

Resampling density values on R -lines into density values on a Cartesian grid

Stanislav Žabić, Stefan Hoppe, Frank Dennerlein, Günter Lauritsch and Frédéric Noo

Abstract— Three dimensional cone-beam reconstruction methods based on the differentiated backprojection accurately reconstruct objects only along measured lines. Thus, the values on a Cartesian grid need to be interpolated from the known data. The quality of the final reconstruction result depends on the chosen interpolation method. In this work, we discuss three solutions to this interpolation problem, compare them, quantify their resolution property and discuss their computational effort. Two of these solutions are original. Methods are tested on simulated data of the ForBilb head and thorax phantoms. Three different source trajectories are investigated: helix, saddle and circle-plus-line. Our results suggest that a carefully chosen interpolation method considerably reduces the computational effort in the reconstruction algorithm while maintaining the image quality.

I. INTRODUCTION

Recently, new cone-beam (CB) reconstruction techniques using differentiated backprojection have been developed [1], [3]– [9]. These techniques are attractive because they can be used for accurate reconstruction of specific regions-of-interest from truncated data. However, they cannot produce directly the sought density function on a Cartesian grid. They first produce the density function on a set of measured lines, and the values on the desired Cartesian grid have to be obtained from there using interpolation. In this work, we consider three solutions to this interpolation problem, compare them, quantify their resolution property and discuss their computational effort. Two of these solutions are original, while the third solution was suggested in [1]. The discussion is focused on interpolation from density values on R -lines – lines in the euclidean space that connect two source positions and are thus measured twice. Extension to general measured lines is however straightforward.

II. THREE METHODS OF INTERPOLATION

For all methods outlined here, we assume that the density values available on each R -line are finely sampled, typically with a sampling distance equal to the sampling distance of the desired Cartesian grid. This assumption is in agreement with the common way to implement a differentiated backprojection.

S. Žabić, F. Dennerlein and F. Noo are with the Department of Radiology, UCAIR, University of Utah, Salt Lake City, Utah, USA. S. Hoppe is with the Institute of Pattern Recognition, University of Erlangen/Nuremberg, Erlangen, Germany. G. Lauritsch is with the Siemens AG, Medical Solutions, Forchheim, Germany.

This work was partially supported by the US National Institutes of Health (NIH) grant R01 EB000627. Its contents are solely responsibility of the authors and do not necessarily represent the official views of the NIH.

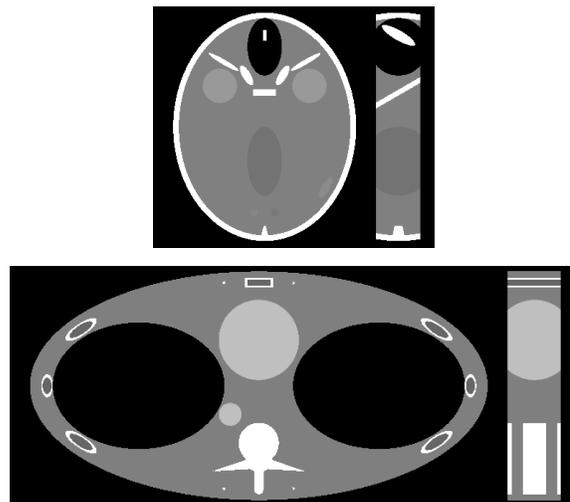


Fig. 1. “Ground truth” images of the ForBilb phantoms. Head phantom without ears is on the top (grayscale window [1, 1.1]) and the thorax phantom is on the bottom (grayscale window [9 1.1]). For each image, central axial slice is on the left and central sagittal slice is on the right.

Method 1: Six Closest R -Lines. In this approach, the value at each point X of the desired Cartesian grid is approximated as an average of six values coming each from one of the six R -lines closest to X . Interpolation weights dependent on the distance from X to the R -lines are used in this average. Let $\{T_i \mid i = 1, \dots, 6\}$ be the set of (orthogonal) projections of X on each of the six closest R -lines. The value of the density function at T_i (denoted here $v(T_i)$) is easily and accurately obtained using linear interpolation of the density values available on the i -th R -line. Assuming the distance d_i from X to the i -th closest line for all i is non-zero, the density value at X is approximated by

$$v(X) := \frac{\sum_{i=1}^6 d_i^{-1} v(T_i)}{\sum_{i=1}^6 d_i^{-1}}.$$

If there exists an i such that $d_i = 0$, then $v(X) := v(T_i)$. We take six R -lines to account for two directions coming from each of the three vectors in the orthogonal basis of \mathbb{R}^3 . However, this algorithm does not guarantee that the points T_i will be spread evenly around X .

Method 2: Six Directed R -Lines. The attempt here is to evenly spread the interpolated values around X in six directions. In a discretized Cartesian grid, each point X can be viewed as the center of a cuboid (rectangular parallelepiped) of dimension $\Delta x \times \Delta y \times \Delta z$, where Δx , Δy and Δz are the sampling distances in the direction of the Euclidean basis

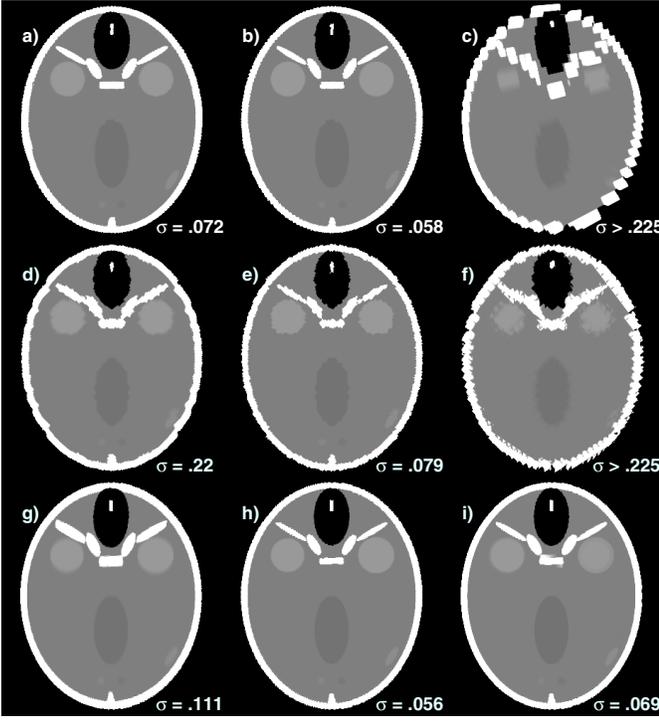


Fig. 2. Results obtained using the head phantom, axial view, central slice. Top row: helix, middle row: saddle and bottom row: circle-plus-line trajectory. Images a), d) and g) are obtained using method 1, b), e) and h) using method 2 and c), f) and i) by method 3. Images are represented in scale [1, 1.1]. Values of the minimizing standard deviation are marked with σ and given in centimeters.

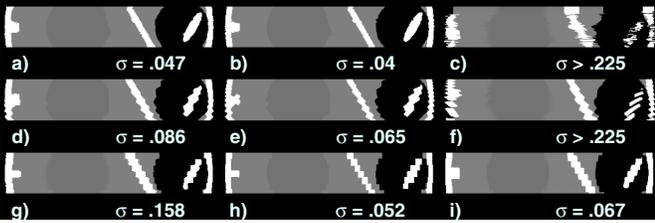


Fig. 3. Results obtained using the head phantom, sagittal view, central slice. Top row: helix, middle row: saddle and bottom row: circle-plus-line trajectory. Images a), d) and g) are obtained using method 1, b), e) and h) using method 2 and c), f) and i) by method 3. Images are represented in scale [1, 1.1]. Values of the minimizing standard deviation are marked with σ and given in centimeters.

vectors, \underline{e}_1 , \underline{e}_2 and \underline{e}_3 respectively. Let $\{M_i \mid i = 1, \dots, 6\}$ be the points at the center of each of the six faces of the voxel centered on X . We first estimate the density value at M_i using the density value at the orthogonal projection of M_i onto the R -line that is closest to M_i . This estimate is obtained using linear interpolation between the density values available on this closest R -line. Then, the density value at X is obtained as the equally-weighted average of values estimated at $\{M_i \mid i = 1, \dots, 6\}$.

Method 3: Interpolation between the R -line surfaces. The R -lines originating from a given source form a surface which we call an R -line surface. The surface is non-planar in general. The method transforms the values from the R -lines into Cartesian coordinates by interpolating between the R -line surfaces. Each value at a point on an R -line surface is

obtained using bilinear interpolation of the density values at four sampled locations lying on the two closest R -lines. With this understanding, the value at (x, y, z_0) for any given z_0 is obtained through linear interpolation of the density value from the closest R -surface above (x, y, z_0) in the z direction and the other value from the closest R -surface below (x, y, z_0) in the z direction. This method, which was suggested in [1] is primarily designed for source trajectories drawn on a cylinder centered on the z axis.

III. EXPERIMENTAL FRAMEWORK

The interpolation methods were tested on two ForBilD phantoms, the head without ears and the thorax, for three source paths: helix, saddle and circle-plus-line. Although the sampling distance Δt on the R -lines was chosen to be quite small for all three paths ($\Delta t = 0.075$ cm), we do not require such a fine sampling on the source positions. For the helical and saddle trajectories, R -lines were obtained using the pairs of source positions that can be created with 200 samples per helix turn and 240 source positions on the saddle. The radius and pitch of the helix were $R = 57$ cm and pitch $p = 3$ cm respectively, while the radius and height of the saddle were $R = 57$ cm and $h = 3$ cm respectively (see equation (16) in [2]). Picturing the field-of-view (FOV) as a centered cylinder of radius 25cm, the number of relevant R -lines (i.e. the R -lines passing through the FOV) that originated from any sampled source position was 58 for the helix and 70 for the saddle trajectory. For the circle-plus-line, the R -lines were defined by those that connect the sources on the line with the sources on the circle. The circle had a radius of 57cm and was located in the plane $z = -2$ cm. The length of the line was $L = 10$ cm. There were 1600 sources on the circle and 7 sources on the line, so that in this geometry 463 R -lines passed through the FOV.

The density values on an R -line were simulated from the definition of the phantoms in the following way:

- Each sample on an R -line is taken to be the average of four samples, each coming from one of four lines in four orthogonal directions: above, below, to the left and to the right. These four lines are parallel to the R -line at distance $\Delta t/2$.
- Any sample on each of these four lines is computed as the average of two subsamples a distance $\Delta t/2$ from each other.

The simulated R -line density values were then interpolated using the three methods described in Section II. The interpolation was done to obtain the density values in volumes partially containing the FOV. Separate images were created by taking the central slice in axial and sagittal view of the interpolated volume using $\Delta x = \Delta y = \Delta z = 0.075$ cm (the spacing was taken to be the same as Δt). The resolution of each interpolated image, denoted f_k , $k = 1, 2, \dots, N$, was compared to the “ground truth” image f by finding the value $\sigma_k > 0$ for which

$$\min_{\sigma_k > 0} \left\| \frac{1}{2\pi\sigma_k^2} f * e^{-(x^2+y^2)/2\sigma_k^2} - f_k \right\| \quad (1)$$

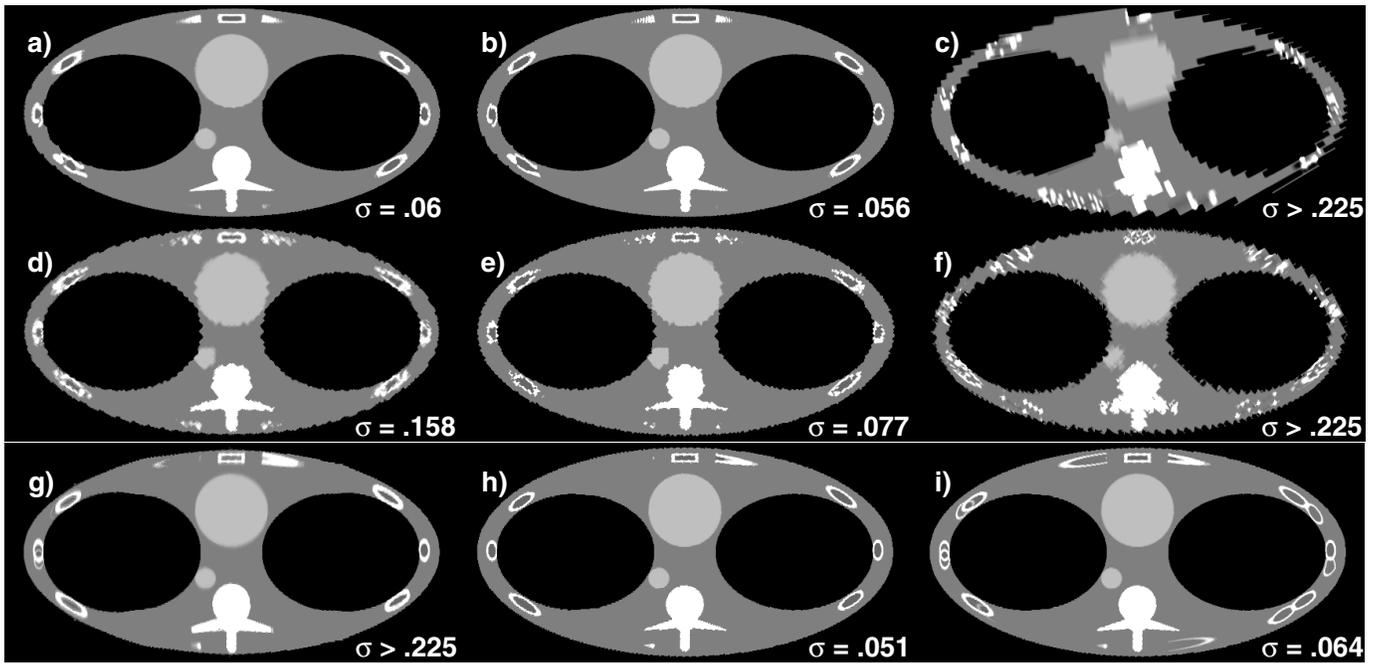


Fig. 4. Results obtained using the thorax phantom, axial view, central slice. Top row: helix, middle row: saddle and bottom row: circle-plus-line trajectory. Images a), d) and g) are obtained using method 1, b), e) and h) using method 2 and c), f) and i) by method 3. Images are represented in scale $[.9, 1.1]$. Values of the minimizing standard deviation are marked with σ and given in centimeters.

is attained. Here, the original image f was chosen to be the central slice through the mathematical phantom f (Fig. 3) with each pixel value obtained as the average of $2 \times 2 \times 2$ subsamples.

In terms of adaptive filtering, the convolution in (1) means blurring the image f using Gaussian low pass filter. The parameter σ_k is the standard deviation of the Gaussian filter which needs to be applied to the “ground truth” image f so that the filtered image minimally differs from the computed image f_k . The smaller the value of σ_k , the closer the image f_k is to image f in terms of resolution.

IV. RESULTS AND DISCUSSION

Images obtained in our experiments are presented in Figures 2, 3, 4 and 5. When compared with the “ground truth” image f , the lowest values of σ in (1) were attained always for method 2. Values of σ obtained from method 1 and method 2 are less than the values of σ obtained for method 3 except in case of circle-plus-line trajectory. For the circle-plus-line trajectory, the value of σ for method 3 were comparable to the values obtained using method 2, but we found that the method 3 returned images with incorrect density values for some of the objects within the phantoms.

Method 3 performed well in [1] because the R -lines were much more finely sampled than in this report. Here, the helix and saddle scenario for a fixed source position had less than 70 R -lines going through the object per sampled source position. This produced R -line surfaces with a very low resolution. These low resolution surfaces are then used in another interpolation which was performed in only two directions; \underline{e}_3 and $-\underline{e}_3$. The new interpolation methods interpolate between

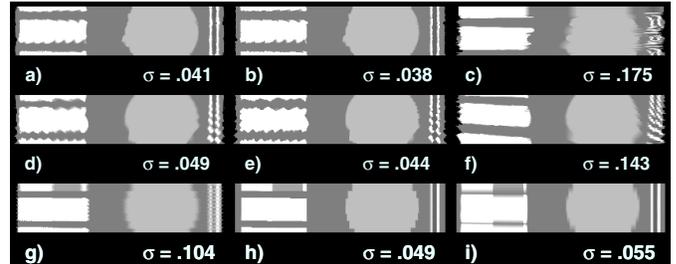


Fig. 5. Results obtained using the thorax phantom, sagittal view, central slice. Top row: helix, middle row: saddle and bottom row: circle-plus-line trajectory. Images a), d) and g) are obtained using method 1, b), e) and h) using method 2 and c), f) and i) by method 3. Images are represented in scale $[.9, 1.1]$. Values of the minimizing standard deviation are marked with σ and given in centimeters.

values from six directions all around each desired target point, disregarding the R -line surfaces.

For the circle-plus-line trajectory, there were in each R -line surface much more R -lines going through the object, so the images obtained by method 3 seem as sharp as those produced by method 2. However, method 3 produced images with incorrect densities for objects in certain regions. The cause of the wrong densities lies in the sparse spacing of the line portion of the path, which led to a large distance between successive R -line surfaces. For example, in figure 4i), each voxel in the (right) rib area was interpolated between the values on two R -line surfaces with z coordinates being far away from the considered voxel. This explains why we see three additional ribs from the slices with higher z coordinates. Additionally, corresponding sagittal view obtained by method 2 (Figure 5h) is clearly closer to the “ground truth” image than the image obtained by method 3 (Figure 5i).

Images produced by method 1 have a lower quality than images produced by method 2 because the R -lines participating in interpolation are in general better distributed in method 2 than in method 1. Method 1 simply takes the closest R -lines for the interpolation, while these are not necessarily evenly distributed around the voxel. Method 2 involves not only the distance from the R -lines to the voxel, but also their orientation. See Fig. 6a) for an illustration (X , T_i and M_i are defined in Section II and V_i are orthogonal projections of M_i to the closest R -lines). One important assumption is that the dimension of the voxels is selected so that not too many R -lines pass through the voxel cube. Fig. 6b) illustrates that if this dimension is chosen to be too large, then method 1 has an advantage over method 2. Method 1 will find a better value for voxel X as it will interpolate between the values on R -lines r_1 , r_2 , r_3 and r_4 . Method 2 will interpolate between the values on R -lines r_2 , r_3 , r_5 and r_6 . However, this also means that the voxel size could be chosen to be smaller, in which case an image with higher resolution and better quality is produced by method 2.

An efficient implementation of method 1 and method 2 is obtained by sorting the closest R -lines and interpolating only for the voxels that belong to a small, yet large enough cylinder around each R -line. Details on the efficient implementation and estimation of the size of the cylinder will be published elsewhere. In this report, both methods are implemented in C on a computer with single central processing unit AMD Athlon™ 64 Processor 3800+ and clock speed of 2400 MHz. Table I gives execution times for the two interpolation methods introduced in this paper. For each experiment, the first column contains amount of voxels and second column contains amount of R -lines participating in the interpolation. Third and fourth column give the execution times for method 1 and method 2 respectively. All times are given in minutes.

V. CONCLUSIONS

In a brute force implementation, the number of operations needed for methods 1, 2 and 3 are respectively proportional to

- $6N_R N_x N_y N_z$,
- $4N_R(N_x + 1)(N_y + 1)N_z$ and
- $N_s N_x N_y N_z$,

where N_R is the number of R -lines, $N_x \times N_y \times N_z$ is the size of the required output volume, and N_s is the number of source positions. For the helix, $N_R = N_s(N_s - 1)$, for the saddle $N_R = N_s^2/2$ and for the circle-plus-line, $N_R = N_{sc}N_{sl}$, where N_{sc} is the number of sources on the circle and N_{sl} is the number of sources on the line. In appearance methods 1 and 2 are much more demanding in terms of computational effort. However, the efficient implementation does not consider all voxels for each R -line, but only those that lie close to it. This dramatically reduced the number of operations required for execution of both algorithms.

The two interpolation methods introduced in this report perform with similar speed, whereas method 2 obtains images with the highest quality. Both methods reveal a strong potential for a large decrease in the number of R -lines onto which

TABLE I
EXECUTION TIMES FOR THE TWO NEW INTERPOLATION METHODS

Scenario	Number of voxels	Number of relevant R -lines	Execution time (in seconds)	
			method 1	method 2
thorax, helix	$580 \times 380 \times 62$	17429	510	687
thorax, saddle	$580 \times 380 \times 62$	8435	198	287
thorax, cpl*	$580 \times 380 \times 62$	3241	86	92
head, helix	$312 \times 338 \times 62$	10217	225	261
head, saddle	$312 \times 338 \times 62$	4820	75	90
head, cpl*	$312 \times 338 \times 62$	1897	51	45

*cpl = circle-plus-line

backprojection and deconvolution needs to be achieved according to the reconstruction steps presented in [1]. This could ultimately decrease the total computational effort required for reconstruction. Tying up the reconstruction steps with the new interpolation method will be the topic of our future research. Preliminary results also suggest that method 2 has certain advantages over method 1. The resulting images obtained by method 2 are interpolated between the values that are more evenly spread around the voxels and therefore produce less distortion. Table I shows that our implementation of method 2 takes a slightly longer time to execute than the implementation of method 1.

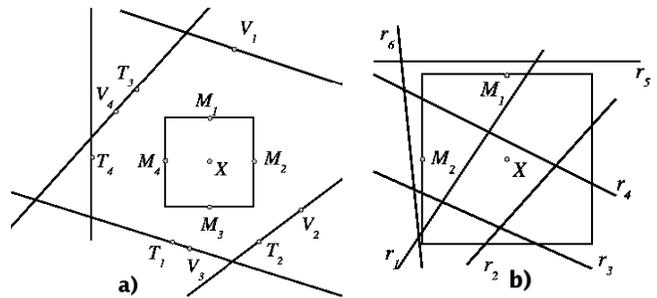


Fig. 6. Figure a) illustrates the distribution of interpolation points for method 1 and method 2. Figure b) illustrates that if the dimension of the voxels is chosen to be large, then method 1 performs better than method 2.

REFERENCES

- [1] J. D. Pack, F. Noo, R. Clackdoyle, "Cone-Beam Reconstruction Using the Backprojection of Locally Filtered Projections", *IEEE Trans Med Imaging*, 2005 Jan; 24(1):70-85.
- [2] J. D. Pack, F. Noo, H. Kudo, "Investigation of saddle trajectories for cardiac CT imaging in cone-beam geometry", *Phys. Med. Biol.*, 2004; 49:2317-36.
- [3] K. Ramamurthi, N. Strobel, R. Fahrig, J.L. Prince, "Fully Truncated Cone-beam Reconstruction on Pi Lines Using Prior CT", *MICCAI* 2005: 631-638
- [4] Y. Ye, S. Zhao, H. Yu, G. Wang, "A General Exact Reconstruction for Cone-Beam CT via Backprojection-Filtration", *IEE Trans Med Imaging*. 2005 Sep; Vol 24, No 9: 1190-98.
- [5] H. Yu, Y. Ye, S. Zhao, G. Wang, "A backprojection-filtration algorithm for nonstandard spiral cone-beam CT with an n-PI-window". *Phys Med Biol*. 2005 May 7;50(9):2099-1111.
- [6] Y. Zou, X. Pan, "Exact image reconstruction on PI-lines from minimum data in helical cone-beam CT", *Phys. Med. Biol.*, 2004 Mar 21; 49(6):941-59.

- [7] Y. Zou, X. Pan, "An extended data function and its generalized backprojection for image reconstruction in helical cone-beam CT", *Phys. Med. Biol.*, 2004 Nov 21; 49(22):N383-7.
- [8] S. Zhao, H. Yu, G. Wang, "A unified framework for exact cone-beam reconstruction formulas", *Med. Phys.*, 2005 June; 32(6):1712-21.
- [9] T. Zhuang, S. Leng, B.E. Nett, G.H. Chen, "Fan-beam and cone-beam image reconstruction via filtering the backprojection image of differentiated projection data", *Phys. Med. Biol.*, 2004 Dec 21; 49(24):5489-503.