DAMMER: Direct Adaptive Multi-resolution MEsh Reconstruction from X-ray measurements

Jannes Merckx[1], Arnold J. den Dekker[1], Jan Sijbers[1], and Jan De Beenhouwer[1]
[1]*imec-Visionlab, University of Antwerp, Universiteitsplein 1, 2610 Antwerp, Belgium*

***Abstract*—X-ray computed tomography (XCT) reconstructs a scanned object using measured projection data, with the object typically represented on a voxel grid during the reconstruction process. However, since material interfaces typically do not align with voxel boundaries, a voxel representation inherently suffers from partial volume effects. This paper presents DAMMER: a method that reconstructs a multi-resolution triangle mesh to represent the attenuation values of a piecewise homogeneous object, often encountered in industrial CT, based on X-ray projection data of this object. DAMMER progressively reconstructs this mesh to match the object. For this, different homogeneous segments are created based on an agglomerative hierarchical clustering procedure, which targets a compact object description by optimizing a weighted sum of projection difference and number of edges between different segments. These segments are progressively optimized to match the homogeneous materials in the object. Simulation and real data experiments show that DAMMER generates significantly more accurate reconstructions compared to pixel grid reconstruction methods, outperforming conventional voxel-based methods in capturing the true geometry of complex material boundaries.**

***Index Terms*—volume mesh, tomographic reconstruction, mesh optimization.**

I. Introduction

CONVENTIONAL X-ray Computed Tomography (XCT) reconstructs a representation of a scanned sample on a regular cubic voxel grid, where the grey value in each voxel is proportional to the (effective) linear attenuation coefficient. However, the faces of these voxels generally do not align with real material interfaces, so that voxels around interfaces often contain more than one material, resulting in partial volume effects (PVE) [1], [2]. PVE significantly compromise the accuracy of interface localization in (industrial) X-ray methods [3]. Moreover, when the scanned sample consists of piecewise homogeneous materials, a regular voxel description is inefficient, as it has redundant volume elements in large homogeneous areas.

Several methods have been proposed to reduce PVE. The most straightforward solution is to reconstruct the object on a high resolution voxel grid. However, this incurs increased memory allocation and computational complexity of the reconstruction. An alternative approach to address PVE is the use of octrees (3D), or quadtrees (2D) [4]. In this approach, the voxel size is locally adapted to generate an efficient multi-resolution representation of the sample [5]–[7]. However, since the orthogonal faces and edges of these representations typically do not align with real material interfaces, PVE still occur.

Another approach to represent a sample consisting of homogeneous materials is approximating interfaces by watertight polygonal surface meshes. These meshes consist of interconnected polygons, most commonly triangles, that collectively approximate the geometric shape of surfaces. Surface mesh representations require storage of vertex positions, connectivity lists, and the attenuation values corresponding to the volumes encapsulated by the surface meshes. Since vertex positions are not bound to grid coordinates, meshes allow a more precise representation of the materials in the sample. One limitation is that a fixed number of interfaces is assumed, which must be either known in advance or accurately extracted from a voxel reconstruction. This extraction is typically performed using the marching cubes algorithm, which exhibits ambiguity problems in certain configurations. Although methods such as signed distance function-based extraction [9] have been proposed to address these ambiguities, surface mesh extraction still relies on a high-resolution voxel reconstructions.

For applications such as metrology, the surface mesh is typically optimized to fit interfaces with a desired level of accuracy. Some research has explored adapting surface meshes to X-ray measurements [8]. However, these models remain limited by the number of meshes extracted from voxel reconstructions. As a result, they require a high-resolution voxel reconstruction to detect unexpected pores or voids, and a reliable way to distinguish such features from noise.

An alternative, in 2D, is extracting interfaces from a CT reconstruction on a triangle mesh, for which a fixed point algorithm with the addition of a Total Variation term was recently proposed [10]. Adaptive triangle mesh representations were also studied for CT by iteratively changing the number of vertices on interface curves extracted by thresholding a level-set representation based on approximate attenuation bins [12]. More recently, Koo et al. proposed DALM, a mesh-based method for accurate 2D reconstruction and segmentation of homogeneous objects directly from projection data [13], assuming the number of materials in the sample is known a-priori. In DALM, an attenuation-labeled mesh (ALM) is deformed to align interface curves with object boundaries. In this deformation procedure, the vertex positions on each interface curve are optimized based on the attenuation values of the domain encapsulated by the curve and the domain outside the curve. However, DALM is inherently limited in handling vertices that neighbor more than two domains with different attenuation values, making it unsuitable for modeling intersecting curves. Moreover, DALM requires a high-resolution full field-of-view reconstruction to extract an initial ALM that correctly identifies all objects.

Existing adaptive mesh reconstruction methods [14]–[17],

primarily focus on generating multi-resolution reconstruction meshes rather than explicitly modeling material interfaces. These methods often incorporate heuristic stopping criteria and typically refine only in one direction (either coarse-to-fine or fine-to-coarse). To improve on these previous works, we introduce Direct Adaptive Multi-resolution MEsh Reconstruction (DAMMER), a 2D triangle mesh-based reconstruction method that adaptively optimizes mesh vertex positions, connectivity, and triangle attenuation values. This approach yields accurate reconstructions of piecewise homogeneous objects commonly encountered in industrial CT on a multi-resolution mesh with high vertex density near interfaces and low density in homogeneous regions, using non-heuristic stopping criteria. DAMMER captures complex interfaces through a segmentation procedure that does not require prior knowledge of the number of materials. All mesh updates are performed locally, eliminating the need for global remeshing, thereby improving efficiency by limiting the need for repeated reconstructions. We evaluated DAMMER under a variety of simulated industrial CT conditions, including noisy projection data, and constrained projection geometries, demonstrating its robustness and effectiveness. We also tested DAMMER on an experimental fan-beam scan. A GitHub-repository with the code can be found at [18].

II. Materials and methods

DAMMER reconstructs attenuation values of a scanned sample on a progressively optimized triangular mesh. An overview of the key steps in the DAMMER algorithm is provided in Algorithm 1, and an overview of its parameters can be found in Table I in Appendix A. The reconstruction process begins with a coarse, regular mesh, defined by a list of vertex coordinates $\boldsymbol{X}$ and a connectivity list $\boldsymbol{T}$ that specifies how the vertices are connected. Starting from a reconstruction on a coarse regular mesh, DAMMER alternates reconstruction steps with mesh optimization steps, in which the mesh is adapted to the (local) content of the reconstructed object, up to a user-defined resolution δ. For the reconstruction steps, we implement the Simultaneous Iterative Reconstruction Technique (SIRT) [19]. First, optimization of attenuation values using SIRT is performed iteratively, interleaved with mesh refinement steps. This method introduces smaller triangles around interfaces. Once the stopping criterion for the mesh refinement method has been satisfied, edges are collapsed to create larger triangles in homogeneous area's. After meeting the stopping criterion for edge collapse, optimization of attenuation values continues iteratively with vertex displacement steps, aligning triangle edges to material interfaces. Analogous to finite element modeling [20], DAMMER aims at close to equilateral triangles for faster computational convergence and a more accurate reconstruction. The quality of the poorest-shaped triangle in the mesh, measured by the maximum circumradius-to-shortest-edge ratio (q_{max}), can be set by the user. In addition to refinement and edge collapse steps, the mesh topology is further optimized, during vertex displacement, through edge flips, where certain edges are reconnected to different vertices to improve overall mesh quality. Enhancing mesh quality through edge flips reduces the need for more computationally expensive refinement steps, making the process more efficient. The optimized mesh reduces PVE, while still providing a compact representation of the sample. The SIRT reconstruction method and the methods for mesh refinement, edge collapse, vertex displacement, and edge flip are described in Sections II-A, II-B, II-C, II-D, and II-D4, respectively.

Algorithm 1 DAMMER

$\boldsymbol{A}$ = System Matrix($\boldsymbol{X}, \boldsymbol{T}$)
$\boldsymbol{\mu}, p_1$ = SIRT($\boldsymbol{b}$, $\boldsymbol{0}$, $\boldsymbol{A}$, N_{it}, 0)
$\boldsymbol{X}, \boldsymbol{T}, \boldsymbol{A}, \boldsymbol{\mu}$ = Refinement($\boldsymbol{X}$, $\boldsymbol{T}$, $\boldsymbol{b}$, $\boldsymbol{\mu}$, $\boldsymbol{A}$, δ, κ_{ref}, S_{thr}, p_1)
$\boldsymbol{X}, \boldsymbol{T}, \boldsymbol{\mu}$ = Edge collapse($\boldsymbol{X}$, $\boldsymbol{T}$,$\boldsymbol{\mu}$, κ_{col}, 0, M_{thr}, q_{max})
$\boldsymbol{A}$ = System Matrix($\boldsymbol{X}, \boldsymbol{T}$)
$\kappa_{\text{seg}} = ||\boldsymbol{A}\boldsymbol{\mu} - \boldsymbol{b}||_2^2/(2N_e)$
$\boldsymbol{\mu}$, N_R = Segmentation($\boldsymbol{b}$, $\boldsymbol{\mu}$, $\boldsymbol{A}$, κ_{seg})
while $N_R < N_{R,\text{thr}}$ **do**
    $\kappa_{\text{seg}} \leftarrow 2\kappa_{\text{seg}}$
    $\boldsymbol{\mu}$, N_R = Segmentation($\boldsymbol{b}$, $\boldsymbol{\mu}$, $\boldsymbol{A}$, κ_{seg})
$\kappa_{\text{seg},0} = \kappa_{\text{seg}}$
for N_{dis} **do**
    $\boldsymbol{X}$ = Displacement($\boldsymbol{X}$, $\boldsymbol{T}$, $\boldsymbol{b}$, $\boldsymbol{\mu}$, $\boldsymbol{A}$, κ_{opt}, N_{BFGS})
    $\boldsymbol{X}, \boldsymbol{T}, \boldsymbol{\mu}$ = Edge collapse($\boldsymbol{X}$, $\boldsymbol{T}$, $\boldsymbol{\mu}$, d_{col}, M_{thr}, q_{max})
    $\boldsymbol{X}, \boldsymbol{T}, \boldsymbol{\mu}$= Edge flip($\boldsymbol{X}$, $\boldsymbol{T}$, $\boldsymbol{\mu}$)
    $\boldsymbol{A}$ = System Matrix($\boldsymbol{X}, \boldsymbol{T}$)
    $\boldsymbol{X}, \boldsymbol{T}, \boldsymbol{A}, \boldsymbol{\mu}$ = Refinement($\boldsymbol{X}$, $\boldsymbol{T}$, $\boldsymbol{b}$, $\boldsymbol{\mu}$, $\boldsymbol{A}$, δ, κ_{ref}, S_{thr}, ∞)
    $\boldsymbol{\mu}$, _ = SIRT($\boldsymbol{b}$, $\boldsymbol{\mu}$, $\boldsymbol{A}$, N_{it}, 0)
    $\boldsymbol{\mu}$, N_R = Segmentation($\boldsymbol{b}$, $\boldsymbol{\mu}$, $\boldsymbol{A}$, κ_{seg})

A. SIRT reconstruction on the triangle mesh

Attenuation values reconstructed on the triangles of the mesh are denoted by $\boldsymbol{\mu} = (\mu_m) \in \mathbb{R}^M$, with M the total number of triangles. These attenuation values are reconstructed from projection data acquired by a detector with P detector elements, rotated over N projection angles. Let $\boldsymbol{b} = (b_i) \in \mathbb{R}^{N \cdot P}$ denote the log-transformed, flat field corrected projection data, and $\boldsymbol{A} = (a_{i,m}) \in \mathbb{R}^{(N \cdot P) \times M}$ the (sparse) system matrix, where $a_{i,m}$ represents the contribution of triangle m to projection value b_i. This contribution corresponds to the intersection length of a ray corresponding with measurement i through triangle m. The attenuation coefficients $\boldsymbol{\mu}$ are estimated from the projection data $\boldsymbol{b}$ by solving the following optimization problem:

$$\hat{\boldsymbol{\mu}} = \arg\min_{\boldsymbol{\mu}} ||\boldsymbol{A}\boldsymbol{\mu} - \boldsymbol{b}||_2, \tag{1}$$

where $||\cdot||_2$ refers to the L_2 norm. SIRT is an iterative algorithm that finds an approximate solution of Eq. (1) by minimizing the objective function $||\boldsymbol{A}\boldsymbol{\mu} - \boldsymbol{b}||_R^2 = (\boldsymbol{A}\boldsymbol{\mu} - \boldsymbol{b})^T \boldsymbol{R} (\boldsymbol{A}\boldsymbol{\mu} - \boldsymbol{b})$, where $\boldsymbol{R} \in \mathbb{R}^{(N \cdot P) \times (N \cdot P)}$ is a diagonal matrix containing the inverse row sums of $\boldsymbol{A}$: $r_{i,i} = 1/\sum_j a_{i,j}$. The SIRT update step is given by

$$\boldsymbol{\mu}^{n_{\text{it}}+1} = \boldsymbol{\mu}^{n_{\text{it}}} + \boldsymbol{C}\boldsymbol{A}^T \boldsymbol{R} (\boldsymbol{b} - \boldsymbol{A}\boldsymbol{\mu}^{n_{\text{it}}}), \tag{2}$$

where $\boldsymbol{C} \in \mathbb{R}^{M \times M}$ is a diagonal matrix containing the inverse column sums of $\boldsymbol{A}$: $c_{m,m} = 1/\sum_i a_{i,m}$.
In this paper, SIRT is applied with two stopping criteria: 1) $n_{\text{it}} > N_{\text{it}}$, with N_{it} a user defined maximum number of

iterations, and 2) $||\boldsymbol{A}^T\boldsymbol{R}(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{\mu}^{n_{\text{it}-1}})||_2^2 < p_1$, with p_1 a user defined threshold on the squared L_2-norm of the update of the attenuation values $\boldsymbol{\mu}$ in iteration n_{it}. The SIRT reconstruction is stopped if one of these criteria is satisfied. Note that the second criterion can be disabled by setting p_1 equal to zero. The pseudocode of the SIRT reconstruction method is shown in Algorithm 2.

Algorithm 2 SIRT

function SIRT($\boldsymbol{b}$, $\boldsymbol{\mu_0}$, $\boldsymbol{A}$, N_{it}, p_1)
  $\boldsymbol{C} = (c_{m,m}) \in \mathbb{R}^{M\times M} : c_{m,m} = 1/\sum_i a_{i,m}$
  $\boldsymbol{R} = (r_{i,i}) \in \mathbb{R}^{(N\cdot P)\times(N\cdot P)} : r_{i,i} = 1/\sum_j a_{i,j}$
  $n_{\text{it}} = 0$
  $\boldsymbol{\mu}^{n_{\text{it}}} = \boldsymbol{\mu_0}$
  while $n_{\text{it}} < N_{\text{it}}$ **and** $||\boldsymbol{A}^T\boldsymbol{R}(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{\mu}^{n_{\text{it}}})||_2^2 > p_1$ **do**
    $\boldsymbol{\mu}^{n_{\text{it}}} = \boldsymbol{\mu}^{n_{\text{it}}-1} + \boldsymbol{C}\boldsymbol{A}^T\boldsymbol{R}\left(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{\mu}^{n_{\text{it}}-1}\right)$
    $n_{\text{it}} \leftarrow n_{\text{it}} + 1$,
  return n_{it}, $||\boldsymbol{A}^T\boldsymbol{R}(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{\mu}^{n_{\text{it}}})||_2^2$

B. Mesh refinement method

Algorithm 3 Refinement

function REFINEMENT($\boldsymbol{X}$, $\boldsymbol{T}$, $\boldsymbol{b}$, $\boldsymbol{\mu}$, $\boldsymbol{A}$, δ, κ_{ref}, S_{thr}, p_1)
  $S_{\text{thr}} = \Delta\mu_{\max} + \kappa_{\text{ref}}\mathbb{E}\left[\frac{r}{l}\right]$
  $S_m = \max_{u\in\boldsymbol{n}(m)} (|\mu_m - \mu_u|) + \kappa_{\text{ref}}\frac{r_m}{l_m}$
  while $\min\{S_m : r_m > \delta\} > S_{\text{thr}}$ **do**
    Split triangle $t = \max_m\{S_m : r_m > \delta\}$
    Update $\boldsymbol{A}$
    $\boldsymbol{\mu}$, _ = SIRT($\boldsymbol{b}$, $\boldsymbol{\mu}$, $\boldsymbol{A}$, 50, p_1)
    $S_m = \max_{u\in\boldsymbol{n}(m)} |\mu_m - \mu_u| + \kappa_{\text{ref}}\frac{r_m}{l_m}$
  return $\boldsymbol{X}$, $\boldsymbol{T}$, $\boldsymbol{A}$, $\boldsymbol{\mu}$

Starting from a (coarse) regular mesh, DAMMER targets a mesh of variable triangle density that accurately represents interfaces up to a desired resolution and at the same time warrants a compact object representation. To this end, triangles near interfaces are progressively split into smaller triangles up to a user-defined resolution δ, expressed in terms of triangle circumradius.

The mesh refinement method starts with a Delaunay triangulated [21] mesh, and initial attenuation values reconstructed using 300 iterations of the SIRT algorithm, assigned to the triangles of this mesh. The pseudocode of the refinement algorithm is presented in Algorithm 3. Each mesh refinement iteration starts by updating, for each triangle m, a so-called splitting weight S_m. This splitting weight, which balances local difference in reconstructed attenuation and triangle equiangularity, is given by

$$S_m = \max_{u\in\boldsymbol{n}(m)} |\mu_m - \mu_u| + \kappa_{\text{ref}}\frac{r_m}{l_m} \quad . \tag{3}$$

The first term on the right hand side of Eq. (3) measures whether triangle m is located at an interface by calculating the maximum absolute difference between the reconstructed attenuation μ_m of triangle m and the reconstructed attenuation of triangles in the list $\boldsymbol{n}(m)$ of triangles neighbouring triangle m. The second term consists of the circumradius (r_m)-to-shortest-edge (l_m) ratio of triangle m. This term measures the triangle's equiangularity. Finally, κ_{ref} is a user-defined regularisation parameter that controls the importance of triangle equiangularity.

To exclude triangles in homogeneous areas from splitting, we derive a threshold value S_{thr} for S_m. To this end, we first estimate a statistical upper bound on the change in reconstructed attenuation $\boldsymbol{\mu}$ due to noise. Assuming that the log-transformed projection data is disturbed by zero mean Gaussian noise with standard deviation σ, a statistical upper limit $\Delta\mu_{\max}$ (with an approximately 95% confidence level) of the change in reconstructed attenuation per triangle due to noise is given by

$$\Delta\mu_{\max} = \frac{\sqrt{N\cdot P}}{\sqrt{M}}\frac{||\boldsymbol{\mu}||_2}{||\boldsymbol{b}||_2}\cdot\xi\left(\boldsymbol{A}\right)\cdot 2\sigma, \tag{4}$$

where $\xi(\boldsymbol{A}) = ||\boldsymbol{A}||\cdot||\boldsymbol{A}^{-1}||$ is the condition number of the system matrix [22]. To exclude triangles in the starting mesh in homogeneous areas from splitting with a 95% confidence, the threshold value S_{thr} for S_m is set to

$$S_{\text{thr}} = \Delta\mu_{\max} + \kappa_{\text{ref}}\mathbb{E}\left[\frac{r}{l}\right]. \tag{5}$$

The first term of Eq. (5) represents the statistical upper limit $\Delta\mu_{\max}$ found by Eq. (4), while the term $\mathbb{E}\left[\frac{r}{l}\right]$ denotes the circumradius-to-shortest-edge ratio of the triangles constituting the starting mesh. Triangles with a score $S_m > S_{\text{thr}}$ are assumed unlikely to be located in a homogeneous region and therefore qualify for splitting. Finally, in this paper, κ_{ref} is set to

$$\kappa_{\text{ref}} = \frac{\Delta\mu_{\max}}{q_{\max}} \quad . \tag{6}$$

With this choice, the splitting weight assigns equal importance to the statistical maximum deviation from the background in homogeneous regions and the maximal desired circumradius-to-shortest-edge ratio in the mesh.

After computing all splitting weights, the triangle t that maximizes this weight (assuming that $S_t > S_{\text{thr}}$) is split by applying Chew's algorithm [23]. This algorithm adds a vertex at the center of the circumscribed circle of triangle t, denoted by coordinates $\boldsymbol{c_t}$, to generate a high quality mesh representation. An exception occurs if $\boldsymbol{c_t}$ is not situated within triangle t or one of its neighbours, implying the presence of at least one additional triangle encroached by the circumscribed circle of t. In this situation, the center of the first edge crossed by a line between $\boldsymbol{c_t}$ and a vertex of t is added to the triangulation. All vertices encompassed by the diametrical circle of this edge are then removed. The required variables for Chew's algorithm, as well as which vertices are added in different situations, are shown in Fig. 1. Following this update, a Delaunay triangulation in the neighborhood of the split triangle is made to update the triangles of the mesh. The attenuation values of the updated triangles are an average of the attenuation values of the original triangles, weighted with their intersection area with the updated triangles. Updating the system matrix then consists of updating the elements

corresponding to a limited number of triangles and rays. After this, a SIRT reconstruction is performed with the updated system matrix. The stopping criterion of this reconstruction is set to $||\boldsymbol{A}^T\boldsymbol{R}(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{\mu})||_2^2 < p_1$, where p_1 represents the squared L_2-norm of the difference between the last and forelast attenuation update of the SIRT reconstruction used to initialize the mesh refinement step. Following each refinement, the splitting weights S_m are updated and the triangle with the highest updated S_m is split using Chew's algorithm. This process - splitting the triangle that maximises Eq. (3) - is repeated until no more triangles with circumradius greater than δ have a splitting weight exceeding S_{thr}.

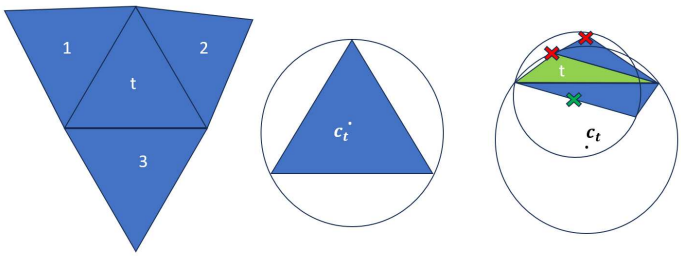


Fig. 1: An illustration of Chew's algorithm. Left: Triangle t and surrounding edges. Middle: center circumscribed circle. Right: Vertices added and removed if $\boldsymbol{c_t}$ is not inside t or one of its neighbours.

C. Edge collapse method

Algorithm 4 Edge collapse

function EDGE COLLAPSE($\boldsymbol{X}$, $\boldsymbol{T}$, $\boldsymbol{\mu}$, κ_{col}, σ_1, d_{col}, M_{thr}, q_{max})
  $\boldsymbol{E}$ = Edges
  $M_{\text{thr}} = -\sigma_1^2 + 2\kappa_{\text{col}}\mathbb{E}\left[\frac{r}{l}\right]$
  for $\boldsymbol{e}$ in $\boldsymbol{E}$ **do**
    $M_e = -\tilde{s}_e + \kappa_{\text{col}}\tilde{p}_e$
    if $\max\{M_e : r_t < \delta\} > M_{\text{thr}}$ **or** $||\boldsymbol{X}(\boldsymbol{e}(1)) - \boldsymbol{X}(\boldsymbol{e}(2))|| < d_{\text{col}}$ **then**
      Calculate $\frac{r}{l}$ of updated triangles
      if $\max\{\frac{r}{l}\} < q_{\text{max}}$ **then**
        Collapse edge
  return $\boldsymbol{X}$, $\boldsymbol{T}$, $\boldsymbol{\mu}$

Following the addition of vertices in interface regions up to a resolution δ, DAMMER proceeds by collapsing edges of the mesh in homogeneous regions. Collapsing an edge f involves merging the vertices $(p, q) = f$ (the vertices connected to f): this means p and q are removed and the center of f is added to the triangulation [24]. Therefore, an edge collapse reduces the number of vertices by 1. A schematic overview of the edge-collapsing procedure is shown in Fig. 2, and the corresponding pseudocode is presented in Algorithm 4. To select the edges to be collapsed, a selection criterion is proposed that combines two distinct objectives: i) reduce the triangle density in homogeneous areas of reconstructed attenuation, ii) promote triangle equiangularity. For the first objective, the local variation in reconstructed attenuation at each vertex v in the mesh is defined as

$$s_v = \frac{1}{U}\sum_{u=1}^{U}\left(\mu_u - \frac{1}{U}\sum_{j=1}^{U}\mu_j\right)^2, \tag{7}$$

where u and j index the U triangles that surround the vertex v. Collapsing an edge $\boldsymbol{e}$ when it has a low $\tilde{s}_e$ value contributes to the first objective described above. Let l_e, r_t and r_u denote the length of edge e and the circumradii of the triangles t and u connected by this edge, respectively. Then, $\tilde{p}_e$ is defined as $\tilde{p}_e = (r_t + r_u)/l_e$. Collapsing $\boldsymbol{e}$ when it has a high $\tilde{p}_e$ value contributes to the second objective described above, as collapsing short edges that connect triangles with comparatively high circumradii promotes triangle equiangularity. Balancing both contributions, the so-called collapsing weight is defined as

$$M_e = -\tilde{s}_e + \kappa_{\text{col}}\tilde{p}_e, \tag{8}$$

where the regularization parameter κ_{col} Eq. (8) is set to

$$\kappa_{\text{col}} = \frac{\sum_{e=1}^{N_e}\tilde{s}_e}{\sum_{e=1}^{N_e}\tilde{p}_e}. \tag{9}$$

All edges with an S_e value exceeding a certain threshold M_{thr} are selected for collapsing. This threshold collapsing weight is set to $M_{\text{thr}} = -\sigma_1^2 + 2\kappa_{\text{col}}\mathbb{E}\left[\frac{r}{l}\right]$, where σ_1^2 refers to the user-defined maximal variance in reconstructed attenuation tolerated in homogeneous regions. Our choice for the threshold weight is guided by the fact that in our starting mesh (introduced in Section III), non-diagonal edges without local attenuation variation (i.e., for which $\tilde{s}_e = 0$) have a collapsing weight equal to $M_e = 2\kappa_{\text{col}}\mathbb{E}\left[\frac{r}{l}\right]$. Therefore, our choice of M_{thr} reflects our target to collapse zero variation and near zero variation edges. Besides collapsing edges with a threshold collapsing weight exceeding M_{thr}, we also collapse edges with a length shorter than a minimal length d_{col}. In this paper, we choose to equate d_{col} to the input resolution parameter δ. All edges with a collapsing weight below the threshold are examined in order of increasing weight. For each edge, it is checked whether collapsing it would cause the maximum circumradius-to-shortest-edge ratio to remain smaller than the desired upper bound q_{max}. If this condition is met, the edge is collapsed. This edge-collapsing procedure is repeated until no edges with $M_e < M_{\text{thr}}$ are present in the mesh. Pseudocode of the Edge collapse method is shown in ALgorithm 4.

D. Vertex displacement method

The previous sections described how the mesh refinement and edge collapse methods return a multi-resolution mesh with resolution δ and maximal circumradius-to-shortest-edge ratio q_{max}, with attenuation values assigned to all triangles. This mesh serves as a suitable initialization for the vertex displacement method. This method aims at segmenting the mesh into homogeneous segments and optimizing the positions of vertices along shared edges between neighboring

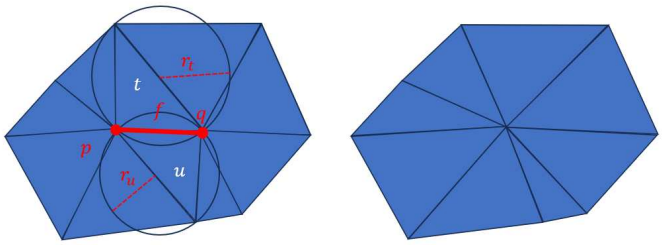


Fig. 2: Left: The edge f, with vertices p and q connected to f, and triangles t and u connected by f. Right: Updated mesh after collapsing f.

segments so that they better align with material interfaces. To achieve this, the method iteratively performs the following six steps over a predefined number of iterations k_{dis}: 1) segmentation, 2) optimization of interface vertex positions, 3) edge flips, 4) additional edge collapses, 5) additional mesh refinement, and 6) optimization of attenuation values with SIRT. Each of these steps is described below. In what follows, the common edges of neighboring segments are referred to as interface edges, whereas the vertices of these interface edges are referred to as interface vertices.

Algorithm 5 Segmentation

function SEGMENTATION($\boldsymbol{b}$, $\boldsymbol{\mu}$, $\boldsymbol{A}$, κ_{seg})
  $\boldsymbol{E}$ = edges ranked by attenuation difference
  $\tilde{\boldsymbol{\mu}} = \boldsymbol{\mu}$
  $N_R = ||\boldsymbol{\mu}||$
  for $\boldsymbol{e}$ in $\boldsymbol{E}$ **do**
    $\boldsymbol{t}_1, \boldsymbol{t}_1$ = triangles connected different sides of $\boldsymbol{e}$
    $\tilde{\boldsymbol{\mu}}(\boldsymbol{t}_1, \boldsymbol{t}_1) = \langle \boldsymbol{\mu}(\boldsymbol{t}_1, \boldsymbol{t}_1) \rangle$
    $\tilde{N}_e$ = number of edges connecting $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$
    if $\frac{1}{2}||\boldsymbol{b} - \boldsymbol{A}\tilde{\boldsymbol{\mu}}||_2^2 - \frac{1}{2}||\boldsymbol{b} - \boldsymbol{A}\boldsymbol{\mu}||_2^2 - \kappa_{\text{seg}}\tilde{N}_e < 0$ **then**
      $N_R = N_R - 1$
      $\boldsymbol{\mu} \leftarrow \tilde{\boldsymbol{\mu}}$
    else
      $\tilde{\boldsymbol{\mu}} \leftarrow \boldsymbol{\mu}$
  return $\boldsymbol{\mu}$, N_R

1) Segmentation: Our segmentation procedure is based on the agglomerative hierarchical clustering procedure [25]. The pseudocode of the segmentation procedure is shown in Algorithm 5. Starting from an initialization where each triangle is an individual segment, neighboring triangles with comparable attenuation values are aggregated to form larger segments. Mathematically, we aggregate the attenuation values of the M triangles of the mesh to minimize the following objective function:

$$\sum_{i=1}^{N \cdot P} \left(b_i - \sum_{m=1}^{M} a_{i,m} \mu_m \right)^2 + \kappa_{\text{seg}} \#\left(\text{interface edges}\right). \quad (10)$$

The first term in this function is the so-called data consistency term (also known as the projection difference), which quantifies the difference between the forward projection of the mesh representation of the object and the measured projection data. The second term scales with the number of interface edges, which is maximal at the start of the segmentation procedure and reduces with the formation of multi-triangle patches. The variable κ_{seg} is a regularization parameter that controls the trade-off between data consistency and the requirement of a compact object representation. Before any aggregations, all interface edges (which initially correspond to all edges in the mesh) are ranked in order of increasing absolute difference between the attenuation values of the edge-neighboring triangles. Next, progressing along this list, all interface edges are consecutively subjected to an aggregation check. An edge passes this check if merging its edge-neighboring triangles into a single homogeneous segment decreases the objective function Eq. (10). Aggregating two triangles entails equalizing their attenuation values to a weighted average, with weights given by the relative areas of the triangles or, in the case of triangles already part of a homogeneous segment, by the relative areas of their respective segments. In case of a positive check, the aggregation is performed accordingly and the interface edge ranking is updated before subjecting the then highest ranked interface edge to the aggregation check. With each aggregation, the number of homogeneous segments reduces by one. The aggregation procedure is continued until no more interface edges pass the aggregation check. Depending on the application, the segmentation parameter κ_{seg} can be set by different methods. In the experiments in this paper, where we assume piecewise homogeneous objects with high contrasts, κ_{seg} is calibrated as follows. First, κ_{seg} is initialized to a fraction of the ratio between the data consistency term and the total number of edges. Next, the aggregation procedure described above is applied, after which κ_{seg} is multiplied by 2 and the aggregation procedure is started over again. This procedure of aggregation and increasing κ_{seg} is repeated iteratively until the total number of homogeneous segments decreases below a user defined maximum number $N_{R,\text{thr}}$. From that moment on, the regularization parameter κ_{seg} is no longer increased, but fixed to its then-current value, coined $\kappa_{\text{seg},0}$. At the end of each segmentation, single triangle segments are aggregated with the neighbouring segment with the lowest attenuation difference compared to the particular single triangle segment.

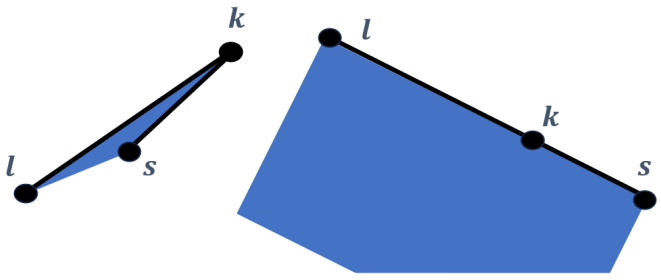


Fig. 3: Left: vertices k, s, and l correspond to a high interface orientation term. Right: vertices k, s, and l correspond to a low interface orientation term.

2) Optimize interface vertex positions: After segmentation, the reconstructed mesh consists of homogeneous segments, separated from each other by interface edges. Let the number

Algorithm 6 Displacement

function DISPLACEMENT($\boldsymbol{X}_{\text{int}}$, $\boldsymbol{b}$, $\boldsymbol{\mu}$, $\boldsymbol{A}$, κ_{opt}, N_{BFGS})
  Calculate $\boldsymbol{\nabla E_d}$ of Eq. (11).
  Initialize $\boldsymbol{H}^{-1}$
  for N_{BFGS} **do**
    $E_0 = E_d\left(\boldsymbol{X}_{\text{int}}\right)$
    $\nu = 1$
    $\boldsymbol{X}_{\text{int}} \leftarrow \boldsymbol{X}_{\text{int}} - \boldsymbol{H}^{-1}\boldsymbol{\nabla E_d}$
    while $E_d\left(\boldsymbol{X}_{\text{int}}\right) > E_0$ **do**
      $\nu \leftarrow \nu/2$
      $\boldsymbol{X}_{\text{int}} \leftarrow \boldsymbol{X}_{\text{int}} + \nu\boldsymbol{H}^{-1}\boldsymbol{\nabla E_d}$
    Update $\boldsymbol{H}^{-1}$ with the BFGS algorithm
  return $\boldsymbol{X}_{\text{int}}$

of interface vertices be denoted by $N_{v,\text{int}}$ and let the coordinates of the vertices connected to these edges be denoted by $\boldsymbol{X}_{\text{int}} = \{\boldsymbol{x}_k\}_{k=1}^{N_{v,\text{int}}} \in \mathbb{R}^{2\times N_{v,\text{int}}}$. The next step of the vertex displacement method is to reposition the interface vertices to better align the interface edges with the real material interfaces. For this, the so-called interface energy function,

$$E_d = \frac{1}{2}\sum_{i=1}^{N\cdot P}\left(b_i - \sum_{m=1}^{M} a_{i,m}\mu_m\right)^2 + \frac{\kappa_{\text{opt}}}{2}\sum_{k=1}^{N_{v,\text{int}}}\sum_{l,s\in N_G(k),l\neq s}||\boldsymbol{x_l} - \boldsymbol{x_k} + \boldsymbol{x_s} - \boldsymbol{x_k}||_2^2, \tag{11}$$

is optimized with respect to $\boldsymbol{X}_{\text{int}}$. The first term of Eq. (11) is the data consistency term, analogous to Eq. (10). This term decreases as the interface edges align more closely with the real material interfaces. The second term is the interface orientation term, where index k traverses all interface vertices, and the index pair (l, s) runs over all pairs of interface vertices in the topological neighbourhood $N_G(k)$ of vertex k [26]. In other words, l and s run over all combinations of interface vertices connected to k. The contribution of the vertex triplet indexed by (k, l, s) to the second term of Eq. (11) is minimal when the three vertices lie on a straight line. Fig. 3 shows two configurations of three vertices, connected by two edges. One configuration corresponds with a high interface orientation term, while the other corresponds to a low interface orientation term. The interface orientation term is included in Eq. (11) to push interface vertices to constant distances of one another and suppress noise effects. The so-called interface orientation parameter κ_{opt} is calibrated before optimising Eq. (11). In particular, κ_{opt} is calibrated as follows:

$$\kappa_{\text{opt},0} = \frac{\sum_{i=1}^{N\cdot P}\left(b_i - \sum_{m=1}^{M} a_{i,m}\mu_m\right)^2}{\sum_{k=1}^{N_{v,\text{int}}}\sum_{l,s\in N_G(k),l\neq s}||\boldsymbol{x_l} - \boldsymbol{x_k} + \boldsymbol{x_s} - \boldsymbol{x_k}||_2^2}. \tag{12}$$

Once calibrated, the value of κ_{opt} is kept fixed throughout the current vertex displacement iteration. Numerical optimization of Eq. (11) is carried out using a quasi-Newton method, specifically the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm. Quasi-Newton methods estimate the inverse Hessian matrix of the objective function to compute a numerical search direction for updating the variable $\boldsymbol{X}$, based on the gradient $\boldsymbol{\nabla E_d} \in \mathbb{R}^{2N_{v,\text{int}}}$ of Eq. (11). The search direction is computed by pre-multiplying the gradient with a numerical approximation of the inverse Hessian matrix $\boldsymbol{H} \in \mathbb{R}^{(2N_{v,\text{int}})\times(2N_{v,\text{int}})}$:

$$\boldsymbol{d} = \boldsymbol{H}^{-1}\boldsymbol{\nabla E_d}. \tag{13}$$

In the first iteration, the inverse Hessian is initialized as a diagonal matrix with all diagonal elements set to the inverse norm of $\boldsymbol{\nabla E_d}$. In subsequent iterations, $\boldsymbol{H}^{-1}$ is updated numerically using successive gradient evaluations, following the BFGS update rule [27]. Pseudocode of the vertex displacement method is shown in Algorithm 6.

3) Additional edge collapses: After optimizing the interface vertex positions with the BFGS algorithm, an additional edge collapse step is applied to remove non-interface vertices from homogeneous segments. To eliminate all edges in homogeneous regions, the collapsing weight parameter κ_{col} (see Eq. (8)) is set to 0, ensuring that the threshold M_{thr} equals $-\sigma_1^2$. Additionally, edges with a length shorter than a specified minimum threshold d_{col} are collapsed, irrespective of their collapsing weight. This helps to eliminate potential excessively short edges around interfaces created by displacing vertices. In the experiments in this work, d_{col} was set based on the resolution δ: $d_{\text{col}} = \delta$.

Algorithm 7 Edge flip

function EDGE FLIP($\boldsymbol{X}$, $\boldsymbol{T}$, $\boldsymbol{\mu}$)
  for e in $\boldsymbol{E}$ **do**
    Let v_{e1}, v_{e2} be the vertices connected by e
    if $v_{e1}, v_{e2} \in \boldsymbol{X}_{\text{int}}$ **then**
      Let $\boldsymbol{T}_1$, $\boldsymbol{T}_2$ be the triangles connected by e
      Define $v_{t1} = \{v \in T_1 \;\&\; v \notin \{v_{e1}, v_{e2}\}\}$, and $v_{t2} = \{v \in T_2 \;\&\; v \notin \{v_{e1}, v_{e2}\}\}$
      Define possible new triangles $\tilde{T}_1 = (v_{e1}, v_{t1}, v_{t2})$, $\tilde{T}_2 = (v_{e2}, v_{t1}, v_{t2})$
      if $\max\left(\left(\frac{r}{l}\right)_{\tilde{T}_1,\tilde{T}_2}\right) < \max\left(\left(\frac{r}{l}\right)_{T_1,T_2}\right)$ **then**
        $T_1, T_2 \leftarrow \tilde{T}_1, \tilde{T}_2$
        $\mu_{T_1}, \mu_{T_1} \leftarrow \left(\mu_{T_1} + \mu_{T_1}\right)/2$
  return $\boldsymbol{X}$, $\boldsymbol{T}$, $\boldsymbol{\mu}$

4) Edge flips: Edge-flipping enhances mesh quality by locally modifying connectivity while preserving the triangulation. An edge flip removes a shared edge between two adjacent triangles and replaces it with a new edge connecting the previously unconnected pair of opposite vertices. This operation preserves the overall mesh topology but can significantly improve element quality. The algorithm targets edges connecting two interface vertices that are not themselves interface edges. For each such edge, the algorithm evaluates whether performing a flip would reduce the maximum circumradius-to-shortest-edge ratio among the affected triangles. If this condition is met, the edge is flipped. An example of this procedure is shown in Fig. 4. The pseudocode of the method is shown in Algorithm 7.

5) Additional mesh refinement: Next, the splitting procedure described in Section II-B is applied to split long triangles

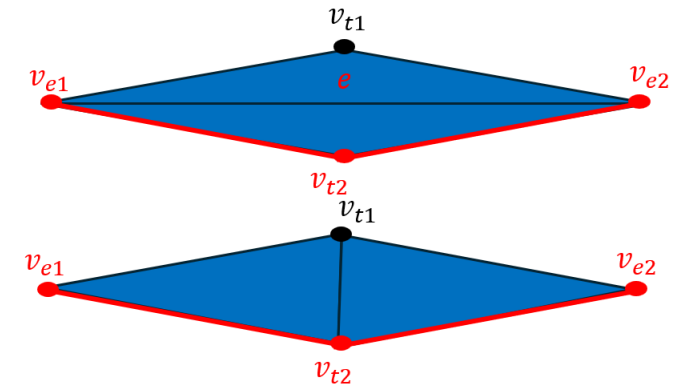


Fig. 4: Triangles before (up) and after (down) edge collapse. Red shows interface edges and vertices, black shows other edges and vertices.

and triangles around interfaces. These additional refinement steps are typically limited due to the approximate correctness of the starting condition.

6) Additional SIRT reconstruction update: Each iteration of the vertex displacement method concludes with an additional SIRT reconstruction. The stopping criterion for this reconstruction is the same number of SIRT iterations N_{it} used for the reconstruction on the starting mesh.

III. Experiments

A. Simulation experiments

First, the performance of DAMMER was evaluated by simulation experiments based on parallel beam projection data generated from 2000×2000 pixel phantom images (shown in Fig. 5b, 5c, and 5d). The size of the square reconstruction domain was set to 1. The forward projections of these phantoms were simulated using the ASTRA toolbox [28]. In particular, the phantoms were projected on a 1D detector consisting of 1000 detector elements of 2 times the phantom pixel size. In each simulation experiment, the starting mesh for DAMMER was chosen equal to the regular mesh shown in Fig. 5a. In this starting mesh, all 1922 triangles have a circumradius of $\frac{\sqrt{2}}{64}$ and a circumradius-to-shortest-edge ratio equal to to $\frac{\sqrt{2}}{2}$.
The number of SIRT iterations to initialize the attenuation values on the starting mesh was set to $N_{\text{it}} = 500$. The tolerated variance in homogeneous areas of the edge collapse method, σ_1^2, was set to 0.01^2. The maximum number of homogeneous segments $N_{R,\text{thr}}$ inputted in the segmentation method to fix κ_{seg} in Eq. (10) was set to one tenth of the number of triangles M found after applying the edge collapse method. Finally, the number of iterations in the vertex displacement method was set to $k_{\text{dis}} = 10$. Finally, the resolution parameter δ was set to 0.01 for the first two phantoms, and 0.005 for the Shepp-Logan phantom (SL phantom), which requires distinguishing the ellipses on the bottom of the image, separated by a minimum distance of 0.0075.

In each experiment, the DAMMER reconstruction was evaluated and compared to a square pixel grid representation. For each phantom, a grid was chosen that corresponds with the maximum memory occupied by DAMMER representations during optimization of the mesh to this phantom. This memory was computed as the sum of the memory needed to store vertex positions, attenuation values, and the connectivity list. This choice resulted in (rounded up to the nearest tenfold) 110×110, 160×160 and 260×260 square pixel grid reconstructions for Phantom 1, Phantom 2, and the SL phantom, respectively. When comparing DAMMER and pixel grid reconstructions, the effect of the segmentation was removed by segmenting the pixels to minimize Eq. (10), just like the triangles of DAMMER (cf. Section II-D1). However, to prevent any bias towards DAMMER, for the pixel grid reconstructions the regularization parameter κ_{seg} in Eq. (10) was not set according to the procedure described in Section II-D1, but tuned to minimize the Mean Squared Error (MSE):

$$\text{MSE} = \frac{\sum_{m=1}^{M} \sum_{j=1}^{J^2} [\boldsymbol{r}_m \cap \boldsymbol{p}_j] \left(\mu_m - \bar{\mu}_j\right)^2}{\sum_{m=1}^{M} [\boldsymbol{r}_m]}. \tag{14}$$

Here, M represents the number of reconstruction elements (either triangles or pixels), and J represents the phantom grid dimension. $\boldsymbol{r}_m$ represents the m^{th} polygon of the reconstruction domain, $\boldsymbol{p}_j$ denotes the j^{th} pixel of the phantom image, and $\boldsymbol{r}_m \cap \boldsymbol{p}_j$ indicates the intersection between these two polygons. The area operator is denoted $[\cdot]$, and outputs the area of its argument. Finally, $\boldsymbol{\mu}$ and $\bar{\boldsymbol{\mu}}$ refer to the reconstructed and phantom attenuation values, respectively. Thus, the MSE quantifies the discretized, integrated squared difference between the reconstructed and true attenuation distributions. In what follows, SIRT pixel grid representations segmented with the same algorithm as DAMMER, but with κ_{seg} tuned to minimize the MSE, are referred to as clustered SIRT reconstructions.

We also compare DAMMER to Total Variation (TV) regularized pixel grid reconstructions. In these, $\boldsymbol{\mu}$ is estimated by optimising

$$\hat{\boldsymbol{\mu}} = \arg\min_{\boldsymbol{\mu}} \left(||\boldsymbol{A}\boldsymbol{\mu} - \boldsymbol{b}||_2^2 + \kappa_{\text{TV}} V(\boldsymbol{\mu}) \right), \tag{15}$$

with κ_{TV} a regularization parameter and $V(\boldsymbol{\mu})$ the isotropic TV term. For a square grid of size $J \times J$, this term is given by:

$$V(\boldsymbol{\mu}) = \sum_{i=1}^{J} \sum_{j=1}^{J} \Big(\left(\mu_{i,j} - \mu_{i+1,j}\right)^2 + \left(\mu_{i,j} - \mu_{i-1,j}\right)^2 + \left(\mu_{i,j} - \mu_{i,j+1}\right)^2 + \left(\mu_{i,j} - \mu_{i,j-1}\right)^2 + \epsilon_{\text{reg}} \Big)^{1/2}, \tag{16}$$

where ϵ_{reg} is a small value that ensures differentiability [30]. Analogous to the clustered SIRT pixel representations, we varied κ_{TV} to get the best possible Total Variation regularized

representation in terms of the MSE.

Besides the MSE, quantitative evaluation was also performed in terms of the Peak Signal-to-Noise Ratio (PSNR) and the structural similarity index measure (SSIM). The PSNR is a complementary measure which provides a normalized measure of image fidelity:

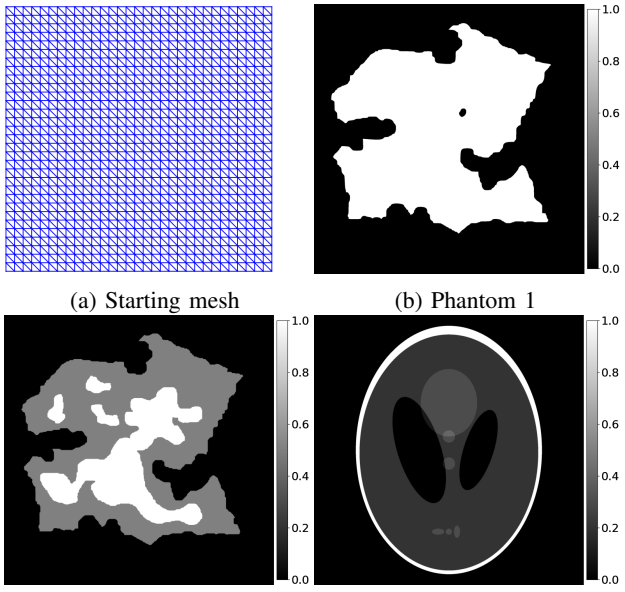
$$\text{PSNR} = 10\log_{10}\left(\frac{\max_j \bar{\mu}_j^2}{\text{MSE}}\right) \tag{17}$$

Finally, the SSIM is defined as:

$$\text{SSIM} = \frac{(2\langle\bar{\boldsymbol{\mu}}\rangle\langle\boldsymbol{\mu}\rangle + c_1)\,(2\sigma_{\bar{\boldsymbol{\mu}}\boldsymbol{\mu}} + c_2)}{(\langle\bar{\boldsymbol{\mu}}\rangle^2 + \langle\boldsymbol{\mu}\rangle^2 + c_1)\left(\sigma_{\boldsymbol{\mu}}^2 + \sigma_{\bar{\boldsymbol{\mu}}}^2 + c_2\right)}, \tag{18}$$

where $\langle\cdot\rangle$ denotes the mean, $\sigma_{\bar{\boldsymbol{\mu}}\boldsymbol{\mu}}$ is the co-variance between $\bar{\boldsymbol{\mu}}$ and $\boldsymbol{\mu}$, and $\sigma_\cdot^2$ refers to the variance. The parameters c_1 and c_2 are, for an image with grey values scaled between 0 and 1, equal to 0.01 and 0.03, respectively [31].

A detailed description of the simulation experiments can be found below.



(a) Starting mesh (b) Phantom 1

(c) Phantom 2 (d) Shepp-Logan phantom

Fig. 5: Starting mesh and geometric phantoms.

1) Noiseless, full angular range projection data: In the first simulation experiment, DAMMER was evaluated on noiseless projection data to assess its intrinsic ability to reduce PVE. For each of the three phantoms introduced in Fig. 7, a dataset of 150 equiangularly distributed projections over the range $[0°, 360°]$ was generated.

For each phantom, we first visually assessed the quality of the multi-resolution mesh representations produced by DAMMER's mesh refinement and edge collapse procedures. Subsequently, the vertex displacement method was applied to the optimized meshes, resulting in the final DAMMER reconstructions. The reconstructions, along with their associated MSE, PSNR, and SSIM metrics, are reported in Section IV-A1.

2) Noisy, full angular range projection data: To evaluate DAMMER's robustness to noise, we repeated the experiments from Section III-A1 with projection data $\tilde{\boldsymbol{b}} = (\tilde{b}_i) \in \mathbb{R}^{N\cdot P}$ perturbed with additive Gaussian white noise. Following the approach in [29], the standard deviation of this Gaussian distribution, divided by the maximum projection measurement, is defined as the noise level. We tested DAMMER on all phantoms for noise levels of 1%. To further assess performance under moderate noise levels, DAMMER was also tested on projection data of Phantom 2 with noise levels of up to 5%. The results of these experiments are presented in Section IV-A2.

3) Limited angular projection data: DAMMER was optimized using different numbers of both noiseless and Gaussian-noised (1% variance) projection data of Phantom 2. In particular, the number of projection angles was gradually reduced from 75 to 30. The resulting MSE values were compared to those of clustered SIRT pixel grid reconstructions and TV regularized reconstructions. The results are provided in Section IV-A3.

4) Limited angular range projection data: We then evaluated DAMMER's performance on 150 projections of Phantom 2 acquired under constrained angular ranges. Specifically, we considered equiangular projections spanning ranges $[0°, 120°]$ and $[0°, 150°]$, both noiseless, and with 1% Gaussian noise. The results of those experiments are presented in Section IV-A4.

5) Effect of resolution parameter δ: Next, we evaluated the effect of the resolution parameter δ, investigating how accurate DAMMER representations can be for infinitesimal resolutions. Results for Phantom 2 are shown in Section IV-A5.

6) Effect of segmentation parameter κ_{seg}: Next, we tested the influence of the segmentation parameter κ_{seg} on the MSE by evaluating reconstructions obtained from projection data of Phantom 2 with different values κ_{seg} (150 projections, 1% noise level). The results are shown in Section IV-A6.

7) Effect of interface orientation term κ_{opt}: As a final simulation experiment on Phantom 2, we studied the effect of varying the interface orientation term κ_{opt} in Eq. (11) from the value $\kappa_{\text{opt},0}$ defined in Eq. (12). In particular, we evaluated DAMMER representations for $\kappa_{\text{opt}} = 0$ and $\kappa_{\text{opt}} = 10\kappa_{\text{opt},0}$. The results of these experiments are shown in Section IV-A7.

B. Sensitivity to small attenuation differences

To study DAMMER's ability to reconstruct small object components, we used a phantom consisting of nested disks with different radii (the "Disk phantom" shown in Fig. 6). The contrast of the smaller disks is 10% compared to the

background, corresponding to an attenuation difference of 0.05. The radii, from bottom to top, are equal to 0.53, 0.35, 0.26, and 0.18 times the circumradius of the triangles of the starting mesh.

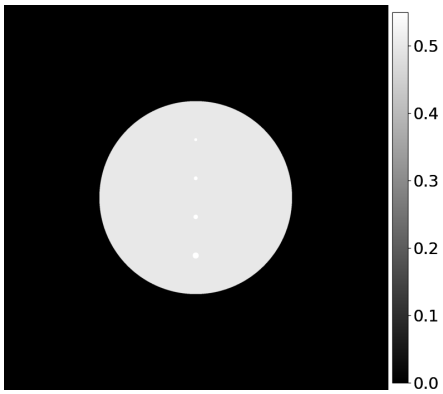
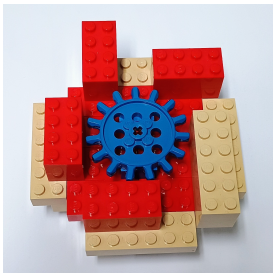


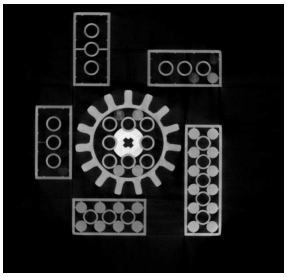
Fig. 6: Disk phantom consisting of smaller disks of varying size encapsulated by a homogeneous disk.

C. Experimental data

We also tested DAMMER on an experimental dataset. In particular, we evaluated DAMMER on fan beam data of a set of lego blocks (shown in Fig. 7a) acquired with the FleXCT scanner located at Visionlab. [32] Specifically, the lego blocks were scanned by an $80keV$ cone-beam, measured on a 2856×456 detector with pixel size 150μm, a Source-to-Object Distance of 50cm, and a Source-to-Detector Distance of 75cm. We tested DAMMER on 360 equiangular central slice projections of this dataset. To match the size of the sample, we set the bounds of the reconstruction domain to $(-10\text{cm}, -10\text{cm})$, and $(10\text{cm}, 10\text{cm})$. Validation of this experiment was performed visually, by comparing it to a high-resolution reconstruction based on 3600 projection angles. This reconstruction image is shown in Fig. 7b.



(a) Lego blocks



(b) Reconstruction

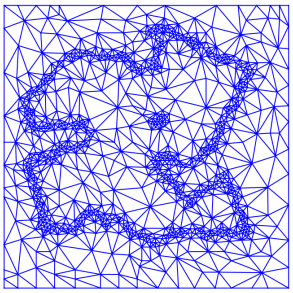
Fig. 7: Lego blocks and a high resolution (central-slice) reconstruction based on 3600 projection angles.

IV. Results

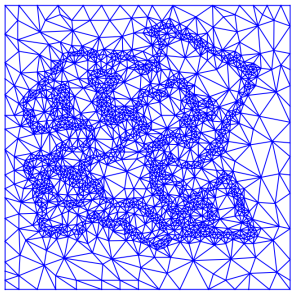
This section presents the results of the experiments outlined in Section III. The reconstructed triangular meshes are visualized using the Matplotlib [33] tripcolor function, which renders triangulations at high resolution for accurate visual interpretation.

A. Simulation data

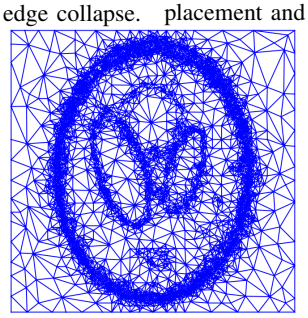
1) Noiseless, full angular range projection data: Fig. 8 displays the multi-resolution meshes acquired by applying DAMMER's mesh refinement and edge collapse methods, described in Section II-B and Section II-C, to noiseless projection data of the three phantoms displayed in Figs. 5b, 5c, and 5d, initialized by a reconstruction on the regular mesh displayed in Fig. 5a. The figures show small triangles around interfaces of the phantoms, and large triangles in homogeneous areas. Fig. 9 compares a SIRT reconstruction of the SL phantom based on 150 noiseless projections on the coarse starting mesh of Fig. 5a to a SIRT reconstruction on the mesh shown in Fig. 8c.



(a) Mesh Phantom 1 after replacement and edge collapse.

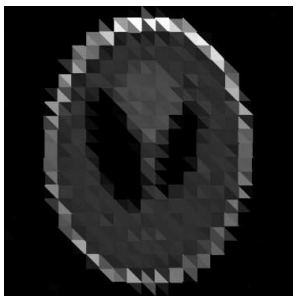


(b) Mesh Phantom 2 after replacement and edge collapse.



(c) Mesh SL-phantom after replacement and edge collapse.

Fig. 8: Triangle reconstruction meshes resulting from applying DAMMER's mesh refinement and edge collapse methods on the starting mesh based on 150 noiseless projections of a) Phantom 1, b) Phantom 2, c) the Shepp-Logan phantom.

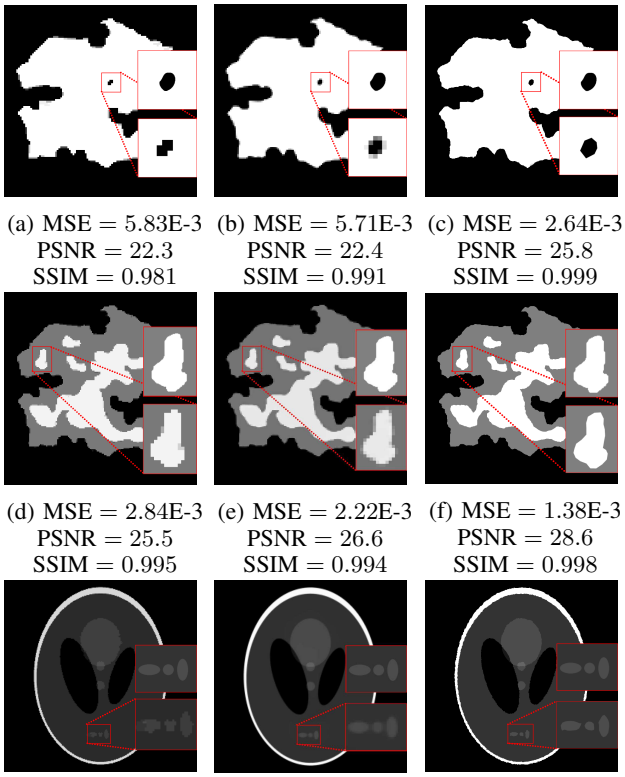


(a) Reconstruction on the starting mesh shown in Fig. 5a.

(b) Reconstruction on the mesh shown in Fig. 8c.

Fig. 9: Reconstruction on the starting mesh and a multi-resolution mesh.

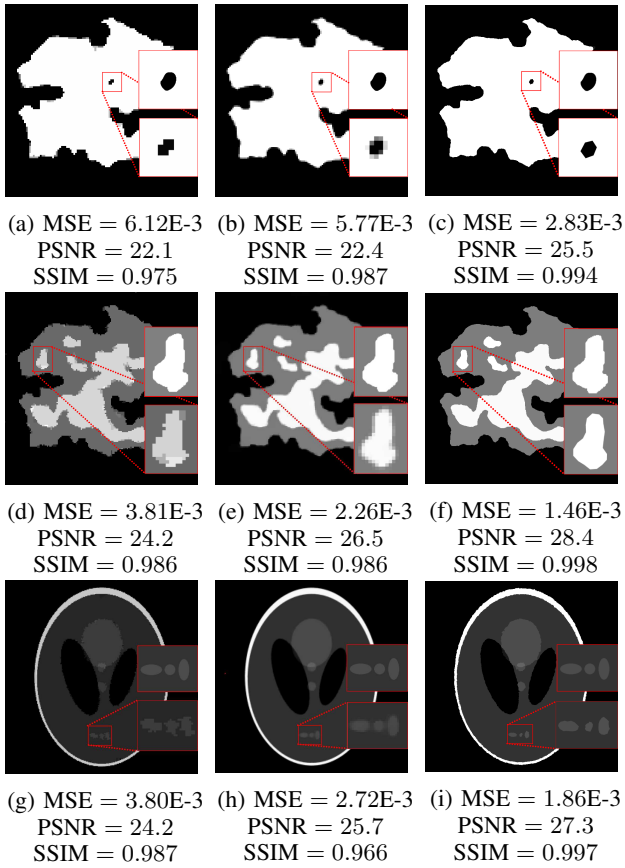
Fig. 10 shows the DAMMER reconstructions (and corresponding MSE, PSNR and SSIM values) for the three phantoms, along with clustered SIRT reconstructions and TV regularised reconstructions on pixel grids of approximately the same memory occupation. The lower insets of the figures zoom in on an ROI while the upper insets show the corresponding groundtruth. The results show that pixel grid reconstructions exhibit artificial, blocky characteristics whereas with DAMMER interfaces are represented with smoother curves.



(a) MSE = 5.83E-3 PSNR = 22.3 SSIM = 0.981
(b) MSE = 5.71E-3 PSNR = 22.4 SSIM = 0.991
(c) MSE = 2.64E-3 PSNR = 25.8 SSIM = 0.999
(d) MSE = 2.84E-3 PSNR = 25.5 SSIM = 0.995
(e) MSE = 2.22E-3 PSNR = 26.6 SSIM = 0.994
(f) MSE = 1.38E-3 PSNR = 28.6 SSIM = 0.998
(g) MSE = 3.73E-3 PSNR = 24.3 SSIM = 0.995
(h) MSE = 2.71E-3 PSNR = 25.7 SSIM = 0.970
(i) MSE = 1.72E-3 PSNR = 27.6 SSIM = 0.996

Fig. 10: Reconstructions from noiseless projection data for Phantom 1 (top row), Phantom 2 (middle row), and the Shepp-Logan phantom (bottom row), along with their corresponding error metrics. The left column displays clustered SIRT pixel reconstructions found by the best possible κ_{seg}. The middle column shows TV regularized reconstructions, found by the best possible κ_{TV}. The right column shows the DAMMER reconstructions. In each image, the lower inset zooms in on a region of interest (ROI), and the upper inset shows the corresponding ground truth.

2) Noisy, full angular range projection data: Fig. 11 shows DAMMER reconstructions based on 150 noisy projections of the three phantoms. Similar to Fig. 10, DAMMER reconstructions are shown next to clustered SIRT and TV regularised reconstructions on a pixel grid of comparable memory occupation, along with the corresponding MSE, PSNR and SSIM values. Similar to the noiseless case, DAMMER was initialized by a reconstruction on the regular mesh displayed in Fig. 5a. For Phantom 2, the MSE of DAMMER representations for increasing noise levels is shown in Fig. 12.



(a) MSE = 6.12E-3 PSNR = 22.1 SSIM = 0.975
(b) MSE = 5.77E-3 PSNR = 22.4 SSIM = 0.987
(c) MSE = 2.83E-3 PSNR = 25.5 SSIM = 0.994
(d) MSE = 3.81E-3 PSNR = 24.2 SSIM = 0.986
(e) MSE = 2.26E-3 PSNR = 26.5 SSIM = 0.986
(f) MSE = 1.46E-3 PSNR = 28.4 SSIM = 0.998
(g) MSE = 3.80E-3 PSNR = 24.2 SSIM = 0.987
(h) MSE = 2.72E-3 PSNR = 25.7 SSIM = 0.966
(i) MSE = 1.86E-3 PSNR = 27.3 SSIM = 0.997

Fig. 11: Reconstructions from noisy (1% noise level) projection data for Phantom 1 (top row), Phantom 2 (middle row), and the Shepp-Logan phantom (bottom row), along with their corresponding error metrics. The left column displays clustered SIRT pixel reconstructions found by the best possible κ_{seg}. The middle column shows TV regularized reconstructions, found by the best possible κ_{TV}. The right column shows the DAMMER reconstructions. In each image, the lower inset zooms in on an ROI, and the upper inset shows the corresponding ground truth.

3) Limited angular projection data: To evaluate DAMMER's performance under limited angular coverage, the number of projections was reduced from the 150 used in the previous experiment. Fig. 13 shows the MSE Eq. (14) of DAMMER and pixel grid reconstructions as a function of the number of equiangular projections (distributed over a full 360° angular range) for Phantom 2. Each figure includes results for both noiseless and noisy projection data. The results show that DAMMER consistently outperforms both pixel grid reconstruction methods.

4) Limited angular range projection data: The top and bottom row of Fig. 14 shows noiseless reconstructions for angular ranges equal to $[0°, 150°]$, and $[0°, 120°]$, where

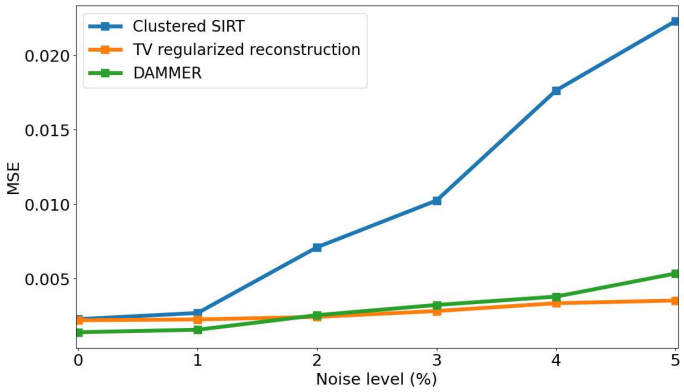


Fig. 12: MSE across different noise levels for DAMMER, clustered SIRT, and TV-regularized pixel reconstructions of Phantom 2. For clustered SIRT and TV, the regularization parameters κ_{seg} and κ_{TV} were tuned to minimize the MSE and set to their optimal values accordingly.

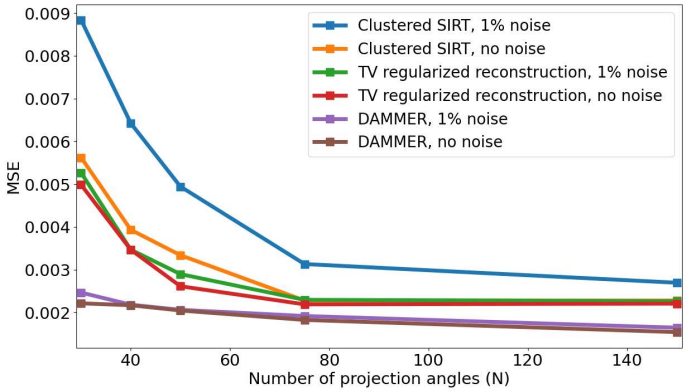
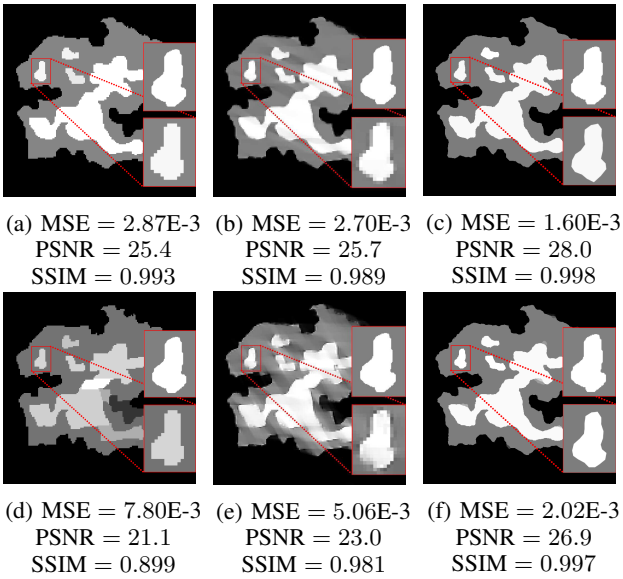


Fig. 13: MSE in function of the number of projections for DAMMER, clustered SIRT, and TV regularized pixel grid reconstructions, both for noiseless and noisy projection data of Phantom 2. For clustered SIRT and TV, the regularization parameters κ_{seg} and κ_{TV} were tuned to minimize the MSE and set to their optimal values accordingly.

projection angle zero corresponds with the horizontal optical axis. Fig. 15 shows the noisy equivalent.

5) Effect of resolution parameter δ: Fig. 16 compares two DAMMER reconstructions based on 150 noiseless projections of Phantom 2. Left with δ set to 0.01 times the size of the reconstruction domain (same experiment as depicted in Fig. 10f), and right with δ set to 0.005 times the size of the reconstruction domain. The interface of the material shown in the inset of the right figure shows curvature of a smaller scale compared to the left. Fig. 17 plots the MSE as a function of the number of triangles, increased by decreasing δ. This figure shows that, although the MSE decreases with increasing resolution, the rate of improvement decreases with increasing number of triangles.

6) Effect of segmentation parameter κ_{seg}: MSE values of DAMMER reconstructions at 1% noise level for different



(a) MSE = 2.87E-3 PSNR = 25.4 SSIM = 0.993
(b) MSE = 2.70E-3 PSNR = 25.7 SSIM = 0.989
(c) MSE = 1.60E-3 PSNR = 28.0 SSIM = 0.998
(d) MSE = 7.80E-3 PSNR = 21.1 SSIM = 0.899
(e) MSE = 5.06E-3 PSNR = 23.0 SSIM = 0.981
(f) MSE = 2.02E-3 PSNR = 26.9 SSIM = 0.997

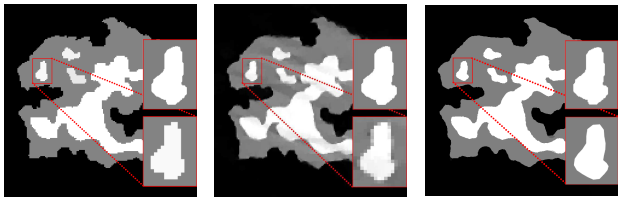
Fig. 14: Clustered SIRT pixel reconstructions (left column), TV regularized pixel reconstructions (middle column) and DAMMER reconstructions (right column) based on 150 noiseless, equiangular projections of Phantom 2 covering an angular range equal to $[0°, 150°]$ (top row), or $[0°, 120°]$ (bottom row), with corresponding error metrics. For clustered SIRT and TV, the regularization parameters κ_{seg} and κ_{TV} were tuned to minimize the MSE and set to their optimal values accordingly. The lower inset zooms in on an ROI, and the upper inset shows the corresponding ground truth.

values of κ_{seg}, expressed in function of the value $\kappa_{\text{seg},0}$ found by the tuning method proposed in this paper, are plotted in Fig. 18. This figure shows little variance in MSE for varying values of κ_{seg}, and a minimum at a value slightly lower compared to $\kappa_{\text{seg},0}$. In Fig. 19, we compare the DAMMER reconstruction found by setting $\kappa_{\text{seg}} = \kappa_{\text{seg},0}$ to the optimal value in Fig. 18, as well as to the worst value in Fig. 18.

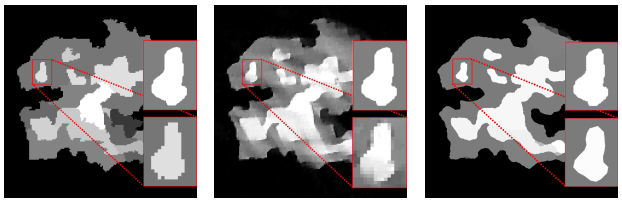
7) Effect of interface orientation parameter κ_{opt}: Fig. 20 shows noiseless and noisy DAMMER reconstructions for different values of the variable κ_{opt} in Eq. (11). These figures show that setting κ_{opt} to a value greater than 0 decreases the MSE, in particular in the presence of noise. However, increasing this parameter may lead to overly smooth object representations.

B. Sensitivity to small attenuation differences

The reconstruction on the starting mesh (Fig. 5a) of the Disk phantom in Fig. 6 is shown in Fig. 21a. The final DAMMER reconstructions are shown in Fig. 21b. These figures show the upper two smaller disks are not reconstructed. For the Disk phantom, the upper bound on the noise induced change, $\Delta\mu_{\text{max}}$, estimated by Eq. (4), is equal to 0.014. The attenuation differences in the triangles containing the different smaller disks (seen in Fig. 21a) are, from bottom to top, 0.036, 0.018, 0.012, and 0.011.

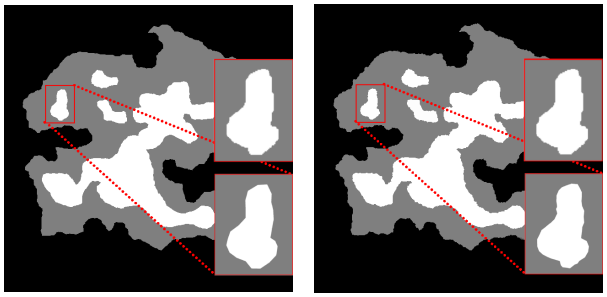


(a) MSE = 4.69E-3 PSNR = 23.3 SSIM = 0.931 (b) MSE = 3.06E-3 PSNR = 25.1 SSIM = 0.980 (c) MSE = 1.80E-3 PSNR = 27.4 SSIM = 0.993



(d) MSE = 8.97E-3 PSNR = 20.5 SSIM = 0.894 (e) MSE = 6.22E-3 PSNR = 22.1 SSIM = 0.963 (f) MSE = 2.13E-3 PSNR = 26.7 SSIM = 0.989

Fig. 15: Clustered SIRT pixel reconstructions (left column), TV regularized pixel reconstructions (middle column) and DAMMER reconstructions (right column) based on 150 noisy (1% noise level), equiangular projections of Phantom 2 covering an angular range equal to $[0°, 150°]$ (top row), $[0°, 120°]$ (lower row), with corresponding error metrics. For clustered SIRT and TV, the regularization parameters κ_{seg} and κ_{TV} were tuned to minimize the MSE and set to their optimal values accordingly. The lower inset zooms in on an ROI, and the upper inset shows the corresponding ground truth.



(a) MSE = 1.38E-3 PSNR = 28.6 SSIM = 0.998 (b) MSE = 8.90E-4 PSNR = 30.5 SSIM = 0.999

Fig. 16: DAMMER reconstructions of Phantom 2 for $\delta = 0.01$ (left), and $\delta = 0.005$ (right). The lower insets of the figures zoom in on an ROI while the upper insets show the corresponding groundtruth.

C. Experimental data

The DAMMER reconstruction of the lego blocks is shown in Figure 22. Fig. 22c shows the full gear and blocks, Fig. 22d shows the mesh triangles of a part of the reconstruction.

V. Discussion

In this paper, we proposed DAMMER, a mesh-based reconstruction method aimed at diminishing PVE in industrial CT applications by optimizing interfaces based on projection data.

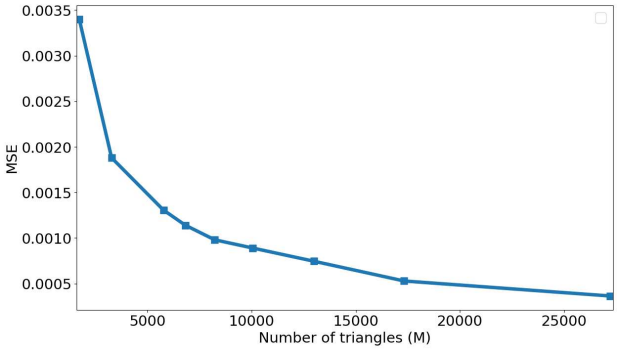


Fig. 17: MSE of DAMMER reconstructions of Phantom 2 as a function of the number of triangles.

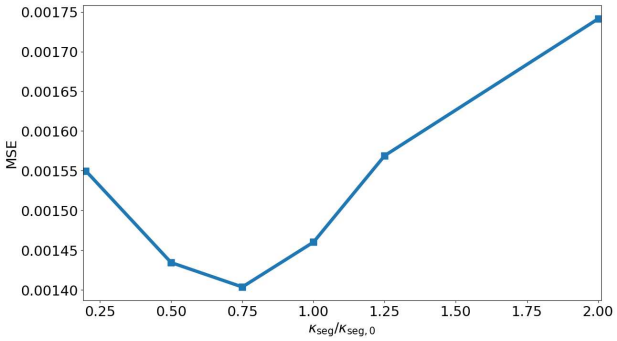
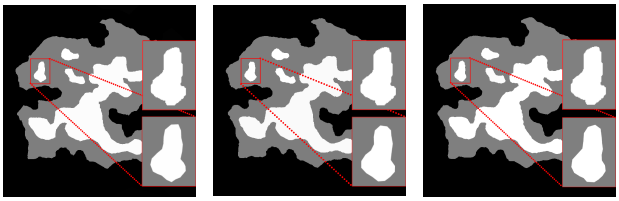


Fig. 18: MSE of DAMMER in function of κ_{seg}.



(a) $\kappa_{\text{seg}} = 0.75\kappa_{\text{seg},0}$ (b) $\kappa_{\text{seg}} = \kappa_{\text{seg},0}$ (c) $\kappa_{\text{seg}} = 2\kappa_{\text{seg},0}$

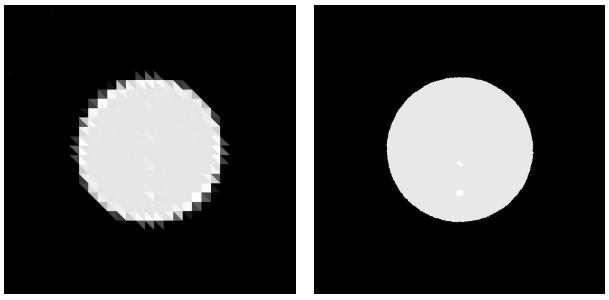
Fig. 19: DAMMER reconstructions based on 150 projections (1% noise level) for $\kappa_{\text{seg},0}$ proposed in this paper, and the κ_{seg} values minimising and maximising the MSE in Fig. 18.

Starting from a coarse regular triangle mesh, triangles near interfaces are progressively refined, while edges in homogeneous areas are collapsed, resulting in a mesh with a high vertex density around interfaces and a low vertex density in homogeneous areas. For a small component to be reconstructed by DAMMER, the triangle encapsulating it must be refined. For this, the reconstructed attenuation change of this triangle compared to the background must exceed the noise-induced upper bound on attenuation variation $\Delta\mu_{\max}$, as estimated in Eq. (4). If this condition is not met, the triangle containing the small region will not be refined, and the component can not be reconstructed. The minimum size of components reconstructed by DAMMER can be decreased by increasing the resolution of the starting mesh. The reconstruction on the mesh is then segmented by aggregating triangles into homogeneous multi-triangle segments by minimizing an objective function composed of

(a) MSE = 1.62E-3 PSNR = 27.9 SSIM = 0.998 (b) MSE = 1.38E-3 PSNR = 28.6 SSIM = 0.998 (c) MSE = 1.93E-3 PSNR = 27.2 SSIM = 0.999

(d) MSE = 2.07E-3 PSNR = 26.8 SSIM = 0.991 (e) MSE = 1.46E-3 PSNR = 28.4 SSIM = 0.994 (f) MSE = 2.76E-3 PSNR = 25.6 SSIM = 0.995

Fig. 20: Noiseless (upper row) and 1% noise level (lower row) DAMMER reconstructions of Phantom 2. Different columns represent different values of κ_{opt}: $\kappa_{\text{opt}} = 0$ (left), $\kappa_{\text{opt}} = \kappa_{\text{opt},0}$ (middle), and $\kappa_{\text{opt}} = 10\kappa_{\text{opt},0}$ (right). The lower inset zooms in on an ROI, and the upper inset shows the corresponding ground truth.



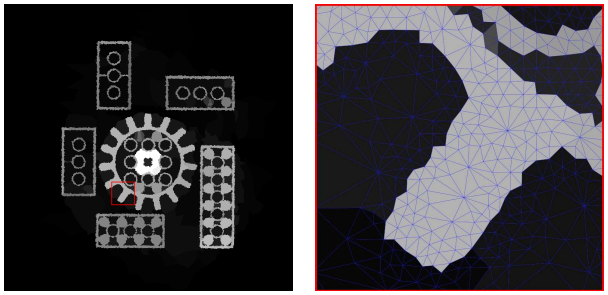
(a) Starting reconstruction (b) DAMMER reconstruction

Fig. 21: DAMMER representations of the Disk phantom.

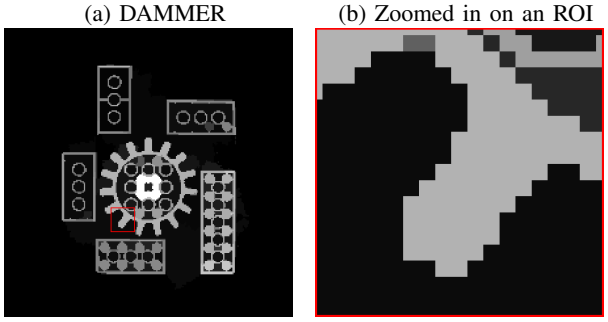
a data-consistency term and a calibrated regularization term. This segmentation procedure reduces the number of vertices to optimize and removes noise effects in homogeneous areas.

Tuning κ_{seg} based on a maximal number of segments enables differentiating higher contrasts typically encountered in industrial CT. We studied the effect of the value of κ_{seg} on the MSE of the DAMMER reconstructions and found the quality of the segmentation is stable to varying values of κ_{seg}. For lower contrasts, which may for example be encountered in medical CT, the regularization parameter κ_{seg} would need to be tuned differently to ensure contrast preserving segmentation.

After segmentation, the vertices of common edges of neighboring segments are repositioned to optimally align these edges with real material interfaces. By iteratively repeating the above steps (i.e., refinement, edge collapses,



(a) DAMMER (b) Zoomed in on an ROI



(c) Clustered SIRT (d) Zoomed in on an ROI

Fig. 22: DAMMER (upper row) and clustered SIRT (bottom row) reconstruction of the lego set.

segmentation, and repositioning interface vertices), DAMMER has the self-correcting ability to model potential new interfaces in split triangles and remove erroneously identified interface edges in homogeneous segments. The vertex optimization is regularized by the interface orientation term κ_{opt}. This term balances the relative importance of data-consistency and a smooth surface. No interface orientation regularization term (setting $\kappa_{\text{opt}} = 0$) leads to worse results, in particular in the presence of noise. Setting κ_{opt} too high may lead to overly smooth object representations.

Simulation experiments with noiseless projection data of various numerical phantoms have demonstrated DAMMER's theoretical potential to diminish PVE (see Fig. 10), and to represent interfaces of various phantoms up to a desired resolution. Figs. 10c and 10f show DAMMER's performance on an irregularly shaped interface, demonstrating more accurate representations compared to equivalent pixel grid reconstructions, though smoothed due to the finite resolution parameter δ. As is shown in Fig. 16, decreasing this resolution parameter results in smaller triangles around interfaces, yielding a reconstruction in which interfaces are smoothed out over a smaller distance, which reduces the MSE. As shown in Fig. 17, the MSE continues to decrease with an increasing number of triangles (or, equivalently, increasing resolution). However, although the plot has not fully converged yet, we expect a lower bound on the MSE due to the finite number of projections and detector pixels. Results for the Shepp-Logan phantom, show DAMMER's ability to separate objects smaller than the resolution of the starting mesh correctly. This is evident by comparing the original reconstruction depicted in Fig 9a to the final reconstruction in Fig. 11h. The results for the Shepp-Logan

phantom also confirm DAMMER's ability to correctly model intersecting interfaces.

For noisy simulated input data, Figs. 11c and 11d show increased smoothing of the ground truth interfaces compared to the noiseless results. This can be explained by the fact that the presence of noise flattens the data consistency term in Eq. (11) around its (noise elevated) minimum, which gives more weight to the interface orientation term in the minimization of the objective function, resulting in smoother interface curves. The results of noisy simulations show that at low noise levels DAMMER can still differentiate the smaller ellipses of the SL phantom (see Fig. 11i). It follows from Figs. 11, 12, and 15 that in all noisy simulation experiments performed in this study, DAMMER outperformed clustered SIRT reconstructions. For high levels of noise, TV regularised pixel grid reconstructions slightly outperformed DAMMER. It is important however to note that this TV reconstruction was tuned using the optimal regularization parameter κ_{TV}, which is unknown in practice. We would also like to note that any regularisation on pixel grid reconstructions, including this TV regularisation, can also be implemented to improve DAMMER results.

For experimental fan-beam data, DAMMER provided a multi-resolution reconstruction of the set of lego blocks. In the absence of ground-truth data, we evaluated DAMMER's performance visually by comparing it to a high-resolution SIRT reconstruction. DAMMER yielded results similar to this ideal reference. Despite this, the reconstructed interfaces were not as sharp as those observed in the simulated images. We attribute this to the higher noise levels in the experimental data. However, the DAMMER reconstruction still exhibited smoother edges compared to the comparable clustered SIRT reconstruction.

Altogether, simulation and experimental results demonstrated that the results of DAMMER consistently outperformed conventional pixel grid reconstructions of comparable memory intake both qualitatively (see Figs. 10-11) and quantitatively, in terms of MSE (see Fig. 13). DAMMER is able to accurately reconstruct interfaces of compound homogeneous objects up to a desired resolution. In contrast to previous methods, it does not require prior knowledge of the (number of) materials in the sample or identifying different objects from a reconstruction on a high resolution grid.

Although DAMMER provides remarkable results, there are still a few challenges to overcome. A first challenge is the robustness to higher noise levels encountered in industrial CT. This can potentially be improved by denoising the reconstructed image with a TV term before the segmentation step. Another limitation is the use of the system matrix, which becomes infeasible for larger 3D reconstructions often encountered in realistic scans. As part of an extension to 3D tetrahedral meshes, we therefore plan to reconstruct attenuation values on the tetrahedra on the fly by computing ray-face intersections through the CAD-ASTRA mesh projector [11]. A final challenge, in particular in 3D, is the high computation time on large reconstructions. DAMMER, which jointly optimizes reconstruction elements and their associated attenuation values, inherently involves a more prolonged optimization process and therefore a higher computational cost than standard pixel-based methods (see Appendix B for a detailed discussion on the time complexity). However, the CAD-ASTRA projector also offers a highly optimized function to compute the gradient of the projection difference to the vertex coordinates. With this function, we plan to optimize the tetrahedra to match real material interfaces.

VI. Summary and conclusions

In this paper, we proposed an innovative multi-resolution, mesh-based 2D reconstruction method that reconstructs a sample's attenuation coefficient on the triangles of an adaptively optimized triangular mesh, based on X-ray projection data. While conventional techniques are limited to reconstructing attenuation coefficients on a fixed regular pixel grid, our newly proposed method, DAMMER, extends this scope by integrating the SIRT reconstruction method within methods that optimize the reconstruction mesh, thereby diminishing partial volume effects. Unlike previously proposed mesh-based reconstruction methods, DAMMER allows accurate modeling of intersecting interfaces and does not require prior knowledge on the number of homogeneous materials present in the sample or their attenuation values. The results of simulation experiments with noisy input data showed that DAMMER outperforms conventional pixel grid reconstructions (of comparable memory occupation) in terms of Mean Squared Error, Peak Signal-to-Noise Ratio and Structural Similarity Index Measure. Moreover, simulation experiments with different phantoms demonstrated DAMMER's robustness to reduced projection angles, constrained projection geometries, and noise, highlighting its potential in challenging imaging scenarios. Although there are still challenges to extend this method to 3D, a preliminary result on an experimental fan-beam dataset showed an efficient representation of a challenging phantom.

Acknowledgements

The authors would like to thank Bart van Lith, Nicholas Francken and Jian Deng (University of Antwerp) for their insights and support of our research. The authors also gratefully acknowledge support from the Research Foundation Flanders, Belgium (FWO) through projects S003421N and 1SG0523N.

References

[1] G.H. Glover and N.J. Pelc, *Nonlinear partial volume artifacts in x-ray computed tomography*, The International Journal of Medical Physics Research and Practise **7**(3), pp. 238-248, 1980, doi.org/10.1118/1.594678

[2] A. Souza, J.K. Udupa and P.K. Saha, *Volume rendering in the presence of partial volume effects*, IEEE Transactions on Medical Imaging **24**(2), pp. 223-235, 2005, doi.org/10.1109/TMI.2004.840295

[3] E. Maire and P. J. Withers, *Quantitative X-ray tomography*, International Materials Reviews **59**(1), pp. 1-43, 2014, doi.org/10.1109/TMI.2004.840295

[4] H. Samet, *An Overview of Quadtrees, Octrees, and Related Hierarchical Data Structures*, Theoretical Foundations of Computer Graphics and CAD **40**, pp. 51-68, 1988, doi.org/10.1007/978-3-642-83539-1_2

[5] H. Guo, el al., *A quadtree-polygon-based scaled boundary finite element method for image-based mesoscale fracture modelling in concrete*, Engineering Fracture Mechanics **211**, pp. 420-441, 2019, doi.org/10.1016/j.engfracmech.2019.02.021

[6] M.G. Cawley, K. Natarajan and J.A. Newell, *Data representation for subsequent image interpretation*, Medical Informatics **16**(2), pp. 125-136, 1991, doi.org/10.1016/j.engfracmech.2019.02.021

[7] L. Xue, K.M. Zhaoxiang Li & J. Wang *Adaptive and High-Precision Isosurface Meshes from CT Data*, *J Nondestruct Eval* **43**(85), 2024, doi.org/10.1007/s10921-024-01102-8

[8] J. Koo et al., *Shape from projections via differentiable forward projector for computed tomography*, Ultramicroscopy **224**, 2021, doi.org/10.1016/j.ultramic.2021.113239

[9] S. Jiang et al., *Research on meshing method for industrial CT volume data based on iterative smooth signed distance surface reconstruction*, Journal of X-Ray Science and Technology **33**(2), pp. 340-349, 2025, doi.org/10.1177/08953996241306691

[10] Y. Chen et al., *A content-adaptive unstructured grid based regularized CT reconstruction method with a SART-type preconditioned fixed-point proximity algorithm*, Inverse problems **38**, 2022, doi.org/10.1088/1361-6420/ac490f

[11] P. Paramonov et al., *CAD-ASTRA: A versatile and efficient mesh projector for X-ray tomography with the ASTRA-toolbox* Optics Express **32**(3), 2024, https://doi.org/10.1364/OE.498194

[12] F. Buyens et al., *Adaptive Mesh Reconstruction in X-Ray Tomography*, MICCAI - Workshop on Mesh Processing in Medical Image Analysis, Nagoya, Japan, 2013, doi.org/10.1109/TMI.2006.879319

[13] J. Koo et al., *DALM, Deformable Attenuation-Labeled Mesh for Tomographic Reconstruction and Segmentation*, IEEE Transactions on Computational Imaging **7**, pp. 151-163, 2021, doi.org/10.1109/TCI.2021.3052034

[14] A. Sitek et al., *Tomographic reconstruction using an adaptive tetrahedral mesh defined by a point cloud*, IEEE Trans. Med. Imag. **25**(9), pp. 1172–1179, 2006, 10.1109/tmi.2006.8793196

[15] N. F. Pereira and A. Sitek, *Evaluation of a 3D point cloud tetrahedral tomographic reconstruction method*, Phys. Med. Biol. **55**(18), pp. 5341–5361, 2010, doi.org/10.1088/0031-9155/55/18/006

[16] P. Lesonen et al., *Anatomy-guided multi-resolution image reconstruction in PET*, Physics in Medicine & Biology, **69**(10), p. 105023, 2024, doi.org/10.1088/1361-6560/ad4082

[17] R.R. Colmeiro et al., *Reconstruction of positron emission tomography images using adaptive sliced remeshing strategy*, J Med Imaging, **8**(2), 2021, doi.org/10.1117/1.JMI.8.2.024001

[18] DAMMER GitHub repository, [retrieved 09 May 2025], https://github.com/JaMerckx/DAMMER

[19] J. Gregor and T. Benson, *Computational analysis and improvement of SIRT*, IEEE Transactions on Medical Imaging **27**, 7, pp. 918–924, 2008, 10.1109/TMI.2008.923696

[20] J. Shimeda and D.C. Gossard, *Automatic triangular mesh generation of trimmed parametric surfaces for finite element analysis*, Computer Aided Geometric Design **15**(3), pp. 199-222, 1998, 10.1016/S0167-8396(97)00037-X

[21] B. Delaunay, *Sur la sphère vide*, Bulletin de l'Académie des Sciences de l'URSS, Classe des Sciences Mathématiques et Naturelles **6**, pp. 793-800, 1934

[22] D.A. Belsley et al., *The Condition Number*. Regression Diagnostics: Identifying Influential Data and Sources of Collinearity. pp. 100–104, (1980)

[23] L.P. Chew, *Guaranteed-quality mesh generation for curved surfaces*, SCG '93: Proceedings of the ninth annual symposium on Computational geometry **9**(32), pp. 274–280, 1993, doi.org/10.1145/160985.161150

[24] H. Hoppe, Progressive meshes, *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 99-108, 1996

[25] T. Hastie et al., *The elements of statistical learning: Data mining, inference, and prediction*, Springer, 2009

[26] A. Cohen, *Local recognition of graphs, buildings, and related geometries*, Finite Geometries, Buildings, and Related Topics, 1990, pp. 85-94

[27] J.A. Ford and I.A. Moghrabi, *Multi-step quasi-Newton methods for optimization*, Journal of Computational and Applied Mathematics **50**, pp. 305-323, 1994, doi.org/10.1016/0377-0427(94)90309-3

[28] W. van Aarle et al., *Fast and flexible X-ray tomography using the ASTRA toolbox*, Optics Express **24**(22), 2016, doi.org/10.1364/OE.24.025129

[29] T. Barbu, *Variational Image Denoising Approach with Diffusion Porous Media Flow*, Abstract and applied analysis, 2013

[30] L.I. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D: Nonlinear Phenomena **60**(1–4), pp. 259-268, 1992, doi.org/10.1016/0167-2789(92)90242-F

[31] Z. Wang et al., *Image Quality Assessment: From Error Visibility to Structural Similarity*, IEEE Transactions on Image Processing **13**(4), pp. 600-612, 2004, doi.org/10.1109/TIP.2003.819861

[32] B. De Samber et al., *FleXCT: a flexible X-ray CT scanner with 10 degrees of freedom*, Optics Express **29**(3), pp. 3438-3457, 2021, https://doi.org/10.1364/OE.409982

[33] J.D. Hunter, *Matplotlib: A 2D graphics environment*, Computing in Science & Engineering **9**(3), pp. 90-95, 2007, doi.org/10.1109/MCSE.2007.55

APPENDIX A
VARIABLES

Table I sums up all the parameters in this paper.

APPENDIX B
TIME COMPLEXITY

Each SIRT iteration (see Section II-A) induces, through the matrix multiplications necessary to calculate the update of $\boldsymbol{\mu}$, a time complexity of approximately $O\left(M \cdot N \cdot P\right)$, so that the time complexity of N_{it} SIRT iterations is approximately of $O\left(N_{\text{it}} \cdot M \cdot N \cdot P\right)$.

For the refinement method (see Section II-B), in the worst case, each SIRT reconstruction requires a number of computations proportional to $N_{\text{it}} \cdot M \cdot N \cdot P$. However, after most refinement steps, the stopping condition defined by p_1 is already satisfied by the triangle reconstruction mesh updated through the weighted average, and SIRT reconstruction updates are therefore not required. In such cases, the time complexity of the refinement method is roughly proportional to the number of system matrix updates required by local updates of the mesh, $\propto N \cdot P$. Due to the local nature of the updates, this does not scale with the number of triangles. In the worst-case scenario, where the entire reconstruction domain is refined to a resolution δ, the number of times each triangle in the starting mesh must be split is proportional to $\delta\sqrt{M_0}^{-1}$. The number of triangles to refine is, in the worst-case scenario, proportional to $\frac{1}{\delta^2}$. For realistic phantoms with interfaces, however, this number scales approximately with $\frac{1}{\delta}$, since only triangles around these 1D interfaces are refined. Substituting $\propto M$ with $\propto 1/\delta^2$, we get an overall worst-case (in terms of δ) time complexity of $O\left(\frac{N \cdot P}{\delta^3\sqrt{M_0}}\right)$ for the refinement method. Substituting $M \propto \frac{1}{\delta}$ for realistic materials with interfaces induces a time complexity of $O\left(\frac{N \cdot P}{\delta\sqrt{M_0}}\right)$.

The edge collapse method involves both computing the variance around a number of edges proportional to the number of triangles, and collapsing edges in homogeneous areas, of which the number is roughly proportional to the number of triangles. Therefore, this method has time complexity of $O\left(M\right)$.

For the segmentation method, checking all edges for an aggregation requires a number of checks proportional to M. Therefore, the total time complexity of the segmentation

TABLE I: Variables of the DAMMER algorithm.

$\boldsymbol{A}$	System matrix
$\boldsymbol{b}$	Projection data
$\boldsymbol{c}_m$	Center circumscribed circle of triangle m
d_{col}	Minimum edge length
δ	Minimal circumradius of refined triangles
$\Delta\mu_{\text{max}}$	Statistical maximum deviation reconstructed attenuation due to noise estimated by Eq. (4)
$\boldsymbol{E}$	Edge list
E_d	Objective function vertex displacement, see Eq (11)
κ_{col}	Parameter that balances the relative contributions of local attenuation variance and triangle equiangularity to the edge collapsing weight Eq. (8)
k_{dis}	Number of vertex displacement iterations
κ_{opt}	Parameter that balances the relative contributions of the projection difference and the interface orientation term in the interface energy function in Eq. (11)
$\kappa_{\text{opt},0}$	Value of κ_{opt} found by Eq. (12)
κ_{ref}	Parameter that balances the relative contributions of triangle equiangularity and local attenuation triangle's splitting weight in Eq. (3)
κ_{seg}	Parameter that balances the relative contributions of projection difference and number of interface edges to the objective function for triangle aggregation in Eq. (10)
$\kappa_{\text{seg},0}$	Segmentation parameter found by the method introduced in Section II-D1
κ_{TV}	Parameter that balances the relative contributions of the projection difference and the TV term in Eq. (15)
l_m	Shortest edge of triangle m
$M_{(0)}$	Number of triangles in the (starting) mesh
$\boldsymbol{\mu}$	Reconstructed attenuation
M_{thr}	Threshold weight for edge collapse
N	Number of projection angles
N_{BFGS}	Number of BFGS optimization steps
N_{dis}	Number of displacement steps
N_e	Number of edges
n_{it}	An iteration of the SIRT algorithm
N_{it}	Number of iterations initial SIRT
$\boldsymbol{n}(m)$	Neighbours of triangle m
N_R	Number of homogeneous segments
$N_{R,\text{thr}}$	Maximum number of connected triangle segments in segmentation
$N_{v,\text{int}}$	Number of interface vertices
$N_{V(,\text{int})}$	Number of (interface) vertices
ξ	Condition number
P	Number of detector pixels
p_1	Stopping condition of SIRT based on change in attenuation
$\tilde{p}_e$	Sum of the circumradius-to-shortest-edge ratios of triangles connected by edge e
q_{max}	Maximal desired circumradius-to-shortest-edge ratio
r_m	Circumradius of triangle m
σ	Standard deviation Gaussian noise (noise level)
σ_1^2	Maximal allowed variance in reconstructed attenuation around edges in the edge collapse method
$\tilde{s}_e$	Variance reconstructed attenuation around edge $\boldsymbol{e}$
s_v	Variance reconstructed attenuation around vertex v, see Eq. (7)
S_{thr}	Threshold splitting weight, see Eq. (5)
$\boldsymbol{T}$	Connectivity list
$\boldsymbol{X}$	Vertex coordinates
$\boldsymbol{X}_{\text{int}}$	Interface vertex coordinates

procedure (that involves computing the projection difference) is of $O\left(M^2 \cdot (N \cdot P)\right)$.

Optimizing interface vertex positions requires computing the gradient of Eq. (11). Both computing this gradient and checking whether the line-search condition is met (through calculating the squared projection difference) require a number of computations that scales linear with $N_{v,\text{int}} \cdot (N \cdot P)$. Computing the BFGS matrix, as well as applying the BFGS update, has time complexity of $O\left(N_{v,\text{int}}^2\right)$, which is however typically orders of magnitude faster than computing the gradient. The vertex optimization is performed for N_{BFGS} iterations, resulting in a total time complexity of $O\left(N_{\text{BFGS}} \cdot N_{v,\text{int}} \cdot N \cdot P\right)$.

The edge flipping procedure computes four circumradius-to-shortest-edge ratios per edge — two before and two after a potential flip — resulting in a constant time cost per edge. Since the number of candidate edges is proportional to the number of triangles M, the edge flip algorithm has time complexity of $O\left(M\right)$.

Out of these methods, the most time-consuming is the mesh refinement, in particular the first, which starts from a coarse mesh. This step takes around 7 minutes for simulated data experiments of Phantom 2. However, this time can be greatly reduced by picking a finer starting mesh. For example, increasing the number of triangles in the starting mesh from 1922 to 5408 reduces the running time of the refinement step to 4 minutes. After this, the most time-consuming part is the vertex displacement method. Even though computing ray-edge intersections can be parallelised on the GPU, this computation still has to be repeated for each iteration of the line-search, which, for the simulation examples described in this paper, leads to typical computation times of about 14 seconds per BFGS iteration. This computation time could be greatly improved by reducing the number of projections. For example, with 50 projections the running time reduces to 5 seconds per BFGS iteration.