Takeda, and W. Osten, "Exploiting scattering media for exploring 3D objects," *Light: Science & Applications*, vol. 6, no. 2, pp. e16 219–e16 219, 2017.

[26] A. Geva, Y. Y. Schechner, Y. Chernyak, and R. Gupta, "X-ray computed tomography through scatter," in *Proc. European Conference on Computer Vision*. Springer, 2018, pp. 34–50.

[27] D. B. Lindell and G. Wetzstein, "Three-dimensional imaging through scattering media based on confocal diffuse tomography," *Nature Communications*, vol. 11, no. 1, pp. 1–8, 2020.

[28] A. E. Bourassa, D. A. Degenstein, and E. J. Llewellyn, "SASKTRAN: A spherical geometry radiative transfer code for efficient estimation of limb scattered sunlight," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 109, no. 1, pp. 52–73, 2008.

[29] A. B. Davis and A. Marshak, "Solar radiation transport in the cloudy atmosphere: a 3D perspective on observations and climate impacts," *Reports on Progress in Physics*, vol. 73, p. 026801, 2010.

[30] A. Levis, Y. Y. Schechner, A. B. Davis, and J. Loveridge, "Multi-view polarimetric scattering cloud tomography and retrieval of droplet size," *Remote Sensing*, vol. 12, no. 17, p. 2831, 2020.

[31] A. Levis, Y. Y. Schechner, A. Aides, and A. B. Davis, "Airborne three-dimensional cloud tomography," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2015, pp. 3379–3387.

[32] A. Aides, A. Levis, V. Holodovsky, Y. Y. Schechner, D. Althausen, and A. Vainiger, "Distributed sky imaging radiometry and tomography," in *Proc. IEEE International Conference on Computational Photography*, 2020, pp. 1–12.

[33] D. Veikherman, A. Aides, Y. Y. Schechner, and A. Levis, "Clouds in the cloud," in *Proc. Asian Conference on Computer Vision*. Springer, 2014, pp. 659–674.

[34] M. Tzabari, V. Holodovsky, O. Shubi, E. Eshkol, and Y. Y. Schechner, "Settings for spaceborne 3D scattering tomography of liquid-phase clouds by the CloudCT mission," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

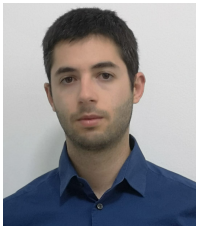
[35] V. Nataraja, S. Schmidt, H. Chen, T. Yamaguchi, J. Kazil, G. Feingold, K. Wolf, and H. Iwabuchi, "Segmentation-based multi-pixel cloud optical thickness retrieval using a convolutional neural network," *Atmospheric Measurement Techniques Discussions*, pp. 1–34, 2022.

[36] R. Masuda, H. Iwabuchi, K. S. Schmidt, A. Damiani, and R. Kudo, "Retrieval of cloud optical thickness from sky-view camera images using a deep convolutional neural network based on three-dimensional radiative transfer," *Remote Sensing*, vol. 11, no. 17, p. 1962, 2019.

[37] H. Chen, S. Schmidt, S. T. Massie, V. Nataraja, M. S. Norgren, J. J. Gristey, G. Feingold, R. E. Holz, and H. Iwabuchi, "The education and research 3D radiative transfer toolbox (EaR^{3}T)–towards the mitigation of 3D bias in airborne and spaceborne passive imagery cloud retrievals," *Atmospheric Measurement Techniques Discussions*, pp. 1–44, 2022.

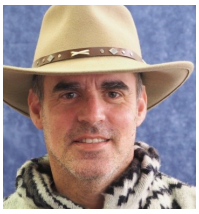
[38] Z.-A. Wertheimer, C. Bar, and A. Levin, "Towards machine learning for heterogeneous inverse scattering in 3D microscopy," *Optics Express*, vol. 30, no. 6, pp. 9854–9868, 2022.

[39] Y. Sde-Chen, Y. Y. Schechner, V. Holodovsky, and E. Eytan, "3DeepCT: Learning volumetric scattering tomography of clouds," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5671–5682.
[40] R. Ronen, Y. Attias, Y. Y. Schechner, J. S. Jaffe, and E. Orenstein, "Plankton reconstruction through robust statistical optical tomography," *Journal of the Optical Society of America A*, vol. 38, no. 9, pp. 1320–1331, 2021.
[41] A. Rosenthal, V. Ntziachristos, and D. Razansky, "Acoustic inversion in optoacoustic tomography: A review," *Current Medical Imaging*, vol. 9, no. 4, pp. 318–336, 2013.
[42] M. Alterman, Y. Y. Schechner, M. Vo, and S. G. Narasimhan, "Passive tomography of turbulence strength," in *Proc. European Conference on Computer Vision*. Springer, 2014, pp. 47–60.
[43] A. K. Trull, J. van der Horst, L. J. Van Vliet, and J. Kalkman, "Comparison of image reconstruction techniques for optical projection tomography," *Applied Optics*, vol. 57, no. 8, pp. 1874–1882, 2018.
[44] S. Chandrasekhar, *Radiative Transfer*. Courier Corporation, 2013.
[45] A. Marshak and A. Davis, *3D radiative transfer in cloudy atmospheres*. Springer Science & Business Media, 2005.
[46] L. Forster, A. B. Davis, D. J. Diner, and B. Mayer, "Toward cloud tomography from space using MISR and MODIS: Locating the "veiled core" in opaque convective clouds," *Journal of the Atmospheric Sciences*, vol. 78, no. 1, pp. 155–166, 2021.
[47] A. Levis, J. Loveridge, and A. Aides, *Pyshdom. 2020. Available online*, https://github.com/aviadlevis/pyshdom.
[48] R. Gupta and R. I. Hartley, "Linear pushbroom cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 9, pp. 963–975, 1997.
[49] A. Levis, Y. Y. Schechner, and A. B. Davis, "Multiple-scattering microphysics tomography," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6740–6749.
[50] V. Holodovsky, Y. Y. Schechner, A. Levin, A. Levis, and A. Aides, "In-situ multi-view multi-scattering stochastic tomography," in *Proc. IEEE International Conference on Computational Photography*, 2016, pp. 1–12.
[51] T. Loeub, A. Levis, V. Holodovsky, and Y. Y. Schechner, "Monotonicity prior for cloud tomography," in *Proc. European Conference on Computer Vision*. Springer, 2020, pp. 24–29.
[52] I. Czerninski and Y. Y. Schechner, "Accelerating inverse rendering by using a GPU and reuse of light paths," *arXiv preprint arXiv:2110.00085*, 2021.
[53] Y. Sde-Chen, *3DeepCT. 2022. Available online*, https://github.com/YaelSdeChen/3DeepCT.
[54] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. European Conference on Computer Vision*. Springer, 2020, pp. 405–421.
[55] B. Attal, E. Laidlaw, A. Gokaslan, C. Kim, C. Richardt, J. Tompkin, and M. O'Toole, "TöRF: Time-of-flight radiance fields for dynamic scene view synthesis," *Advances in Neural Information Processing Systems*, vol. 34, 2021.
[56] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh, "Neural volumes: Learning dynamic renderable volumes from images," *arXiv preprint arXiv:1906.07751*, 2019.
[57] Q. Zheng, G. Singh, and H. P. Seidel, "Neural relightable participating media rendering," *Advances in Neural Information Processing Systems*, vol. 34, 2021.
[58] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelNeRF: Neural radiance fields from one or few images," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4578–4587.
[59] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.
[60] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
[61] E. Eytan, A. Khain, M. Pinsky, O. Altaratz, J. Shpund, and I. Koren, "Shallow cumulus properties as captured by adiabatic fraction in high-resolution LES simulations," *Journal of the Atmospheric Sciences*, vol. 79, no. 2, pp. 409–428, 2022.
[62] R. H. Heiblum, L. Pinto, O. Altaratz, G. Dagan, and I. Koren, "Core and margin in warm convective clouds–part 1: Core types and evolution during a cloud's lifetime," *Atmospheric Chemistry and Physics*, vol. 19, no. 16, pp. 10 717–10 738, 2019.
[63] A. Kleinschrodt, N. Reed, and K. Schilling, "A comparison of scheduling algorithms for low cost ground station networks," in *67st International Astronautical Congress.*, 2016, pp. 1–15.
[64] K. Schilling, Y. Y. Schechner, and I. Koren, "CloudCT - computed tomography of clouds by a small satellite formation," in *Proc. IAA Symposium on Small Satellites for Earth Observation*, 2019.
[65] M. Tzabari, V. Holodovsky, O. Shubi, E. Eytan, O. Altaratz, I. Koren, A. Aumann, K. Schilling, and Y. Y. Schechner, "CloudCT 3D volumetric tomography: Considerations for imager preference, comparing visible light, short-wave infrared, and polarized imagers," in *Polarization Science and Remote Sensing X*, vol. 11833. SPIE, 2021, pp. 19–26.
[66] Y. Bertschy and Y. Y. Schechner, "Vicarious spaceborne polarimetric camera calibration using solar power stations," in *Polarization: Measurement, Analysis, and Remote Sensing XV*, vol. 12112. SPIE, 2022, pp. 132–142.
[67] K. N. Kutulakos and S. M. Seitz, "A theory of shape by space carving," *International Journal of Computer Vision*, vol. 38, no. 3, pp. 199–218, 2000.
[68] R. Ronen, Y. Y. Schechner, and E. Eytan, "4D cloud scattering tomography," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5520–5529.
[69] M. J. Garay, M. L. Witek, R. A. Kahn, F. C. Seidel, J. A. Limbacher, M. A. Bull, D. J. Diner, E. G. Hansen, O. V. Kalashnikova, H. Lee *et al.*, "Introducing the 4.4 km spatial resolution Multi-Angle Imaging SpectroRadiometer (MISR) aerosol product," *Atmospheric Measurement Techniques*, vol. 13, no. 2, pp. 593–628, 2020.



Roi Ronen is a Ph.D. student at the Technion - Israel Institute of Technology. His research explores the interface between machine learning, physics-based imaging and computational photography. He obtained his B.Sc. in 2019 and a M.Sc. in 2021, both at the Electrical and Computer Engineering Department, Technion, Israel. In 2021 he conducted a summer internship at Amazon Web Service - AI labs. His paper won the JOSA A Emerging Researcher Best Paper Prize for 2021.

Vadim Holodovsky received his M.Sc. degree in Electrical Engineering, Technion, Israel, in 2016. He is currently a Senior Research Associate in the Hybrid-Imaging lab, Technion, Israel. His research interests include computer vision algorithms for 3D retrievals and remote sensing. Vadim's work includes experiment designs, optics, cameras, polarization, and electronic hardware.



Yoav Y. Schechner is a graduate of the Technion - Israel Institute of Technology: BA (Physics 1990), MSc (Physics 1994), PhD (EE 2000). Afterwards, he was a research scientist at Columbia University (CS). Since 2002, he is a faculty member in the Technion's Viterbi Faculty of Electrical Engineering. He is the Diane and Mark Seiden Chair in Science. He is a principal investigator and coordinator of the CloudCT project, funded by ERC. In 2010 and 2011 he was a visiting scientist at Caltech and NASA's Jet Propulsion Laboratory (JPL). He won the Best Student Paper Award at CVPR in 2017, the Best Paper Awards at ICCP in 2013 and 2018, and the Distinguished Lecturer Award, Technion 2020. He is the recipient of the Ray and Miriam Klein Research Award, the Henry Taub Prize for Academic Excellence, the Otto Schwarz Foundation Excellence Award and the Landau Fellowship. His research interests involve outdoor phenomena and all aspects of imaging.

Supplementary Material: Variable Imaging Projection Cloud Scattering Tomography

Roi Ronen, Vadim Holodovsky and Yoav Y. Schechner

Abstract—This is a supplementary material for the main manuscript on Variable Imaging Projection Cloud Scattering Tomography (VIP-CT). First, we show an ablation study demonstrating the contribution of different parameters to the overall system performance. We further present an ablation study on the geometry encoder architecture. Then, we describe the feature sampling component of VIP-CT. Finally, we detail statistics that relate to plots in the main manuscript. This document also presents additional real-world results and visualizations of cloud reconstructions, that did not enter the main manuscript due to space limitations.

Index Terms—Inverse rendering, Physics-based Vision, Scientific imaging, Participating media

1 Ablation studies

We conducted ablation studies to evaluate the contribution of some of the components of VIP-CT. All tests in this section were done on the BOMEX dataset with a 10-Viewpoint geometry and 20 m resolution. The results are summarised in Table 1 herein.

Removing the top-down pathway in the image feature extractor increases ϵ by 6%. When excluding $\mathbf{g}^{\mathrm{domain}}(\mathbf{X})$, that is, by inference without the voxel 3D location, ϵ increases by 11%. This points to the importance of the voxel 3D location during inference. Similarly, ϵ increases when $\mathbf{g}^{\mathrm{cam}}(\mathbf{X}_c|\mathbf{X})$ is excluded. Moreover, ϵ increases by 14% when passing to the decoder only image features, that is, excluding all geometry encoding. Furthermore, there is an increase of ϵ by 10% when changing the sampler spatial support $h \times w$ from 40×40 m, which is about the voxel size, to 160×160 m.

We study the effect of parameters of the geometry encoding. Recall that a voxel location $\mathbf{X}$ and the camera location $\mathbf{X}_c|\mathbf{X}$ are encoded to the vectors $\mathbf{g}^{\mathrm{domain}}(\mathbf{X})$ and $\mathbf{g}^{\mathrm{cam}}(\mathbf{X}_c|\mathbf{X})$, correspondingly. A visualization of the coordinate encoder is presented in Fig. 1 herein.

We assess the impact of the embedding lengths $l_{\mathrm{domain}}, l_{\mathrm{cam}}$ of the domain and camera features. We train and test VIP-CT on the BOMEX dataset with Large magnitude perturbed views. Results are shown in Fig. 2 herein. Training and inference times are similar for all cases, as computing times of $\mathbf{g}^{\mathrm{domain}}$ and $\mathbf{g}^{\mathrm{cam}}$ are negligible (see Sec. 3.3 of the main manuscript).

TABLE 1
Ablation studies. The "Top-down pathway" column stands for an image feature extractor that includes (by default) the top-down pathway, as described in the main manuscript. The columns $\mathbf{g}^{\mathrm{domain}}$ and $\mathbf{g}^{\mathrm{cam}}$ columns stand for a model that (as default) encodes 3D coordinates in a learned high dimensional representation. The $h \times w$ column is the support used in feature sampling.

Model	Top-down pathway	$\mathbf{g}^{\mathrm{domain}}$	$\mathbf{g}^{\mathrm{cam}}$	$h \times w$ [m]	ϵ %
ResNet 50	✗	✓	✓	40×40	42 ± 36
No coordinate embedding	✓	✗	✗	40×40	50 ± 12
No $\mathbf{X}$ embedding	✓	✗	✓	40×40	47 ± 22
No camera embedding	✓	✓	✗	40×40	45 ± 10
Wide sampling support	✓	✓	✓	160×160	46 ± 11
VIP-CT	✓	✓	✓	40×40	$\mathbf{36 \pm 25}$

2 Feature sampling

In camera c, $\mathbf{X}$ is projected to a continuous-valued $\mathbf{x}$. A kernel that has spatial support $h \times w$, centered at $\mathbf{x}$, is then defined. This kernel is sampled on a sub-grid having $h_{\mathcal{S}} \times w_{\mathcal{S}}$ samples. Let q index one of these $h_{\mathcal{S}} w_{\mathcal{S}}$ samples. The spatial location of this sample in the image is denoted $\mathbf{x}^{(q)}$.

Recall that, for the c camera image, $\mathcal{P}_c$ is the set of L feature pyramid levels. Using bilinear spatial interpolation at $\mathbf{x}^{(q)}$ on each of the C channels of the L levels (Fig. 3 herein), the discrete feature map pyramid $\mathcal{P}_c$ yields an interpolated feature vector $\mathbf{u}_c^{(q)}$. Then, vectors in these kernel samples are summed by:

$$\mathbf{u}_c(\mathbf{X}) = b(c) + \sum_{q=1}^{h_{\mathcal{S}} w_{\mathcal{S}}} a(q,c)\mathbf{u}_c^{(q)} \ . \tag{1}$$

The parameters $b(c), a(q,c)$ constitute a learnable kernel. We use $h_{\mathcal{S}} = w_{\mathcal{S}} = 9$. Then, the vectors $\{\mathbf{u}_c(\mathbf{X})\}_{c=1}^{N^{\mathrm{cam}}}$ are concatenated across viewpoints to

$$\mathbf{v}(\mathbf{X}) = [\mathbf{u}_1(\mathbf{X}), \mathbf{u}_2(\mathbf{X}), \ldots, \mathbf{u}_{N^{\mathrm{cam}}}(\mathbf{X})] \ . \tag{2}$$

This is Eq. (13) of the main manuscript.

3 Additional Real Data Results

We show in Fig. 4 herein an additional cross-validation view of the cloud in Sec. 5 of the main manuscript. The results are

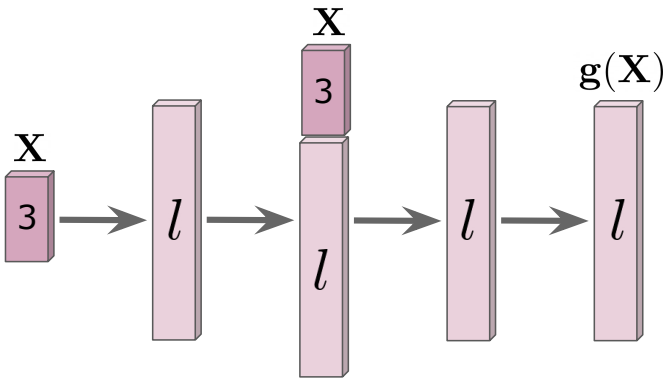


Fig. 1. Visualization of the coordinate encoder architecture. A 3D coordinate $\mathbf{X}$ is embedded to an l dimensional feature vector $\mathbf{g}(\mathbf{X})$. The arrows indicate a fully-connected layer with ReLU activation. The output size of all intermediate layers is l.

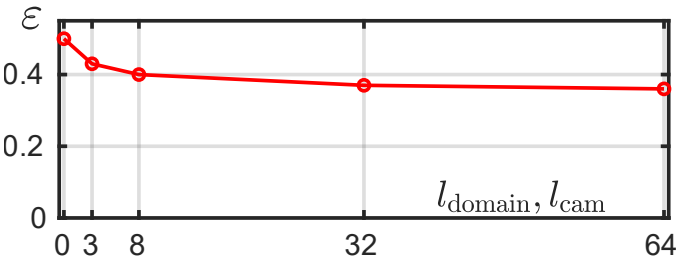


Fig. 2. Ablation study of the geometry encoder feature length. For $l_{\text{domain}} = l_{\text{cam}} = 0$, no geometry information is used during recovery. Here $l_{\text{domain}} = l_{\text{cam}} = 3$ stands for using the raw 3D coordinate values in VIP-CT.

comparable, but the run time of VIP-CT is about 10^4 faster than that of the physics-based solver. In Fig. 5 herein, we show a cross-validation result of a different cloud.

4 Simulation Results

We detail numerical results for simulations described and plotted in the main manuscript. Herein, performance is evaluated by these criteria

$$\epsilon = \frac{\|\boldsymbol{\beta}^{\text{true}} - \hat{\boldsymbol{\beta}}\|_1}{\|\boldsymbol{\beta}^{\text{true}}\|_1} \ , \ \delta = \frac{\|\boldsymbol{\beta}^{\text{true}}\|_1 - \|\hat{\boldsymbol{\beta}}\|_1}{\|\boldsymbol{\beta}^{\text{true}}\|_1} \ , \tag{3}$$

and

$$\gamma = \frac{\|\boldsymbol{\beta}^{\text{true}} - \hat{\boldsymbol{\beta}}\|_2^2}{\|\boldsymbol{\beta}^{\text{true}}\|_2^2} \ . \tag{4}$$

In Fig. 6 herein and Table 2 herein we compare the results obtained by VIP-CT (our approach) and by 3DeepCT [2] for different geometries and datasets.

Then, Figs. 7, 8 and 9 herein visualize additional reconstructions of scenes from the *Subset of Seven Clouds*, discussed in the manuscript and in Sde-Chen et al. [2]. These figures also plot the corresponding scatter plots. Statistics of the *varying geometry* simulation are listed in Table 3 herein. Two failure cases from the BOMEX dataset are presented in Fig. 10 herein.

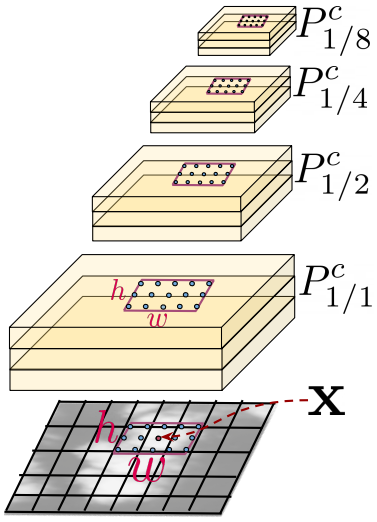


Fig. 3. There are four levels of feature arrays. Each has C channels. The arrays are laterally sampled. First, a kernel that has spatial support $h \times w$, centered at $\mathbf{x}$ is defined. Then, this kernel is sampled on a subgrid having $h_{\mathcal{S}} \times w_{\mathcal{S}}$ samples. In this illustration $h_{\mathcal{S}} = 3$, $w_{\mathcal{S}} = 5$. These samples then undergo weighted summation, using learned weights.

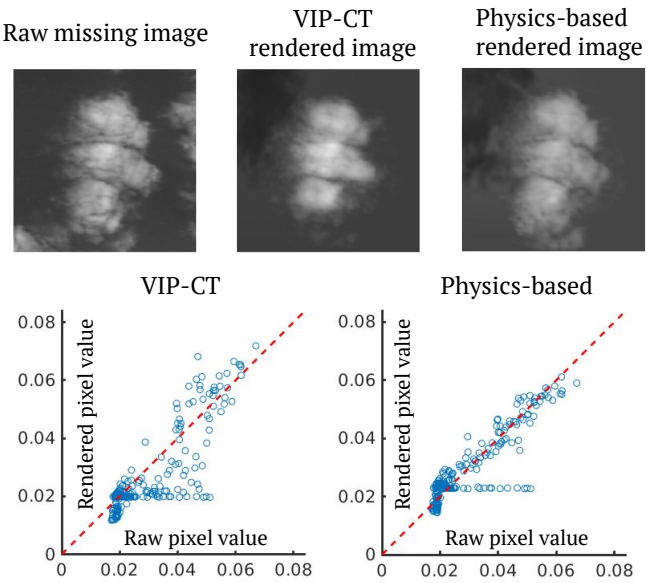


Fig. 4. [Top] Cross-validation appearance results of the cloud in Sec. 5 of the main manuscript, in a different viewpoint. [Bottom] Scatter plots of the pixel values in this view. The correlation coefficient of the scatter plot resulting from our approach (VIP-CT) is 0.85. For physics-based optimization [1], it is 0.9.

References

[1] A. Levis, Y. Y. Schechner, A. Aides, and A. B. Davis, "Airborne three-dimensional cloud tomography," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2015, pp. 3379–3387.

[2] Y. Sde-Chen, Y. Y. Schechner, V. Holodovsky, and E. Eytan, "3DeepCT: Learning volumetric scattering tomography of clouds," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5671–5682.

TABLE 2
Result for fixed geometries. The statistics of ϵ, γ are plotted in Fig. 8 of the main manuscript and Fig. 6 herein, respectively. The row on a *VIP-CT unified* reports a single model that is trained and tested on both BOMEX and CASS datasets.

Dataset	Method	Geometry	Image resolution	Training Time	$\epsilon\%$	$\delta\%$	$\gamma\%$	Test time
BOMEX	3DeepCT [2]	32 satellites	50 m	11 [hours]	82 ± 10	32 ± 16	59 ± 29	7 ± 0.9 millisec
		10 satellites	50 m	8 [hours]	86 ± 10	44 ± 16	67 ± 31	7 ± 0.7 millisec
	VIP-CT	32 satellites	50 m	10 [hours]	48 ± 15	29 ± 25	31 ± 21	0.25 ± 0.25 sec
		10 satellites	50 m	15 [hours]	51 ± 13	13 ± 27	23 ± 24	0.27 ± 0.24 sec
		32 satellites	20 m	14 [hours]	34 ± 24	17 ± 28	13 ± 30	0.67 ± 0.48 sec
		10 satellites	20 m	17 [hours]	36 ± 25	14 ± 29	13 ± 17	0.5 ± 0.37 sec
	VIP-CT unified	10 satellites	20 m	24 [hours]	48 ± 12	29 ± 14	21 ± 13	0.5 ± 0.37 sec
CASS	3DeepCT [2]	10 satellites	50 m	48 [hours]	96 ± 18	3 ± 50	72 ± 33	18 ± 2 millisec
	VIP-CT	10 satellites	50 m	11 [hours]	53 ± 63	-1 ± 47	28 ± 40	0.58 ± 0.47 sec
		10 satellites	20 m	24 [hours]	41 ± 14	25 ± 17	16 ± 17	1 ± 0.6 sec
	VIP-CT unified	10 satellites	20 m	24 [hours]	42 ± 14	18 ± 20	15 ± 25	1 ± 0.6 sec

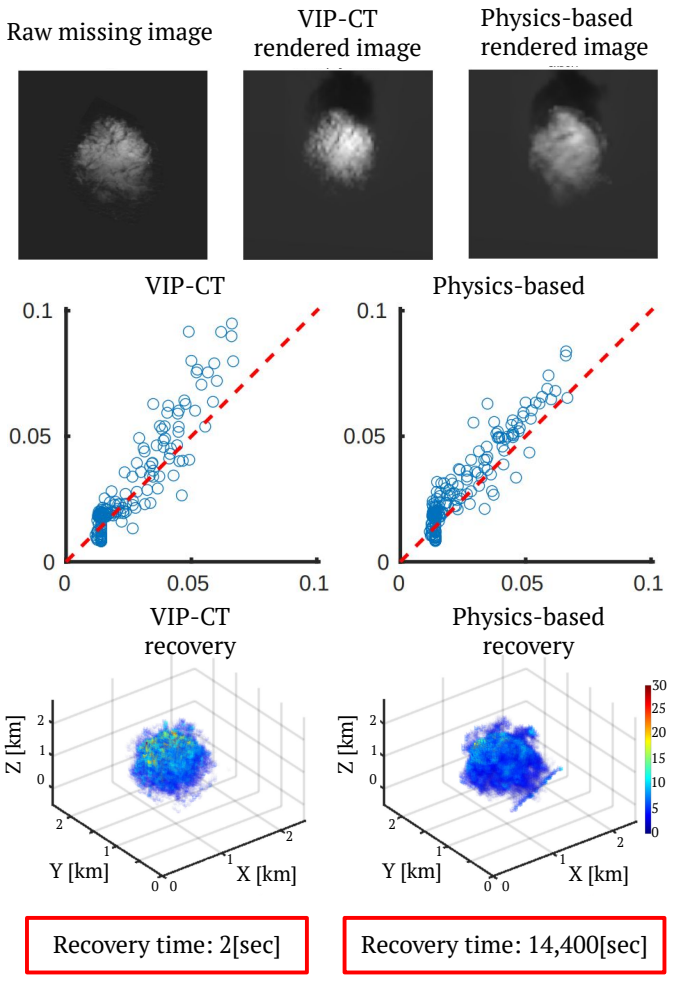


Fig. 5. [Top] An AirMSPI missing view and corresponding rendered views of an additional cloud that is not presented in the main manuscript. The cloud is estimated either by our VIP-CT approach or a physics-based optimizer [1]. Both use data that excludes this view. [Middle] Scatter plots of the pixel values in these views. The correlation coefficient of the scatter plot resulting from our approach (VIP-CT) is 0.93. For physics-based optimization, it is 0.94. [Bottom] Visualization of the recovered cloud by the respective methods.

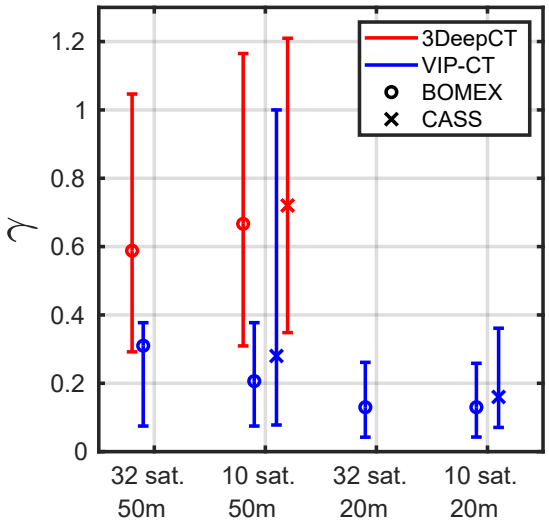


Fig. 6. Results for a fixed geometry, comparing our results (by VIP-CT) to 3DeepCT. Bars represent the 5% and 95% percentiles. In terms of γ, VIP-CT outperforms 3DeepCT [2] across test data-sets and geometries.

TABLE 3
Result for perturbed viewpoints. The perturbation magnitude is detailed in Table 1 of the main manuscript. The statistics of ϵ herein are plotted in Fig. 10 of the main manuscript.

Training geometry	Test perturbation	ϵ	δ
Fixed	-	$36 \pm 25\%$	$14 \pm 29\%$
Perturbed	-	$36 \pm 13\%$	$14 \pm 16\%$
Fixed	S	$36 \pm 22\%$	$15 \pm 26\%$
Perturbed	S	$36 \pm 13\%$	$15 \pm 16\%$
Fixed	M	$38 \pm 21\%$	$16 \pm 25\%$
Perturbed	M	$36 \pm 13\%$	$15 \pm 16\%$
Fixed	L	$41 \pm 20\%$	$16 \pm 24\%$
Perturbed	L	$37 \pm 12\%$	$16 \pm 15\%$
Fixed	XL	$42 \pm 18\%$	$16 \pm 22\%$
Perturbed	XL	$37 \pm 11\%$	$17 \pm 14\%$

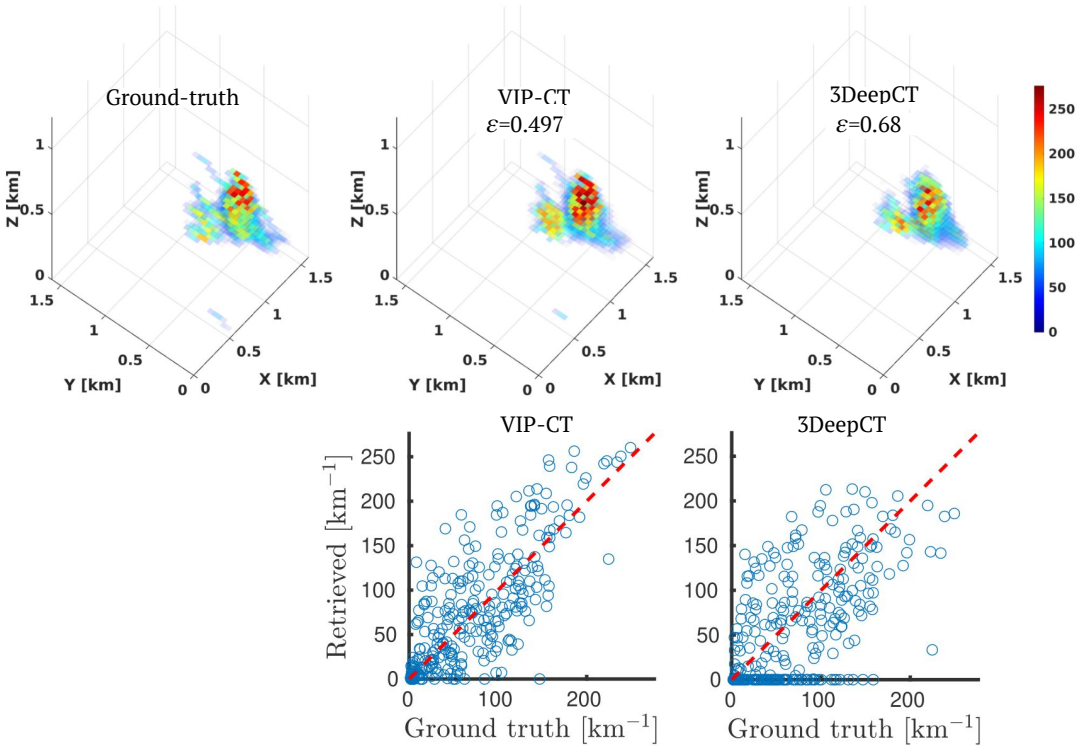


Fig. 7. 3D reconstructions of the extinction coefficient; corresponding values of ϵ; and scatter plots of $\hat{\beta}$. These results correspond to an example scene, out of the *Subset of Seven Clouds* mentioned in the main manuscript.

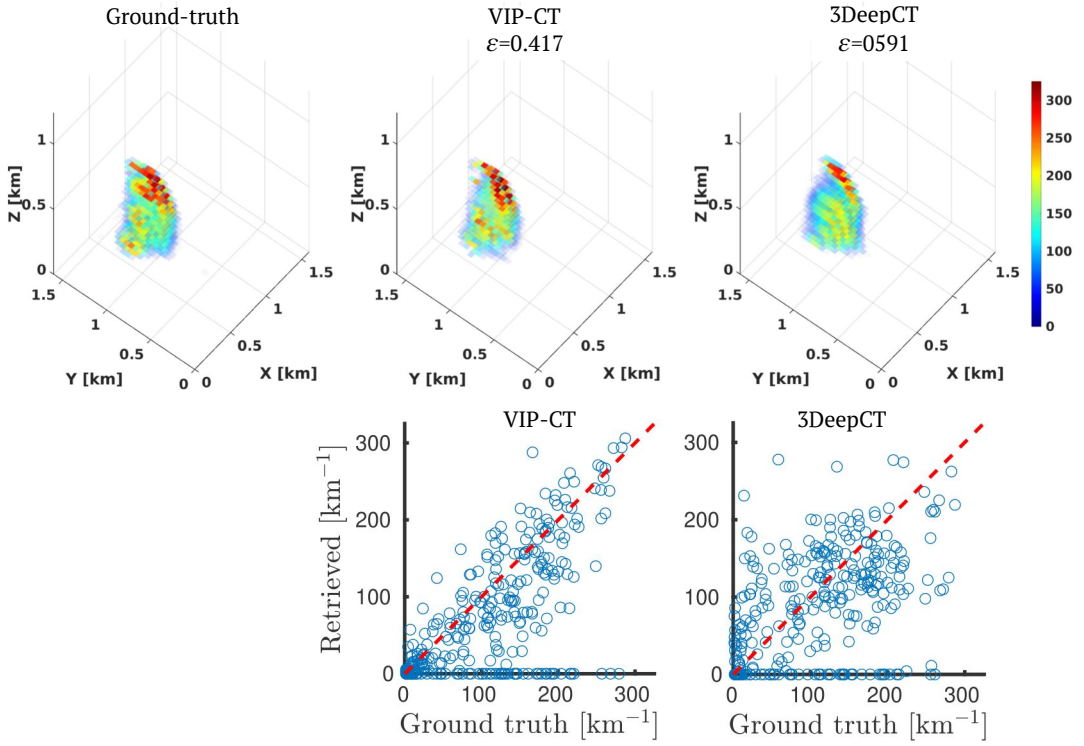


Fig. 8. 3D reconstructions of the extinction coefficient; corresponding values of ϵ; and scatter plots of $\hat{\beta}$. These results correspond to an example scene, out of the *Subset of Seven Clouds* mentioned in the main manuscript.

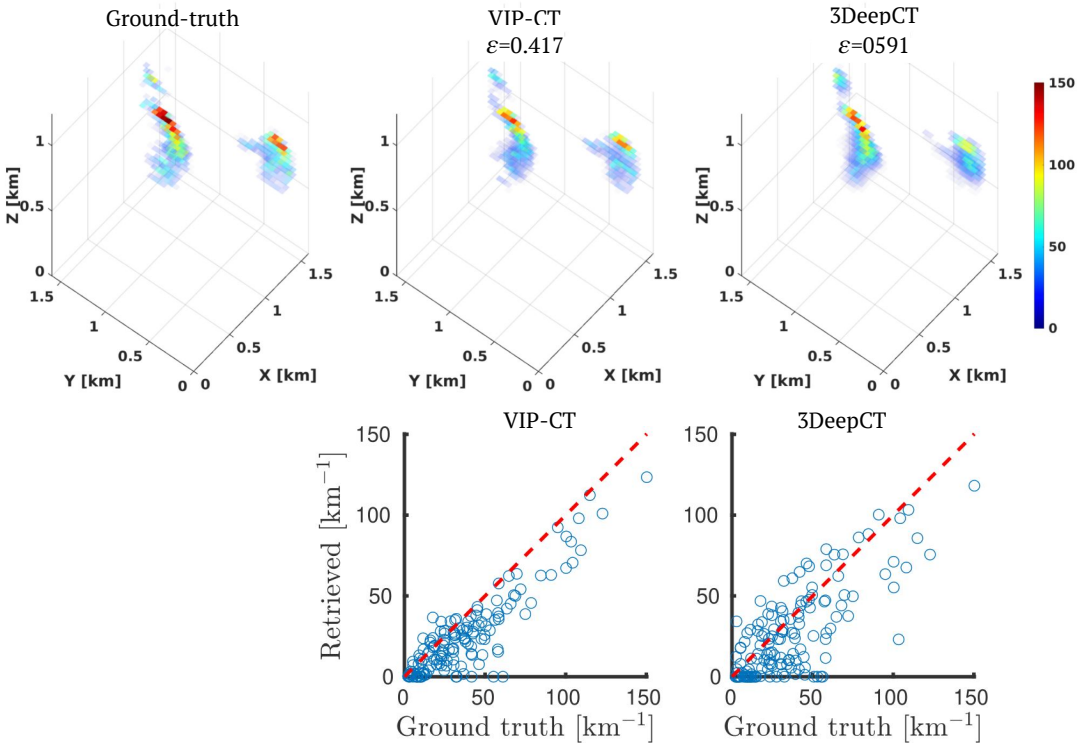


Fig. 9. 3D reconstructions of the extinction coefficient; corresponding values of ϵ; and scatter plots of $\hat{\beta}$. These results correspond to an example scene, out of the *Subset of Seven Clouds* mentioned in the main manuscript.

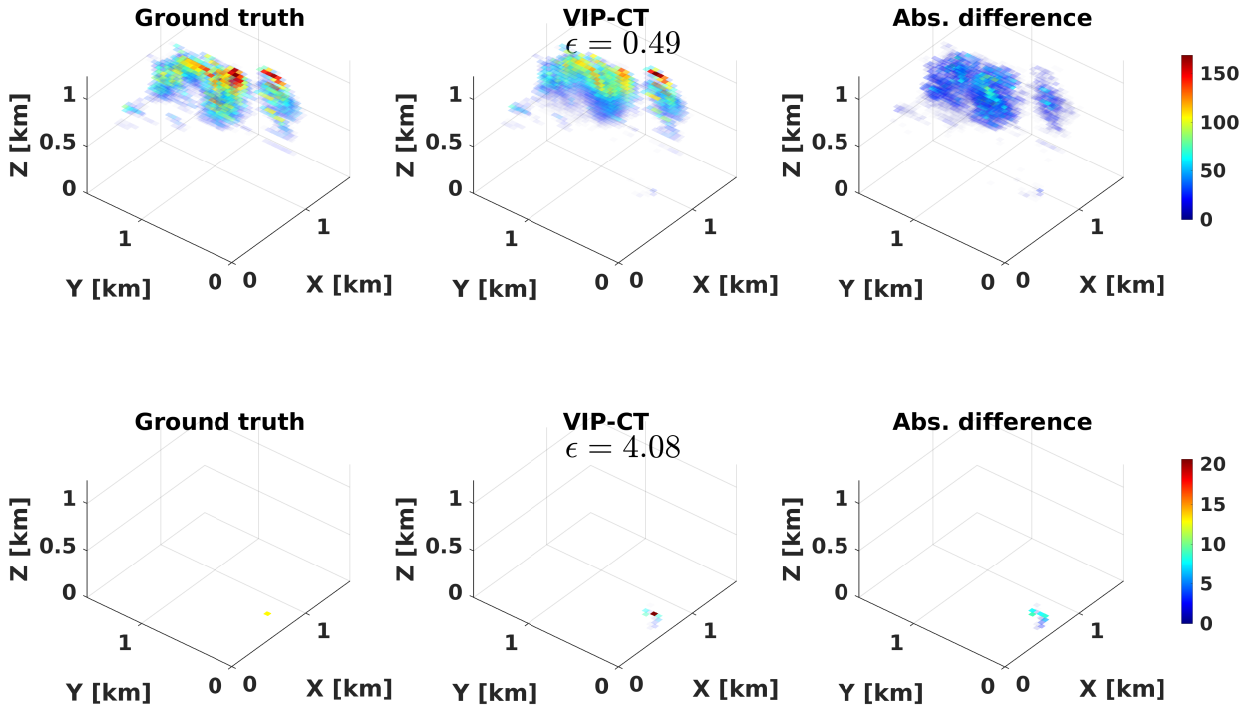


Fig. 10. Two cloud examples, from the BOMEX dataset, poorly recovered by VIP-CT. For clarity, we show the absolute difference per voxel of the estimated and true clouds. A very high ϵ value is obtained for clouds with a very small number of voxels.