

Constructing System Matrices for SPECT Simulations and Reconstructions

Nirantha Balagopal

April 28th, 2017

M.S. Report
The University of Arizona
College of Optical Sciences

Acknowledgement

I would like to thank my adviser, Dr. Matthew Kupinski, for inspiring my interest in image science and for his guidance on this report.

Contents

1	Introduction	3
1.1	An overview of select medical imaging systems	3
1.1.1	Radiotracers	3
1.1.2	SPECT Cameras	3
1.1.3	Image Reconstruction	4
2	Constructing system matrices	4
2.1	Linear imaging equation	4
2.2	A simplified 2D system	5
2.3	Maximum-Likelihood Expectation Maximization	6
2.4	Single point illuminated	8
2.5	Reconstruction with multiple angles	9
2.6	Reconstructing a skull	12
3	Constructing system matrices: a 3D case	12
3.1	Setting up a 3D system	12
3.2	Verifying the matrix	15
3.3	Single point illuminated	15
3.4	Multiple illuminated points	15
4	Discussion	21
4.1	Issues and difficulties	21
4.2	Next Steps	21

1 Introduction

1.1 An overview of select medical imaging systems

Nuclear imaging techniques are used in diagnostic medicine to show physiological functions in the body. In these techniques (most notably Single Photon Emission Computed Tomography, or SPECT, and Positron Emission Tomography, or PET), radiation is emitted from a radiotracer distributed throughout the patient's body. This is in contrast with anatomical imaging systems such as computed tomography (CT) or magnetic-resonance imaging (MRI) that probe the body with external sources.

In this report we focus on modeling pinhole SPECT imaging systems that are used in small-animal imaging and in some human applications. SPECT is an indirect imaging system, and can be understood as a system made up of a radiating object, camera systems that include collimators and detectors, and a reconstruction method. The camera systems generally acquire projection data at multiple angles around the subject. This information is used by the image-reconstruction method to reconstruct a 3D representation of the distribution of the radiotracer in the body. I will briefly outline some of the components in the subsections below.

1.1.1 Radiotracers

SPECT employs injecting a direct gamma emitter. The tracers typically used in SPECT imaging incorporate Technetium-99m (^{99m}Tc). This is a high Z tracer that emits at about 142 keV which can readily exit the body without much scatter. The half-life of this isotope is approximately 6 hours which makes performing clinical imaging studies easy compared to short half-life isotopes often used in PET imaging (^{18}F often used in PET has a half-life of 109 minutes). The 6-hour half life is still short enough to minimize the amount of radioactive material administered to the patient.

1.1.2 SPECT Cameras

Conventionally, parallel-hole collimators (often made of lead or similar materials) are used as the image-forming element in many SPECT systems. Due to the geometry of this configuration, a parallel-hole collimator typically has low sensitivity, as for rays to reach

the detector they must travel close to parallel with the collimator holes (normal to the collimator surface). This also results in “undistorted projection images and a FOV that is constant with distance” [1].

It has been believed that the image-forming element is the limiting factor in resolution [2], but pinhole collimation can be used instead of parallel-hole to obtain a higher resolution. For applications including small-animal imaging or brain imaging, pinhole collimation is often utilized as the image-forming element.

A typical SPECT detector is a scintillation crystal coupled with a series of photodetectors. Scintillation crystals emit visible photons after absorbing gamma rays. The visible photons are detected by a series of photomultiplier tubes (PMTs) used to amplify the signal. The PMT outputs can then be used to estimate the position and energy of the gamma-ray interaction within the crystal.

1.1.3 Image Reconstruction

Image reconstruction is the process of using the SPECT projection images to reconstruct an estimate of the distribution of the radiotracer within the body. Numerous techniques exist for image reconstruction. In this report, I will focus on the use of maximum-likelihood, expectation maximization (MLEM) which will be discussed in greater detail in a later section.

2 Constructing system matrices

2.1 Linear imaging equation

SPECT imaging systems are well approximated with a linear imaging equation. Thus, we represent the imaging equation mathematically as $\vec{g} = \mathbf{H}\vec{f}$ for a noise-free system, or $\vec{g} = \mathbf{H}\vec{f} + \vec{n}$ when noise is present. For now, we will focus on the former. It should be noted that \vec{g} , \vec{f} , and \vec{n} are all vectors and \mathbf{H} is a matrix transform representing the imaging transform. The data \vec{g} are used as input to the image-reconstruction algorithm.

The object \vec{f} is an N -dimensional column vector whose values represent basis coefficients for a continuous object. Typically, the basis function used are voxels and thus the components of \vec{f} are voxel values. Objects are three dimensional but \vec{f} is represented as a column

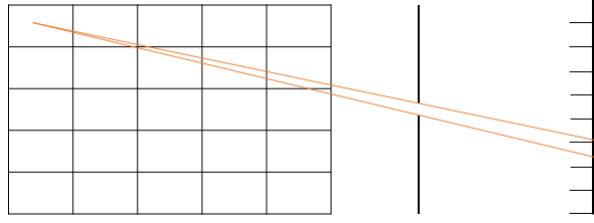


Figure 1: An illustration of a simple 2D system. The ‘voxelized’ rectangle on the left represents an object. Gamma rays from the center of one voxel fan through the pinhole to the ‘pixelized’ detector (right)

vector for mathematical convenience. You can think of \vec{f} as a “flattened” object where all the voxel values are placed in a \vec{f} in a specified order. Similarly, the image data \vec{g} and noise \vec{n} are also vectors (M -dimensional in this case) and represent the image data and noise, respectively. Multiple projections are typically acquired and projection images are generally 2 dimensional. Thus, much like \vec{f} , there is a specified order for the elements of \vec{g} and \vec{n} to represent these as column vectors. Finally, \mathbf{H} is an $M \times N$ dimensional matrix transform that represents how components of the object are transformed by the imaging system to image data.

This equation is linear. Recall that an operator (in this case \mathbf{H}) is linear if for $\vec{g}_1 = \mathbf{H}\vec{f}_1$ and $\vec{g}_2 = \mathbf{H}\vec{f}_2$ and some constant α :

$$\begin{aligned}\vec{g}_1 + \vec{g}_2 &= \mathbf{H}(\vec{f}_1 + \vec{f}_2) \\ \alpha\vec{g}_1 &= \mathbf{H}[\alpha\vec{f}_1]\end{aligned}$$

2.2 A simplified 2D system

Consider a 2D system as shown in Figure 1. The “object” (in this case, a two-dimensional rectangular grid) is voxelized with N voxels which will be j indexed. There is a pinhole a distance d away from the object and the detector is made of a one-dimensional array of M pixels, i indexed.

Our initial simulations focused on creating a simulation of this simplified two-dimensional system. We created an \mathbf{H} matrix ($M \times N$) where element \mathbf{H}_{ij} corresponds to the probability that a ray from voxel j reaches pixel i . This matrix will be used to simulate projection data and to help perform image reconstruction. We use the following specifications (with

relative units):

- Size of object: 128 voxels by 128 voxels
- Number of voxels, N : 16,384
- Side length of each (square) voxel: 2
- Distance from object to pinhole: 192
- Diameter of pinhole: 0.1
- Distance from pinhole to detector: 10
- Number of pixels, M : 128
- Rays leave each voxel from the voxel center.

In the MATLAB code to create the system matrix, we first had to define a Cartesian coordinate system, where the origin was located in the center of the pinhole.

Next, consider the angle α created by two rays, \vec{r}_1 and \vec{r}_2 , leaving a voxel, where \vec{r}_1 is the ray leaving the center of voxel j and grazing the top of the pinhole, and \vec{r}_2 is the ray leaving the voxel and grazing the bottom of the pinhole as shown in Figure 2. We modeled 100 rays that leave the center of each voxel, spread uniformly from \vec{r}_1 to \vec{r}_2 , that hit the detector. We defined the size of the detector array by finding the vectors (from all possible voxels) with the smallest y-coordinate at the detector plane, and the largest y-coordinate at the detector plane. To construct the \mathbf{H} matrix we estimated the probabilities of photons emitted from voxel j hitting detector i by the fraction of the 100 rays that hit each detector element. The matrix is shown as an image in Figure 3.

It should be noted that this matrix does take a lot of memory, however, it also has many zeros and can be saved as a sparse matrix to save memory and speed up calculations.

2.3 Maximum-Likelihood Expectation Maximization

The maximum-likelihood expectation maximization (MLEM) algorithm is based on Poisson-noise statistics and is an iterative method to produce object estimates that increase the likelihood of the image data observed. True convergence is rarely desired with the MLEM algorithm because of data inconsistencies due to noise. [3]

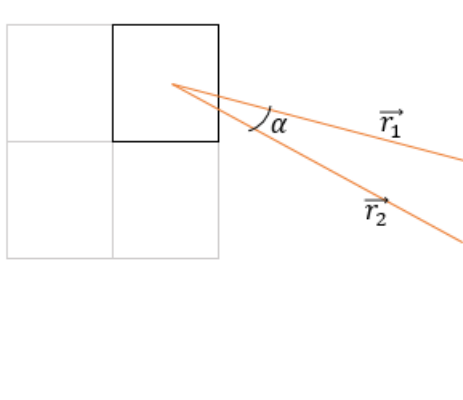


Figure 2: Ray geometry of a simple 2D system where α represents the angle between two rays leaving the center of a voxel and reaching the edges of a pinhole.

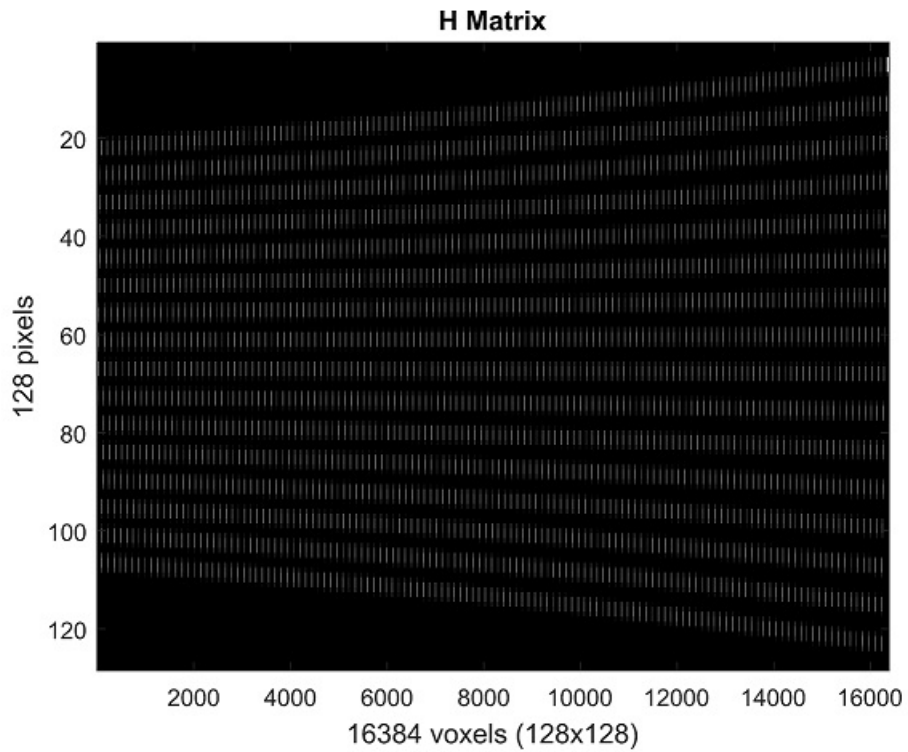


Figure 3: Image representation of a system matrix for the 2D case.

The steps of the MLEM algorithm are given by,

$$\bar{f}_j^{(k+1)} = \bar{f}_j^{(k)} \frac{1}{s_j} \sum_{i=0}^M \frac{g_i}{(\mathbf{H}\bar{f}^{(k)})_i} H_{ij}$$

where s_j is defined as

$$s_j = \sum_{i=0}^M H_{ij}$$

that is, s_j is the “probability that a photon emitted from voxel j is detected somewhere” [4].

We look to first create an image \vec{g} by passing a known object through our imaging matrix. Then, we create an initial guess for our object \vec{f} and use the MLEM algorithm as above with repeated iterations until there is a “reasonable” reconstructed image defined using subjective methods. Note that in simulation, this can work with repeated trials, but in real applications, too many iterations can create a noisy image, especially when the algorithm is forced to reconstruct from noisy data.

2.4 Single point illuminated

What if the object is made up of only 1 illuminated voxel?

Now, we create the image of this on our 128 x 1 detector by multiplying the \mathbf{H} matrix by this (flattened) \vec{f} (the forward problem).

If we run an MLEM algorithm from this object, we need a guess, so we will use a vector of all ones as this 1st iteration (k) guess. The corresponding final object created by this algorithm is shown in Figure 6, for various iteration numbers.

The artifacts of back-projection are shown in Figure 6, as the object is appearing as a line instead of a point, with the line getting thinner per iteration. With projection of the 3D object onto a 2D detector, backprojection will smear the 2D image back into the 3D space resulting in the “streak”. No amount of MLEM iterations will truly rid the reconstruction of this smear, so we have to consider backprojection from multiple angles.

It is interesting to note that a very similar reconstruction may have been obtained from an object that had more than one illuminated point along the line formed by the backprojection in the reconstructed object. For an object made up of only one illuminated point, at least two viewing angles must be considered.

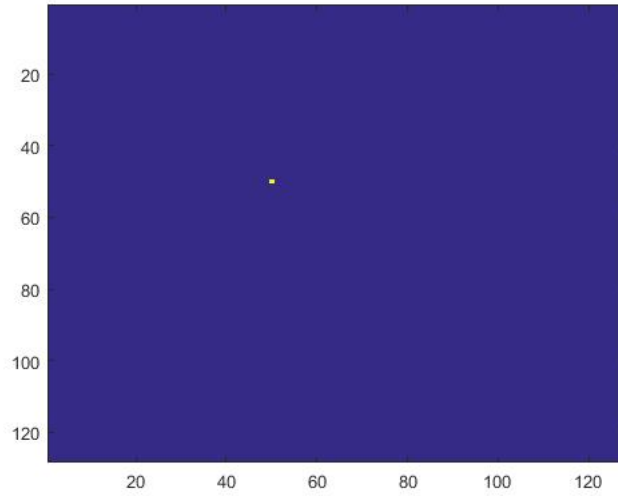


Figure 4: Test object with one illuminated voxel at (50, 50). This noise-free object will be passed through the already constructed \mathbf{H} matrix and will be used as a known for image reconstruction.

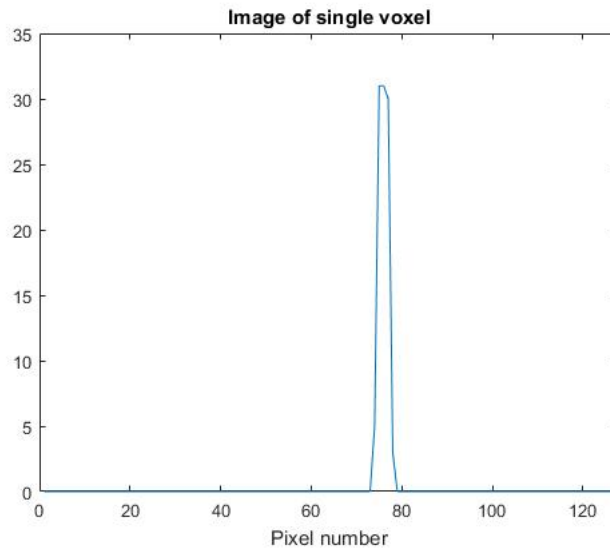


Figure 5: Image vector, \vec{g} , of first test object (with one illuminated voxel) using our \mathbf{H} operator

2.5 Reconstruction with multiple angles

Clearly a single projection is not sufficient to accurately reconstruct the distribution of the radiotracer. Thus, we have to consider a setup where the pinhole and detector are

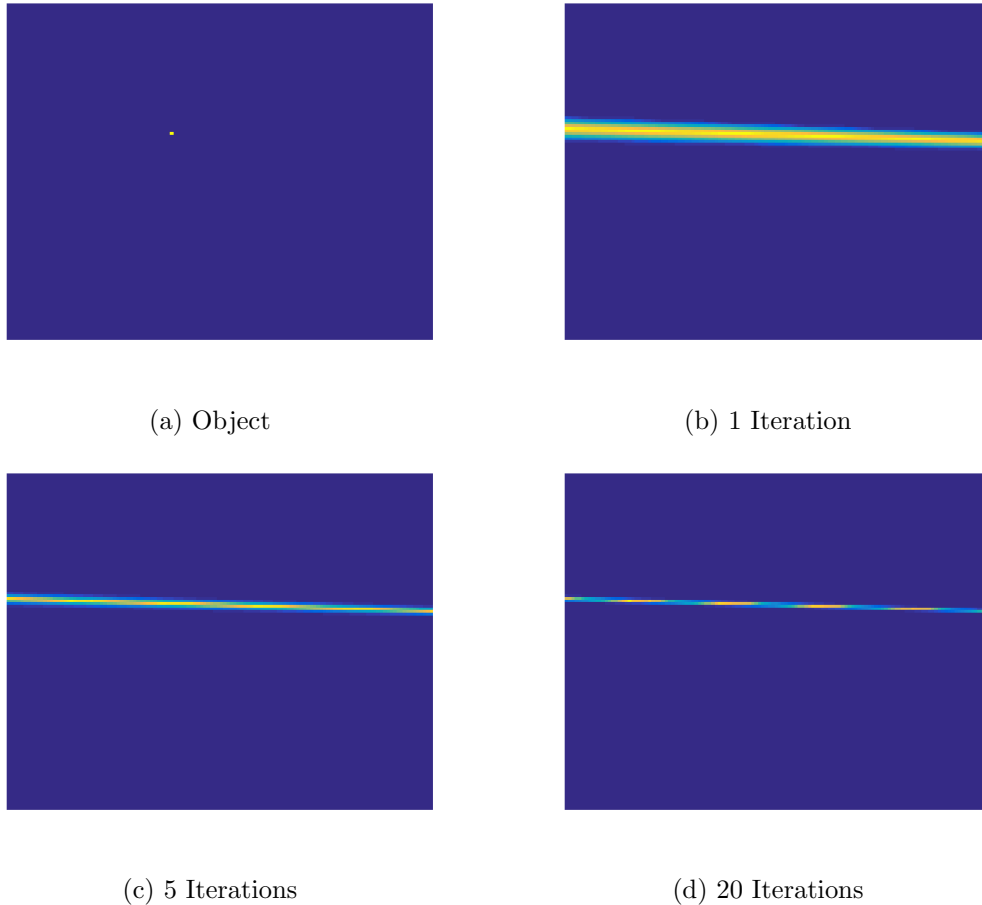


Figure 6: An object with a single illuminated voxel is shown in (a). MLEM reconstructions for various iterations all show a linear streak, geometrically from the point of detection through the pinhole and back onto the object plane.

positioned at a series of different angles around the object. (This is equivalent to rotating the object by that angle, which is the approach we will take for this simulation.)

Consider 5 different angles equally spaced around the object. When using just these 5 angles, the single point reconstructions improve substantially as shown in Figure 7. In these figures, the artifacts cause by the insufficient sampling seen in the previous method (with one angle) are still apparent, but with angular directions based on the angle of the projection. However, with as little as 3 iterations, the “streaks” are virtually unnoticeable. With 20 iterations, there is little noticeable difference between the true object and the reconstruction. We show the error more closely in Figure 8. The point in the object was

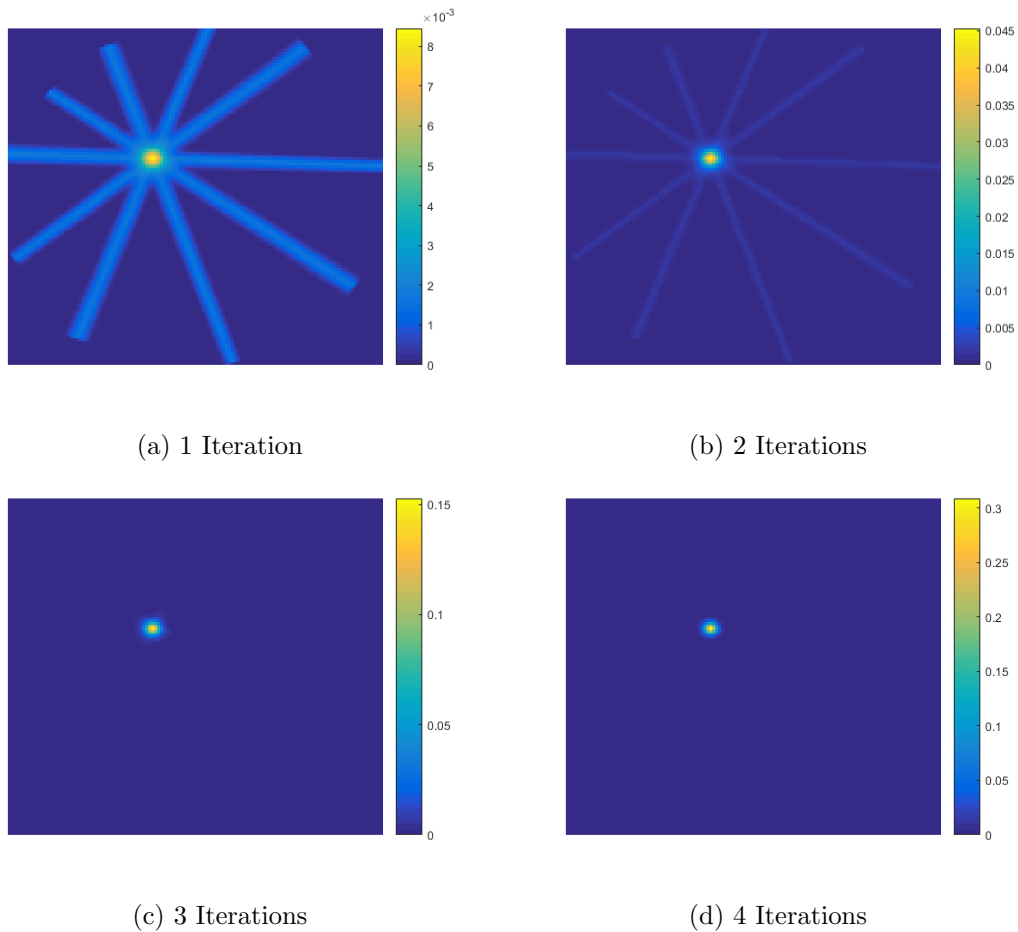


Figure 7: Reconstructing a single-point illuminated object using MLEM iterations, with five angular projections. With as few as 3 iterations, evidence of the “streaked” angles seems to have disappeared, while intensity in the true bright spot increases with each iteration.

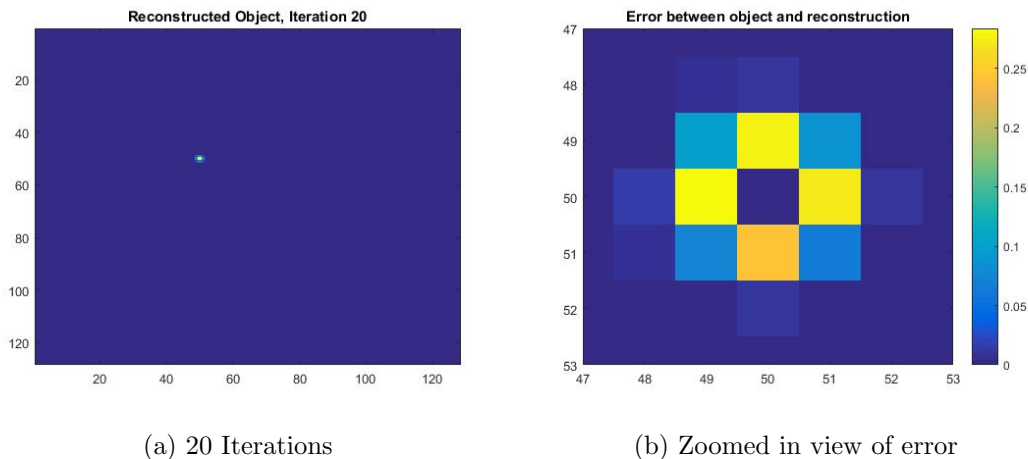


Figure 8: 20 iterations of MLEM with five angular projections and its reconstruction error, centered around the region of interest.

defined to be at voxel (50, 50). Looking at Figure 8b, there is no error at the actual illuminated voxel, but there is a small error in its neighboring voxels.

2.6 Reconstructing a skull

The previous example was simple and had predictably good results. What about a more complicated object? To study this, we utilized a cartoon image of a skull as the object to be imaged and reconstructed (see Figure 9).

For this reconstruction, I chose to use 5 angles. The reconstructions at various iterations of the MLEM algorithm are shown in Figure 9. Artifacts are visible after 20 iterations but it is still easy to determine the object in the field of view.

3 Constructing system matrices: a 3D case

3.1 Setting up a 3D system

Now consider a three dimensional system illustrated in Figure 10. Each voxel is then a three-dimensional volume of size s^3 . The pinhole is a circle with radius r . The detector is made up of two-dimensional square pixels and is itself rectangular. The coordinate system

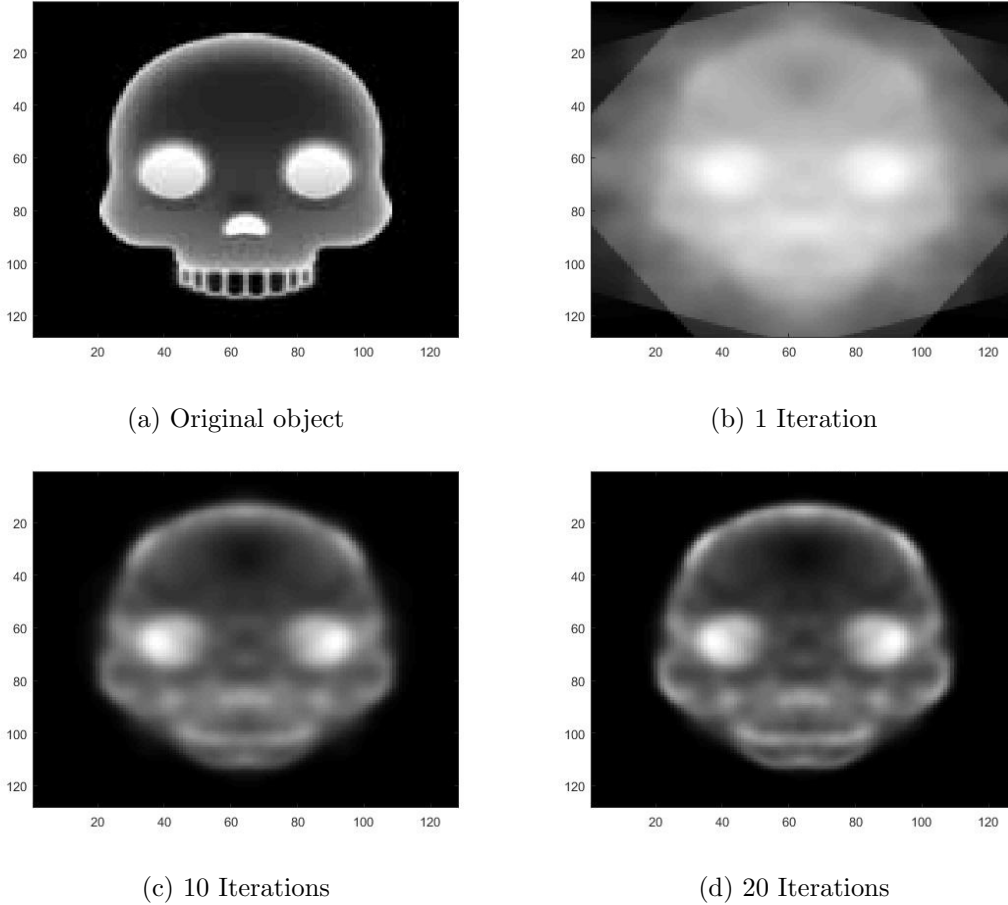


Figure 9: (a) A more complicated test object to reconstruct. (b-d) MLEM reconstructions for various iterations with five angular projections. From 10 to 20 iterations there is not noticeable change in the reconstruction.

is defined as shown in Figure 10, again with the origin in the center of the pinhole. For this more complicated system, we will no longer choose the center of voxel of ray origination, nor will these rays spread or fan evenly through the pinhole. Instead we make the following changes:

- Rays will originate from a random location within the cubic voxel.
- Each ray will pass through a different random location within the pinhole.

Note that a random pinhole location is found by choosing coordinate points (y_r, z_r) where both of these variables are random numbers between $-r$ and r . A new random pinhole co-

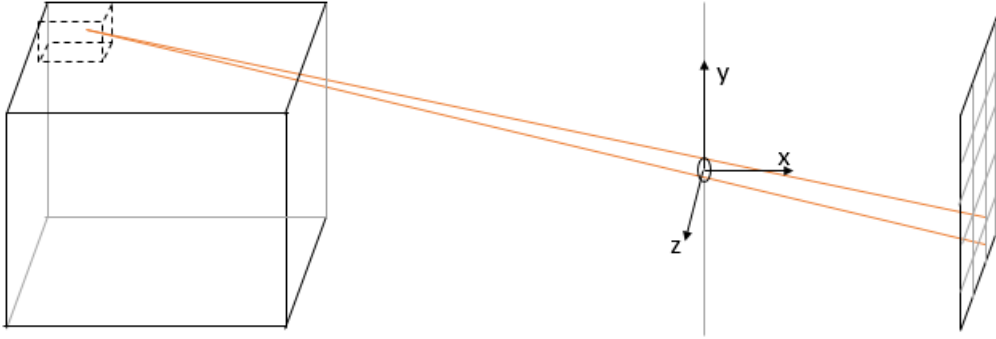


Figure 10: 3D System

ordinate (y_r, z_r) is found if $\sqrt{y_r^2 + z_r^2} > r^2$. This can insure a random spread throughout the pinhole without a concentration in any one area.

For the 3D system, we will also use the following specifications (again, units are arbitrary):

- Size of object: 51 voxels by 51 voxels by 51 voxels
- Number of voxels, N : 132,651
- Side length of each (cube) voxel: 2
- Distance from object to pinhole: 15
- Radius of pinhole: 0.3
- Distance from pinhole to detector: 10
- Number of pixels on detector, M : 2601 (51×51)
- 1000 rays leave each voxel from random locations in the voxel.

The size of the detector is determined automatically by tracing rays from the center of the corner voxels closest to the pinhole through the center of the pinhole. This ensures that our field of view is primarily visible on the detector face.

Each column of the system matrix still represents one specific voxel. For this case, the N voxels are ordered first in the y direction, then the x direction, then the z direction. Each row of the matrix once again represents a particular pixel from the detector, ordered first

by detector columns.

3.2 Verifying the matrix

Viewing the matrix for a 3D system as we did for the 2D system in Figure 3 is not as intuitive, both because the matrix is so large and because it is mostly filled with zeros. Instead, we utilized an alternative method in verifying the accuracy of the matrix. We took an object with one emitting voxel. Let this voxel be in the plane of $(x_o = 26, y_o = 26)$ (where x_o and y_o refer to a coordinate system of the object with respect to the back corner) and have this illuminated voxel move throughout the z direction. As the detector is defined in the $y - z$ plane, we expect an illuminated voxel moving as such to correspond to a pixel moving horizontally across the center of the detector. We can do the same with the illuminated voxel moving in the y direction. Now we expect the pixel on the detector to move vertically (see Figure 12).

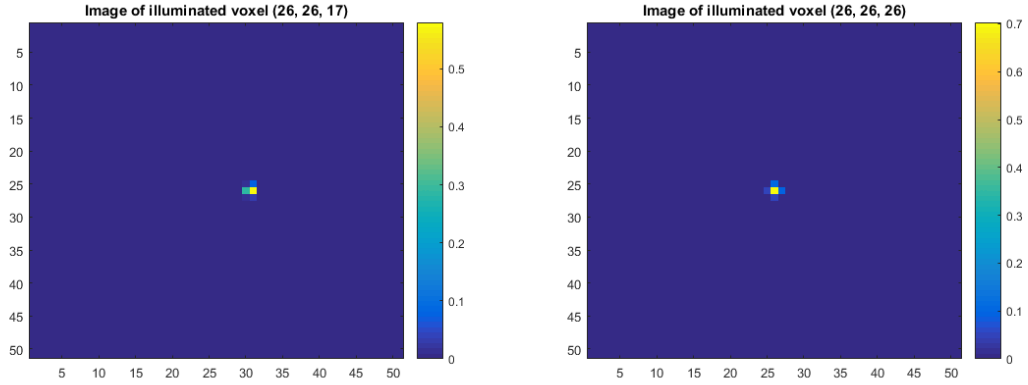
3.3 Single point illuminated

As before, we can test the system matrix with a single illuminated voxel. Let that point be located at $(20, 20, 20)$ on the object. We can actually run the same MLEM algorithm as previously detailed for this 3D object.

Figure 13 shows an MLEM reconstruction with 20 iterations for this single pixel. However, in this first run, only one angle was used. What if we rotate the object as before? Let's try 20 iterations with 5 angles evenly spaced around the object. Compare Figures 13 and 14. Though both of these used 20 iterations, adding 5 angles created a significantly more accurate reconstruction. Slice 20 of the reconstruction had the highest intensity relative to any of the other nearby slices which were fairly negligible.

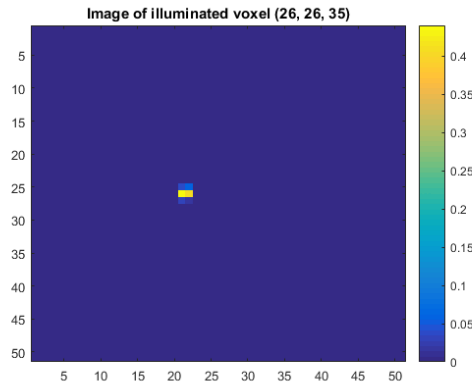
3.4 Multiple illuminated points

Now consider a more complicated object. We shall try 27 symmetrically spaced illuminated voxels in object space (illuminated with an intensity of 1) and each of these voxels surrounded by an illuminated voxel with an intensity of 0.5, to simulate an intensity fall-off. The x , y , and z planes with these 9 high intensity voxels are defined to be 21, 26, and 31. MLEM will be performed with 20 iterations and 5 angular projections.



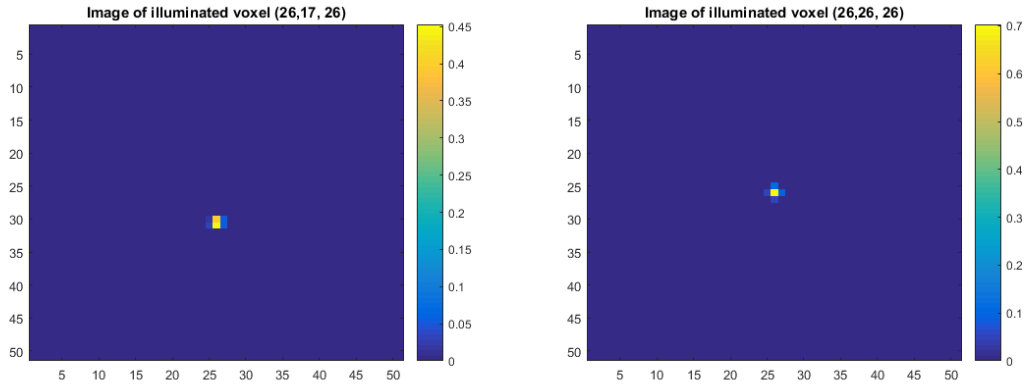
(a) Voxel illuminated at $z_o = 17$

(b) Voxel illuminated at $z_o = 26$



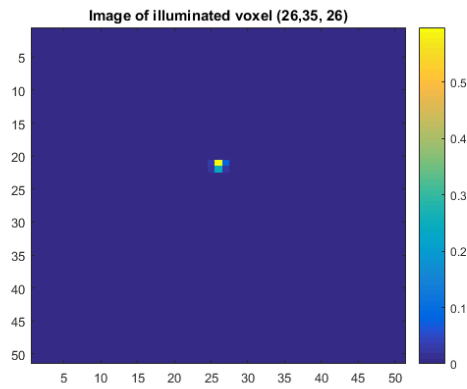
(c) Voxel illuminated at $z_o = 35$

Figure 11: Verifying the 3D matrix with an illuminated voxel moving in the z direction. Each plot shows the image of a voxel illuminated at object point $(26, 26, z_o)$. As expected, the illuminated point moves horizontally, due to the definitions of the system axes.



(a) Voxel illuminated at $y_o = 17$

(b) Voxel illuminated at $y_o = 26$



(c) Voxel illuminated at $y_o = 35$

Figure 12: Verifying the 3D matrix with an illuminated voxel moving in the y direction. Each plot shows the image of a voxel illuminated at object point $(26, y_o, 26)$. As expected, the illuminated point moves vertically, due to the definitions of the system axes.

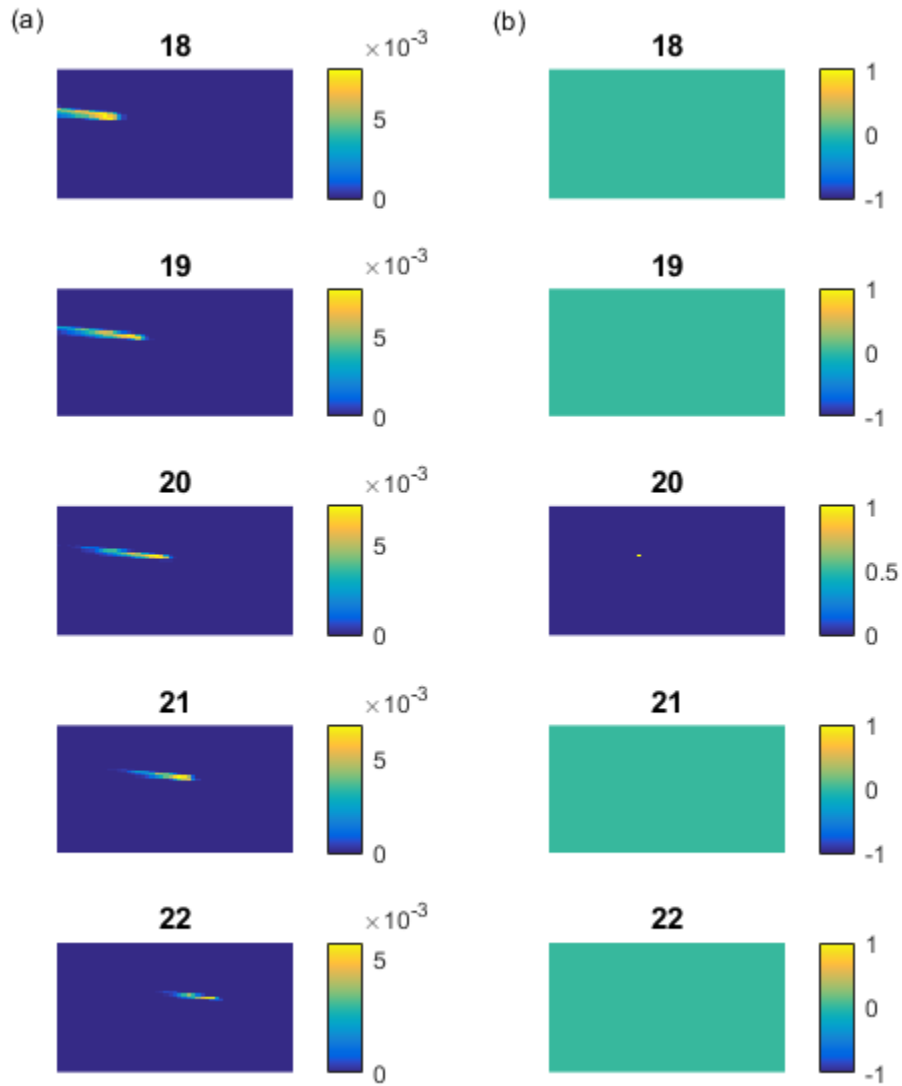


Figure 13: (a) Reconstructing a single illuminated voxel at object slice $z = 20$.
 (b) The original object. The titles for each plot represent the z object slice number.

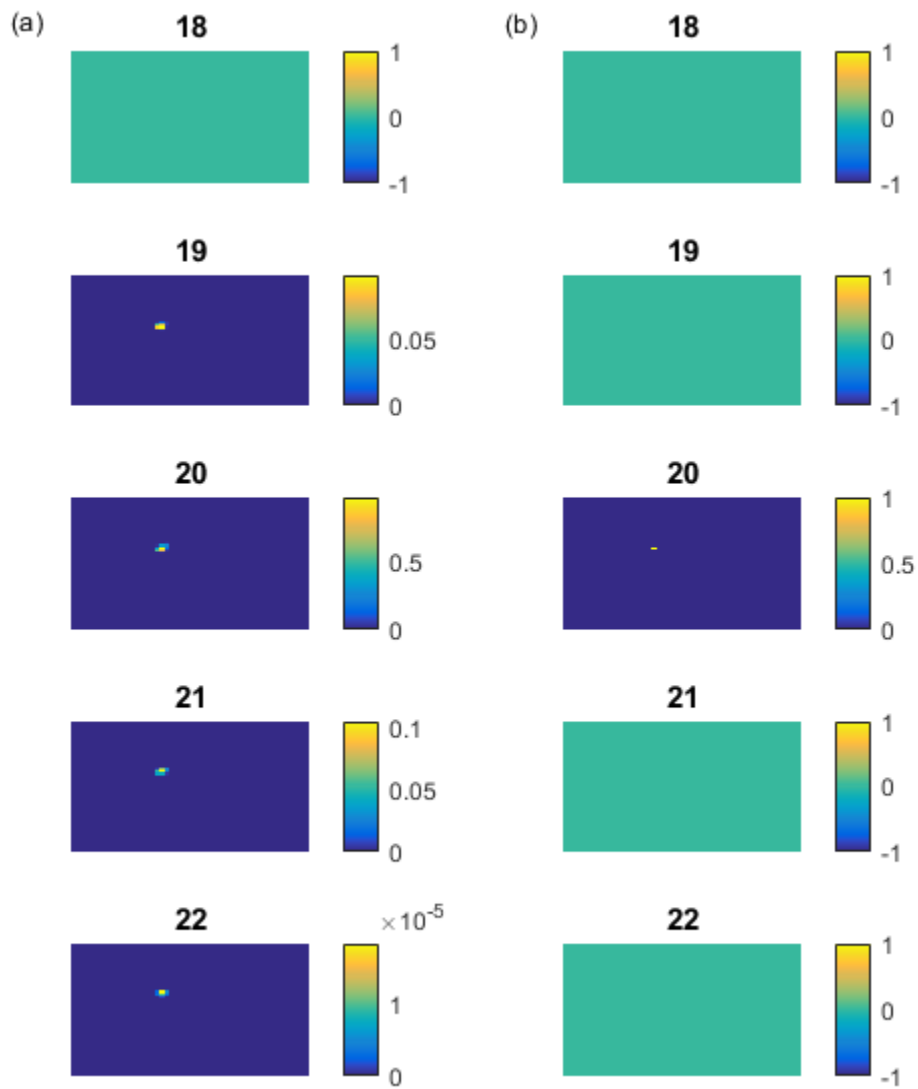


Figure 14: (a) Reconstructing a single illuminated voxel at object slice $z = 20$ with 5 different angles. (b) The original object. The titles for each plot represent the z object slice number.

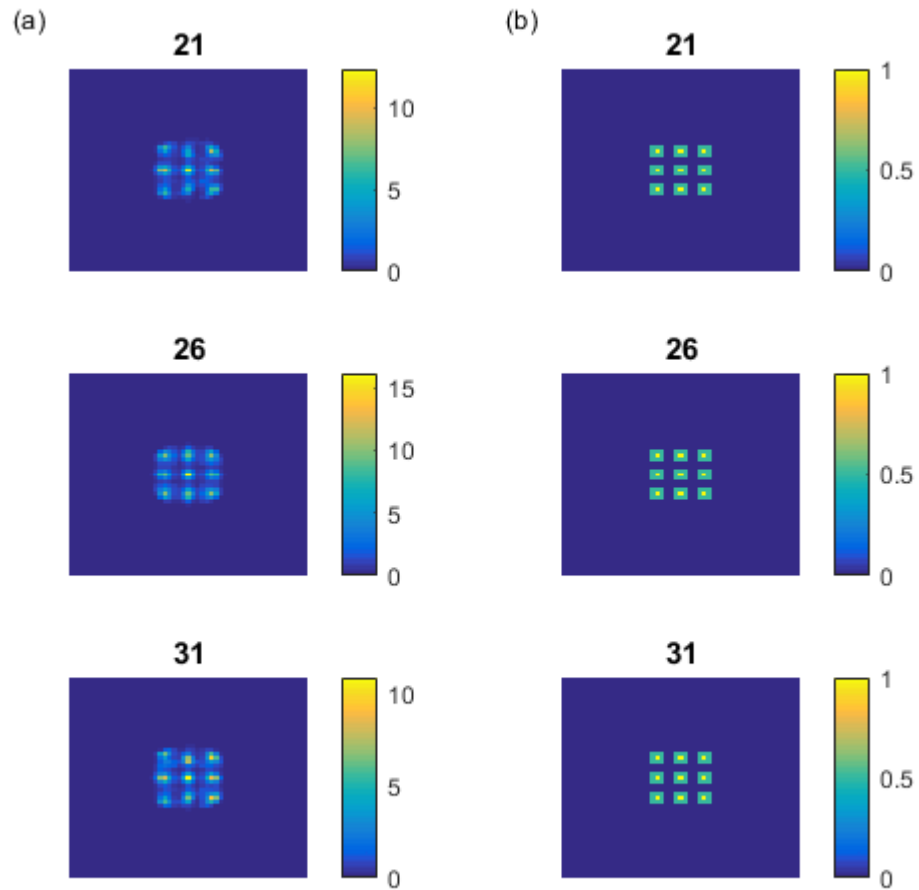


Figure 15: (a) Reconstructing an object with 27 bright voxels surrounded by dimmer voxels. (b) The original object. The titles for each plot represent the z slice number.

As shown in Figure 15, an intensity-varied object was reconstructed with some interesting results. Though each of the three z object slices which had the 9-spot pattern was identical to each other, each of these slices' reconstructions were not. There is a general shape of the 9-spot pattern with significant intensity fall-off surrounding each bright spot.

4 Discussion

4.1 Issues and difficulties

For these results, MATLAB R2016B was used. Calculation-wise, the most time-intensive step of running these simulations is constructing the system matrix. Another difficulty lies in storing the matrix. While the 2D system had an \mathbf{H} matrix with a size 128×128^2 , or 2,097,152 elements, when made sparse it could be saved as a variable of size 2.2 kilobytes. However, in the 3D case, we defined a system with a matrix sized $51^2 \times 51^3$, or 345,025,251 elements. (approximately 172 times more elements than the 2D system matrix). This cannot be saved and stored through MATLAB, even when converted to a 'sparse' matrix (a method which was discussed in Section 2). Instead, clever space-saving functions have to be written which save the non-zero elements and their location in order to rebuild the matrix rather than saving the matrix outright. However, once this matrix is rebuilt, performing image simulations with it is trivial from a computational memory standpoint.

4.2 Next Steps

The next obvious step in the analysis is to consider the effects of detector noise on imaging performance. Beyond that, there should be considerations made to scatter in the object as well as in the collimator and detector system. Further goals include using lessons from these simulations to aid in the design of imaging system geometries. To achieve this, there will need to be modeled realistic physics of the gamma rays, especially when considering their interaction with the pinhole collimator. From there, we could look towards multiple bed positions to achieve a larger field of view for the SPECT system.

References

- [1] M. Dahlbom, *Physics of PET and SPECT Imaging*. Taylor & Francis, 2016.
- [2] M. A. Kupinski and H. H. Barrett, *Small-Animal SPECT Imaging*. Springer, 2005.
- [3] M. N. Wernick and J. N. Aarsvold, *Emission Tomography: The Fundamentals of PET and SPECT*. Elsevier Science, 2004.
- [4] H. H. Barrett and K. J. Myers, *Foundations of Image Science*. Wiley Series in Pure and Applied Optics, Wiley, 2013.