Flexible algebraic technique for multiview reconstruction: incremental learning in reflective tomography

Jean-Baptiste Bellet

▶ **To cite this version:**

Jean-Baptiste Bellet. Flexible algebraic technique for multiview reconstruction: incremental learning in reflective tomography. Optical Engineering, 2019, 58 (10), pp.103102. 10.1117/1.OE.58.10.103102 . hal-02316413

HAL Id: hal-02316413
https://hal.science/hal-02316413

Submitted on 5 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Jean-Baptiste Bellet, “Flexible algebraic technique for multiview reconstruction: incremental learning in reflective tomography,” Opt. Eng. 58(10), 103102 (2019), doi: 10.1117/1.OE.58.10.103102.

Copyright 2019 Society of Photo-Optical Instrumentation Engineers (SPIE). One print or electronic copy may be made for personal use only. Systematic reproduction and distribution, duplication of any material in this publication for a fee or for commercial purposes, and modification of the contents of the publication are prohibited.

Flexible algebraic technique for multi-view reconstruction: incremental learning in reflective tomography

Jean-Baptiste Bellet[a,*]
[a]Université de Lorraine, CNRS, IECL, F-57000 Metz, France

Abstract. Reflective tomography reconstructs a scene from calibrated reflective images, using algorithms from X-ray tomography. Many works in the subject are based on analytical formulas such as the filtered backprojection. However these formulas require constraints on the acquisition geometry, such as a circular rotation. We want to avoid such constraints: they may be seriously violated in some practical cases. To tackle this problem, we tune the Algebraic Reconstruction Technique from X-ray tomography. More precisely we look for a model of the scene such that the X-ray projections of the model approximate recorded calibrated reflective images. The model is computed by an iterative algebraic method: a Kaczmarz algorithm. In this way we perform incremental supervised learning in optics, where the hypothesis space emulates reflective tomography. We get a flexible method for multiple-view reconstruction based on linear algebra. It accepts a general calibrated acquisition such as: several cameras arbitrarily located/oriented, with visible-near infrared wavelengths. It could reconstruct a scene using several devices simultaneously, such as air-ground cameras combined with ground-ground cameras. The relevance of the approach is numerically shown, from calibrated CCD images of the Middlebury datasets. In particular we get reconstructions from 16 views.

Keywords: three-dimensional imaging, optical computational imaging, machine learning in optics, reflective tomography, algebraic reconstruction technique.

* Jean-Baptiste Bellet jean-baptiste.bellet@univ-lorraine.fr

1 Introduction

1.1 Reflective tomography

Reflective tomography emerged at the end of the 80's.[1,2] The initial method computes a tomographic reconstruction from reflective projections obtained with laser radars, such as range-resolved data. The reconstruction solver operates an X-ray inversion, despite the wavelengths are much larger than the X-ray wavelengths. This heuristic approach takes benefit from geometric similarities between X-ray projections and reflective projections: it is linked with geometric tomography.[3] The method is successful for several kinds of reflective data and has been introduced in various frameworks. The same principle has been proposed in object modeling from photographs.[4] The method has been tested for imaging satellites.[5,6] More recently, it has been shown[7–9] that reflective tomography achieves three-dimensional optical imaging from bi-dimensional images of

backscattered intensities in the visible or near-infrared band. In particular, the method overcomes occlusion issues and enables visualization of concealed objects.[10]

1.2 Need for flexibility

Most of the works in reflective tomography invert the Radon transform, or the X-ray transform, by the means of analytical formulas such as the filtered backprojection in 2D, or the Feldkamp-Davis-Kress (FDK) algorithm for the cone-beam scan in 3D. Nevertheless, we can imagine practical cases where these analytical formulas cannot be directly applied.

For example, the ideal acquisition for the FDK algorithm would require: a camera with fixed intrinsic parameters, that moves on a circular trajectory with a constant angular step, and that points towards the center of the trajectory. These constraints may be violated in practice. Hence correcting algorithms have been designed to relax the constraints:[7] they re-calibrate the images. These methods need less stringent constraints, such as a trajectory contained in a plane.

But more generally the recorded images may come from several acquisition devices, located at arbitrarily positions and oriented along arbitrarily directions. We would like a flexible imaging method based on reflective tomography that directly tolerates such a general scenario. This would overcome some geometric limitations[11] of usual reflective tomography and would extend its possibilities.

1.3 Proposed strategy

Two classes of methods can be distinguished in X-ray tomography: the analytical methods (as above), and the algebraic methods. The algebraic methods consider the problem of X-ray tomography as a linear system, or as an optimization problem. Then this problem is solved by an iterative

method. The most widely spread one is the Kaczmarz algorithm (and its variants), known in the field of tomography as the Algebraic Reconstruction Technique (ART).[12,13] This method is very flexible, since it is based on linear algebra considerations, and not on special formulas. In reflective tomography, some algebraic methods have already been tested in 2D: the gradient method and the conjugate gradient method.[14]

In this paper we introduce ART in 3D reflective tomography, in order to get a flexible algebraic technique for 3D multiple-view reconstruction. We derive a frame-driven Kaczmarz method. Basically the method tries to model the recorded images (the frames) as X-ray projections of a 3D model of the scene. The iterative method is a loop over the frames. Each iteration updates the 3D model, using the constraint that the X-ray transform of the model should reproduce the selected frame. Using the machine learning terminology,[15] the method realizes online learning[16] in optics.

Concerning the validation of the approach, we test numerically the method on calibrated images captured with a CCD camera, extracted from the famous datasets of Middlebury.[17,18] We show several scenarios, including cases where the images are sampled on a hemisphere, and cases where the number of available images is very limited. We estimate the quality of the computed models by cross-validation.

1.4 Organization

The paper is organized as follows. First, we describe X-ray projections in the pinhole geometry of an ideal visible-near infrared camera. Then, we propose a Kaczmarz algorithm for the multi-view reconstruction in the framework of reflective tomography; we discuss the set of parameters of the method and we estimate the computational costs. Finally, we show numerically the relevance of the method on various examples from the Middlebury datasets.

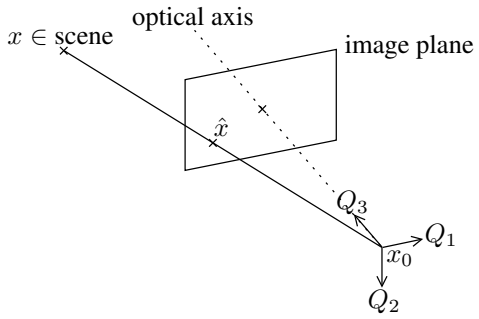


Fig 1 Perspective projection through an ideal camera.

2 Perspective projection

2.1 Geometric model of image formation

In the visible-near infrared domain (VIS-NIR), an ideal camera is a pinhole,[19,20] modeled as Figure 1. In a world reference frame, the camera is located at the optical center $x_0 \in \mathbb{R}^3$, while the orientation of the camera is represented by an orthogonal matrix $Q = (Q_1, Q_2, Q_3) \in \mathbb{R}^{3\times 3}$ such that: the unit vector Q_1 is the horizontal direction in the image plane, the unit vector Q_2 is orthogonal to Q_1 and is the vertical direction in the image plane, while the vector $Q_3 = Q_1 \times Q_2$ is aligned with the optical axis, and points towards the scene. From a geometric point of view, the pinhole realizes an ideal perspective projection through the optical center x_0: a point $x \in \mathbb{R}^3$ is projected onto a point $\hat{x}$ that belongs to the image plane and such that $x, \hat{x}$, and x_0 are aligned.

We assume that in the image plane, a pixel is a parallelogram with a horizontal side (parallel to Q_1) and we introduce an image frame based on the horizontal pixel coordinate of the image ($1 \leq i_1 \leq m_1$) and the vertical pixel coordinate ($1 \leq i_2 \leq m_2$); see Figure 2. Then it is known[19]

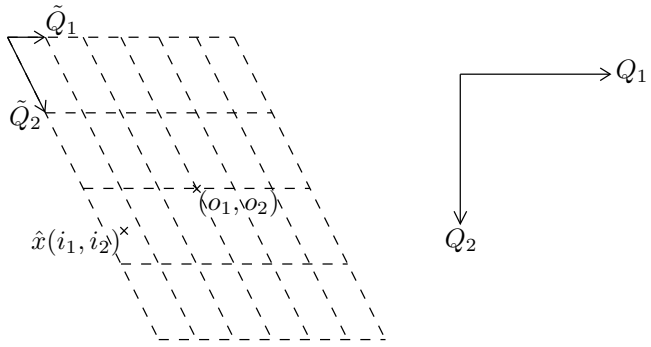


Fig 2 Image plane of Figure 1. The sides $(\tilde{Q}_1, \tilde{Q}_2)$ of a pixel, combined with an origin such as the top left corner define pixel coordinates. In pixel coordinates, the optical axis is projected onto (o_1, o_2); and $\hat{x}$ has coordinates (i_1, i_2). The parameters of the calibration matrix K are such that $Q_1 = s_1\tilde{Q}_1$ and $Q_2 = s_{12}\tilde{Q}_1 + s_2\tilde{Q}_2$.

that the coordinates (i_1, i_2) of $\hat{x}$ satisfy a relationship of the form:

$$\lambda \begin{bmatrix} i_1 \\ i_2 \\ 1 \end{bmatrix} = K[Q^*, -Q^* x_0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix}, \quad \text{with } \lambda \geq 0. \tag{1}$$

Here, $\cdot^*$ denotes the transpose. The triplet (x_1, x_2, x_3) denotes the coordinates of x in the world frame. The parameter $\lambda = Q_3^*(x - x_0)$ represents the depth of x in the camera frame (x_0, Q). The extrinsic matrix of the camera is $[Q^*, -Q^* x_0]$; it depends only on the position x_0 and on the orientation Q of the camera. The upper triangular matrix

$$K = \begin{bmatrix} fs_1 & fs_{12} & o_1 \\ 0 & fs_2 & o_2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_1 & s_{12} & o_1 \\ 0 & s_2 & o_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3\times 3} \tag{2}$$

is the intrinsic calibration matrix of the camera. The focal length $f > 0$ is the distance between the optical center x_0 and the image plane $x_0 + fQ_3 + \mathrm{span}(Q_1, Q_2)$. The optical axis $\{x_0 + \lambda Q_3, \lambda \geq 0\}$ is projected onto the pixel whose coordinates are (o_1, o_2) (in pixel coordinates). The parameters s_1, s_2, s_{12} encode the shape of a pixel. For square pixels, $s_{12} = 0$, and $s_1 = s_2$ is the inverse of a side length. More generally the sides of a pixel are the vectors $\tilde{Q}_1 = \frac{1}{s_1} Q_1$ and $\tilde{Q}_2 = \frac{1}{s_2} Q_2 - \frac{s_{12}}{s_1 s_2} Q_1$. So the size of unit length in horizontal, respectively vertical, pixels is s_1, resp. s_2, and s_{12} represents the skew of a pixel. The intrinsic matrix K depends on the focal length and on the shape of a pixel on the receiver array: it depends only the camera, and not on the position, nor on the orientation.

In a word, an ideal camera is associated with a quadruplet $C = (x_0, Q, K, m)$, where x_0 is the position, Q is the orientation, K is the calibration matrix, and $m = (m_1, m_2)$ is the size of the image. From a geometric point of view, the projection is governed by the relation (1). The main assumption of this paper is that we consider only such cameras, and whose parameters (x_0, Q, K, m) are known, or at least pre-computed. The method that we will derive does not aim at computing the parameters (x_0, Q, K, m). In particular, the computation (or measurement) of the camera parameters is a preliminary step to our method; we refer to methods of vision[19] for that step.

2.2 X-ray transform

The X-ray transform $\mathcal{X}$ integrates along lines.[12] Let $\phi : \mathbb{R}^3 \to \mathbb{R}$ be a scalar function defined on the space $\mathbb{R}^3$. Let $L(x_0, u) = \{x_0 + \lambda u, \lambda \geq 0\}$ be the ray of origin $x_0 \in \mathbb{R}^3$ and whose direction is the unit vector $u \in \mathbb{S}^2$. The X-ray transform of ϕ, along the ray $L(x_0, u)$, is the integral (if defined)

$$\mathcal{X}[\phi](x_0, u) = \int_0^\infty \phi(x_0 + \lambda u) \mathrm{d}\lambda. \tag{3}$$

The unit of $\mathcal{X}[\phi](x_0, u)$ is the unit of ϕ multiplied by a length (λ is a length). In X-ray tomography, the Beer-Lambert law establishes that this transform models the attenuation of X-rays along lines: if an X-ray emanates from x_0 in the direction u, with intensity I_0, then at infinity, on the ray $L(x_0, u)$, the intensity I is modeled by $\ln \frac{I_0}{I} = \mathcal{X}[\phi](x_0, u)$. In that case, ϕ is an inverse length that represents a linear attenuation coefficient of the crossed materials, while the global attenuation $\mathcal{X}[\phi](x_0, u)$ is dimensionless.

We recall a sufficient theoretical geometric condition[12] under which the X-ray transform (3) can be inverted. Assuming that x_0 scans a curve γ, if the Tuy's condition is satisfied: the curve γ intersects each plane hitting supp ϕ transversely, then the function ϕ can be reconstructed from the $\mathcal{X}[\phi](x_0, u), x_0 \in \gamma, u \in \mathbb{S}^2$.

In practice the function ϕ belongs to a finite dimensional vector space for computational purposes:[13] $\phi = \sum_{1 \leq k \leq N} \varphi_k \psi_k$, where $\{\psi_k\}_{1 \leq k \leq N}$ is a basis of the functional space and $\varphi = (\varphi_k)_{1 \leq k \leq N} \in \mathbb{R}^N$. The coefficients φ_k of the linear combination have the same unit than ϕ and the basis functions ψ_k are dimensionless. In this paper, we consider functions ϕ that are defined on a finite grid of voxels. Roughly speaking, the function ϕ is constant on small cubes (voxels). The number of voxels is N, φ_k is the value of ϕ inside the voxel numbered k, while the function ψ_k represents the geometry of this voxel. More precisely, we discretize a parallelepiped $[a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$ on a uniform grid of $N = n_1 \times n_2 \times n_3$ voxels of side h:

$$\{(x_1, x_2, x_3) : (k_1 - 1)h \leq x_1 - a_1 \leq k_1 h, (k_2 - 1)h \leq x_2 - a_2 \leq k_2 h, (k_3 - 1)h \leq x_3 - a_3 \leq k_3 h\}; \tag{4}$$

$1 \leq k_1 \leq n_1$, $1 \leq k_2 \leq n_2$ and $1 \leq k_3 \leq n_3$ are the voxel coordinates. The basis function ψ_k is the characteristic function of the voxel $k \equiv (k_1, k_2, k_3)$: it takes the value 1 inside the cube (4) and

on the three "upper" faces $x_i = a_i + k_i h$; its value is 0 otherwise.

Using the basis functions, the X-ray evaluation (3) becomes:

$$\mathcal{X}[\phi](x_0, u) = \sum_{k=1}^{N} \varphi_k \mathcal{X}[\psi_k](x_0, u); \tag{5}$$

here $\mathcal{X}[\psi_k](x_0, u)$ is the intersection length of the ray $L(x_0, u)$ with the voxel numbered k. Only a few voxels really contribute: most of the intersections are void. The projection $\mathcal{X}[\phi](x_0, u)$ is computed by an efficient scheme.[21,22] The contributing voxels are identified by ray tracing: the volume is crossed along the ray, from voxel to voxel. For each voxel encountered along the ray, its contribution $\varphi_k \mathcal{X}[\psi_k](x_0, u)$ is computed and added to the sum. At the end, we get (5) in $O(\|n\|)$ operations, where the number of diagonal voxels $\|n\| = \sqrt{n_1^2 + n_2^2 + n_3^2}$ is an upper bound over the number of crossed voxels.

We define now an X-ray image of ϕ for the same cone-beam geometry than the pinhole geometry. We set the perspective rays by the means of a quadruplet $C = (x_0, Q, K, m)$ as before. Each pixel (i_1, i_2) of the X-ray image has the intensity $\mathcal{X}[\phi](x_0, u)$, where the ray $L(x_0, u)$ passes through (i_1, i_2) and x_0:

$$L(x_0, u) = \{(x_1, x_2, x_3) : (1), \lambda \geq 0\}, \quad \text{with } u = \frac{\tilde{u}}{\|\tilde{u}\|}, \tilde{u} = QK^{-1} \begin{bmatrix} i_1 \\ i_2 \\ 1 \end{bmatrix}. \tag{6}$$

We write the X-ray image of ϕ, for the geometry $C = (x_0, Q, K, m)$, as a matrix product. We number the pixel coordinates and the corresponding directions u by $1 \leq i \leq M = m_1 \cdot m_2$. Then

the X-ray image of ϕ, associated with C, is given by the vector:

$$X_C\varphi, \quad \text{with} \quad X_C = [\mathcal{X}[\psi_k](x_0, u_i)]_{\substack{1\le i\le M \\ 1\le k\le N}}, \quad \varphi = [\varphi_k]_{1\le k\le N}. \tag{7}$$

The vector $\varphi \in \mathbb{R}^N$ contains the coordinates of ϕ in the basis $\{\psi_k\}$. The matrix $X_C \in \mathbb{R}^{M\times N}$ (unit: length) contains the X-ray transforms of the basis functions ψ_k, along the rays defined by $C = (x_0, Q, K, m)$; as soon as the set of basis functions $\{\psi_k\}_{1\le k\le N}$ is fixed, this matrix X_C depends only on C. The vector $X_C\varphi \in \mathbb{R}^M$ of (7) represents in a convenient way the X-ray image of ϕ for the configuration C. The matrix X_C is sparse but huge; so in practice X_C is not directly computed: $X_C\varphi$ is rather computed by ray tracing, in $O(\|n\|M)$ operations.

We can now define the backprojection associated with the projection X_C: it is the transpose X_C^*. The backprojection of an image $[g_i]_{1\le i\le M}$ is also computed by ray-tracing in $O(\|n\|M)$ operations, due to the expression:

$$X_C^*[g_i]_{1\le i\le M} = \left[\sum_{i=1}^{M} g_i \mathcal{X}[\psi_k](x_0, u_i)\right]_{1\le k\le N}. \tag{8}$$

2.3 Maximum Intensity Projection

We consider a function defined on voxels as before: $\phi = \sum_k \varphi_k\psi_k$. We have presented so far the projection of this function according to the X-ray transform. We recall now the principle of the Maximum Intensity Projection (MIP): it is a volume rendering method that we will use along perspective rays.

The geometry of a MIP camera is defined by $C = (x_0, Q, K, m)$ as before. In the image plane, each pixel (i_1, i_2) records the maximum of the function ϕ, along the ray defined by (6).

Furthermore the rendering can be adjusted by thresholding; in this paper, we will use 0 as a lower threshold. Using the same notations $1 \leq i \leq M$ and u_i as in (7), the MIP image of $\phi = \sum_k \varphi_k \psi_k$, associated with C is:

$$\Pi_C \varphi = \left[\max \left(0, \max_{L(x_0, u_i)} \phi \right) \right]_{1 \leq i \leq M}. \tag{9}$$

The coefficients of $\Pi_C \varphi$ define pixel values; these values have the same unit than ϕ. They can be efficiently computed using ray tracing as for the X-ray transform. The main difference is that the sum is replaced by the maximum. The cost is again $O(M\|n\|)$ operations.

3 Flexible reflective tomography

3.1 Multiple view geometry

We capture several images of a fixed scene, using VIS-NIR cameras. Several scenarios are possible: the same camera is used for several locations, several orientations, several focal lengths, with a motion which is continuous or not. And/or several cameras are employed. For multi-channel images, one can consider for example that each channel provides one image.

The collection of the recorded images is denoted by $(g_s)_{1 \leq s \leq S}$, where S is the number of images, and g_s is the image number s. We assume that for all $1 \leq s \leq S$, the image g_s is modeled by an ideal camera associated with a quadruplet $C_s = (x_{0s}, Q_s, K_s, m_s)$ as described in the previous section: x_{0s} is the location of the optical center, Q_s is the orientation, K_s is the calibration matrix, and $m_s = (m_{s,1}, m_{s,2})$ represents the size. As for the X-ray images, we assume that g_s is a column vector in $\mathbb{R}^{M_s}$, based on a numbering of the pixels $1 \leq i \leq M_s = m_{s,1} \cdot m_{s,2}$. We denote by $M = \sum_{s=1}^{S} M_s$ the total number of records (*i.e.* the number of rays of projection). We assume that the parameters C_s are known or pre-computed by another method. Also this model

assumes that the eventual distortions have already been corrected.

A classical problem in three-dimensional vision: how can we reconstruct the geometry of the original scene from the recorded calibrated views $(g_s, C_s)_{1\leq s\leq S}$?

3.2 Reflective tomography

Reflective tomography gives a heuristic answer to this question. If the images g_s are collected in the X-ray spectrum, then we look for an attenuation $\phi = \sum_{1\leq k\leq N} \varphi_k \psi_k$ such that

$$X_{C_s}\varphi = g_s, \quad 1 \leq s \leq S. \tag{10}$$

Reflective tomography proposes to solve the same system of linear equations (10), despite the images g_s are measured using VIS-NIR cameras. Concerning the units, the matrices X_{C_s} contain lengths. In the case of X-ray tomography, the vectors g_s contains logarithms of intensities ratios and are dimensionless, and the attenuation coefficients of φ are inverse lengths. For reflective tomography, the situation is different. The unit of g is indeed the unit of the recorded images g_s: assuming that the records are irradiances, g contains powers per unit area (W.m^{-2}). So the sought function $\phi = \sum_{1\leq k\leq N} \varphi_k \psi_k$ and its coefficients φ_k represent powers per unit volume (irradiance divided by length).

For special occurrences of the parameters C_s, the system (10) is often solved using analytical formula based on X-ray inversion, such as the filtered backprojection or the Feldkamp-Davis-Kress algorithm. In this paper we focus rather on algebraic methods because they enable general configurations for the C_s. We compute a voxel model $\phi = \sum_k \varphi_k \psi_k$ of the scene by ART. Then we synthesize[4,23] new images of the scene based on the MIP (9): this volume rendering method

is efficient in reflective tomography, because the reconstruction ϕ is essentially supported by the surfaces of the scene (up to artifacts).

3.3 Algebraic reconstruction technique

The equations (10) define blocks of equations for the linear system:

$$X\varphi = g, \quad \text{with} \quad X = [X_{C_s}]_{1\leq s\leq S}\,, \quad g = [g_s]_{1\leq s\leq S}\,. \tag{11}$$

The matrix $X \in \mathbb{R}^{M\times N}$ contains the X-ray projections of the basis functions along every ray of projection. The right hand-side $g \in \mathbb{R}^M$ is a column vector containing all the records. The matrix X may be rank deficient and the right-hand side g may be outside the range of X. So we should consider instead a least squares solution such as the Moore-Penrose generalized inverse: the element $\varphi \in \mathbb{R}^N$ with the smallest norm in the set of minimizers of $\|X\varphi - g\|$. It is the unique solution to the normal equation $X^*X\varphi = X^*g$ in the range of X^*. But we do not solve directly this huge problem of N unknowns and M equations. We propose instead an iterative algebraic method: a Kaczmarz algorithm, inspired by the Algebraic Reconstruction Technique (ART) of tomography.

We propose a frame-driven version of ART. The principle: looping over the frames in order to incorporate into the reconstruction φ the linear constraints (10), frame by frame. We start with $\varphi^{(0)} = 0$ (or another initial guess if available). Let us assume that the k-th iterate $\varphi^{(k)}$ has been computed. We select the frame g_{s_k}, were the number s_k satisfies $1 \leq s_k \leq S$. The next iterate $\varphi^{(k+1)}$ is defined as follows:

$$\varphi^{(k+1)} = \varphi^{(k)} + \omega X^*_{C_{s_k}} \left(X_{C_{s_k}} X^*_{C_{s_k}} + \sigma \mathrm{I} \right)^{-1} (g_{s_k} - X_{C_{s_k}} \varphi^{(k)}), \tag{12}$$

with $0 < \omega < 2$ (dimensionless), and $\sigma > 0$ (unit: area). We comment this definition below.

Concerning the numbers s_k, we propose to consider each frame once per cycle of S iterations: $\{s_k\}_{j \le k \le j+S-1} = \{1, \dots, S\}$. Then after κ cycles, *i.e.* κS iterations, the constraint of each frame has been used κ times. Other strategies could be possible to weight the contribution of the images.

For the recurrence relation (12), let us consider the case $\omega = 1$. If the matrix $X_{C_{s_k}} X_{C_{s_k}}^* \in \mathbb{R}^{M_s \times M_s}$ was invertible and $\sigma = 0$, then[12] the relation (12) would define $\varphi^{(k+1)}$ as the minimizer of $\left\|\varphi - \varphi^{(k)}\right\|$ with the constraint $X_{C_{s_k}} \varphi = g_{s_k}$; in other words this would model the selected image g_{s_k} as an X-ray image, where the power per unit volume φ would be chosen as close as possible to the current estimate $\varphi^{(k)}$. But $X_{C_{s_k}} X_{C_{s_k}}^*$ may be rank deficient, so a Tikhonov regularization[24] is performed with parameter $\sigma > 0$: (12) defines $\varphi^{(k+1)}$ as the minimizer of $\|X_{C_k} \varphi - g_{s_k}\|^2 + \sigma \left\|\varphi - \varphi^{(k)}\right\|^2$, without constraint. Roughly speaking, this minimization tries to find φ such that $X_{C_k} \varphi \approx g_{s_k}$ and $\varphi \approx \varphi^{(k)}$; the parameter σ controls the relative importance of these two conditions. More generally, with a relaxation parameter $0 < \omega < 2$, the relation (12) combines the Tikhonov solution with the estimate $\varphi^{(k)}$, with weights ω and $1 - \omega$.

The vector $\left(X_{C_{s_k}} X_{C_{s_k}}^* + \sigma \mathrm{I}\right)^{-1} (g_{s_k} - X_{C_{s_k}} \varphi^{(k)})$ is computed by solving a linear system:

$$\left(X_{C_{s_k}} X_{C_{s_k}}^* + \sigma \mathrm{I}\right) v = g_{s_k} - X_{C_{s_k}} \varphi^{(k)}. \tag{13}$$

The matrix of the regularized system (13) is symmetric positive definite due do $\sigma > 0$; this enables a safe inversion. We solve this system using another iterative method: the conjugate gradient algorithm.[25] One iteration of this method costs about one evaluation of the matrix against a vector;

so during the procedure, we take advantage of the relation

$$(X_{C_{s_k}} X^*_{C_{s_k}} + \sigma \mathrm{I})v = X_{C_{s_k}}(X^*_{C_{s_k}} v) + \sigma v. \tag{14}$$

It enables efficient computations by ray-tracing; the matrices themselves are never computed.

To finish with, after κ cycles of iterations, *i.e.* κ scans of the full dataset, a root mean square error RMSE (power per unit area, as the record g) can be computed if desired:

$$\eta^{(\kappa)} = \sqrt{\frac{1}{M} \sum_{s=1}^{S} \| X_{C_s} \varphi^{(\kappa S)} - g_s \|^2}. \tag{15}$$

The RMSE provides some indicator about the convergence of the process; we can monitor the decay rate $\tau^{(\kappa+1)} = \frac{\eta^{(\kappa)} - \eta^{(\kappa+1)}}{\eta^{(\kappa)}}$ and decide that convergence has been reached as soon as the decay rate is below a fixed threshold $0 < \tau < 1$. Furthermore we can normalize the RMSE, by comparison with the standard deviation $\hat{\sigma}_g$ of the dataset g: we get in this way the root relative squared error RRSE (dimensionless)

$$\rho^{(\kappa)} = \frac{\eta^{(\kappa)}}{\hat{\sigma}_g}. \tag{16}$$

3.4 Online learning

We construe the ART (12) as machine learning[15] in optics. The dataset $(C_s, g_s), 1 \leq s \leq S$, contains labeled training data. The camera parameters C_s are considered as the observations, while the VIS-NIR images g_s are considered as their labels. Ideally we would like to infer a function F such that for all configuration C, $F(C)$ is a VIS-NIR image of the original scene, taken with

a camera associated with the parameters of C. We design a supervised learning algorithm that analyses the training dataset: we try to find F such that $F(C_s) \approx g_s, 1 \leq s \leq S$. Of course this problem is very ill-posed.

Reflective tomography takes benefit from the geometrical similarities between the perspective projections of VIS-NIR cameras and of X-ray images, and adds a strong hypothesis about the unknown F: it assumes that we can find a reasonable F under the form $F(C) = X_C\varphi$, where $\varphi \in \mathbb{R}^N$ defines a reasonable voxel model of the scene $\phi = \sum_{k=1}^{N} \varphi_k \psi_k$. We design a hypothesis space based on this principle: we look for $F \in \{F_\varphi : C \mapsto X_C\varphi, \varphi \in \mathbb{R}^N\}$.

Then we define a cost function J that measures the modeling error for the occurrence (C, g), when φ defines the model (and thus $F = F_\varphi$):

$$J(\varphi, C, g) = \frac{1}{2} \left\| \left(X_C X_C^* + \sigma \mathrm{I}\right)^{-\frac{1}{2}} \left(X_C\varphi - g\right) \right\|^2, \quad \text{with} \quad \sigma > 0. \tag{17}$$

J represents the sum of squared residuals for the linear system $X_C\varphi = g$, with preconditioner $(X_C X_C^* + \sigma \mathrm{I})^{\frac{1}{2}}$. Its gradient with respect to φ is: $\frac{\partial J}{\partial \varphi}(\varphi, C, g) = X_C^* \left(X_C X_C^* + \sigma \mathrm{I}\right)^{-1} \left(X_C\varphi - g\right)$.

We define now an online learning based on this cost. At the beginning, we set $\varphi^{(0)} = 0$ (for example). Let us assume that the k-th model $\varphi^{(k)}$ has been computed. We select a new frame g_{s_k} with $1 \leq s_k \leq S$. The modeling error for this frame is $J(\varphi^{(k)}, C_{s_k}, g_{s_k})$. We would like to find a new model $\varphi^{(k+1)}$ such that this modeling error decreases. So we compute $\varphi^{(k+1)}$ as the first iterate of the gradient method for the preconditioned least squares problem $\inf_{\varphi \in \mathbb{R}^N} J(\varphi, C_{s_k}, g_{s_k})$, with $\varphi^{(k)}$ as the starting point and $\omega > 0$ as the step. We get:

$$\varphi^{(k+1)} = \varphi^{(k)} - \omega \frac{\partial J}{\partial \varphi}(\varphi^{(k)}, C_{s_k}, g_{s_k}). \tag{18}$$

We recognize exactly the relation (12). The parameter ω plays now the role of a learning rate. And κS iterations mean κ scans of the whole training set.

So, the proposed method (12) is an incremental gradient method[26] for a least squares problem with block-preconditioning, associated with the blocks (10) of the equation (11):

$$\inf_{\varphi \in \mathbb{R}^N} \sum_{s=1}^{S} J(\varphi, C_s, g_s). \tag{19}$$

To finish with, the visualization of the reconstruction φ is based on the MIP: if $C = C_s$ is an observation of the training set, then the projection $\Pi_{C_s}\varphi$ is a kind of re-projection that must have similarities with the label g_s; otherwise $\Pi_C\varphi$ is a kind of prediction of what the scene looks like for a camera with parameter C.

3.5 Parameters of the method

3.5.1 Grid of voxels

At the beginning, we set a box by the means of the opposite corners $a = (a_1, a_2, a_3)$ and $b = (b_1, b_2, b_3)$. The box must contain the part of the scene that we want to reconstruct. Then we define a grid of voxels (4) for this box, by the means of the side h of a voxel. We take h on the order of the object resolution on the images g_s when enough data are available. Eventually h can be taken larger for reduction of the computational time and for safety reasons, especially for very limited data or inaccurate calibrations. The number N of voxels is then the product of the $n_i = 1 + \lceil \frac{b_i - a_i}{h} \rceil, 1 \leq i \leq 3$.

3.5.2 *Kaczmarz iterations*

Then we must set the rules for the Kaczmarz iterations. Let us recall three known facts[12,27] about ART in standard X-ray tomography. Firstly, the relaxation parameter takes often a small value such as $\omega = 0.05$. This enables to reconstruct in priority the low-frequency components of the sought attenuation during the iterations. The high-frequency components, including noise, appear earlier if ω is close to 1. The second point is about the ordering of the rays: scanning the rays in a random order (with significant rotations between two successive projections) during the iterations can improve the speed of convergence, and it is a way to reconstruct rapidly all the spatial frequencies. And the last point is the so-called *semi-convergence* behavior: the first iterates capture rapidly desired information with many details; then the method slows down, while the iterates deteriorate and capture undesired noise. We keep these three points in mind to guess satisfactory values for ART in reflective tomography. For the relaxation, we are looking for surfaces; this is linked with high-frequency spatial information. So we suggest $\omega = 0.5$ for instance; we will choose this value in the numerical experiments of the paper. Concerning the "random" ordering, we can choose $s_0 = 1$, and $s_{k+1} - 1 = s_k + p - 1 (\text{mod } S)$, where the step p and the number of images S are relatively prime numbers. To finish with, for the stop criterion, we can set in advance a small number of cycles κ such as $\kappa = 2$; in that case the computation of the RMSEs is optional. Alternately we stop the iterations as soon as the decay rate $\tau^{(k+1)}$ of the RMSE between two successive cycles is below a threshold $0 < \tau < 1$: $\eta^{(k)} - \eta^{(k+1)} \leq \tau \eta^{(k)}$; in that case we also introduce a safety bound κ_{max} over the number of cycles κ. In this paper, we will set $\tau = 5\%$ and $\kappa_{\max} = 8$.

3.5.3 *Inner regularization*

By the way each iteration of the Kaczmarz algorithm solves the linear system (13). We suggest to choose the regularization parameter σ on the order of Lh where L is a characteristic length of the box, such as the diagonal $L = \|b - a\|$. This is motivated by the following empirical reason: the matrix $X_{C_{s_k}} X_{C_{s_k}}^*$ of (13) is expected to be $O(Lh)$; and so the regularizing term may be negligible if $\sigma = o(Lh)$, while it may dominate if $Lh = o(\sigma)$. Indeed, the diagonal entries look like $\sigma + \sum_{k=1}^{N} \mathcal{X}[\psi_k](x_0, u_i)^2, 1 \leq i \leq M$. Roughly speaking the sum contains $O(L/h)$ contributing terms, each of them being $O(h^2)$; so the diagonal entries are expected to be $\sigma + O(Lh)$. By the way each non-diagonal entry (i, j) looks like $\sum_{k=1}^{N} \mathcal{X}[\psi_k](x_0, u_i)\mathcal{X}[\psi_k](x_0, u_j)$; this is bounded by $(\sum_k \mathcal{X}[\psi_k](x_0, u_i)^2)^{1/2}(\sum_k \mathcal{X}[\psi_k](x_0, u_j)^2)^{1/2}$, and thus it is expected to be $O(Lh)$ for the same reasons.

3.5.4 *Intermediate solver*

Once σ is fixed, the system (13) is solved by the conjugate gradient algorithm. This resolution is only an intermediate step in the global iterative procedure (Kaczmarz algorithm). So we do not need to solve the problem very precisely, and we perform only a few iterations of the conjugate gradient for efficiency reasons: for the stopping rule, we use a tolerance of 1% for the relative residual (with respect to the right hand side), combined with a bound of 10 iterations. We will see that this is enough to get satisfactory results.

3.5.5 *Visualization*

Concerning the visualization with our own cameras, we set a MIP camera with square pixels: for the calibration matrix (2), $s_{12} = 0$ and $s_1 = s_2 = s$. We want to get well-resolved images of voxels

of side h. So we require: $h \geq r$, where $r = \frac{\text{WD}}{fs}$ is the object space resolution (unit: length), WD being the working distance.

3.6 Costs

We estimate the global costs of the proposed approach. To simplify the expressions, we assume here: the recorded images are square of m^2 pixels ($m = m_1 = m_2$), the computed volume is a cube of $N = n^3$ voxels ($n = n_1 = n_2 = n_3$), and the MIP images are square of m^2 pixels.

3.6.1 Computational costs

We show that one cycle of iterations of the algorithm roughly costs about $O(nM)$ operations, where $M = Sm^2$ is the total number of recorded pixels, and n estimates the number of diagonal voxels of the reconstruction.

Let us consider indeed one iteration (12). The computational cost for the residual $g_{s_k} - X_{C_{s_k}} \varphi^{(k)}$ is dominated by the cost of an X-ray projection by ray-tracing: $O(m^2 n)$ operations. For the linear system (13) of the form $\left(X_{C_{s_k}} X^*_{C_{s_k}} + \sigma \mathrm{I} \right)^{-1} \bullet$, a single iteration of the conjugate gradient method costs about one evaluation of (14), estimated by the cost of a projection followed by a backprojection: $O(2m^2 n)$ operations. So the gradient conjugate iterations, with an upper bound of 10 iterations, costs $O(20m^2 n)$ operations. The final operation $\varphi^{(k)} + \omega X^*_{C_s} \bullet$ costs again about $O(m^2 n)$ operations, as a backprojection. And so, one iteration (12) costs about $O(22m^2 n)$, where the O implicitly contains the constant induced by the projection of a single voxel during ray-tracing. At the end, one cycle of iterations, *i.e.* S iterations, needs $O(22Sm^2 n) = O(Mn)$ operations. If we evaluate the RMSE (15) (optional), we add a cost dominated by the computational cost of the S residual images: it is again $O(Mn)$.

To finish with, the visualization costs about $O(m^2 n)$ operations per computed view (MIP).

3.6.2 Memory cost

The main memory cost comes typically from the unknown $\varphi \in \mathbb{R}^N$: $8N$ bytes for double precision. For moderate sizes, φ is stored directly in the RAM. Concerning the records g_s, each iteration loads a single recorded image in the RAM, due to the frame-driven approach. At the end, taking into account the various intermediate steps (including the computations of the $X^*_{C_{s_k}} v$ during the calls of the conjugate gradient), the RAM contains about $16N$ bytes.

3.7 Cross-validation

In practice, we know a set of images; we use the data to create a model, and then we generate new images based on the model. There is a question about the validity of the new images, or the "error of generalization" induced by the model. We can take benefit from the usual strategies of machine learning such as the cross-validation[15] to answer this question. In brief, one way consists in separating the dataset in two parts: a training set that is used to train the model, and a test set that is used to compare predictions of the model with known data. Furthermore this procedure is repeated for several partitions in order to compute statistics of performance.

For our study, the principle of the "k-fold cross-validation" is the following. We divide randomly the data set of S images into k subsets of (about) S/k images, where k is a fixed integer. For all $1 \leq i \leq k$, we select the i-th subset of S/k images as a test set. The other $S(1 - 1/k)$ images provide the i-th training set. We compute a reconstruction $\varphi[i]$ of the scene based on the i-th training set, using ART (12). Then we evaluate the i-th X-ray model $F_{\varphi[i]}(C) = X_C \varphi[i]$ on

the i-th test set by the means of the following RMSE, or the RRSE:

$$\eta[i] = \sqrt{\frac{1}{\sum_{s\in S[i]} M_s} \sum_{s\in S[i]} \|X_{C_s}\varphi[i] - g_s\|^2}, \quad \rho[i] = \frac{\eta[i]}{\hat{\sigma}[i]}, \quad 1 \leq i \leq k, \tag{20}$$

where $S[i]$ is the set containing the S/k indexes of the i-th test set, $\sum_{s\in S[i]} M_s$ is the total number of pixels over which the RMSE is computed, and $\hat{\sigma}[i]$ is the standard deviation of the corresponding pixel values $g_s, s \in S[i]$. It is worth mentioning that $\|X_{C_s}\varphi[i] - g_s\|^2$ aims at comparing the recorded image g_s with the X-ray image $X_{C_s}\varphi[i]$ generated by the voxel model $\varphi[i]$. However in practice, we visualize MIP images $\Pi_C\varphi$ rather than X-ray images $X_C\varphi$. So it would be more appropriate to compare $\Pi_{C_s}\varphi[i]$ and g_s; but it is difficult to define a suitable criterion. That is the reason why we focus rather on the RMSE/RRSE (20).

At the end, we get the RMSEs $\eta[i], 1 \leq i \leq k$, each one being computed from about S/k test images and about $S(1-1/k)$ training images; each initial image has been used exactly $k-1$ times as a training one, and once as a test image. Then we estimate some error of generalization of the model by the average of the RMSEs:

$$\hat{\eta} = \frac{1}{k}\sum_{i=1}^{k} \eta[i]. \tag{21}$$

Furthermore, the standard deviation $\hat{\sigma}_\eta$ of the RMSEs gives an idea on how much the quality of the predictions depends on the training set. And of course the same statistics can be computed for the RRSEs: average $\hat{\rho}$, and standard deviation $\hat{\sigma}_\rho$.

Lastly cross-validations can be eventually used to choose between several models corresponding to several sets of values of the model parameters (such as h, σ, τ, and even the camera param-

eters if needed): one cross-validation is realized for each of the models. At the end, we decide that the best set of parameters is the one for which the error of generalization is the smallest. This approach is attractive from a theoretical point of view, but we must emphasize that it multiplies the number of reconstructions to be computed. That is why in the numerical experiments of this paper, we will set the parameters from the considerations of subsection 3.5 rather than multiple cross-validations.

4 Experiments

4.1 Implementation

The full algorithm has been sequentially implemented in Fortran 2003. It includes the frame-driven Kaczmarz algorithm combined with the conjugate gradient for the intermediate solver. It also includes ray-tracing on a grid of voxels, for the computation of X-ray images, for the backprojection, and for the MIP. The code is executed on a workstation HP Z820 with processors Intel Xeon E5-2609, 2.40GHz. We will measure the time dedicated to the computation of the reconstructions: initialization, iterative updates of the model, iterative loading of the images, evaluation of RMSEs.

4.2 Middlebury datasets

The website[18] contains famous datasets for the evaluation of multi-view stereo reconstruction algorithms in computer vision.[17] The datasets contain various calibrated images of size $m = (640, 480)$, with three channels: RGB (Red, Green, Blue). The images have been corrected to remove radial distortion, and they have been calibrated with accuracy on the order of a pixel; a pixel spans about 1/4 mm on the object.

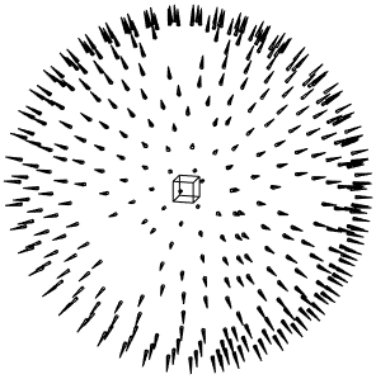


Fig 3 Camera positions for the Dino dataset. The reconstruction is computed inside the box.

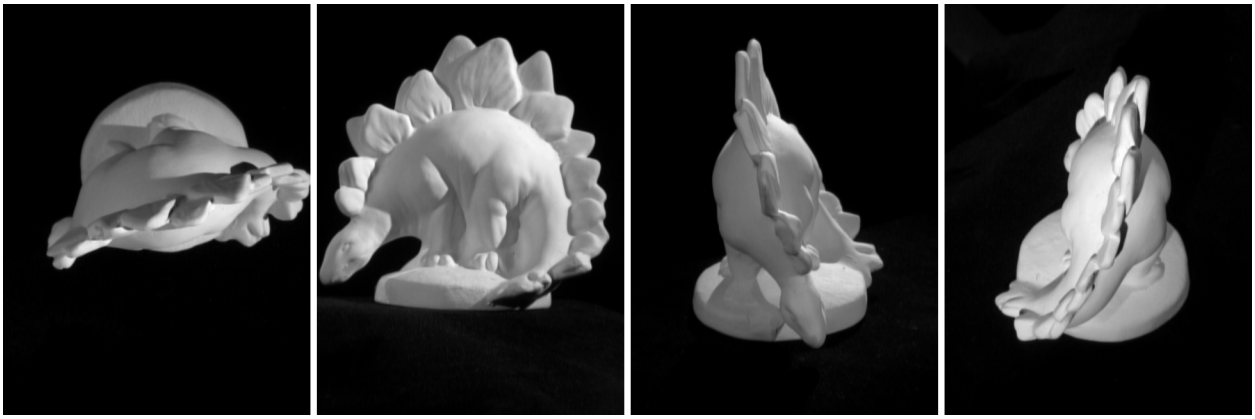
So the Middlebury datasets enter in the framework of this paper. We propose to test the ART (12) on images extracted from these datasets. We compute various reconstructions, on tight boxes suggested by the website.

4.3 Dino

We consider here the *Dino* data set, containing $S = 363$ views sampled on a hemisphere; see Figure 3 for the camera positions. For the image number $1 \leq s \leq S$, we set g_s as a grayscale image by adding the three channels RGB (0/765 represents black/white). See Figure 4 for samples of the sequence. We check here that the method is relevant for this relatively full dataset. (The unit length will be the meter, unless stipulated otherwise.)

4.3.1 Reconstruction

We compute the reconstruction with voxels of side $h = 0.00025$, in the box delimited by $a = (-0.041897, 0.001126, -0.037845)$ and $b = (0.030897, 0.088227, 0.03549)$; the diagonal length is $L = 0.13514$. We set: $\sigma = Lh$ for the regularization, $\omega = 0.5$ for the relaxation, $p = 13$ for the step. We iterate during 8 cycles, in 140400 s. In Table 1, we report some indicators of convergence. On Figure 5, we represent the evolution of re-projections based on rendering with a



(a) (b) (c) (d)

Fig 4 Samples of the Dino dataset: (a) g_{298} (b) g_{29} (c) g_{359} (d) g_{227}.

κ	0	1	2	3	4	5	6	7	8
$\eta^{(\kappa)}$	240.9	104.5	103.5	102.9	102.5	102.1	101.8	101.6	101.4
$\rho^{(\kappa)}$	1.207	0.5233	0.5185	0.5154	0.5131	0.5113	0.5099	0.5087	0.5076
$\tau^{(\kappa)}$		0.57	0.0092	0.006	0.0044	0.0035	0.0029	0.0024	0.0021

Table 1 Reconstruction from the Dino dataset: evolution of the RMSE $\eta^{(\kappa)}$, the RRSE $\rho^{(\kappa)}$ and the decay rate $\tau^{(\kappa)}$ of the RMSE; κ is the number of cycles of iterations.

MIP camera (9). The Figure 6 contains the evolution of a predicted view, using a new MIP camera: its parameters are not in the original dataset.

We observe a semi-convergence behavior: the RMSE rapidly decays at the beginning and the useful information is rapidly recovered; noise appears at a later stage, while the RMSE slowly decreases. It is worth mentioning that with a threshold $\tau = 0.05$ for the decay rate of the RMSE, the method would have stopped after $\kappa = 2$ cycles, with a RMSE that is almost the same than the RMSE after a single cycle. Furthermore, even if the renderings show the relevance of the reconstruction, the values of the RRSEs are relatively high (about 51%); we somehow recover that the dataset (VIS-NIR images) does not belong to the range of the X-ray transform.

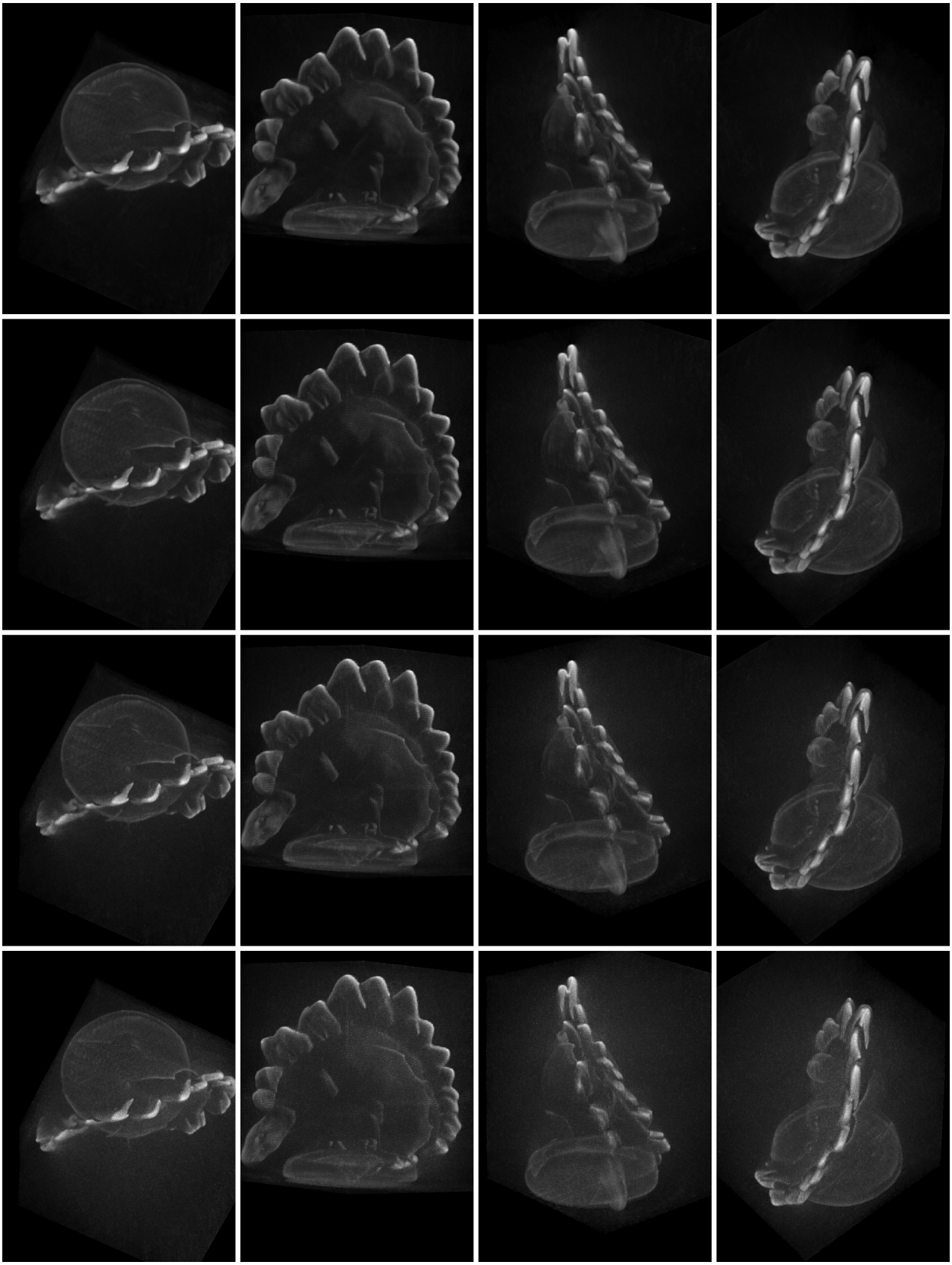
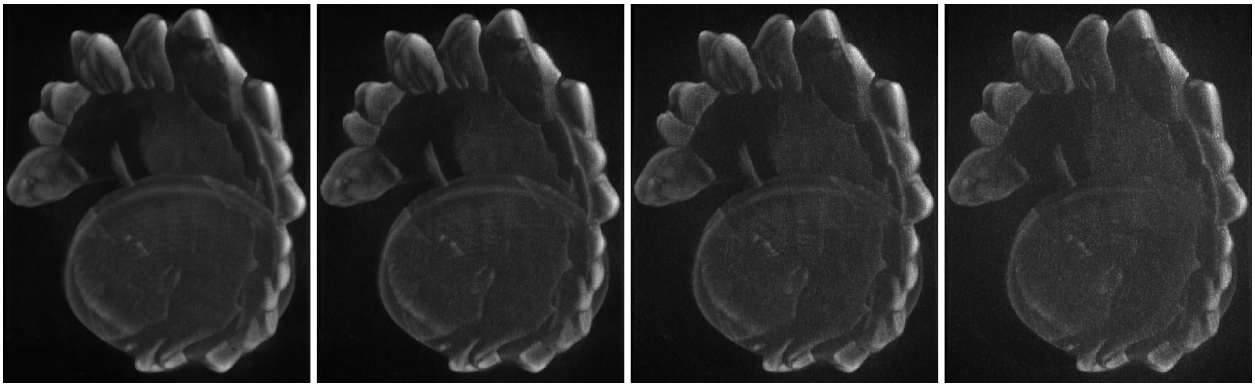


Fig 5 Re-projections of iterates from the Dino dataset: $\Pi_{C_s}\varphi^{(\kappa S)}$. From left to right: $s = 298, 29, 359, 227$; from top to bottom: $\kappa = 1, 2, 4, 8$ cycles of iterations. See Figure 4 for ground truth.



(a) (b) (c) (d)

Fig 6 Prediction from the Dino dataset after κ cycles: (a) $\kappa = 1$, (b) $\kappa = 2$, (c) $\kappa = 4$, (d) $\kappa = 8$. The MIP camera, with object resolution $r = 0.00015737$, is at working distance WD $= 2$.

4.3.2 Cross-validation

We test now the property of generalization: we realize a 4-fold cross-validation. The data set is divided into 4 subsets. We realize 4 experiments: we compute 4 reconstructions based on the 4 training sets. The subsets are such that the j-th image of Figure 4 is a test image for the j-th experiment, and a training image for the other experiments. We keep the same parameters than before, except the step: $p = 11$. For the stopping criterion, we iterate until the decay rate $\tau^{(k)}$ of the RMSE is below the threshold $\tau = 0.05$. We summarize in Table 2 some indicators measured after convergence, including the number of iterations κ to reach convergence, and the RMSE $\eta[i]$ computed over the training set. On Figure 7, the image (i, j) is the MIP of the i-th reconstruction, with the camera parameters of the j-th image of Figure 4. In particular the diagonal of Figure 7 contains predicted views, while the views outside the diagonal are re-projections.

The RMSEs/RRSEs over the test sets are larger but comparable with the RMSEs/RRSEs over the training sets, while the re-projections and the predictions look visually similar. Furthermore the quality of prediction does not severely depend on the training set.

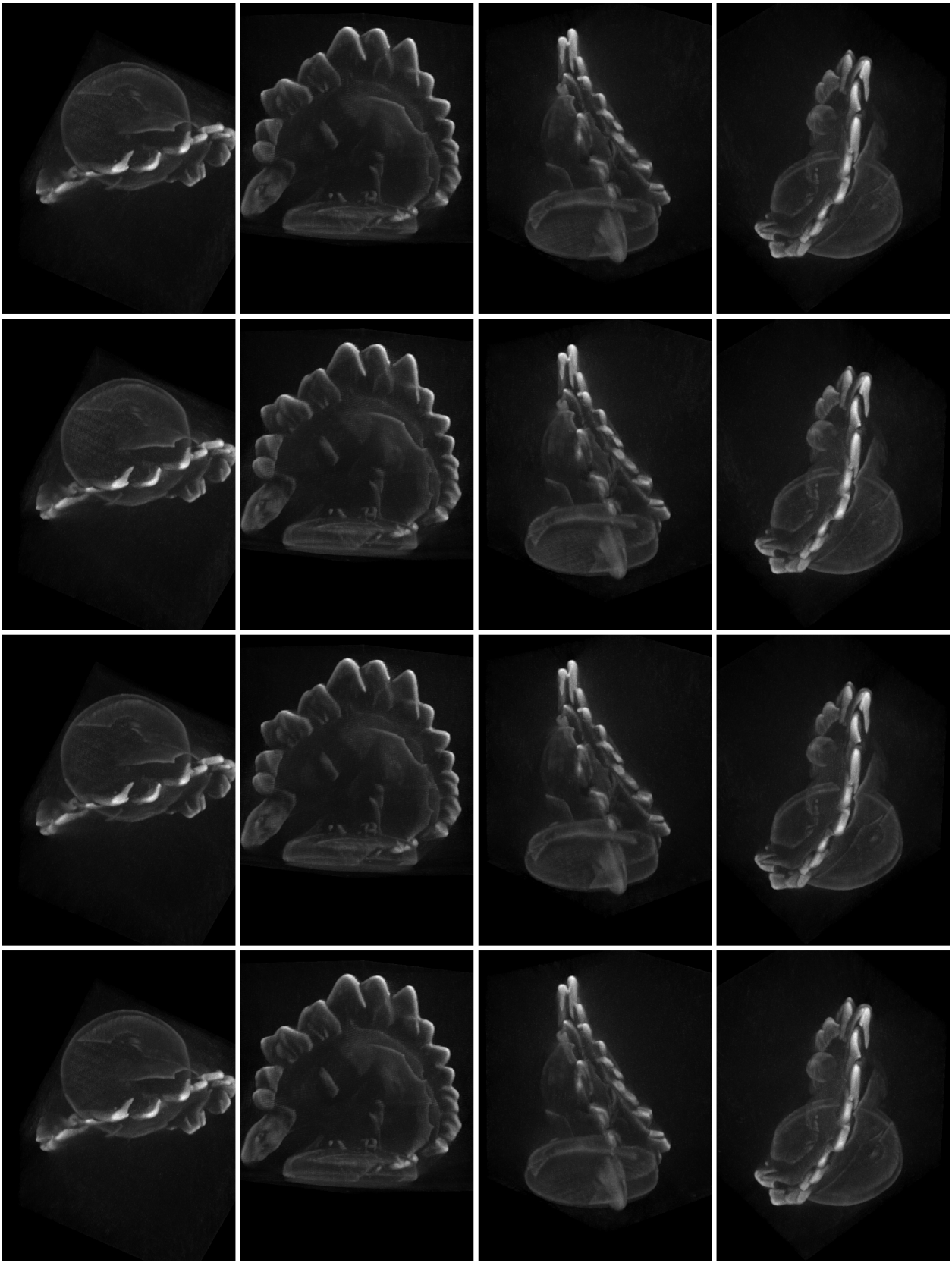


Fig 7 4-fold cross-validation for the Dino reconstruction. The line i contains MIP views from the i-th training set. The views are predictions on the diagonal; otherwise they are re-projections. See Figure 4 for ground truth.

Experiment i	1	2	3	4	
Number of training images	272	272	272	273	
Number of test images	91	91	91	90	
Number of cycles κ	2	2	2	2	
Time (s)	27730	28050	28710	28240	
RMSE $\eta^{(\kappa)}$ over the training set	101.1	101.4	100.3	103.1	
RRSE $\rho^{(\kappa)}$ over the training set	0.5067	0.5061	0.5051	0.5154	
RMSE $\eta[i]$ over the test set	104.8	106.4	109.1	104.3	$\hat{\eta} = 106, \quad \hat{\sigma}_\eta = 1.87$
RRSE $\rho[i]$ over the test set	0.5236	0.5394	0.5377	0.5260	$\hat{\rho} = 0.532, \hat{\sigma}_\rho = 0.00696$

Table 2 4-fold cross-validation for the Dino reconstruction. The last lines evaluate the trained models over the test sets.

4.4 Dino Sparse Ring

We consider here the *DinoSparseRing* dataset. It is similar with the Dino dataset, but it contains only $S = 16$ lateral views sampled on a ring around the object; see Figure 8. We construct the images g_s by adding the three channels RGB. We check that the method is still relevant even if the number of views is relatively small. Furthermore we investigate the choice of the side h of the voxels; it conditions the number of unknowns in the equation (11): doubling h divides by eight the number of unknowns.

4.4.1 Reconstruction

The tight box for the computation is given by $a = (-0.061897, -0.018874, -0.057845)$ and $b = (0.010897, 0.068227, 0.015495)$. We set: $\sigma = 2 \cdot Lh$ for the regularization, $\omega = 0.5$ for the relaxation, $p = 3$ for the step, and $\tau = 0.05$ for the stopping criterion. We apply the method for several voxel resolutions h: see Table 3 for efficiency/accuracy indicators. See Figure 9 for lateral re-projections, and see Figure 10 for top views predicted by a new MIP camera.

The reconstruction is sharper for the smallest values of h. But we also observe a pixelized noise for the resolution of the input images: $h = 0.00025$. Here the number of available data is $M \approx 4.9 \cdot 10^6$, while the number of unknowns is $N \approx 30 \cdot 10^6$ for $h = 0.00025$, and $N \approx 3.8 \cdot 10^6$

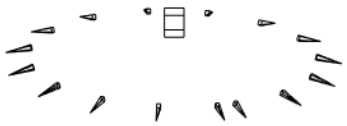
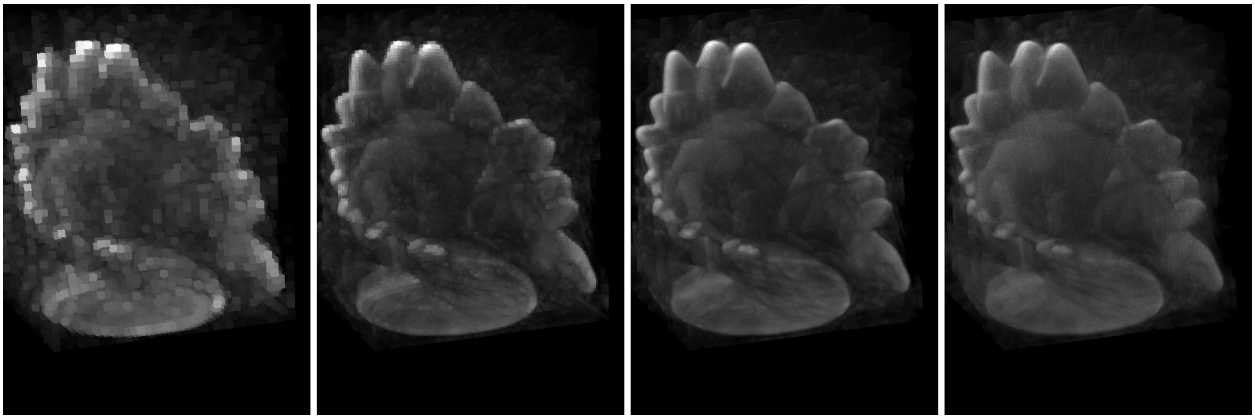


Fig 8 Camera positions for the DinoSparseRing dataset. The reconstruction is computed inside the box.

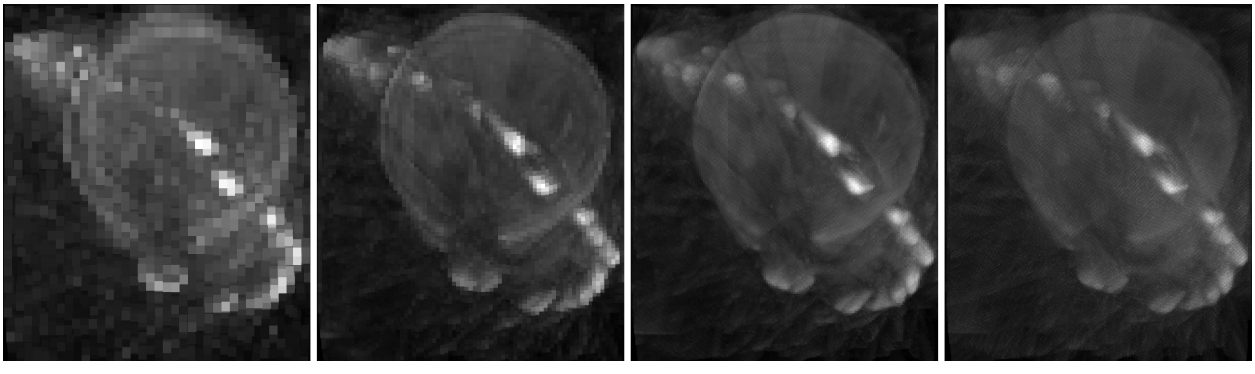
Resolution h (mm)	2	1	0.5	0.25
Number of cycles κ	2	3	3	4
Time (s)	423.4	770.7	1001	3091
RMSE $\eta^{(\kappa)}$	92.00	87.55	86.40	87.00
RRSE $\rho^{(\kappa)}$	0.4692	0.4465	0.4406	0.4436

Table 3 Indicators for the DinoSparseRing reconstruction, for several voxel resolutions h.



(a) (b) (c) (d)

Fig 9 Lateral re-projection of the DinoSparseRing reconstruction with voxel resolution (a) $h = 2$, (b) $h = 1$, (c) $h = 1/2$ and (d) $h = 1/4$ (mm).



(a) (b) (c) (d)

Fig 10 Top view predicted by the DinoSparseRing reconstruction. In mm, the voxel resolution is: (a) $h = 2$, (b) $h = 1$, (c) $h = 1/2$ and (d) $h = 1/4$; the MIP camera, object resolution $r = 0.15737$, is at working distance WD $= 2000$.

Experiment i	1	2	3	4	
Number of training images	12	12	12	12	
Number of test images	4	4	4	4	
Number of cycles κ	3	3	3	3	
Time (s)	725.6	731.7	746.0	775.9	
RMSE $\eta^{(\kappa)}$ over the training set	90.88	81.16	91.47	87.22	
RRSE $\rho^{(\kappa)}$ over the training set	0.4521	0.4295	0.4675	0.4399	
RMSE $\eta[i]$ over the test set	114.0	138.7	105.50	114.50	$\hat{\eta} = 118, \quad \hat{\sigma}_\eta = 12.3$
RRSE $\rho[i]$ over the test set	0.6319	0.6458	0.5349	0.6085	$\hat{\rho} = 0.605, \hat{\sigma}_\rho = 0.0428$

Table 4 4-fold cross-validation for the DinoSparseRing reconstruction.

for $h = 0.0005$. So we can understand that is safer to take $h = 0.0005$: bounding the value of h plays the role of a regularization procedure. Finally, the visual rendering and the RRSE both recommend the resolution $h = 1/2$ mm.

4.4.2 Cross-validation

We realize a 4-fold cross validation; the reconstructions are computed from 12 training images, and are tested against 4 test images. We set the following parameters: $h = 0.0005$, $\omega = 0.5$, $\sigma = 2Lh$, $\tau = 0.05$. We summarize some indicators in Table 4, and we represent MIP views on Figure 11.

The test sets and the training sets are relatively small, so we could expect strong variations in the quality of the reconstruction, and in its evaluation on the test set. Table 4 shows significant variations, but not huge ones. Furthermore the re-projections and the predictions look relatively similar. At the end, even if the number of training views (12) is relatively small, the voxel model keeps some ability to generalize.

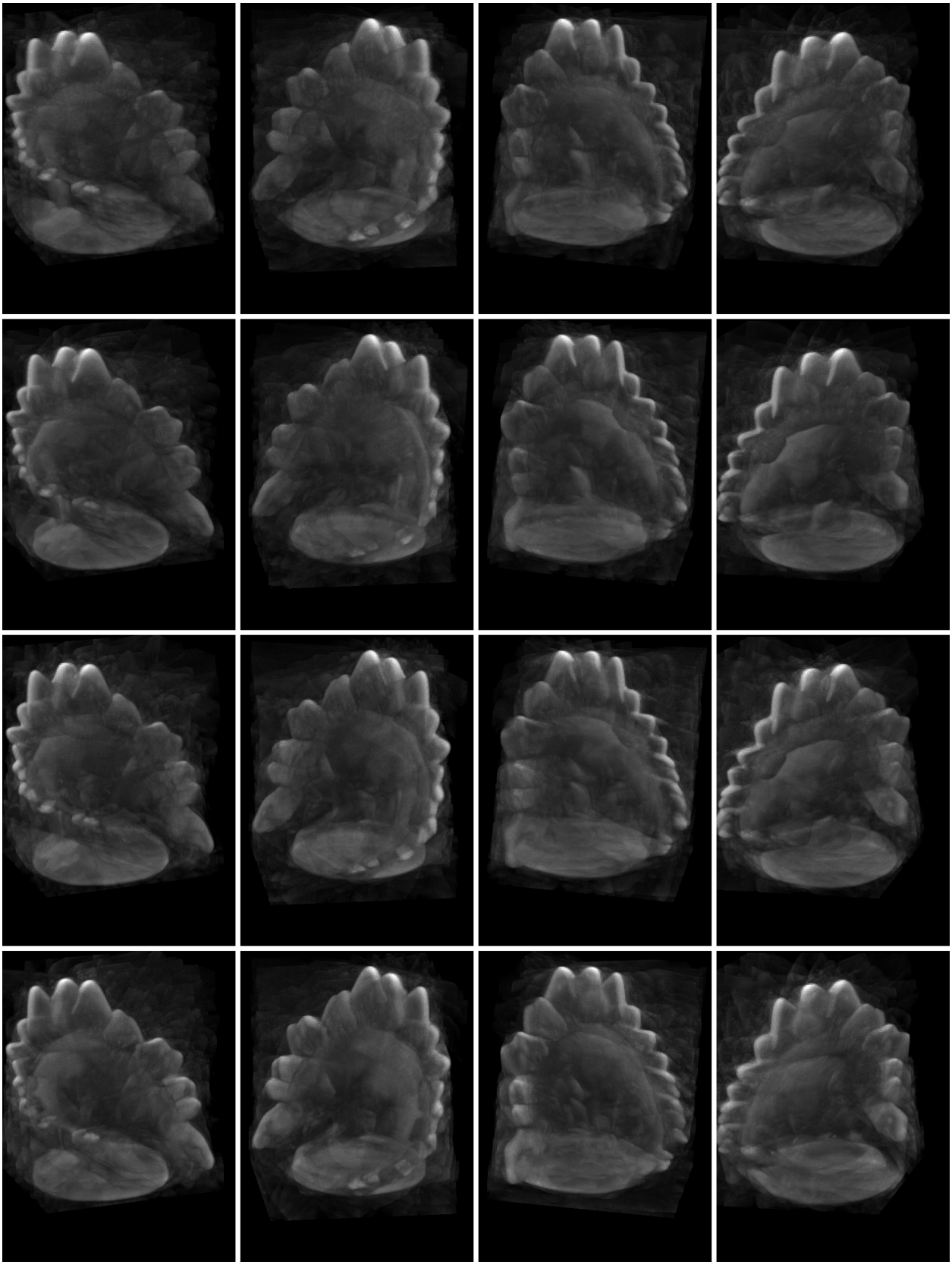


Fig 11 4-fold cross-validation for the DinoSparseRing reconstruction. The line i contains MIP views of the i-th trained model. The views are predictions on the diagonal; otherwise they are re-projections.

4.5 *Temple Sparse Ring*

We consider the *TempleSparseRing* dataset. It contains $S = 16$ RGB lateral views sampled on a ring around the object; see Figure 12. We construct the images g_s by extracting the Blue component (0/255 represents black/white). See Figure 13 for samples of the sequence. We illustrate the influence of the inner Tikhonov regularization.

We set: $a = (-0.073568, 0.021728, -0.012445)$ and $b = (0.028855, 0.181892, 0.062736)$ for the box, $L = 0.20443$ for the diagonal, $h = 0.0005$ for the voxels, $\omega = 0.5$ for the relaxation, $p = 3$ for the step, $\tau = 0.05$ for the stopping rule. We apply the method for $\frac{\sigma}{Lh} = 0.01, 0.1, 1, 10, 100$; due to $Lh \approx 0.0001$, these successive values correspond also to several powers of Lh: $\sigma \approx (Lh)^{1.5}, (Lh)^{1.25}, Lh, (Lh)^{0.75}, (Lh)^{0.5}$ (up to a multiplicative 1 with the right unit). We summarize several indicators in Table 5. On Figure 14 we represent re-projections. Furthermore, we predict air-ground images on Figure 15.

For the smallest regularization, the conjugate gradient stops always due to the bound (10) on the number of iterations. The regularization is not really efficient in that case and we obtain dark images with strong peaks; they should be thresholded/rescaled to be useful. Otherwise the conjugate gradient stops always once the admitted tolerance 10^{-2} is reached. The reconstructions for the strongest regularization are fastly obtained but are blurred: the regularization term predominates and the preconditioner of (17) is not efficient enough. The most acceptable results are the intermediate regularizations, for which σ is roughly on the order of Lh; $\sigma = Lh$ realizes indeed a good compromise between accuracy (small RMSE) and efficiency (small computational time). Is is also a compromise for MIP rendering, between dark images with peaks and blurred bright images.

Fig 12 The TempleSparseRing dataset contains 16 "ground-ground" views. The reconstruction is computed inside the box, and is used to predict 4 "air-ground" views (cameras in bold).

(a) (b) (c) (d)

Fig 13 Samples of the TempleSparseRing dataset: (a) g_1, (b) g_5, (c) g_9 and (d) g_{13}.

Ratio $\frac{\sigma}{Lh}$ (dimensionless)	0.01	0.1	1	10	100
Number of cycles κ	4	4	5	5	1
Time (s)	2287	2049	1318	1036	162.3
RMSE $\eta^{(\kappa)}$	15.64	15.36	16.15	23.39	43.33
RRSE $\rho^{(\kappa)}$	0.4035	0.3961	0.4166	0.6032	1.118

Table 5 Indicators for the TempleSparseRing reconstruction, for several inner regularizations σ.

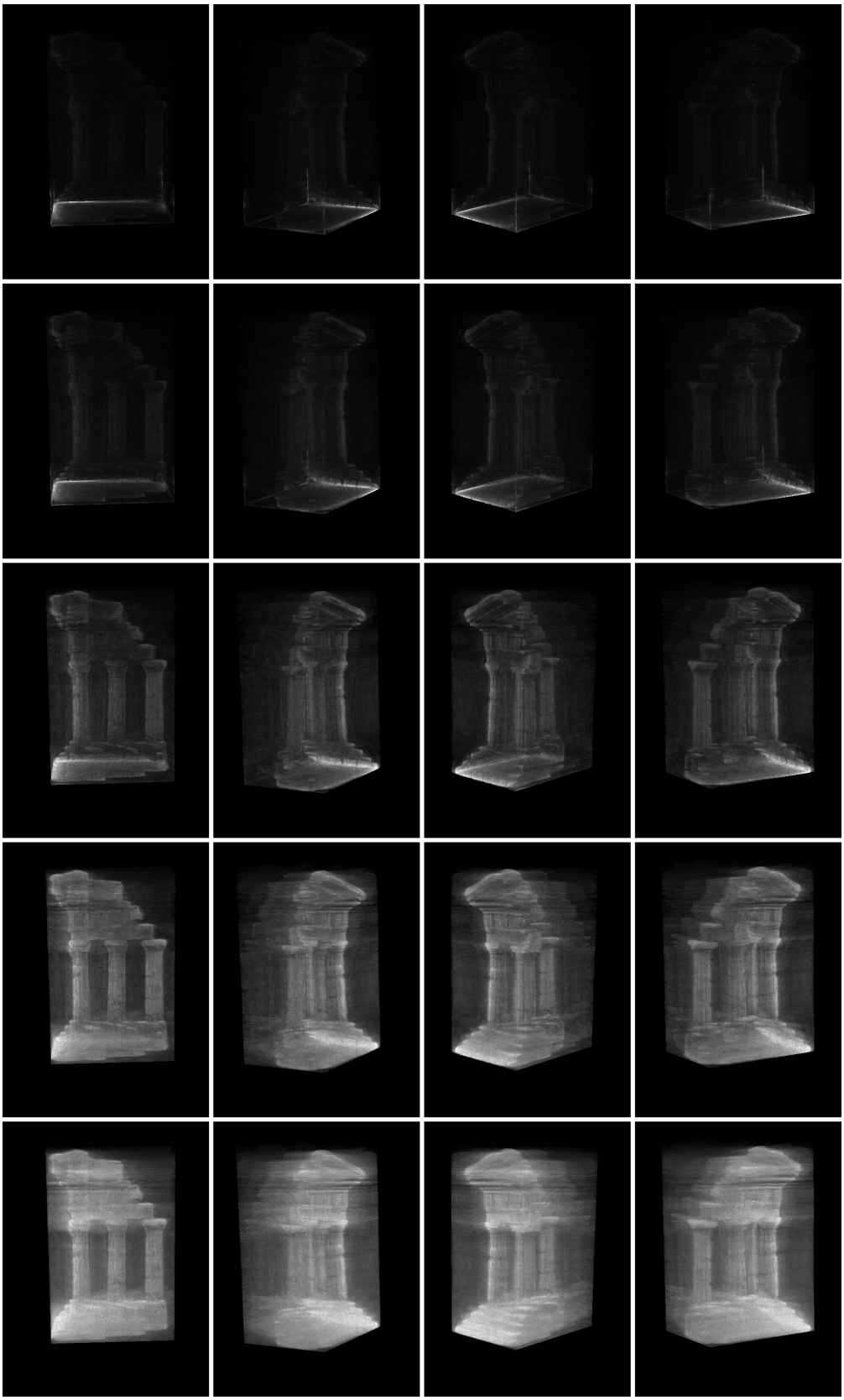


Fig 14 Re-projections of the TempleSparseRing reconstruction, for several inner regularizations: from top to bottom, $\frac{\sigma}{Lh} = 0.01, 0.1, 1, 10, 100$. See Figure 13 for ground truth.

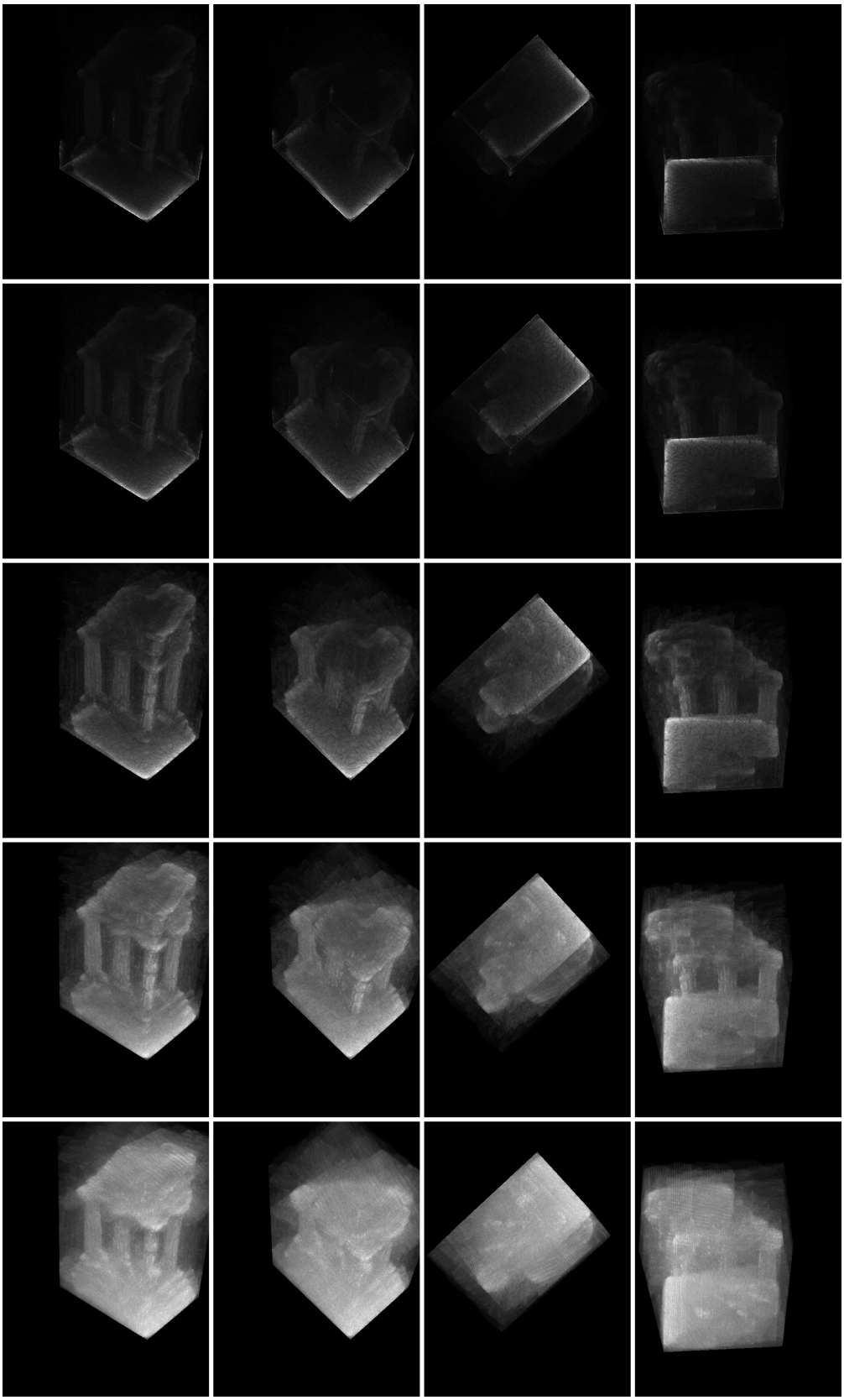


Fig 15 Air-ground views predicted by the TempleSparseRing reconstruction, for several inner regularizations: from top to bottom, $\frac{\sigma}{Lh} = 0.01, 0.1, 1, 10, 100$. See Figure 12 for the camera positions.

4.6 Temple

We consider the *Temple* dataset. It contains 312 RGB views sampled on a hemisphere; see Figure 16 for the camera positions. On Figure 17, we represent the three components (R,G,B) separately, and the sum R+G+B, for the views number 220, 245 and 302. It appears that the specular reflection is more pronounced on the Red; while the Blue is the most diffuse and the sharpest. We test several ways to deal with the multi-channel property.

For the test (a), the dataset is obtained by extraction of the Red component. For the test (b), we extract the Blue component. For the test (c), we add the channels (R,G,B) of the original images. For the test (d), we consider one RGB image as three different images. For (a), (b), (c), the dataset contains $S = 312$ images, while for (d), it contains $S = 936$ images (312 R, then 312 G, then 312 B). We set: $a = (-0.054568, 0.001728, -0.042945)$ and $b = (0.047855, 0.161892, 0.032236)$ for the box, $h = 0.0005$ for the voxel side, $\sigma = Lh$ for the regularization, $\omega = 0.5$ for the relaxation, $p = 11$ for the step, and $\tau = 0.05$ for the stopping criterion. See Table 6 for indicators, see Figure 18 for re-projections, and see Figure 19 for predictions.

We get similar reconstructions, despite some slight differences. It is worth mentioning that the RRSE is an indicator of quality without being an absolute criterion: the smallest RRSE is the

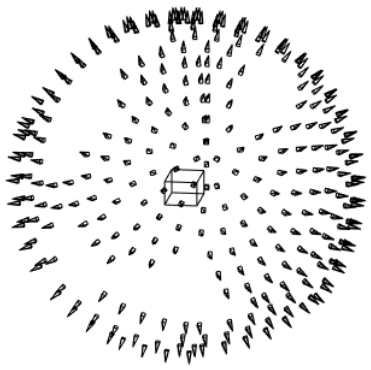
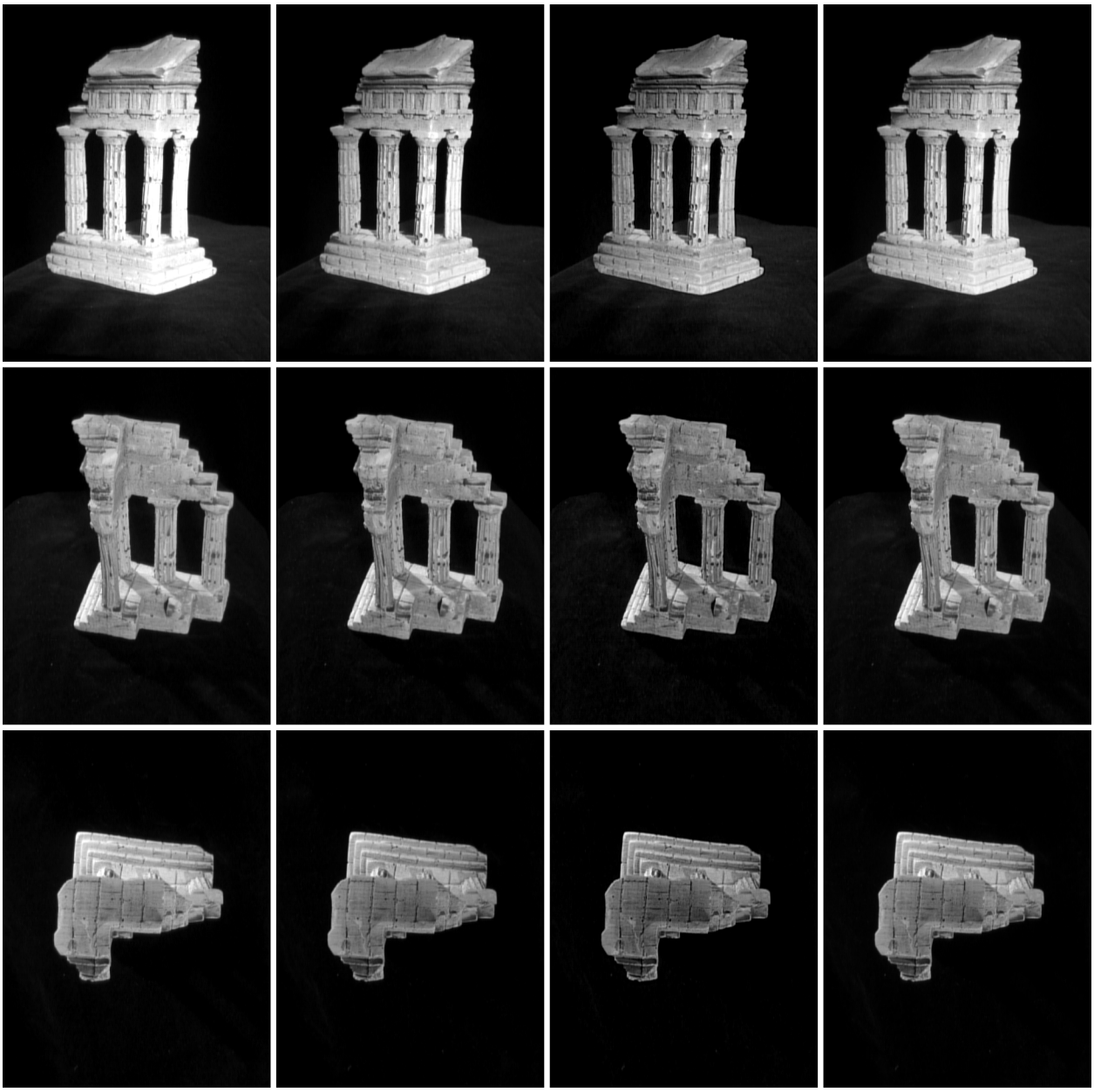


Fig 16 The Temple dataset contains 312 views on a hemisphere. The reconstruction is computed inside the box.

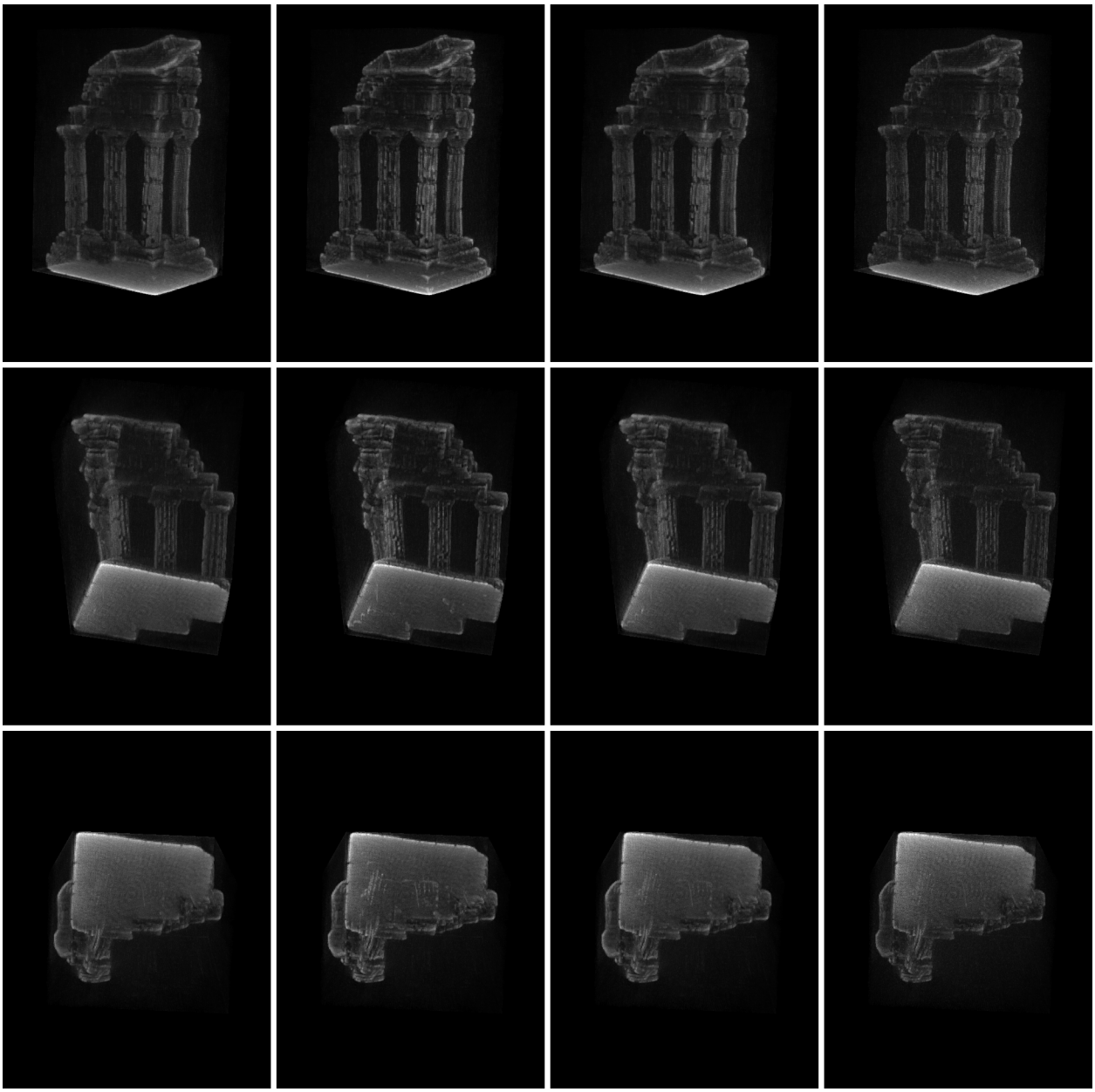


(a) (b) (c) (d)

Fig 17 Channels from the Temple dataset: (a) Red, (b) Green, (c) Blue and (d) Gray=Red+Green+Blue.

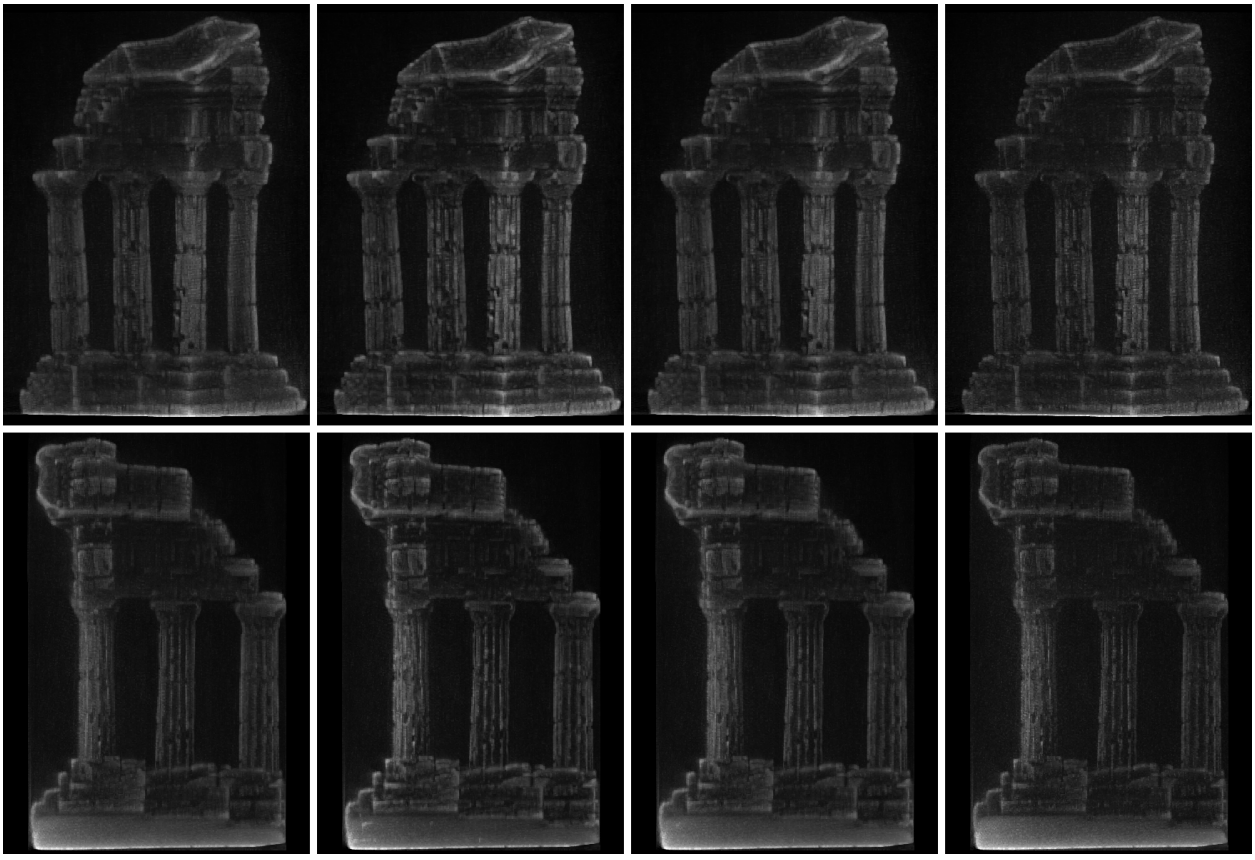
Channel	R	B	R+G+B	R,G,B
Number of cycles κ	2	2	2	2
Time (s)	11520	11840	10680	35150
RMSE $\eta^{(\kappa)}$	37.73	22.06	91.15	33.01
RRSE $\rho^{(\kappa)}$	0.5548	0.5754	0.5594	0.5858

Table 6 Indicators for the reconstructions from several Temple channels.



(a) (b) (c) (d)

Fig 18 Re-projections of the reconstruction, for several Temple channels: (a) R, (b) B, (c) R+G+B and (d) R,G,B. See Figure 17 for ground truth.



(a) (b) (c) (d)

Fig 19 Predictions from the Temple reconstruction, for several channels: (a) R, (b) B, (c) R+G+B and (d) R,G,B. The MIP camera, object resolution $r = 0.00028612$, is at working distance WD $= 5$.

case (a) from the Red component while the case (b) from the Blue component looks the sharpest. Furthermore the ART manages successfully the redundancies and the inconsistencies between the three channels, case (d): it extracts itself the useful information.

4.7 Extreme scenario

To finish with we create a very restricted dataset from the Temple dataset: a training set containing $S = 3$ images. The first image g_1 is the Red component of the image number 194 of the Temple dataset, the second image g_2 is the Green component of the image 32, and the third image g_3 is the Blue component of the image 41. The images g_1, g_2, g_3 could represent images taken from

three different cameras; g_1 is air-ground, g_2 and g_3 are ground-ground: see Figure 20 and the first line of Figure 21. Furthermore we create a test set containing the Red component of the 309 other images of the Temple dataset. Reconstructing the scene using only the proposed training set is challenging. In particular, even if the $g_s, 1 \leq s \leq 3$ were X-ray images, the Tuy's condition would be very seriously violated. Even if we cannot expect reflective tomography to recover perfectly the scene, the approach of this paper is applicable, at least. Furthermore we will test here the computed model against the test set, considered as ground truth.

We set: $a = (-0.054568, 0.001728, -0.042945)$ and $b = (0.047855, 0.161892, 0.032236)$ for the box, $h = 0.001$ for the voxel side, $\sigma = 3Lh$ for the regularization, $\omega = 0.5$ for the relaxation, $p = 1$ for the step, and $\tau = 0.05$ for the stopping criterion. See Table 7 for the indicators. For the rendering, we replace the lower threshold of the MIP (9) by 800 (instead of 0) in order to start denoising. We represent the three re-projections of the reconstruction on the second line of Figure 23. We represent test views against MIP predictions on Figure 22; the camera positions are in bold on Figure 20. And we represent predictions based on a rotation of our own MIP camera, on Figure 23.

We see that even if the number of views is very restricted, the ART is still able to capture some features and details and has still some ability to generalize, especially for views that are not too far from the training ones.

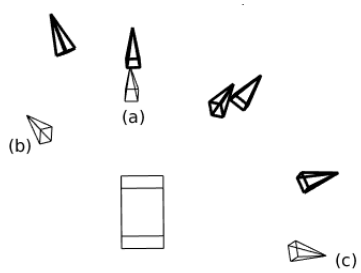
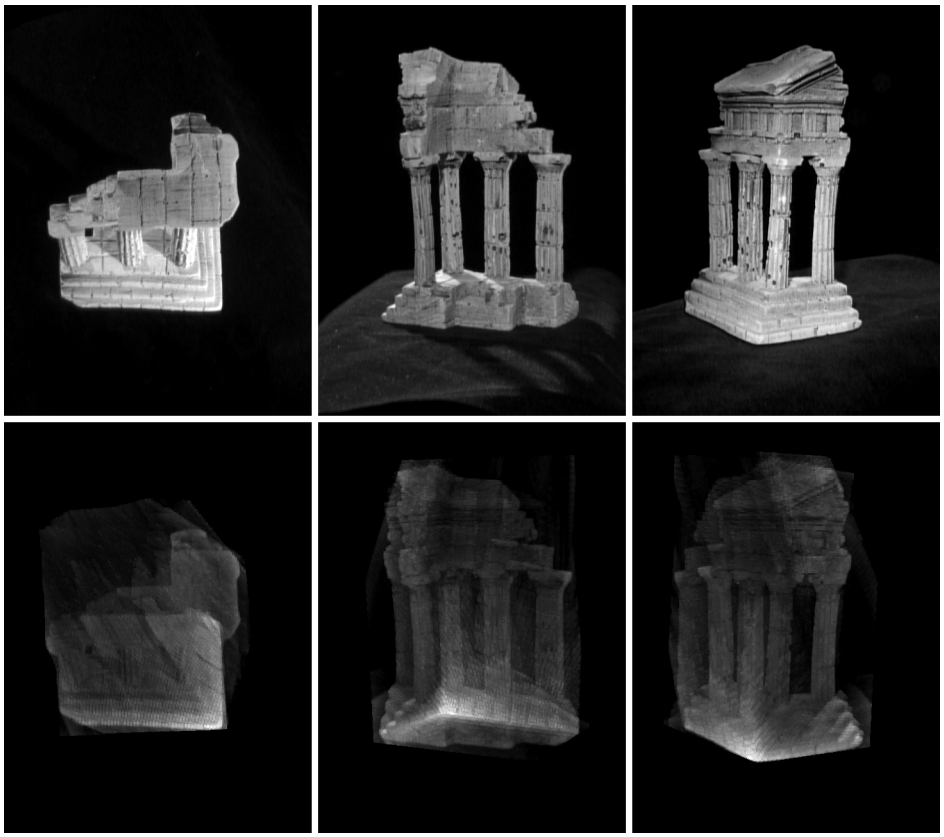


Fig 20 The restricted dataset contains: (a) one air-ground view (b-c) two ground-ground views. The reconstruction is computed inside the box, and will be displayed on the cameras in bold.



(a) (b) (c)

Fig 21 Reconstruction from three training images. Top: dataset (a) g_1, (b) g_2 and (c) g_3. Bottom: re-projections.

Number of training images	3
Number of test images	309
Number of cycles κ	6
Time (s)	156.9
RMSE $\eta^{(\kappa)}$ over the training set	19.72
RRSE $\rho^{(\kappa)}$ over the training set	0.3585
RMSE $\eta[1]$ over the test set	44.58
RRSE $\rho[1]$ over the test set	0.6560

Table 7 Cross-validation for the very restricted dataset, using the Temple dataset as ground truth. The last lines evaluate the trained model over the test set.

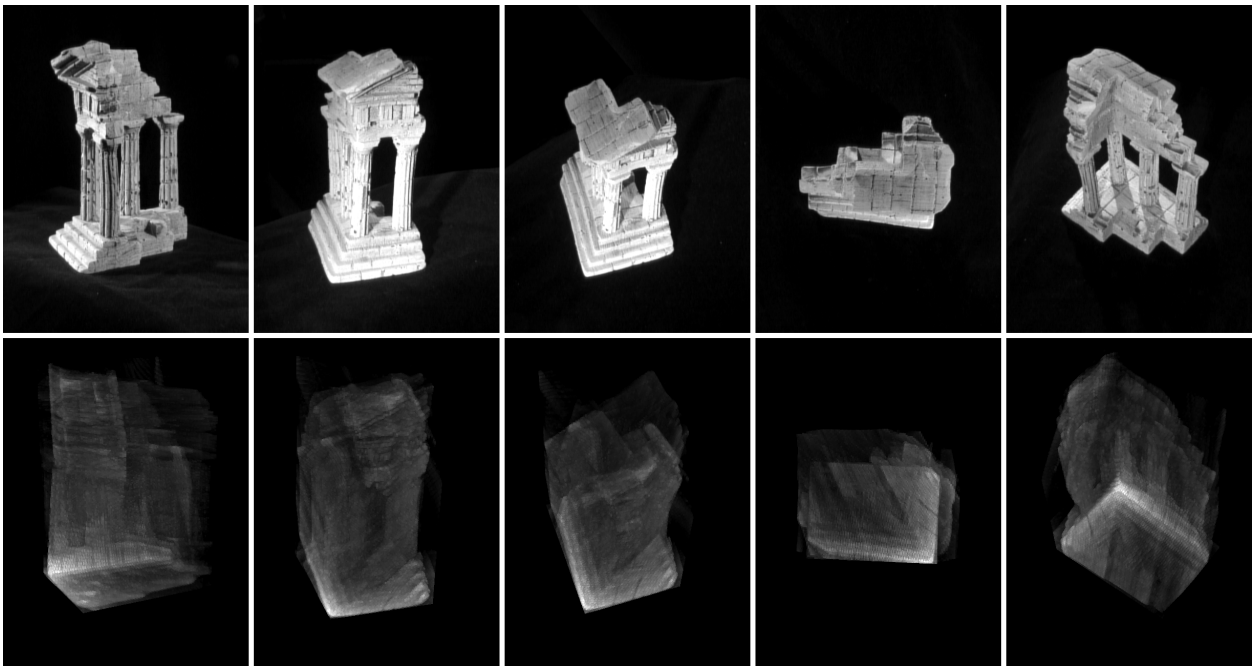


Fig 22 Reconstruction from three views. Top: test views. Bottom: MIP predictions. See Figure 20 for the camera positions.

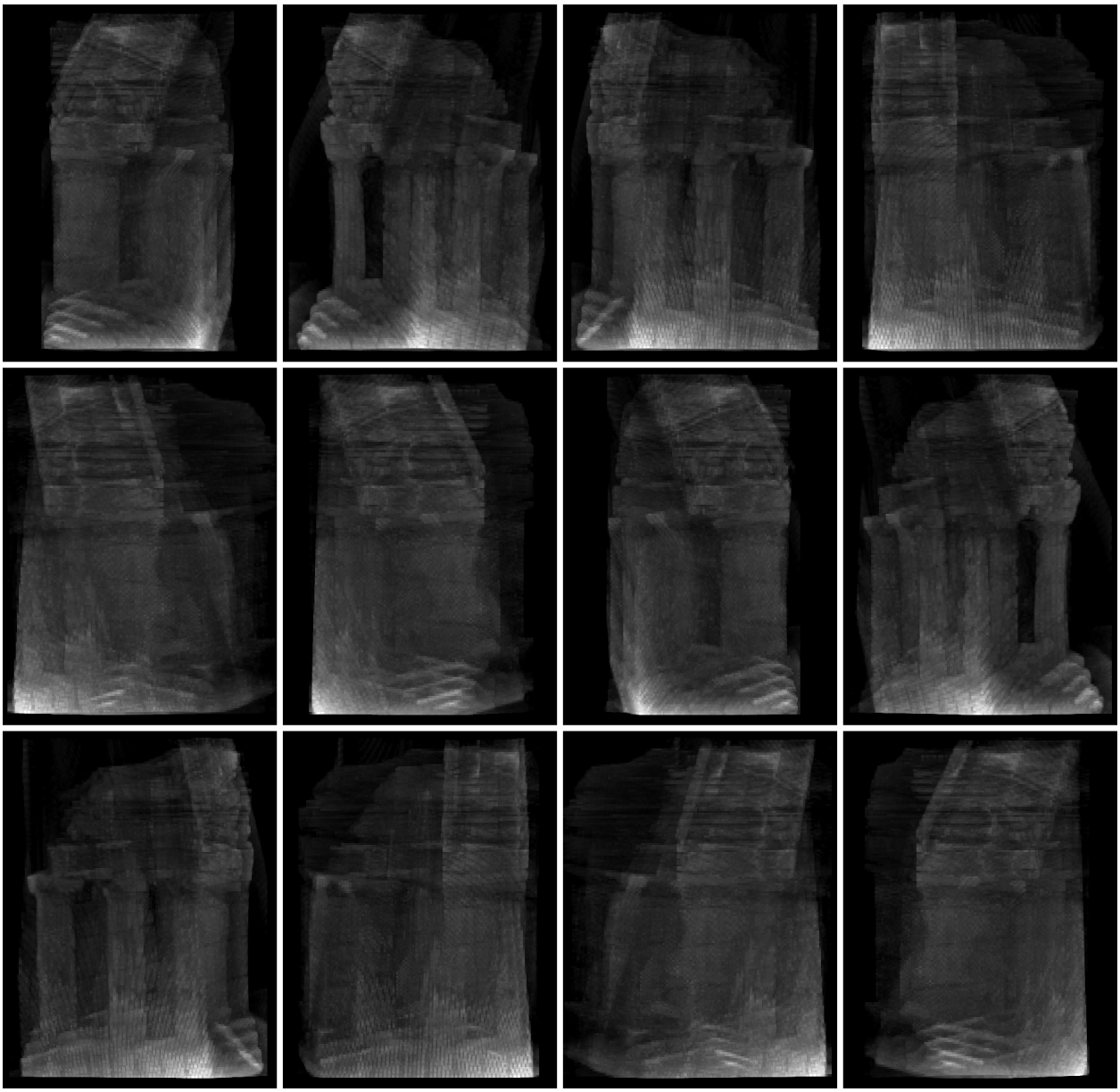


Fig 23 Predictions from three views. The MIP camera, object resolution $r = 0.00028612$, working distance WD $= 5$, rotates on a circle around the reconstruction.

5 Conclusion

This paper formulates reflective tomography under the form of a least squares problem with block-preconditioning, and solves it by an incremental learning algorithm. The method is a frame-driven Kaczmarz algorithm, inspired by the ART of X-ray tomography. It provides a voxels method for multiple-view reconstruction in VIS-NIR optics, when the recorded images are calibrated. For the computation of a cube of N voxels from M recorded pixels, one cycle of iterations of the algorithm costs about $O(MN^{1/3})$ operations. For a practical use where the scene is unknown, the cross-validation is a way of estimating the quality of the computed model, if enough data are available. Numerical experiments on real datasets show the relevance of the algorithm, even if the number of available images is relatively small. Also this paper is one more empirical proof that the X-ray transform is able to capture features of images and to build a relevant model containing the geometry of the scene, despite the dataset is not in the range of the transform.

The algorithm of this paper is purely based on linear algebra techniques and automatically catches geometric features. So it is robust and flexible, and various scenarios of acquisition are practicable. We can imagine a camera with a continuous motion, such as an onboard camera, with arbitrarily trajectory. The method accepts also the merge of several datasets. For example ground-ground images measured by a pedestrian could be combined with air-ground images measured by an aircraft or by a satellite. In fact, merging datasets is a way to augment the set of measured features and so it is a way to perfect the reconstruction.

Finally the author hopes that the relevance of the algebraic methods will offer new opportunities in three-dimensional optical imaging, such as new practical uses of reflective tomography.

References

1 R. Marino, R. Capes, W. Keicher, *et al.*, "Tomographic image reconstruction from laser radar reflective projections," in *Laser radar III*, **999**, 248–269, International Society for Optics and Photonics (1989).

2 F. Knight, D. Klick, D. Ryan-Howard, *et al.*, "Two-dimensional tomographs using range measurements," in *Laser radar III*, **999**, 269–281, International Society for Optics and Photonics (1989).

3 A. G. Ramm and A. I. Katsevich, *The Radon transform and local tomography*, CRC press (1996).

4 D. T. Gering and W. Wells, "Object modeling using tomography and photography," in *Multi-View Modeling and Analysis of Visual Scenes, 1999.(MVIEW'99) Proceedings. IEEE Workshop on*, 11–18, IEEE (1999).

5 C. L. Matson, D. E. Holland, D. F. Pierrottet, *et al.*, "Satellite feature reconstruction using reflective tomography: field results," in *Optics in Atmospheric Propagation and Adaptive Systems II*, **3219**, 65–73, International Society for Optics and Photonics (1998).

6 J. B. Lasche, C. L. Matson, S. D. Ford, *et al.*, "Reflective tomography for imaging satellites: experimental results," in *Digital Image Recovery and Synthesis IV*, **3815**, 178–189, International Society for Optics and Photonics (1999).

7 G. Berginc and M. Jouffroy, "Optronic system and method dedicated to identification for formulating three-dimensional images." US patent 20110254924 A1, European patent 2333481 A1, FR 09 05720 B1 (2009).

8 G. Berginc and M. Jouffroy, "Simulation of 3D laser systems," in *Geoscience and Remote Sensing Symposium, 2009 IEEE International, IGARSS 2009*, **2**, 440–444, IEEE (2009).

9 G. Berginc, "Scattering models for 1-D-2-D-3-D laser imagery," *Optical Engineering* **56**(3), 031207 (2016).

10 G. Berginc, J.-B. Bellet, I. Berechet, *et al.*, "Optical 3D imaging and visualization of concealed objects," in *Proc. SPIE*, **9961**, 99610Q (2016).

11 M. Henriksson, T. Olofsson, C. Grönwall, *et al.*, "Optical reflectance tomography using TC-SPC laser radar," in *Proc. SPIE*, **8542**, 85420E (2012).

12 F. Natterer and F. Wübbeling, *Mathematical methods in image reconstruction*, SIAM (2001).

13 G. T. Herman, *Handbook of mathematical methods in imaging*, ch. 16 Tomography, 691–733. Springer Science & Business Media (2010).

14 E. P. Magee, C. L. Matson, and D. Stone, "Comparison of techniques for image reconstruction using reflective tomography," in *Image Reconstruction and Restoration*, **2302**, 95–103, International Society for Optics and Photonics (1994).

15 C.-A. Azencott, *Introduction au Machine Learning*, Dunod (2018).

16 J. Lin and D.-X. Zhou, "Learning theory of randomized Kaczmarz algorithm," *The Journal of Machine Learning Research* **16**(1), 3341–3365 (2015).

17 S. M. Seitz, B. Curless, J. Diebel, *et al.*, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, **1**, 519–528, IEEE (2006).

18 S. Seitz, B. Curless, J. Diebel, *et al.*, "Multi-view stereo evaluation web page," *URL http://vision.middlebury.edu/mview* (2006).

19 Y. Ma, S. Soatto, J. Kosecka, *et al.*, *An invitation to 3-D vision: from images to geometric models*, vol. 26, Springer Science & Business Media (2012).

20 B. Horn, *Robot vision*, MIT press (1986).

21 R. L. Siddon, "Fast calculation of the exact radiological path for a three-dimensional ct array," *Medical physics* **12**(2), 252–255 (1985).

22 F. Jacobs, E. Sundermann, B. De Sutter, *et al.*, "A fast algorithm to calculate the exact radiological path through a pixel or voxel space," *Journal of computing and information technology* **6**(1), 89–94 (1998).

23 S. Berechet, I. Berechet, J.-B. Bellet, *et al.*, "Method for discrimination and identification of objects of a scene by 3-D imaging." Patent EP3234914B1 (2018).

24 P. C. Hansen, *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*, vol. 4, Siam (2005).

25 Y. Saad, *Iterative methods for sparse linear systems*, vol. 82, siam (2003).

26 D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM Journal on Optimization* **7**(4), 913–926 (1997).

27 P. C. Hansen, "Regularization in tomography - dealing with ambiguity and noisy data," in *COST workshop Advanced X-Ray Tomography: Experiment, Modeling, and Algorithms, Lorentz Center*, http://people.compute.dtu.dk/pcha/Talks (Feb. 10-14, 2014).

Jean-Baptiste Bellet is an assistant professor in applied mathematics at the University of Lorraine. He received his engineering degree in applied mathematics from the Institut National des Sciences Appliquées of Rouen in 2007, his MS degree in mathematics from the University of Rouen in

2007, and his PhD degree in applied mathematics from the Ecole Polytechnique of Palaiseau in 2010. His current research interests include imaging and scientific computing.

List of Figures

9 Lateral re-projection of the DinoSparseRing reconstruction with voxel resolution (a) $h = 2$, (b) $h = 1$, (c) $h = 1/2$ and (d) $h = 1/4$ (mm).

10 Top view predicted by the DinoSparseRing reconstruction. In mm, the voxel resolution is: (a) $h = 2$, (b) $h = 1$, (c) $h = 1/2$ and (d) $h = 1/4$; the MIP camera, object resolution $r = 0.15737$, is at working distance WD $= 2000$.

11 4-fold cross-validation for the DinoSparseRing reconstruction. The line i contains MIP views of the i-th trained model. The views are predictions on the diagonal; otherwise they are re-projections.

12 The TempleSparseRing dataset contains 16 "ground-ground" views. The reconstruction is computed inside the box, and is used to predict 4 "air-ground" views (cameras in bold).

13 Samples of the TempleSparseRing dataset: (a) g_1, (b) g_5, (c) g_9 and (d) g_{13}.

14 Re-projections of the TempleSparseRing reconstruction, for several inner regularizations: from top to bottom, $\frac{\sigma}{Lh} = 0.01, 0.1, 1, 10, 100$. See Figure 13 for ground truth.

15 Air-ground views predicted by the TempleSparseRing reconstruction, for several inner regularizations: from top to bottom, $\frac{\sigma}{Lh} = 0.01, 0.1, 1, 10, 100$. See Figure 12 for the camera positions.

16 The Temple dataset contains 312 views on a hemisphere. The reconstruction is computed inside the box.

17 Channels from the Temple dataset: (a) Red, (b) Green, (c) Blue and (d) Gray=Red+Green+Blue.

18 Re-projections of the reconstruction, for several Temple channels: (a) R, (b) B, (c) R+G+B and (d) R,G,B. See Figure 17 for ground truth.

List of Tables

7 Cross-validation for the very restricted dataset, using the Temple dataset as ground truth. The last lines evaluate the trained model over the test set.