Fast distance-driven projection and truncation management for iterative cone-beam CT reconstruction

Simon Rit and Marcel van Herk and Jan-Jakob Sonke

***Abstract*—The purpose of this study was to improve the speed of the Simultaneous Algebraic Reconstruction Technique (SART) for truncated cone-beam (CB) projections using a distance-driven forward projector. First, we optimized the innermost loop of the projector for a cone-beam scanner with a circular source trajectory and a flat-panel imager. We showed that the innermost loop of the distance-driven projection reduces to the mapping of two segments of samples with constant spacings. Second, we implemented the projector to handle a multi-resolution grid in the planes orthogonal to the rotation axis in order to reduce the overhead required to manage the truncation of the cone-beam projections. The distance-driven projector was 1.4 times faster than an optimized ray-driven projection with Joseph interpolation. Moreover, the multi-resolution reduced 4 times the overhead required to handle an half-truncated Shepp-Logan phantom.**

***Index Terms*—Optimization, distance-driven projection, cone-beam reconstruction, truncation.**

I. Introduction

In the past few years, the use of cone-beam (CB) computed tomography (CT) has rapidly grown for different clinical applications, e.g. the acquisition of CT images in the treatment room for patient positioning [1]. In-room acquisition requires efficient CB reconstruction to provide a CT image to the operator within a few seconds after the end of the acquisition. This requirement has logically led to the use of analytical methods, more specifically to the Feldkamp algorithm [2] for circular source trajectories, and hindered the use of iterative techniques. However, iterative techniques have different properties that can turn to their advantage in some situations. Therefore, faster iterative reconstruction techniques could increase their clinical relevance. This contribution is a step in this direction and focuses more specifically on algebraic reconstruction.

The algebraic reconstruction technique (ART) has been one of the first CT reconstruction techniques proposed [3]. Starting from an initial guess of the CT image (generally 0 for all samples), the algorithm iterates over each pixel of the CB projections and updates the reconstructed image in three steps: (1) project the current image along the ray corresponding to the sample, (2) compute the difference between the estimated and the acquired projection and (3) backproject the estimated difference in the image. In [4], Andersen and Kak proposed an improvement of ART, the Simultaneous ART (SART), which is a block version of ART: an update process simultaneously all rays of a given CB projection. Its implementation simplicity and its computational efficiency have brought SART to be one of the main iterative reconstruction techniques used nowadays.

However, SART remains computationally expensive compared to Feldkamp algorithm. The main reason is its higher algorithmic complexity due to the additional forward projection (step 1) and the need to go over each CB projection a few times. These issues were not investigated in this study because they can not be addressed without changing the core of ART. This study identifies and addresses two other reasons: the computational cost of the forward projection (step 1) compared to the backprojection (step 3) and the management of truncation.

II. Fast distance-driven projection

Forward projection of a CT volume produces a Digitally Reconstructed Radiograph (DRR). Different methods have been proposed for DRR generation which can be classified in two main approaches: voxel-driven and ray-driven [5]. The latter has often been preferred because it produces aliasing-free DRRs but its computational cost is quite high compared to backprojection which can now be implemented *hyperfast* [6].

Hyperfast backprojection takes advantage of the alignment of one axis of the CT volume with one axis of the projection image in a circular geometry (Fig. 1). This alignment implies that all voxels belonging to a segment of the CT volume parallel to this axis projects on a segment of the DRR parallel to the same axis. One can take advantage of this property to optimize both the computation and the memory alignment of the innermost loop of the backprojection [6].

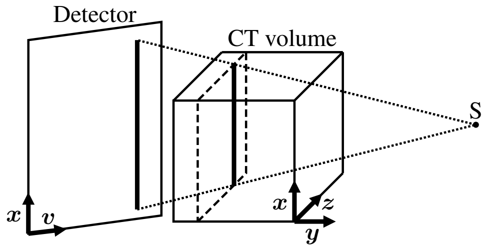

Fig. 1. Geometry of the scanner: we assumed a circular source trajectory in a plane orthogonal to $\boldsymbol{x}$. The axis $\boldsymbol{x}$ is shared by the volume and the detector which allows an efficient innermost loop for distance-driven projection.

S. Rit, M. van Herk and J.-J. Sonke are with the Department of Radiation Oncology, The Netherlands Cancer Institute-Antoni van Leeuwenhoek Hospital, Plesmanlaan 121, 1066 CX Amsterdam, The Netherlands `j.sonke@nki.nl`

Using the same property in a forward projection algorithm is not straightforward. Ray-driven methods are designed to run along the ray corresponding to each DRR pixel which will generally not align with any axis of the CT volume due to the beam divergence. A voxel-driven strategy can be used instead but it is subject to aliasing in simple implementations, e.g. splatting each CT voxel on the DRR using a constant kernel size. De Man and Basu have proposed an alternative solution, the *distance-driven projection* [5], which bijectively maps each slice of the CT volume to the DRR. The bijection produces aliasing-free forward and back-projection but also allows the choice of the axis for the innermost loop. We used this property to implement efficiently a distance-driven forward projection.

We assumed a circular trajectory of the source orthogonal to the CT volume and selected the $\boldsymbol{x}$ axis for the innermost loop for the above-mentioned reasons (Fig. 1). The optimization of the innermost loop comes then to an efficient mapping of a scaled segment of the CT volume with constant voxel width `vw` to a segment of the DRR, also with a constant pixel width `pw` (Fig. 2).

We distinguished two cases: shrinking (`vw<pw`) and stretching (`vw>=pw`) the segment of CT voxels. Both cases address the splitting of a CT voxel in two DRR pixels in the same way: (1) a fraction of the current voxel value corresponding to the distance `dx` between the inferior corner of the current CT voxel and the superior corner of the current DRR pixel is accumulated in the current DRR pixel, (2) the rest `vw-dx` is accumulated in the next DRR pixel, and (3) the distance `dx` for the next voxel is computed by taking the complement `pw-(vw-dx)`. Moreover, case 1 addresses the specific case of voxels that do not split but entirely map to one pixel, i.e. `dx>vw`, and case 2 addresses the specific case of voxels that split in more than 2 pixels, i.e. `vw-dx>pw`. Listing 1 details our optimized implementation of this innermost loop. The rest of the projection algorithm will be described in the following because it also incorporates efficient truncation management.

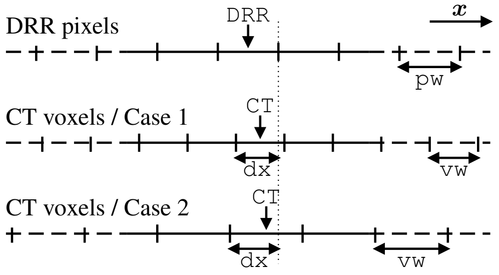

Fig. 2. Graphical illustration of cases and variables for Listing 1. Case 1 and 2 are respectively a shrinking and a stretching of a segment of the CT volume to a segment of the DRR.

Listing 1. Bijective mapping of `n` voxels of a segment of the CT volume with constant `vw` width to pixels of the DRR with constant `pw`.

```
void MapSeg(float *CT,   // Pointer to first CT voxel
            float *DRR,  // Pointer to first DRR pixel
            int   n,     // Number of voxels
            float dx,    // Distance between:
                         // - current voxel inferior corner
                         // - current pixel superior corner
            float vw,    // Projected voxel width
            float pw,    // Pixel width
            float f)     // Voxel correction factor
{
  int i;                 // Voxel index
  float vv;              // Voxel value weighted with f

  if (vw<pw)             /***   Case 1: 0 or 1 split   ***/
  { for(i=0; i<n; i++)   // Loop over the voxels
    { vv = (*CT)*f;      // Get and weight voxel value
      if (dx>vw)
      {                  // Voxel projects in 1 pixel only
        *DRR += vv*vw;   // Accumulate full voxel
        dx -= vw;        // Update distance for next voxel
      }
      else
      {                  // Voxel splits in two pixels
        *DRR += vv*dx;   // Accumulate first part of voxel
        dx = vw-dx;      // Compute rest of voxel width
        ++DRR;           // Go to next pixel
        *DRR += vv*dx;   // Accumulate second part of voxel
        dx = pw-dx;      // Update distance for next voxel
      }
      ++CT;              // Go to next voxel
    }
  }
  else                   /*** Case 2: 1 or more splits ***/
  { for(i=0; i<n; i++)   // Loop over the voxels
    { vv = (*CT)*f;      // Get and weight voxel value
      *DRR += vv*dx;     // Accumulate first part of voxel
      dx = vw - dx;      // Compute rest of voxel width
      ++DRR;             // Go to next pixel
      while(dx>=pw)
      {                  // Rest covers an entire pixel
        *DRR += vv*pw;   // Accumulate a pixel of voxel
        dx -= pw;        // Compute rest of voxel width
        ++DRR;           // Go to next pixel
      }
      *DRR += vv*dx;     // Accumulate last part of voxel
      dx = pw-dx;        // Update distance for next voxel
      ++CT;              // Go to next voxel
    }
  }
}
```

III. Efficient truncation management

Truncation of the CB projections is a common problem with in-room CBCT due to the limited size of the flat panels. In theory, Feldkamp can not reconstruct the image of an object when its CB projections are laterally truncated but, in practice, feathering strategies correct most of the artifacts [7], and their cost is limited because they only act on the 2D CB projections. On the contrary, iterative reconstruction is theoretically less limited by truncation [8] but it requires a larger field-of-view (FOV) during reconstruction (*Reconstruction FOV*) than the usable FOV after reconstruction (*CT FOV*).

The Reconstruction FOV is the part of space which is hit at least once by the X-ray beam while the CT FOV is the part of space which is hit by the X-ray beam for every source position (Fig. 3). Both FOVs are circles in the planes parallel to the source trajectory but the radius of the Reconstruction FOV is fixed by the isocenter-detector distance and can be much larger than the CT FOV which is fixed by the detector size at the isocenter. Of course, the Reconstruction FOV can be reduced with *a priori* information on the maximum patient dimensions but it will generally remain significantly larger than the CT FOV.

For iterative reconstruction, the CT grid must encompass the reconstruction FOV during reconstruction which can lead to a huge number of voxels when using a fine resolution. However, the more the voxels are away from the CT FOV, the less they are used during reconstruction. Therefore, we used a multi-resolution grid with a progressively lower resolution for voxels outside the CT FOV. An example of such a grid is provided in Fig. 3b.

Most previous contributions dealing with fast forward and back-projection assumed that the CT volume is represented on a regular grid, i.e. that the CT voxels are regularly spaced along $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ (Fig. 1). Of course, using a regular spacing simplifies the computation per voxel so we kept a regular spacing along $\boldsymbol{x}$ (Fig. 3a) to preserve our simple innermost loop (Listing 1). Nevertheless, we adapted the projection to handle a set of segments arbitrarily spaced along $\boldsymbol{y}$ and $\boldsymbol{z}$, e.g. a rectilinear grid (Fig. 3b). This set is represented by a cloud of points containing the 3 spatial coordinates of the inferior and superior corners Ci and Cs of each CT segment parallel to $\boldsymbol{x}$ (Fig. 3a) as well as the position of the corresponding voxel index in the CT volume.

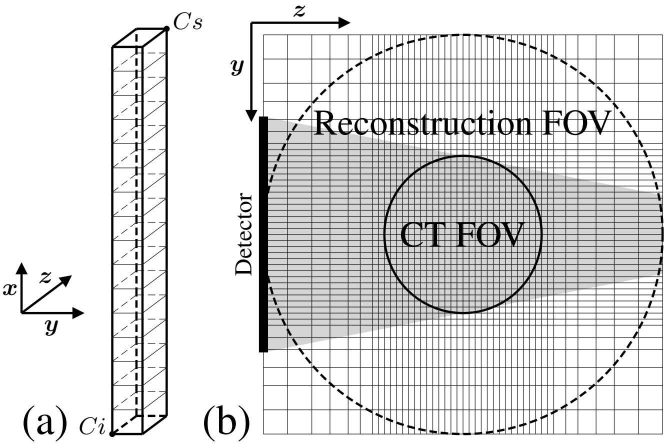

Fig. 3. (a) Example of a segment of voxels along the $\boldsymbol{x}$ axis (Fig. 1): the coordinates of its corners C_i and C_s are passed to the reconstruction algorithm in order to know its dimensions and its location. (b) Rectilinear grid used during reconstruction from truncated data to cover the CT FOV with a fine resolution and the Reconstruction FOV with a gradually coarser resolution.

The main loop of the projection goes over all segments of this cloud. For each segment, its corners Ci and Cs are projected on the DRR. Similarly to [5], the segment projects to a rectangle which sides are parallel to $\boldsymbol{x}$ and $\boldsymbol{v}$. Depending on its dimensions, the rectangle is then mapped to one or more segments of DRR pixels using the function `MapSeg` which parameter `f` is set to the fraction along $\boldsymbol{v}$ multiplied by the length of the intersection between the ray and the segment.

IV. Experiments

We analytically simulated 640 CB projections of the Shepp-Logan phantom regularly spaced around a full revolution using a 1 m source-isocenter distance to reproduce a typical acquisition of an existing CBCT scanner [1]. The resolution of the CB projections was 512×512 pixels of 0.5×0.5 mm^2 at the isocenter.

The performance of our distance-driven projector was evaluated using a CT volume having $512 \times 512 \times 512$ voxels of $0.5 \times 0.5 \times 0.5$ mm^3 which computation requires 80 Giga-voxel Updates (GUP). DRRs were computed for every angle corresponding to the 640 CB projections. The performance was compared to local implementations of an optimized ray-driven algorithm using Joseph interpolation [9] with adequate orientation of the volume [10] and an optimized voxel-driven backprojection [6]. Only single-threaded implementations were evaluated because the full (back-)projections were sequentially computed in each direction which is known to be mainly limited by communication bandwidth [6]. The test was performed on an Intel Core Duo 2.1 GHz processor.

The truncation management was evaluated by generating a new set of CB projections with the center of the Shepp-Logan phantom shifted to the border of the CT FOV in the $\boldsymbol{y}$ direction. First, reference images were reconstructed using the same $512 \times 512 \times 512$ grid. Second, the superior half of the grid along $\boldsymbol{y}$ (part in the Reconstruction FOV but not in the CT FOV) was replaced with a multi-resolution grid. The voxel width along $\boldsymbol{y}$ was increased by a factor 2 every 16 voxels such that only 64 voxels covered the space normally occupied by 256 voxels. All CT images were reconstructed using 3 iterations of SART and a 0.2 convergence factor.

V. Results

Table I summarizes the speed of the different operators evaluated in this study. Backprojection is hyperfast because its innermost loop requires very little operations per voxel and uses an optimized memory layout [6]. In contrast, the ray-driven implementation is 4.5 times slower because the memory layout is less optimal and it requires more computation. Finally, our distance-driven operator is only 3.3 times slower than backprojection, i.e. 1.4 times faster than the ray-driven implementation because it uses the same memory layout as the backprojection and its innermost loop remains sufficiently simple (Listing 1). The overhead required to manage a rectilinear grid along $\boldsymbol{y}$ and $\boldsymbol{z}$ is limited as the innermost loop along $\boldsymbol{x}$ still consumes more than 90% of the total time.

TABLE I
Speed in GUP/s of the 3 operators evaluated in this work.

Voxel-driven backprojection	0.236
Ray-driven projection	0.052
Distance-driven projection	0.072

Fig. 4 shows slices of the reconstructed images. When there is no truncation, the full Shepp-Logan phantom is adequately reconstructed in the CT FOV. We observe noise introduced by the edges of the phantom but it is a known artefact of both ray-driven and distance-driven projectors [11]. When there is truncation, the Shepp-Logan is adequately reconstructed in the CT FOV although it is less accurate at the edge of the CT FOV. Finally, the multi-resolution correction of the truncation reconstructs a very similar CT image but with $512 \times 320 \times 512$ voxels instead of $512 \times 512 \times 512$. The small ellipsoids are still well defined and only the edge noise is more pronounced.

VI. Discussion and conclusion

In this paper, we presented an implementation of SART with an optimized distance-driven operator and management of the truncation of the CB projections.

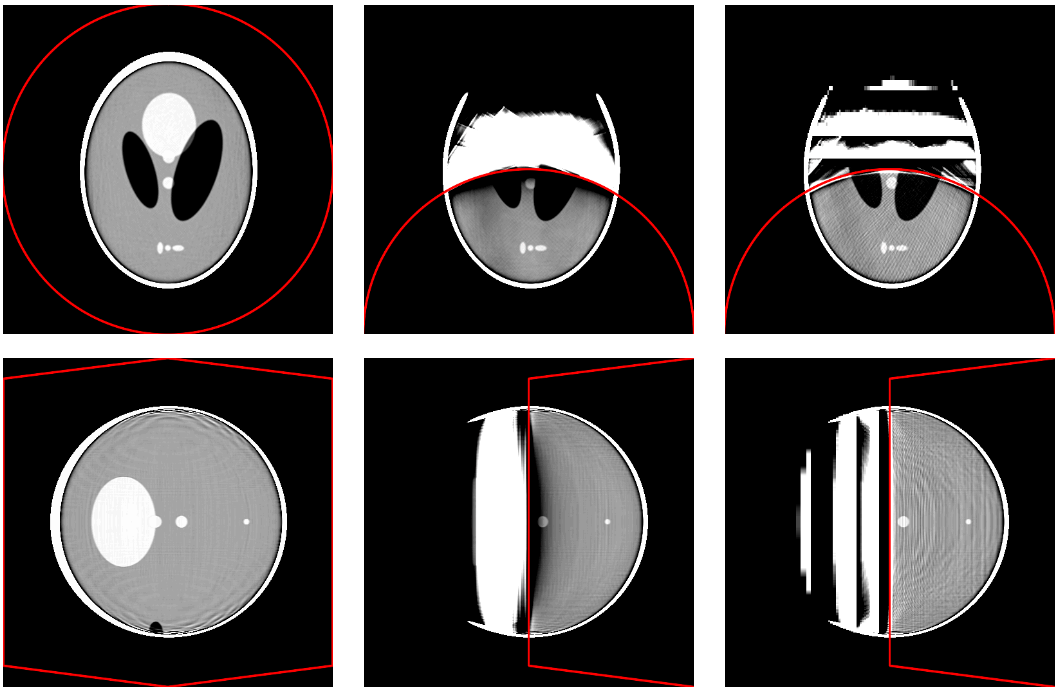

Fig. 4. Slices of the reconstructed images. Top: axial slices. Bottom: sagittal slices. Left: reconstruction without truncation. Middle: reconstruction with truncation. Right: reconstruction with truncation and a multi-resolution grid. The lines indicate the limit of the CT FOV. Greyscale window: [1.00, 1.03].

First, we showed that the innermost loop of the distance-driven algorithm can be efficiently implemented if it runs along the rotation axis x in contrast to what other papers have reported, e.g. [12]. Indeed, all the voxels having the same y and z coordinates project with a constant width along x on the detector. Using this property, we developed an optimized mapping of two parallel segments with constant spacings of their samples (Fig. 2, Listing 1). Therefore, we obtain a faster implementation than our optimized implementation of a ray-driven projection with Joseph interpolation. Moreover, in contrast to ray-driven projection [10], this implementation does not require flipping the volume depending on the source angle which is an advantage for SART. Note that faster forward and back-projections can be obtained with multithreading as well as with additional tricks to optimize the flow of data through the memory cache [10].

Second, we used a multi-resolution grid in the y and z directions to correct for the truncation of the CB projections. We evaluated the algorithm on a half-truncated Shepp-Logan phantom and observed similar convergence with 4 times less voxels in the part added to the CT FOV to correct for the truncation. However, the edge noise is enhanced because the larger voxels of the Reconstruction FOV have an influence on the CT FOV. Among other solutions, future works will include adequate prefiltering of the CB projections [13] or more constrained iterative reconstruction techniques, e.g. the conjugate gradient method.

The object size was known in advance in this case and the reconstruction grid tailored to fit the object but it is not possible in practice. If truncation can be expected at every side of the object, the multi-resolution grid schematically represented in Fig. 3 reduces even more significantly the overhead required for full coverage of the Reconstruction FOV.

References

[1] D. Jaffray, J. Siewerdsen, J. Wong, and A. Martinez, “Flat-panel cone-beam computed tomography for image-guided radiation therapy,” *Int J Radiat Oncol Biol Phys*, vol. 53, no. 5, pp. 1337–1349, Aug 2002.

[2] L. Feldkamp, L. Davis, and J. Kress, “Practical cone-beam algorithm,” *J Opt Soc Am A*, vol. 1, no. 6, pp. 612–619, 1984.

[3] R. Gordon, R. Bender, and G. Herman, “Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography,” *Journal of Throretical Biology*, vol. 29, no. 3, pp. 471–481, Dec 1970.

[4] A. Andersen and A. Kak, “Simultaneous algebraic reconstruction technique (SART): a superior implementation of the art algorithm,” *Ultrason Imaging*, vol. 6, no. 1, pp. 81–94, Jan 1984.

[5] B. De Man and S. Basu, “Distance-driven projection and backprojection in three dimensions,” *Phys Med Biol*, vol. 49, no. 11, pp. 2463–2475, Jun 2004.

[6] M. Kachelriess, M. Knaup, and O. Bockenbach, “Hyperfast parallel-beam and cone-beam backprojection using the cell general purpose hardware,” *Med Phys*, vol. 34, no. 4, pp. 1474–1486, Apr 2007.

[7] B. Ohnesorge, T. Flohr, K. Schwarz, J. Heiken, and K. Bae, “Efficient correction for CT image artifacts caused by objects extending outside the scan field of view,” *Med Phys*, vol. 27, no. 1, pp. 39–46, Jan 2000.

[8] B. Zhang and G. Zeng, “Two-dimensional iterative region-of-interest (ROI) reconstruction from truncated projection data.” *Med Phys*, vol. 34, no. 3, pp. 935–944, Mar 2007.

[9] P. Joseph, “An improved algorithm for reprojecting rays through pixel images,” *IEEE Trans Med Imaging*, vol. 1, no. 3, pp. 192–196, 1982.

[10] M. Knaup and M. Kachelriess, “Acceleration techniques for 2D parallel and 3D perspective forward and backprojections,” in *Fully 3D Image Reconstruction in Radiology and Nuclear Medicine*, 2007.

[11] J. Sunnegardh and P.-E. Danielsson, “A new anti-aliased projection operator for iterative CT reconstruction,” in *Fully 3D Image Reconstruction in Radiology and Nuclear Medicine*, 2007.

[12] A. Ziegler, T. Köhler, T. Nielsen, and R. Proksa, “Efficient projection and backprojection scheme for spherically symmetric basis functions in divergent beam geometry,” *Med Phys*, vol. 33, no. 12, pp. 4653–4663, Dec 2006.

[13] H. Kunze, K. Stierstorfer, and W. Härer, “Pre-processing of projections for iterative reconstruction,” in *Fully 3D Image Reconstruction in Radiology and Nuclear Medicine*, Salt Lake City, Utah, USA, 2005, pp. 84–87.