

## A comparison of rotation- and blob-based system models for 3D SPECT with depth-dependent detector response

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

2004 Phys. Med. Biol. 49 2157

(<http://iopscience.iop.org/0031-9155/49/11/003>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 141.213.15.65

The article was downloaded on 30/01/2013 at 20:33

Please note that [terms and conditions apply](#).

# A comparison of rotation- and blob-based system models for 3D SPECT with depth-dependent detector response

A Yendiki and J A Fessler

Department of Electrical Engineering and Computer Science, University of Michigan,  
Ann Arbor, MI, USA

E-mail: [nastazia@eecs.umich.edu](mailto:nastazia@eecs.umich.edu)

Received 1 October 2003

Published 19 May 2004

Online at [stacks.iop.org/PMB/49/2157](http://stacks.iop.org/PMB/49/2157)

DOI: 10.1088/0031-9155/49/11/003

## Abstract

We compare two different implementations of a 3D SPECT system model for iterative reconstruction, both of which compensate for non-uniform photon attenuation and depth-dependent system response. One implementation performs fast rotation of images represented using a basis of rectangular voxels, whereas the other represents images using a basis of rotationally symmetric volume elements. In our simulations the blob-based approach was found to slightly outperform the rotation-based one in terms of the bias-variance trade-off in the reconstructed images. Their difference can be significant, however, in terms of computational load. The rotation-based method is faster for many typical SPECT reconstruction problems, but the blob-based one can be better-suited to cases where the reconstruction algorithm needs to process one volume element at a time.

## 1. Introduction

Statistical reconstruction is now widely used in computed tomography, since it produces higher image quality than analytical methods such as filtered back-projection (FBP). However, this improved quality is achieved at the cost of increased computation time. A key component of the accuracy-computation trade-off in the iterative algorithms used for statistical image reconstruction is that typically one projection and one backprojection operation have to be performed at each iteration of the algorithm. The implementation of these operations requires a model for the imaging system at hand. In general, more detailed models result in higher image quality but also in higher computational load, which can become especially cumbersome in 3D problems.

In this paper we focus on system models for 3D single-photon emission computed tomography (SPECT), which compensate for non-uniform photon attenuation as well as the depth-dependent response inherent in SPECT systems. In particular, we consider here rotation- and blob-based implementations. Several rotation-based models have been found to achieve superior accuracy when compared to the simple line-integral model in 2D reconstruction (Di Bella *et al* 1995). Furthermore, models that represent images in terms of spherically symmetric blobs have been applied to 3D positron-emission tomography (PET) and have been found to possess better bias-variance properties than models using cubic voxel representations (Matej and Lewitt 1996). The blob-based methods are computationally tractable for shift-variant PET system models when the ‘footprint’ (2D projection) of the 3D blob is pre-computed and saved. However, it is not clear if the same holds true for SPECT, where the depth-dependent blur causes these footprints to expand considerably with increasing distance from the detector. Also, for the purpose of SPECT reconstructions, it is of interest to know how these blob-based models compare in terms of bias-variance properties and speed to the rotation-based models.

We have implemented both types of system models for fully 3D SPECT. We present an overview of the two implementations in section 2 and analyse them in terms of computational load and memory usage in sections 3 and 4 respectively. We present results from reconstructions of simulated phantom data in section 5 and further discuss those results in section 6.

## 2. System models

For both types of system models we have implemented the back-projector as the exact adjoint of the respective projector. Using a simpler back-projector can speed up reconstruction but it does not preserve the convergence properties of the iterative reconstruction algorithm and it introduces artefacts that may accumulate with every iteration (Zeng and Gullberg 2000). To ensure a comparison untarnished of such artefacts, we utilized adjoint pairs.

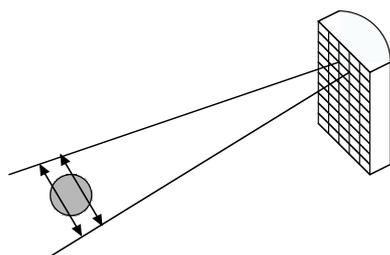
Both types of system models compensate for non-uniform attenuation. For the sake of speed, the attenuation factors are pre-computed for every volume element (cubic voxel or blob) at every projection angle. Utilizing the central ray approximation, an attenuation factor is computed by summing over the attenuation map along the perpendicular line from the centre of the volume element to the detector.

### 2.1. Rotation-based projector

The rotation-based projector we use follows the processing steps listed in Zeng and Gullberg (1992) and performs the rotation step using the fast and accurate three-pass separable rotation method described in Unser *et al* (1995). The three-pass method decomposes the rotation of a 2D image into three sets of 1D interpolations. These interpolations are equivalent to applying appropriate non-integer shifts first to each row, then to each column, and then again to each row of the image.

Projectors utilizing the three-pass method with linear and cubic interpolation have been compared in Di Bella *et al* (1995) and the latter outperformed the line-integral projector in projection accuracy. However, cubic interpolation introduces negative values in the rotated image, which is unnatural in emission tomography, where image values represent photon counts. Thus, we use linear interpolation hereafter.

The rotation-based 3D SPECT projector implemented for this comparison compensates for depth-dependent detector response. For every projection angle, it first rotates the input



**Figure 1.** If the FWHM of the system blur does not increase very fast with distance (relatively to the size of the blob), assume that it is approximately constant throughout the blob.

emission volume around its  $z$ -axis by applying the three-pass method to every slice of the volume separately. It then applies to every voxel the appropriate attenuation factor. Subsequently, every image plane that corresponds to a different distance from the detector face is blurred by a 2D convolution kernel that models the detector response at the corresponding distance. Finally, the voxels are summed along the direction that is perpendicular to the detector to produce a 2D projection of the emission volume.

## 2.2. Blob-based projector

Rotationally symmetric volume elements can provide a basis that is more natural than cubic voxels for representing smooth radioactivity distributions. Moreover, their symmetry makes them appealing for some image reconstruction applications since the line integrals through them are independent of projection angle and only depend on the distance of the line of integration from the centre of the element (Lewitt 1990, 1992).

Desired properties of the rotationally symmetric basis functions are effective band-limitedness (to match the band-limitedness of tomographic data) and finite spatial support (to facilitate computation). By these criteria, a choice that is nearly optimal yet easy to compute is the Kaiser–Bessel blob:

$$b_{R,\alpha,m}(r) = \begin{cases} [\sqrt{1 - (r/R)^2}]^m I_m[\alpha\sqrt{1 - (r/R)^2}] / I_m(\alpha) & 0 \leq r \leq R \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where  $I_m$  denotes the modified Bessel function of the first kind of order  $m$ ,  $R$  controls the blob support,  $\alpha$  controls the blob taper and  $m$  controls the number of continuous derivatives at  $r = R$ .

We have implemented a 3D SPECT projector utilizing Kaiser–Bessel blobs, similar to the one described for PET in Matej and Lewitt (1996). The additional step in our projector is that the blobs at each different distance from the detector now need to be blurred by the corresponding point-spread function (PSF) of the SPECT detector.

Of course, different points within a 3D blob lie at different distances from the detector and therefore correspond to different PSFs. However, when the full width at half-maximum (FWHM) of the depth-dependent system blur changes slowly with distance relatively to the blob size, as illustrated in figure 1, we can assume that the blur FWHM is approximately the same throughout a blob as it is at the blob's centre. This approximation simplifies the implementation significantly, since it makes projecting the blurred blob equivalent to blurring the projections of the blob. Thus, we can pre-compute and save a set of finely sampled 2D footprints of the blob, blurred by the detector PSF at each of a set of distances from the detector.

The spacing of the stored samples for a single footprint, as well as the number of different distances for which footprints are stored, depends on the required accuracy of the implementation. Clearly, although the continuous projections of the rotationally symmetric volume elements are angle-independent, the discrete projections are not necessarily so. In general, the same footprint is sampled at slightly different points (by a small shift) at different angles. However, we do not have to store a different footprint for each angle, as we may have done if applying this method to, say, cubic voxels. Storing a finely sampled version of the footprint and picking out samples based on the nearest-neighbour criterion will yield adequate accuracy for SPECT. A similar nearest-neighbour approach is taken with respect to distance from the detector when choosing which of the stored footprints to use. In some sense, this approximation is used implicitly by the rotation-based model as well, since after rotation by a non-trivial angle, some edges of the cubic voxel will be at different distances from the detector than others.

We also model the attenuation as being constant throughout a single blob, as we did for cubic voxels in the rotation-based implementation. We arrange the blobs over a uniform 3D Cartesian grid for simplicity, although more efficient blob arrangements are also possible (Matej and Lewitt 1995).

Taking all of the above into account, a projection operation with the blob-based model is performed as follows. At each projection angle and for each blob in the volume, we select the footprint that corresponds to the distance of the blob from the detector in a nearest-neighbour sense. For every sample on this footprint that corresponds to a projection bin in a nearest-neighbour sense, we multiply the sample by the appropriate attenuation factor and the coefficient  $c_j$  of the blob and add the product onto the projection plane.

In general, we assume that the object  $x(\vec{r})$  can be represented as a linear combination of shifted copies of some volume element  $b(\vec{r})$ :

$$x(\vec{r}) = \sum_j c_j b(\vec{r} - \vec{r}_j)$$

and we attempt to reconstruct the coefficients  $c_j$  from noisy projections of the object. The rotation-based model assumes  $b(\vec{r})$  to be a rect function, whereas the blob-based model assumes  $b(\vec{r})$  to have the form (1), where  $r = \|\vec{r}\|$ . Therefore, an extra step of convolving the reconstructed coefficients with the blob function is needed when using the blob-based model, if the images are to be displayed or compared to images represented in voxel space.

### 3. Computation

We provide here an assessment of the number of floating-point operations (additions and multiplications) required by each implementation to compute one projection of an  $n_x \times n_y \times n_z$  image.

The operations performed by the rotation-based projector are as follows.

- *Rotation.* Let  $n_i$  be the length of the 1D interpolation filter used by the separable three-pass rotation method. To rotate one  $n_x \times n_y$  slice of the input volume, each of the three passes convolves the interpolation filter with all rows or all columns of the slice. This amounts to a total of  $\frac{3\pi}{4}(2n_i - 1)n_x n_y n_z$  operations to rotate all slices of the input volume. The  $\frac{\pi}{4}$  factor stems from the fact that the rotation algorithm is not applied to all  $n_x n_y$  voxels in a slice but only to those that lie within the circle inscribed in the slice. More computation is required when the interpolation filter must be preceded by an IIR filter, e.g. for cubic interpolation (Unser *et al* 1993). For linear interpolation, where  $n_i = 2$  and no IIR pre-processing is needed, the rotation requires  $\frac{9\pi}{4}n_x n_y n_z \approx 7.1n_x n_y n_z$  operations.

- *Attenuation.* Applying the pre-computed attenuation factors to all voxels of the rotated volume requires  $n_x n_y n_z$  multiplications.
- *Blurring.* In general, convolving all  $n_y$  planes in the volume with 2D PSFs of average size  $n_h \times n_h$ , requires  $(2n_h^2 - 1)n_x n_y n_z$  operations. However, this step can be significantly speeded up for certain convenient PSF shapes. If, for example, the system PSFs are separable, each 2D convolution is reduced to two 1D convolutions, in which case this step can be implemented with as few as  $2(2n_h - 1)n_x n_y n_z$  operations.
- *Projection.* Summing the  $n_y$  blurred planes of size  $n_x \times n_z$  requires  $n_x n_y n_z$  additions.

Therefore, with linear interpolation in the rotation step, the rotation-based projector performs a total of  $(2n_h^2 + 8.1)n_x n_y n_z$  operations in the general case of non-separable PSFs or  $(4n_h + 7.1)n_x n_y n_z$  operations in the case of separable PSFs.

On the other hand, the blob-based projector has to perform the following operations for each blob in the object support.

- *Attenuation.* 1 multiplication of the blob coefficient by a pre-computed attenuation factor.
- *Projection.*  $n_{bb}^2$  multiplications to scale the blob footprint by the blob coefficient and  $n_{bb}^2$  additions to add it onto the projection view, where  $n_{bb} \times n_{bb}$  is the average size of the blob footprint after it has been blurred by the appropriate PSF.

On paper, the  $2n_{bb}^2$  operations needed for the projection step above could be reduced to  $\frac{\pi}{2}n_{bb}^2$ , if we only projected the footprint samples that lay within the circular support of the footprint, rather than the entire  $n_{bb} \times n_{bb}$  square. However, we do not consider this reduction further, as it is not trivial to implement. In the rotation-based projector, the  $\frac{\pi}{4}$  improvement corresponds to a unique circle, the one inscribed in the image matrix, and it is simple to pre-compute its limits and reuse them throughout the computations. In the blob-based case, the  $\frac{\pi}{4}$  factor would correspond to a different circle for each blob, whose limits change with blob position from angle to angle. Pre-computing and reusing those limits would require a large amount of memory and the overhead due to memory access would undermine the computational savings.

Therefore the operations add up to  $2n_{bb}^2 + 1$  per blob. Assuming that the blob grid is the same as the voxel grid used by the rotation-based projector above, the blob-based projector performs a total of  $(2n_{bb}^2 + 1)n_x n_y n_z$  operations for the entire projection view. If  $n_b$  is the width of the basic blob before blurring, then  $n_{bb} = n_b + n_h - 1$ . For any blob radius between 1 and 2 times the grid spacing,  $n_b = 3$  and therefore the blob-based projector performs a total of  $(2n_h^2 + 8n_h + 9)n_x n_y n_z$  operations.

Based on the calculations above, the rotation-based implementation would always be faster than the blob-based implementation with a regular cubic grid of the same size. However, when the blobs are arranged on a body-centred cubic (BCC) grid instead of the Cartesian grid, the operations required by the blob-based method can be reduced by a factor of  $\sqrt{2}$  without loss in image quality, as described in Matej and Lewitt (1995).

If the system PSFs are separable (e.g. Gaussian), the rotation-based model with separable 1D convolutions would be faster than the blob-based model even with the BCC arrangement. In the general case of non-separable PSFs, the blob-based model with the BCC grid will be faster only when  $2n_h^2 + 8n_h + 9 < \sqrt{2}(2n_h^2 + 8.1)$ , which is true for  $n_h \geq 10$  pixels. Whether this is a realistic PSF size will depend on the properties of the imaging system, the radius of the detector orbit and the size of the reconstruction. For example, in the case of the two SPECT reconstruction problems tackled in the simulations of section 5 (technetium and iodine studies), where the diameters of the system PSF at the centre of the field of view were about 31 and 52 mm respectively, this would mean that the blobs would be faster for a pixel size

**Table 1.** Projection speed (s) on a 3 GHz xeon processor.

$n_x \times n_y \times n_z$	$n_\theta$	Blob	Blob BCC	Rot 2D	Rot 1D
$64 \times 64 \times 25$	60	3.3	2.3	3.2	1.6
$128 \times 128 \times 50$	120	113.	80.	69.	31.

Column 3: CPU times for blob-based model with  $R = 2$  times the grid spacing,  $\alpha = 10.4$ ,  $m = 2$ , and the same Cartesian grid as the rotation-based model. Column 4: CPU times from column 3 divided by  $\sqrt{2}$  (a prediction of CPU times for blobs with the BCC grid). Column 5: CPU times for rotation-based model with 2D FFTs. Column 6: CPU times for rotation-based model with separable 1D convolutions.

smaller than 3.1 and 5.2 mm respectively, which would be true for matrix sizes of  $128 \times 128$  or larger.

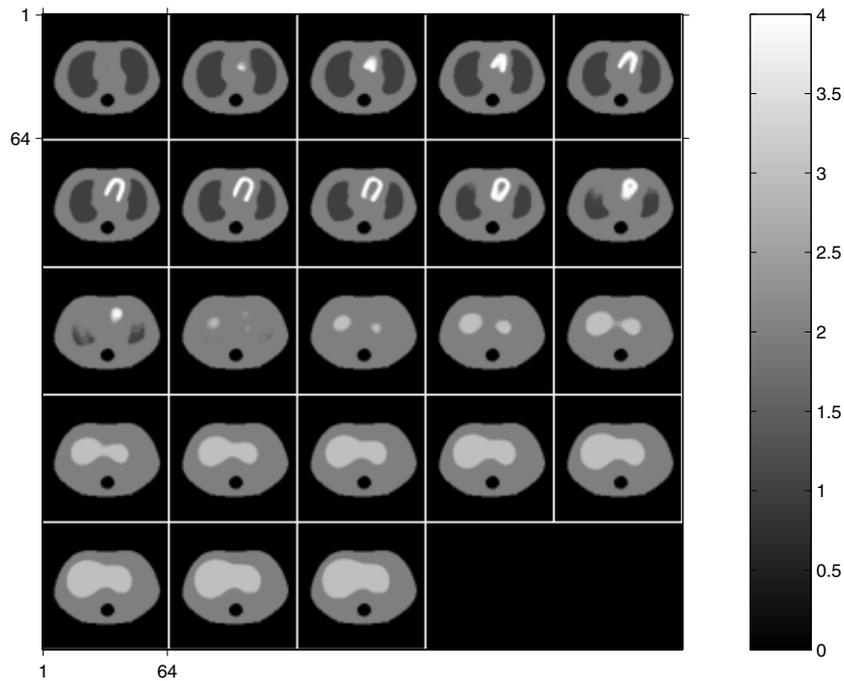
Furthermore, because the blob-based implementation processes the blobs in the image grid sequentially, it can very naturally take advantage of the object support, which is readily available in SPECT from the attenuation map. If  $\lambda \leq 1$  is the ratio of blobs inside the object support, the projector or back-projector only needs to process  $\lambda n_x n_y n_z$  blobs (or approximately  $\frac{1}{\sqrt{2}} \lambda n_x n_y n_z$  blobs when using the BCC grid). The rotation-based projector could also be modified to take advantage of the object support to some extent. However, it would be less natural than it would be in the blob case and it would significantly increase the complexity of the rotation-based implementation and its underlying data structure.

#### 4. Memory usage

Both system models pre-compute and store attenuation factors corresponding to all volume elements at all angles, to reuse them over iterations. This amounts to an array of type float and size  $n_\theta n_x n_y n_z$ , where  $n_\theta$  is the number of projection views, and it is the dominant term in the total memory usage of both models. For a total of  $128 \times 128 \times 50$  volume elements and 120 projection views, this requires roughly 400 Mbytes of memory, which is reasonable for modern workstations.

#### 5. Simulation results

The CPU time required to calculate a full projection set with each system model on a 3 GHz xeon processor with 2 GB RAM, gcc and Linux is shown in table 1. Times are reported for a system with a pixel size of 4 mm and 2 mm in the  $64 \times 64 \times 25$  and  $128 \times 128 \times 50$  cases respectively, and PSF widths identical to those described for the simulations below. We assume the object support to be the circle that is inscribed in the  $n_x \times n_y$  image matrix, therefore the blob-based model benefits by only processing  $\frac{\pi}{4} n_x n_y n_z$  grid positions. Since our blob-based implementation uses a Cartesian grid, we also provide a prediction of the speed that could be attained with the BCC grid by dividing the practical CPU times by a factor of  $\sqrt{2}$ . CPU times for the rotation-based model are reported both for separable PSFs (implemented with 1D separable convolutions as in the analysis above) and for non-separable PSFs (implemented here with 2D FFTs). For the  $64 \times 64 \times 25$  matrix the computational savings afforded by the BCC grid and the circular object support make the blob-based model faster than the rotation-based model with 2D FFTs, whereas this is not the case for the  $128 \times 128 \times 50$  matrix. The blob-based model with a Cartesian grid is consistently the slowest and the rotation-based model with separable 1D convolution is consistently the fastest.



**Figure 2.** Anthropomorphic chest phantom used in the simulations (downsampled to  $64 \times 64 \times 23$ ).

To evaluate the two models in terms of the bias-variance trade-off, we reconstructed  $64 \times 64 \times 23$  images of an anthropomorphic chest phantom (shown in figure 2) from projections taken at 60 uniformly spaced angles over a  $[0, 2\pi)$  range. The heart, liver, lungs, spine and surrounding tissue in the phantom have uniform radioactivity densities with ratio 4:3:1:0:2 and uniform attenuation coefficients equal to 0.15, 0.15, 0.027, 0.191 and  $0.15 \text{ cm}^{-1}$  respectively, values appropriate for the 140 keV photons emitted by Tc-99m.

The pixel size of the system was set to 4 mm. The depth-dependent PSFs were assumed to be 2D Gaussians with a FWHM increasing from 9.4 at the plane of the emission volume that is nearest to the detector to 22.6 mm at the one that is furthest. The same set of PSFs was used at all projection angles, as in a circular orbit.

The noiseless  $256 \times 256$  projections were generated from a high-resolution version of the phantom image using the rotation-based projector (therefore the central ray approximation used by both system models to apply attenuation is here consistent with the data) and were then downsampled to  $64 \times 64$  by averaging<sup>1</sup>. We normalized this set of downsampled projections to a total of  $5 \times 10^6$  counts. To crudely simulate scatter, we uniformly added to the projection set a number of scattered counts equal to 10% of the total true counts. We used this new projection set to generate Poisson-distributed measurements.

We reconstructed images from 200 realizations of the noisy Poisson data using each system model and the ordered subset expectation maximization (OSEM) algorithm with six subsets. The system models that we used in the OSEM reconstructions assumed the same PSFs as the system that we used to generate the noiseless projections. Although it would be of interest

<sup>1</sup> Since the 2D projections are downsampled along two dimensions, if the reconstructed images are to be compared with the true 3D object, a scaling factor equal to the downsampling ratio must be applied. The bias in the reconstructed images as they appear in Yendiki and Fessler (2003) is due to lack of this scaling.

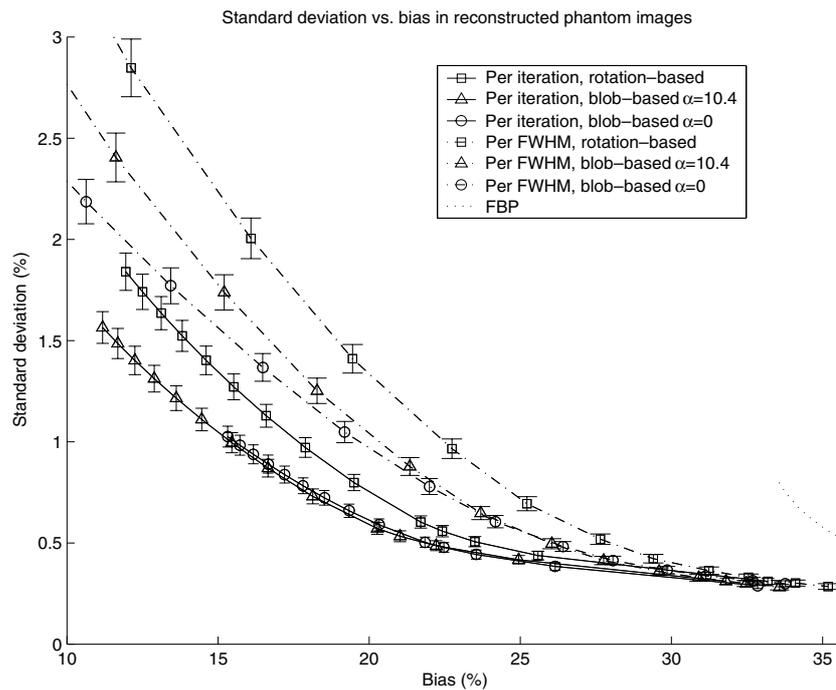
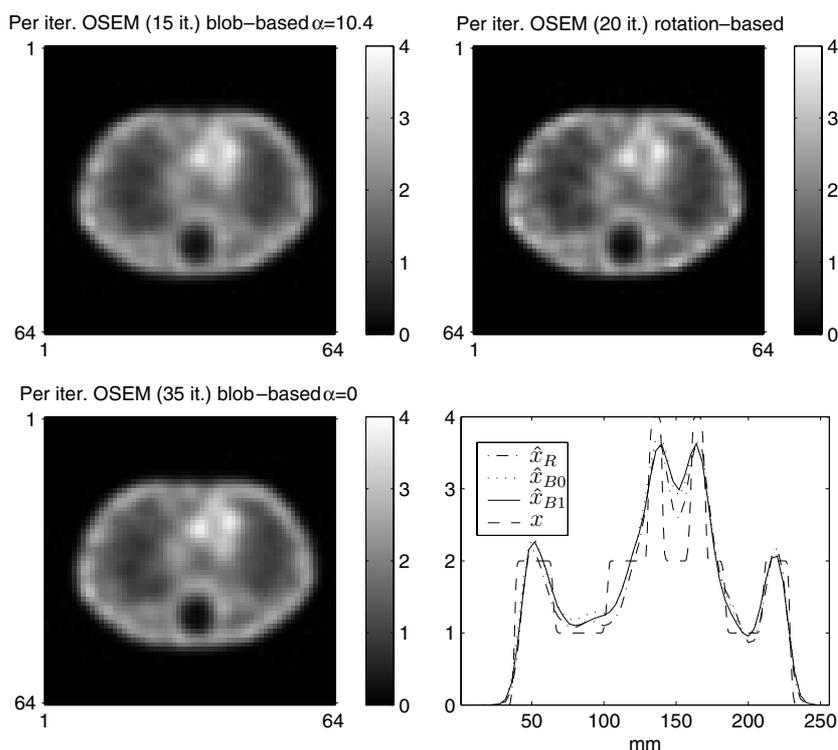


Figure 3. Bias-variance trade-off per OSEM iteration and per post-smoothing FWHM.

to compare the performance of the system models under PSF mismatch, this was beyond the scope of this paper. Our intention here was to isolate the effects of the choice of volume elements from the effects of such a mismatch. The OSEM reconstructions also assumed knowledge of the average scattered counts in the measurements (Daube-Witherspoon *et al* 1992).

We produced two different sets of bias versus standard deviation plots by varying two different parameters that are commonly used to control the bias-variance trade-off, namely the number of iterations or the FWHM of a post-filter. In the first case we initialized the OSEM reconstructions with a uniform image and plotted the bias and standard deviation of the total uptake in the heart area per iteration, up to 50 iterations. In the second case we initialized the OSEM reconstructions with a smooth FBP image (whose non-positive values had been set to a very small positive value) and allowed the algorithm to run for 1000 iterations. We then smoothed the reconstructed images (in the blob-based case after synthesizing them from the reconstructed coefficients) using 3D Gaussian post-filters of various FWHMs, up to 7 pixels. Similarly, we recorded the bias and standard deviation of the total uptake in the heart area for each post-filter FWHM. Figure 3 shows plots of the standard deviation versus bias normalized as a percentage of the true total heart activity. The FBP curve in figure 3 was acquired from plain FBP reconstructions with a Hanning window of variable width and Chang attenuation correction but no attempt to compensate for the depth-dependent system blur. These curves are just provided for reference, as the focus here was on comparing the blob- and rotation-based system models rather than iterative reconstruction and FBP.

Figure 4 shows the seventh slice from an image reconstructed with OSEM using the rotation-based model and two different blob-based models. The images were chosen at different iterations to have a similar bias level (about 16.5%) on the per iteration bias-variance



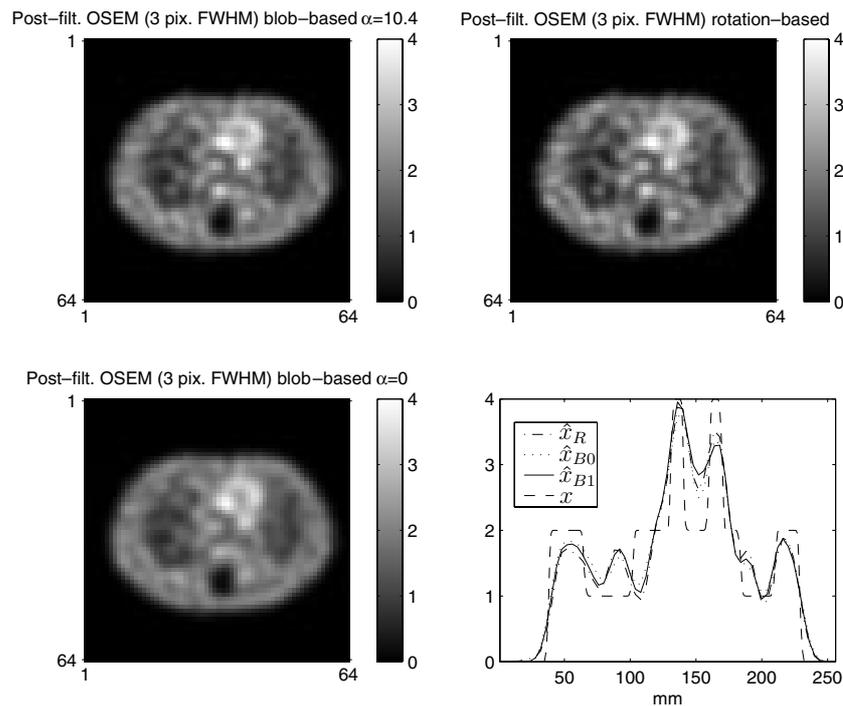
**Figure 4.** A slice and profiles through OSEM-reconstructed images of a single realization of the noisy data. Superimposed on the profile of the true high-resolution phantom ( $x$ ) are those of the images reconstructed with the rotation-based model ( $\hat{x}_R$ , NMSE = 3.26%), the blob-based model with  $\alpha = 0$  ( $\hat{x}_{B0}$ , NMSE = 2.76%) and the blob-based model with  $\alpha = 10.4$  ( $\hat{x}_{B1}$ , NMSE = 2.87%).

plot. A profile of these slices through the heart area is also shown, superimposed on the corresponding profile of the noiseless high-resolution phantom. Similarly, figure 5 shows the seventh slice from a set of the post-smoothed 1000-iteration OSEM reconstructions with the three different system models, chosen to have a similar bias level (about 19%) on the per FWHM bias-variance plot. All images in figures 4 and 5 were reconstructed from the same realization of the noisy data.

## 6. Discussion

For the blob-based reconstructions above, we tried two different blob shapes, both of them with radius  $R = 2$  times the grid increment and  $m = 2$  (continuous and with continuous first derivative at  $r = 2$ ) but with different taper parameters and therefore FWHMs. The blob with  $\alpha = 10.4$  has a FWHM 1.33 times the grid increment and has been shown in Matej and Lewitt (1996) to possess good properties with respect to the approximation of the constant 3D function. The blob with  $\alpha = 0$  does not possess such properties and in fact is a uniform sphere with a FWHM equal to its diameter.

Although the shape of these two blobs is quite different, their bias-variance performance was not characterized by statistically significant differences at the commonly acceptable range



**Figure 5.** A slice and profiles through over-iterated and post-smoothed OSEM-reconstructed images of a single realization of the noisy data. Superimposed on the profile of the true high-resolution phantom ( $x$ ) are those of the images reconstructed with the rotation-based model ( $\hat{x}_R$ , NMSE = 4.12%), the blob-based model with  $\alpha = 0$  ( $\hat{x}_{B0}$ , NMSE = 2.99%) and the blob-based model with  $\alpha = 10.4$  ( $\hat{x}_{B1}$ , NMSE = 3.60%).

of bias values, as shown by the error bars in figure 3. In addition to these two choices of blob shape, we also considered the other choices listed in table 1 of Matej and Lewitt (1996) and they resulted in similar performance, although these results are not shown here to avoid excessive clutter in the plots. No further effort was made here to optimize the blob parameters for SPECT reconstruction. It appears that the wide PSFs of the SPECT detector dominated the bias-variance properties of the reconstruction and varying the blob shape did not make a significant difference in the simulations presented here.

The plots do show, however, that the bias-variance performance of the blob-based projectors was somewhat better than that of the rotation-based one, similarly to what investigators have reported for PET (Matej *et al* 1994, Matej and Lewitt 1996). It is possible that spherically symmetric volume elements were a better match for the contours of the shapes in the high-resolution phantom used in our simulations. However, as shown by the bias-variance plots and confirmed by the reconstructed images, the improvement is modest for SPECT.

The spacing of the pre-computed blob footprint samples that we used in all the simulations above was equal to 1/10 of the system pixel size. Decreasing this sample spacing did not affect the quality of the reconstructed images.

Similar results (not shown here) were obtained when simulating an I-131 study with the same phantom and system parameters appropriate for an ultra-high energy collimator (PSF FWHM from 3 to 43 mm, pixel size 7.2 mm).

In general, our simulations indicated only a slightly superior performance of the blob-based implementations of the 3D SPECT system model in terms of the bias-variance trade-off. On the other hand, the differences in terms of computational load were significant. When the system PSFs can be considered separable, such as the Gaussians used above, the rotation-based model with 1D separable convolutions will always be faster. If we wanted to model collimator properties in greater detail with non-symmetric PSFs, we would resort to 2D convolutions, and in that case the blob-based model with an efficient grid could be faster for large PSF sizes. However, FFTs can be used instead of 2D convolutions to speed up the rotation-based model with those larger PSFs.

The fact that use of the object support can be very naturally built into the blob-based system model makes it very suitable for cases where we only need to reconstruct a few or even a single volume element from the object, such as in a recently proposed method for fast estimation of resolution and covariance in SPECT (Stayman and Fessler 2002) and in algorithms that perform element-by-element processing, such as the space-alternating generalized EM (SAGE) algorithm (Fessler and Hero 1995).

Finally, our simulations did not include realistic Compton scatter effects. We speculate that since Compton scatter is relatively spatially smooth, and since scatter computations are usually performed with relatively coarse sampling (Ollinger 1996, Watson *et al* 1999, Beekman *et al* 2002), including a realistic scatter model would not significantly alter the outcome of our comparisons in favour of any choice of basis function.

## Acknowledgments

The authors gratefully acknowledge Dr Kenneth F Koral for ongoing discussions and collaboration, as well as the anonymous reviewers for their helpful suggestions. This work was supported by NCI grants R01 CA-87955 and CA-60711 and NSF grant BES-9982349.

## References

- Beekman F J, de Jong H W A M and van Geloven S 2002 Efficient fully 3D iterative SPECT reconstruction with Monte Carlo based scatter compensation *IEEE Trans. Med. Imaging* **21** 867–77
- Daube-Witherspoon M E, Carson R E, Yan Y and Yap T K 1992 Scatter correction in maximum likelihood reconstruction of PET data *Proc. IEEE Nucl. Sci. Symp. Med. Im. Conf.* vol 2, pp 945–7
- Di Bella E V R, Barclay A B, Eisner R L and Schafer R W 1995 Comparison of rotation-based methods for iterative reconstruction algorithms *Proc. IEEE Nucl. Sci. Symp. Med. Im. Conf.* vol 2, pp 1146–50
- Fessler J A and Hero A O 1995 Penalized maximum-likelihood image reconstruction using space-alternating generalized EM algorithms *IEEE Trans. Image Process.* **4** 1417–29
- Lewitt R M 1990 Multidimensional digital image representations using generalized Kaiser-Bessel window functions *J. Opt. Soc. Am. A* **7** 1834–46
- Lewitt R M 1992 Alternatives to voxels for image representation in iterative reconstruction algorithms *Phys. Med. Biol.* **37** 705–16
- Matej S, Herman G T, Narayan T K, Furuie S S, Lewitt R M and Kinahan P E 1994 Evaluation of task-oriented performance of several fully 3D PET reconstruction algorithms *Phys. Med. Biol.* **39** 355–67
- Matej S and Lewitt R M 1995 Efficient 3D grids for image reconstruction using spherically-symmetric volume elements *IEEE Trans. Nucl. Sci.* **42** 1361–70
- Matej S and Lewitt R M 1996 Practical considerations for 3D image reconstruction using spherically symmetric volume elements *IEEE Trans. Med. Imaging* **15** 68–78
- Ollinger J M 1996 Model-based scatter correction for fully 3D PET *Phys. Med. Biol.* **41** 153–76
- Stayman J W and Fessler J A 2002 Fast methods for approximation of resolution and covariance for SPECT *Proc. IEEE Nucl. Sci. Symp. Med. Im. Conf.* vol 2, pp 786–8
- Unser M, Aldroubi A and Eden M 1993 B-spline signal processing: part II—efficient design and applications *IEEE Trans. Signal. Process.* **41** 834–48

- Unser M, Thevenaz P and Yaroslavsky L 1995 Convolution-based interpolation for fast, high quality rotation of images *IEEE Trans. Image Process.* **4** 1371–81
- Watson C C, Schaefer A, Luk W K and Kirsch C M 1999 Clinical evaluation of single-photon attenuation correction for 3D whole-body PET *IEEE Trans. Nucl. Sci.* **46** 1024–31
- Yendiki A and Fessler J A 2003 A comparison of rotation- and blob-based system models for 3D SPECT with depth-dependent detector response *Proc. Int. Mtg. on Fully 3D Image Reconstruction in Radiation and Nuclear Medicine*
- Zeng G L and Gullberg G T 1992 Frequency domain implementation of the three-dimensional geometric point response correction in SPECT imaging *IEEE Trans. Nucl. Sci.* **39** 1444–53
- Zeng G L and Gullberg G T 2000 Unmatched projector/backprojector pairs in an iterative reconstruction algorithm *IEEE Trans. Med. Imaging* **19** 548–55