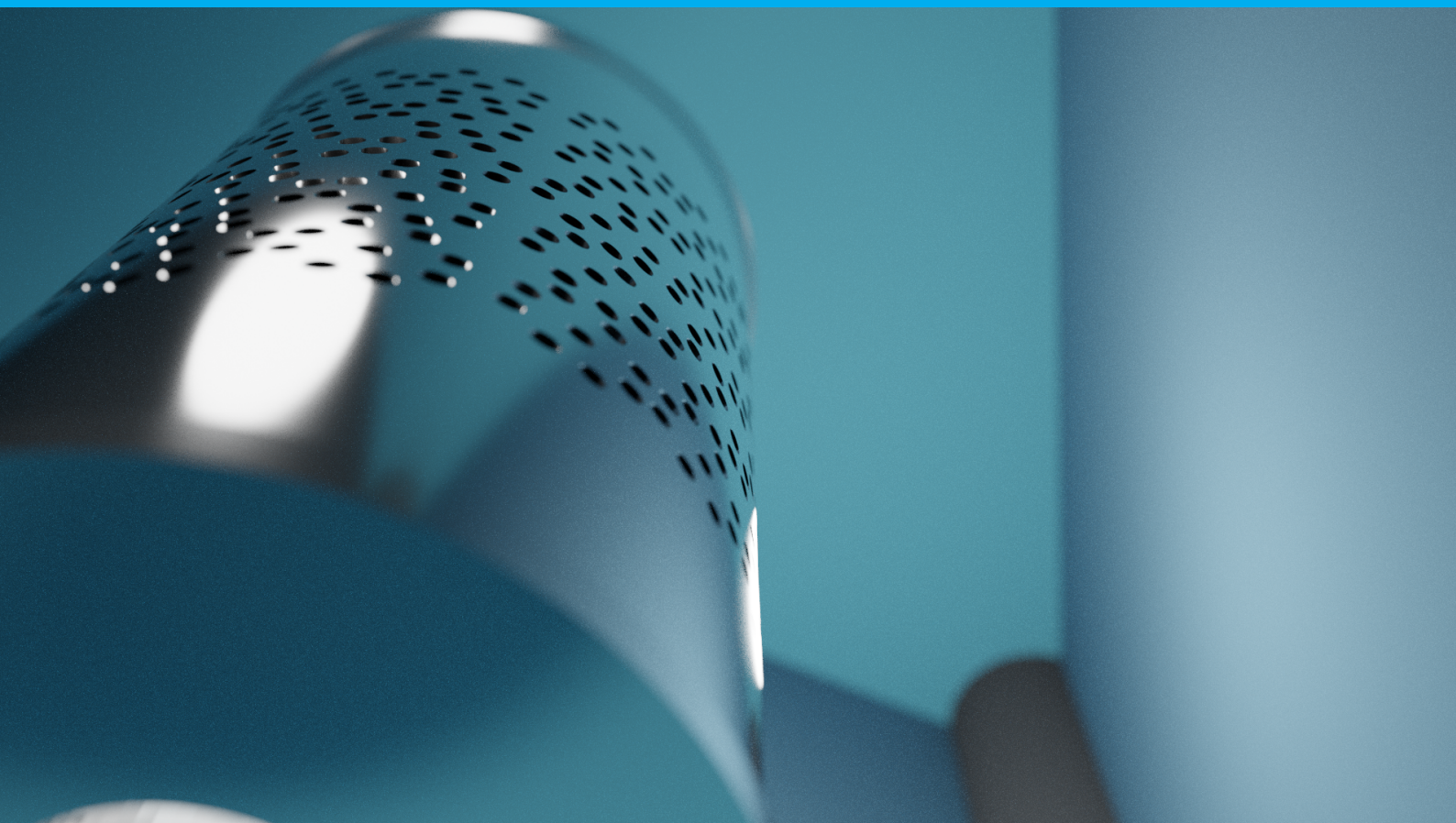


High-energy multi-pinhole col- limator design

Using 3x3 twisted pinhole clus-
ters for high-energy small-
animal SPECT/PET imaging

Sven Kwinkelenberg



High-energy multi-pinhole collimator design

Using 3x3 twisted pinhole clusters for
high-energy small-animal SPECT/PET
imaging

by

Sven Kwinkelenberg

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended publicly on Thursday July 16, 2020 at 3:00 PM.

Student number: 4719522
Project duration: April 20, 2020 – July 16, 2020
Thesis committee: Mrs. Dr. ir. Marlies C. Goorden, TU Delft, supervisor
Prof. dr. Freek J. Beekman, TU Delft

Abstract

With multi-pinhole collimation systems, high resolutions can be reached with both SPECT and PET. Using tracers with higher-energy γ -photons increases the effects of pinhole edge penetration, decreasing the resolution of the system. This research aims to design a collimator using 3x3 twisted pinhole clusters to be used in the VECTor system. This new cluster design should allow high-energy tracers such as ^{89}Zr to be imaged. The sensitivity of this new design was compared at 511 keV and 909 keV with the sensitivity of the VECTor collimator by simulating the sensitivity using GATE.

A collimator design is characterised by its degree of multiplexing and detector coverage. These values are aimed to be the same as for the VECTor collimator to ensure a fair comparison at the simulation stage. Several design approaches have been tested. The final design had the clusters placed in 5 rows on the collimator. The three inner rows had 21 clusters, the two outer rows had 15 clusters. The inner radius of the collimator was increased to prevent pinholes from intersecting.

The pinhole diameters of the final design were modified such that the resolution of the system was the same as the VECTor resolution for both 511 keV and 909 keV photons. To prevent the pinholes from intersecting, either the inner radius of the collimator had to be further increased, or the pinholes had to be smaller. It was chosen to test both options, resulting in a total of three collimator designs to be simulated. A scan lasting 1 hour was simulated with a source the size of the VECTor CFOV. This was done with 511 keV photons using 2 MBq/mL ^{18}F and with 909 keV photons using 2 MBq/mL ^{89}Zr with their respective designs. These simulations resulted in a 1.77 times higher sensitivity to direct photons for the 511 keV collimator, and either a 2.04 or a 2.69 times higher sensitivity to direct photons for the 909 keV photons. Additionally, the total sensitivity of the 511 keV collimator did not significantly change, and the total sensitivity of the 909 keV collimator designs increased with either 8% or 40%. It is concluded that the implementation of 3x3 twisted pinhole clusters in a collimator to be used in the VECTor system has significant benefits over the current collimator when used with high-energy γ -photons.

Contents

1	Introduction	1
1.1	SPECT and PET	1
1.2	Gamma ray collimation	1
1.3	Image reconstruction	3
1.4	Pinhole clusters	3
2	Theory	5
2.1	Multiplexing and detector coverage	5
2.2	Resolution	5
3	Method	7
3.1	Resources	7
3.2	Cluster parameters and pinhole parameters	7
3.3	Multiplexing and detector coverage	8
3.4	Cluster design	10
3.5	Cluster placement	10
3.6	Optimising	11
3.7	Approaches	13
3.7.1	Stacking	13
3.7.2	Single fitting	14
3.7.3	Double fitting	14
3.7.4	Shifted fitting	14
3.8	Equivalent resolution	16
3.9	GATE simulations	16
4	Results	19
4.1	Reference values	19
4.2	Stacking	19
4.3	Single fitting	20
4.4	Double fitting	20
4.5	Shifted fitting	21
4.6	Inner radius increase	22
4.7	Resolution optimisation	22
4.8	Sensitivity simulations	25
5	Discussion	27
6	Conclusion	29
A	Matlab code	31
A.1	Projection	31
A.2	Collimator creation	35
	Bibliography	45

1

Introduction

1.1. SPECT and PET

Molecular imaging using single photon emission computed tomography (SPECT) and positron emission tomography (PET) is an essential part of modern-day biomedical research. The use of radiolabelled biomolecules allows for in vivo imaging of biological processes. For both techniques, a variety of tracers is available, their uses ranging from bone metabolism visualisation to cancer identification [1]. These technologies do not only have clinical applications; small-animal SPECT and PET research is an important field in the development of new, and the improvement of existing pharmaceuticals [2]. Modern small-animal SPECT systems are able to reach resolutions of up to 0.25 mm [3] and dedicated small-animal PET systems are able to reach resolutions of up to 0.6 mm [4].

Both PET and SPECT use radioactive tracers. PET tracers emit positrons, which, upon encountering an electron, annihilate into two γ -photons. Traditionally, PET uses a ring of detectors around the subject to detect these photon pairs. This coincidence detection uses the line connecting the two detected photons and the time difference between the photon counts to compute the origin of the photons [5]. SPECT tracers directly emit γ -radiation, in the form of single γ -photons. Using a collimator to focus these γ -photons on the detector, the origin of the photons can be computed. This is done by blocking photons not on the right path to preserve the information about the position of origin in the detected photons [2]. This same collimation technique can also be applied to the γ -photons originating from positron-electron annihilation in PET. As a result, no coincidence detection is necessary, and image reconstruction is done in the same way as with SPECT image reconstruction. This report will focus on this type of PET imaging, and not on traditional coincidence PET.

As both techniques can be used for a similar type of imaging, complementary sets of tracers can be chosen to image certain biological processes in greater detail. For example, ^{18}F -fluoride and $^{99\text{m}}\text{Tc}$ -HDP are respectively a PET and SPECT tracer, used for the imaging of bone structure [6]. Combining SPECT and PET images made with these tracers could provide more detailed information than either of these images could on their own. A system for simultaneous imaging using PET and SPECT, called VECTor (Versatile Emission Computed Tomography system), has been developed to facilitate the combination of both SPECT and PET images with resolutions of 0.25mm and 0.75mm respectively [7].

1.2. Gamma ray collimation

In order to obtain a projection that can be turned into an image, the γ -photons emitted during a scan are selectively blocked using a collimator to retain the information about their origin. In most clinical systems, this is done using a parallel hole collimator. This type of collimator consists of a slab of highly absorbant material with many small parallel holes. With a sufficiently thick slab, these holes allow only photons perpendicular to the collimator plane to pass through [2]. This gives an image with no magnification of the source object. If the holes are sufficiently small, the resolution of this system is equal to the intrinsic resolution of the detector. This intrinsic resolution in modern systems is of the order of 3.5 mm [2], which is sufficient for most imaging applications on human organs.

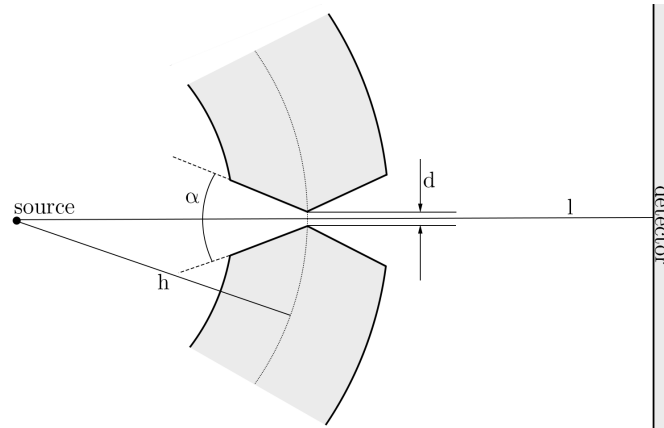


Figure 1.1: A schematic view of a pinhole with opening angle α and diameter d in a collimator tube. The pinhole centre is a distance h away from the source, and the detector is a distance l away from the source.

However, this is often too low for small-animal imaging. In this case, sub-millimetre resolution is desired. To achieve this, either the image must be magnified, or a detector with a better intrinsic resolution has to be used. However, current high-resolution detectors are mostly at the prototype stage, and only suitable for very small systems. Obtaining a large-area, high-resolution detector for commercial systems would therefore be expensive. Magnifying the image on a readily available larger detector is therefore more cost-effective than using a high-resolution detector [2].

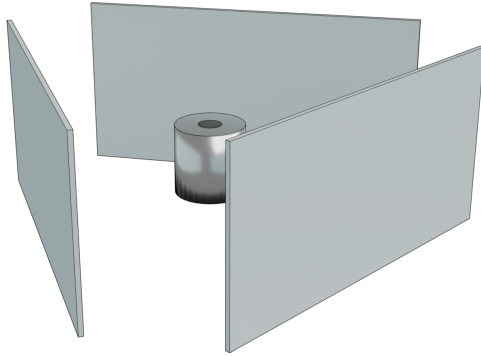
Several different imaging systems exist, all with their own approach to image magnification. Slit-slat systems use a combination of knife-edge slits and a shutter-like plate [8]. Other systems only use slits without the shutters [9]. Some systems use pinholes to magnify the projection [10]. This can either be done with one, or with multiple pinholes. Every system can be either stationary, have a moving or rotating detector, or have a moving bed to image the subject from all sides. This research focuses on a multi-pinhole system with a stationary collimator and stationary detectors [1].

As this system is meant for small-animal imaging, the subject is always close to the collimator. Let h be the distance between the subject and the pinhole, and $l - h$ the distance between the pinhole and the detector as shown in figure 1.1. The image is magnified with a factor $M = (l - h)/h$. While the image is now a factor M larger, the intrinsic resolution of the detector has remained the same [2]. This effectively increases the resolution of the system as a whole. Placing the detectors relatively far from the collimator further increases this magnification and thus the resolution. Pinhole imaging compared to imaging using a parallel hole collimator has a relatively low sensitivity for objects at larger distances from the pinholes. Because photons are emitted from tracers in a random direction, only a very small fraction of the total amount of γ -photons will pass through a pinhole. However, placing an object closer to a pinhole increases this fraction, and thus the sensitivity. For small-animal imaging, a large magnification is desired, which is achieved by placing the subject close to the pinholes, decreasing h and increasing M . This coincidentally increases the sensitivity of the pinholes. As a result, small-animal pinhole systems are not less efficient than clinical SPECT systems, regardless of the fact that they use pinholes. The combination of a large magnification and good sensitivity makes pinhole imaging a popular choice for small-animal SPECT and PET [2].

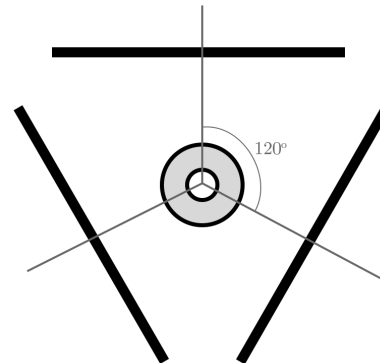
To further increase the sensitivity of a pinhole system, multi-pinhole (MP) systems are developed. Such a system uses multiple pinholes that are all looking at a single area, called the central field of view (CFOV). When a photon from this CFOV departs in a random direction, it has a larger probability to pass through one of the pinholes compared to the probability in a single-pinhole system, increasing the overall sensitivity of the system [11].

The system used during this project is the VECTor system [6], which is based on the U-SPECT-II imaging system [1]. It uses a tube-shaped multi-pinhole collimator. The object to be studied is placed inside the tube. The CFOV lies in the centre of this tube. The collimator is surrounded by three planar gamma detectors. Each

pinhole on the tube projects on one of these detectors. As the tube is surrounded by the detectors, it is not necessary to move either the bed or the detectors to obtain a complete data set for image reconstruction of the CFOV.



(a) A 3D model of the detector and collimator setup. The cylinder in the middle is the collimator tube, and the planes represent the detectors.



(b) A schematic top view of the detector and collimator setup. The circles in the centre represent the collimator tube, while the black lines represent the detectors. The three-fold rotational symmetry of the system is clearly visible.

Figure 1.2: Two schematic representations of the detector and collimator setup of the U-SPECT-II system.

1.3. Image reconstruction

The most popular image reconstruction algorithm used in multi-pinhole systems is maximum-likelihood expectation maximisation (MLEM) [12]. This algorithm uses an iterative approach to estimate the distribution of tracers with increasing accuracy [12].

The volume that is scanned is divided into small, cubic elements called voxels. The system matrix represents, for every voxel-pixel combination, the probability that a photon emitted from a specific voxel is detected in a specific pixel. This system matrix is unique for every system, and has to be either measured by scanning the imaging volume with a point source, or by accurately simulating the system [6] [13] [1]. The algorithm starts by assuming a tracer distribution in the reconstruction volume. Using the system matrix, the projection of this assumed distribution is calculated. This expected projection is then compared to the measured projection, and the assumed tracer distribution is modified so that its theoretical projection more closely resembles the measured projection. This new projection is again compared to the real projection. Repeating this process many times (iterations) brings the assumed distribution closer to the real distribution [12].

This is only a basic explanation of the image reconstruction process. Modern systems use more advanced versions of the MLEM algorithm, such as OSEM. These modifications to the algorithm allow the calculations to be greatly accelerated compared to traditional MLEM [14]. The VECTOR system uses the similarity-regulated ordered subsets expectation maximisation (SROSEM). This algorithm, compared to pixel based OSEM algorithms (POSEM) [14], can speed up the image reconstruction up to 11.5 times [15].

1.4. Pinhole clusters

The photons used for SPECT imaging typically have an energy in the range of 115 - 260 keV [6]. Even though the collimator material is highly absorbant for γ -radiation, some photons will be able to penetrate a small distance into the collimator material. When this happens near the edge of a pinhole, the photon has a chance to penetrate through this edge and reach the detector. This pinhole edge penetration is one of the main sources of image degradation in pinhole imaging [2]. Photons produced by the positron-electron annihilation during PET have a much higher energy: 511 keV. This higher energy allows the photons to penetrate further into the

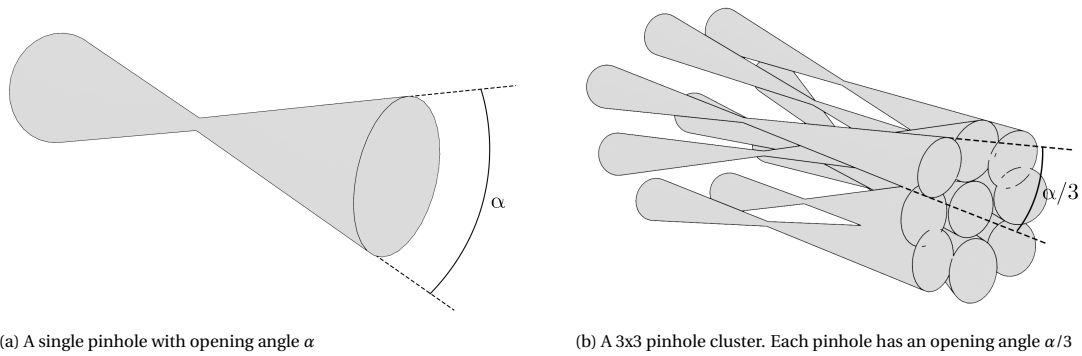


Figure 1.3: The concept of a 3x3 pinhole cluster. The 9 pinholes in the right image observe the same FOV as the single pinhole on the left, but the opening angles of the pinholes in the cluster are three times as small.

collimator material, increasing the influence of pinhole edge penetration.

The effects of pinhole edge penetration can be reduced by decreasing the opening angle of the pinholes. A smaller opening angle gives the photons more material to penetrate through, decreasing their chances of reaching the detector. However, as the CFOV is determined by the opening angles of the pinholes, shrinking the opening angles shrinks the CFOV. As a solution, VECTor implements pinhole clusters. A pinhole cluster consists of several pinholes grouped together. The pinholes are placed such that they collectively observe the entire CFOV. Every individual pinhole now only has to view a smaller portion of the CFOV, thus its opening angle can be decreased. This way, the opening angles of the individual pinholes can be reduced without sacrificing the CFOV or the sensitivity of the system [16]. The VECTor system uses 48 2x2 clusters, divided into four rings [17].

Further increasing the amount of pinholes per cluster allows even smaller opening angles to be used. This, in turn, allows higher energies to be imaged. This could for example allow ^{89}Zr to be used as a tracer for both PET and SPECT. ^{89}Zr traditionally is a PET tracer, used for the tracking of antibodies or the identification of different tumour types [18] [19]. It decays to $^{89\text{m}}\text{Y}$ by emitting a positron. This positron emission can be used for PET imaging. $^{89\text{m}}\text{Y}$ in turn decays to the stable ^{89}Y by sending out a 909 keV photon. Traditionally, as the difference in energy between the PET photons (511keV) and the 909 keV photons is so large, the $^{89\text{m}}\text{Y}$ -radiation is simply filtered away [18]. Using VECTor, it is possible to use this 909keV radiation for imaging instead. The development of a high-energy collimator using 3x3 twisted pinhole clusters could allow the 909 keV peak of ^{89}Zr to be used for SPECT imaging with even higher resolution.

Different types of twisted pinhole clusters are currently being researched for use with VECTor. This project focuses on the implementation of 3x3 twisted pinhole clusters. These clusters consist of 9 pinholes, collectively observing a CFOV the size of the VECTor CFOV. The use of 'twisted' clusters allows the opening angles to be even smaller than is possible when using their non-twisted counterparts [20]. This is done by 'twisting' the pinholes around each other. This twisting also makes the cluster more compact, allowing more clusters to be placed on the collimator. An example of what a 3x3 twisted pinhole cluster looks like is shown in figure 1.3. The goal of this project is to design a collimator using 3x3 twisted pinhole clusters that can be used for SPECT/PET imaging using ^{89}Zr with decreased effects of pinhole edge penetration.

2

Theory

Several parameters are important in determining the quality of a collimator design. This chapter defines and explains the most important concepts used during this study.

2.1. Multiplexing and detector coverage

Two important characteristics of a collimator design are the degree of multiplexing [21] and the detector coverage. Together, they determine how efficiently the detectors are used.

When a pixel on a detector has multiplexing, it means that at least two pinholes are projecting on that pixel. The degree of multiplexing is calculated using

$$\% \text{multiplexing} = \frac{\text{number of pixels with multiplexing in the projection}}{\text{number of pixels with nonzero value in the projection}}. \quad (2.1)$$

Typically, a higher value for the multiplexing results in a system with a higher sensitivity. Since a higher multiplexing usually means there are multiple pinholes, and sensitivity increases with the number of pinholes, sensitivity increases with the degree of multiplexing. The downside of multiplexing is that some information is lost in the regions where there is multiplexing; the origin of a photon found here is no longer certain.

To maximise the efficiency of the system, it is crucial that as many pixels as possible are used in the image reconstruction. To quantify this, the detector coverage is defined:

$$\text{detector coverage} = \frac{\text{number of pixels with nonzero value in the projection}}{\text{total number of pixels}}. \quad (2.2)$$

A higher value for the detector coverage means that the detector is more efficiently used, with a value of 1 suggesting that no pixel is left unused.

Generally, a system with a higher detector coverage has a higher sensitivity. A high degree of multiplexing generally is not desired, as it results in an increase in image noise and reconstruction artefacts [21]. A certain degree of multiplexing is tolerated if it results in a higher detector coverage. The effects of higher or lower degrees of multiplexing on the overall image quality and efficiency of the system are explained in [21].

2.2. Resolution

Not only the sensitivity is important for an imaging system; its spatial resolution is also an important characteristic. This resolution typically consists of two parts: the intrinsic resolution of the detectors, and the geometrical resolution of the collimator. The former is a characteristic of the detectors and can not be changed without changing the detectors themselves. The geometric resolution is a result of the geometric properties of the collimator used in the system. This resolution is normally defined as the full width half maximum

(FWHM) of the point spread function. Assuming an ideal system with no edge penetration, the geometric resolution is given by

$$\lambda_g = d \left(1 + \frac{1}{M} \right), \quad (2.3)$$

where d is the pinhole diameter, and M is the magnification of the system [22]. M can be calculated as

$$M = \frac{l-h}{h}. \quad (2.4)$$

In this equation, l , h and d are defined as in figure 1.1.

As a result of increasing pinhole edge penetration with increasing energy, the point spread function of the pinhole widens. This, in turn, lowers the overall resolution of the system. To take the effects of edge penetration into account, the diameter d in equation (2.3) is replaced by the resolution effective diameter d_{re} . This diameter is defined as the diameter of a pinhole without edge penetration, which would have the same geometric resolution as a real pinhole with edge penetration [6]. The resolution effective diameter can be calculated as follows:

$$d_{re} = d \left(1 - \ln(k) \frac{A}{d} \right) \quad \text{with} \quad A = \frac{\tan\left(\frac{\alpha}{2}\right)}{\mu}. \quad (2.5)$$

Here, $k = 0.5$ for FWHM. d is the diameter of the pinhole without correction for edge penetration, α is the opening angle of the pinhole, and μ is the linear attenuation coefficient of the collimator material. For the VECTor system, this material is tungsten [6]. The attenuation coefficient depends on the energy of the imaging photons. A derivation of equations (2.5) can be found in [23]. The collimator in this project will be tested for photons with energies of 511 keV and 909 keV, matching the photon energies of respectively ^{18}F and ^{89}Zr [18] [19].

In equation (2.5), the second term between brackets in the first equation represents the influence of edge penetration. As $\ln(k)$ is negative for $k = 0.5$, a larger value for A increases the resolution effective diameter. A smaller opening angle α would decrease the value of A , and with it the influence of edge penetration. As can be seen from equation (2.3), a smaller resolution effective diameter, as a result of smaller opening angles, gives a higher spatial resolution.

To allow for a fair comparison of the sensitivity of the new collimator designs with the VECTor collimator, the resolution of the new designs is set equal to the VECTor resolution. Setting the expressions for the geometrical resolutions (2.3) of the VECTor and twisted pinhole system equal to each other, the following expression is found:

$$d_{\text{twist}} = d_{\text{VECTor}} - \frac{\ln(k)}{\mu} \cdot \left[\tan\left(\frac{\alpha_{\text{VECTor}}}{2}\right) - \left(\frac{\alpha_{\text{twist}}}{2}\right) \right]. \quad (2.6)$$

Here, d_{twist} and d_{VECTor} are respectively the diameters of the pinholes of the twisted clusters and the diameters of the VECTor pinholes. The angles α_{twist} and α_{VECTor} respectively are the opening angles of the twisted cluster pinholes and the VECTor pinholes. Equation (2.6) gives the necessary diameter of the pinholes in the 3x3 clustered collimator in terms of the parameters of the clusters on the VECTor collimator. Using this diameter for the pinholes of a 3x3 clustered collimator should yield the same resolution as the VECTor system.

3

Method

This chapter describes the process of designing and optimising the collimator design. The working principles of the scripts will be described, and the different approaches to designing a collimator are explained. All scripts described in this chapter are written and executed in Matlab [24].

3.1. Resources

The performance of the collimator designs made during this project will be compared to the VECTor collimator. To ensure a fair comparison, it is attempted to have the degree of multiplexing and the detector coverage of the new designs as close to the VECTor values as possible. Therefore, the multiplexing and detector coverage of the VECTor system have to be calculated. For this, a file containing the exact specifications of the VECTor collimator was provided. A script reads and loads the necessary data into variables used in the code. The collimator specification file contains for each pinhole the detector it is projecting on, a set of three spatial coordinates (x_{ph} , y_{ph} , z_{ph}), the orientation angles θ and ϕ , the opening angle α and the pinhole diameter. This data is formatted in the form of a table, each column containing one of the aforementioned variables in the order they were mentioned. Each row contains the complete set of variables for one pinhole. This formatting is kept during the entirety of this project.

A script detailing the placements of the three detectors in the VECTor system is also provided. The detector specification file contains three sets of corner coordinates of the detectors, as well as the number of pixels on each detector. A separate file specifies a subset of pixels at the edges and corners of the detectors that should not be used when calculating the multiplexing.

The design for a 3x3 clustered pinhole is provided as a script, which creates a set of positions, orientations and opening angles for 9 pinholes. This script forms the basis for the scripts that create and place the 3x3 clusters later on in this project.

Finally, a file is provided that takes in the collimator specifications from a file, and converts this data into an OpenSCAD [25] file (.scad). This file can be opened in the program OpenSCAD, showing a 3D model of the collimator tube.

3.2. Cluster parameters and pinhole parameters

During this project, each cluster and each pinhole is characterised by a set of parameters. For the clusters, these are: three spatial coordinates x , y and z , two orientation angles θ and ϕ and the opening angle α . The clusters also have a twisting angle ϕ_r . This determines how much the pinholes are twisted around each other. The value used for this throughout this report will be 6.124 degrees. The pinholes have two additional parameters: the detector they are projecting on and the pinhole diameter d . In order to differentiate between the coordinates and angles of the clusters and of the pinholes, the coordinates and angles of the clusters get a subscript c . The positions of the pinholes get a subscript 'ph', to prevent confusion with the coordinates of the pixels later on. The definitions of the parameters are shown in figure 3.1.

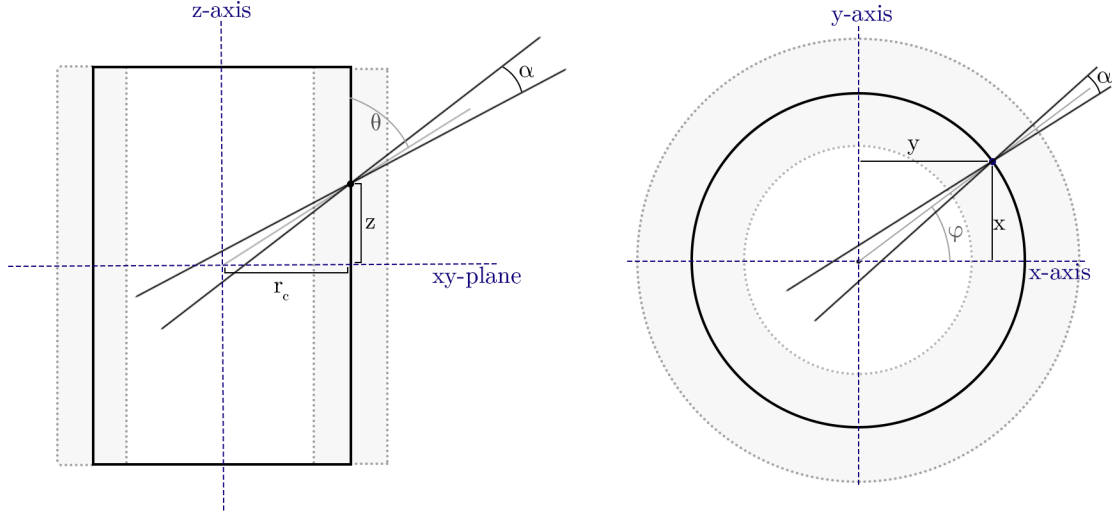


Figure 3.1: An overview of the pinhole and cluster parameters, as seen from the side (left) and the top (right) of the collimator tube. The coordinates are given in a Cartesian coordinate system, with its origin in the centre of the collimator tube and the z -axis parallel to the collimator tube. The grey shaded areas represent the collimator material. The solid black lines represent the radius on which the pinhole and cluster centres lie.

3.3. Multiplexing and detector coverage

To calculate the degree of multiplexing and the detector coverage, first the projection of the collimator must be known. To calculate the projection, the detectors are divided into separate pixels. Every pixel is assigned a 3D position. The x, y, z -position of each pixel is now represented by three variables: $d, i,$ and j . These respectively represent the detector the pixel is a part of and the column and row of the pixel.

When starting the calculation to find the projection, three matrices are defined, one for each detector. These matrices, called the projection matrices P_d , store the information about the projection. Each matrix is a $N_i \times N_j$ matrix, with N_i and N_j respectively the number of columns and rows on the detector. The subscript d shows which detector this projection matrix belongs to. Initially, all values in all three projection matrices are 0.

The projection is calculated by testing, for each pinhole, which pixels can be 'seen' by this pinhole. A pixel is defined as 'visible' by a particular pinhole if the line connecting the pinhole centre and the pixel makes an angle smaller than half the opening angle with the pinhole axis. If this angle is larger than half the opening angle, the pixel is deemed 'invisible' by this pinhole. To calculate if a pixel is visible, the following vectors are defined:

$$\vec{p}_{\text{ph}} = \begin{pmatrix} x_{\text{ph}} \\ y_{\text{ph}} \\ z_{\text{ph}} \end{pmatrix}, \quad \vec{p}_{\text{px}} = \begin{pmatrix} x_{\text{px}} \\ y_{\text{px}} \\ z_{\text{px}} \end{pmatrix} \quad (3.1)$$

where \vec{p}_{ph} is the vector pointing from the origin to the pinhole centre, and \vec{p}_{px} is the vector pointing from the origin to the pixel centre. These two vectors are used to calculate the vector pointing from the pinhole centre to the pixel, \vec{p}_{proj} , using

$$\vec{p}_{\text{proj}} = \vec{p}_{\text{px}} - \vec{p}_{\text{ph}}. \quad (3.2)$$

In order to find the angle between this projection vector and the pinhole axis, the pinhole axis has to be quantified. Here, the pinhole axis is defined as the axis which passes through the centre of the pinhole and is parallel to the orientation of the pinhole. With this, an orientation vector \vec{o} with the same direction as the pinhole axis can now be defined. The orientation vector points in the direction the pinhole is projecting in, which is towards the detector. The orientation vector is calculated using θ and ϕ :

$$\vec{o} = \begin{pmatrix} r \sin \theta \cos \phi \\ r \sin \theta \sin \phi \\ r \cos \theta \end{pmatrix} \quad (3.3)$$

where choosing $r = 1$ returns a normalised orientation vector. The angle between the projection and orientation vector is the same as the angle between the projection vector and the pinhole axis. The angle between two vectors \vec{a} and \vec{b} is easily calculated using $|\vec{a}||\vec{b}|\cos\alpha = \vec{a} \cdot \vec{b}$ where α is the angle between the two vectors and the dot signifies an inner vector product. For the vectors defined above, this now becomes:

$$\gamma = \arccos \frac{\vec{p}_{\text{proj}} \cdot \vec{o}}{|\vec{p}_{\text{ph}}||\vec{o}|}. \quad (3.4)$$

Here, γ is the angle between the orientation and projection vector (γ is used to prevent confusion with the opening angle α). When γ is smaller than $\alpha/2$, with α the opening angle of the pinhole, the pixel defined as visible. Otherwise it is invisible.

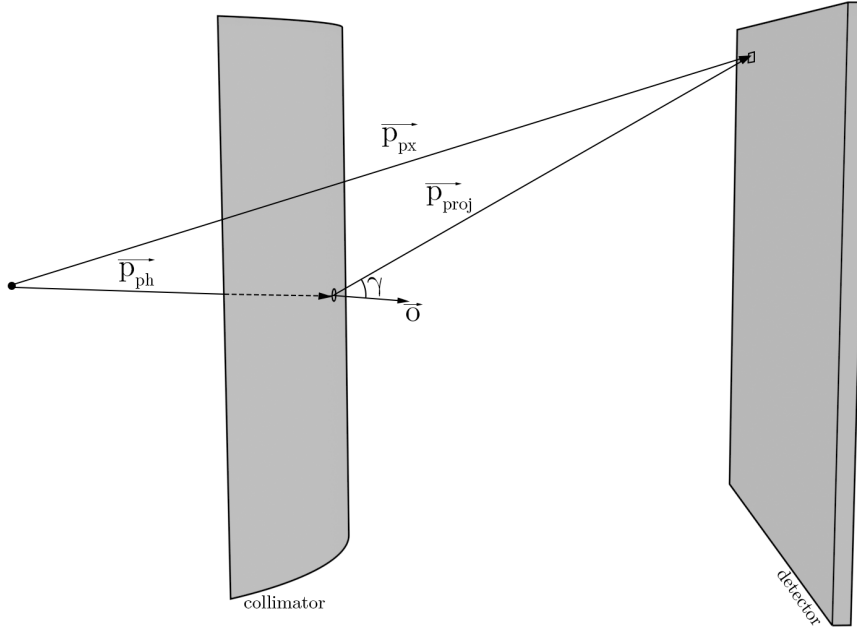


Figure 3.2: The vectors and angles determining whether a pixel is visible for a pinhole or not. If the angle γ between the orientation vector and the projection vector is larger than $\alpha/2$, the pixel is not visible. Otherwise, the pixel is visible.

Now, a script is written with a loop that checks every pixel for every pinhole on the collimator to see whether it is visible or invisible. If a pixel is visible, its corresponding value in the projection matrix is increased by one. If it is invisible, it remains the same. This way, the value of a pixel in the projection matrix reflects how many pinholes can project on this pixel.

Every pixel that has a value of 1 or more in the projection matrix P_d is a pixel with nonzero value in the projection (see equation (2.1)). Every pixel with a value of 2 or more in the projection matrix is seen by at least two pinholes, and therefore a pixel that has multiplexing. By simply counting the number of pixels that have a value of 2 or more in the projection matrix, and dividing this by the number of pixels that have a value of 1 or more in the projection matrix, the degree of multiplexing is obtained.

Due to the nature of the detectors, the pixels close to the edges and corners do not provide reliable information. Because of this, the information of these 'bad pixels' is often discarded during the image reconstruction process. To have the degree of multiplexing more accurately reflect the system properties, a correction has to be made for these pixels. Directly after the projection matrices have been calculated, all the values that correspond to the 'bad pixels' are set to 0. This is equivalent to saying that these pixels can not be seen by any pinhole. They therefore take no part in the calculation of the multiplexing.

The detector coverage is calculated simply by counting the number of pixels with a value of 1 or more in the projection matrix, and dividing this by the total number of pixels, not including the 'bad pixels'.

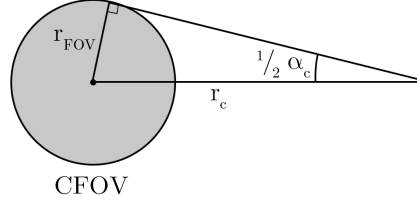


Figure 3.3: A schematic overview of the parameters determining the opening angle of a cluster.

3.4. Cluster design

As described in section 3.1, a script creating a 3x3 twisted pinhole cluster at a certain position and with a certain orientation is provided. The cluster is created based on the parameters r_{FOV} , r_c and ϕ_r . Respectively, these are the radius of the FOV of the cluster, the distance from the cluster centre to the central axis of the collimator (here the z -axis), and the twisting angle. Here, r_c is equivalent to h in figure 1.1. A cluster created by this script is placed with the cluster centre on the z -axis, at $z_c = -r_c$ (left image of figure 3.4). The general orientation of the cluster is in the positive z -direction, towards the origin. All individual orientations, given as normalised orientation vectors, signify in which direction the pinholes are 'looking'. A modified version of this script is used in the following sections, where all the orientation vectors are inverted to show in which direction the pinholes are projecting. This is so that the orientation vectors created by this script are consistent with the orientation vectors defined with equation (3.3).

The FOV and r_c together define the opening angles of the pinholes. The total opening angle of the cluster is given by

$$\alpha_c = 2 \arcsin \frac{r_{\text{FOV}}}{r_c} \quad (3.5)$$

where α_c is the opening angle of the cluster. The opening angle of each pinhole is then $\alpha = 1/3\alpha_c$. The twisting angle determines how much the pinholes twist around each other. The default value for a non-overlapping projection of a cluster with an FOV of 6 mm is $\phi_r = 6.124$ degrees.

3.5. Cluster placement

The script above only creates a single cluster, and always at the same position. These clusters then have to be re-orientated according to θ_c and ϕ_c and moved to their final positions x_c, y_c and z_c . The cluster is brought to the correct position with the following steps:

1. The cluster is rotated from its initial position into the xy -plane by rotating it $-1/2\pi$ around the y -axis.
2. From the final x_c and y_c coordinates, the angle ξ is calculated, defined as $\xi = \arctan_2(y_c/x_c)$.
3. The cluster is rotated ξ degrees around the z -axis.
4. The cluster is shifted up an amount z_c to its final position.
5. The orientations of the cluster pinholes are rotated $\pi + \theta_c$ radians around the y -axis, and ϕ_c radians around the z -axis.

Rotating the cluster around a certain axis is done by rotating the 9 vectors describing the centre positions of the individual pinholes around this axis. Similarly, rotating the orientation is done by rotating the orientation vectors around the given axis. Rotating the vectors around an axis is done using rotation matrices, defined as:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, \quad R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \quad R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

where $R_x(\theta)$, $R_y(\theta)$, $R_z(\theta)$ are the matrices to rotate a vector an angle θ around respectively the x -, y - or z -axis. The matrix-vector product of a rotation matrix and a position or orientation vector gives the vector rotated around the axis belonging to that rotation matrix. The process of rotating and shifting the cluster into position is shown in figure 3.4.

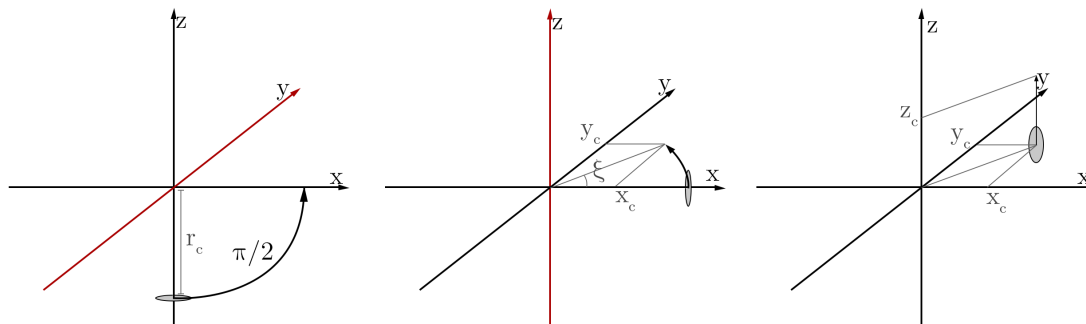


Figure 3.4: The process of placing a cluster at the right coordinates. The left image corresponds to placement step 1. The middle image corresponds to placement steps 2 and 3. The right image corresponds to placement step 4. If something is rotated around an axis, this axis is shown in red.

As rotating and shifting the position of the cluster does not affect its orientation, the orientation of a cluster is still pointing along the negative z -direction after it is brought to the correct position. A rotation of π around the y -axis points the orientation of the cluster in the positive z -direction. A second rotation of θ_c around this same axis ensures that the orientation of the cluster has the right θ -value. A final rotation of ϕ_c around the z -axis also sets the ϕ value for all the pinholes to the correct value. These steps are implemented in a script so that only a file detailing the positions and orientations of the clusters has to be provided. This script then places the clusters at these positions and returns a file with the individual pinholes.

3.6. Optimising

For the optimisation of the placement of the clusters on the collimator, an iterative approach is chosen. First, positions and orientations for all the cluster centres on the collimator are defined and saved in a file. Then, the clusters themselves are placed on their respective positions as defined by this file. This is done using the steps shown in figure 3.4. The positions and orientations of all the individual pinholes are then saved in a new collimator specifications file. Using this file and the calculations described in section 3.3, the projection, degree of multiplexing and detector coverage of the new collimator design are calculated. These values, together with the projection, can then be used as a basis for new modifications to the design to get closer to the desired values.

During the optimisation process, the clusters are often placed close together. Although so far, pinholes were treated as infinitely small points mathematically, they are not single points with infinitely small diameter. In reality, pinholes can best be described as hourglass-shaped tubes through the collimator material. When clusters are placed too close together, these tubes can start to intersect each other. It is important that in the final design, there must at no point be any intersection of the pinholes, as this makes the manufacturing of the collimator virtually impossible. As can be seen from figure 3.5, the points in the collimator material where pinhole intersection is most likely to occur is close to either the inner or outer wall of the collimator. The pinholes are widest here, and therefore take up the most space on the collimator.

It can not be easily deduced from the information provided by the multiplexing script whether the pinholes intersect or not. To assure that no overlap is present in the collimator, the 3D representation is used. By changing the outer radius of the collimator in the 3D model, it is possible to check for overlap at different depths in the collimator material.

The VECTor system possesses three-fold rotational symmetry. Therefore, the clusters on the collimator will be divided into three groups. All groups are identical, apart from a shift in position; each group is rotated $2/3\pi$ around the central collimator axis. All clusters in a group project on the same detector. The centre of a group is defined such that a cluster placed there would project in the centre of a detector. To place a cluster in a group, only two coordinates are needed: z_c and ξ_c . Here, z_c is again the z -coordinate of the cluster centre, and ξ_c is the rotation of the cluster angle in the xy -plane, relative to the cluster centre.

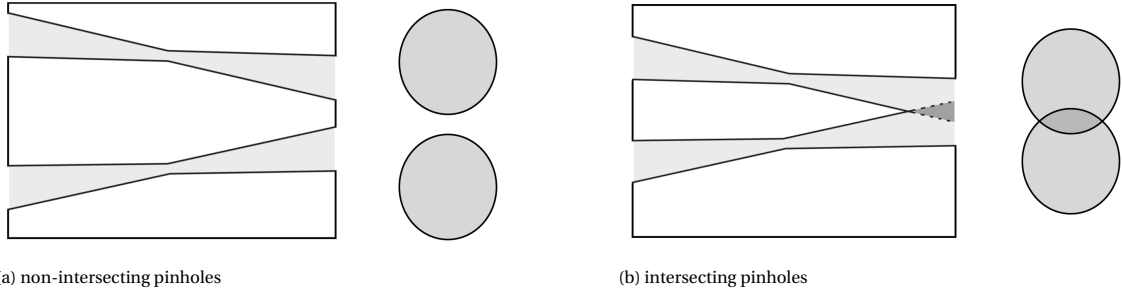


Figure 3.5: An example of intersecting and non-intersecting pinholes. The grey shaded areas are the pinhole tubes, the white area is the collimator material. The grey circles show a cross section of the pinhole tubes at the right side of the side view.

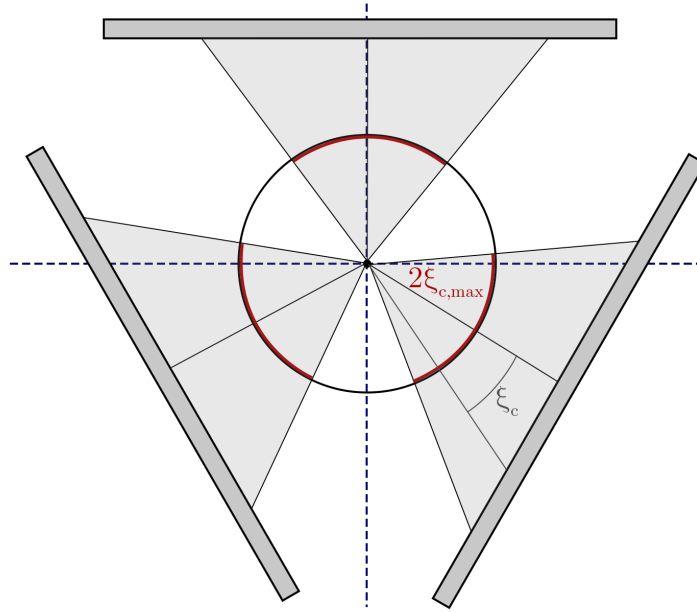


Figure 3.6: A schematic representation of the groups and group angles. Each grey area is represents a group, with the red curves representing the width of the group. The black line in the centre of each group represents the angle $\xi_c = 0$. Each group is identical to the others, but rotated $2/3\pi$ around the z -axis.

Every group is divided into rows. The rows are numbered from top to bottom, with the lowest row being number 1. Each row is centred around the group centre, and the width of a row is defined as twice the maximum angle $\xi_{c,\max}$, as shown in figure 3.6. All the clusters in a row are linearly spaced between $-\xi_{c,\max}$ and $\xi_{c,\max}$. If a row only contains one cluster, its ξ_c is set to 0. The cluster FOV and twisting angle ϕ_r can be manually set for each row.

With the number of rows, and for each row the number of clusters, z -position, width, FOV and twisting angle manually specified, a script is used to calculate the coordinates of the cluster centres. These are then used to calculate θ_c and ϕ_c for each cluster, so that it looks at the origin of the collimator tube (figure 3.1):

$$\theta_c = \arctan \frac{r_c}{z_c}, \quad \phi_c = \arctan \frac{y_c}{x_c}. \quad (3.7)$$

Because of the rotational symmetry, the parameters of only one group have to be defined. The positions and orientations of the clusters are then calculated for one group. The other two groups are created by simply copying the existing group, and rotating all positions and orientations $2/3\pi$ around the z -axis. After this, the individual pinholes can be placed at the correct positions and the projection can be calculated.

Pinholes from clusters that are far away from the centre of their group can sometimes overlap with pinholes from neighbouring groups. These pinholes can be safely removed from the collimator, as clusters this

far from the group centre do not project on the detector, and are therefore not important for image reconstruction or sensitivity purposes.

3.7. Approaches

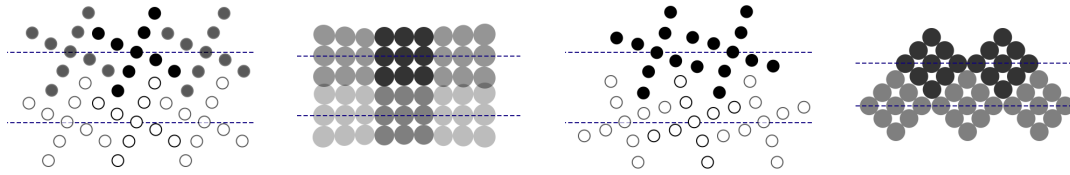
During this project, several different approaches to designing a collimator are explored. All parameters described here are for a single group, but as the groups are identical, these parameters effectively describe all groups. Unless specified otherwise, the default parameters for the collimators are the same as for the VECTOR system. These values are given in table 3.1 [1].

Table 3.1: The default parameters of interest of the VECTOR system, along with their symbols and values.

Parameter	Symbol	Value
Inner collimator diameter	r_{in}	24 mm
Outer collimator diameter	r_{out}	67 mm
Pinhole-to-centre radius	r_c	32 mm
Distance to the detector	r_{det}	210 mm
Pinhole diameter	D	0.700 mm

3.7.1. Stacking

The first approach to optimising the collimator design is by creating a central row at $z = 0$. The projection of a single cluster on the detector is approximately square. The width of the cluster row is set to a value where the projections of the clusters connect at the sides, but do not overlap. For a single row, this is the most efficient way of distributing the cluster projections. The number of clusters in the central row is determined by the amount of clusters necessary to cover the full detector width. New rows are placed directly above and below the central row. The widths of these rows are optimised using the same principle as the central row. New rows are added on top of and below the existing rows until the detector is fully covered. This approach will be referred to as 'stacking'. An example of two rows stacked on top of each other is shown in figure 3.7a. In this figure, 9 circles of the same shade of grey form the projection of one cluster.



(a) The method where the square-shaped projections are 'stacked' on top of each other, here shown for 2 rows of 3 clusters each.

(b) The method where the diamond-shaped projections are 'fitted' in between each other, here shows for 2 rows, where the bottom row has 3 clusters, and the top row has 2 clusters. The clusters in the top row are also flipped to allow this kind of placement.

Figure 3.7: A schematic visualisation of the different methods of placing the cluster rows together on the collimator. The blue dashed lines show the z_c positions of the clusters. The left image shows what the clusters would look like on the inner wall of the collimator. The right image shows what the projection would look like in an ideal case.

Because clusters with a larger ξ_c are further away from the detector than the ones with smaller ξ_c (see figure 3.6), their projections cover a larger area on the detector. If the row directly above the central row were to be placed such that its projection exactly touched the projection of the central row for $\xi_c = 0$, large areas of overlap would exist at the edges of the detectors. This is already visible for three clusters per row in figure 3.7a. To prevent these large areas of overlap, the z -factor is introduced:

$$z_c = z_c - \left| \frac{\xi_c}{\xi_{c,\max}} \right| \cdot \text{sign}(z_c) \cdot z\text{-factor}. \quad (3.8)$$

Here, $\text{sign}(z_c)$ is -1 if z_c is negative, 1 if z_c is positive, and if $z_c = 0$, it is 0. The z -factor moves clusters up or down on the collimator depending on their value of ξ_c . A negative z -factor shifts clusters with a larger absolute value of ξ_c away from the xy -plane. Using this z -factor, the large areas of overlap at the edges of the projection can be prevented.

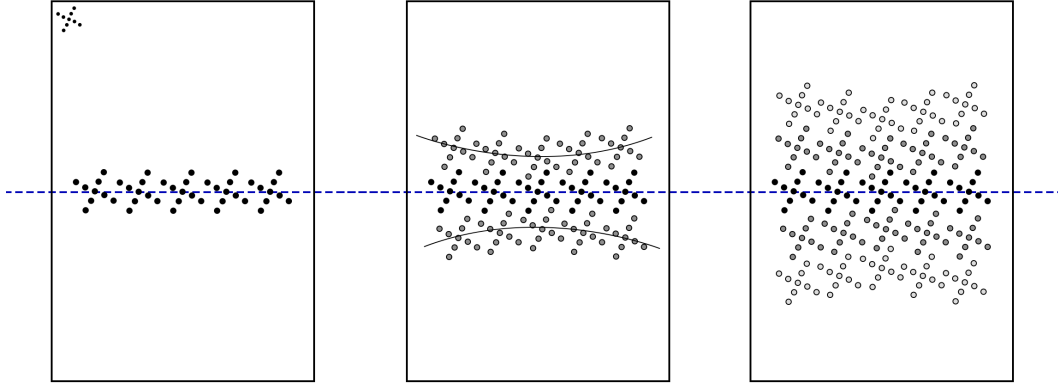


Figure 3.8: The process of stacking on a collimator. First, a central row is placed on the collimator (left). Then, a row above and below this row are added (middle). The curved lines in this image show the effect of a negative z -factor. In the right figure, more rows are added on top and below until the detector is filled. The blue line represents the $z_c = 0$ plane. The shape in the top left corner of the left figure represents the shape of a single cluster on the inner wall of the collimator tube.

3.7.2. Single fitting

Another approach to placing the clusters in the groups is a process referred to as 'fitting'. For this, the clusters are rotated $1/4\pi$ radians around their axis. This changes the shape of the projection from a square to a diamond. Rotating the cluster around its axis is done by rotating the cluster θ_r around the z -axis before bringing it to its final position on the collimator.

Again, first a central row is placed at $z_c = 0$. The clusters are spaced such that the corners of the now diamond-shaped projections are touching, but not overlapping. As this approach uses a single central row, it will be referred to as 'single fitting'. Assuming the central row contains n clusters, the rows above and below it now need to contain $n \pm 1$ clusters. The projections of these upper and lower rows can then be 'fitted' in between the projections of the central row. An example of the projection of two rows fitted together in an ideal configuration is shown in the right image of figure 3.7b. Rows are added above and below the existing rows until the detector is fully covered. Every new row always has 1 cluster more or 1 cluster less than the row above or below it.

The shape of the clusters does not allow the clusters to be packed as closely together as desired. To fit the clusters closer together still, they have to be mirrored; right after the creation of a cluster, the positions and orientations of the pinholes are mirrored (flipped) with respect to the x -axis. The clusters are flipped every other row. The results of both the rotation and mirroring of the clusters is illustrated in figure 3.7b, where the bottom cluster row is flipped. An example of the process of designing a collimator with the single fitting approach is shown in figure 3.9.

3.7.3. Double fitting

Contrary to single fitting, 'double fitting' uses two central rows, placed at $z_c = \pm z$. The pattern of flipping every other row, as well as every row always having one cluster more or less than its neighbours, is maintained. An example of the double fitting approach is shown in figure 3.10.

3.7.4. Shifted fitting

All previously described approaches have only one way of changing the vertical position of a cluster projection on a detector, which is by shifting the cluster up or down on the collimator. This fourth approach, called 'shifted fitting', introduces a second way to change the vertical position of a cluster projection. This is done by changing the θ_c -value of the clusters. As assured by equation (3.7), each cluster always looks directly at the centre of the collimator tube. This ensures that the entire CFOV is always seen by every cluster. Manually increasing or decreasing the value of θ_c shifts the projection of the cluster down or up on the detector. Now, the θ -shift θ_{sh} is defined:

$$\theta_{c,f} = \theta_{c,i} + \theta_{sh} \quad (3.9)$$

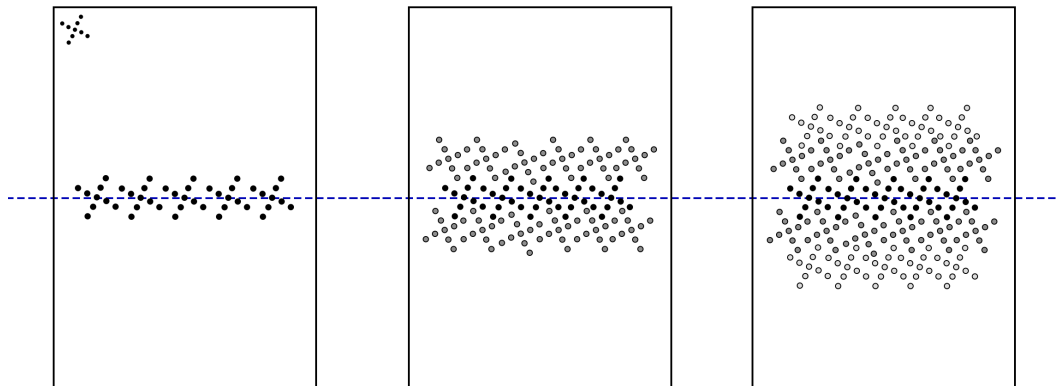


Figure 3.9: The process of designing a collimator using the single fitting approach. First, a central row is placed on the collimator (left). Then, rows above and below this central row are added (middle). These rows have one cluster more or less than the central row. The clusters in these rows are flipped. Finally, more rows are added on top and below the existing rows (right) following the same pattern of flipping. The blue line represents the $z_c = 0$ plane. The shape in the top left corner of the left figure represents the shape of a single cluster on the inner wall of the collimator tube.

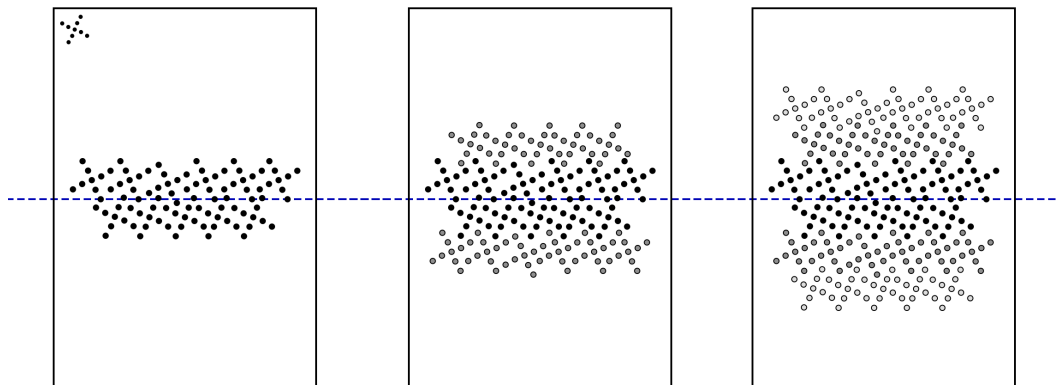


Figure 3.10: The process of designing a collimator using the double fitting approach. First, two central rows are placed at equal distances from the $z_c = 0$ plane (left). One of these rows is flipped. Then, rows are placed on top and below these central row, where adjacent rows always have one cluster more or less than their neighbours (middle). Every other row is flipped. In this figure, the bottom row and the third row are flipped. Finally, more rows are added on top and below the existing rows (right), preserving the pattern of flipping every other row. The blue line represents the $z_c = 0$ plane. The shape in the top left corner of the left figure represents the shape of a single cluster on the inner wall of the collimator tube.

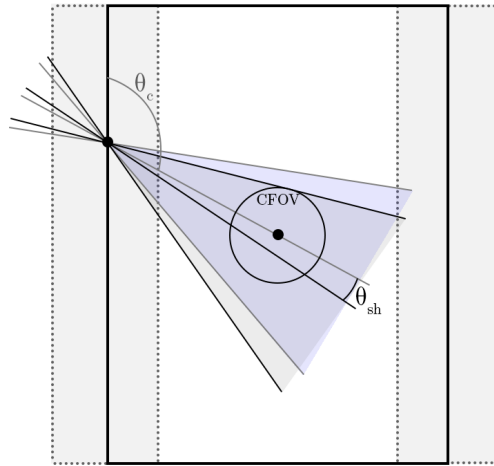


Figure 3.11: A schematic view of the effects of θ_{sh} . The blue area is the FOV of a cluster with orientation angle θ_c . The grey area is the FOV of the same cluster, but shifted an angle θ_{sh} . The circle in the middle is the CFOV that should always be observed.

where $\theta_{c,f}$ is the final value of θ_c after the shift, and $\theta_{c,i}$ is the initial value, before the shift. Changing the orientation of a cluster also changes the FOV of this cluster. The FOV remains the same size, but is shifted up or down. The result is that a θ -shifted cluster is no longer observing the entire CFOV. Increasing the FOV of the individual clusters allows to slightly change the orientation angle while still observing the entire CFOV (figure 3.11).

Increasing the FOV of the clusters does however increase the opening angles of the pinholes. This in turn increases the amount of pinhole edge penetration, which is exactly what is to be prevented with the new design.

Apart from the θ -shifting and increasing the FOV, the process of designing a collimator with this approach is the same as with double fitting.

3.8. Equivalent resolution

After a design made with one of the approaches above is chosen as the final, optimal design, the pinhole diameters have to be set so that the resolution of this new collimator is the same as for the VECTor collimator. All projections and 3D visualisations in the previous sections are done for pinholes with an infinitely small diameter. The diameters of the pinholes of the final design are calculated using equation (2.6). This way, the new collimator designs can be compared to VECTor at equal resolution. The attenuation coefficient μ in equation (2.6) is energy-dependent. This means that for every energy, a different pinhole diameter is needed to compare the collimator design to VECTor. The increase in diameter could cause some pinholes that were previously not intersecting to now intersect each other. After the pinhole diameter has been changed, the designs are manually optimised again to remove any possible overlap as a result of the new diameter.

3.9. GATE simulations

Sensitivity scans using the final designs are simulated with GATE (version 8.0) [26], a simulation package designed to be used in combination with GEANT4 [27]. GEANT4 is a software package specifically developed to simulate the interaction between particles and matter using monte carlo simulations. GATE uses the physical models and calculations available through GEANT4, and applies them specifically to emission tomography [26].

A simulation is run on the collimator design where a sample the size of the VECTor CFOV and a known activity is placed in the collimator tube. In this project, the CFOV size was an ellipsoid with size 4.5x6x6 mm. It was filled with 2 MBq/mL of either ^{18}F or ^{89}Zr . The projection of this sample is simulated for an hour (sim-

ulated time) to obtain the sensitivity of the system. The total sensitivity is given as the percentage of the total amount of emitted photons that is detected by the detector in a 20% window around the photopeak. The three components making up this total are given as well. These are the direct photons, the attenuated photons and the scattered photons. The sensitivity to the direct photons is given as the percentage of emitted photons that passed through a pinhole and directly reached the detector, triggering a count there. A high sensitivity to direct photons is desired, as these are the photons that are best suitable for image reconstruction. The sensitivity to the attenuated photons is given as the percentage of the emitted photons that were attenuated on their way to the detector, but still detected. This attenuation is a result of the photons passing through the collimator material. The sensitivity to attenuated photons therefore is a measure of the amount of edge penetration taking place during the simulation. The sensitivity to scattered photons is given as the percentage emitted photons that were detected, but were scattered along the way. As a result of the scattering, these photons do not end up where they are expected on the detector, decreasing the contrast of the reconstructed images.

The VECTor collimator is also simulated with a sample the size of its CFOV for the same time as the new design. This allows the values of the new design to be compared to the VECTor values.

4

Results

This chapter explains the different approaches to optimising the collimator design, as well as the problems that were found in the process. For the calculations done in this chapter, the values in table 3.1 are used as the default parameters for the VECTor system.

4.1. Reference values

The projection of the VECTor collimator on the detectors was calculated. From this, the degree of multiplexing and the detector coverage were obtained. For this design, the degree of multiplexing is 0.1472, and the detector coverage is 0.9558. The projection of this collimator is shown in figure 4.1.

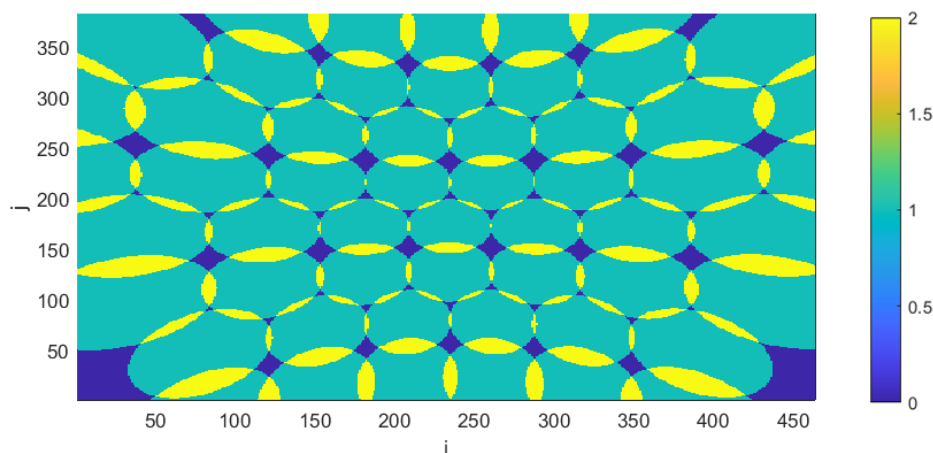


Figure 4.1: The projection of the VECTor collimator on detector 1 of the system. Only the projection on detector 1 is given, because the projections on detector 2 and 3 are identical.

4.2. Stacking

The first collimator design is made using the 'stacking' approach of the previous chapter. This design has a total of 669 pinholes, divided into 5 rows with 15 clusters each. The few missing pinholes are a result of the removal of overlapping pinholes at the group edges. The degree of multiplexing is 0.0248, and the detector coverage is 0.7652. The projection of this design is shown in figure 4.2. A 3D model of this collimator is shown in figure 4.7a.

Using this approach, no satisfactory design could be reached. As a result of the cluster design, it was not possible to place the rows close enough together to optimally cover the detector. Placing the rows close

enough together for the projections to touch would inevitably result in intersecting pinholes. This intersection primarily took place near the inner wall of the collimator tube. Here, there is less space for the pinholes than near the outer wall. In the remainder of this report, all the mentioned overlap will be at the inner radius of the collimator. All possible overlap at other depths in the collimator will always be visible at the inner diameter.

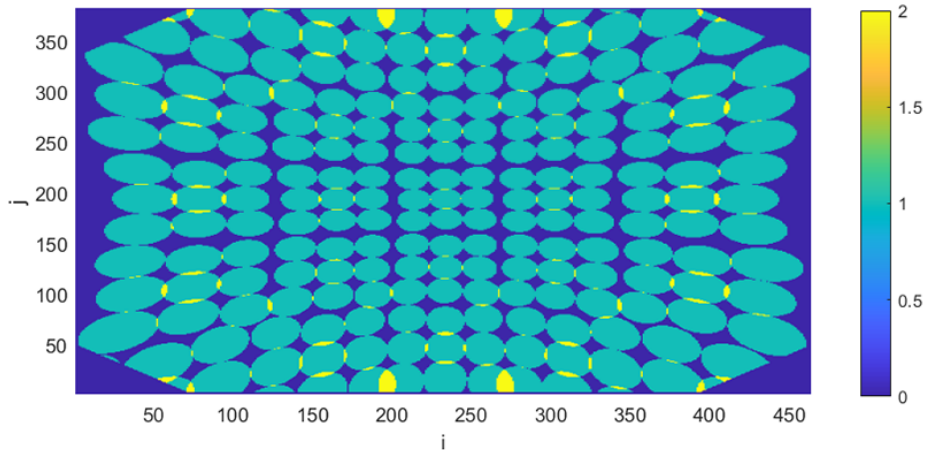


Figure 4.2: The projection of the first collimator design using 3x3 clusters on detector 1. As a result of the cluster groups, the projections on detector 2 and 3 are identical and thus omitted. It can be seen that the projections in the middle row (row 3) do not completely touch. Also, row 2 and 3, as well as 3 and 4 do not connect.

4.3. Single fitting

A second design is made using the 'single fitting' approach. This design has a total of 669 pinholes, divided over 5 rows with 15 clusters per row. The degree of multiplexing of this system is 0.1124, and the detector coverage is 0.7163. The projection of this design is shown in figure 4.3. A 3D model of this collimator is shown in figure 4.7b.

The single fitting design has both a higher value for the multiplexing and a lower detector coverage than the stacking design, both of which is undesirable. It is important to note that the detector coverage of this design could be improved by decreasing the width of the rows. This would however also increase the degree of multiplexing. Projections of individual pinholes would almost completely overlap, resulting in more reconstruction artefacts and a lower image contrast. This brings more disadvantages than are compensated for by the increase in detector coverage.

Additionally, this design has large unused strips of detector area between the centre row (row 3) and the rows directly above and below it (respectively row 4 and 2) in the projection. These strips are a result of the fact that the rows can not be placed closer together without the clusters overlapping.

4.4. Double fitting

Following the single fitting design, a design is made using the 'double fitting' approach to try to eliminate the unused detector strips of the single fitting design. The projection of the design made using the double fitting approach is shown in figure 4.4. This design has a total of 654 pinholes, divided over 6 rows. From top to bottom, these rows have 9, 12, 15, 12, 15 and 12 clusters. This approach has a detector coverage of 0.7312 and a degree of multiplexing of 0.0525. A 3D model of this collimator is shown in figure 4.7c.

As can be seen from figure 4.4, the unused detector strips between the central rows and the outer rows are smaller. However, now there is an extra strip present between the two central rows. Again, it is not possible to place these rows closer together without the pinholes intersecting. The detector coverage of this design is

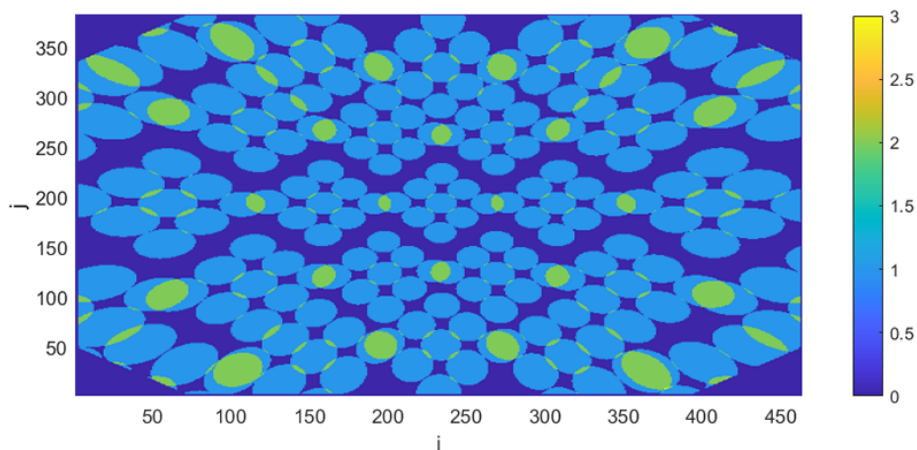


Figure 4.3: The projection of the collimator design using diamond-shaped cluster projections. As a result of the cluster groups, the projections on detector 2 and 3 are identical and thus omitted. The gaps between row 2 and 3, and 3 and 4 are clearly visible. It can also be seen that there are pinholes whose projections almost completely overlap in the outer rows.

slightly higher than when using single fitting, but it is still lower than when using the stacking approach. This means that it is still not an improvement compared to the first design.

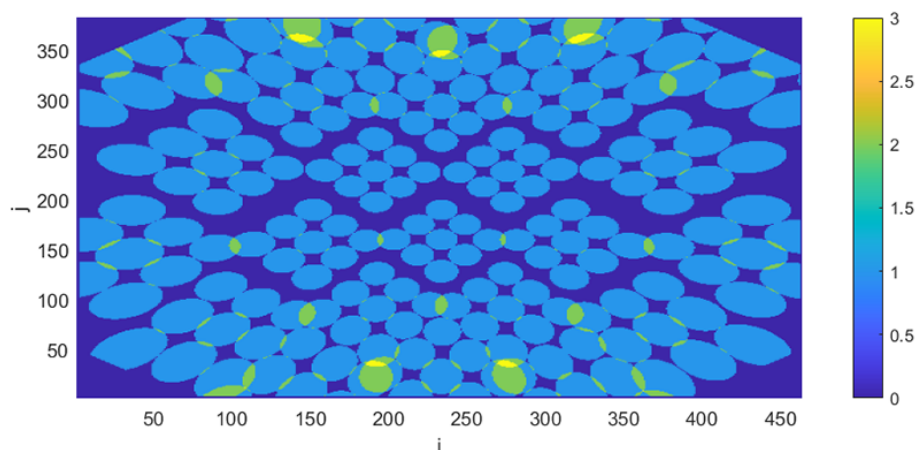


Figure 4.4: The projection of the collimator design using diamond-shaped cluster projections and two centre rows. As a result of the cluster groups, the projections on detector 2 and 3 are identical and thus omitted. The gaps between rows 2, 3 and 4 are clearly visible. It can also be clearly seen that there are pinholes whose projections almost completely overlap in the outermost rows.

4.5. Shifted fitting

As none of the previous approaches resulted in a satisfactory design, a design was made with the 'shifted fitting' approach. The design has 345 pinholes, divided into 4 rows with respectively 4, 3, 4 and 3 clusters. The total degree of multiplexing is 0.0504, and the detector coverage is 0.7454. The projection of this design can be seen in figure 4.5. A 3D model of this collimator is shown in figure 4.7d.

Using the shifted fitting approach, the large unused spaces between the centre rows can be eliminated. However, the larger cluster FOV increases the physical size of the clusters on the collimator. As a result, significantly less clusters can fit on the collimator tube. The pinholes making up the larger clusters now also have a larger opening angle, according to equation (3.5). All previous designs had pinhole opening angles of 7 degrees, compared to VECTor's 17 degrees. The shifted fitting design has pinholes with an opening angle

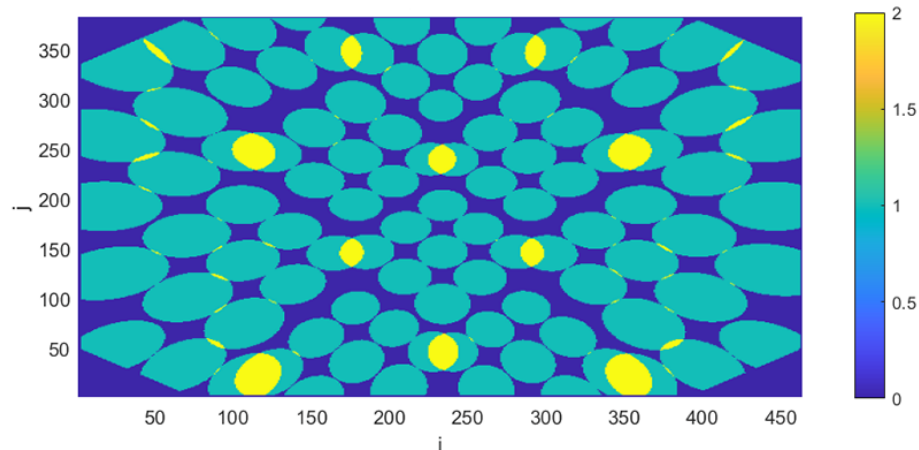


Figure 4.5: The projection of the collimator design with rows with variable FOV's. As a result of the cluster groups, the projections on detector 2 and 3 are identical and thus omitted. At the top and bottom of this projection, the triangle-shaped unused parts of the detector (not in the corners) exist because it is not possible to place additional rows above and below the existing rows to fill in these gaps.

of 10 degrees on average. This will increase the amount of edge penetration, which is exactly what is to be prevented. Additionally, even though the unused detector strips have been eliminated, there is still much unused detector area between the projections. This prevents the detector coverage from being significantly higher than for the previous approaches. And while the degree of multiplexing is low, there are positions where pinhole projections almost completely overlap. As previously explained, this is not desirable.

4.6. Inner radius increase

Previous designs all focused on optimally using the available space on the collimator. However, no optimal design could be reached. Now, first the space each cluster takes up on the inner collimator wall is decreased. This is done by increasing the inner radius of the collimator tube from 24 mm to 25 mm. The value of r_c is kept the same. As a result, each cluster takes up slightly less space on the inner wall of the collimator, allowing the clusters to be placed closer together.

The previously used inner radius of 24 mm is a characteristic dimension of the VECTor collimator. In order to ensure a fair comparison between the VECTor system and the new designs, this value was aimed to remain the same as for the original VECTor collimator. However, as none of the previous designs were able to reach satisfactory values, it is believed that the small increase from 24 mm to 25 mm is justified. As this change is small, it should not lead to significant faults in the comparison.

The small increase in available space allows the stacking approach to achieve a higher detector coverage and degree of multiplexing than before. Additionally, with this approach the θ -shift is no longer necessary. Therefore, the cluster FOV no longer has to be increased, resulting in a larger number of pinholes on the collimator, each with the smaller opening angle of 7 degrees, while still observing the same CFOV as the original VECTor collimator. The exact values for the multiplexing and detector coverage of this design are given in table 4.1. The projection of this design is shown in figure 4.6. A 3D model of this collimator is shown in figure 4.7e.

4.7. Resolution optimisation

The stacking design with larger inner diameter, shown in figure 4.6, is chosen as the optimal design that can be reached with 3x3 pinhole clusters. This design still has pinholes with an infinitely small diameter. The resolution of this design is to be set the same as the VECTor resolution for photons with energies of 511 keV (^{18}F) and 909 keV (^{89}Zr). The VECTor collimator has pinholes with a diameter of 0.700 mm. Two rows on this collimator have opening angles of $\alpha = 18$ degrees, and two rows have opening angles of $\alpha = 16$ degrees. The average of these two ($\alpha_{\text{VECTor}} = 17$ degrees) is used, together with the diameter, in equation (2.6) to cal-

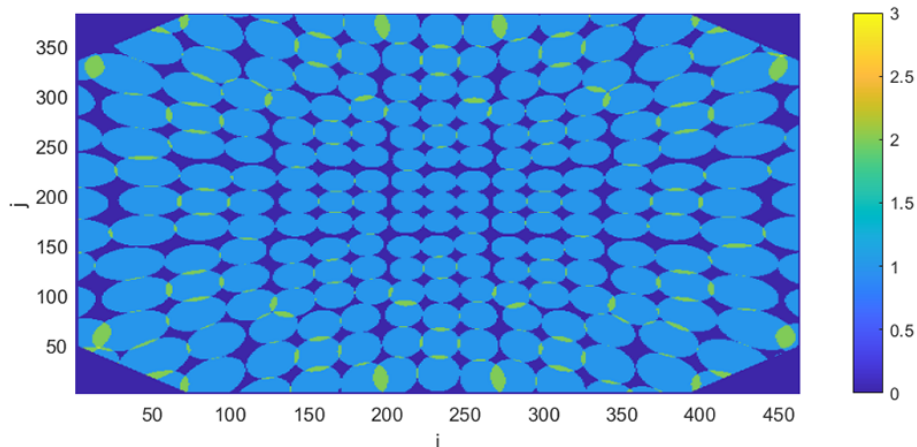


Figure 4.6: The projection of the collimator design using the same technique of section 4.2, but with a larger inner diameter. As a result of the cluster groups, the projections on detector 2 and 3 are identical and thus omitted. This is the design that is chosen as the optimal design found during this project.

culate the necessary diameters for the twisted cluster. The opening angles of the clustered pinholes in the final design are $\alpha_{cp} = 7.08$ degrees. The resulting resolution optimised diameters, together with the energy dependent attenuation coefficients μ , are given in table 4.1. The old pinhole diameters in the collimator

Table 4.1: The pinhole diameter, degree of multiplexing, detector coverage and the number of pinholes for three collimator designs. The first row is the design created following section 4.6. The second and third rows provide the values of the designs derived from this first design, but with different diameters. The diameters, and the attenuation coefficient μ are also given.

Design name	E (keV)	μ (mm^{-1})	r_{in} (mm)	d_{cp} (mm)	Degree of multiplexing	Detector coverage	Number of pinholes
25mm0D	-	-	25	0	0.0434	0.8257	729
25mmF	511	0.2283	25	0.965	0.0369	0.8257	723
25mmZr	909	0.1355	25	1.070	0.0226	0.7886	693
26mmZr	909	0.1355	26	1.148	0.0405	0.8111	717

specification file were substituted with the new values, resulting in two designs. As the pinholes were placed as close together on the collimator as possible, the substitution of larger diameters into this design resulted in intersecting pinholes. This problem could be solved easily for the collimator using the 511 keV pinholes. The only new overlap was between clusters. Moving the clusters slightly further apart prevented any intersections. This gave a slightly lower detector coverage and degree of multiplexing, but the increased pinhole diameter increases the sensitivity of the pinholes, compensating for this change.

Changing the radius to the 909 keV radius resulted not only in overlap between clusters, but also in overlap of pinholes in the same cluster. This second kind of intersection can not be solved by moving the clusters further apart. The self-intersecting clusters were located in the outermost rows (rows 1 and 6). A way to decrease the amount of self-intersecting was by increasing the twisting angle ϕ_r of these rows. A downside to this is that the size of the clusters increases. The clusters now no longer fit as close together as in the original design, decreasing the detector coverage.

With the pinhole diameters increased to accommodate for 909 keV photons, no way was found to place the clusters in such a way that no overlap was present on the collimator. Again, the inner radius was increased with one millimetre, to 26 mm. This allowed a design to be made.

Because increasing the inner radius is not ideal, a third collimator design was made. This design has pinholes with a diameter of 1.070 mm. This was the largest diameter that could still be placed on the collimator without intersection. In order to see if the increase in inner radius and pinhole diameter have any significant

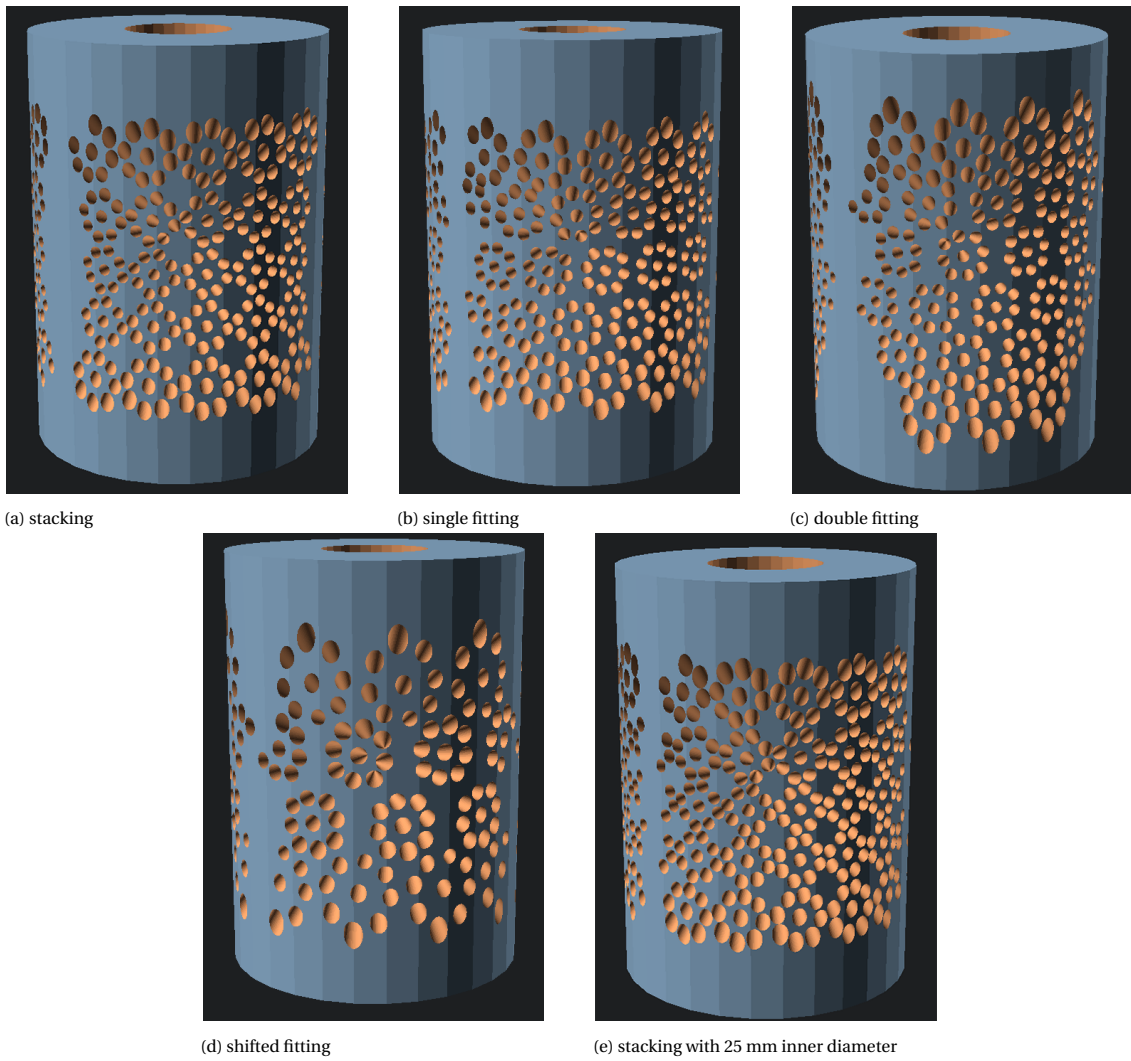


Figure 4.7: 3D models of the different collimator designs, made in OpenSCAD. The outer radius for these models is 67 mm.

Table 4.2: The results of the GATE simulations for the three twisted cluster collimator designs and the VECTor collimator. The results for the reference (VECTor) system are given above the horizontal line, the results for the new designs are given below this line.

Design name	E (keV)	r_{in} (mm)	Total sensitivity (%)	Direct photons (%)	Attenuated photons (%)	Scattered photons (%)
VECTorF	511	24	0.2839	0.0743	0.1908	0.0188
VECTorZr	909	24	0.1516	0.0167	0.1298	0.0051
25mmF	511	25	0.2733	0.1313	0.1230	0.0190
25mmZr	909	25	0.1633	0.0341	0.1264	0.0058
26mmZr	909	26	0.2124	0.0450	0.1604	0.0070

effects on the sensitivity, all three designs are simulated.

4.8. Sensitivity simulations

For the simulations, a source the size of the CFOV of the VECTor collimator is simulated for 1 hour. From this, the total sensitivity of the collimator is given, as well as the components making up this total. These are: direct photons, attenuated photons and scattered photons. The results of the simulations for the three new designs, as well as the values for the VECTor collimator are given in table 4.2.

For the 511 keV photons, the total sensitivity of the 25mmF collimator design is comparable with the sensitivity of the VECTorF collimator under the same conditions (Total VECTor sensitivity: 0.2839%). However, the percentage of direct photons is 1.77 times larger than with the VECTor collimator (VECTor direct photons: 0.0743%). This factor was expected to be 1.9 [20]. The percentage of detected photons that fall under the attenuated photons is lower for the 25mmF design than for the VECTor collimator. As the attenuated photons are the photons that have passed through the collimator material, a lower sensitivity to attenuated photons indicates that the amount of edge penetration is decreased. This is exactly what the 3x3 twisted cluster design had to accomplish. The amount of scattered photons is roughly the same for VECTor and the new design. The sensitivities of both the VECTor system and the new design for 511keV photons are shown figure 4.8a for easy comparison.

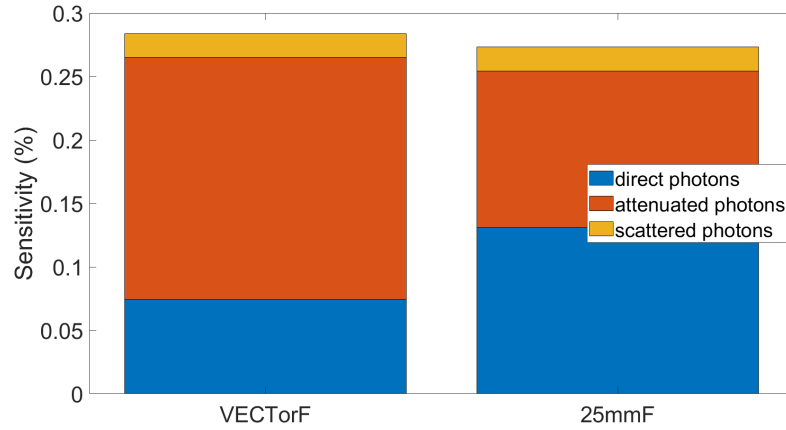
The sensitivity of the VECTor collimator is also simulated for 909 keV photons (VECTorZr). The resulting values are: a total sensitivity of 0.1516%, and a sensitivity to respectively direct, attenuated and scattered photons of 0.0167%, 0.1298% and 0.0051%. This means that the direct photons make up 11.0% of the detected photons. The attenuated photons make up 85.6%, and the scattered photons make up 3.5%.

The 25mmZr collimator has a sensitivity of 0.1633%. This is of the same order of magnitude as the sensitivity of the VECTor collimator. However, the sensitivity to direct photons is 0.0341%, which is 2.04 times the value of VECTor. The sensitivity to attenuated photons is lowered for this new collimator, to 0.1264%. This suggests that the amount of edge penetration is indeed decreased using the 3x3 twisted pinhole clusters. The amount of detected scattered photons has stayed almost the same, with a new value of 0.0058%.

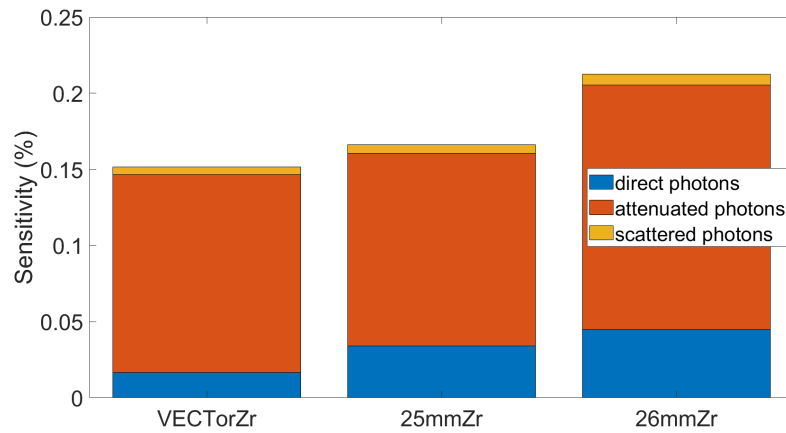
Compared to this, the collimator with pinhole diameters of 1.148 mm and a larger inner radius (26mmZr) has an even higher sensitivity. The total percentage of detected photons for this design is 0.2124%. This is 1.40 times the sensitivity of the VECTor collimator. The sensitivity to direct photons is 0.0450%, 2.69 times the VECTor value. The sensitivity to attenuated photons is also slightly larger than when using VECTor; it is 0.1604%. The sensitivity to scattered photons is also slightly higher, with a value of 0.0070%.

The sensitivities for 909 keV photons for the two new designs and the VECTor collimator are shown in figure 4.8b for easy visual comparison.

Based on these values, both designs optimised for 909 keV have a higher total sensitivity than VECTor. The first design of the two has only a slight increase, but the second design increases the sensitivity with 40%. For the first design, the direct photons make up 20.5% of the detected photons. This is 1.86 times the percentage of VECTor. For the second design, the direct photons make up 21.2%, which is 1.92 times the VECTor value. Even though the sensitivity of the second design to attenuated photons is larger than with the VECTor collimator,



(a) The sensitivity components for the simulations with 511 keV photons.



(b) The sensitivity components for the simulations with 909 keV photons.

Figure 4.8: Bar graphs showing the three components making up the total system sensitivity for the different collimator designs.

mator, the relative sensitivity to these photons is lower.

Both 909 keV designs have a general increase in both total sensitivity and direct photon sensitivity. However, the 26mmZr design (with larger inner radius and larger pinholes) has both the highest total sensitivity and the highest direct photon sensitivity. The relative sensitivity to direct photons is also the largest of the two designs. When using 3x3 twisted clusters, the direct photon sensitivity was expected to increase with a factor 2.8 [20]. The design that comes closest to this value is the second 909 keV design with 2.69. Together with the better values for the total and relative sensitivity, this design seems to be the better of the two.

For both the design for 511 keV and 909 keV, the increase in direct photon sensitivity is lower than the expected value. A possible cause for this is the fact that the degree of multiplexing and the detector coverage are not the same as for the VECTor collimator: both values are lower. As the predictions were made based off of the VECTor collimator, deviating from the VECTor values can produce different results than expected. Increasing the detector coverage of the new designs could improve the total sensitivity of the system. This could also further improve the sensitivity to direct photons, bringing it closer to the expected value. This expected value is calculated with the idea that it should be possible to completely cover the detector. It did not take into account the intersection of pinholes, which prevents an optimal detector coverage. This prevents the collimator from reaching the predicted values.

5

Discussion

This project can serve as a basis for further research. Here, some suggestions for future research are given, as well as suggestions to improve upon the scripts and approaches used here.

Firstly, the script used to calculate the projections of a collimator design could be made significantly more efficient than it currently is. In the script used during this project, every pixel on the detector had to be checked for visibility for every pinhole. However, only a small part of the pixels on a detector is visible for a pinhole, and all these pixels are located close together. Including a heuristic step in the script to find the approximate location of these visible pixels could greatly decrease the amount of pixels that have to be checked for each pinhole. Based on the size of the individual pinhole projections in figure 4.1, every individual projection is fully contained within a 120x120 pixel area. Such an area is only around 10% of the total detector area. Including a heuristic step and only checking these smaller areas could therefore improve the speed of the script by almost a factor of 10. A faster script allows for faster calculation of the multiplexing and detector coverage of designs, improving the overall speed of the design process. As the development of the multiplexing script was not the main focus of this project, it was chosen to not include this heuristic step. Instead the slower version was used, so the project could advance to the collimator design stage.

The script that is used to define the placement of the clusters on the collimator could be greatly improved by the introduction of a graphical user interface (GUI). During this project, defining the placement of the clusters was done by entering numerical values into different arrays in a Matlab script. After entering the required values, the script needed to run, creating an OpenSCAD file. This file then had to be either opened or refreshed in OpenSCAD to see the placement of the pinholes. This made the design process tedious and slow. A GUI could provide immediate feedback on the position of the pinholes without having to run any other programs on the side. Using sliders or input fields, the process of placing the pinholes could also be made more efficient and intuitive. With the right implementation, a GUI could also be used for designing new collimators using other types of pinhole clusters, or collimators for different systems.

As described in the previous chapters, the main approach during this project was to place the clusters in rows on the collimator. All clusters in the same row had the same characteristics (apart from position and orientation). The main advantage of this is that the resulting symmetry makes it relatively easy to manufacture collimators designed this way. However, it gives less control over the characteristics of individual clusters. Because of this reduced control, there may be a more optimal placement of the clusters that has not been found during this project. Additionally, although it makes it easier to define placements, the placement of clusters in rows is in no way proven to be the optimal configuration for the system used. Further research could focus on other, different approaches of placing the clusters to create a more optimal collimator design.

Other follow-up research could focus on the development of collimators using clusters with even more pinholes. Increasing the amount of pinholes per cluster decreases the opening angles of the pinholes and ultimately the effect of edge penetration. However, based on the results of this project, the development of a functional collimator with these higher-order clusters will only work if they are not significantly larger than the 3x3 clusters, as lack of space on the collimator tube was already a significant issue during this project.

6

Conclusion

The aim of this project was to design a collimator to be used in the VECTor system for simultaneous SPECT and PET imaging using high-energy isotopes. This collimator uses 3x3 pinhole clusters, to decrease the opening angles of the individual pinholes and allow tracers with high-energy photons to be imaged. The smaller opening angle should decrease the image degradation effects of pinhole edge penetration. Each design was characterised by its degree of multiplexing, detector coverage and sensitivity. The degree of multiplexing should be as close to the value of the VECTor system as possible. This value is 0.1472. The detector coverage should be as close to 1 as possible. The sensitivity to direct photons should be higher than the sensitivity of the VECTor system.

To find an optimal design, several different approaches were used. First, the clusters were placed in rows, and stacked on top of each other on the collimator tube. In order to reach optimal values, these clusters had to intersect, which was not allowed. This intersection mainly took place on the inner wall of the collimator tube. Secondly, the clusters were rotated 45 degrees around their axes, giving diamond-shaped projections that could be fitted together. Again, this approach either yielded sub-optimal values or intersecting pinholes. Finally, the inner radius of the collimator tube was slightly increased to decrease the size of the clusters on the inner wall of the collimator. This allowed the rows to be placed closer together, and sufficiently cover the detector surface. The optimal result was achieved using the first approach, stacking the rows on top of each other with one row in the centre of the collimator.

The diameters of the pinholes of this design were increased as to achieve the same resolution as the VECTor system. This was done for photons with 511 keV and 909 keV energy. To account for the pinhole intersections that arose as a result of the diameter increase, three final designs were made: one for the 511 keV photons, and two for the 909 keV photons. A sensitivity scan using these designs was simulated with GATE to obtain the sensitivity and the sensitivity components of each design.

For the resolution optimised collimator for 511 keV photons, the total sensitivity was 0.2733%. This is comparable to the sensitivity of VECTor, which is 0.2839%. The sensitivity to direct photons is 0.1313%, which is a factor 1.77 more than with VECTor. The direct photons make up 48% of the detected photons, compared to 26.3% with the VECTor collimator. Overall, the use of 3x3 pinhole clusters therefore has a positive effect on the amount of edge penetration during scanning with 511 keV photons when the resolution of the collimator is made the same as with VECTor.

For the resolution optimised collimator for 909 keV photons, two designs were made. The first design had an inner collimator radius of 25 mm, and pinholes with a diameter of 1.040 mm. This collimator had a total sensitivity of 0.1633%, which is slightly higher than the VECTor value: 0.1516%. The sensitivity to direct photons of this collimator is 0.0341%, which is 20.5% of the total sensitivity. For VECTor, at this energy the direct photons make up 11.0% of its total sensitivity. The absolute sensitivity to direct photons is 2.04 times the VECTor value.

The second design had an inner collimator radius of 26 mm and pinholes with a diameter of 1.148 mm. The total sensitivity of this design was 0.2124%, which is 1.40 times the VECTor sensitivity. The direct photon sensitivity of this collimator was 0.0450%, which is 21.2% of the total sensitivity. The absolute direct photon sensitivity was 2.69 times the VECTor value.

For both designs, the total sensitivity, as well as the relative sensitivity to direct photons increased compared to the VECTor collimator in the same circumstances. It is concluded that the use of 3x3 pinhole clusters has significant benefits over VECTor when used at high energies. The collimator designs made during this project effectively implement these clusters to decrease the effects of pinhole edge penetration.

Although the sensitivities for all designs were higher for the direct photons than the VECTor values, they were not as high as predicted. For 511 keV photons, an increase of 1.9 was predicted, yet 1.77 was found. For 909 keV photons, an increase of 2.8 was predicted, yet at most 2.69 was found. A possible explanation for this is the fact that the multiplexing and detector coverage of the new designs is lower than for the VECTor collimator. As the predictions were made based on the VECTor characteristics, lower values for the detector coverage and multiplexing can result in lower sensitivity values than expected.

Follow-up research could focus on other approaches to placing the clusters, in hopes of increasing the detector coverage of the design. Improvements could be made to this research by increasing the efficiency of the scripts used to test the characteristics of the collimator, so that more approaches and designs can be tested in the same amount of time.

A

Matlab code

This appendix contains the Matlab scripts used during this project. It explains the uses of the different scripts and the connections between them. For some scripts, multiple versions exist. Here, only the final or most versatile version is included.

A.1. Projection

The script `projection.m` calculates the projection of a collimator design, specified in a collimator specification file given by `clusterFile`, on the detectors. The detectors are specified with the script `det_make.m`. The script `LoadVpx` is a script provided to load an array with bad pixels into variables. The script `projection.m` is shown below.

```
% File: projection.m

% Make sure to run readInCollimatorSpecifications.m and det_make.m first
% for all the variables
close all, clc;

readfile = clusterFile;
run readInCollimatorSpecifications;
run det_make;

%% Let's make some pixels
clc;

% The size of a pixel on the detector
pixelsize = 1.072000;

% i is in the xy-plane, j is parallel to the z-axis
i_size = 464;
j_size = 383;

% vert is easier to type than det_ideal.vertices
vert = det_ideal.vertices;

% A vector with the same length and direction as the top edge of each
% detector
% Actually three vectors in one
span = vert([4 8 12], [1 2]) - vert([1 5 9], [1 2]);

% This is the size of a pixel, projected on either the x or y axis
% Dividing the x and y component of the detector by the amount of pixels in
```

```

% the i-direction
pixel_size_x = span(:, 1)/i_size;
pixel_size_y = span(:, 2)/i_size;

% Linspace does not work with vectors
% This means I have to make each of the pixel vectors separately
% It's not pretty, but it works
pixel_x1 = linspace(vert(1, 1), vert(4, 1)- pixel_size_x(1), i_size);
pixel_x2 = linspace(vert(5, 1), vert(8, 1)- pixel_size_x(2), i_size);
pixel_x3 = linspace(vert(9, 1), vert(12, 1)- pixel_size_x(3), i_size);

pixel_y1 = linspace(vert(1, 2), vert(4, 2)- pixel_size_y(1), i_size);
pixel_y2 = linspace(vert(5, 2), vert(8, 2)- pixel_size_y(2), i_size);
pixel_y3 = linspace(vert(9, 2), vert(12, 2)- pixel_size_y(3), i_size);

% Combining all the x and y coordinates of all the pixels into one matrix
% Indices are: coordinate (x or y), pixel number (i), detector (1:3)
pixel_xy = cat(3, [pixel_x1; pixel_y1], [pixel_x2; pixel_y2], [pixel_x3; pixel_y3]);

% Clearing all the intermediate variables to keep the workspace clean
clear pixel_x1 pixel_x2 pixel_x3 pixel_y1 pixel_y2 pixel_y3 span;

% Now for the z-coordinates of the pixels:
% As all the j-coordinates lie on the z-axis, the real pixel_size can be
% used, and no projection and division etc is needed here
pixel_z1 = linspace(vert(1, 3), vert(2, 3) - pixel_size, j_size) - pixel_size/2;
pixel_z2 = linspace(vert(5, 3), vert(6, 3) - pixel_size, j_size) - pixel_size/2;
pixel_z3 = linspace(vert(9, 3), vert(10, 3) - pixel_size, j_size) - pixel_size/2;

% Combining all the z coordinates of all the pixels into one matrix
% indices are: detector (1:3), pixel number (j)
pixel_z = cat(1, pixel_z1, pixel_z2, pixel_z3);

% Clearing all the intermediate variables to keep the workspace clean
clear pixel_z1 pixel_z2 pixel_z3;

% Finding the number of holes on the collimator
[hole_number, ~] = size(c);

% creating an empty matrix for the three detectors. This is where the
% intensity will be stored
det_I = zeros(j_size, i_size, 3);

% Loop over all the holes
for h = 1:hole_number

    % Check which detector hole h is projecting on
    detector = ph(h, 2) + 1;

    % Loop over all i and j coordinates
    for i = 1:i_size
        for j = 1:j_size
            % Retrieve the xyz-coordinates of pixel [i, j], and the
            % coordinates of the centre of hole h
            pixel = [pixel_xy(1, i, detector) pixel_xy(2, i, detector) pixel_z(
                detector, j)];
        end
    end
end

```



```

    pinhole = center(h, :);

    % Create the projection vector from h to [i, j], and calculate
    % its magnitude
    proj = pixel - pinhole;
    projection_mag = sqrt(proj(1)^2 + proj(2)^2 + proj(3)^2);

    % Create the orientation vector of hole h, and calculate its
    % magnitude
    theta = phiTheta(h, 2);
    phi = phiTheta(h, 1);
    orientation = [sin(theta)*cos(phi) sin(theta)*sin(phi) cos(theta)];
    orientation_mag = sqrt(orientation(1)^2 + orientation(2)^2 + orientation
        (3)^2);

    % Calculate the angle between the projection and orientation
    % vector (in degrees)
    cos_angle = dot(proj, orientation)/(projection_mag*orientation_mag);
    angle = acosd(cos_angle);

    % Set the threshold angle to half the opening angle of pinhole
    % h (also in degrees)
    thresh_angle = alpha(h, :)/2;

    % Check if the projection angle exceeds the threshold angle. If
    % it does not, one intensity is added to the pixel on the
    % detector the pinhole is projecting on
    if (angle < thresh_angle)
        det_I(j, i, detector) = det_I(j, i, detector) + 1;
    end
end
end
% Display a progress bar. Each # is a percent. This is just because the code
% takes quite a long time to run
clc;
progress = floor(h/hole_number * 100);
fprintf('Progress: \nLast pinhole completed: %d \n[', h);
for q = 1:progress
    fprintf('#');
end
for q = 1:100-progress
    fprintf('-');
end
fprintf(']\n');
pause(0.01)
end

% Load the data that determines which pixels contribute and which don't.
% Then turn in into the same format as the det_I variable, so it is easier
% to use.
% 1 means live pixel, 0 means 'dead' pixel
data = LoadVpx('Correction');
live_pixels = cat(3, data(1:383, :), data(384:766, :), data(767:1149, :));

% Multiplying the intensity matrix with the live_pixels matrix. This way,

```

```

% every dead pixel will have value 0, independent of the projection. It
% will therefore also not count in the multiplexing calculations, as this
% only looks for pixels with a value of 1 or higher.
det_I = det_I.*live_pixels;

disp("Done");

%% Multiplexing
used_pixels = 0;
multi_pixels = 0;

% Loop over the three detectors
for d = 1:3
    % Loop over all pixels per detector
    for i = 1:i_size
        for j = 1:j_size
            % If the pixel has a value of 1 or higher, it is projected on,
            % and it will be counted in used_pixels
            if (det_I(j, i, d) >= 1)
                used_pixels = used_pixels + 1;
            end

            % If the pixel value is 2 or higher, it is part of an
            % overlapping region, and it is counted in multi_pixels
            if (det_I(j, i, d) >= 2)
                multi_pixels = multi_pixels + 1;
            end
        end
    end
    fprintf('Calculating multiplexing for detector: %d \n', d);
end

% Calculate and display the multiplexing value
multiplexing = multi_pixels/used_pixels;

clc;
fprintf('The total multiplexing is: %10.8f \n', multiplexing);

pixel_fraction = used_pixels/(3*177712);
fprintf('The total fraction of pixels used is: %10.8f \n', pixel_fraction);

pixel_fraction = used_pixels/sum(sum(data));
fprintf('The EFFECTIVE total fraction of pixels used is: %10.8f \n', pixel_fraction)
;

%% plot the final results
% Plot the intensities 'measured' on the detector in three separate plots,
% but in one image.
f1 = subplot(2, 2, 1);
det1 = pcolor(det_I(:, :, 1));
xlabel("i");
ylabel("j");
title("Detector 1");
set(det1, 'LineStyle', 'none');

f2 = subplot(2, 2, 2);

```

```

det2 = pcolor(det_I(:, :, 2));
xlabel("i");
ylabel("j");
title("Detector 2");
set(det2, 'LineStyle', 'none');

f3 = subplot(2, 2, 3);
det3 = pcolor(det_I(:, :, 3));
xlabel("i");
ylabel("j");
title("Detector 3");
set(det3, 'LineStyle', 'none');

```

A.2. Collimator creation

The script `makeCollimator.m` is the script which contains all the variables that specify the placement of the clusters on the collimator tube. All variables are given only for one cluster group. The variable `newColl` gives the name of the file containing the positions and orientations of the clusters. The variable `clusterFile` gives the name of the final collimator specification file, where the clusters have been placed based on `newColl`. `makeCollimator.m` is shown below.

```

% File: makeCollimator.m

clear, clc;

% The name of the file that will specify where all the clusters will be
% placed and how they will be orientated
newColl = 'newCollExample.txt';
specFile = newColl;

% Specify the parameters for creating the collimator
rows = 5; % Integer value
fillFactor = [5 7 7 7 5]; % Must have the same # of elements as the value of rows
z_shift = [-23 -10.5 0 10.5 23]; % Must have the same # of elements as the value of
rows
phi_shift = [0.23 0.37 0.29 0.37 0.23]; % Radians, positive is wider
theta_shift = [0 0 0 0 0]; % Degrees, positive is up
z_factor = [0 0 0 0 0]; % Negative is away from equator
flipOrNot = [0 0 0 0 0]; % Value 1: flip, value 0: don't flip
rotate = [2 0 3 0 2]; %Degrees, positive is clockwise
phi_rot = [6.024 6.024 5.924 6.024 6.024]; % Standard is 6.124
r_fov = [6 6 6 6 6]; % Radius of the CFOV per row
phi_factor = 1;
phi_comp = 1.15;

% The name of the file that will contain the positions and orientations of
% all the individual pinholes
clusterFile = 'clusterFileExample.txt';

% Create the cluster spec file
run pinholePlacement_semiAuto;

% Place the clusters at the position specified in the cluster spec file
run clusterCollimator;

% Stop pinholes from overlapping at group edges
run stopEdgeOverlap;

```

```

% Make an OpenSCad file for easy 3D viewing
run GetOpenscadFileFromTextFile;
disp ("Done!");

%% Calculate and show the projection of the recently created collimator
run projection;
% Sound a notification beep when the calculations are done
beep;

```

In this file, after the cluster group parameters are defined, the file `pinholePlacement_semiAuto.m` is run. This script defines the position and orientation of each cluster to be placed on the collimator. The positions and orientations are saved in a file with the same name as `newColl`. The script is shown below.

```

% File: pinholePlacement_semiAuto.m

% clear, clc;

filename = newColl;
% filename = 'test.txt';

% System characteristics
% r_fov = 8; % central field of view radius
r_c = 32; % distance of pinhole centres to collimator axis
x_det = 210; % Distance of detector to collimator axis (shortest)
z_max = 205.287; % Max coordinate of the detector in the z-direction
w_max = 258.94; % Max (half) width of the detector

fillMatrix = zeros(rows, 2);
fillMatrix(:, 2) = fillFactor;

% Apply z-shift
for r = 1:rows
    z_c(r) = z_shift(r);
end

% Add z_c to the first column of the fillMatrix
fillMatrix(:, 1) = z_c;

counter = 1;
for r = 1:rows
    clear phi; syms phi;

    z_c = fillMatrix(r, 1);

    alpha = 2*asin(r_fov(r)/r_c); % Opening angle (radians)

    % Equation to estimate the maximum allowed phi for a certain z_c and
    % detector size
    eq_y = r_c.*sin(phi) + ((x_det./cos(phi) - r_c)./cos(atan(z_c./r_c))) .* tan(phi
        + alpha./2) == w_max;

    phi_max = double(vpasolve(eq_y, phi)) * phi_comp;

    phi_max = phi_max - (abs(z_c/z_max) * phi_factor);

    phi_max = phi_max + phi_shift(r);

```

```

clear phi;
if fillMatrix(r, 2) == 1
    phi = 0;
else
    phi = linspace(-phi_max, phi_max, fillMatrix(r, 2));
end

for i = 1:fillMatrix(r, 2)
    z_c = fillMatrix(r, 1);
    z_c = z_c - abs(phi(i)/phi_max) * sign(z_c) * z_factor(r);

    theta = pi/2 - atan(z_c/r_c);
    x_c = r_c * cos(phi(i));
    y_c = r_c * sin(phi(i));

    theta = theta - (theta_shift(r)*pi/180);

    writeMatrix(counter, :) = [counter-1 0 -x_c y_c -z_c -phi(i) theta alpha
        0.7 r];
    counter = counter + 1;
end
end

writeMatrix = transpose(writeMatrix);

% Rotate all the pinholes into position
writeMatrix([3 4 5], :) = rotationMatrix('z', -7*pi/6) * writeMatrix([3 4 5], :);
writeMatrix(6, :) = writeMatrix(6, :) - 7*pi/6;

ph_num = counter - 1;

% Duplicate the pinhole plane twice
writeMatrix(:, ph_num+1:2*ph_num) = writeMatrix(:, 1:ph_num);
writeMatrix(:, 2*ph_num+1:3*ph_num) = writeMatrix(:, 1:ph_num);

% Rotate the second plane to the second detector
writeMatrix([3 4 5], ph_num+1:2*ph_num) = rotationMatrix('z', -2*pi/3) * writeMatrix
([3 4 5], ph_num+1:2*ph_num);
writeMatrix(6, ph_num+1:2*ph_num) = writeMatrix(6, ph_num+1:2*ph_num) - 2*pi/3;
writeMatrix(2, ph_num+1:2*ph_num) = 1;

writeMatrix(1, ph_num+1:2*ph_num) = ph_num:2*ph_num-1;

% Rotate the third plane to the third detector
writeMatrix([3 4 5], 2*ph_num+1:3*ph_num) = rotationMatrix('z', -4*pi/3) *
writeMatrix([3 4 5], 2*ph_num+1:3*ph_num);
writeMatrix(6, 2*ph_num+1:3*ph_num) = writeMatrix(6, 2*ph_num+1:3*ph_num) - 4*pi/3;
writeMatrix(2, 2*ph_num+1:3*ph_num) = 2;

writeMatrix(1, 2*ph_num+1:3*ph_num) = 2*ph_num:3*ph_num-1;

file = fopen(filename, 'w');

fprintf(file, '%d %d %10.8f %10.8f %10.8f %10.8f %10.8f %10.8f %4.3f %
d\n', writeMatrix);

```

```
fclose ( file );

fprintf ( 'file saved as %s \nThis file contains the locations of the clusters.\n',
         filename );
```

After this script, the script `clusterCollimator.m` is run. This script reads in the cluster parameters saved in the `newColl`-file, and places clusters on the collimator depending on these parameters. The parameters of the individual pinholes are then saved in a `.txt`-file whose name is specified by `clusterFile`. `clusterCollimator.m` is shown below.

```
% File: clusterCollimator.m

close , clc;

% 'center' contains all the xyz coordinates of all the pinhole/cluster
% positions.
% 'phiTheta' contains all the orientations of all the pinholes/clusters
% 'r_colli' is the radius of the collimator, measured from centre to
% pinhole/cluster centres
readfile = specFile;
run readInCollimatorSpecifications;
clear c alpha j d p r_in r_out;

% Find the amount of selected positions
[cluster_number , ~] = size(center);

% The pinhole cluster twisting angle
phi_r = phi_rot * pi/180;

% Loop over all selected positions
for i = 1:cluster_number
    row = ph(i, 10);
    [pos_out, ori_out] = createCluster(phi_r(row), r_fov(row));

    % Check if flipOrNot exists. If it does, execute:
    if exist('flipOrNot', 'var') == 1
        if flipOrNot(row) == 1
            % Bring cluster to the origin
            pos_out(3, :) = pos_out(3, :) + 32;

            % flip around the x-axis
            pos_out(2, :) = -pos_out(2, :);
            ori_out(2, :) = -ori_out(2, :);

            % Shift back to original position
            pos_out(3, :) = pos_out(3, :) - 32;
        end
    end

    % Check if the clusters have to be rotated
    if exist('rotate', 'var') == 1
        rot_angle = rotate(row)*pi/180;
        if flipOrNot(row) == 1
            % Rotate the cluster by 45 degrees
            pos_out = rotationMatrix('z', -rot_angle) * pos_out;
            ori_out = rotationMatrix('z', -rot_angle) * ori_out;
```

```

    else
        % Rotate the cluster by 45 degrees
        pos_out = rotationMatrix('z', rot_angle) * pos_out;
        ori_out = rotationMatrix('z', rot_angle) * ori_out;
    end
end

% Invert orientation vector to look outwards
ori_out([1 2 3], :) = -ori_out([1 2 3], :);

% Calculate theta orientation
theta = atan2(center(i, 2), center(i, 1));

% Compensate if coordinates are not exactly 32 from collimator axis
R = norm(center(i, [1 2]));
pos_out(3, :) = pos_out(3, :) - (R - 32);

% rotate around the y-axis into the xy-plane
pos_out = rotationMatrix('y', -pi/2) * pos_out;
ori_out = rotationMatrix('y', -pi+phiTheta(i, 2)) * ori_out;

% rotate around the z-axis to right theta position
pos_out = rotationMatrix('z', theta) * pos_out;
ori_out = rotationMatrix('z', phiTheta(i, 1)) * ori_out;

% shift to correct z-position
pos_out(3, :) = pos_out(3, :) + center(i, 3);

pos_out = transpose(pos_out);
ori_out = transpose(ori_out);

c(i*9-8:i*9, :) = pos_out;
ori(i*9-8:i*9, :) = ori_out;

det(i*9-8:i*9) = ph(i, 2);

opening_angle(i*9-8:i*9) = ph(i, 8)/3;

D(i*9-8:i*9) = 1.148;
end

% Calculate the theta and phi angles of the global orientation of every
% pinhole
for i = 1:cluster_number*9
    ori_angles(i, [1 2]) = [atan2(ori(i, 2), ori(i, 1)) ...
        acos(ori(i, 3)/sqrt(ori(i, 1)^2 + ori(i, 2)^2 + ori(i, 3)^2))];
end

% clear write_matrix if it still exists from previous script
clear write_matrix;
for i = 1:cluster_number*9
    write_matrix(i, :) = [i-1 det(i) c(i, [1 2 3]) ori_angles(i, [1 2])
        opening_angle(i) D(i)];
end

% This has to be inverted to work with other programs

```

```

% It is not important during these calculations and is inverted back when needed
write_matrix(:, [3 5]) = -write_matrix(:, [3 5]);
write_matrix(:, [6]) = -write_matrix(:, [6]);
write_matrix = transpose(write_matrix);

fileID = fopen(clusterFile, 'w');

fprintf(fileID, '%d %d %10.8f %10.8f %10.8f %10.8f %10.8f %10.8f %4.3f\n', write_matrix);

fclose(fileID);

```

clusterCollimator.m relies upon the function createCluster(phi_r, r_fov). This function creates a twisted pinhole cluster at x, y, z -position $(0, 0, r_c)$ with orientation facing down. This function is a modified version of the script that was provided that created a pinhole cluster at the same position, but with inverse orientation. The code of createCluster is shown below.

```

% File: createCluster.m

function [ph_pos, ph_ori] = createCluster(phi_r, r_fov)
% System information
% r_fov = 6; % fov radius
r_c = 32; % cluster centre radius
r_det = 210; %distance from centre to detector
r_col_in = 24; %inner radius of collimator cylinder

opening_angle = 2*asin(r_fov/r_c); % opening angle of the entire cluster
beta = 1/3 * opening_angle;

% Pinhole counter, starting at 0
ph_n = 0;

% Create the first five pinholes
for x = -1:1:1
    for y = -1:1:1
        if abs(x*y) == 0 % in the [x y]-matrix, ignore the corners
            ph_n = ph_n + 1; % create the next pinhole

            ph_pos(:, ph_n) = [0 0 -r_c]; % pinhole position
            ph_ori(:, ph_n) = [-x*tan(beta) -y*tan(beta) 1]; % pinhole orientation

            % Normalise the pinhole orientation length
            ph_ori(:, ph_n) = ph_ori(:, ph_n) / norm(ph_ori(:, ph_n));

            if ph_n ~= 3 % in the [x y]-matrix, ignore the centre
                % Rotate the pinhole position around the x and y axis, depending on
                % their position in the [x y]-matrix
                ph_pos(:, ph_n) = rotationMatrix('x', x*phi_r) * ph_pos(:, ph_n);
                ph_pos(:, ph_n) = rotationMatrix('y', y*phi_r) * ph_pos(:, ph_n);

                % Rotate the pinhole orientation around the x and y axis, depending
                % on
                % their position in the [x y]-matrix
                ph_ori(:, ph_n) = rotationMatrix('x', x*phi_r) * ph_ori(:, ph_n);
                ph_ori(:, ph_n) = rotationMatrix('y', y*phi_r) * ph_ori(:, ph_n);
            end
        end
    end
end
end

```



```

    end
end

% Now create 4 new holes by twisting the 4 outer holes 1/4pi around the
% z-axis
for i = 1:5
    if i ~= 3 % Don't use pinhole 3
        % Increment the pinhole counter again
        ph_n = ph_n + 1;

        % Create the new position and orientation of pinhole 6, 7, 8 and 9
        ph_pos(:, ph_n) = rotationMatrix('z', -pi/4) * ph_pos(:, i);
        ph_ori(:, ph_n) = rotationMatrix('z', -pi/4) * ph_ori(:, i);
    end
end

% Get the signs of the x and y position of the pinholes
ph_sign_x = sign(ph_pos(1, :));
ph_sign_y = sign(ph_pos(2, :));

% Loop over all the 'new' pinholes
for i = 6:9
    % For equal signs, the sqrt(2) factor works on the y angle
    if ph_sign_x(i)*ph_sign_y(i) == 1
        % Twist position
        ph_pos(:, i) = rotationMatrix('x', ph_sign_x(i)*phi_r) * ...
            rotationMatrix('y', ph_sign_y(i)*phi_r*sqrt(2)) * ph_pos(:, i);

        % Twist orientation
        ph_ori(:, i) = rotationMatrix('x', ph_sign_x(i)*phi_r) * ...
            rotationMatrix('y', ph_sign_y(i)*phi_r*sqrt(2)) * ph_ori(:, i);

    % For equal signs, the sqrt(2) factor works on the x angle
    elseif ph_sign_x(i)*ph_sign_y(i) == -1
        % Twist position
        ph_pos(:, i) = rotationMatrix('x', ph_sign_x(i)*phi_r*sqrt(2)) * ...
            rotationMatrix('y', ph_sign_y(i)*phi_r) * ph_pos(:, i);

        % Twist orientation
        ph_ori(:, i) = rotationMatrix('x', ph_sign_x(i)*phi_r*sqrt(2)) * ...
            rotationMatrix('y', ph_sign_y(i)*phi_r) * ph_ori(:, i);
    end
end
end
end

```

Finally, `stopEdgeOverlap` is run. This script makes sure there is a gap between the cluster groups to prevent the groups from intersecting. The width of this gap is defined by `space` in degrees. This script reads in the collimator specification file, removes all pinholes in the gaps between the groups, and replaces the original specification file with the one without these pinholes. The script is shown below.

```
% File: stopEdgeOverlap.m
```

```
close all, clc;
```

```
readfile = clusterFile;
```

```

run readInCollimatorSpecifications

phi = phiTheta(:, 1);

splits = [1/6*pi, 5/6*pi, -2*pi + 9/6*pi];
space = 7 * pi/180;

upperBounds = splits + space;
lowerBounds = splits - space;

ph_num = size(phi, 1);
counter = 1;
toDelete = [];
for i = 1:ph_num
    delete = false;
    if (phi(i) > lowerBounds(1)) && (phi(i) < upperBounds(1))
        % disp([lowerBounds(1) phi(i) upperBounds(1)]);
        delete = true;
    end
    if (phi(i) > lowerBounds(2)) && (phi(i) < upperBounds(2))
        % disp([lowerBounds(2) phi(i) upperBounds(2)]);
        delete = true;
    end
    if (phi(i) > lowerBounds(3)) && (phi(i) < upperBounds(3))
        % disp([lowerBounds(3) phi(i) upperBounds(3)]);
        delete = true;
    end

    if delete == true
        toDelete(counter) = i;
        counter = counter + 1;
    end
end

ph(toDelete, :) = [];

%%
file = fopen(readfile, 'w');

write_matrix = transpose(ph);
fprintf(file, '%d %d %10.8f %10.8f %10.8f %10.8f %10.8f %10.8f %4.3f\n', write_matrix);

fclose(file);

```

The scripts and functions `pinholePlacement_semiAuto`, `clusterCollimator` and `createCluster` rely heavily on a function called `rotationMatrix(axis, angle)`. This is a function which returns the 3D rotation matrix for rotation of `angle` radians around axis `axis`. This function is shown below.

% File: `rotationMatrix.m`

```

function matrix = rotationMatrix(axis, angle)
% ROTATIONMATRIX is a simple function that returns the rotation matrix for
% rotating 'angle' degrees around axis 'axis'. This prevents the user from having
% to define these matrices mid-script and keeps the code clean.

% Check for x-rotation matrix

```

```
if (axis == 1 || axis == 'x')
    matrix = [1 0 0; 0 cos(angle) -sin(angle); 0 sin(angle) cos(angle)];

% Check for y-rotation matrix
elseif (axis == 2 || axis == 'y')
    matrix = [cos(angle) 0 sin(angle); 0 1 0; -sin(angle) 0 cos(angle)];

% Check for z-rotation matrix
elseif (axis == 3 || axis == 'z')
    matrix = [cos(angle) -sin(angle) 0; sin(angle) cos(angle) 0; 0 0 1];

end

end
```

The scripts `readInCollimatorSpecifications.m` and `GetOpenscadFileFromTextFile.m` in the scripts above are scripts used to respectively load the information from the collimator specification files into variables in the code, and to create an OpenSCAD-file from the collimator specification file. These scripts were provided from the beginning and were therefore not written during this project. They will therefore not be shown here.

Bibliography

- [1] Frans Van Der Have, Brendan Vastenhouw, Ruud M Ramakers, Woutjan Branderhorst, Jens O Krah, Changguo Ji, Steven G Staelens, and Freek J Beekman. U-SPECT-II: an ultra-high-resolution device for molecular small-animal imaging. *Journal of Nuclear Medicine*, 50(4):599–605, 2009.
- [2] Freek Beekman and Frans van der Have. The pinhole: gateway to ultra-high-resolution three-dimensional radionuclide imaging. 2007.
- [3] Oleksandra Ivashchenko, Frans van der Have, Jose L Villena, Harald C Groen, Ruud M Ramakers, Harrie H Weinans, and Freek J Beekman. Quarter-millimeter-resolution molecular mouse imaging with U-SPECT+. *Molecular imaging*, 14(1):7290–2014, 2015.
- [4] Frans Van Der Have, Brendan Vastenhouw, Ruud M Ramakers, Woutjan Branderhorst, Jens O Krah, Changguo Ji, Steven G Staelens, and Freek J Beekman. U-spect-ii: an ultra-high-resolution device for molecular small-animal imaging. *Journal of Nuclear Medicine*, 50(4):599–605, 2009.
- [5] Gaspar Delso, Edwin Ter Voert, and Patrick Veit-Haibach. How does pet/mr work? basic physics for physicians. *Abdominal imaging*, 40(6):1352–1357, 2015.
- [6] Marlies C Goorden, Frans van der Have, Rob Kreuger, Ruud M Ramakers, Brendan Vastenhouw, J Peter H Burbach, Jan Booiij, Carla FM Molthoff, and Freek J Beekman. VECTor: a preclinical imaging system for simultaneous submillimeter SPECT and PET. *Journal of Nuclear Medicine*, 54(2):306–312, 2013.
- [7] Marlies C Goorden, Jarno van Roosmalen, Frans van der Have, and Freek J Beekman. Optimizing modelling in iterative image reconstruction for preclinical pinhole PET. *Physics in Medicine & Biology*, 61(10):3712, 2016.
- [8] Shelan T Mahmood, Kjell Erlandsson, Ian Cullum, and Brian Forbes Hutton. Design of a novel slit-slat collimator system for SPECT imaging of the human brain. *Physics in Medicine & Biology*, 54(11):3433, 2009.
- [9] Gengsheng L Zeng. A skew-slit collimator for small-animal SPECT. *Journal of nuclear medicine technology*, 36(4):207–212, 2008.
- [10] D Paix. Pinhole imaging of gamma rays. *Physics in Medicine & Biology*, 12(4):489, 1967.
- [11] Freek J Beekman, Frans van der Have, Brendan Vastenhouw, Annemarie JA van der Linden, Peter P van Rijk, J Peter H Burbach, and Marten P Smidt. U-SPECT-I: a novel system for submillimeter-resolution tomography with radiolabeled molecules in mice. *Journal of Nuclear Medicine*, 46(7):1194–1200, 2005.
- [12] Brian F Hutton, H Malcolm Hudson, and Freek J Beekman. A clinical perspective of accelerated statistical reconstruction. *European journal of nuclear medicine*, 24(7):797–808, 1997.
- [13] MC Goorden, F Van Der Have, R Kreuger, and FJ Beekman. An efficient simulator for pinhole imaging of PET isotopes. *Physics in Medicine & Biology*, 56(6):1617, 2011.
- [14] Woutjan Branderhorst, Brendan Vastenhouw, and Freek J Beekman. Pixel-based subsets for rapid multi-pinhole SPECT reconstruction, 2010.
- [15] Marlies C Goorden, Chris Kamphuis, Ruud M Ramakers, and Freek J Beekman. Accelerated image reconstruction by a combined dual-matrix dual-voxel approach. *Physics in Medicine & Biology*, 2020.
- [16] Marlies C Goorden and Freek J Beekman. High-resolution tomography of positron emitters with clustered pinhole SPECT. *Physics in Medicine & Biology*, 55(5):1265, 2010.

- [17] Kenta Miwa, Masayuki Inubushi, Yasuto Takeuchi, Tetsuro Katafuchi, Mitsuru Koizumi, Tsuneo Saga, and Masayuki Sasaki. Performance characteristics of a novel clustered multi-pinhole technology for simultaneous high-resolution SPECT/PET. *Annals of nuclear medicine*, 29(5):460–466, 2015.
- [18] Floor CJ Van de Watering, Mark Rijpkema, Lars Perk, Ulrich Brinkmann, Wim JG Oyen, and Otto C Boerman. Zirconium-89 labeled antibodies: a new tool for molecular imaging in cancer patients. *BioMed research international*, 2014, 2014.
- [19] Martin Walther, Peter Gebhardt, Philipp Grosse-Gehling, Lydia Würbach, Ingo Irmeler, Stephan Preusche, Muhammad Khalid, Thomas Opfermann, Thomas Kamradt, Jörg Steinbach, et al. Implementation of ^{89}Zr production and in vivo imaging of B-cells in mice with ^{89}Zr -labeled anti-B-cell antibodies by small animal PET/CT. *Applied radiation and isotopes*, 69(6):852–857, 2011.
- [20] "QUARAT: Quantitative Universal Radiotracer Tomography", STW-OTP proposal granted in 2018.
- [21] Greta SP Mok, Yuchuan Wang, and Benjamin MW Tsui. Quantification of the multiplexing effects in multi-pinhole small animal SPECT: a simulation study. *IEEE transactions on nuclear science*, 56(5):2636–2643, 2009.
- [22] Victor Bom, Marlies Goorden, and Freek Beekman. Comparison of pinhole collimator materials based on sensitivity equivalence. *Physics in Medicine & Biology*, 56(11):3199, 2011.
- [23] Roberto Accorsi and Scott D Metzler. Analytic determination of the resolution-equivalent effective diameter of a pinhole collimator. *IEEE transactions on medical imaging*, 23(6):750–763, 2004.
- [24] Matlab, 2020. Version R2020a (9.8.0.1323502). The MathWorks Inc.
- [25] OpenSCAD, 2020. Version 2019.05. The OpenSCAD Developers.
- [26] D Strulab, G Santin, D Lazaro, V Breton, and C Morel. GATE (Geant4 Application for Tomographic Emission): a PET/SPECT general-purpose simulation platform, 2003.
- [27] G Santin, D Strul, Delphine Lazaro, L Simon, M Krieguer, M Vieira Martins, Vincent Breton, and C Morel. GATE: A Geant4-based simulation platform for PET and SPECT integrating movement and time management. *IEEE Transactions on nuclear science*, 50(5):1516–1521, 2003.