Error Backprojection Algorithms for Non-Line-of-Sight Imaging

Marco La Manna, Fiona Kine, Eric Breitbach, Jonathan Jackson, Andreas Velten

Abstract—Recent advances in computer vision and inverse light transport theory have resulted in several non-line-of-sight imaging techniques. These techniques use photon time-of-flight information encoded in light after multiple, diffuse reflections to reconstruct a three-dimensional scene. In this paper, we propose and describe two iterative backprojection algorithms, the additive error backprojection (AEB) and multiplicative error backprojection (MEB), whose goal is to improve the reconstruction of the scene under investigation over non-iterative backprojection algorithms. We evaluate the proposed algorithms' performance applied to simulated and real data (gathered from an experimental setup where the system needs to reconstruct an unknown scene). Results show that the proposed iterative algorithms are able to provide better reconstruction than the unfiltered, non-iterative backprojection algorithm for both simulated and physical scenes, however are more sensitive to errors in the light transport model.

Index Terms—Non-line-of-sight (NLOS) Imaging, Time-of-Flight, Seeing-around-corners, Algebraic Reconstruction Technique (ART), Kaczmarcz method.

1 Introduction

Non-line-of-sight (NLOS) imaging has attracted a lot of interest due to recent breakthroughs made in computer vision and inverse transient light transport theory. Specifically, researchers have shown ways to exploit the information embedded in multiple, diffuse reflections [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. In these time-of-flight NLOS techniques, a relay surface in a scene is illuminated by short light pulses and the returned light from the relay surface is detected in a dataset $\boldsymbol{s}(p, q, t)$ that encodes light intensity as a function of the illuminated and detected pixels on the wall (p and q) and the time between illumination and detection (t) (refer to Fig. 1).

We can generally describe the reconstruction for a given collected dataset $\boldsymbol{s}$ as an inverse rendering problem. For a given scene geometry $\boldsymbol{b}$, the corresponding $\boldsymbol{s}$ can be estimated using standard computer graphics rendering techniques described by a function $\mathcal{F}(\cdot)$. The task is then to find its inverse, namely

$$\boldsymbol{b} = \mathcal{F}^{-1}(\boldsymbol{s}), \tag{1}$$

to reconstruct the scene geometry from the collected data [8], [12], [13]. If the function $\mathcal{F}(\cdot)$ can be approximated by a linear operator, say $\boldsymbol{A}$, the resulting linear inverse problem can be solved in different ways, such as convex optimization [4], [14]. However, this approach has two challenges. First, the matrix $\boldsymbol{A}$ becomes too large to handle even for fairly moderate resolutions of $\boldsymbol{b}$ and $\boldsymbol{s}$. Second, incorporating light transport effects like Lambertian shading, occlusions, multiple scattering in the scene, and a scene's bidirectional reflectance distribution function (BRDF) make $\mathcal{F}(\cdot)$ nonlinear.

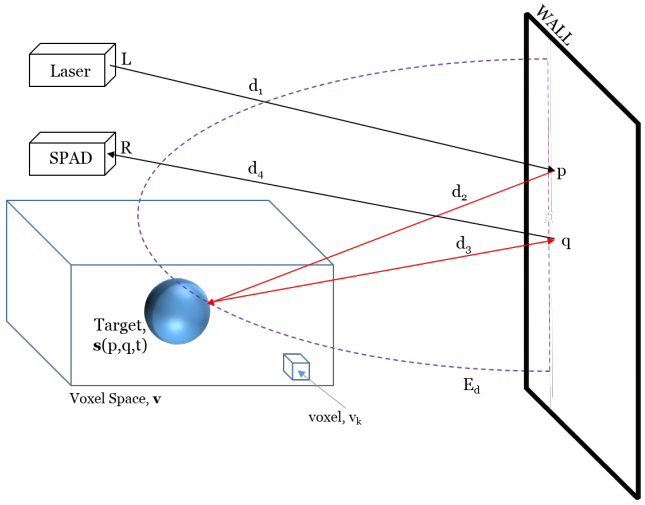


Fig. 1. Light path in a simple scenario that comprises a spherical target which is not in the direct line of sight of the laser and camera.

Backprojection methods [6], [15], similar to backprojections used in the computational tomography field [16], [17], allow for robust, highly parallelizable reconstructions using limited memory and computation time, but provide only an approximate solution and cannot incorporate priors or complex light transport. In this paper, we provide two iterative algorithms, which we call additive error backprojection (AEB) and multiplicative error backprojection (MEB). These algorithms have been implemented in a memory efficient way and have been tested with both simulated and real data. Results show that these iterative algorithms can improve the reconstruction over the non-iterative algorithm,

- *M. La Manna, F. Kine, E. Breitbach, A. Velten (corresponding author) are with the Department of Biostatistics and Medical Informatics, University of Wisconsin – Madison, Madison, WI 53706 USA; e-mail: {lamanna2, fkine, ebreitbach2, velten}@wisc.edu. Jonathan Jackson is with the Physics Department at University of California, Berkeley, Berkeley, CA 94720; e-mail: jonathanj@berkeley.edu.*

This work was supported by DARPA REVEAL Program (Grant no. DARPA-BAA-15-44 MSN189781), NASA NIAC Program (Grant no. NNH15ZOA001N-15NIAC A2) AFRSO (Grant no. AFOSR-2014-0003-cidYIP-2015), ONR (Grant no. Open BAA 15-001), and Morgridge Institute for Research.

but are more sensitive to errors in the light transport model. Furthermore, the AEB and MEB are functional implementations of the algebraic reconstruction technique (ART) [18], [19], [20], [21], [22] (also known in numerical algebra as the Kaczmarz method [23]), commonly used for computed tomography reconstructions. The goal of this technique is to provide an iterative algorithm for solving consistent, linear systems of the form

$$\boldsymbol{A}\,\boldsymbol{b} = \boldsymbol{s}. \tag{2}$$

If (2) has a solution, it has been shown that the method converges [24]. Note, however, that the complete form of ART requires a linear matrix implementation of the forward model resulting in prohibitive memory requirements for most real datasets. Our iterative backprojection algorithms implement the forward projection and backprojection operations as memory efficient functions that are not strictly limited to linear light transport models.

The remainder of the paper is organized as follows. In Sections 2 and 3, we discuss the backprojection and forward projection algorithms, respectively. In Section 4, we describe the proposed iterative algorithms. In Section 5, we show the algorithms' performance results by applying them to simulated scenarios, whereas in Section 6, we use publicly available experimental data (cf. [15]). In Section 7, we draw conclusions and discuss potential future work.

2 Backprojection

Since the proposed algorithms are based on the backprojection routine, let us briefly review the algorithm proposed in [6]. Our goal is to reconstruct an unknown 3D scenario given a dataset created by illuminating points on a visible relay wall in the scene with short pulses. In Fig. 1, the light source, located at L, illuminates individual points (or small patches) on a relay wall at known positions. For simplicity, we have shown a single point, p, in Fig. 1. For each of the illuminated positions, a fast detector or detector array collects light reflected from known positions on the wall (represented by q, in Fig. 1), as a function of time. Between the illumination and the detection, time is proportional to the total distance traveled from the light source to the detector, indicated as $d = d_1 + d_2 + d_3 + d_4$, in Fig. 1. Assuming multiple laser positions $p_1, p_2, \ldots, p_M$ and camera positions, $q_1, q_2, \ldots, q_N$, the collected light intensities are stored in $\boldsymbol{s}$, a 7-dimensional space, composed of laser coordinates, camera coordinates, and time, $t \in [0, T]$. In other words, the dataset $\boldsymbol{s}$ contains the measured light intensities (photons) for each laser and camera position pair combination over an interval of time T.

In order to recover the scene, we define a 3D reconstruction volume, $\boldsymbol{v}$, uniformly discretized into K voxels, $\boldsymbol{v} = [v_1, v_2, \ldots, v_K]$, where v_k is a unit-cube with a predefined edge length (see Fig. 1). The signal collected at the detector is due to light scattering off of a target located within this voxel space. For a given photon, the subset of voxels in $\boldsymbol{v}$ that could have been the associated target locations form a surface of locations with equal travel time through points p_m and q_n; this surface is an ellipsoid, E_d, whose foci are p_m and q_n, as shown in Fig. 1.

The goal of the backprojection algorithm is to project each data sample in $\boldsymbol{s}$ onto the voxels on the associated ellipsoid E_d. Therefore, for each voxel v_k, we obtain a real value, which we call b_k; the higher this value, the higher the likelihood that a target surface is inside the considered voxel. Loosely speaking, we refer to b_k as the *confidence* of having a target inside the v_k-th voxel.

For computational purposes, it is easier to define the backprojection as a sum over data samples for a given voxel, rather than a set of voxels for a given data sample. For $k = 1, \ldots, K$, we can calculate b_k as

$$b_k = \begin{cases} \sum_{m=1}^{M} \sum_{n=1}^{N} \alpha_{kmn}\, s(p_m, q_n, \frac{d_{kmn}}{c}) & 0 \le \frac{d_{kmn}}{c} < T, \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where

$$\begin{aligned} d_{kmn} = d_1(L, p_m) + d_2(p_m, v_k) + \\ d_3(v_k, q_n) + d_4(q_n, R), \end{aligned} \tag{4}$$

in which L and R are the laser and camera origin, respectively; $s(p_m, q_n, \frac{d_{kmn}}{c})$ is the measured data projected back to the ellipses[1] containing the k-th voxel. Moreover,

$$\alpha_{kmn} = \alpha_d(v_k, p_m, q_n)\, \alpha_{ls}(v_k, p_m, q_n) \tag{5}$$

is a weight factor that takes into account the distance attenuation factor, $\alpha_d(v_k, p_m, q_n)$, and the Lambertian shading $\alpha_{ls}(v_k, p_m, q_n)$. The former is calculated as

$$\alpha_d(v_k, p_m, q_n) = \frac{d_2^2(p_m, v_k) d_3^2(v_k, q_n) d_4^2(q_n, R)}{A_s\, A_{cam}\, A_{ap}}, \tag{6}$$

where A_s is the target's area, A_{cam} and A_{ap} are the camera area looking at the wall and the camera's aperture, respectively. Note that $d_1(L, p_m)$ is not considered in $\alpha_d(v_k, p_m, q_n)$, because we assume that the laser source is collimated. The Lambertian shading is calculated as

$$\alpha_{ls}(k, m, n) = 1 / \left\langle \frac{u_2(p_m, v_k)}{||u_2(p_m, v_k)||}, \frac{u_3(v_k, q_n)}{||u_3(v_k, q_n)||} \right\rangle, \tag{7}$$

in which $\langle \cdot, \cdot \rangle$ represents the dot product, u_2 is the surface normal w.r.t the wall's point p_m and voxel v_k and u_3 the normal w.r.t the voxel v_k and the wall's point q_n. The operation expressed in (3) is called backprojection, which we will indicate in vector form as

$$\boldsymbol{b} = \mathcal{B}(\boldsymbol{s}), \tag{8}$$

where $\boldsymbol{b} = [b_1, \cdots, b_K] \in \mathcal{R}^K$ is a vector that contains the confidence of all the voxels. In Algorithm 1, we provide a pseudocode that explains how the backprojection operates.

Once the values of the $\boldsymbol{b}$ vector have been calculated, [6], [15] apply some post-processing filters to enhance the target features. More specifically, a spatial high-pass filter is employed to enhance surfaces. Adaptive and denoising steps may be considered, but are not included here.

1. Recall that the foci of the ellipses are the laser and camera position on the relay wall.

Algorithm 1 Pseudocode for the backprojection algorithm, $\mathcal{B}(\boldsymbol{s})$, proposed in [6].

Input: Dataset, $\boldsymbol{s}$
Voxel space: $\boldsymbol{v}$
Laser positions: $p_1, \ldots, p_M$
Camera positions: $q_1, \ldots, q_N$
Output: Confidence, $\boldsymbol{b}$

for k = 1 to K **do**
  Voxel v_k coordinates
  Initialize $b_k = 0$
  for m = 1 to M **do**
    Compute $d_1(L, p_m)$ and $d_2(p_m, v_k)$
    for n = 1 to N **do**
      Compute $d_3(v_k, q_n)$ and $d_4(q_n, R)$
      Compute $d = \sum_{j=1}^{4} d_j$
      if ($t = \frac{d}{c} \leq T$) **then**
        Compute $b_k = b_k + \alpha_{kmn}\, s(p_m, q_n, \frac{d}{c})$
        $\alpha_{kmn} = \alpha_d(v_k, p_m, q_n)\, \alpha_{ls}(v_k, p_m, q_n)$
        α_d *see (6)*
        α_{ls} *see (7)*
      end if
    end for
  end for
end for
return

3 Forward Projection

The forward projection function, $\mathcal{F}(\cdot)$, has the goal of creating a correspondence between the voxel space and the data space, and it is an implementation of the rendering equation, outlined in (1). Its use is twofold: assuming a non-zero b_k value obtained from the backprojection, the forward projection creates the corresponding data *as if* a target were present in v_k. This operation is repeated for all the voxels in $\boldsymbol{v}$ such that $b_k > 0$ ($b_k = 0$ implies target absence in the considered v_k voxel). We provide the forward projection pseudocode for this case in Algorithm 2. Alternatively, the function $\mathcal{F}(\cdot)$ can create a simulated dataset by directly providing the target location (corresponding to a non-zero b_k value in the previous description).

Our implementation is optimized for fast execution and models the impulse response of the detector as a Gaussian with a specific full width half maximum (FWHM).

4 Iterative Algorithms

The proposed iterative algorithms correlate time-of-flight data and measured intensities to reconstruct the scattering object positions that make up the scene.

Consider a generic scenario, such as the one in Fig. 1, with multiple laser positions $p_1, \ldots, p_M$ and camera positions $q_1, \ldots, q_N$. Given the scenario dataset, $\boldsymbol{s}_1$, the AEB starts by computing a backprojection $\mathcal{B}(\boldsymbol{s}_1)$, cf. (3), projecting each collected data sample over an ellipsoid in the reconstruction volume, $\boldsymbol{v}$. In other words, at the first iteration, given the dataset, $\boldsymbol{s}_1$, the AEB algorithm computes a backprojection, $\boldsymbol{b}_1 = \mathcal{B}(\boldsymbol{s}_1)$, in a similar manner to [6]. At the next step, we find the reprojected dataset, $\boldsymbol{s}_2 = \mathcal{F}(\boldsymbol{b}_1)$, namely a forward projection of all non-zero voxels in $\boldsymbol{b}_1$. Now, we proceed in calculating the error (element-wise subtraction), $\boldsymbol{\Delta}_2$, between the true dataset, $\boldsymbol{s}_1$ and $\boldsymbol{s}_2'$ (the reprojected dataset, $\boldsymbol{s}_2$, normalized version w.r.t. the maximum of $\boldsymbol{s}_1$),

$$\boldsymbol{\Delta}_2 = \boldsymbol{s}_1 - \boldsymbol{s}_2', \tag{9}$$

and then find its backprojection, $\mathcal{B}(\boldsymbol{\Delta}_2)$. Defining $\gamma_A \in (0, 1]$ as the step size (namely, a weighting factor), the AEB corrected backprojection at the second iteration is

$$\boldsymbol{b}_2 = \boldsymbol{b}_1 + \gamma_A\, \mathcal{B}(\boldsymbol{\Delta}_2). \tag{10}$$

At the i-th step, the AEB corrected backprojection can be expressed iteratively as

$$\boldsymbol{b}_i = \boldsymbol{b}_{i-1} + \gamma_A\, \mathcal{B}(\boldsymbol{s}_1 - \mathcal{F}(\boldsymbol{b}_{i-1})), \tag{11}$$

where the step size, γ_A, is constant throughout all the iterations. Similarly to (9), $\mathcal{F}(\boldsymbol{b}_{i-1})$ has been normalized w.r.t. the maximum of $\boldsymbol{s}_1$, but we have omitted the superscript, to avoid a heavy notation.

In a similar manner to the AEB, the MEB starts with a backprojection of the provided dataset. We can express the MEB updated step as

$$\boldsymbol{b}_i = \gamma_M \odot \boldsymbol{b}_{i-1} \odot \mathcal{B}\left(\boldsymbol{s}_1 \oslash \mathcal{F}(\boldsymbol{b}_{i-1})\right), \tag{12}$$

where γ_M is the MEB step size (constant throughout all the iterations); '$\odot$' and '$\oslash$' represent the element-wise multiplication and division operators, respectively.

As a stopping criterion for both of the proposed iterative algorithms, we use exploit the mean square error (MSE). At the i-th iteration, the MSE, $\mathcal{E}_i$, can be defined as

$$\mathcal{E}_i = ||\boldsymbol{b}_i - \boldsymbol{b}_{i-1}||^2, \tag{13}$$

Algorithm 2 Pseudocode for the forward projection algorithm, $\mathcal{F}(\boldsymbol{b})$.

Input: Confidence $\boldsymbol{b}$
Voxel space: $\boldsymbol{v}$
Laser positions: $p_1, \ldots, p_M$
Camera positions: $q_1, \ldots, q_N$
Output: Dataset $\boldsymbol{s}$

for k = 1 to K **do**
  if $b_k > 0$ **then**
    Voxel v_k coordinates
    for m = 1 to M **do**
      Compute $d_1(L, p_m)$ and $d_2(p_m, v_k)$
      for n = 1 to N **do**
        Compute $d_3(v_k, q_n)$ and $d_4(q_n, R)$
        Compute $d = \sum_{j=1}^{4} d_j$
        if ($\frac{d}{c} \leq T$) **then**
          Compute
          $s(p_m, q_n, t) = \alpha_{kmn}\, e^{-\frac{(t-d/c)^2}{2\sigma^2}}$
          $t \in [0, T]$
          $\sigma = f(FWHM)$
          $\alpha_{kmn,f} = \alpha_{d,f}(v_k, p_m, q_n)\, \alpha_{ls,f}(v_k, p_m, q_n)$
          $\alpha_{d,f} = 1/\alpha_d$
          $\alpha_{ls,f} = 1/\alpha_{ls}$
        end if
      end for
    end for
  end if
end for
return

for $i > 2$, and where $\boldsymbol{b}_i$ is the backprojection at the current iteration and $\boldsymbol{b}_{i-1}$ is the backprojection at the previous iteration. More specifically, if $\mathcal{E}_i < \mathcal{E}_{i-1}$, the algorithms proceed in calculating the backprojection results. However, the algorithms stop if one of the following conditions becomes true: if $\mathcal{E}_i < 10^{-20}$ or if $\mathcal{E}_i > \mathcal{E}_{i-1}$. In the latter situation, the results obtained at the previous iteration, $(i-1)$, are considered the final ones, whereas the results at the i-th iteration are discarded.

As we discussed in Section 1, if we assume that $\mathcal{F}(\cdot)$ is linear, the proposed algorithms are an implementation of ART [18], [20]. This technique reconstructs a scene using multiple projections, each corresponding to a different row[2] in (2). Using our notation, we can write the additive ART equation as [25]

$$\boldsymbol{b}_i = \boldsymbol{b}_{i-1} + \boldsymbol{A}^T \boldsymbol{M}^{-1} (\boldsymbol{s}_1 - \boldsymbol{A}\, \boldsymbol{b}_{i-1}) \tag{14}$$

where $\boldsymbol{b}_i$ and $\boldsymbol{b}_{i-1}$ are the backprojection at the i-th and $i-1$-th iteration, respectively; $\boldsymbol{A}^T$ is the transpose of the matrix $\boldsymbol{A}$, $\boldsymbol{M}$ is a diagonal scaling matrix, $\boldsymbol{s}_1$ is the measured projection (available dataset), whereas $\boldsymbol{A}\, \boldsymbol{b}_{i-1}$ is the estimated projection (or reprojected dataset). Note that $\boldsymbol{A}^T$ is fundamentally the backprojection function $\mathcal{B}(\cdot)$ that we introduced in (11) and (12). For this reason, apart from the scaling factors in $\boldsymbol{M}$ and the added weight factors (γ_A and γ_M), the AEB expressed in (11), is equivalent to (14). By replacing the addition and subtraction operators in (14) with the multiplication and division operators, we obtain the MEB equation shown in (12).

5 Application to Simulated Data

In the following subsections, we provide some numerical results relative to the iterative algorithms that have been discussed above. We consider two simulated scenarios and, in both cases, embed the target in additive noise. This serves to approximate afterpulsing, the dominant noise found in SPAD detectors (our cameras), which can be modeled as uniformly random variables; we choose to generate these variables such that their amplitude corresponds to 1% of the maximum of the noiseless data. Moreover, to generate our datasets, $\boldsymbol{s}_1$, we utilize the forward projection discussed in Section 3, using $M = 25$ laser positions on the wall and $N = 2$ camera positions, and FWHM $= 10\, ps$. In addition to the stopping criterion described in the previous subsection, we also set the maximum number of iterations to 40.

5.1 Scenario 1: Two isotropic spheres

Consider the scenario depicted in Fig. 2, where the laser and camera are located in L (red '$\times$') and R (black '$\times$'), respectively. For this simulation, there are two isotropic spheres of $1\ cm$ in diameter. The noisy simulated dataset is shown in Fig. 3 for a single camera position. The considered voxel space is delimited by the blue lines in Fig. 2, a $100\ cm \times 100\ cm$ square in the (x, z)-plane.

In Fig. 4, we provide the unfiltered and filtered backprojection results[3], at the first iteration of the iterative algorithms (corresponding to the original backprojection, namely the non-iterative method). More specifically, Fig. 4a shows the "raw" results, while Fig. 4b depicts the output of a Laplacian filter applied to the previous result (the spatial high pass filter discussed previously). The AEB backprojection results at the last iteration are depicted in Figs. 5a-5b. For this scenario, the AEB algorithm was stopped after 5 iterations. As expected, for the MEB algorithm, the first iteration yields the same results shown in Figs. 4a- 4b. In Fig. 5c-5d, we provide the MEB backprojection results at the 3rd and last iteration.

Comparing the unfiltered results of the AEB (Fig. 5a) and MEB (Fig. 5c) to the non-iterative backprojection (Fig. 4a), we can see that both algorithms provide improvements over a single backprojection result. The AEB and MEB unfiltered results are similar, whereas -for the filtered results- the AEB provides a smaller target recovery area than the MEB.

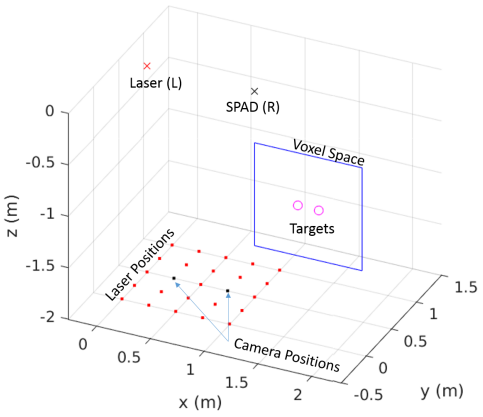


Fig. 2. Section 5.1 simulated scenario. The laser and camera origins are indicated with a red and black '$\times$', respectively. The wall is parallel to the (x, y)-plane; the laser and camera positions are indicated by red and black '.', respectively. The targets, two isotropic spheres, are represented by 'o' in magenta, whereas the voxel space is shown using a continuous blue line.

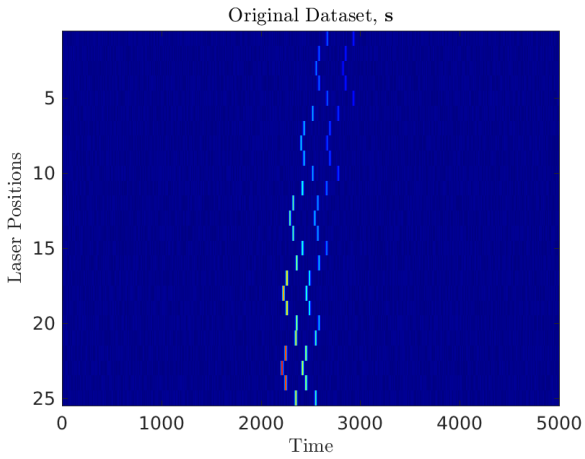
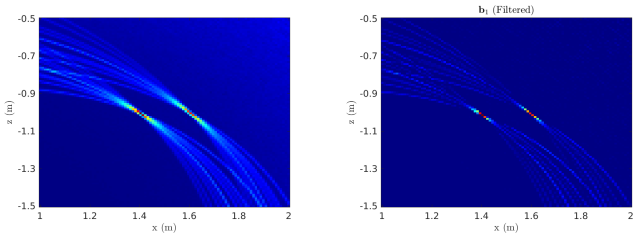


Fig. 3. Simulated dataset, $\boldsymbol{s}_1$, corresponding to the scenario described in Section 5.1.

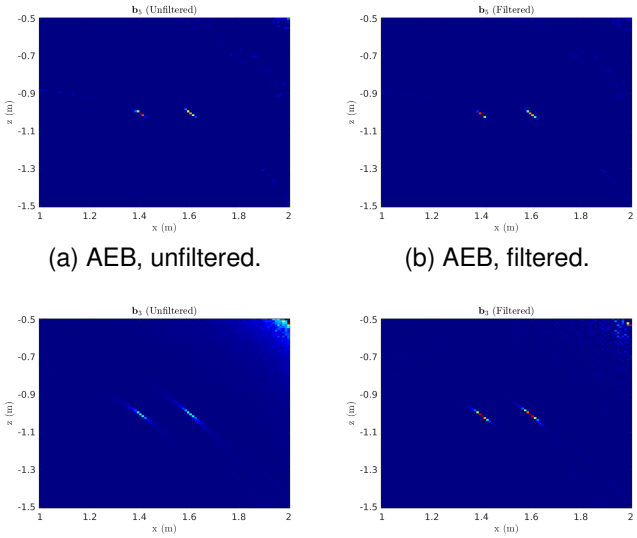
2. In Computational Tomography, each row corresponds to a *sinogram* of the scene, that is a projection of the scene.

3. These and the following results have been normalized and visualized using the standard jet(256) MATLAB colormap.



(a) Non-iterative, unfiltered. (b) Non-iterative, filtered.

Fig. 4. Non-iterative backprojection results, considering the scenario of Section 5.1.



(a) AEB, unfiltered. (b) AEB, filtered.

(c) MEB, unfiltered. (d) MEB, filtered.

Fig. 5. AEB and MEB backprojection results, considering the scenario of Section 5.1.

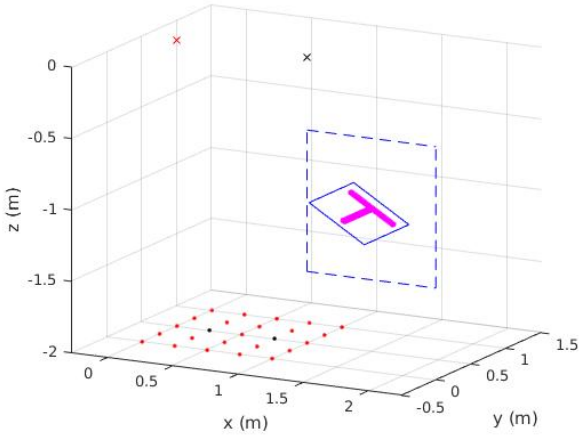


Fig. 6. Simulated scenario in Section 5.2. The laser and camera origin are indicated with a red and black '$\times$', respectively. The wall is parallel the (x, y)-plane and the laser and camera positions are shown using red and black '$\cdot$', respectively. The "T"-shaped target is represented by '$\cdot$' in magenta, whereas the two considered voxel spaces are shown using continuous and dotted blue lines.

5.2 Scenario 2: "T"-shaped target

For this subsection, the scenario is depicted in Fig. 6, where we assume that a laser and a camera are located in L (red '$\times$') and R (black '$\times$'), respectively. We consider a $38\ cm$ by $42\ cm$ "T"-shaped target placed $105\ cm$ from the wall. The simulated dataset, for a single camera position, is shown in Fig. 7.

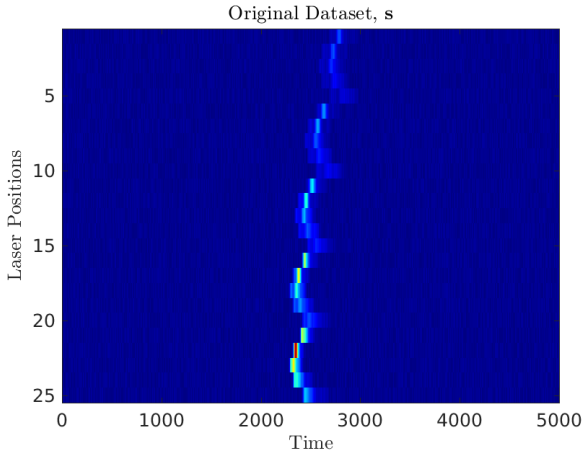


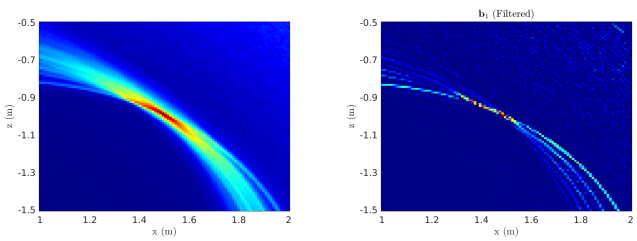
Fig. 7. Simulated dataset, s_1, corresponding to the scenario described in 5.2 for a camera position.

Now, consider the $100\ cm \times 100\ cm$ square on the (x, z)-plane in Fig. 6 (refer to the space delimited by the dotted blue lines) as our voxel space. This slice encompasses the crossbar of the "T"-shaped target and our goal is to use the iterative methods to recover this target feature.

We provide the unfiltered and filtered backprojection results at the first iteration (corresponding to the original backprojection) in Fig. 8. For this particular realization, the AEB algorithm stopped after 12 iterations and we show the results in Figs. 9a-9b. The MEB algorithm stopped after 4 iterations and the unfiltered and filtered backprojection results are shown in Figs. 9c-9d. As it can be seen, the AEB performs better than the MEB and, again, the noise has affected the results of the Laplacian filter (cf. Fig. 9b and 9d).

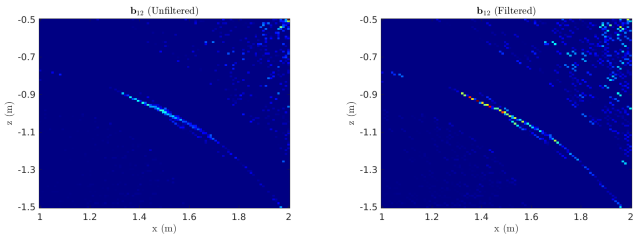
Let us now consider the diagonal slice shown in Fig. 6 (refer to the continuous blue line), sliced parallel to the "T"-shaped target to consider its entire geometry (rather than just its crossbar, as in the previous case). If we compare the unfiltered results (Fig. 10a, Fig. 11a and Fig. 11c), it is possible to see that both the AEB and MEB have provided an improvement over the non-iterative result. The former was stopped after 16 iterations, while the latter converged after 5 iterations. The filtered results for the AEB reconstruction (Fig. 11b) are significantly less noisy than the non-iterative one (Fig. 10b), whereas the filtered MEB reconstruction is poor (Fig. 11d). Neither AEB nor MEB results can be further improved with the use of a Laplacian filter.

Despite their superior performance in the presence of noise, the error backprojection methods are difficult to use with real data. This is due to their sensitivity to bias in the data or in the light transport model. To illustrate this, in Figs. 12-13, we plot the results of the algorithms when the dataset is embedded in additive uniform noise (as above) plus ambient noise (simulated as a constant value added to each point in the dataset). As it can be seen from Figs. 13a-13d, the error backprojection algorithms are

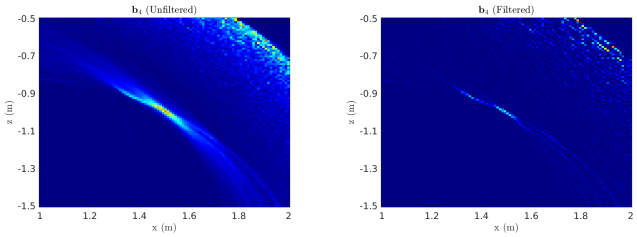


(a) Non-iterative, unfiltered. (b) Non-iterative, filtered.

Fig. 8. Non-iterative backprojection results, considering the scenario described Section 5.2 and the slice on the (x, z) plane, shown in Fig. 6 (dotted blue line).

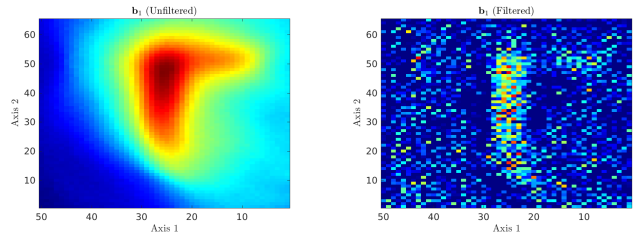


(a) AEB, unfiltered. (b) AEB, filtered.



(c) MEB, unfiltered. (d) MEB, filtered.

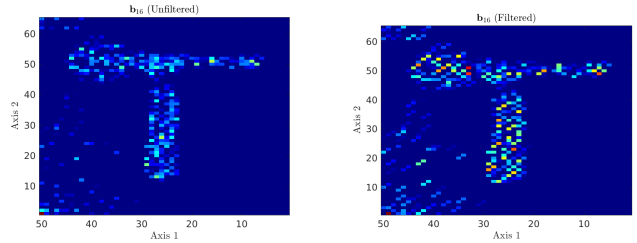
Fig. 9. AEB and MEB backprojection results, considering the scenario described Section 5.2 and the slice on the (x, z) plane, shown in Fig. 6 (dotted blue line).



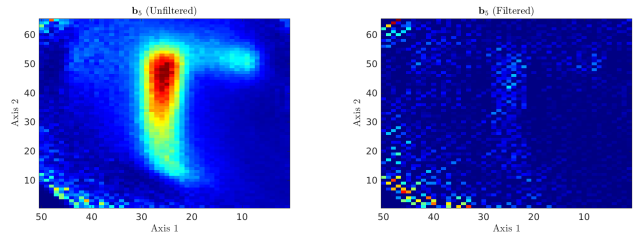
(a) Non-iterative, unfiltered. (b) Non-iterative, filtered.

Fig. 10. Non-iterative backprojection results, considering the scenario described Section 5.2 and the diagonal slice shown in Fig. 6 (continuous blue line).

able to provide some improvement over the non-iterative one. In this study case, the forward projection model is not accounting for the ambient background; namely there is a mismatch between the *real* forward projection model (dataset and ambient noise) and the one *assumed* by the AEB (MEB) algorithm (dataset only), leading to the results shown in Fig. 13. Since the addition of a constant background only contributes backprojection errors at low spatial frequencies,

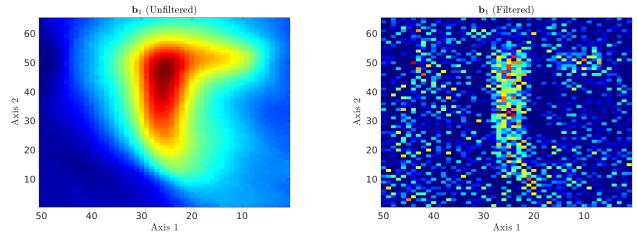


(a) AEB, unfiltered. (b) AEB, filtered.



(c) MEB, unfiltered. (d) MEB, filtered.

Fig. 11. AEB and MEB backprojection results, considering the scenario described Section 5.2 and the diagonal slice shown in Fig. 6 (continuous blue line).



(a) Non-iterative, unfiltered. (b) Non-iterative, filtered.

Fig. 12. Non-iterative backprojection results, considering the scenario described Section 5.2, where -in this case- the dataset is embedded in uniform noise plus ambient noise. We reconstruct the diagonal slice shown in Fig. 6 (continuous blue line).

the Laplacian filter acts as a high pass filter. The resulting filtered backprojection remains essentially unaffected by the background.

6 Application to Real Data

To evaluate the performance of the proposed algorithms using real data, we obtained the dataset from [15], collected using the scenario shown in Fig. 14.

6.1 Capture Setup

The light source is an ultra-fast laser (Amplitude Systems Mikan), which is able to generate 250 fs-long pulses, with a Pulse Repetition Frequency of 55 MHz and wavelength 1030 nm. The photon detector is a Single-Photon Avalanche Detector (SPAD), with 20 μs active area and its time jitter is in the order of 30 ps. The detector is focused on a 1 cm^2 spot covering on the wall. The diameter of the lens is 2.54 cm and its focal length is 2.54 cm.

Through the use of galvanometer-actuated mirrors (indicated with L in Fig. 14), the light source sequentially strikes

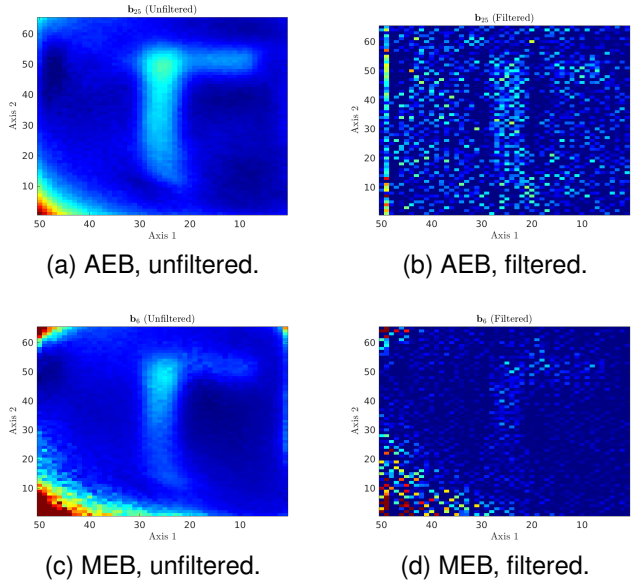


Fig. 13. AEB and MEB backprojection results, considering the scenario described Section 5.2, where -in this case- the dataset is embedded in uniform noise plus ambient noise. We reconstruct the diagonal slice shown in Fig. 6 (continuous blue line).

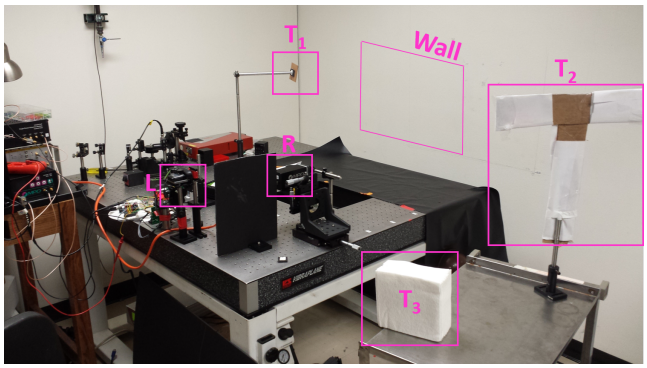


Fig. 14. Experimental setup that was used in [15]. T_1, T_2 and T_3 correspond to the three considered targets, as explained in Section 6.1. The laser and camera position are located in L and R, respectively.

a set of predefined spots on the wall. Considering one laser spot, part of the light is first scattered throughout the scene, then hits the target, reflects back to the wall and, if it hits the wall within the patch the SPAD is focused on, is captured by the SPAD (marked as R in Fig. 14). As it can be seen from Fig. 14, the considered targets are a small square patch (T_1), a 38 $cm \times 41$ cm letter "T" (T_2) and a square patch (T_3), all of them made of white material.

The captured data was taken considering $M = 185$ laser spots on the wall, $N = 1$ camera positions, an exposure time of $T_{et} = 1$ s and total capture time $T_{ct} = 300$ s. The captured dataset is shown in Fig. 15.

6.2 Results

For the reconstruction, we consider a 200 $cm \times 90$ $cm \times 40$ cm volume, which wraps around the location of the targets. Considering a slice near the center of the reconstruction volume, in Fig. 16, we show the non-iterative, unfiltered

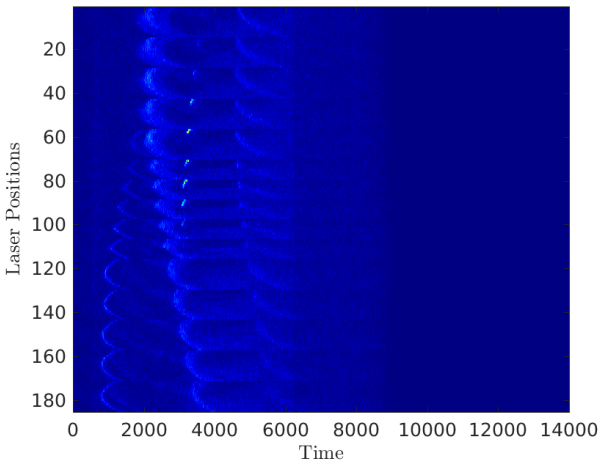
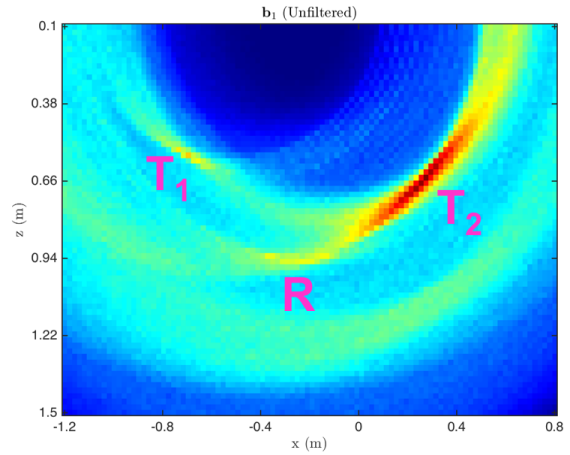


Fig. 15. To test the iterative algorithms, we have used the experimental dataset that was captured in this scenario, which is available from [15].

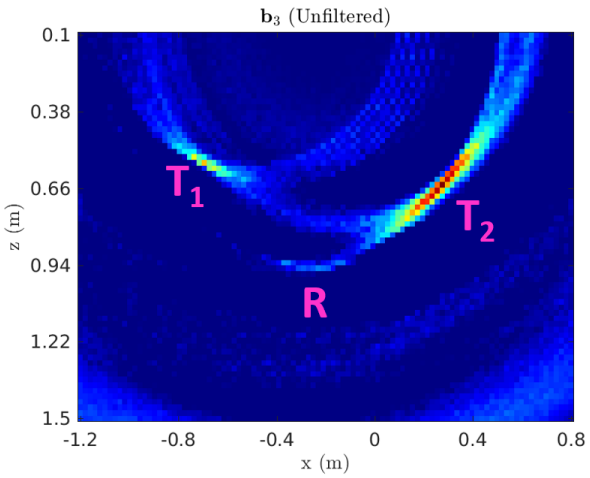
backprojection result and the AEB and MEB unfiltered results (at the $i = 3$-th iteration). It is possible to see that the AEB has improved over the non-iterative algorithm; specifically, the AEB is able to determine the presence of the targets and reduce part of the unwanted signals. The MEB is able recover the targets as well, however there seems to be less noise, with respect to the AEB (and the non-iterative backprojection). Note that the big patch, T_3, is located below the cross section shown in the figure and is therefore not supposed to appear in the reconstruction slice. In Fig. 17, we show the filtered results, where the MEB provides a less noisy result than the original backprojection and the AEB. Although there is an improvement in the unfiltered results, the high frequencies are minimally affected by the iterative methods. This is due to the fact that the data used has a non-uniform background. This background light is coming from the laser pulse; it has been identified in [15] and is most likely due to light from a previous pulse that is still present in the room. The best way to remove this background is to lower the repetition rate of the laser. Unfortunately this is not possible on our current hardware.

Ref. [15] also provides a 3D rendering of the entire volume, achieved by applying the Laplacian filter, a global thresholding algorithm to the entire volume and, finally, plotting the 3D volume with a rendering software. In Fig. 18, we show the entire reconstructed volume for the non-iterative backprojection (Fig. 18a), and the reconstructed volume for the AEB (Fig. 18b) and MEB (Fig. 18c) at the 3rd iteration, using Chimera [26] as our rendering software. It can be seen that the AEB is able to provide a similar reconstruction of the letter "T" (T_2), better reconstruction of the patch T_3, whereas the MEB algorithm is able to decrease the size of the camera (R) and improve the reconstruction of T_1 and T_3.

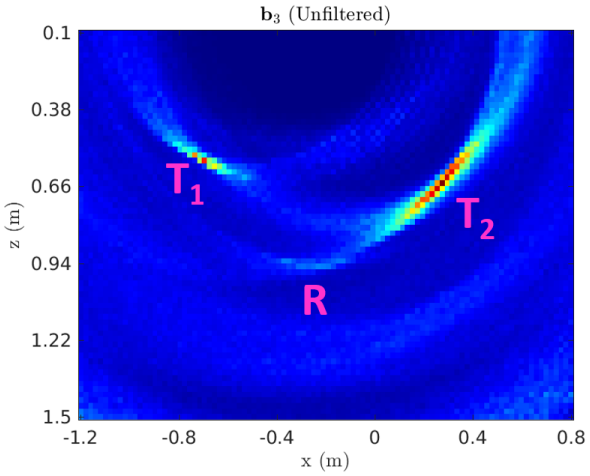
Both the filtered results in Fig. 17 and the volume reconstructions in Figs. 18b-18c show little to no improvement w.r.t. to the non-iterative algorithms. However, it is important to note that the AEB and MEB algorithm work under the assumption that the considered forward model corresponds to the true forward model. However, the noise



(a) Original backprojection.

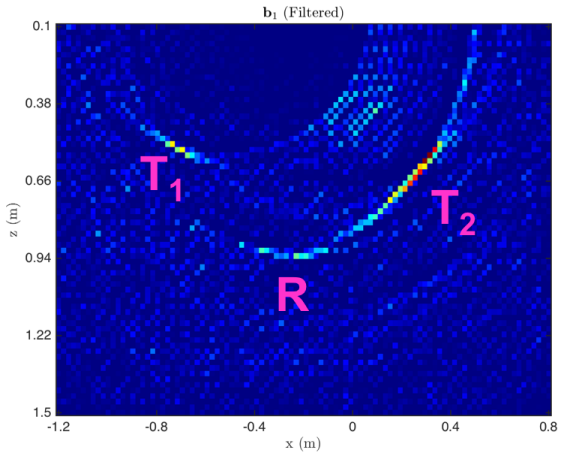


(b) AEB last iteration.

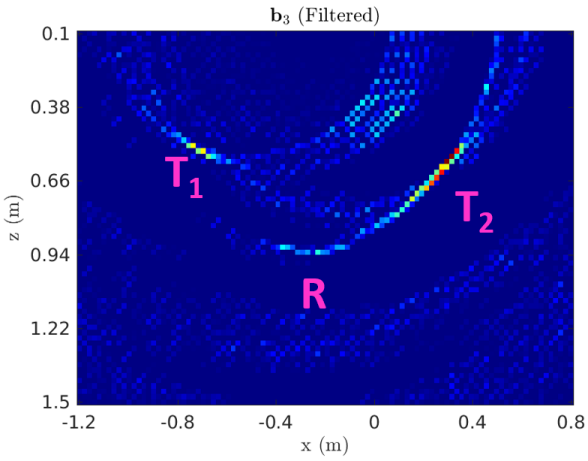


(c) MEB last iteration.

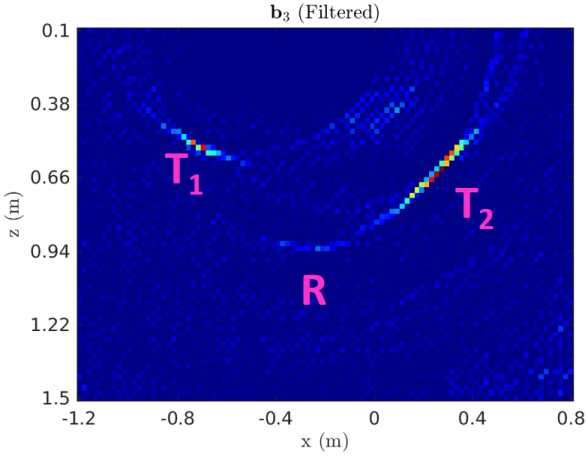
Fig. 16. Unfiltered backprojection results.



(a) Original backprojection.



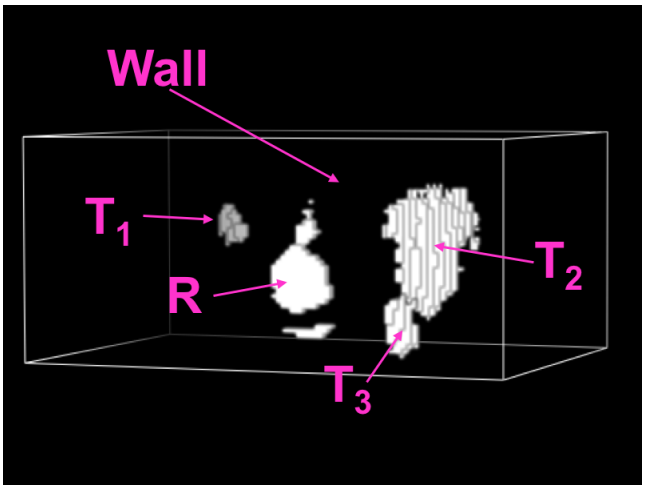
(b) AEB last iteration.



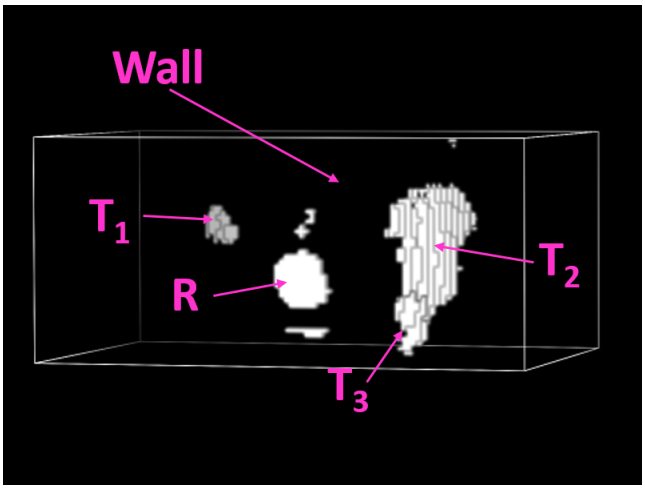
(c) MEB last iteration.

Fig. 17. Filtered backprojection results.

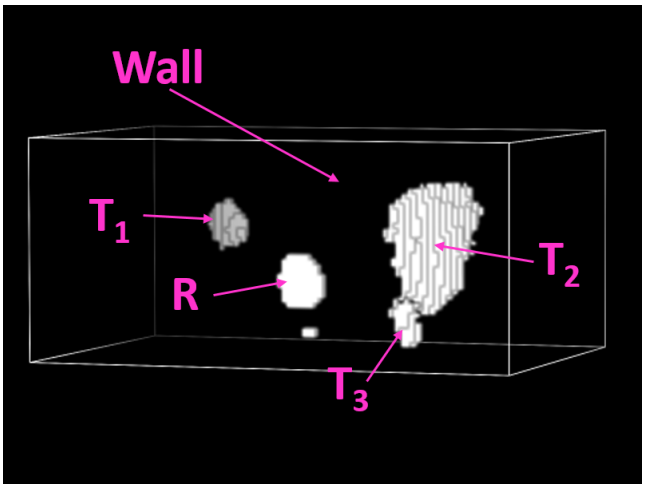
in the experimental dataset is mainly due to scattering from previous pulses (reflected from objects outside of the considered voxel space) as well as ambient light, which can be modeled as low frequency and non-uniform noise. This is a type of systemic noise and is not currently considered by our forward model. We have shown in Fig. 12 and 13 that our approaches are particularly sensitive to errors in the light transport model and data describing the laser and camera positions, $p_1, \ldots, p_M$, and $q_1, \ldots, q_N$, in the visible scene.



(a) Non-iterative algorithm.



(b) AEB algorithm.



(c) MEB algorithm.

Fig. 18. Reconstruction volume comparison between the original backprojection and the proposed iterative algorithms, after 3 iterations. The white lines represent the considered volume for our reconstructions.

7 Conclusions

Non-line-of-sight imaging has attracted a lot of interest thanks to advancements in computer vision and inverse light studies. Several recovery algorithms have been developed to date. In this paper we considered the backprojection algorithm and provided two possible iterative variations, which we called additive error backprojection (AEB) and multiplicative error backprojection (MEB) algorithm. We described their functional implementation (which is memory and time efficient), as well as noted similarities with the algebraic reconstruction technique (ART).

Results show that the algorithms have improved over the unfiltered, non-iterative one proposed in [6], although the error backprojection algorithms are sensitive to errors in the assumed forward projection models. Future work should focus on acquiring real data using a laser with a lower repetition rate, and doing background subtraction (before applying the AEB or MEB), to decrease the errors in the light transport model.

References

[1] A. Kadambi, H. Zhao, B. Shi, and R. Raskar, "Occluded imaging with time-of-flight sensors," *ACM Trans. Graph.*, vol. 35, no. 2, pp. 15:1–15:12, Mar. 2016. [Online]. Available: http://doi.acm.org/10.1145/2836164

[2] O. Gupta *et al.*, "Reconstruction of hidden 3D shapes using diffuse reflections," *Opt. Express*, vol. 20, no. 17, pp. 19 096–19 108, Aug 2012.

[3] F. Heide *et al.*, "Low-budget transient imaging using photonic mixer devices," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 45:1–45:10, Jul. 2013.

[4] ——, "Diffuse mirrors: 3D reconstruction from diffuse indirect illumination using inexpensive time-of-flight sensors," in *2014 IEEE Conf. Comput. Vision Pattern Recognition*, June 2014, pp. 3222–3229.

[5] ——, "Doppler time-of-flight imaging," in *ACM SIGGRAPH 2015 Emerging Tech.*, ser. SIGGRAPH '15. ACM, 2015, pp. 9:1–9:1. [Online]. Available: http://doi.acm.org/10.1145/2782782.2792497

[6] A. Velten *et al.*, "Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging," *Nat Commun*, vol. 3, p. 745, 03 2012. [Online]. Available: http://dx.doi.org/10.1038/ncomms1747

[7] A. Kirmani *et al.*, "Looking around the corner using ultrafast transient imaging," *Int. J. Comput. Vision*, vol. 95, no. 1, pp. 13–28, 2011. [Online]. Available: http://dx.doi.org/10.1007/s11263-011-0470-y

[8] S. M. Seitz, Y. Matsushita, and K. N. Kutulakos, "A theory of inverse light transport," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1440–1447.

[9] O. Steinvall, M. Elmqvist, and H. Larsson, "See around the corner using active imaging," in *SPIE Security and Defence*. International Society for Optics and Photonics, 2011, pp. 818 605–818 605.

[10] G. Gariepy, F. Tonolini, R. Henderson, J. Leach, and D. Faccio, "Detection and tracking of moving objects hidden from view," *Nature Photonics*, 2015.

[11] S. K. Nayar, K. Ikeuchi, and T. Kanade, "Shape from interreflections," *International Journal of Computer Vision*, vol. 6, no. 3, pp. 173–195, 1991.

[12] R. Ramamoorthi and P. Hanrahan, "A signal-processing framework for inverse rendering," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 117–128.

[13] R. Ramamoorthi, "A signal-processing framework for forward and inverse rendering," Ph.D. dissertation, Stanford University, 2002.

[14] J. Klein, C. Peters, J. Martín, M. Laurenzis, and M. B. Hullin, "Tracking objects outside the line of sight using 2d intensity images," *Scientific Reports*, vol. 6, 2016.

[15] M. Buttafava *et al.*, "Non-line-of-sight imaging using a time-gated single photon avalanche diode," *Opt. Express*, vol. 23, no. 16, pp. 20 997–21 011, Aug 2015. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-23-16-20997

[16] A. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*. IEEE Press, 1998.

[17] A. Katsevich, "An improved exact filtered backprojection algorithm for spiral computed tomography," *Advances in Applied Mathematics*, vol. 32, no. 4, pp. 681–697, 2004.
[18] R. Gordon, R. Bender, and G. T. Herman, "Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and x-ray photography," *J. Theoretical Biology*, vol. 29, no. 3, pp. 471 – 481, 1970.
[19] P. Gilbert, "Iterative methods for the three-dimensional reconstruction of an object from projections," *J. Theoretical Biology*, vol. 36, no. 1, pp. 105 – 117, 1972.
[20] R. Gordon, "A tutorial on ART (algebraic reconstruction techniques)," *IEEE Trans. Nuclear Sci.*, vol. 21, no. 3, pp. 78–93, June 1974.
[21] A. H. Andersen and A. C. Kak, "Simultaneous algebraic reconstruction technique (SART): A superior implementation of the ART algorithm," *Ultrasonic Imaging*, vol. 6, no. 1, pp. 81–94, 1984.
[22] H. Guan and R. Gordon, "Computed tomography using algebraic reconstruction techniques (arts) with different projection access schemes: a comparison study under practical situations," *Physics in medicine and biology*, vol. 41, no. 9, p. 1727, 1996.
[23] S. Kaczmarz, "Approximate solution of systems of linear equations," *International Journal of Control*, vol. 57, no. 6, pp. 1269–1271, 1993.
[24] L. Dai and T. B. Schön, "On the exponential convergence of the Kaczmarz algorithm," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1571–1574, 2015.
[25] G. Cimmino, "Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari," *La Ricerca Scientifica*, vol. 16, no. 9, pp. 326–333, 1938 [In Italian].
[26] E. Pettersen *et al.*, "UCSF Chimera – A visualization system for exploratory research and analysis," *J. Comput. Chem.*, vol. 25, pp. 1605–1612, 2004.