Fast and efficient tetrahedral volume mesh reconstruction with CAD-ASTRA

Jannes Merckx[a], Arnold J. den Dekker[a], Jan De Beenhouwer[a], and Jan Sijbers[a]

[a]imec-Vision Lab, Department of Physics, University of Antwerp, Belgium

ABSTRACT

Reconstructing X-ray CT data on cubic voxels leads to discretization errors due to the finite size of these voxels. These errors are known as Partial Volume Effects (PVE). Reconstructing X-ray data on a tetrahedral mesh adapted to the sample generates a memory efficient multi-resolution representation that has the potential to diminish PVE. Such an adaptive method requires reconstructing attenuation values of the scanned sample on tetrahedra of the mesh. While a wide variety of open source software is available for iterative reconstruction methods on cubic voxels, research on fast iterative reconstructions on tetrahedral meshes is limited. In this work, we present an extension of the GPU accelerated CAD-ASTRA[1,2] mesh projector that solves the inverse problem on tetrahedra. With an appropriate starting condition, such a representation has the potential to greatly increase the accuracy of a CT reconstruction.

Keywords: Tetrahedral mesh reconstruction, Iterative reconstruction algorithms

1. INTRODUCTION

In X-ray Computed Tomography (CT), the attenuation values of a sample are typically reconstructed on a regular voxel grid. Since it is unlikely that the faces of the cubic voxels align with real material interfaces, voxels often contain two or more materials, causing reconstruction errors known as partial volume effects (PVE).[3–5] PVE significantly compromise the accuracy of interface localization in (industrial) X-ray methods.[6]

Alternatively, interfaces in a sample can be represented by watertight polygonal surface meshes. These meshes consist of interconnected polygons, most commonly triangles, that collectively approximate the geometric shape of the interface. Since vertex positions are not bound to grid coordinates, meshes allow a more precise representation of the materials in the sample. Furthermore, the vertex positions of the mesh can be optimised to match interfaces.[7] A limitation of surface mesh representations is that the number of meshes and attenuation values are fixed, and assumed to be extracted from a high resolution voxel reconstruction through the marching cubes algorithm.[8]

To improve the efficiency of mesh models in representing interfaces, reconstructed data can be represented by an adaptive tetrahedral mesh or point cloud.[9–12] However, splitting and merging of tetrahedral cells requires a fast method to compute attenuation values on the tetrahedra of the mesh. For this, we propose adding a function in the CAD-ASTRA forward projector that calculates the attenuation values corresponding to tetrahedra with a GPU accelerated iterative reconstruction technique. In this paper, we will study the performance of this tetrahedral reconstruction method in terms of computation time. We will also show how improving the tetrahedral mesh improves the result of the reconstruction compared to a conventional voxel reconstruction with the same number of volume elements.

Further author information: E-mail: jannes.merckx@uantwerpen.be

2. MATERIALS AND METHODS

A CT reconstruction on a tetrahedral (or square voxel) tessellation of the image domain involves solving the linear system of equations

$$\boldsymbol{A}\boldsymbol{\mu} = \boldsymbol{b}. \tag{1}$$

In (1), $\boldsymbol{\mu} = (\mu_m) \in \mathbb{R}^M$ refers to the reconstructed attenuation values corresponding to M tetrahedra (or voxels). The measured projection data, acquired under P projection angles by a detector with $N \times N$ detector pixels (assuming a square detector), are denoted $\boldsymbol{b} = (b_i) \in \mathbb{R}^{N^2 \cdot P}$. Finally, and $\boldsymbol{A} = (a_{i,m}) \in \mathbb{R}^{(N^2 \cdot P) \times M}$ corresponds with the (sparse) system matrix, in which $a_{i,m}$ represents the contribution of tetrahedron m to projection value b_i. Directly solving the system of equations (1) for an exact solution $\boldsymbol{\mu}$ is typically infeasible, since noise and discretization effects render the system inconsistent. Therefore iterative methods, including ART, SART, SIRT and gradient descent,[13] typically minimize the projection distance $||\boldsymbol{A}\boldsymbol{\mu} - \boldsymbol{b}||$ for some norm $|| \cdot ||$. In this paper, the SIRT algorithm is implemented on tetrahedra meshes. SIRT is an iterative reconstruction algorithm known to converge to a solution of

$$\hat{\boldsymbol{\mu}} = \arg\min_{\boldsymbol{\mu}} ||\boldsymbol{A}\boldsymbol{\mu} - \boldsymbol{b}||^2_{\boldsymbol{R}} \ . \tag{2}$$

where $\boldsymbol{R} \in \mathbb{R}^{(N \cdot P) \times (N \cdot P)}$ is the diagonal matrix that contains the inverse row sums of the system matrix $\boldsymbol{A}$ (its diagonal elements are given by $r_{i,i} = 1/\sum_j a_{i,j}$), and $||\boldsymbol{A}\boldsymbol{\mu} - \boldsymbol{b}||^2_{\boldsymbol{R}} = (\boldsymbol{A}\boldsymbol{\mu} - \boldsymbol{b})^T \boldsymbol{R} (\boldsymbol{A}\boldsymbol{\mu} - \boldsymbol{b})$. Starting from an initial reconstruction $\boldsymbol{\mu}^0 = \boldsymbol{0}$, the SIRT algorithm iteratively updates the reconstruction as follows:

$$\boldsymbol{\mu}^k = \boldsymbol{\mu}^{k-1} + \boldsymbol{C}\boldsymbol{A}^T\boldsymbol{R}\left(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{\mu}^{k-1}\right), \tag{3}$$

with $\boldsymbol{C} \in \mathbb{R}^{M \times M}$ the diagonal matrix that contains the inverse column sums of the system matrix $\boldsymbol{A}$ (i.e., $c_{m,m} = 1/\sum_i a_{i,m}$.

Although the matrices $\boldsymbol{R}$ and $\boldsymbol{C}$ can be computed in an initial iteration and stored in memory, the system matrix A is too large to store. Therefore the practical implementation of (3) requires an efficient ray tracer to quickly compute elements of the system matrix and update attenuation values based on these elements. Each update with SIRT can be split in updates of individual tetrahedra t as

$$\mu_t^k = \mu_t^{k-1} + c_{t,t} \sum_i \left(a_{i,t} r_{i,i} \left(b_i - \sum_j \left(a_{i,j} \mu_j^{k-1} \right) \right) \right). \tag{4}$$

A SIRT update of the attenuation values associated with the tetrahedral mesh elements consists of multiplying all system matrix elements with the corresponding values of $\boldsymbol{C}$ and $\boldsymbol{R}\left(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{\mu}^{k-1}\right)$, before adding each result to the attenuation value associated with the corresponding tetrahedron. The system matrix elements can be computed by tracing all intersections between rays corresponding to the measurements and faces of the tetrahedra. These are exactly the computations the CAD-ASTRA forward projector performs to update projection data in parallel on the GPU.[1,2] Each ray-triangle intersection, CAD-ASTRA either adds or subtracts the product of the intersection distance and the attenuation of the tetrahedron to the projection value corresponding to the ray, depending on whether the ray enters or leaves the tetrahedron at this face. Rather than updating projection data from attenuation values for each ray-triangle intersection, we update reconstructed attenuation values of the tetrahedra by multiplying with the corresponding elements of $\boldsymbol{C}$ and $\boldsymbol{R}\left(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{\mu}^{k-1}\right)$. For an intersection of ray l (with direction $\boldsymbol{z}_l$), with face f, attenuation values of the tetrahedra that share f, are updated by

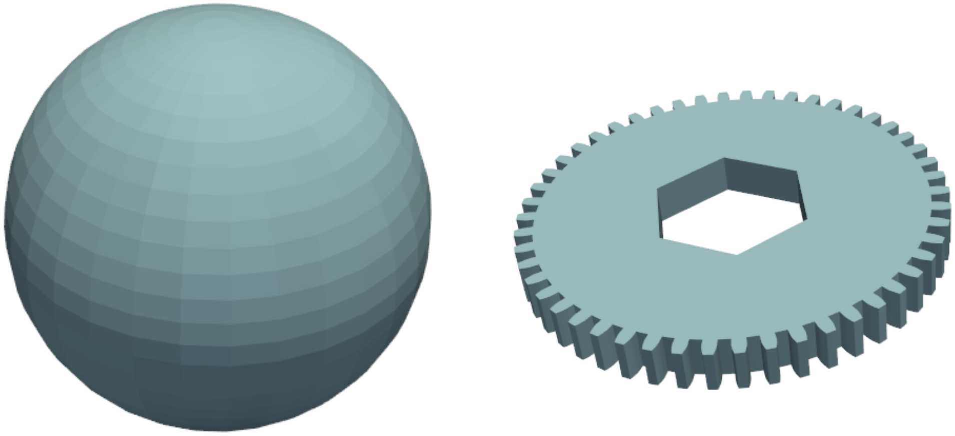
$$\mu_t^k = \mu_t^{k-1} + \operatorname{sgn}\left(\boldsymbol{n}_{f,t} \cdot \boldsymbol{z}_l\right) c_{t,t} r_{l,l} \left(b_l - \sum_j a_{l,j} \mu_j^{k-1} \right), \tag{5}$$

where $\boldsymbol{n}_{f,t}$ refers to the outside normal of tetrahedron t on face f. Therefore, each SIRT iteration consists of first computing the difference between simulated and measured projection data $(b_l - \sum\limits_j a_{l,j}\mu_j^{k-1})$ with CAD-ASTRA, and then updating attenuation values by (5).

Following an initial tetrahedral SIRT reconstruction, interface vertices can be extracted with the marching tetrahedra algorithm.[14] We propose creating a new tetrahedral mesh, containing these extracted interface vertices, with roughly the same number of tetrahedra as the original reconstruction mesh. We expect reconstructions on this updated mesh to improve the result of the initial tetrahedral reconstruction.

3. EXPERIMENTS

The phantoms studied in this paper were a uniform sphere and a gear, shown in Fig. 1.



(a) Spherical phantom

(b) Gear phantom

Figure 1: Phantoms

Forward projection data $\bar{\boldsymbol{b}} = \left(\bar{b}_i\right) \in \mathbb{R}^{N^2 \cdot P}$ of the phantoms shown in Fig. 1 were computed using the CAD-ASTRA forward projector on a high resolution surface mesh representation. These noiseless forward projections were simulated on a 512 × 512 pixel detector under 200 projection angles equiangulary distributed over a full angular range.

The tetrahedral reconstruction method was first tested on tetrahedral meshes created by splitting cubes of a voxel grid in 6 tetrahedra each, cutting each pair of opposing faces in half through a diagonal of these faces. In this paper, we split voxels of a $33 \times 33 \times 33$ grid, corresponding to a tetrahedral mesh of 215622 tetrahedra. This number is equivalent to a $60 \times 60 \times 60$ voxel reconstruction. We compared reconstructions on both this original regular tetrahedral mesh and on tetrahedral meshes of the same size generated based on extracted interface vertices to the equivalent $60 \times 60 \times 60$ voxel grid.

The quality of the SIRT reconstructions was measured in terms of the mean squared residual (MSR) in the projection space, which is defined as

$$\mathrm{MSR} = \frac{1}{N^2 \cdot P} \sum_i \left(\bar{b}_i - \sum_j a_{i,j} \hat{\mu}_j \right)^2 . \tag{6}$$

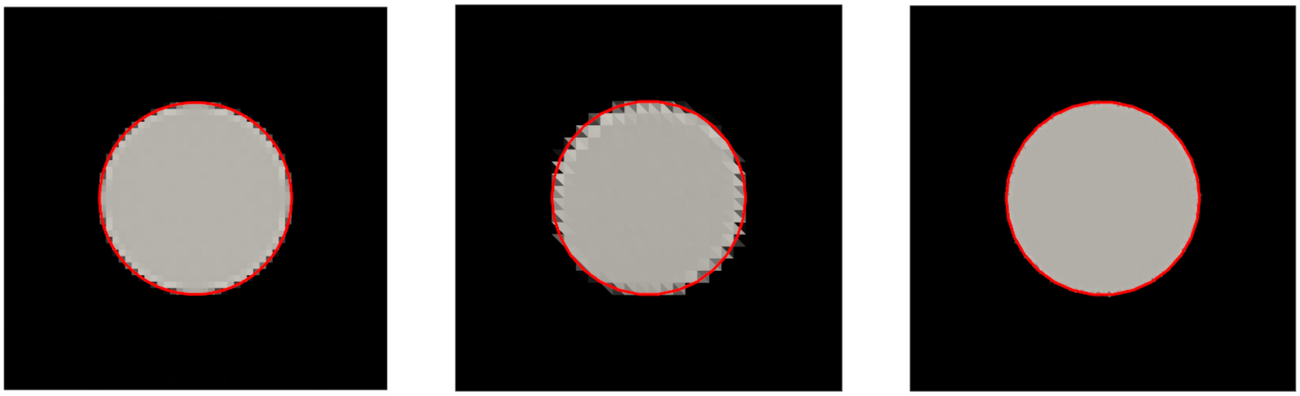
The tetrahedral reconstructions were compared with cubic voxel SIRT reconstructions on approximately the same number of reconstruction elements, generated by the ASTRA toolbox.[15]

4. RESULTS AND DISCUSSION

Figs. 2 and 3 compare central slices of 60 × 60 × 60 voxel reconstructions of the sphere and gear phantoms with central slices of tetrahedral mesh reconstructions. The regular tetrahedra of the mesh generated from the $33 \times 33 \times 33$ voxel grid have a larger dimension than the voxels of the $60 \times 60 \times 60$ voxel grid, which means that the resulting reconstructions, of which slices are shown in Figs. 2b and 3b, show larger overlaps with interfaces compared to the slices shown in Figs. 2a and 3a. For the sphere phantom, the MSR is indeed lower for the regular voxel reconstruction compared to the regular tetrahedral reconstruction. For the gear phantom, however, the MSR shows a better value for a regular tetrahedral grid compared to the equivalent voxel grid. This shows that whether a regular tetrahedral mesh reconstruction improves on a regular voxel reconstruction depends on how well both representations match with the particular phantom.

Updating the tetrahedral meshes based on extracted interface vertices results in reconstructions of which central slices are shown in Figs. 2c and 3c. These slices clearly show better results compared to the slices of equivalent square voxel reconstructions. This can also be seen in the MSR value that decreases an order of magnitude compared to reconstructions on regular tetrahedra, and improves on the value associated with equivalent square voxel reconstructions for both phantoms.

In terms of computation time, a comparison with the ASTRA toolbox is shown in Fig. 4. This figure shows that, although the tetrahedral voxel reconstruction is performed in parallel on the GPU, it is not yet competitive with established voxel reconstruction methods in terms of reconstruction time.

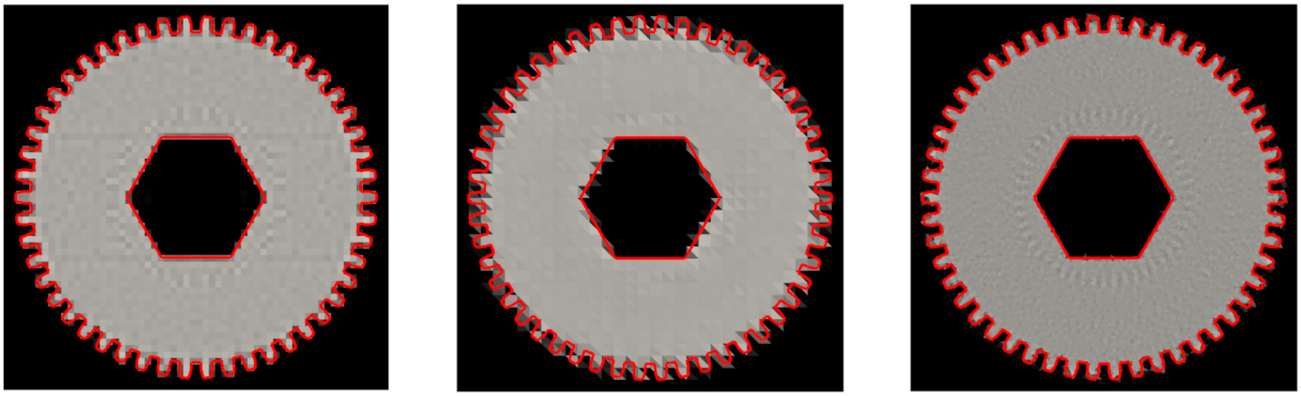


(a) $60 \times 60 \times 60$ voxel grid
MSR $= 4.79 \cdot 10^{-6}$

(b) 215622 regular tetrahedra
MSR $= 8.86 \cdot 10^{-6}$

(c) 200500 adapted tetrahedra
MSR $= 8.70 \cdot 10^{-7}$

Figure 2: Comparison of a voxel reconstruction (left), a tetrahedral reconstruction on a regular starting mesh (middle) and a tetrahedral reconstruction on an updated reconstruction mesh (right) of the sphere phantom in Fig. 1a.



(a) $60 \times 60 \times 60$ voxel grid
MSR $= 1.13 \cdot 10^{-4}$

(b) 215622 regular tetrahedra
MSR $= 8.81 \cdot 10^{-5}$

(c) 205706 adapted tetrahedra
MSR $= 7.18 \cdot 10^{-6}$

Figure 3: Comparison of a voxel reconstruction (left), a tetrahedral reconstruction on a regular starting mesh (middle) and a tetrahedral reconstruction on an updated reconstruction mesh (right) of the gear phantom in Fig. 1b.

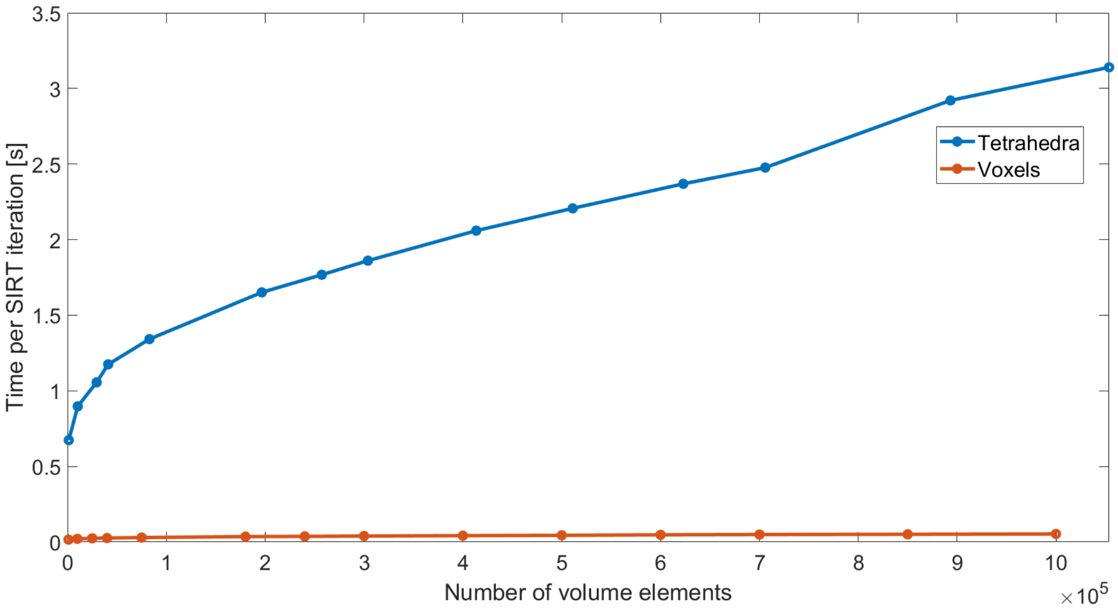


Figure 4: Time per iteration for different numbers of reconstruction elements.

5. CONCLUSION

We studied the performance an iterative solver that reconstructs X-ray CT data on a tetrahedral volume mesh based on the CAD-ASTRA forward projector. Specifically, we compared the correctness of the tetrahedral reconstructions to equivalent voxel reconstructions, both for a regular starting mesh and a mesh updated to contain more vertices around interfaces. This showed that, although the performance of the regular mesh depended on the phantom, an updated mesh outperformed the corresponding voxel reconstruction for both phantoms studied. We monitored the computation time for different numbers of tetrahedra and concluded that our method, despite being GPU accelerated, is not yet competitive with established voxel-based reconstruction methods in terms of reconstruction time. Future work will involve studying more efficient mesh update techniques as well as improving computational efficiency of the tetrahedral mesh reconstruction method.

REFERENCES

[1] Paramonov, P., Renders, J., Elberfeld, T., De Beenhouwer, J., and Sijbers, J., "Efficient x-ray projection of triangular meshes based on ray tracing and rasterization," in [*Proc. SPIE*], *SPIE's Conference on Developments in X-Ray Tomography XIV* (2022).

[2] Paramonov, P., Francken, N., Renders, J., Iuso, D., Elberfeld, T., De Beenhouwer, J., and Sijbers, J., "Cad-astra: A versatile and efficient mesh projector for x-ray tomography with the astra-toolbox projector for computed tomography," *Optics Express,* **32(3)** (2024).

[3] Glover, G. and Pelc, N., "Nonlinear partial volume artifacts in x-ray computed tomography," *The International Journal of Medical Physics Research and Practise* **7(3)**, 238–248 (1980).

[4] Souza, A., Udupa, J. K., and Saha, P. K., "Volume rendering in the presence of partial volume effects," *IEEE Transactions on Medical Imaging* **24(2)**, 223–225 (2005).

[5] Inou, N., Koseki, M., and Maki, K., "Individual modeling method based on the x-ray ct images : Influence of partial volume effect on the modeling(special issue bioengineering of biological and medical materials)," *Transactions of the Japan Society of Mechanical Engineers Series A* **69(677)**, 109–122 (2003).

[6] Maire, E. and Withers, P. J., "Quantitative x-ray tomography," *International Materials Reviews* **59(1)**, 1–43 (2014).

[7] Koo, J., Dahl, A. B., Bærentzen, J. A., Chen, Q., Bals, S., and Dahl, V. A., "Shape from projections via differentiable forward projector for computed tomography," *Ultramicroscopy* **224** (2021).

[8] Lorensen, W. and Cline, H., "Marching cubes: A high resolution 3d surface construction algorithm," *ACM SIGGRAPH Computer Graphics* **21(4)** (1987).

[9] Sitek, A., Huesman, R. H., and Gullberg, G. T., "Tomographic reconstruction using an adaptive tetrahedral mesh defined by a point cloud," *IEEE Transactions on Medical Imaging* **25(9)**, 1172–1179 (2006).

[10] Cazasnoves, A., Buyens, F., and Sevestre-Ghalila, S., "Adapted sampling for 3d x-ray computed tomography," in [*HAL open science*], (2015).

[11] Pereira, N. F. and Sitek, A., "Practical implementation of tetrahedral mesh reconstruction in emission tomography," *Phys Med Biol* **55(18)**, 5341–5361 (2010).

[12] Boutchko, R., Sitek, A., and Gullberg, G. T., "Evaluation of a 3d point cloud tetrahedral tomographic reconstruction method," *Phys Med Biol* **58(9)**, 3001–3022 (2013).

[13] Liaptsis, G., Clarke1, A. L., and Nithiarasu, P., "Comparison of different iterative reconstruction algorithms for x-ray volumetric inspection," *Special Issue of e-Journal of Nondestructive Testing* **23(8)**, 1435–4934 (2018).

[14] Doi, A. and Koide, A., "An efficient method of triangulating equi-valued surfaces by using tetrahedral cells," *IEICE Transactions of Information and Systems* **E74-D(1)** (1991).

[15] Van Aarle, W., Palenstijn, W. J., Cant, J., Janssens, E., Bleichrodt, F., Dabravolski, A., De Beenhouwer, J., Batenburg, K. J., and Sijbers, J., "Fast and flexible X-ray tomography using the ASTRA Toolbox," *Optics Express* **24**(22), 25129–25147 (2016).