Ray tracing through a grid of blobs

Lucretiu M. Popescu and Robert M. Lewitt

***Abstract*— In this paper we describe two ray tracing algorithms for images represented using spherically symmetric basis functions (blobs) on regular grids. The method presented here allows more realistic modeling of the forward projection by considering tube shaped kernels, rather than simple lines. Each kernel is a function of the radial distance r from its center and can vary with the position l along the projection line. The forward projections are computed by convolutions of the kernel with the blob line integrals. Both ray tracing procedures presented incrementally compute the square distance r^2 for each visited blob enabling the appropriate resolution kernel to be used. The second variant also computes the l coordinate along the line of response axis allowing for longitudinal variations of the resolution kernel to be considered as well as time-of-flight (TOF) modeling.**

I. Introduction

The forward projection operation involves computing the individual contributions of each volume element to each projection data element, and represents a critical step in many tomographic image reconstruction techniques. In the case of iterative algorithms, the computed values are assumed to be an accurate model of the data acquisition process. On the other hand, the accuracy of modeling is often limited because projection calculations are the dominant contribution to the reconstruction time. Therefore, much effort has been spent in designing fast ray tracing techniques [1–6] taking into account idealized physical models, but there remains a need for methods that combine the speed of these techniques with more realistic modeling. This is particularly true in the case of emission tomography, both PET and SPECT, where the nature of the data acquisition process is much more far from the ideal than in the case of CT, for example.

In this paper we describe a ray tracing algorithm for line-integral forward projection similar to one described in [7], and we present an extension of this method for more realistic models where the forward projection involves tube shaped kernels, rather than simple lines.

II. Object projection model

Let us consider the case of a PET scanner that reports a detected event — a line of response (LOR) — as the spatial coordinates of the coincidence detected photon pair $\mathbf{y} \equiv (\mathbf{a}, \mathbf{b})$, as sketched in Fig. 1. Due to the limited resolution of the detector as well as other factors such as the non-collinearity of the positron annihilation gamma rays, the reported positions are only approximate estimates, the true positions being distributed with a certain probability density around the reported values [8]. If the detectors locally have isotropic resolution the distributions are cylindrically symmetric. The resulting shape of the distribution $h(\mathbf{y}|\mathbf{x})$, the probability density of detecting an event in the point $\mathbf{y}$ due to an emission in the point $\mathbf{x}$, is also cylindrically symmetric if the attenuation around the symmetry axis is cylindrically symmetric, a fact that is implicitly assumed due to the limited resolution of the attenuation map. In the reference frame relative to the detected LOR $\mathbf{y}$, $h(\mathbf{y}|\mathbf{x})$ can be expressed as $K(r, l)$, where r and l are the radial and longitudinal coordinates of the point $\mathbf{x}$ relative to the LOR $\mathbf{y}$.

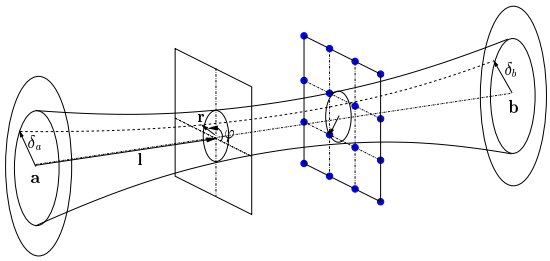

Fig. 1. Schematic representation of the shape of a line of response.

The problem consists in accurately determining the probability density of detecting an event in the point $\mathbf{y}$ due to an emission in the volume element $\mathcal{V}_i$, represented by a basis function $b_i(\mathbf{x})$

$$h_i(\mathbf{y}) = \int\limits_{\mathbf{x}\in\mathcal{V}_i} b_i(\mathbf{x}) h(\mathbf{y}|\mathbf{x}) \mathrm{d}\mathbf{x} \,. \tag{1}$$

Usually the image is represented using translation invariant basis functions $b_i(\mathbf{x}) = b(\mathbf{x} - \mathbf{x}_i)$ with $b(\xi) = 0$ for $\|\xi\| > B$ and $\int_{\|\xi\|\le B} b(\xi)\mathrm{d}\xi = 1$, the points $\mathbf{x}_i$ following the nodes of a grid. We also have $K(r, l) = 0$ for $r > D$, where B, D are positive real numbers.

For convenience we shift the reference frame relative to the LOR so that the grid node $\mathbf{x}_i$ is placed at the origin and the z axis is parallel with the LOR axis at the distance r in the plane xz, as shown in Fig. 2. For spherically symmetric basis functions (blobs) $b(\mathbf{x} - \mathbf{x}_i) = b(\|\mathbf{x} - \mathbf{x}_i\|)$, and $h_i(\mathbf{y})$ will depend only on the distances (r, l). If in addition we assume that $K(r, l)$ varies slowly with l compared with the

This work is supported by grants CA92060/EB002131 and CA93962/EB002135 from the National Institutes of Health, USA.

The authors are with University of Pennsylvania, Department of Radiology, 423 Guardian Drive, 4-th floor Blockley Hall, Philadelphia, PA 19104-6021, e-mail: {popescu,robert}@mipg.upenn.edu

Paper presented at *Nuclear Science Symp. Medical Imaging Conf.*, 2004, Rome, M10-220

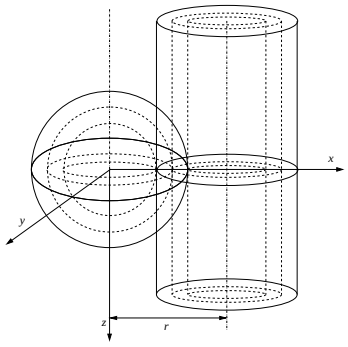

Fig. 2. Intersection between a projection kernel and a blob.

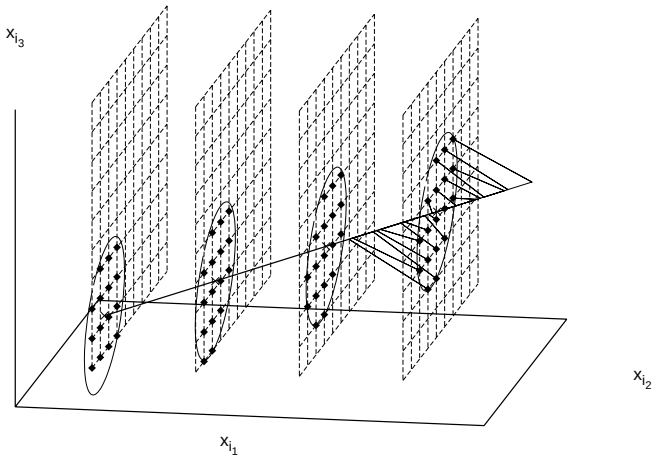

Fig. 3. Ray tracing with computation of the distances to the neighboring grid points.

blob variation then we have

$$h(r,l) = \int_{x_{\min}}^{x_{\max}} \mathrm{d}x \int_{-\sqrt{B^2-x^2}}^{\sqrt{B^2-x^2}} \mathrm{d}y\, K(\sqrt{(r-x)^2+y^2}, l) \times \int_{-\sqrt{B^2-x^2-y^2}}^{\sqrt{B^2-x^2-y^2}} \mathrm{d}z\, b(\sqrt{x^2+y^2+z^2}) \quad (2)$$

with $x_{\min} = \max(r-D, -B)$ and $x_{\max} = \min(r+D, B)$.

In this manner the problem of modeling the projection is reduced to tracing a straight line (the LOR axis) and determining the distances r and l for each grid point within a certain distance from the line as shown in Fig. 3.

III. The Ray Tracing Algorithm

A. First approach

The LOR is specified by the point $\mathbf{a} \equiv (a_1, a_2, a_3)$ and direction $\mathbf{u} \equiv (u_1, u_2, u_3)$, $\|\mathbf{u}\| = 1$. The square distance from an arbitrary point in space $\mathbf{x} \equiv (x_1, x_2, x_3)$ to the line defined by the point $\mathbf{a}$ and direction $\mathbf{u}$ is $r^2 = \|(\mathbf{x}-\mathbf{a}) \times \mathbf{u}\|^2$ which expanded gives us the following formula

$$\begin{aligned} r^2 &= [(x_2-a_2)u_3 - (x_3-a_3)u_2]^2 + \\ &\quad [(x_3-a_3)u_1 - (x_1-a_1)u_3]^2 + \\ &\quad [(x_1-a_1)u_2 - (x_2-a_2)u_1]^2 \quad (3) \\ &= r_{23}^2 + r_{31}^2 + r_{12}^2 \quad (4) \end{aligned}$$

where

$$\begin{aligned} r_{i_1 i_2} &= (x_{i_1} - a_{i_1})u_{i_2} - (x_{i_2} - a_{i_2})u_{i_1} \quad (5) \\ &= d_{i_1 i_2} - d_{i_2 i_i} \end{aligned}$$

with $d_{i_1 i_2} = (x_{i_1} - a_{i_1})u_{i_2}$. The $r_{i_1 i_2}$ components have the property $r_{i_1 i_2} = -r_{i_2 i_1} \Rightarrow |r_{i_1 i_2}| = |r_{i_2 i_1}|$, so we have freedom in choosing the indices order. The point $\mathbf{x}$ follows the grid points j_1, j_2, j_3

$$x_{ij_i} = x_{0i} + \Delta x_i \cdot j_i \quad (6)$$

We have

$$\begin{aligned} d_{i_1 i_2} &= (x_{0i_1} - a_{0i_1})u_{i_2} + \Delta x_{i_1} u_{i_2} \cdot j_{i_1} \quad (7) \\ &= c_{i_1 i_2} + b_{i_1 i_2} \cdot j_{i_1} \end{aligned}$$

Hence the components $d_{i_1 i_2}$ can be expressed using a constant part $c_{i_1 i_2} = (x_{0i_1} - a_{i_1})u_{i_2}$ and an incremental part varying with $b_{i_1 i_2} = \Delta x_{i_1} u_{i_2}$ when moving from one grid position to the next along the i_1 direction. An implementation of a ray tracing algorithm using above equations is presented in Fig. 5.

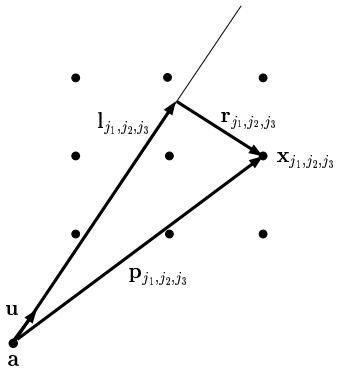

Fig. 4. Distances from the points of a grid to an arbitrary straight line.

B. Second approach

The algorithm presented above does not directly track the distance l along the LOR and needs additional computations in order to obtain it. A different approach which is based on the equation $\|\mathbf{p}\|^2 = l^2 + r^2$, where $\mathbf{p} = \mathbf{x} - \mathbf{a}$ and $l = \mathbf{u} \cdot \mathbf{p}$, automatically computes both r and l for each visited grid point. As shown in Fig. 4 for each component $i \in \{1, 2, 3\}$ we have

$$\begin{aligned} p_{ij_i} &= x_{0i} - a_i + j_i \cdot \Delta x_i \quad (8) \\ l_{ij_i} &= (x_{0i} - a_i)u_i + j_i \cdot \Delta x_i u_i \quad (9) \end{aligned}$$

sort $\{u_i\}_{i\in\{1,2,3\}}$ so that $u_{i_1} \geq u_{i_2} \geq u_{i_3}$;
determine $j_{1\min}$, $j_{1\max}$, the first and last grid planes intersected along the direction i_1;
$d_{12} = c_{i_1 i_2} + b_{i_1 i_2} \cdot j_{1\min}$;
$d_{13} = c_{i_1 i_3} + b_{i_1 i_3} \cdot j_{1\min}$;
for $j_1 = j_{1\min}$ **to** $j_{1\max}$ **do**
  determine active region limits $j_{2\min}$, $j_{2\max}$, $j_{3\min}$, $j_{3\max}$;
  $r_{12} = d_{12} - c_{i_2 i_1} - b_{i_2 i_1} \cdot j_{2\min}$;
  $d_{23} = c_{i_2 i_3} + b_{i_2 i_3} \cdot j_{2\min}$;
  for $j_2 = j_{2\min}$ **to** $j_{2\max}$ **do**
    $r_{12}^{\text{sq}} = r_{12}^2$;
    $r_{13} = d_{13} - c_{i_3 i_1} - b_{i_3 i_1} \cdot j_{3\min}$;
    $r_{23} = d_{23} - c_{i_3 i_2} - b_{i_3 i_2} \cdot j_{3\min}$;
    for $j_3 = j_{3\min}$ **to** $j_{3\max}$ **do**
      $r^2 = r_{12}^{\text{sq}} + r_{13}^2 + r_{23}^2$;
      perform the desired action with current blob
      at the distance r from the axis of the LOR;
      r_{13} -= $b_{i_3 i_1}$;
      r_{23} -= $b_{i_3 i_1}$;
    endfor
    r_{12} -= $b_{i_2 i_1}$;
    d_{23} += $b_{i_2 i_3}$;
  endfor
  d_{12} += $b_{i_1 i_2}$;
  d_{13} += $b_{i_1 i_3}$;
endfor

Fig. 5. Ray tracing through a grid of blobs first algorithm.

The distances to a point $\mathbf{x}_{j_1 j_2 j_3}$ on the grid are then

$$l_{j_1 j_2 j_3} = \sum_{i\in\{1,2,3\}} c_i u_i + j_i \cdot \Delta x_i u_i \tag{10}$$

$$r^2_{j_1 j_2 j_3} = \sum_{i\in\{1,2,3\}} (c_i + j_i \cdot \Delta x_i)^2 - l^2_{j_1 j_2 j_3} \tag{11}$$

where here $c_i = x_{0i} - a_i$, $i \in \{1,2,3\}$.

In (10) and (11) one can notice that the differences between the computation of r and l for one grid point and a consecutive grid point along any i axis, involve only incremental changes with Δx_i and $\Delta x_i u_i$. Hence one can obtain an efficient algorithm that performs projection tracing through a grid by determining the distances from the grid points to the projection axis and the positions along the trajectory. A pseudo-code example of such an algorithm is presented in Fig. 6.

In this second variant one has to make sure that the floating point representation used is accurate enough so that when p^2 and l^2 are at their maximal values their representation error is not of the magnitude of the difference $r^2 = p^2 - l^2$. Double precision representation is satisfactory for typical medical image sizes.

The speed of both algorithms is determined by the number of volume elements that are visited. Larger projection kernels or higher overlap of the blobs result in slower projection calculations, that being the price paid for the increased accuracy

sort $\{u_i\}_{i\in\{1,2,3\}}$ so that $u_{i_1} \geq u_{i_2} \geq u_{i_3}$;
determine $j_{1\min}$, $j_{1\max}$, the first and last grid planes intersected along the direction i_1;
$p_1 = c_{i_1} + j_{1\min} \cdot \Delta x_{i_1}$;
$l_1 = c_{i_1} u_{i_1} + j_{1\min} \cdot \Delta x_{i_1} u_{i_1}$;
for $j_1 = j_{1\min}$ **to** $j_{1\max}$ **do**
  determine active region limits $j_{2\min}$, $j_{2\max}$, $j_{3\min}$, $j_{3\max}$;
  $p_1^{\text{sq}} = p_1^2$;
  $p_2 = c_{i_2} + j_{2\min} \cdot \Delta x_{i_2}$;
  $l_2 = l_1 + c_{i_2} u_{i_2} + j_{2\min} \cdot \Delta x_{i_2} u_{i_2}$;
  $p_{30} = c_{i_3} + j_{3\min} \cdot \Delta x_{i_3}$;
  $l_{30} = c_{i_3} u_{i_3} + j_{3\min} \cdot \Delta x_{i_3} u_{i_3}$;
  for $j_2 = j_{2\min}$ **to** $j_{2\max}$ **do**
    $p_{12}^{\text{sq}} = p_1^{\text{sq}} + p_2^2$;
    $p_3 = p_{30}$;
    $l = l_2 + l_{30}$;
    for $j_3 = j_{3\min}$ **to** $j_{3\max}$ **do**
      $r^2 = p_{12}^{\text{sq}} + p_3^2 - l^2$;
      perform the desired action with current blob
      at the distances l, along the trajectory,
      and r from the axis of the LOR;
      p_3 += Δx_{i_3};
      l += $\Delta x_{i_3} u_{i_3}$;
    endfor
    p_2 += Δx_{i_2};
    l_2 += $\Delta x_{i_2} u_{i_2}$;
  endfor
  p_1 += Δx_{i_1};
  l_1 += $\Delta x_{i_1} u_{i_1}$;
endfor

Fig. 6. Ray tracing through a grid of blobs second algorithm.

of the model. Measuring the speed in terms of the number of image elements visited per unit time, the ray tracing through blobs algorithms are competitive with the ray tracing through voxels algorithms. The actual performance depends on the computer architecture, the programming style, as well as the nature of the additional computations done in the reconstruction algorithm.

IV. Conclusion

Image representation using spherically symmetric basis functions on a regular grid enables in a convenient and flexible way more realistic modeling of the forward projection by considering tube shaped kernels. Ray tracing procedures based on the method described above have been implemented and tested demonstrating great flexibility, being used for list mode PET image reconstruction, including the case of time of flight PET [9], and for iterative transmission image reconstruction with cone beam data [10].

References

[1] R. L. Siddon, "Fast calculation of the exact radiological path for a three-dimensional CT array," *Med. Phys.*, vol. 12, no. 2, pp. 252–255, 1985.

[2] F. Jacobs, E. Sundermann, M. Christiaens, B. De Sutter, and I. Lemahieu, "A fast algorithm to calculate the exact radiological path, through a pixel or voxel space," *Journal of Computing and Information Technology*, vol. 6, no. 1, pp. 89–94, 1998.

[3] M. Christiaens, B. De Sutter, K. De Bosschere, J. Van Campenhout, and I. Lemahieu, "A fast, cache-aware algorithm for the calculation of radiological paths exploiting subword parallelism," *Journal of Systems Architecture*, vol. 45, pp. 781–790, 1999.

[4] G. Han, Z. Liang, and J. You, "A fast ray-tracing technique for TCT and ECT studies," in *IEEE Nuclear Science Symposium. Conference Record*, vol. 3, pp. 1515–1518, 1999.

[5] B. De Man and S. Basu, "Distance-driven projection and backprojection," in *IEEE Nuclear Science Symposium, Conference Record*, vol. 3, pp. 1477– 1480, 2002.

[6] H. Zhao and A. J. Reader, "A fast ray-tracing technique to calculate line integral paths in voxel arrays," in *IEEE Nuclear Science Symposium. Conference Record*, 2003. M11-197.

[7] R. M. Lewitt, "Alternatives to voxels for image representation in iterative reconstruction algorithms," *Phys. Med. Biol.*, vol. 37, no. 3, pp. 705–716, 1992.

[8] S. Staelens, Y. D'Asseler, S. Vandenberghe, M. Koole, I. Lemahieu, and R. Van de Walle, "A three-dimensional theoretical model incorporating spatial detection uncertainty in continuous detector PET," *Phys. Med. Biol.*, vol. 49, no. 11, pp. 2337–2350, 2004.

[9] J. S. Karp, S. Surti, A. Kuhn, L. M. Popescu, M. E. Daube-Witherspoon, and G. Muehllehner, "Time of flight PET with lanthanum bromide scintillators," *J. Nucl. Med.*, vol. 45, no. SNM 51-st Ann. Meet., Abs. Bk. Supp., p. 99P, 2004.

[10] M. E. Daube-Witherspoon, L. M. Popescu, S. Matej, C. A. Cardi, R. M. Lewitt, and J. S. Karp, "Rebinning and reconstruction of point source transmission data for positron emission tomography," in *IEEE NSS-MIC Conf. Rec.*, 2003. M11-218.