Mapping Gamma Sources and Their Flux Fields Using Non-directional Flux Measurements

Jack Buffington

CMU-RI-TR-18-32

August 14, 2018

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Red Whittaker
Michael Kaess
Joseph Bartels

Submitted in partial fulfillment of the requirements for the degree of Masters of Science in Robotics.

Abstract

There is a compelling need to robotically determine the location and activity of radiation sources from their flux. There is also a need to create dense flux maps from sparse flux measurements. This research addresses these dual problems.

An example use would be at the location of a nuclear accident. A mobile robot could collect gamma flux measurements. Using these measurements, dense flux maps and likely locations for fissile material could be created to guide cleanup efforts.

Previous research has largely focused on locating point sources of radiation while ignoring distributed sources. Additionally, little research has been put into creating quality flux maps except in the field of geological survey. Nearly all prior research has employed the use of directional sensors which limits the usefulness of their approaches.

This thesis demonstrates a set of algorithms that can locate sources and generate maps of expected flux within and surrounding surveyed regions using measurements from non-directional gamma ray sensors.

The efficacy of these solutions is demonstrated by comparing estimated versus actual flux maps as well as estimated versus actual source maps .

Acknowledgments

Thanks to my wife for sticking by my side while I went back to school for this degree. You followed me from California even though you would have preferred to stay. My parents, John and Kay Buffington for instilling the values that caused me to seek out this degree. My grandmother, Helen Rockey, who provided the means for me to go back to school. My sister Heidi Buffington for her help with proof-reading. Red Whittaker for being my adviser. You took me on despite my unusual background. Finally, thank you to Michael Kaess and Joseph Bartels for being part of my thesis committee.

Contents

List of Figures

Chapter 1

Introduction

During a nuclear facility cleanup, whether for decommissioning purposes or after an accident presents a situation where having accurate maps of radioactive source locations and the flux generated by them would be useful. Figure 1.1 shows a situation where these maps would have assisted a cleanup effort. This picture was taken at the Chernobyl nuclear facility in 1986 where a steam explosion and subsequent graphite fire destroyed a nuclear reactor. Workers are shown searching for and removing radioactive graphite that was used in the reactor. If a mobile robot had taken a series of measurements in and around this location before workers approached, maps built from those measurements could have guided the workers directly to larger pieces of graphite without the need to search. This would have limited the workers exposure to radiation damage.

The fundamental particles that all things in existence can be chemically divided into are called atoms. Atoms are made of protons, neutrons, and electrons. Protons and neutrons exist within a small, central nucleus and electrons orbit around them. Atoms are categorized as different elements by the number of protons contained within their nucleus. These elements are further differentiated by the number of neutrons within their nucleus. Two atoms of the same element with a differing number of neutrons are called isotopes of that element. Some isotopes are stable and will retain the same number of electrons, protons, and neutrons forever if not influenced by external forces. Other isotopes are unstable. These isotopes are described as radioactive and can spontaneously undergo a process called decay where they emit high-energy particles which can break chemical bonds or even transmute other elements. These interactions are hazardous to lifeforms and can have negative effects upon man-made materials.

While there are three main products of nuclear decay, only gamma rays have the ability to penetrate more than two meters from the atom which emitted them. Because of this, only gamma rays will be considered in this thesis since it is desirable to locate radioactive materials from larger distances. Gamma rays are high energy photons which exist in the same spectrum as radio waves, visible light, and X-rays. They are not affected by magnetic fields and unlike their lower energy relatives, experience no reflection at all and nearly no refraction. Gamma rays are measured by their flux. Flux is typically measured in units of counts per unit time and like other photons on the electromagnetic spectrum, gamma ray flux falls off with the square of the distance from a point source.

A few terms will be repeatedly used throughout this thesis. Sources and emitters will both refer to materials that produce gamma rays. Emitters will typically be point sources. The term

Figure 1.1: Workers removing radioactive graphite at the Chernobyl nuclear facility in 1986

samples will be used to describe measurements taken at discrete locations by any sensor that can detect gamma rays. These samples will represent a location and gamma flux measurement. A sampled region will describe the area within the convex hull defined by measurement locations. Finally, the rate which nuclear material decays will be referred to as its activity.

Previous research has largely focused on identifying the locations and activities of sources. Typically, mechanical means are used to attenuate gamma rays unevenly to give a sense of directionality to one or more omnidirectional sensors. There are two methods frequently used to locate sources. The first generates a two dimensional representation of what is being observed from a single location. Approximately half of prior research projects that used a single location resolved sources to points. The remaining projects provided a true image. The second method used to locate sources was to observe an object or location from multiple viewpoints. Of these, approximately half resolved sources to points, often in two dimensions. Most of the remaining projects located sources in three dimensions and could resolve sources that were larger than points.

While some research could produce images of sources in front of the sensor from a single location, these images are not maps but rather perspective projections of the sources. It is likely that those methods could provide maps if observations were made from multiple locations. All of the research projects that observed a scene from multiple locations were capable of producing a true map that gave coordinates in two or three dimensions for the observed sources. The vast majority of sensing strategies that have been used to create source maps used directional sensors which preclude their use in many situations due to their cost, size, or weight. While all prior

research related to mapping used flux measurements, only in the field of geological survey were these measurements used to create a dense map of expected flux.

This thesis describes three methodologies that create dense flux maps as well as characterize sources from sparse nondirectional gamma flux measurements. These measurements can be collected using a sensor such as a Geiger counter, which is inexpensive, readily available, and light enough that it can be used on nearly any robotic platform. This thesis starts by describing the simulated annealing algorithm which is similar to one of the algorithms that will be presented later. It then reviews several algorithms that have been used to locate sources of radiation. Descriptions of devices that can detect gamma rays in directional and non directional manners will follow. The next chapter will briefly describe the four algorithms developed for this research and then discuss how they are used together to create source and flux maps. It will follow with detailed descriptions of the individual workflows and the algorithms work. After each workflow is described, results generated by them will be shown and discussed. Future work will then be discussed and this thesis will conclude.

Chapter 2

Related work

2.1 Simulated annealing

Simulated annealing is an algorithm that has a wide variety of uses such as integrated circuit layout [1], modeling seismic waveforms, financial predictions, and logistics [2]. In more general terms, simulated annealing is a good solution for combinatorial optimization problems that require near optimal solutions when the set of possible solutions is extremely large and can't be exhaustively examined. While no examples of simulated annealing could be found where it is applied to the problem of locating and estimating the activity of radioactive sources, it shares a strong similarity to an algorithm used in this thesis. This algorithm will be described in Chapter 5. Simulated annealing is loosely modeled after physical annealing where a metal is initially heated to a high temperature and then slowly lowered This causes the internal crystal structures to grow large which makes the material become more ductile and more easily worked. In a fully annealed state, the internal stresses in the material are the lowest possible.

Simulated annealing algorithm starts with an initial state that is set either randomly or based on an initial best guess. It then calculates a score for that state. After this initial setup, it begins an iterative process of refining the state. For each iteration it perturbs the state and then calculates a score for it. If the new score is better then that state is accepted and iteration continues. If the new state has a worse score then this state is conditionally accepted according to a cooling schedule [3]. The concept of accepting worse solutions prevents the simulated annealing algorithm from becoming stuck in a local minimum. Without this, simulated annealing is just a hill-climbing algorithm [4]. Pseudocode for the simulated annealing process is shown in algorithm 1.

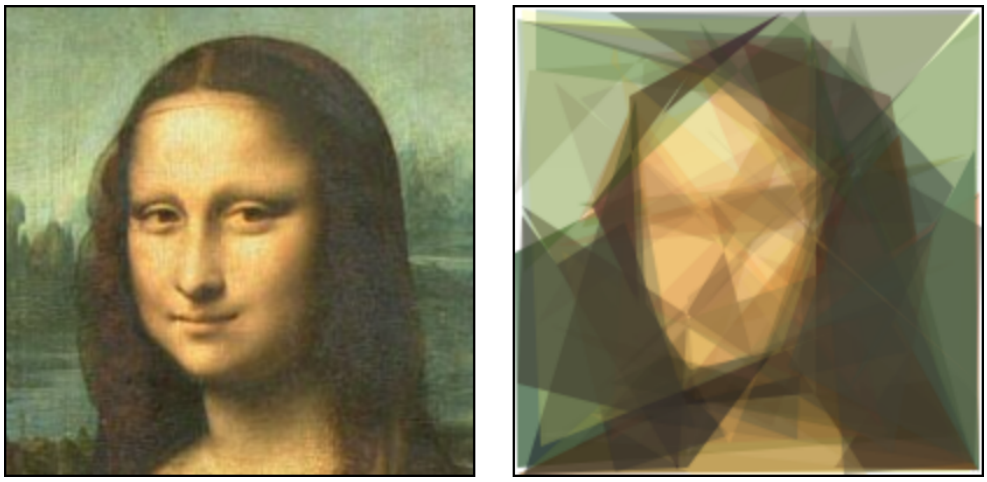
The temperature referenced in Algorithm 1 is related to the cooling schedule. There is no consensus about how a cooling schedule should be set. Instead, the temperature should be modified to best fit the needs of the task. It has been proven that the temperature should decrease logarithmically for the algorithm to be guaranteed to arrive at the global minimum. There are many variants to this cooling schedule such as simulated quenching [2], which uses an exponential decay to increase the speed of convergence. In all cases the temperature will initially be very high and will proceed towards zero in some manner. This causes the algorithm to initially accept a large proportion of changes that decrease the level of fitness. As more iterations happen, progressively fewer decreases to the level of fitness will be allowed. Figure 2.1 shows exam-

Algorithm 1 Pseudocode for the simulated annealing algorithm

$currentSolution \leftarrow generateInitialSolution()$
$currentScore \leftarrow calculateScore(currentSolution)$
while iterations < maxIterations AND score > stopScore **do**
    $tempSolution \leftarrow perturbCurrentSolution(currentSolution)$
    $temperature = calculateTemperature(iteration, maxTemperature)$
    $tempScore \leftarrow calculateScore(tempSolution)$
    if tempScore > currentScore **then**
        $currentScore \leftarrow tempScore$
        $currentSolution \leftarrow tempSolution$
    else
        $acceptanceProbability \leftarrow exp(\frac{currentScore - tempScore}{temperature})$
        if acceptanceProbability > rand(0..1) **then**
            $currentScore \leftarrow tempScore$
            $currentSolution \leftarrow tempSolution$
        end if
    end if
end while

ple output from a program which uses simulated annealing to replicate images using a series of semi-transparent colored polygons.

Figure 2.1: Example output from a simulated annealing algorithm[1]



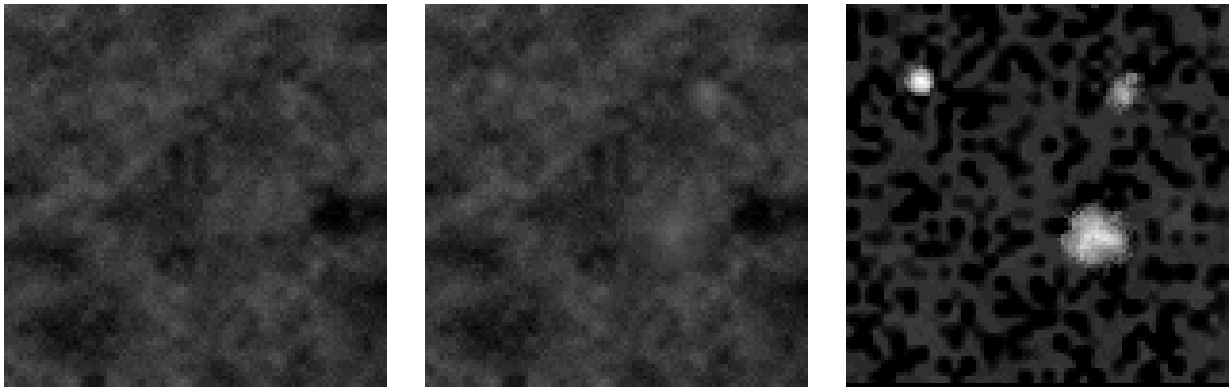
[1]Source: `http://alteredqualia.com/visualization/evolve/`

2.2 Algorithms for locating gamma ray sources

Background radiation subtraction for aerial surveys to isolate man-made sources

Previous processes that have been described dealt with mapping the location and activity of naturally occurring sources of radiation. When trying to locate man-made sources of radiation, it can sometimes be difficult to determine what is a natural source of radiation and what is caused by human activity. If prior aerial surveys were performed, that data could be subtracted from newly recorded data to remove most of the natural background radiation from the survey area. This leaves just the sources of radiation caused by human activity [5]. Figure 2.2 shows a simulated example of a situation where background subtraction reveals three sources that weren't previously there. On the left is a previously created map. In the middle is a newly created map which contains the three new sources. On the right is the result of subtracting the old map from the new map with enhanced contrast. The three new sources show up clearly. This method is not without false positives. Due to the inherent noise in the measurements, there will be differences between the two maps. In order to be detected, the new sources will have to be active enough to be seen above this noise floor.

Figure 2.2: An example of background subtraction

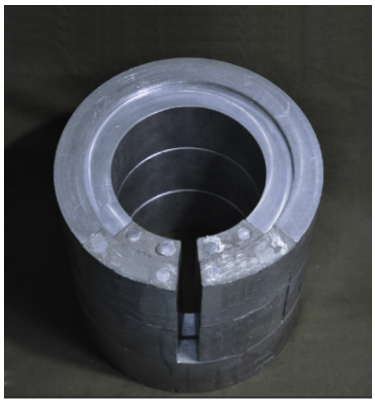


2.2.1 Maximum Likelihood Expectation Maximization

Maximum Likelihood Expectation Maximization, which is abbreviated as MLEM and sometimes referred to as just EM is a common algorithm used to locate sources of radiation from recorded data. MLEM is an iterative algorithm that progressively tries to find better solutions. It does this by first performing an expectation step, which weights how much each data point is represented by each variable in the current model. Next it performs a maximization step that adjusts the model based on the weights applied in the previous step [6]; [7].

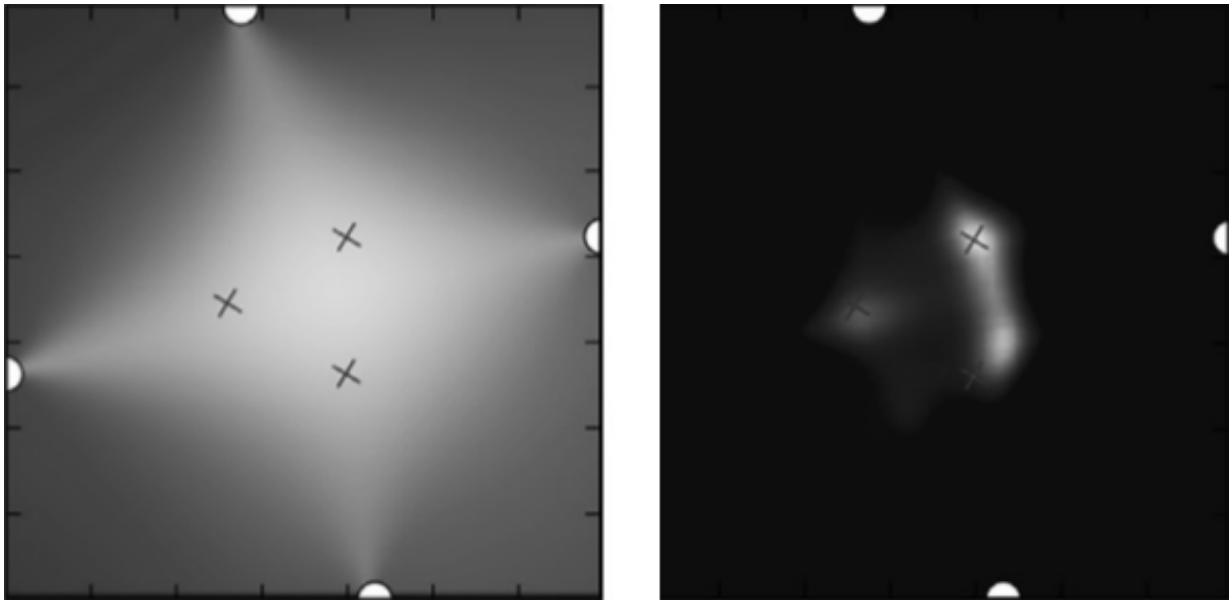
An example of where MLEM is used when locating nuclear radiation is given in [8] where a tubular rotating slit collimator was used in conjunction with a non-directional germanium detector. The collimator allowed the researchers to collect data about the observed flux for a 360 degree sweep around each location where the sensor was used. In the experiment, three caesium point sources were used. Because the collimator had a slit with a width that was greater than zero, the resulting graphs of the observed flux looked similar to a Gaussian for every observed source. The original data was sharpened using the Richardson-Lucey deconvolution algorithm and then a MLEM estimation was made using data from all sample locations to determine the original positions of the sources. Figure 2.3 shows the collimator that was used for this experiment and Figure 2.4 shows some results. The image on the left of Figure 2.4 shows the output after one iteration of MLEM and the image on the right is after fifteen iterations. The locations where measurements were taken are shown as white half-circles on the perimeter of the images and the locations of the sources are shown as black X's.

Figure 2.3: A rotating slit collimator[2]



[2]Image from [8]

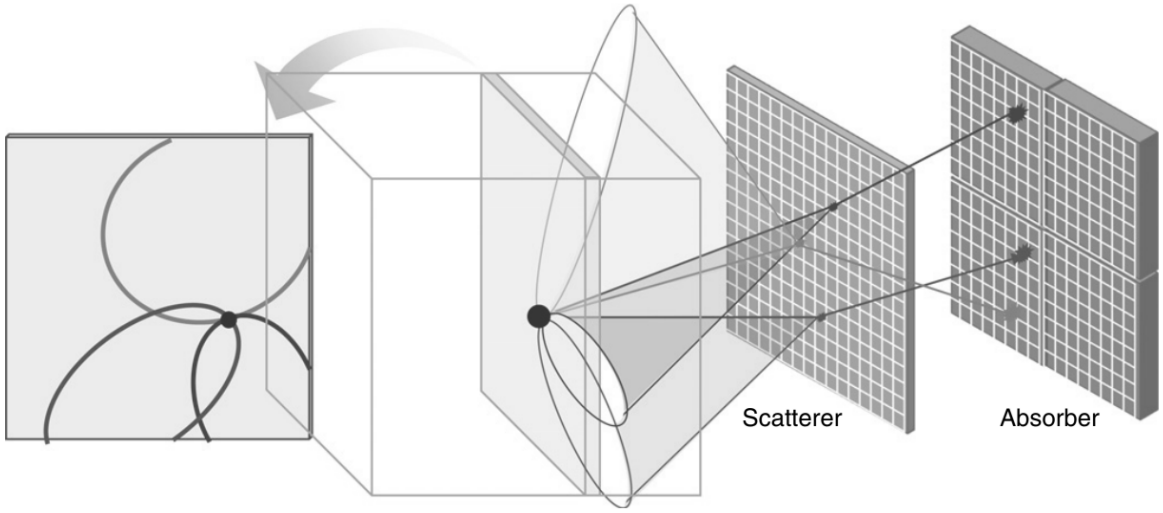
Figure 2.4: Results from the rotating slit collimator experiment[3]



[3]Image from [8]

Another example of where the MLEM algorithm is used to locate sources of radiation is when the raw data from a Compton camera is being processed. When a high-energy photon strikes the scintillator inside of a camera and the result is a Compton scattering, the gamma ray will lose some of its energy and an electron will be liberated from one of the scintillator's atoms. If the gamma ray interacts once again with part of the scintillator, then the direction of the original gamma ray can be resolved down to the surface of a cone due to the physics of Compton scattering[6]. For a single point source, the direction to the source can be quickly determined using simple math. If, on the other hand, the source is an area source or multiple point sources, using the MLEM algorithm becomes necessary[9]. Figure 2.5 shows three cones resulting from interactions with three gamma rays from the same source. Given enough of these interactions, the MLEM algorithm can produce an image that shows likely source locations.

Figure 2.5: Intersecting cones used to determine direction to gamma ray source[4]



Rotational modulator collimators are a type of collimator where there are two parts. Both are identical with the exception that they are sometimes scaled versions of each other. The individual pieces are made of many parallel slits. When these pieces are rotated in tandem, they alternately attenuate or let gamma rays pass freely depending on the angle to the sensor. The sensing element is a non-directional sensor. MLEM is one algorithm that can be used with this sort of collimator to figure out where a source of gamma rays is located [11].

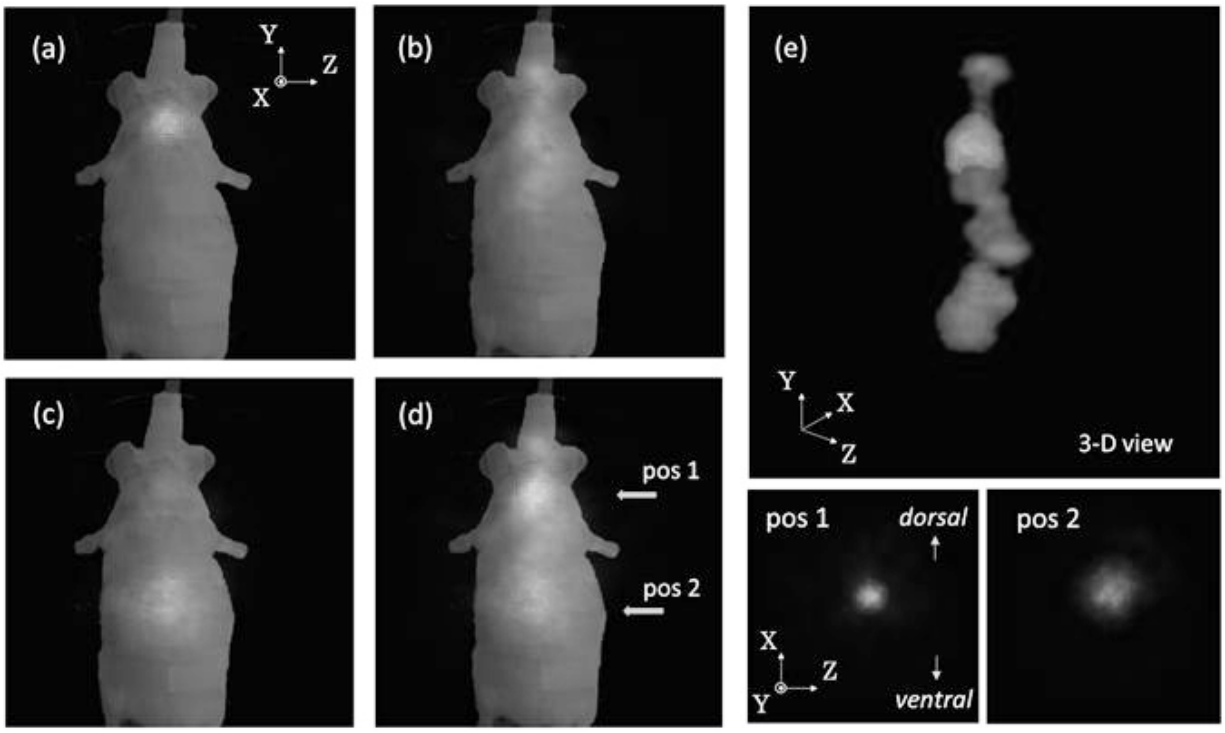
[4]Image from [10]

2.2.2 Computed tomography algorithms

Another class of algorithms that can be used to locate the source of gamma radiation are variations of the algorithms used to decode the data from computed tomography (CT) scans. In [12], the researchers reconstructed a 3D representation of a scene using an off-axis coded aperture rotating slit in conjunction with a scintillation camera. While details were lacking in the paper, it reported "the imaging technique can be modeled mathematically with a formalism analogous to the one employed in CT". CT scans have two major classes of algorithms used to reconstruct their images [13]. These two classes are analytical and iterative. Analytic versions are typically a filtered back projection algorithm such as the inverse Radon transform such as described in [14]. These algorithms are popular because of their computational efficiency. Iterative reconstruction methods typically work by repeatedly forward projecting and then back projecting. Often, iterative reconstruction leads to a result that has less noise than an analytical solution. Recent trends in iterative reconstruction also incorporate expectation maximization within the algorithm. [15]

One other notable experiment that used a computed tomography algorithm to locate sources was [16]. For this experiment, a live mouse was injected with small amounts of three different radioactive materials. As can be seen in Figure 2.6, these materials migrated to different parts of the mouse's body where they were then imaged from multiple angles using a Compton camera. A 3D representation of the sources was reconstructed using a computed tomography algorithm. This representation differentiated between the three sources and displayed them in different colors.

Figure 2.6: Output from the experiment that imaged a mouse[5]



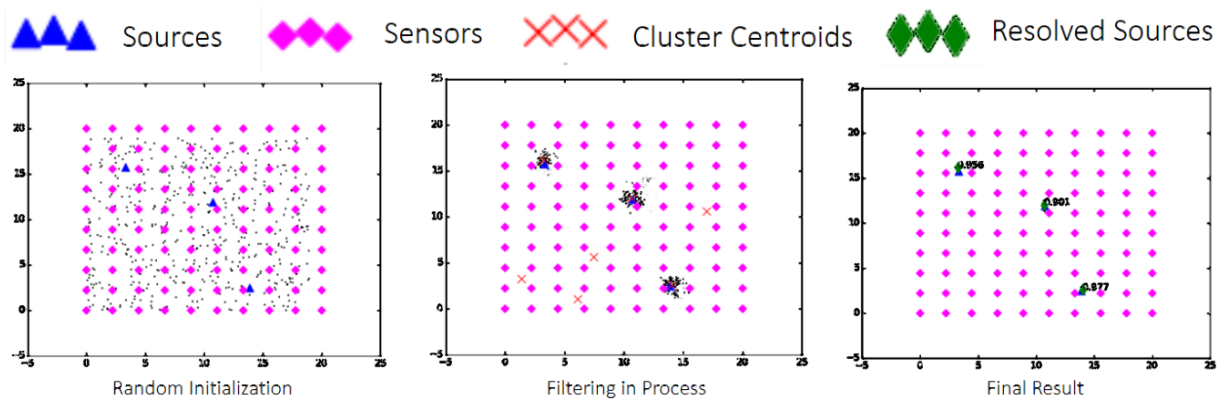
2.2.3 Particle filter

A type of algorithm that has been explored to locate sources of radiation with some success is the particle filter. A particle filter is a sequential Monte Carlo algorithm which starts with a randomly distributed population of particles throughout the solution space. At each iteration, the particle filter moves all of the current particles forward using a control input, which includes a model of the system noise. Then it evaluates how well each particle individually solves the given problem and assigns it a score. Next, it resamples the current set of particles to create a new set of particles before continuing on to the next iteration [17]; [18].

Reference [19] describes a process where multiple sources of radiation were located using a particle filter combined with a clustering algorithm. This process resolved sources into point sources by running the particle filter as previously described. Every nth iteration, it stopped and evaluated the particles by clustering them and representing the whole set of particles by the cluster centers. These are labeled as candidate sources and are evaluated for their quality. If the candidate's quality is high enough, it is labeled as a source which is then considered in later iterations. This process continues until the requested number of sources have been labeled.

Example output from this experiment is shown in Figure 2.7.

Figure 2.7: Locating sources using a particle filter[6]



[5]Image from [16]
[6]Image from [19]

2.2.4 Landform surveying methods

When gamma flux is measured by an aerial survey, sample locations are relatively sparse. Additionally, sample locations are much closer together along the flight path than they are between other nearby passes. In order to estimate the distribution of radioactive material on the ground in the form of deposits, two main methods have historically been used.

Gridding

Gridding is a process where the original data is interpolated onto a set of regularly-spaced grid locations. The grid points can then be used to draw contours or displayed directly. There are a number of gridding algorithms. One such algorithm is bidirectional gridding. This can be used when flight paths are roughly parallel. This algorithm interpolates data points along the flight path at an equal spacing to the grid spacing using spline interpolation. Spline interpolation is once again used between these interpolated values on successive flight paths to arrive at the final grid [20]. A similar gridding algorithm is minimum curvature gridding. This algorithm uses two-dimensional splines. The splines are adjusted using an iterative process and are then used to create the final map. Minimum curvature gridding produces a better result than bidirectional gridding as long as there is no dominant geological trend perpendicular to the flight paths.

Kriging

Kriging is an interpolation method that works well for gamma ray interpolation because it splits data into three components: the general trend, a spatially dependent component, and a noise component. The spatially dependent and noise components are calculated using the semivariances of the data with different time lags [20]. Reference [21] states that "kriging is concerned with prediction of one part of a stochastic process from observations on other parts."

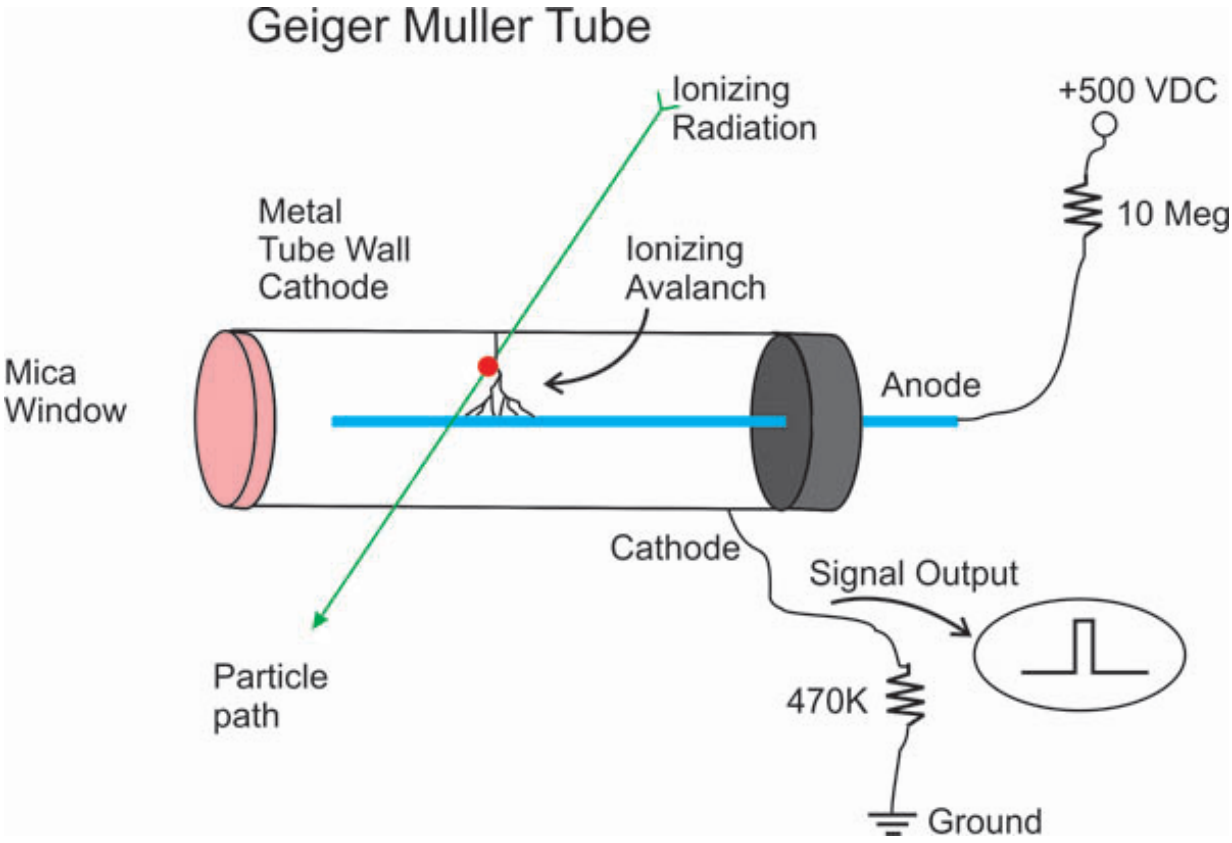
2.3 Nondirectional sensing

2.3.1 Geiger-Müller tubes

One way that gamma rays are detected is through the use of a Geiger-Müller tube, which is a partially-evacuated tube that has an anode running through the center of it. A high voltage is placed between the anode and the outer wall of the tube, which is either made of metal or is coated with a conductive material. When a gamma ray enters the tube and strikes another atom inside of the tube, this atom becomes ionized. Due to the voltage gradient inside of the tube, the freed electrons will be collected at the center of the tube. In a Geiger-Müller tube, the voltage gradient is sufficient that the freed electrons may pick up enough kinetic energy as they move towards the center of the tube that they can also free other electrons [22]. Geiger-Müller tubes have a high enough voltage differential that no determination can be made about the initial energy of an incident gamma ray and therefore can only provide a count of how many gamma rays struck them. One additional characteristic of a Geiger-Müller tube is that there is an amount of time after detecting a count where it will be unable to detect further counts. Currently, the best

available Geiger-Müller detectors can measure up to about 100,000 counts per second [23]. The Geiger-Müller tube is generally used as a non-directional sensor but if it is modified so that its length is much less than its diameter, it can have some sense of directionality.

Figure 2.8: A Geiger-Müller tube[7]



2.3.2 Proportional counters

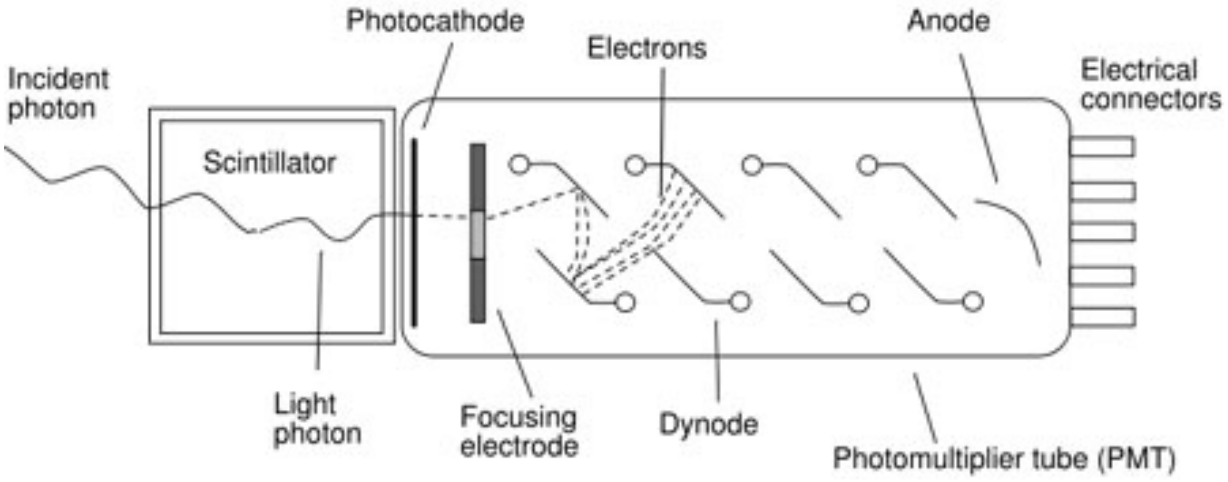
If an evacuated tube uses a lesser voltage than a Geiger-Müller tube between the outside wall and the center anode, it becomes possible to not only detect when a gamma ray hits the detector but also what its energy was. The output pulses can be counted as they were with a Geiger-Müller tube but the amplitude of the pulse will be proportional to the initial energy of the gamma ray.

[7]Source: `https://www.imagesco.com/geiger/geiger-counter-tube.html`

2.3.3 Scintillators

Some materials emit visible light when gamma rays strike them. The incoming gamma rays cause some electrons within the scintillator to move to higher excitation states. When they return to a lower excitation state, they emit a photon that has the energy difference between the two states [24]. This photon can then be detected through a variety of means. One way that these photons are detected is with a photomultiplier tube, which initially converts each photon into an electron via the photoelectric effect. This electron is then passed between a series of plates that have a voltage gradient along the length of the tube. Each time an electron strikes one of the plates, several other electrons are freed and move further down the tube. This has the effect that while there were initially only a handful of electrons that hit the first plate, a very large number of electrons, which are more easily detected, strike the final plate. Another method of detecting photons is through the use of photodiodes, which, unlike photomultiplier tubes, directly detect the photon in one step. One additional property of scintillators is that the number of photons created by a gamma ray is proportional to its energy. This in turn leads to a proportional change in the number of electrons detected by the photomultiplier tube. Because of this, it is possible to identify with some accuracy what isotope a group of gamma rays came from when you consider the spectrum of detected energies.

Figure 2.9: A scintillator and photomultiplier tube[8]



[8]Source: https://web.stanford.edu/group/scintillators/scintillators.html

2.3.4 Semiconductor detectors

Gamma sensors can be built from semiconductor materials. Typically semiconductor detectors have superior energy resolution when compared to scintillating detectors. They are currently limited to small sizes, which depending on the intended use could be beneficial or detrimental. Semiconductor detectors have the downside of degradation due to the radiation that they are sensing so that over time they become less sensitive. This reduction of sensitivity can be largely eliminated by operating the semiconductor detector at very low temperatures. Semiconductor detectors are made out of materials such as silicon, germanium, or diamond. By doping them so that they act as a diode and then reverse-biasing them, it is possible to detect gamma rays that strike them directly [25].

2.4 Directional sensing

2.4.1 Radiation-sensitive materials

While the previous types of detectors have all been electrically operated, radiation-sensitive materials require no electrical connection at all. The way that they function is that when a gamma ray strikes these materials, they have a property change. An example of this is the Radball detector [26]. It is made of a radiation sensitive polymer sphere which is surrounded by a tungsten sphere with collimation holes covering its surface. By placing this device within a contaminated area and leaving it for an amount of time, damage to the inside sphere changes its color along the paths that gamma rays passed. When the radball is removed and disassembled, the inner sphere can be optically examined to determine the direction and activity of sources within the examined area. By its very nature, this type of detector gives a sense of what direction radiation is coming from. Another example of a radiation sensitive material is radiographic film. This type of material was first used by Roentgen when he discovered X-rays. Newer radiographic film is sensitive to gamma rays as well.

Figure 2.10: The Radball detector[9]

[9]Image from [26]

2.4.2 Planar detectors

It is possible to construct an electronic gamma detector that is directional. One method is to construct it so that the device is thin and wide like a pancake or consisting of many thin layers. In this configuration, the detector will have a non-uniform response to a source as it is rotated. To avoid confusion about which side of the device the gamma source is coming from, one side of the device should be covered with a layer of high-impedance material such as lead and the other with something that has low impedance like plastic. Reference [27] reports that for gamma rays that have greater than 0.6 MeV of energy, the sensor response is greater when gamma rays enter through the low impedance side. The trend reverses with gamma rays that have less than that amount. The end result is that in order to determine the true direction to a source, one must first locate a direction where a peak in response is found, then the detector must be flipped 180 degrees so that the other face is towards the source. Using knowledge of the detector orientation and the measured energy of the gamma rays, the true direction to the source can be determined.

2.4.3 Tubular slit collimator

The planar direction towards a source of gamma radiation can be found when using a non-directional sensor if it is used in conjunction with a tubular slit collimator. Tubular slit collimators are tubes of a high impedance material that resists the passage of gamma rays. Along one side is a slit. By rotating the collimator, the slit selectively allows gamma rays from one direction to enter freely while blocking most other gamma rays from other directions. With this type of setup, there is a trade off between directional resolution and how quickly measurements can be taken. Widening the slit allows more gamma rays to enter but also decreases the angular selectivity. The width of the slit also determines the minimum angle between two sources where they can no-longer be resolved as separate [8].

2.4.4 Pinhole aperture

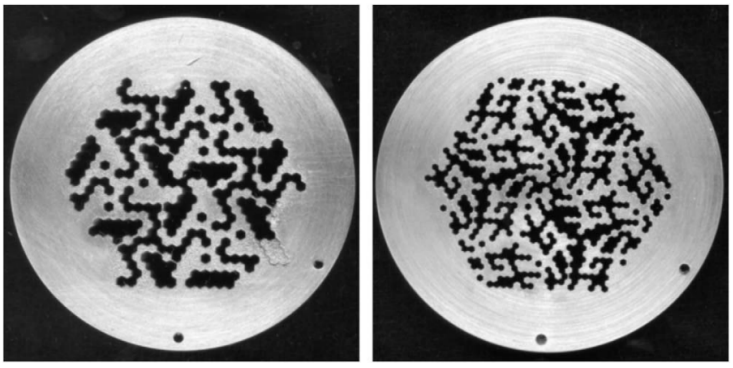
Similar to visible light, it is possible to build a pinhole gamma camera. Unlike its visible light equivalent, the gamma camera must not only have an aperture that is circular when viewed parallel to the image plane but this aperture must be thick and have a cross section that is conical. This sort of arrangement will produce a fuzzy image because gamma rays can still penetrate the aperture where it is thin near the opening. An additional drawback of pinhole apertures is that exposure times must be long if the aperture is small because relatively few gamma rays will pass through the opening. Widening the size of the aperture allows more gamma rays to enter, which decreases the exposure time, but the trade off is that resolving power is reduced as the gamma image becomes fuzzier [28].

2.4.5 Coded apertures

An alternative to using a pinhole aperture is to use multiple apertures. Although the images from each aperture will overlap, the true image can be reconstructed using Fourier convolution. Coded apertures are made of sheets of a dense material that can be cut or drilled in a known

pattern. There are a variety of different strategies for arranging the holes. In all cases, the hole locations are chosen to reduce the amount of noise that results from the Fourier transform. Holes are typically arranged in a grid or hexagonal pattern as can be seen in Figure 2.11 [29];[30].

Figure 2.11: Coded apertures[10]

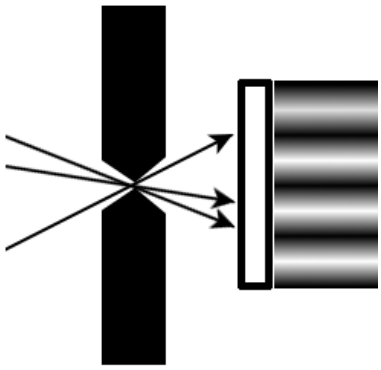


2.4.6 Scintillator with multiple detectors

While a single scintillator with an associated detector is useful for non-directional sensing, pairing a scintillator crystal with some form of aperture allows a scintillator to perform imaging. The way that this is done is to use an array of photomultiplier tubes behind the scintillation crystal. Physically, photomultiplier tubes are fairly large so using the output from them directly would result in a very low resolution image. The resolution issue can be overcome by examining the output of all of the photomultiplier tubes in conjunction. By comparing the relative outputs of the photomultiplier tubes, the two-dimensional position where the scintillation took place can be determined [32].

Figure 2.12 shows a pinhole aperture imaging device. Gamma rays pass through the aperture where they then strike a scintillator. Multiple photomultiplier tubes located behind the scintillator detect the location of the interaction.

Figure 2.12: A pinhole aperture imaging device



[10]Image from [31]

2.4.7 Compton camera

More recently, Compton cameras are starting to be used to determine the activity and direction that a gamma ray came from using no aperture at all. A Compton camera is made from a cubic block of semiconductor made from a material such as cadmium zinc telluride. On one side of the cube there is a single cathode. On the opposite side there is an array of anodes. The three-dimensional position of the interaction can be determined by observing the output of the anodes and cathode. When a gamma ray strikes the semiconductor detector it produces a cloud of electrons. The cathode will always detect the interaction but there will be a weighting potential for the anodes based on where the gamma ray interacted with the detector. The signal measured at the cathode is proportional to the depth through the semiconductor and the charge imparted by the gamma ray. The anode's signals will only be proportional to the charge. Using the information that was measured, the three-dimensional location of the interaction can be computed [33]. If two events are measured in rapid succession in different areas, it can be assumed that these are due to Compton scattering. Using multiple two-event interactions and knowledge of Compton scattering, a probabilistic direction map indicating where gamma rays may have come from can be made. This sort of sensor is omnidirectional and its directional sensitivity is only reduced by the shielding effect of other equipment used to run the sensor such as circuit boards and batteries [34]; [35].

Chapter 3

Algorithms and workflows for mapping source locations and gamma flux fields

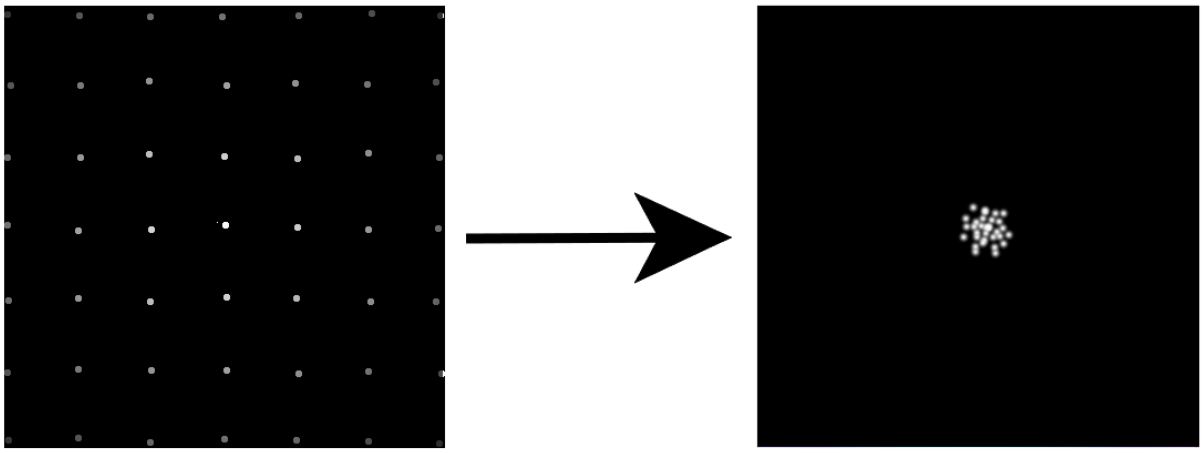
3.1 Algorithms

Four algorithms were developed for this research. This section will give a quick overview of each and discuss its strengths and weaknesses. Later chapters will describe these algorithms in greater detail.

3.1.1 Triangulated gradient method

The triangulated gradient method takes a set of sparse gamma flux measurements and outputs a set of source location estimates. It starts by triangulating the measurement locations. For each triangle, it fits an inverse square gradient across the triangle so that the measurements at the vertices match the values of the gradient at those same locations. It uses the direction of the gradient and the knowledge of where the inverse square curve goes to infinity to determine the source location. This algorithm is computationally efficient and could run on a mobile robot to help it make decisions about where it should take gamma flux measurements. A drawback of this algorithm is that multiple sources, large area sources, and linear sources can cause incorrect though still useful estimates. Additionally, noise in gamma flux measurements can cause estimated source locations to move have positional error.

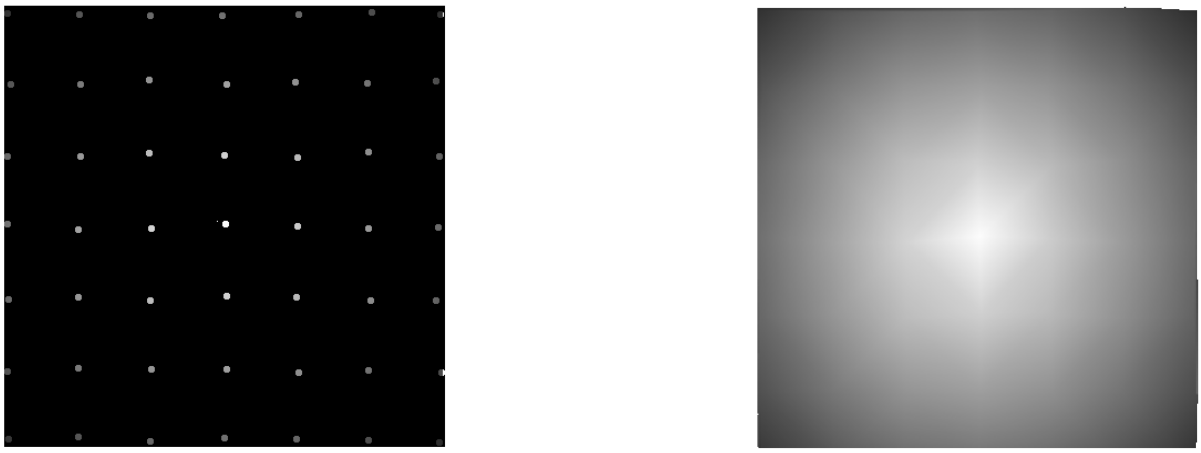
Figure 3.1: Triangulated gradient method



3.1.2 Inverse square renderer

The inverse square renderer takes a set of sparse gamma flux measurements and outputs a dense gamma flux map for the area within the sampled region. This algorithm is very similar to the triangulated gradient method. The main difference is that after the gradient is fit to the triangle it is rendered within the triangle to create a map of dense flux values within the triangle. This algorithm is computationally efficient and is suitable for use on a mobile robot that needs a dense flux map. Because it is based on the triangulated gradient method, this algorithm can also optionally provide estimates about where sources may be located. This algorithm has some drawbacks. Although it outputs a dense gamma flux map, the error is higher than what can be achieved using other methodologies detailed later in this thesis. The inverse square renderer is also more susceptible to noise in the gamma flux measurements because, in essence, it is interpolating directly between these measurements. Finally, while helpful, the estimated source locations may not be as accurate as needed if there are multiple sources in the region being surveyed or if there is a single source that is relatively large.

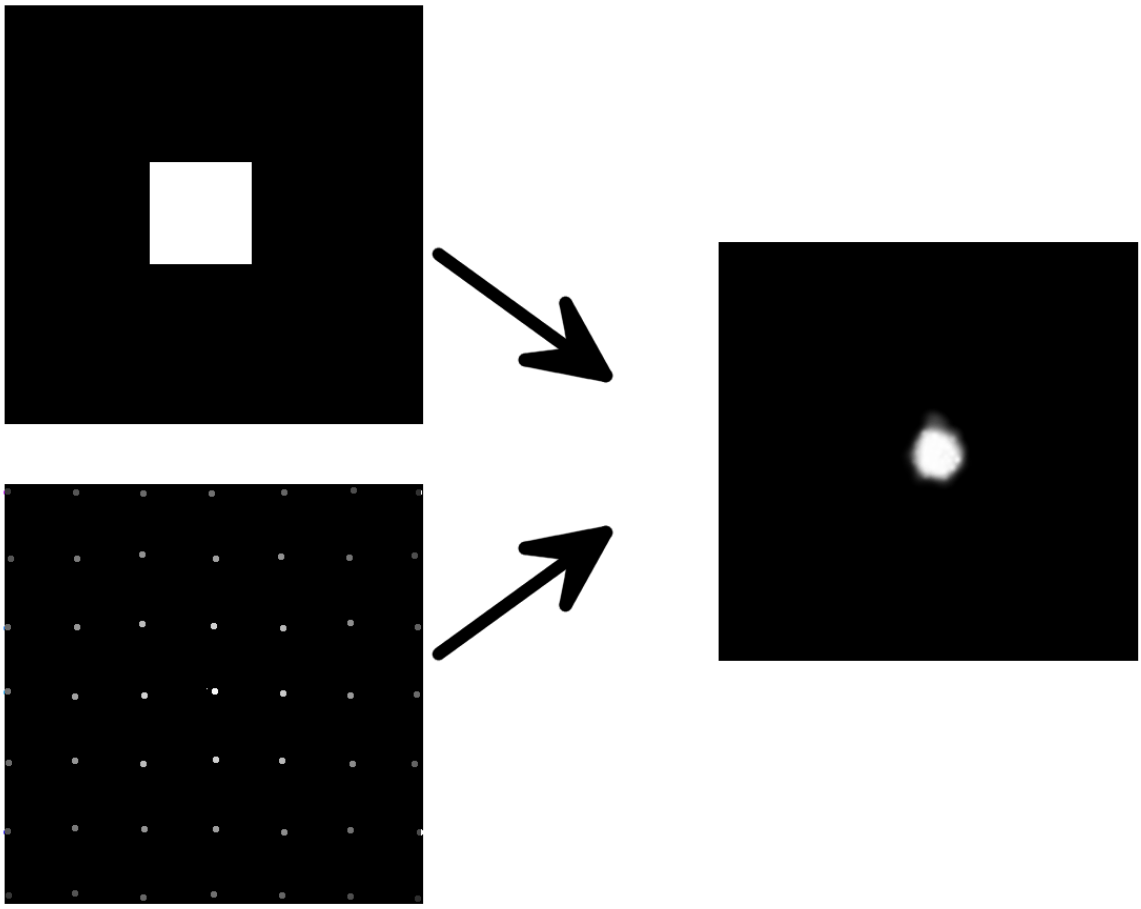
Figure 3.2: Inverse square renderer



3.1.3 Flux match annealing

The flux match annealing algorithm takes a set of sparse gamma flux measurements and a set of emitters that are placed in likely source locations and outputs the emitters with activity levels that cause a flux field that matches the flux measurements as closely as possible. This algorithm is similar to the simulated annealing algorithm. Flux match annealing replicates flux fields with low amounts of error. It is mostly immune to the shot noise present in all gamma flux measurements and can still produce usable results even when the median number of counts at each location is less than twenty. A drawback to this algorithm is that it is computationally expensive and is better suited to offline mapping rather than on a mobile robot. Typical processing times based on the number of emitters and sample locations used in this research varied between one and five minutes. Code was written in Matlab and ran on a mid-range i7 laptop purchased in 2017. While this algorithm estimates source locations and activities that produce flux that closely matches measured flux, the resulting emitter activity map is often not representative of the actual source locations and activities.

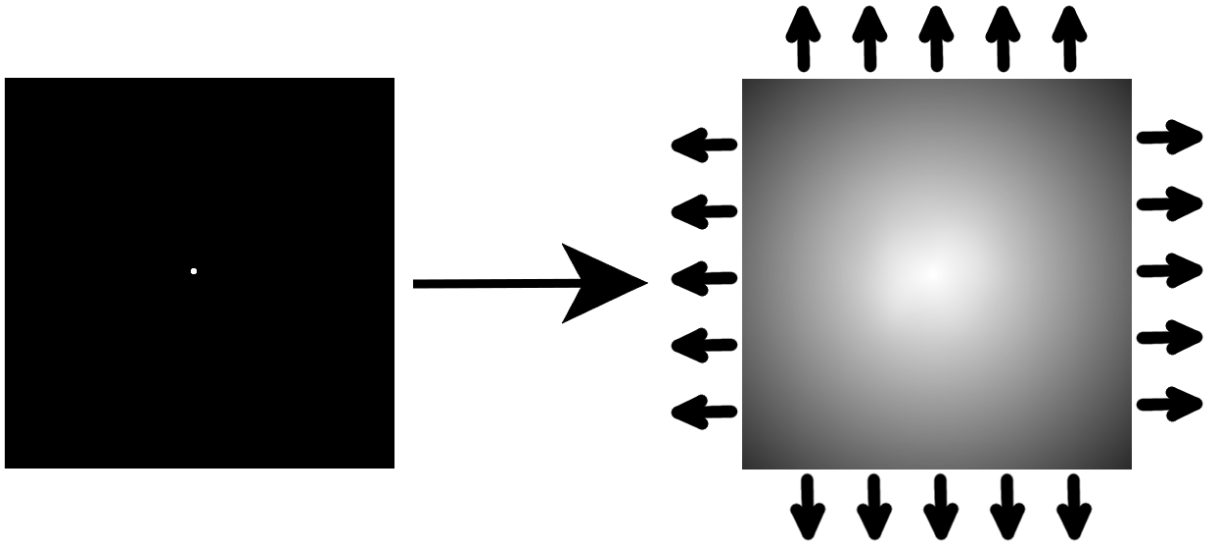
Figure 3.3: Flux match annealing



3.1.4 Point source renderer

The point source renderer takes a set of emitter locations and activities and outputs a dense gamma flux map. A modified version of this algorithm that renders flux only at specified locations is used by the flux match annealing algorithm at each iteration to estimate the flux at the measurement locations. This algorithm runs relatively quickly and is capable of rendering flux maps for any desired region. Its speed is linearly proportional to both the requested number of flux estimates and the number of emitters passed to it.

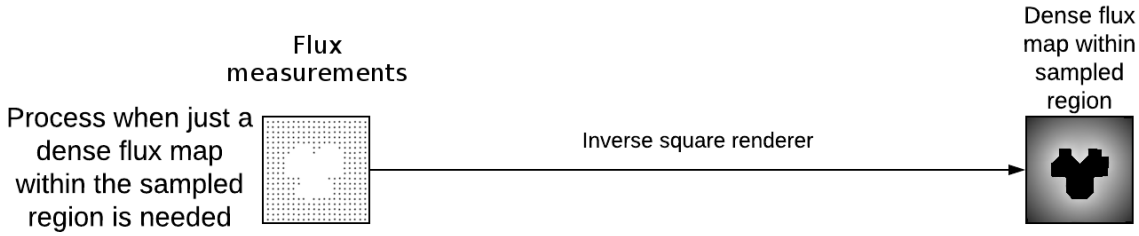
Figure 3.4: Point source renderer



3.1.5 Workflows

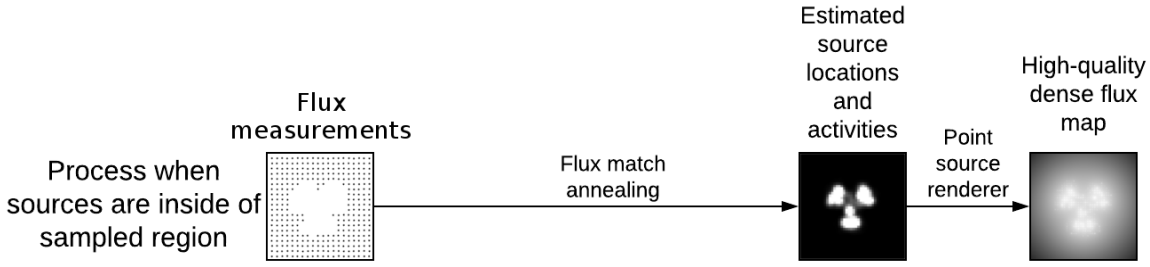
There are three workflows that use the algorithms described in this thesis. The first workflow takes a set of sparse gamma flux measurements and outputs a dense gamma flux map within the convex hull of the measurement locations. It optionally produces estimates of source locations.

Figure 3.5: First workflow



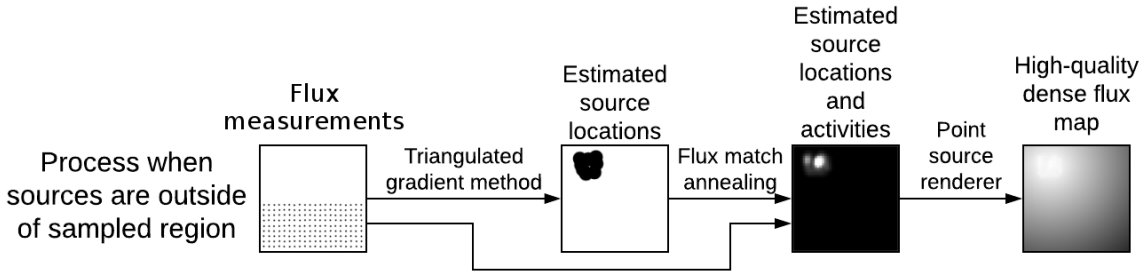
The second workflow takes a set of sparse gamma flux measurements and outputs a dense gamma flux map that can be extrapolated beyond the sampled region. This workflow is optimal for situations where the measurement locations surround all sources.

Figure 3.6: Second workflow



The third workflow takes a set of sparse gamma flux measurements and outputs a dense gamma flux map that can be extrapolated beyond the sampled region. This workflow is suggested for situations where the sources are partially or fully outside of the sampled region.

Figure 3.7: Third workflow



Chapter 4

The point source renderer algorithm

The point source renderer takes a set of emitter locations and activities and uses them to generate estimated gamma flux values for locations within a map. There are two variants of the point source renderer used in this thesis. One generates a dense flux map within a specified region. The other only generates flux values for specific locations. The first variant is used for final output where an image or dense flux map is required. The second is used in the main loop of the flux match annealing algorithm as a part of assessing the suitability of a solution.

The operation of the point source renderer is relatively straightforward. For each location where an estimate of the gamma flux is required, the renderer calculates the distance from that location to the locations of each of the emitters. Next it uses Equation 4.1 to arrive at the individual contribution to the total flux from each emitter.

$$flux = \frac{activity}{4\pi r^2} \tag{4.1}$$

The flux contributions from all emitters are summed together to arrive at a final flux value for the desired location. Algorithm 2 shows pseudocode for the point source renderer.

Algorithm 2 Pseudocode for calculating flux values from a set of point source emitters

for every desired location **do**
  $flux[location] \leftarrow 0$
  for every point source emitter **do**
    $dist \leftarrow euclideanDistance(location, emitter)$
    $activity \leftarrow thisEmitter.activity$
    $thisEmittersContribution \leftarrow \frac{activity}{4*\pi*dist^2}$
    $flux[location] \mathrel{+}= thisEmittersContribution$
  end for
end for

Chapter 5

The flux match annealing algorithm

The flux match annealing algorithm takes a set of sparse gamma flux measurements and a set of emitters and outputs the emitters with activities optimized to produce flux that closely matches the provided measurements.

Flux match annealing is similar to simulated annealing, which was described in the related work section. The main difference is how solutions that increase the error between estimated and actual flux are accepted. In simulated annealing, any solution that increases this error can potentially be accepted using Formula 5.1. In this formula, the temperature variable is controlled by a cooling schedule and is initially high, which allows many increases in error to be accepted. As the number of iterations increases, the temperature is progressively reduced to allow fewer increases in error to be accepted. Flux match annealing uses a different process that moves faster towards a solution. Every Nth iteration, it accepts any solution that doesn't increase the error by more than a certain fixed percent. This works because of the second change that has been made. While simulated annealing uses a fixed method to modify the state, flux match annealing modifies the state in a continually variable manner. In each iteration, the activities of some of the emitters are adjusted using a Gaussian random variable with a mean of zero. The standard deviation of this random variable is proportional to the sum of the squared flux errors at the measurement locations.

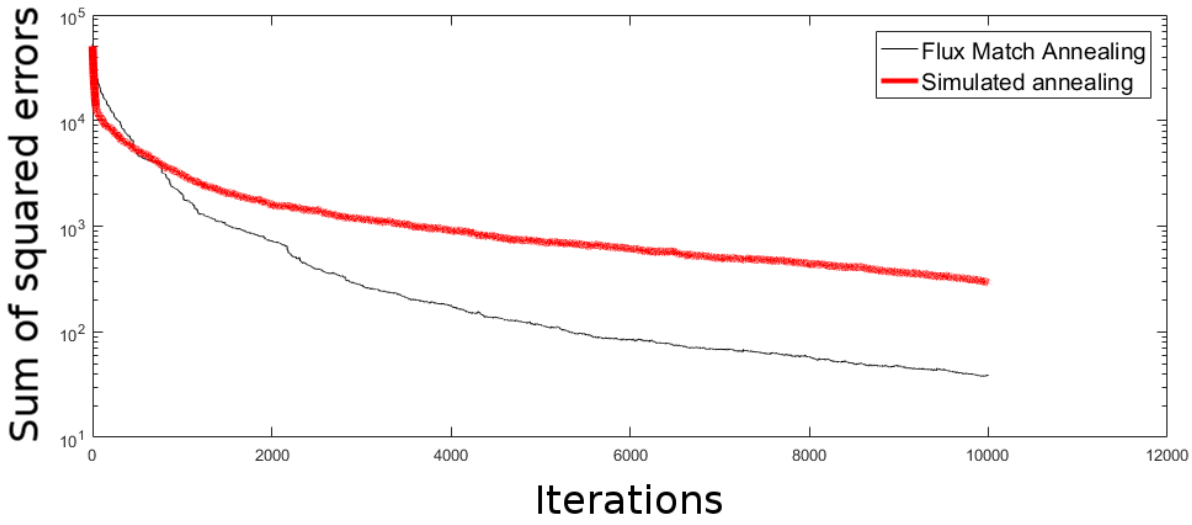
Figure 5.1 shows a comparison of the sum of the squared flux errors over time for code using the simulated annealing acceptance function and the acceptance function used in flux match annealing. Both plot lines represent the average of 40 trials which ran for 10,000 iterations each. Pseudocode for flux match annealing is shown in Algorithm 3.

$$accept = \exp \frac{\Delta fitness}{temperature} > rand(0 \rightarrow 1) \tag{5.1}$$

Algorithm 3 Pseudocode for the Flux Match Annealing algorithm

$emitterLocations \leftarrow decideOnEmitterLocations()$
$sigmaConstant \leftarrow 20$
$acceptanceFreq \leftarrow 8$
$maxPctIncrease \leftarrow 0.2$ ▷ 20%
$activities \leftarrow initializeEmitterActivities()$
$lastScore \leftarrow \sum measuredFlux^2$
$sigma \leftarrow \frac{lastScore}{sigmaConstant}$
while $iteration \neq maxIterations$ **do**
  $tempActivities \leftarrow activities$
  $rectangle \leftarrow randomlyGenerateRectangle()$
  $selectedEmitters \leftarrow selectEmittersUsingRectangle(rectangle, emitterLocations)$
  $adjustment \leftarrow gaussian(0, sigma)$
  $tempActivities \leftarrow tempActivities + adjustment$
  $tempActivities \leftarrow max(tempActivites, 0)$ ▷ Don't let them go negative
  $computedFlux \leftarrow computeFlux(tempActivities, emitterLocations)$
  $thisScore = \sum(measuredFlux - computedFlux)^2$
  if thisScore < lastScore **then**
    $lastScore \leftarrow thisScore$
    $activities \leftarrow tempActivities$
  else
    $pctIncrease \leftarrow \frac{thisScore - lastScore}{lastScore}$
    if pctIncrease < maxPctIncrease AND (iteration MOD acceptanceFreq == 0) **then**
      $lastScore \leftarrow thisScore$
      $activities \leftarrow tempActivities$
    end if
  end if
  $sigma \leftarrow \frac{lastScore}{sigmaConstant}$
end while

Figure 5.1: A comparison of error over time for simulated annealing versus flux match annealing



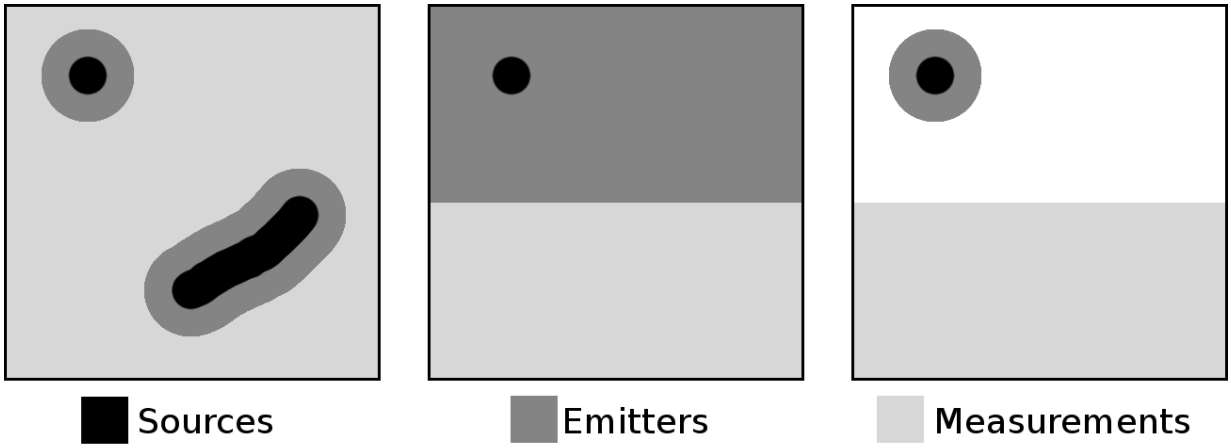
Before the flux match annealing algorithm can run, emitter locations must be decided upon. The initial activity for all emitters is zero. The examples in this thesis place emitters using the three strategies shown in Figure 5.2.

The first placement strategy is used for situations where sources are mostly or completely surrounded by measurement locations. This strategy operates on the assumption that there are spaces around sources where no measurements were taken. These spaces are filled with emitter locations in a grid pattern. The rationale behind taking measurements in this manner is that a mobile robot designed to take flux measurements would be likely to take the measurements in a manner that sampled as closely as possible to the sources without traversing directly over them.

The second strategy to place emitters is used when a source is located outside of the sampled region. In this situation, a large area that includes the source locations is filled with a grid of emitters and the flux match annealing algorithm is left to determine where the sources are located on its own.

The third method is also used when sources are located outside of the sampled region. It is used to improve the speed and accuracy of the flux match annealing algorithm. This strategy uses the triangulated gradient method to estimate likely source locations. Emitters are then placed in a grid pattern within a fixed distance of the estimated source locations.

Figure 5.2: The three strategies for placing emitters



Once the locations of the emitters have been decided, the flux match annealing algorithm can start. First, some constants are set to values that were determined through experimentation. These experiments will be described later. An initial score is calculated which is the sum of the squares of the measured fluxes. The last step before starting the iterative portion of the code is to initialize a sigma variable that is used when perturbing the emitters' activity levels.

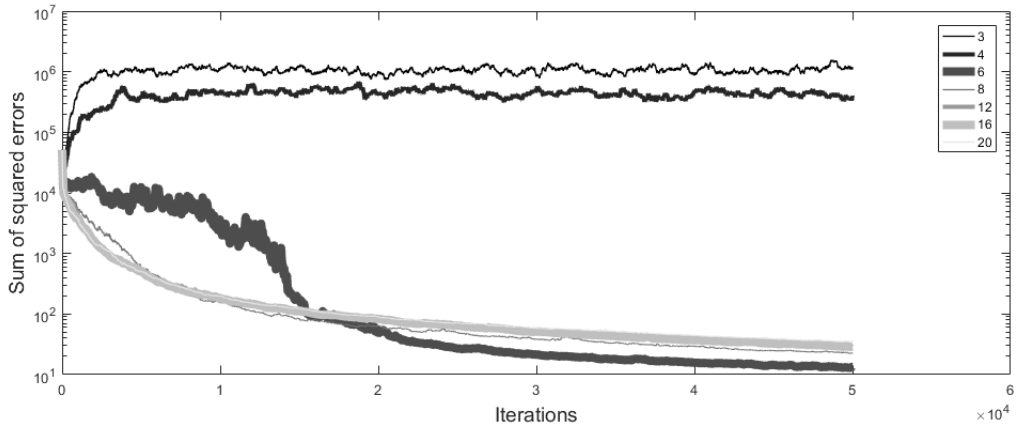
In the iterative portion, rectangular regions are randomly selected until a region contains at least one emitter. Then, a Gaussian random number with a mean of zero and a standard deviation of 'sigma' is found. This value is added to the selected emitters activity. A check is done to make sure that no emitter's activity has gone below zero. A score is found by summing the squared flux errors between the calculated and actual fluxes. Lower scores are better as they indicate that the solution more closely matches what was measured. If the new score is lower than the old score this set of emitters is accepted as the best solution. Every Nth iteration, if the score went up, it will still accept the new set of emitters as the best solution as long as the score didn't go up by more than a certain percentage. The last thing that is done within the iterative section is that a new sigma value is generated. This value is the last score divided by the constant "sigmaConstant", which will be discussed later.

There are three variables that can be changed to tune the performance of the flux match annealing algorithm. The first determines the maximum allowable increase in error. This is maxPctIncrease in the pseudocode. This variable has a wide range of acceptable values which range from around 2% to 1000%. At very low values, this variable tends to prevent the algorithm from jumping out of local minimums. This causes the final solution to have more error than it could have had with a higher percentage. At very high values, the algorithm can be unstable when it first starts. When starting, changes in the score of over 1,000 percent are possible. This instability is more common with maps that have a source that is physically large. Limiting the maximum increase in score helps the algorithm through the initial period of instability. The value used by this research was twenty percent.

The second variable that can be adjusted to refine the performance of this algorithm determines how often it allows an increasing amount of error to be accepted. In the pseudocode this

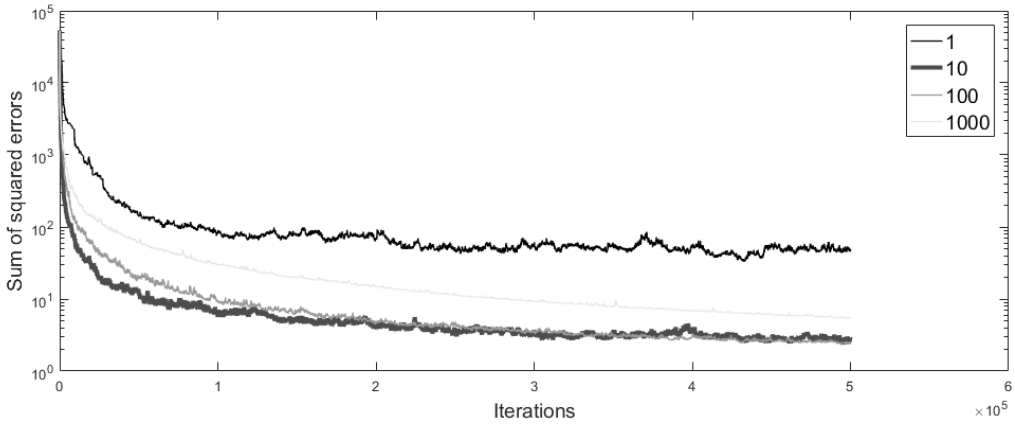
variable is "acceptanceFreq". Figure 5.3 plots out the progress of the score for different acceptance frequency values. Lower scores are better. Each line represents the average of 40 trials using a particular acceptance frequency over 50,000 iterations. A maximum of 40 percent error increase was allowed for the tests shown. Acceptance frequency settings of three and four diverge. On average, an acceptance frequency of six starts to converge after about 5,000 iterations but several runs had trouble converging at the beginning which throws off the average until about 15,000 iterations. Although a setting of six on average achieves a better score, this research used an acceptance frequency of eight because of its increased stability.

Figure 5.3: Graph showing running scores for different acceptance frequencies



The final variable that can be adjusted to tune the results of this algorithm is the variable that controls how large the standard deviation is when randomly adjusting emitter values. In the pseudocode, this is the variable "sigmaConstant". Changing the sigmaConstant value controls the convergence rate as well as how low of a flux error the algorithm can converge to. Figure 5.4 shows a series of trials which used different sigmaConstant values. Each line represents the average of five trials that have been run for 500,000 iterations each. There are a wide range of values that sigmaConstant can be set to where the algorithm will converge to similar final amounts of flux error. Setting sigmaConstant to a very low value will cause the emitter values to fluctuate wildly which doesn't allow for good results. In Figure 5.4, when sigmaConstant is set to one, the final result is more than an order of magnitude worse than other results. Setting sigmaConstant to a value that is too high causes the emitter values to fluctuate by very small amounts, which has two effects. The first is that because the emitter activities can only change by small amounts, the algorithm takes a large number of iterations to to achieve similar results when compared to when sigmaConstant is lower. The second effect is that the flux match annealing algorithm can't get out of local minimums with too high of a sigmaConstant. One interesting thing to note is that even though values of 10 and 100 cause the curve to mostly level off by around 300,000 iterations, it never becomes completely flat and improvement is still seen through the remainder of the iterations. This was not the case when sigmaConstant was set to 1. Instead, the graph levels off around 100,000 iterations where it remains flat until 200,000 iterations where it drops a small amount again and then stays flat for the remaining 300,000 iterations.

Figure 5.4: Graph showing running scores with varying sigmaConstants



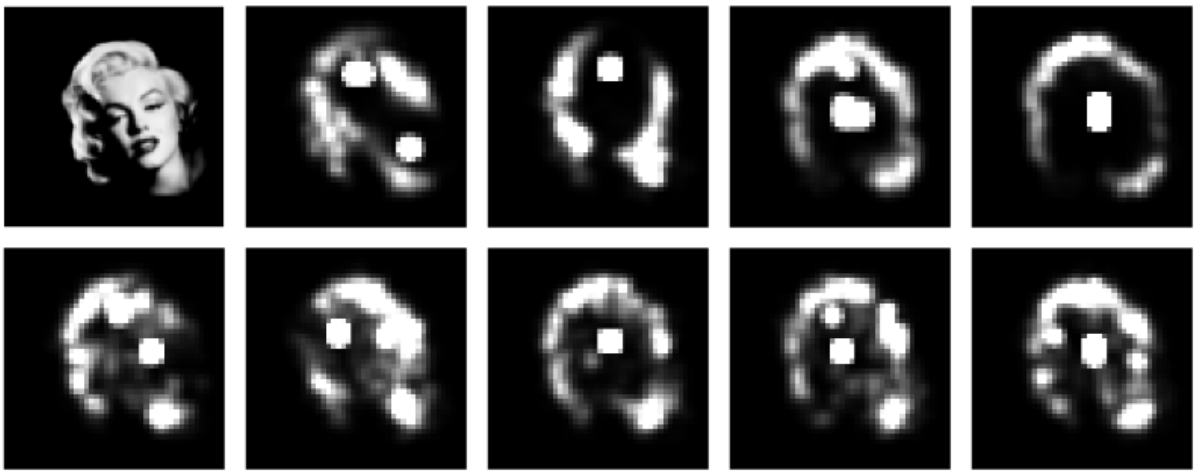
5.1 Replicating source shape and activity

While this algorithm is, in part, intended to help determine the shape and activity level of arbitrarily-shaped area sources, the reality is that no algorithm can truly replicate exactly what is happening in areas that haven't been directly surveyed. This algorithm creates a set of emitters that can nearly replicate the measured flux at the sampled locations. When the actual and computed source activities are compared side by side, there can be some profound differences. Figure 5.5 shows a grid of images.

The source map on the top left was used to construct a set of emitters based on its pixels. One point source emitter was created for each pixel. The pixel's value determined the activity of its associated emitter. From these emitters, a map of the flux field emitted by them was created using the point source renderer. A new set of emitters were created for use by the flux match annealing algorithm. These emitters were arranged in a grid everywhere within a fixed radius of any pixel in the source map with a non-zero value. These emitters were given an initial activity of zero. Measurements were taken from the flux field in a sparse grid pattern everywhere that emitters weren't placed. Multiple runs of the flux match annealing algorithm used these measurements to generate the remaining source maps seen in Figure 5.5. Each image shown is the result of 500,000 iterations of the flux match annealing algorithm. When measurements are taken as they were for this experiment, the flux match annealing algorithm tends to replicate the edges of large area sources and then usually places one or more strong sets of emitters towards the center of the active region.

The emitter map used for Figure 5.5 was the most complex map that was used in testing. Figure 5.6 shows the results from six other maps and Figure 5.7 shows the regions where emitters were placed and where measurements were taken for these maps. Where Figure 5.7 is white emitters were placed in a grid pattern. Black regions indicate where measurements were taken. The left column in Figure 5.6 shows the original source maps. The next column to the right shows the flux maps resulting from these sources. The remaining columns show the reconstructed locations and activity levels for sources which were computed using the flux maps. Like the

Figure 5.5: Actual (top left) vs. computed emitters for multiple runs of the annealing algorithm



previous example, the reconstructed source maps tend to have clumps of emitters with high levels of activity. On the other hand, with the exception of the map used in row six, flux match annealing tended to confine the activity to places where it was actually located. This is perhaps most clearly seen in rows two and four. In the second row, the sources are small, round areas. The regions around them where emitters could be placed are much larger than the actual sources and yet the reconstructions are approximately the same size. In row four, the original map is a manipulated version of the Robotics Institute's logo. An editing error left a horizontal strip of pixels with a non-zero value around the person's midsection between the two vertical areas. This was then expanded when creating the map that determined where emitters would be placed. Despite having this unintended area where emitters were placed, relatively few emitters within that region finished with an activity level that wasn't zero. The map in row six is not very representational of what existed in the actual source map. Despite being a mess of seemingly randomly-placed sources, these sources emit flux fields that closely match the actual flux field. Flux field errors will be examined in detail in later chapters.

Figure 5.6: A comparison of actual (left column) and computed emitters for multiple maps

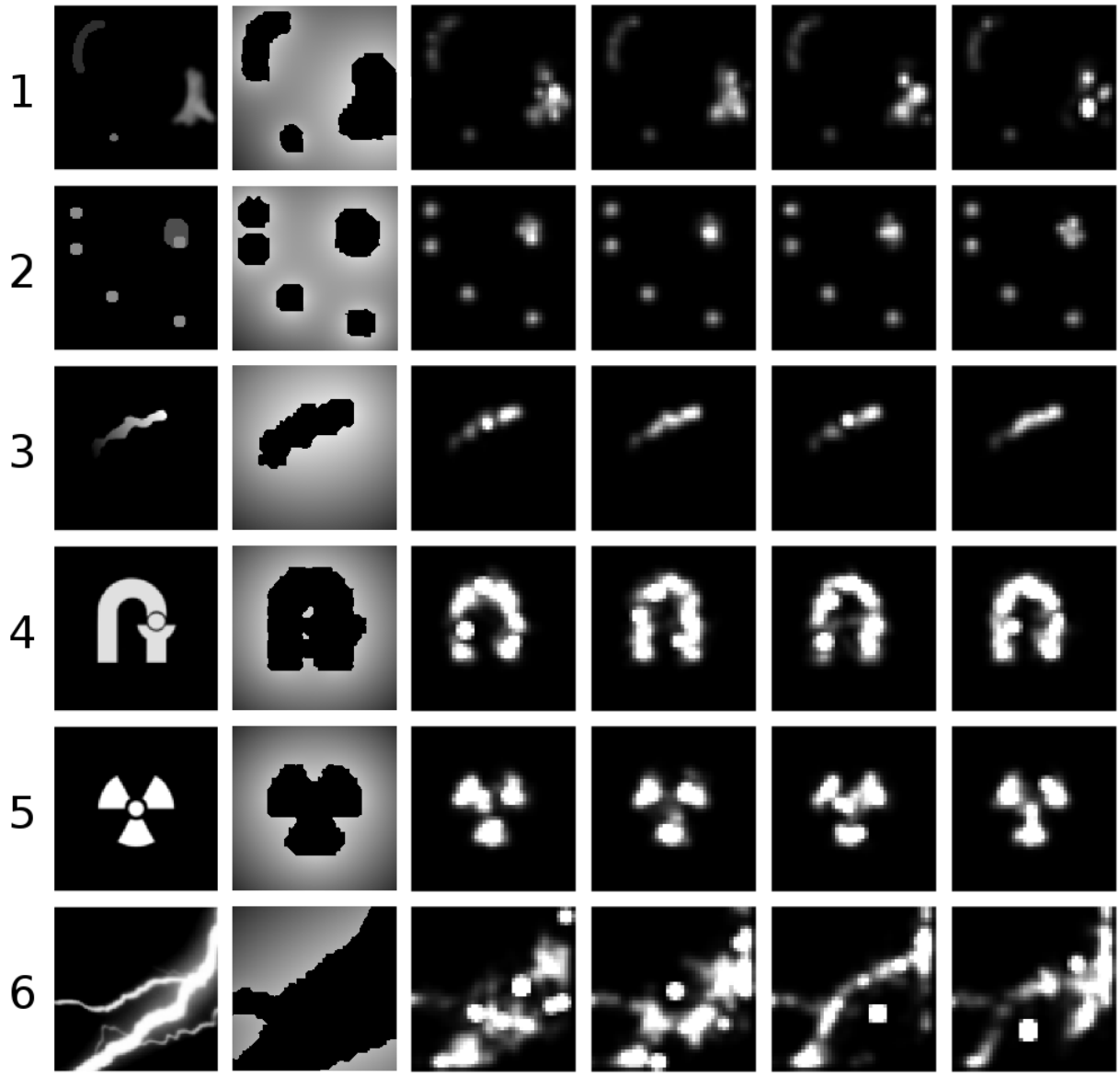


Figure 5.7: The exclusion maps used for the maps in Figure 5.6

Chapter 6

The triangulated gradient method

6.1 Triangulated gradient method overview

The triangulated gradient method is an algorithm that estimates the locations of gamma ray sources from a set of sparse gamma flux measurements. The functionality of this algorithm is described in simple terms starting with Figure 6.1 and then a more detailed description will follow.

In Figure 6.1, sources are seen as grey pluses and measurement locations are shown as black dots. The measurement locations are triangulated as shown in Figure 6.2. For each triangle, the direction of the gradient is found. The gradient directions are represented by arrows in Figure 6.3. Finally, using the gradient direction and the flux values at the vertices of the triangle, a source location can be determined. These estimates are shown in Figure 6.4 as black dots.

Figure 6.1: Map of source and sample locations

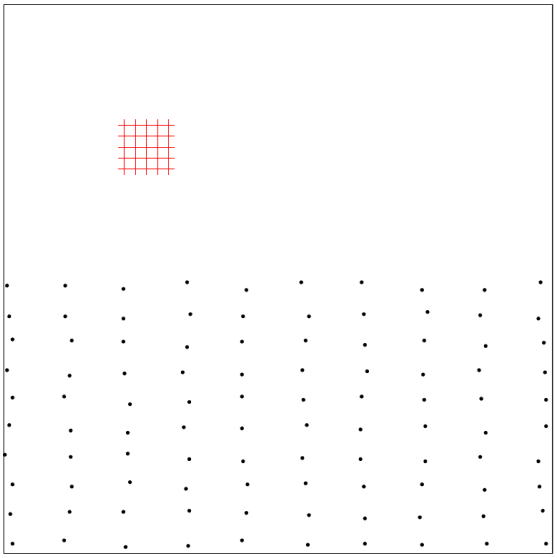


Figure 6.2: Triangulation of the map

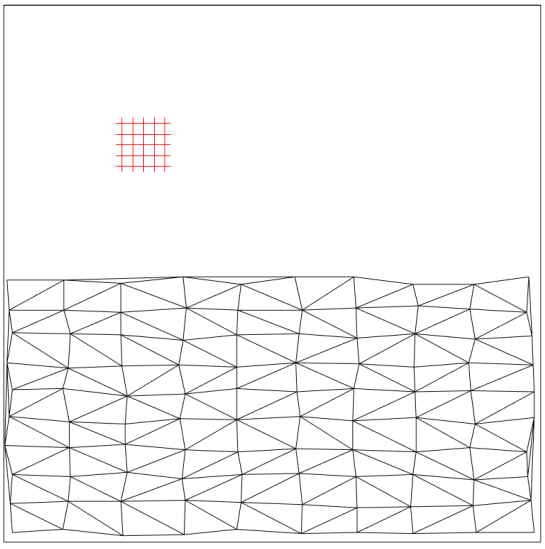


Figure 6.3: Map showing calculated gradient direction for each triangle

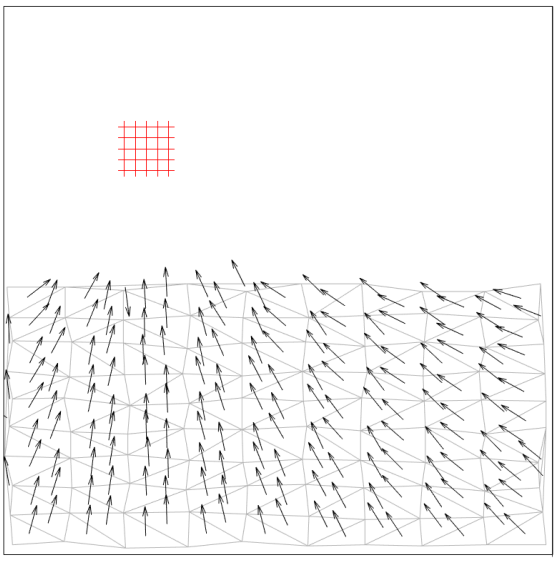
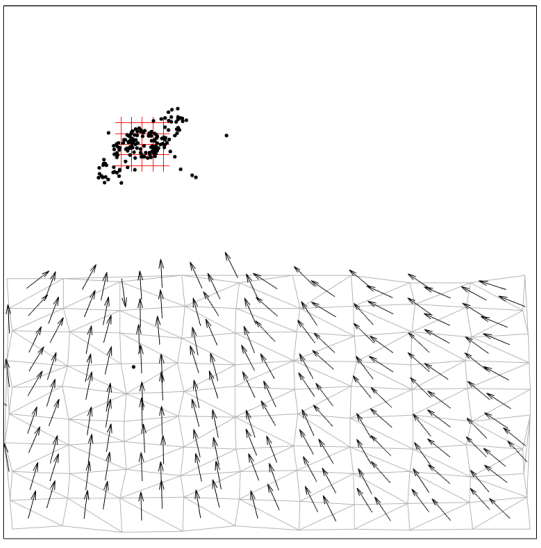


Figure 6.4: Map of estimated source locations



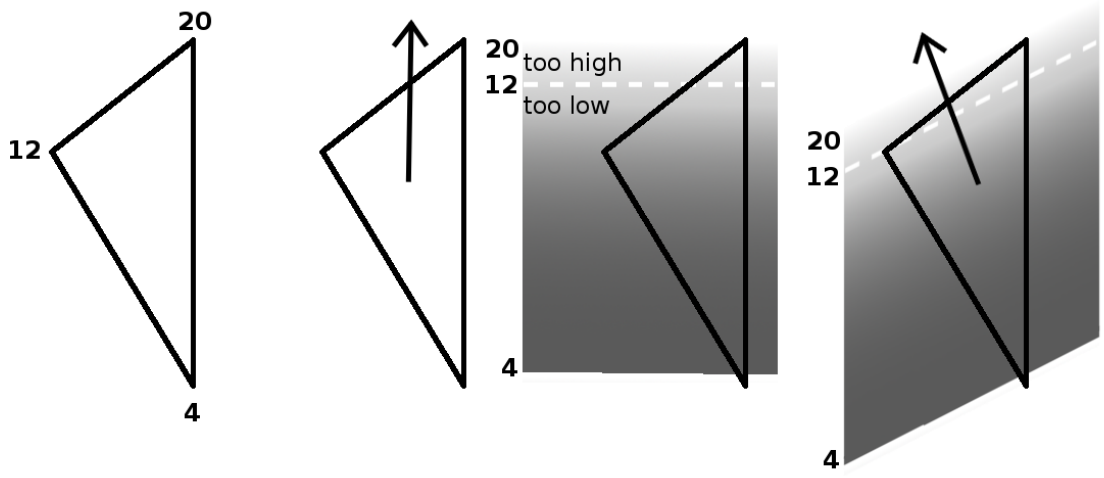
6.2 Finding the gradient direction

An important step in the triangulated gradient method is the process of finding the gradient for each triangle. This is done with an iterative algorithm. Pseudocode for this is shown in Algorithm 4. To start, the two vertices that have the minimum and maximum flux measurements are found. The initial estimate of the gradient angle is set to be the direction from the vertex that has the minimum amount of flux to the one that has the maximum amount. An initial angular step size and direction is set. The initial step angle used for this thesis was negative 45 degrees. The estimated flux at the middle vertex that was nether a minimum nor maximum is calculated based on the current estimate of the gradient direction. Finally an initial error amount is stored. This error is the difference between the estimated and measured flux at the middle vertex.

The iterative section starts now and repeats until the magnitude of the angular step size is below a predefined threshold. The current gradient direction estimate is adjusted by the current step angle. It then calculates the predicted flux at the middle vertex and finds the error between this value and the measured value. If the sign of this error is different from the sign of the last error it halves the angular step size and reverses the step direction. If the sign for both errors was the same it does nothing. It then continues to the next iteration.

Figure 6.5 shows this process pictorially. The triangle on the left shows a triangle with hypothetical flux readings at its vertices. A unit vector is created in the direction from minimum to maximum as shown in the second triangle. This becomes the current guess of the gradient direction. The third triangle gives a visual example of dense flux based on the initial gradient direction. The dotted line shows where in this gradient the actual flux value for the middle vertex is. This algorithm iteratively changes the gradient direction until the estimated flux at the middle vertex matches the measured flux as shown by the triangle on the right.

Figure 6.5: Steps towards finding the gradient direction



Algorithm 4 Pseudocode for iteratively finding the gradient direction

$sortedVertices \leftarrow sortVerticesMinToMaxBasedOnFlux(vertices)$
$minVertex \leftarrow sortedVertices(1).position$
$maxVertex \leftarrow sortedVertices(3).position$
$currentDirection \leftarrow vectorToAngle(maxVertex - minVertex)$
$normal \leftarrow angleToUnitVector(currentDirection)$
$stepAngle \leftarrow -initialStepSize$ ▷ -45 degrees
$middleFlux \leftarrow sortedVertices(2).flux$
$computedFlux \leftarrow findFluxAtMiddleVertex(normal, vertices)$
$lastError \leftarrow middleFlux - computedFlux$
while $abs(stepAngle) < finalStepSize$ **do**
    $currentDirection \leftarrow currentDirection + stepAngle$
    $normal \leftarrow angleToUnitVector(currentDirection)$
    $computedFlux \leftarrow findFluxAtMiddleVertex(normal, vertices)$
    $thisError \leftarrow middleFlux - computedFlux$
    if sign(thisError) != sign(lastError) **then**
        $stepAngle \leftarrow stepAngle * -1$
    end if
end while

6.3 Triangulated gradient method algorithm

Pseudocode for the triangulated gradient method is shown in Algorithm 5.

Algorithm 5 Pseudocode for the triangulated gradient method

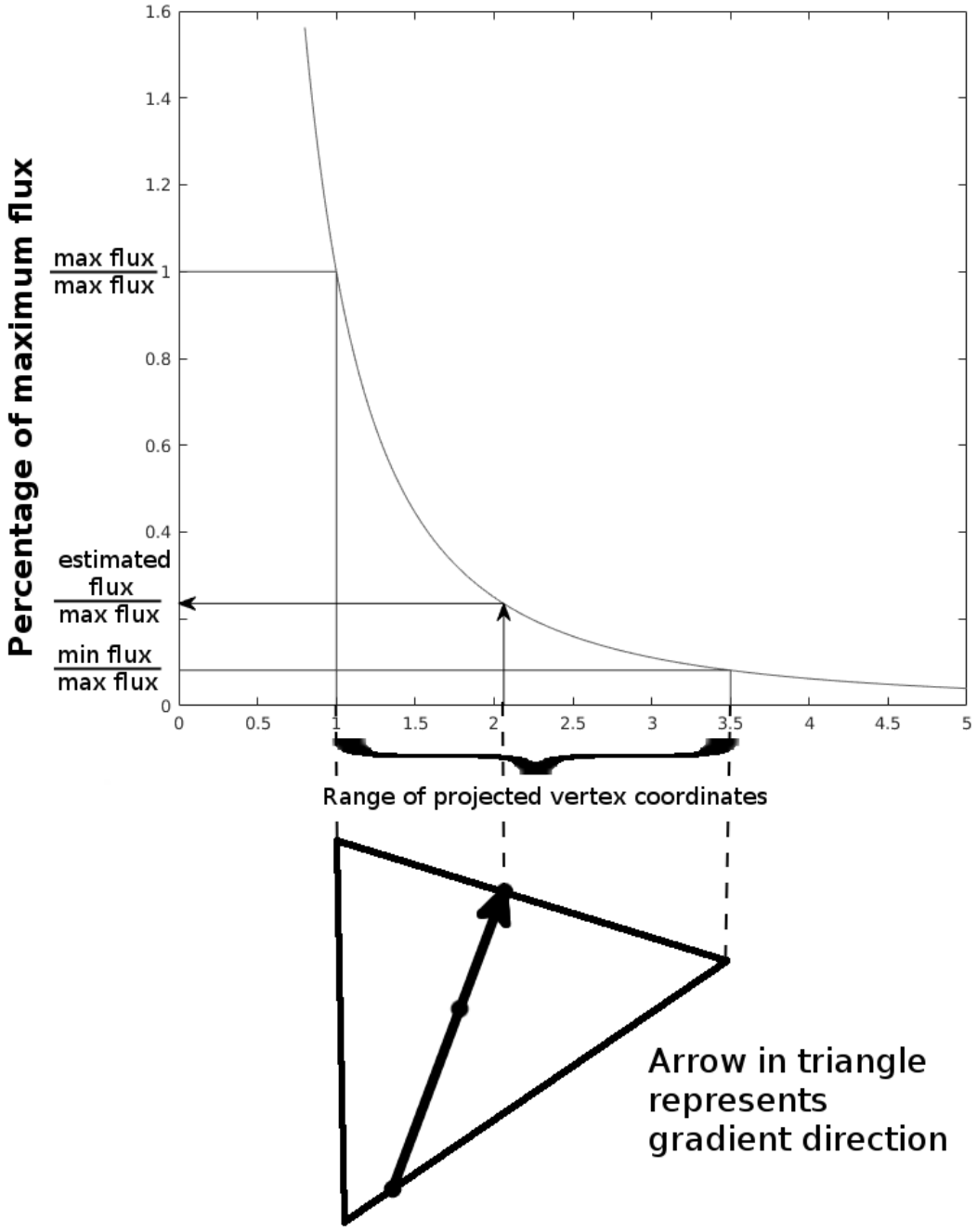
$triangles \leftarrow delauneyTriangulation(sampleLocations)$
for all triangles **do**
$gradientDirection \leftarrow findGradientDirection(triangle)$
$intersections \leftarrow findInstersections(gradient, triangle)$
$fluxes \leftarrow findFluxAtIntersections(triangle, intersections)$
$intersectionDistance \leftarrow euclideanDistance(intersections)$
$percentRemaining \leftarrow \frac{fluxes.min}{fluxes.max}$
$X \leftarrow \sqrt{\frac{1}{percentRemaining}}$
$Xrange \leftarrow X - 1$
$distanceToPointSource \leftarrow \frac{intersectionDistance}{Xrange}$
end for

The initial triangulation of the measurement locations is performed using Delauney triangulation. The gradient direction is then found using the algorithm described previously. A line is constructed which passes through the centroid of the triangle in the direction of the gradient. The two intersections between the line and the triangle are found. At each intersection the flux is estimated. This is done using inverse square interpolation which is calculated as follows. For each side where there is an intersection, the flux measurements at its endpoints are sorted. The lower flux measurement is divided by the higher flux measurement which results in a value between zero and one. The adjusted lower flux is then used to find an X coordinate using an inverse square curve. Formula 6.1 shows how this X value is found. The larger flux will correspond to an X value of one. The range of X values between one and the X value that corresponds to the lower amount of flux represents the distance between the vertices of the edge of the triangle that is being intersected. Figure 6.6 shows this relationship. The distance between the intersection point and the vertex with the larger flux is scaled into the X value range that was just found. This scaled intersection point is used to find an estimated flux value using Formula 6.2.

$$X = \sqrt{\frac{1}{Y}} \tag{6.1}$$

$$flux = \frac{maxflux}{X^2} \tag{6.2}$$

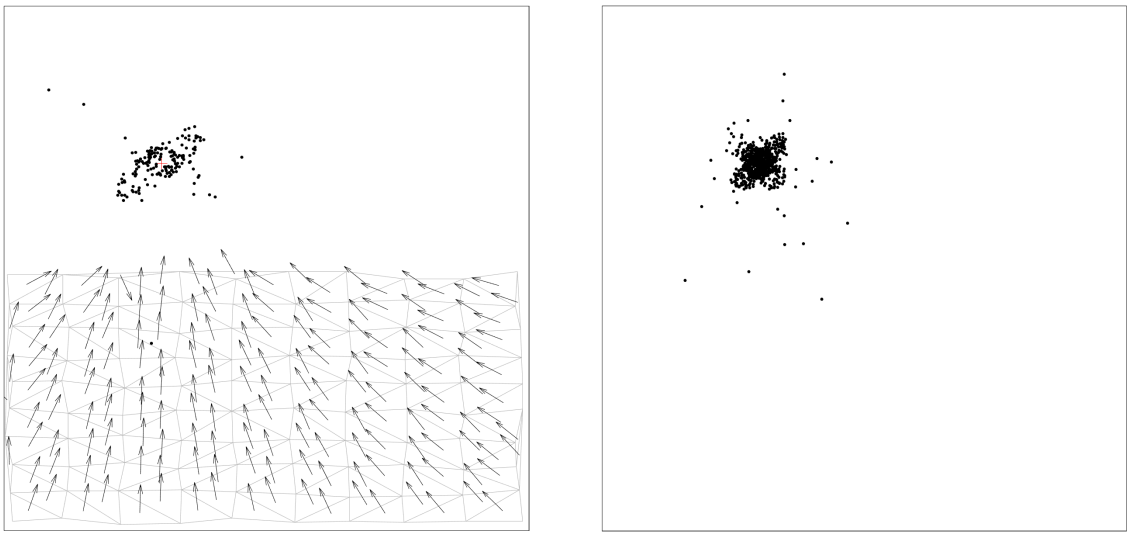
Figure 6.6: Graphic showing how estimated flux is found



6.4 Results

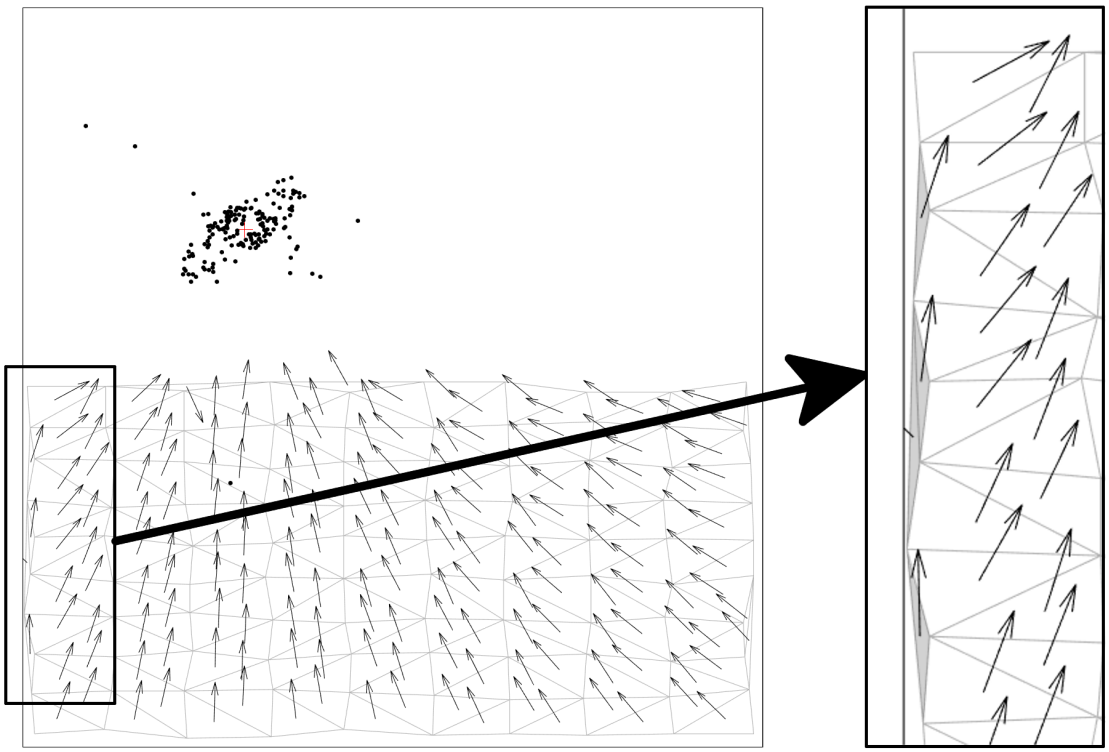
The triangulated gradient method is meant to give an estimate of source locations. Previously, estimates were shown for a small area source that was located outside of the sampled region. Figure 6.7 shows the results of estimating the location of a single point source emitter. The image on the left shows the estimates when measurements are only taken in the bottom half. The image on the right shows estimates when measurements are taken everywhere within the map. The results for a point source look very similar to the results for the small area source. This is due to most triangles being relatively distant from the sources. As the distance between sources and triangles increases, the flux from area sources combine to appear as a single, more active source.

Figure 6.7: Estimated locations of a point source



One further observation from this example is of the triangles on the edges. Thin triangles tend to not give good estimates and should be eliminated. Figure 6.8 shows an expanded view of the edge. Triangles that give bad estimates are highlighted in grey.

Figure 6.8: Bad source estimates from edge triangles



The triangulated gradient method doesn't always provide estimates like the previous examples. In situations where there are multiple sources, estimates will tend to connect the two sources. This can be seen in Figure 6.9. This figure shows two area sources which are drawn with grey pluses. Estimates are shown with black dots. The estimates on the left use measurements that were taken on the bottom half of the map only. The estimates on the right use measurements from everywhere within the map. Taking measurements from everywhere places many more estimates on the actual source locations. There are a number of estimates connecting the two source locations. This can be understood by looking at a map that shows the flux. Figure 6.10 shows the same map with the flux drawn as zebra stripes rather than a gradient. The edges of each stripe represent areas with equal amounts of flux. Triangles that are equally distant from both sources have gradient directions that point directly between the two sources. Triangles that are closer to one source than the other will give better estimates of where the closer source is located. Because of this, it is important to take measurements as closely as possible to actual sources. This suggests measurement strategies where a robot might initially take measurements with wide sample spacing to get a general overview of source locations then navigate towards estimated locations to refine its estimates. There are some stray estimates in the image on the right of Figure 6.9. These are due to triangles that overlapped the two sources. When that happens, the estimates are wildly inaccurate so triangles should be removed if it is suspected that they may contain sources. One way to automate this it to remove estimates from triangles that in turn have estimated sources within their boundaries.

Figure 6.9: Estimated locations of two area sources

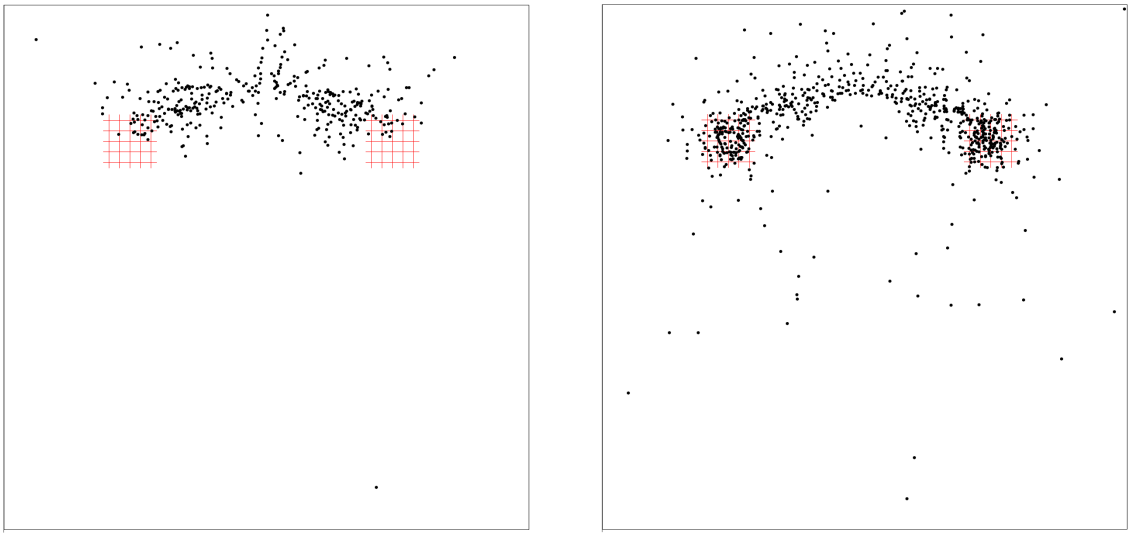


Figure 6.10: Map with two sources and flux lines

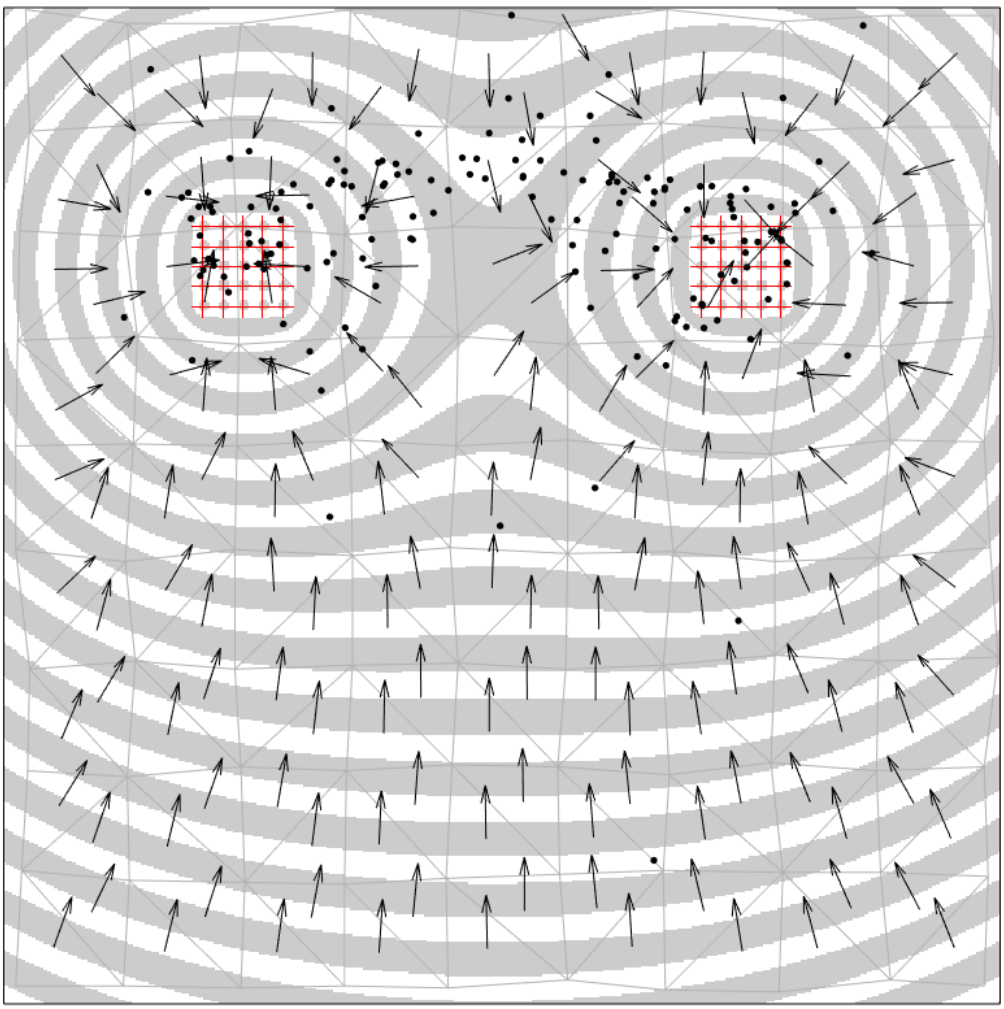
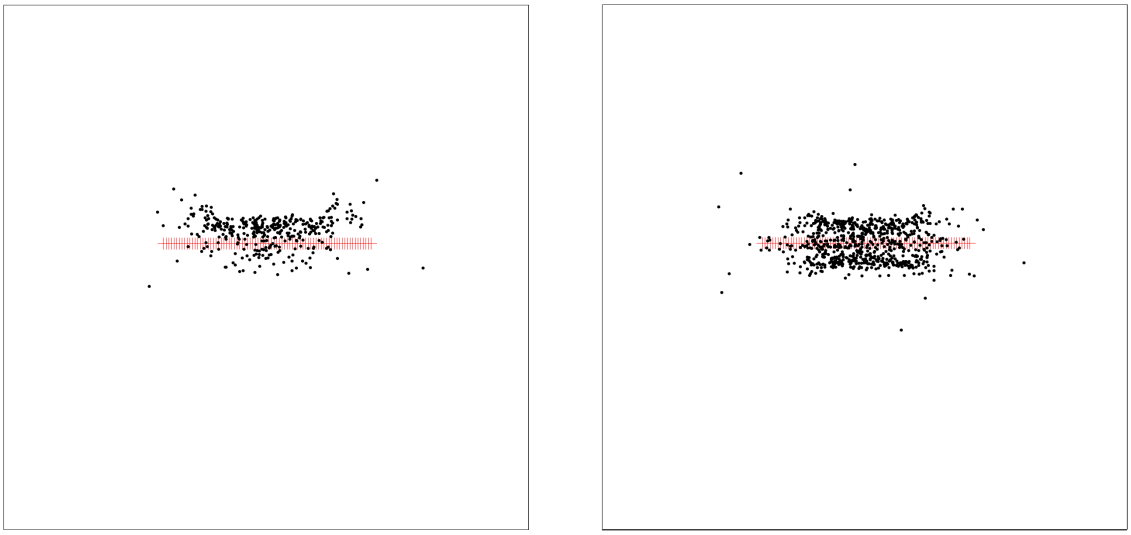


Figure 6.11 shows two maps of estimates from linear sources. On the left are estimates from measurements that were only taken in the bottom half of the map. On the right are estimates from measurements that were taken everywhere in the map. There are two things to note in these maps. The first is that the end points of the linear source are not well represented. This will be the case for any thin source or sharp corners for area sources. This is because the estimates tend towards the center of mass of a source. The second thing to note is that many of the estimates overestimated the distance to the source. In the left map, these estimates are above the linear source since measurements were only taken on the bottom half. In the right map, triangles on both sides of the linear source overestimated, creating lines of estimates on either side of the actual source location. The reason for the overestimates is because an inverse square falloff with respect to distance is used when estimating. With a linear source, if it is long enough, the falloff is the inverse of the distance. This leads to errors in estimation.

Figure 6.11: Map with a linear source



One final situation that produces potentially unexpected results is shown in Figure 6.12. This figure has two representations of a map that has a large rectangular area source. Measurements were taken from everywhere within the map. The majority of the estimates lie towards the center of the source in a relatively linear manner. The map on the right shows the flux generated by this source as well as the estimated gradient directions for each triangle. The flux lines that are close to the source are nearly linear causing a linear pattern of estimates. The flux lines near corners have a lot of curvature and cause estimates to radiate from the corner towards the central line of estimates Assuming that the source has a uniform level of activity throughout, when the number of measurements are taken to an extreme, the result is that the estimates cluster into a shape that resembles the lines of a Voronoi diagram that has seeds along the perimeter of the area source. Figure 6.13 shows this relationship.

Figure 6.12: Map with a large area source

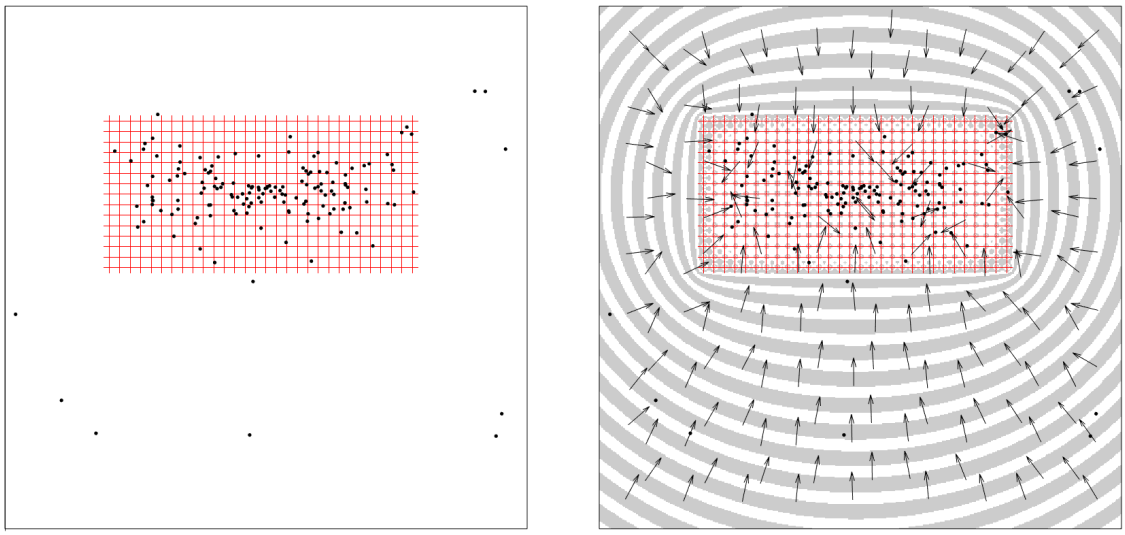
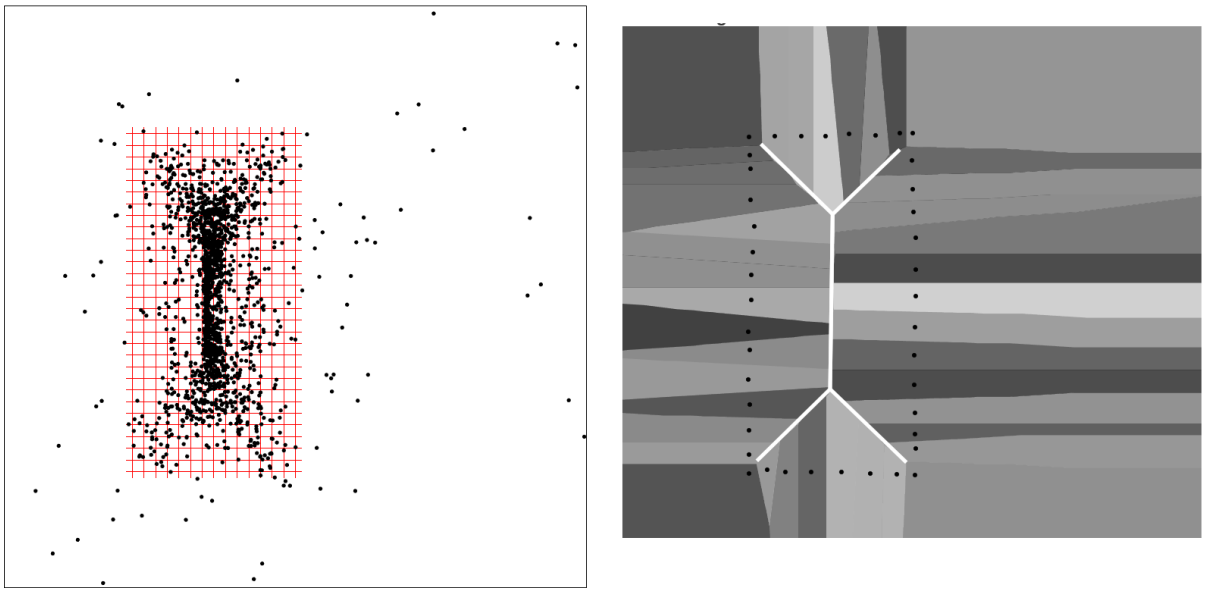


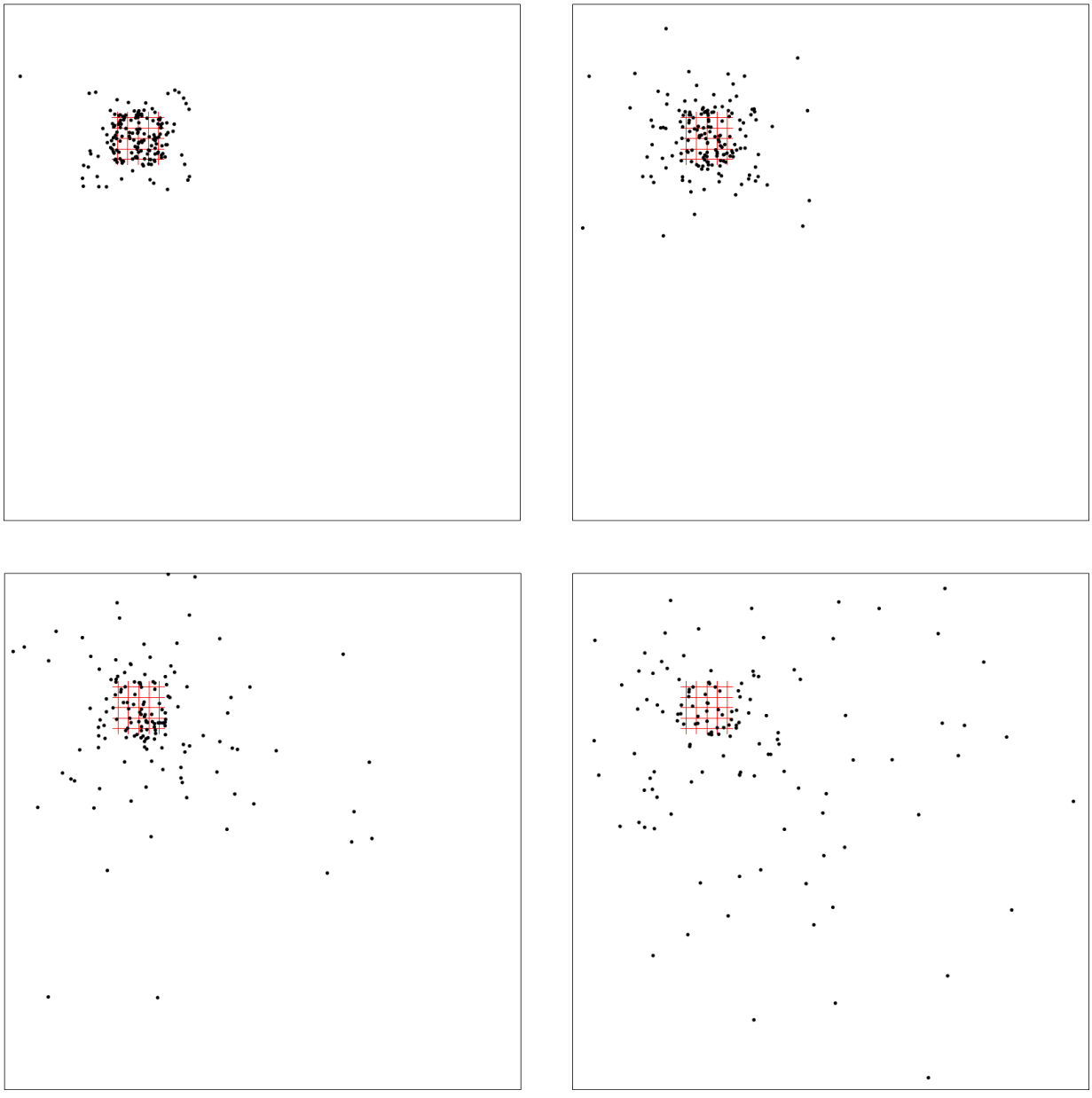
Figure 6.13: Source estimates compared with a Voronoi diagram



6.5 Shot noise

So far every example has used measurements that were perfect. Actual gamma flux measurements have shot noise because gamma rays are emitted one at a time rather than in a continuous manner. Figure 6.14 shows how this algorithm holds up when noise is present. Each map in this figure has a small area source in its top left corner. Source location estimates are shown as black dots. The top left map shows results with perfect data. The top right map has a similar distribution but this comes with a somewhat heavy cost. The median number of counts at each measurement location in this map is 3,044. It isn't practical to take such long measurements at every sample location. It may be an acceptable strategy for a robot to periodically take a set of three longer measurements in an equilateral triangle to help navigate to better locations where it will take shorter measurements. The bottom left map shows the results if measurements have an order of magnitude fewer counts per measurement. In that map, the median number of counts is 297. Moving one more order of magnitude downward, for map on the bottom right the median number of counts per measurement is 30. While the bottom right map still shows a more dense concentration of estimates around the actual source location, it is likely that in practice measurements with so few counts wouldn't produce useable results.

Figure 6.14: How shot noise affects the triangulated gradient method

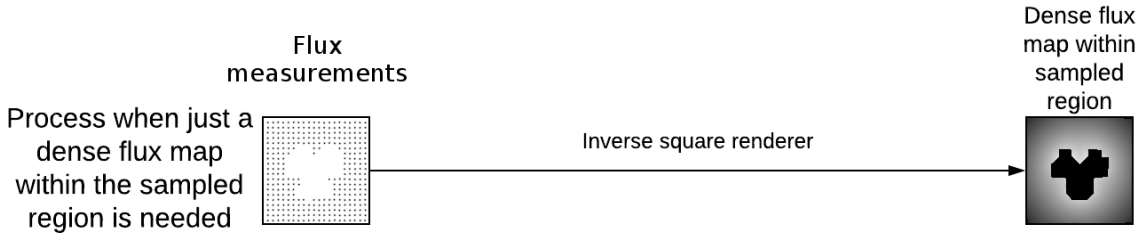


Chapter 7

The inverse square renderer workflow

The first workflow that was described in Chapter 3 uses the inverse square renderer to take a set of sparse gamma flux measurements and directly output a dense gamma flux map for all areas within the convex hull of the sampled region. The inverse square renderer is based on the triangulated gradient method. The only difference is that instead of estimating a source location for each triangle, the inverse square renderer uses the estimated gradient direction and the flux at the vertices to fill the triangle with dense flux values. Since this workflow is based on the triangulated gradient method, it shares all of the strengths and weaknesses previously described for that algorithm.

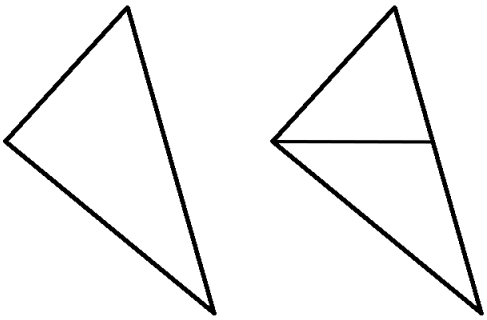
Figure 7.1: The inverse square renderer workflow



The initial steps for the inverse square renderer are exactly the same as the triangulated gradient method through finding the gradient direction. At this point it can now start rendering the triangle. Directly rendering a general triangle isn't easily done. Each triangle is split by dividing it into top and bottom halves at the vertex that is in the middle vertically. See Figure 7.2 for reference. This leaves two triangles with a horizontal edge.

Splitting the triangle makes it easy to figure out the coordinates for the first and last pixels in each row of pixels that must be rendered. Before starting to render, the vertices of the original triangle with the minimum and maximum flux are projected onto the gradient vector and the resulting coordinates are saved for later. The triangulated gradient method projected the vertex with the middle flux value onto the gradient vector during each iteration in order to estimate the flux at that vertex. The same process is used in the inverse square renderer when calculating the dense flux values. The coordinates of each pixel being rendered are projected onto the gradient

Figure 7.2: Original triangle before split (left) and after (right)

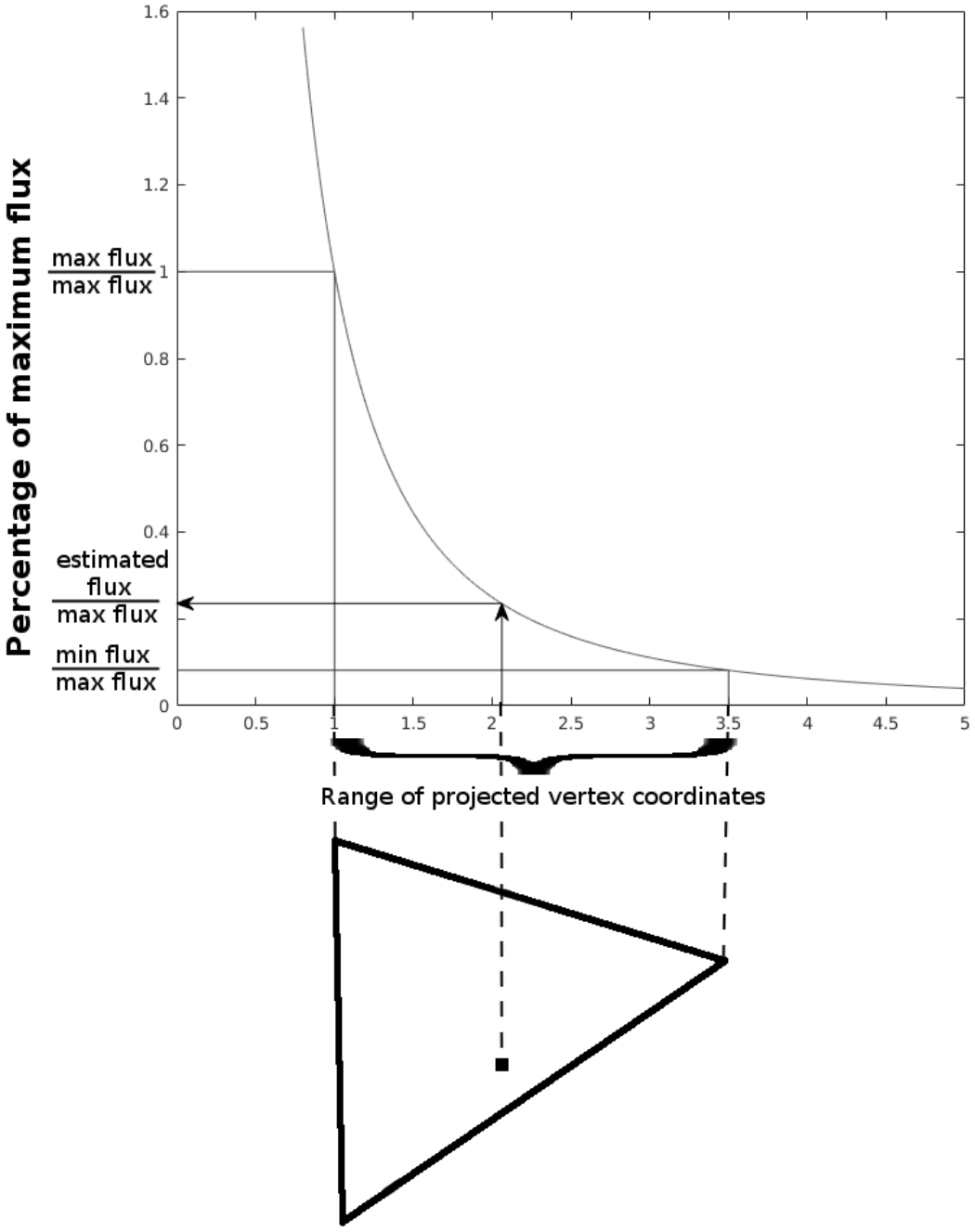


vector which results in a coordinate along the gradient. This coordinate lies somewhere between the two coordinates that were saved previously. By scaling the pixel's coordinate so that it fits into the range shown at the bottom of Figure 7.3, a final flux value can be found using Equation 7.1. Algorithm 6 shows pseudocode for the inverse square renderer.

Figure 6.6 provides a visual reference for the next part. The plot within the graph is of Equation 7.1.

$$flux = maximum flux / X^2 \tag{7.1}$$

Figure 7.3: Graphic showing how the value for a pixel is found



Algorithm 6 Pseudocode for rendering a dense flux map

$triangles = triangulate(sampleLocations)$
for all $triangles$ **do**
  $minFlux \leftarrow min(vertex1flux, vertex2flux, vertex3flux)$
  $maxFlux \leftarrow max(vertex1flux, vertex2flux, vertex3flux)$
  $percentRemaining \leftarrow \frac{minFlux}{maxFlux}$
  $minX \leftarrow \sqrt{\frac{1}{percentRemaining}}$
  for all vertices in triangle **do**
    $projectVertex()$ ▷ Project vertex onto gradient vector
  end for
  $minVal \leftarrow min(projectedVertices)$
  $maxVal \leftarrow max(projectedVertices)$
  $splitTriangle()$ ▷ Split the triangle into two, each having a horizontal side
  for all split triangles **do**
    for all rows of pixels in the triangle **do**
      $calculateFirstAndLastPixelWithinTriangle()$
      for all pixels in each row **do**
        $projectedCoord \leftarrow projectPixel()$ ▷ Project pixel onto gradient vector
        $percentBetween \leftarrow findPercent(minVal, maxVal, projectedCoord)$
        $X \leftarrow ((minX - 1) * percentBetween) + 1$ ▷ Scale into range (1,minX)
        $Y \leftarrow \frac{1}{X^2}$
        $PixelValue \leftarrow maxFlux * Y$
      end for
    end for
  end for
end for

7.1 Results

Figure 7.4 shows a flux map that was caused by a single point source emitter. The top left image is the actual flux map. Measurements were taken of this flux in a 12x12 grid. These measurements were then passed to the inverse square renderer which generated the flux map on the bottom left. The image on the top right is a map of the percent error in flux between the actual and rendered flux maps. The image on the bottom right is a histogram of the percent error in flux for this map. In the maps shown here, triangles that contain sources are still rendered. In actual use, there are likely to be areas where measurements were not taken around the actual source locations. In that situation, it is recommended that triangles that span the unmeasured area be removed because they will have high amounts of error. Note that along the bottom and right of the percent error map there are thin sections with high error. This is due to the edges having very thin triangles like what was seen in Figure 6.8 in the triangulated gradient method chapter. If this algorithm is to be used by a mobile robot which will make decisions base on these maps, it is recommended that these triangles be removed.

Figure 7.4: Map of point source with 12x12 measurement grid

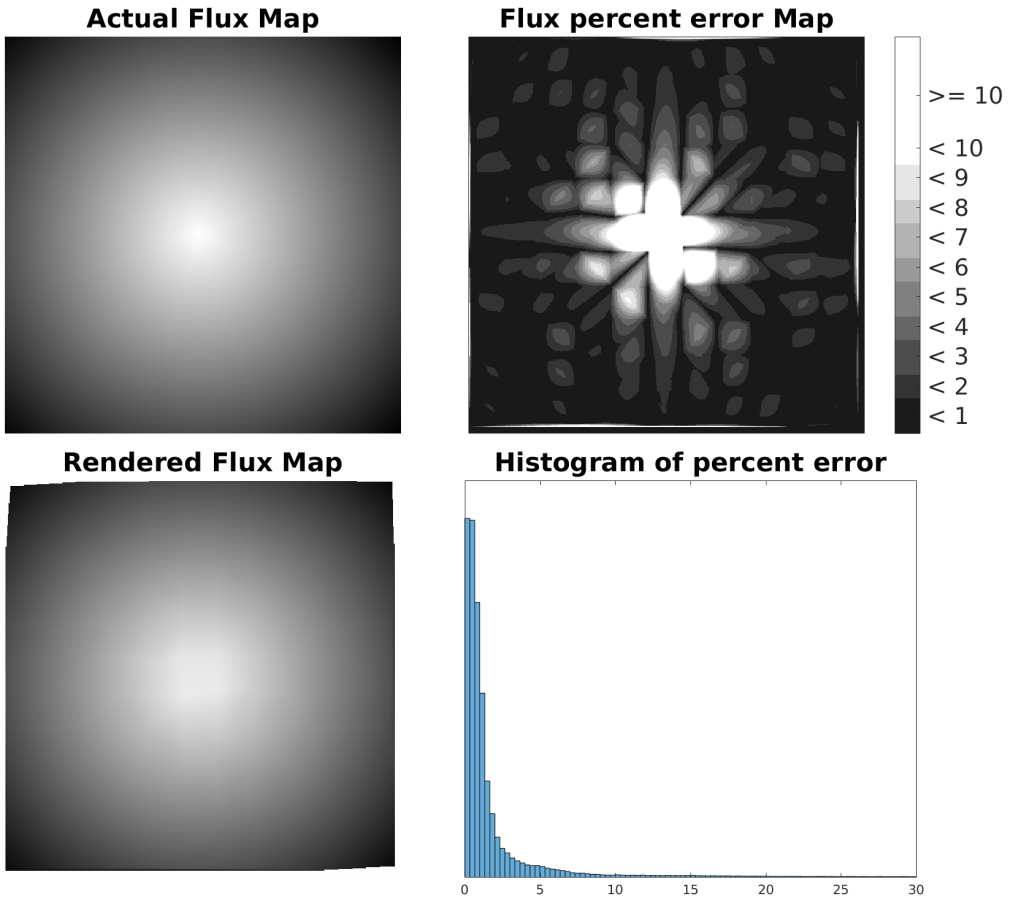


Figure 7.5 shows the same map where measurements were taken in a 7x7 grid. Note that the error is much higher in this map. Larger sample spacing will lead to higher amounts of error. This is particularly true along triangle edges that are perpendicular to the gradient direction. Figure 7.6 helps show why this happens. In this figure there is a point source and a triangle filled with a representation of the actual flux that would be caused by the point source. The inverse square renderer uses a linear gradient direction instead of the radial gradient that a point source causes. The three corners of the triangle will be rendered with correct flux values. This is why the sample locations in the maps all have no error and are drawn in black. The center of the vertical edge in Figure 7.6 will have a large amount of error because it will be rendered with the same amount of flux as the corners nearest to the point source due to the linear gradient direction. This value will be too low. The dotted line in the figure shows the locations in the triangle that share the same flux values as the near corners.

Figure 7.5: Map of point source with 7x7 measurement grid

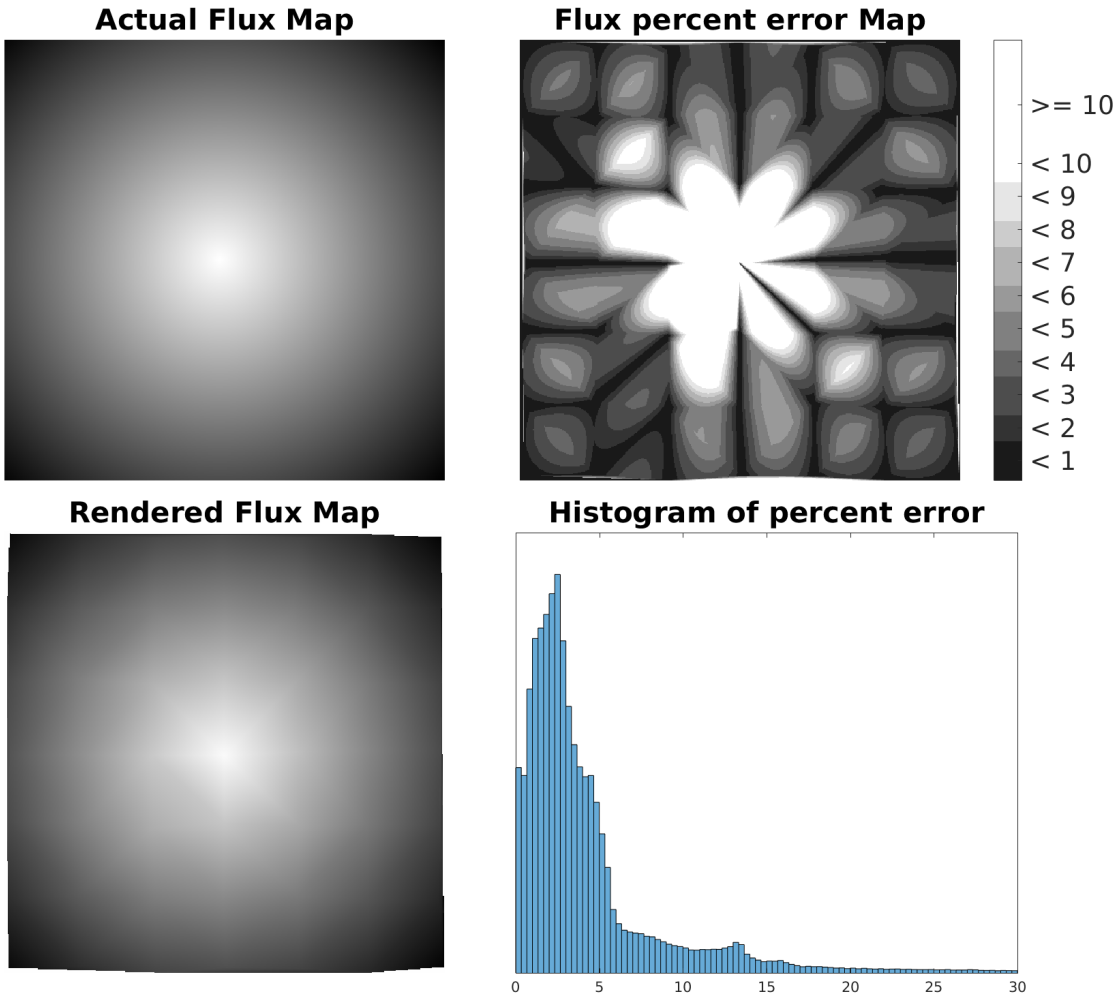


Figure 7.6: Example triangle showing why interpolation can fail along some edges

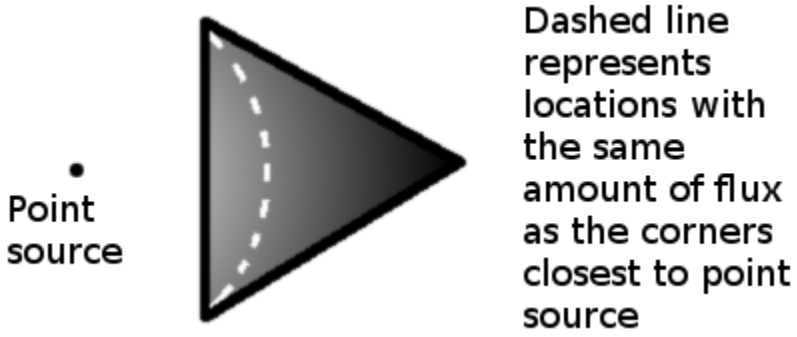


Figure 7.7 shows a map with a large rectangular area source. This map has less error than the point source but that may be due to the source's shape aligning nicely with the spacing of the measurements. Note that although there are large sections of this map that have larger than ten percent error, these are only within the boundaries of the source. The histogram doesn't have a spike for these areas because they are distributed fairly evenly over a large range of values.

Figure 7.7: Map of a large area source

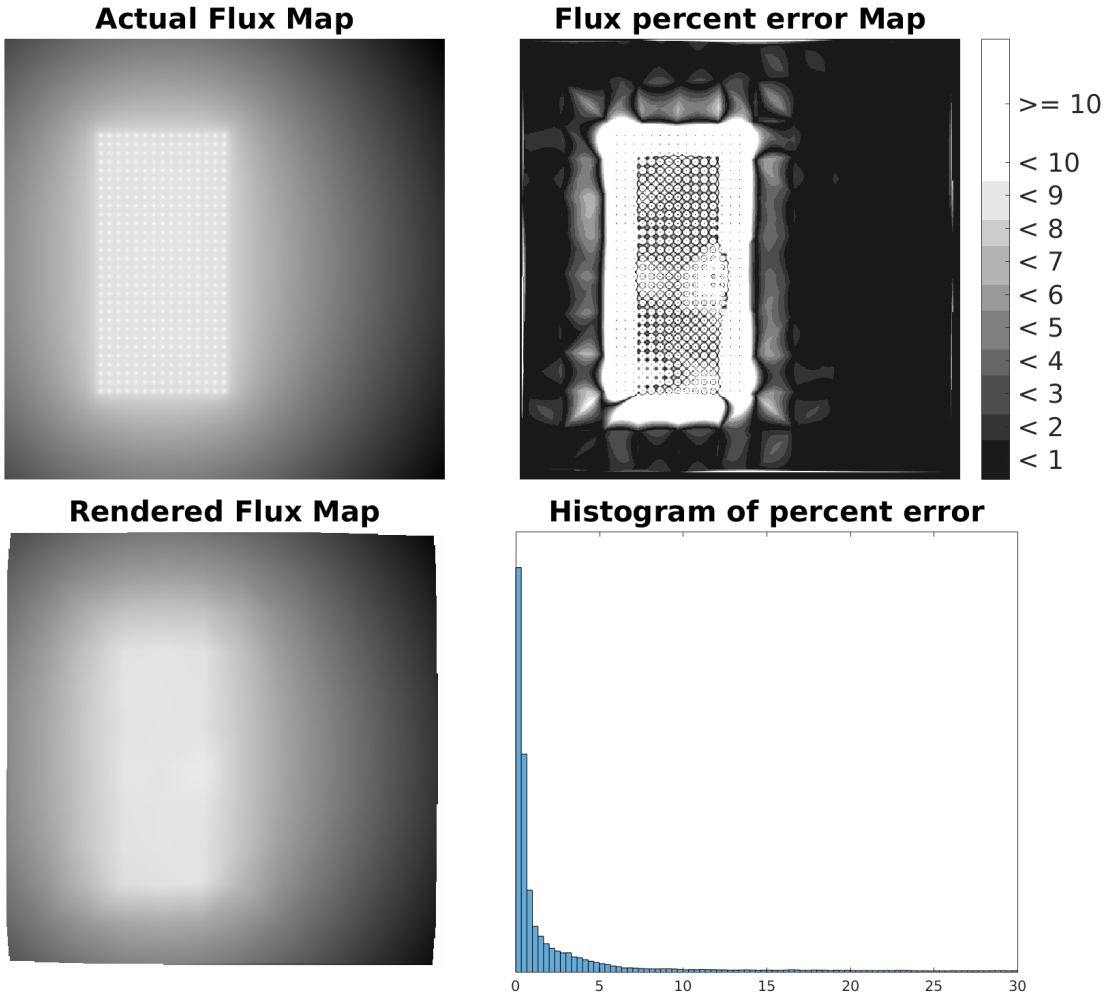
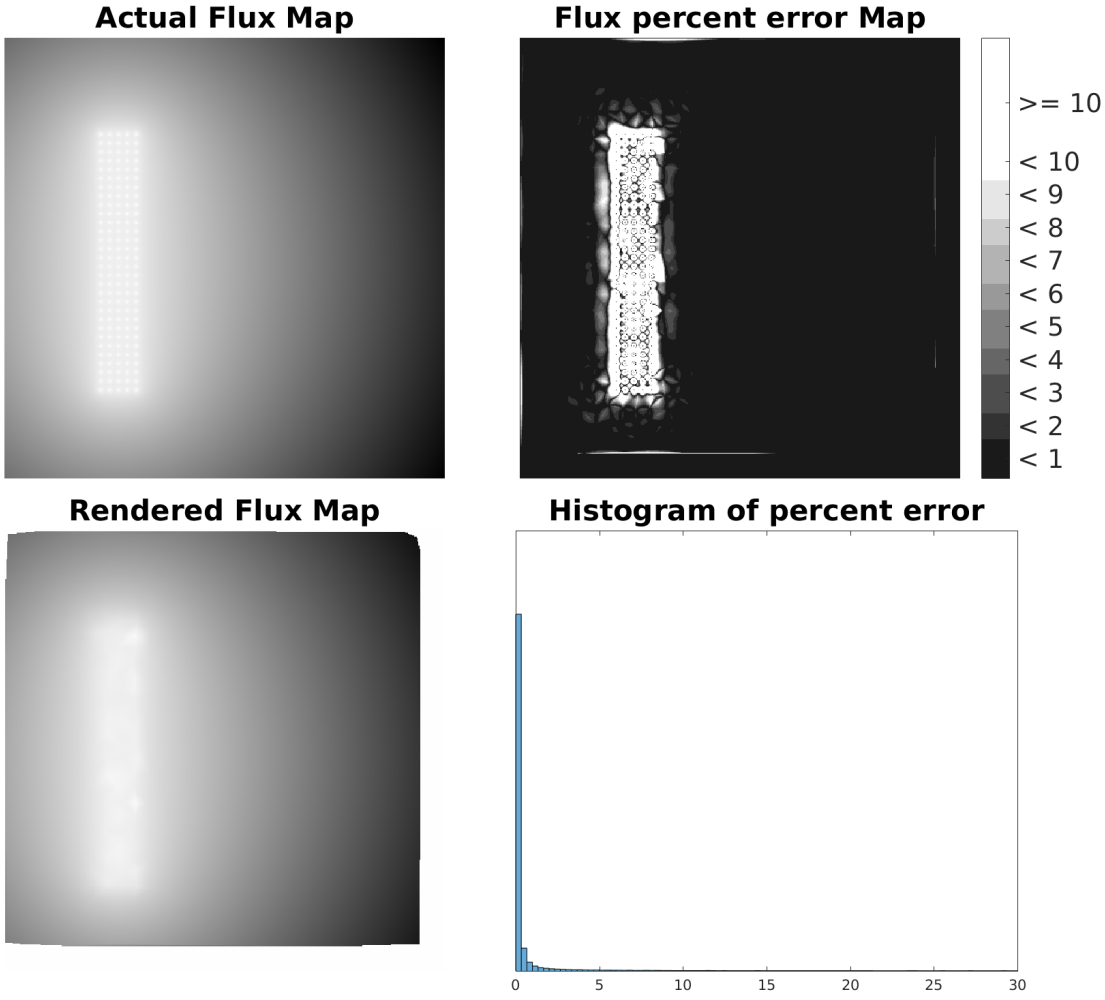


Figure 7.8 shows another area source. Measurements were taken from this map with little space between them. This results in a map that has very low amounts of error even when triangles are close to the source.

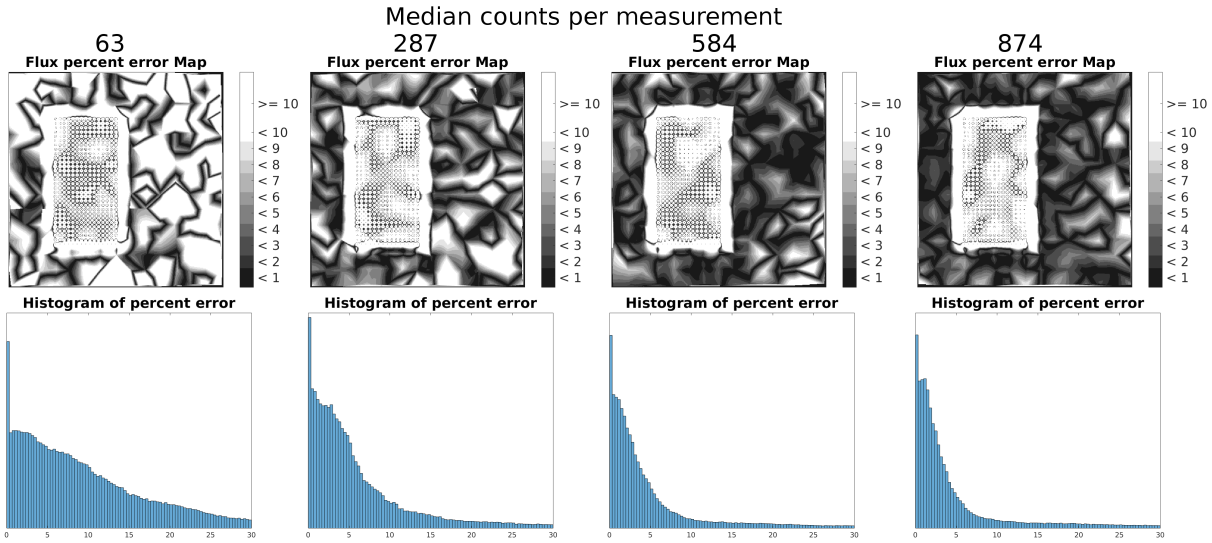
Figure 7.8: Map of a thin area source



7.2 Noisy measurements

All previous maps have had perfect measurements. Actual gamma flux measurements have shot noise because gamma rays are emitted one at a time from a radioactive source instead of continuously. Figure 7.9 shows four maps where measurements were taken for varying amounts of time. This causes the median number of gamma rays that have been detected to vary. These median counts are listed above each map. While some difference can be seen in the maps themselves, the difference can be more clearly seen in the histograms. The longer that measurements can be taken, the better the results of this algorithm will be.

Figure 7.9: A comparison of results with different measurement durations



7.3 Suggested measurement strategies

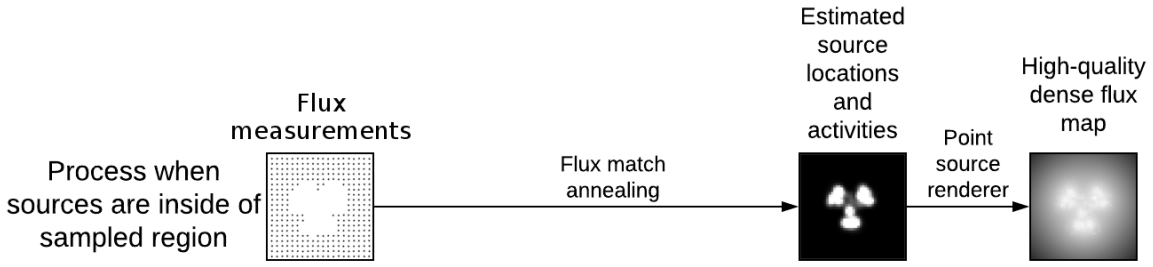
The main source of error within a triangle is caused by the direction of the gradient changing. Small triangles reduce how much the gradient direction can change across a triangle. The gradient direction farther away from a source changes less quickly thus triangles that are far from a source have less error and could be larger. Because sometimes it is only required that error stay lower than some threshold, a suggested measurement strategy is to take measurements more frequently in areas where the flux is high and less frequently in areas where there is less flux. A way to check if measurements are being taken often enough is to have the robot which is taking measurements navigate to the center of a triangle. It will then take one more measurement. If that measurement is within a prespecified tolerance of the value that the inverse square renderer predicts then measurements are being taken often enough in that area.

Chapter 8

The flux match annealing and point source renderer workflow

The second workflow uses the flux match annealing algorithm and point source renderer to take a set of sparse gamma flux measurements and emitter locations and outputs a dense gamma flux map as well as a set of emitters that are optimized to produce flux that closely matches the provided measurements. Figure 8.1 shows this workflow. Measurements were taken wherever a dot is located in the map on the left.

Figure 8.1: The second suggested workflow



This workflow is intended for situations where sources are located within the convex hull of the sample locations and operates on the assumption that samples were not taken directly over an actual source. The areas where there are no samples are filled with a grid of emitters that initially have an activity of zero. These emitters and the flux measurements are provided to the flux match annealing algorithm which adjusts their activity levels to produce flux that closely matches the provided flux measurements. The optimized emitters are then passed to the point source renderer which outputs a dense gamma flux map that can be extrapolated beyond the surveyed region. This workflow provides a higher-quality flux map when compared to the inverse square renderer workflow and is much more tolerant of the shot noise inherent in gamma flux measurements. Drawbacks to this workflow are that the flux match annealing algorithm is computationally expensive and it does not provide good estimates of source locations when the sources are not surrounded by sample locations. While this workflow can match the flux radiated

by sources which are outside the sampled region, it tends to be an inefficient process due to how many emitters must be used. Additionally, the flux match annealing algorithm requires a much higher number of iterations to match the provided flux measurements as closely as a situation where the source was completely surrounded by measurement locations.

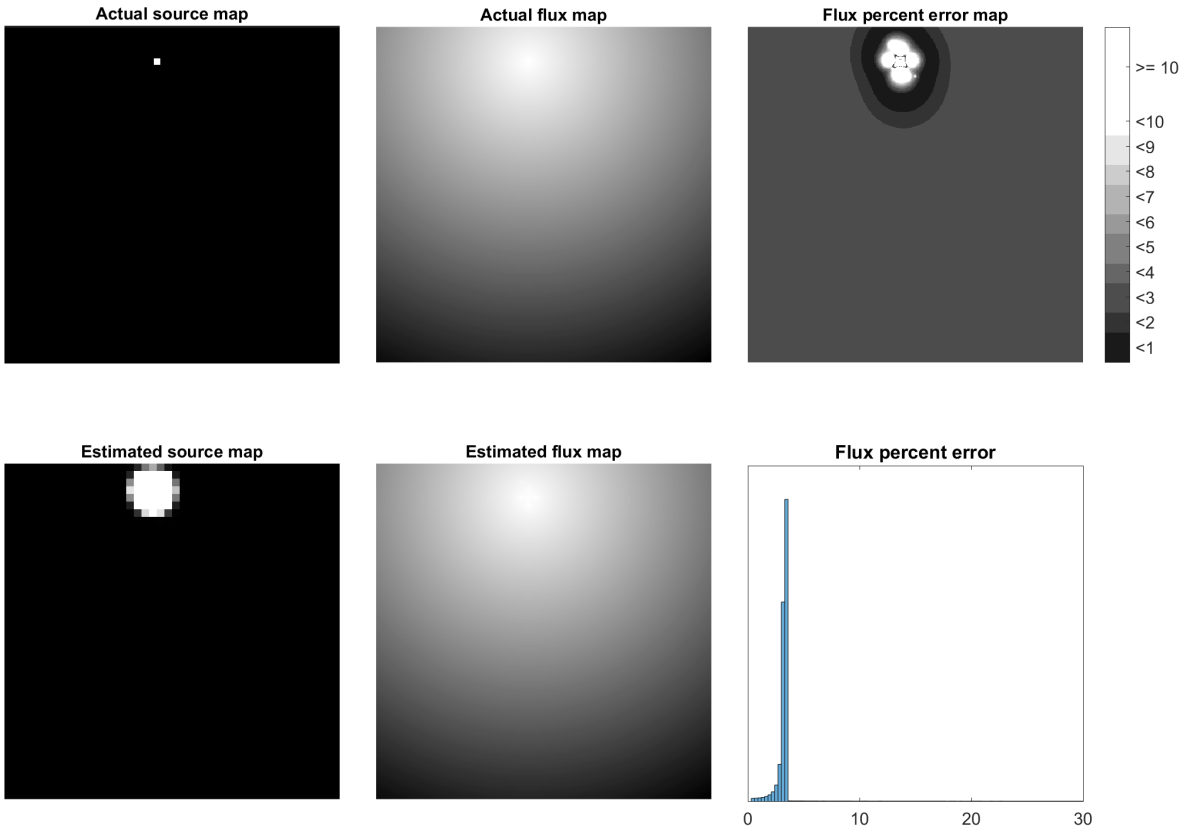
8.1 Results with different source maps

8.1.1 Map with single point source

Figure 8.2 shows results from a source map which has a single point source. The image on the top left shows the actual source map. In source maps, white pixels represent locations with high activities and black pixels represent areas with no activity. Varying levels of gray represent varying levels of activity. The image on the bottom left shows an estimated source map. Note that the algorithm that drew the estimated source map was written to draw activities from emitters that weren't necessarily in a grid pattern. It uses a predetermined grid spacing and shows the sum of the activities of all emitters within each grid location. Because of this, estimated source maps look a bit different in shape when compared to estimated flux maps which show the actual emitter locations. The estimated source maps should be used as a general reference for what areas in a map are active. The image at the top center shows the actual gamma ray flux produced by the source in the source map. Below it is an image which shows the gamma ray flux produced by the emitters in the estimated source map. On the right at the top is a map that shows the percent error between the two gamma ray flux maps. Finally, on the bottom right is a histogram of the percent error seen within the map.

The flux match annealing algorithm has difficulty matching the flux from point sources or sources which are very thin. In these situations, while the overall level of error may still be low, the percent error seen on the map will tend towards higher values rather than lower values.

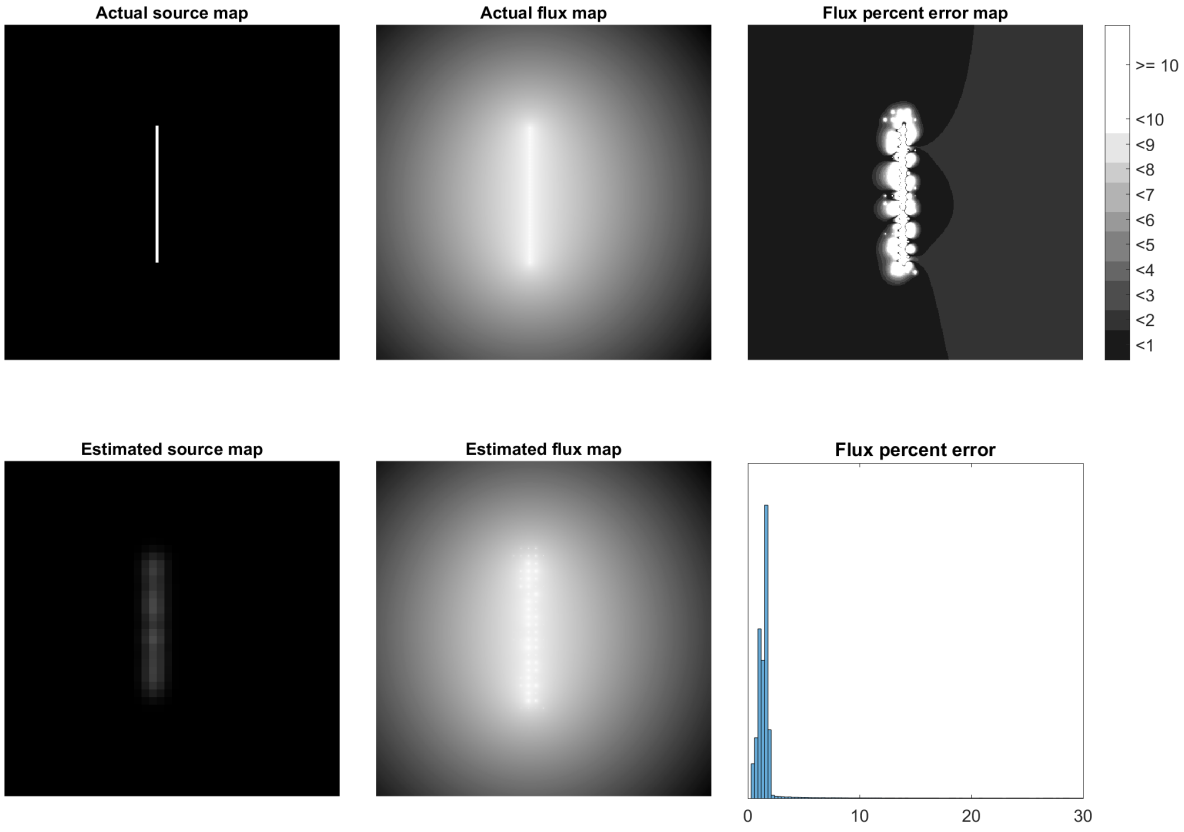
Figure 8.2: Map with a single point source



8.1.2 Map with a linear source

Figure 8.3 shows a map that has a linear source. The percent error map has a large section on the right side that is between one and two percent error. This is because this is a thin source and the grid of emitters that was placed in unexplored regions doesn't align perfectly with the actual source.

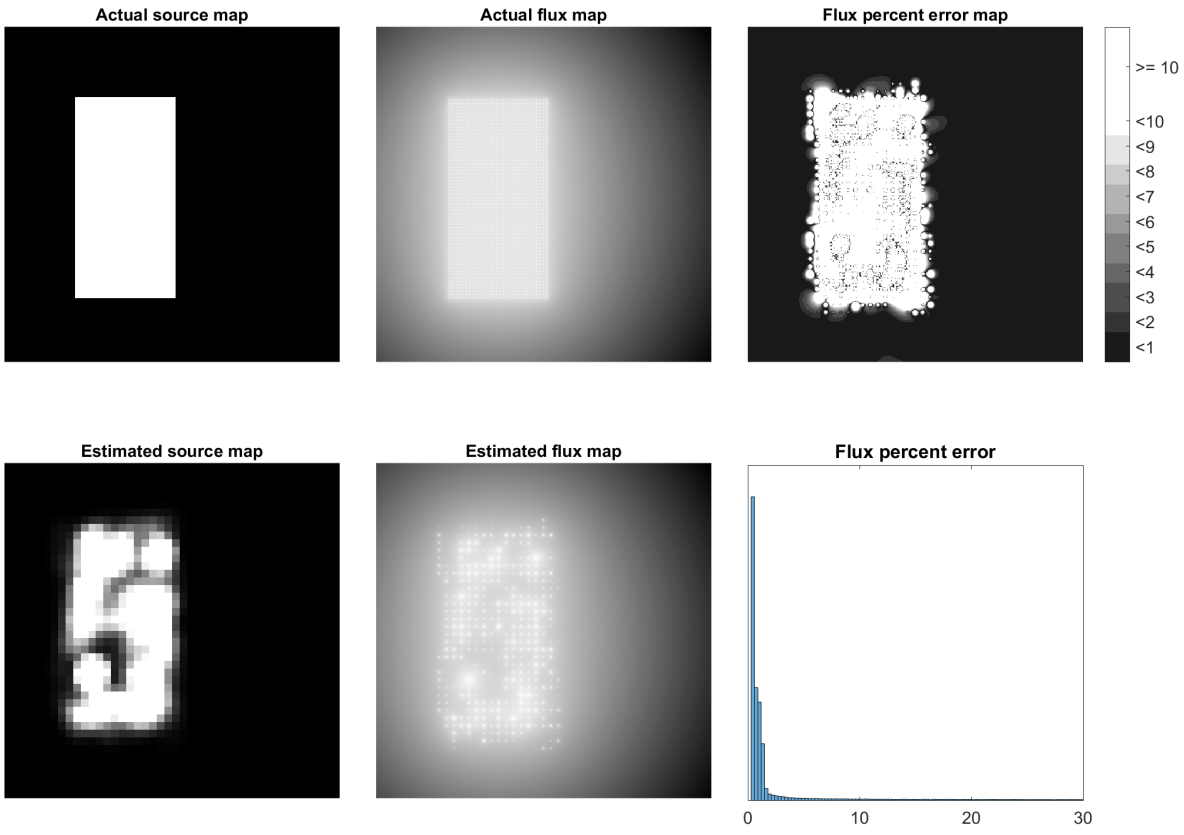
Figure 8.3: Map with a linear source



8.1.3 Map with an area source

Figure 8.4 shows a large rectangular area source. Because this map has an area source, flux match annealing can more easily replicate the flux within the sampled region. While the previous two examples had large regions of up to three percent error, this map's error is well below one percent in most locations. The errors seen where the source is located are generally above ten percent. Wile the flux match annealing algorithm can closely replicate measured flux, it doesn't model sources as well. These errors don't show up as a lump on the percent error histogram because they span a large range with a low number of counts per value.

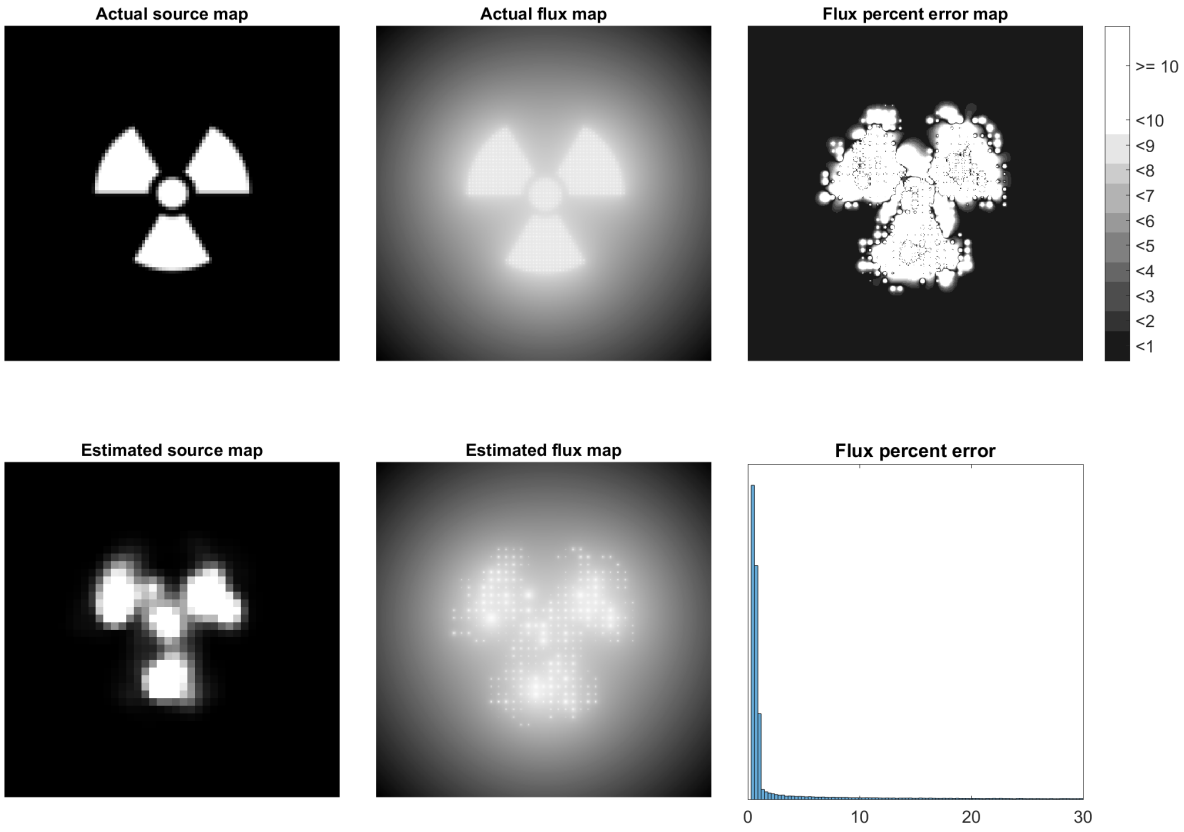
Figure 8.4: Map with an area source



8.1.4 Map with a complex source

Figure 8.5 shows a map that has a source with a complex shape. Although the source is easily identifiable as a symbol indicating radiation hazard, don't think of it as an image but rather a map of where radioactive paint may have been painted. This map shows that the flux match annealing algorithm works equally well regardless of the complexity of a source's shape. Once again, error is concentrated well below one percent for areas where the source is not located.

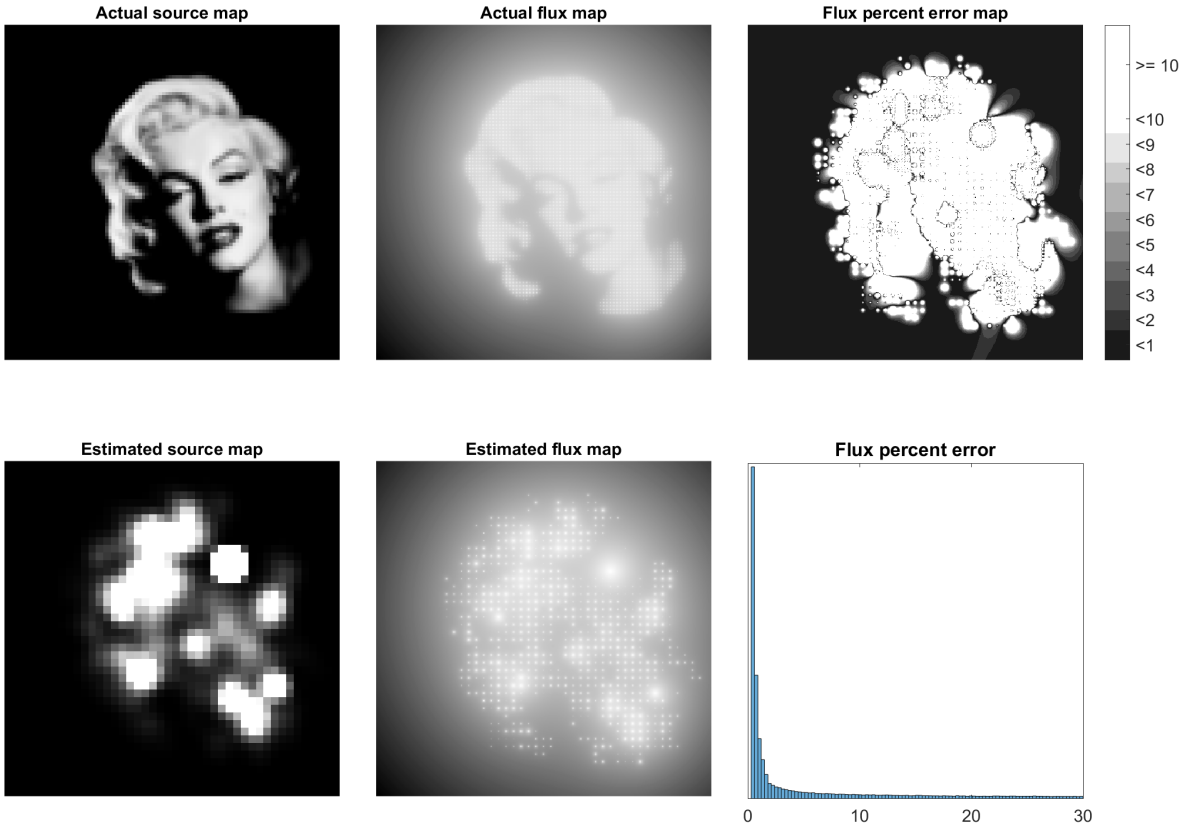
Figure 8.5: Map with a complex source



8.1.5 Map with a complex source having variable activity

Figure 8.6 shows a large area source that not only has a complex shape but also has a range of activities. While this map has a bit more error than the previous map, the bulk of the error is still well below one percent.

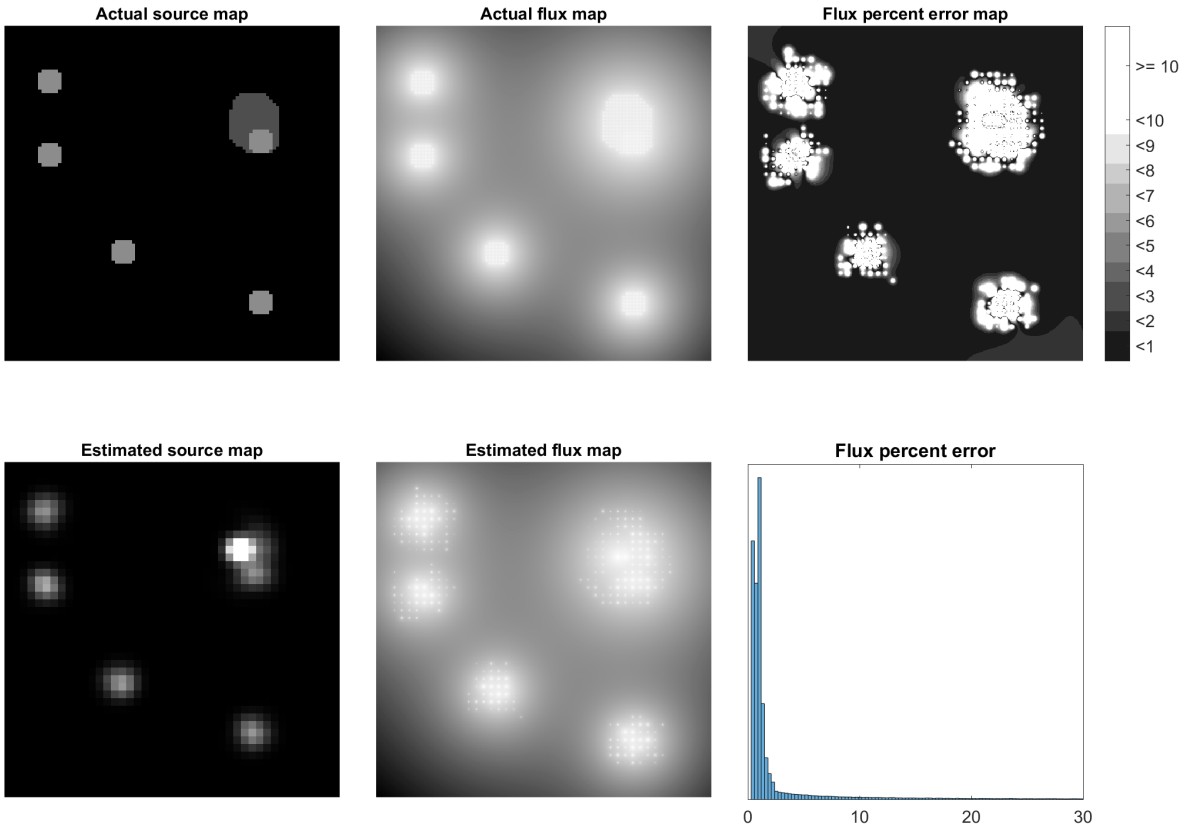
Figure 8.6: A second map with a complex source



8.1.6 Map with multiple sources and variable activity

The last example shown in this section shows a map that has multiple sources and varying activities. Flux match annealing has more difficulty matching flux when there are multiple sources rather than one within a map. This can be seen in the corners of the percent error map where error is between one and two percent as well as in the shape of the error histogram.

Figure 8.7: A map with multiple sources and variable activity



The flux match annealing algorithm was run for 100,000 iterations for each of the maps presented so far. Processing time for each iteration of the flux match annealing algorithm is linearly related to the product of the number of emitters and the number of measurement locations. For these examples, measurement spacing was larger than the emitter spacing. Because of this, as the size of a source increased, so did the processing time. Run time ranged from just above one minute for the point source map to 3.7 minutes for the second map with a complex source. These maps were processed on a desktop computer with an Intel i7-6700K CPU that ran at 4.0 GHz. The code for this thesis was written in Matlab. It would likely run several times faster if the algorithm was rewritten in a compiled language.

8.2 Error progression over time

The flux match annealing algorithm iteratively refines its solution over time. The following figures show results at various points along a single map's refinement. The actual source map is shown in Figure 8.8. The number of iterations shown in the figures are, going left to right and top to bottom, 245, 665, 1808, 4915, 13360, 36316, 98716, and 268337. These values were chosen because they are evenly spaced when plotted on a log scale and show a progression from a low number of iterations to a likely stopping point. During early stages, large changes happen often. Towards the end, changes can still be seen in the activity levels of the emitters but the total amount of flux in the various source locations has largely stabilized. When computing the results for this example, the random number generator was seeded to the same number each time so that the estimations all progressed in the same manner.

Figure 8.8: Actual source map used in this section

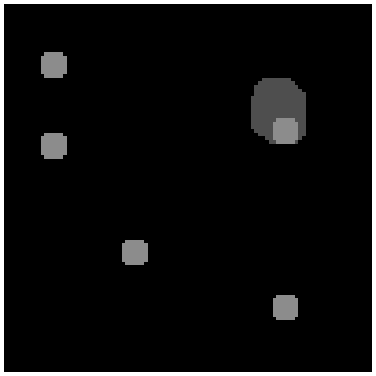


Figure 8.9: Estimated flux maps after different numbers of iterations

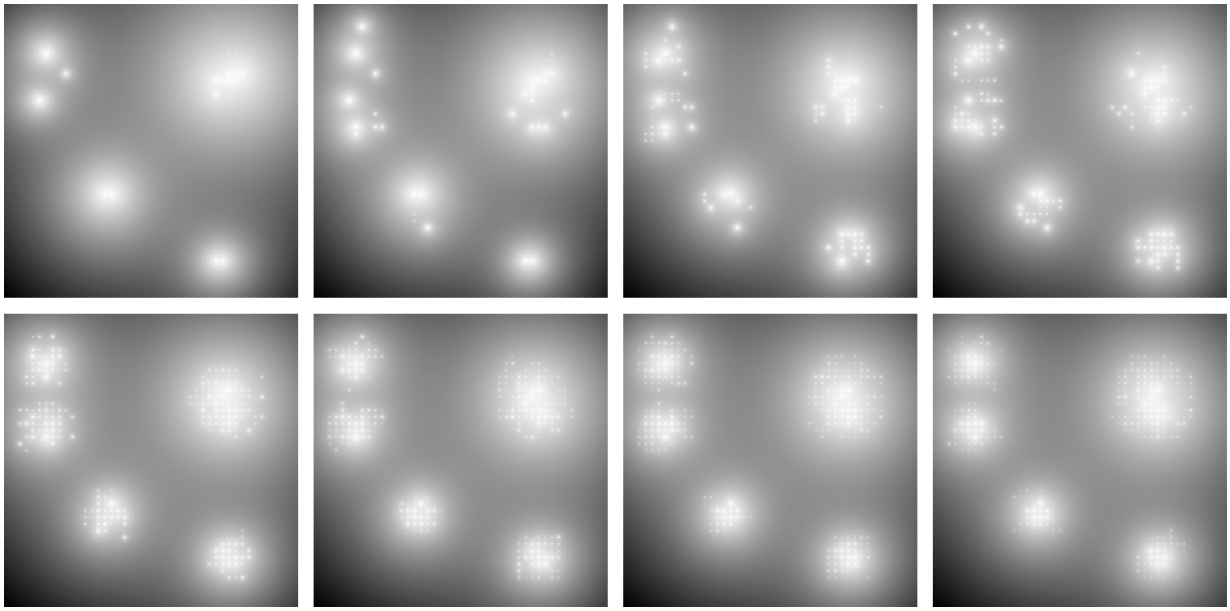


Figure 8.11 shows the progression of the flux error map. When the number of iterations are low, the error is significant and the algorithm's results would likely be unusable. At 4915 iterations, most error is below ten percent. The result at 36,316 iterations, which was calculated in 51 seconds, is likely to be a good trade-off between computation time and accuracy if this algorithm were to be used on a mobile robot. At 98,716 iterations, the majority of improvement has completed.

In the flux match annealing algorithm, the sum of the squared flux errors at the measurement locations is calculated. While this variable isn't directly relatable to the percent error, it is helpful to understand how this value progresses over time. It was found that this variable roughly follows the curve generated by Equation 8.1. In this equation, C is a variable that roughly relates to the number of emitters used. Figure 8.10 shows a graph where the sum of squared flux errors over time are compared to the predicted value.

$$sumOfSquaredFluxError = \frac{C}{iterations} \tag{8.1}$$

Figure 8.10: Flux error maps after different numbers of iterations

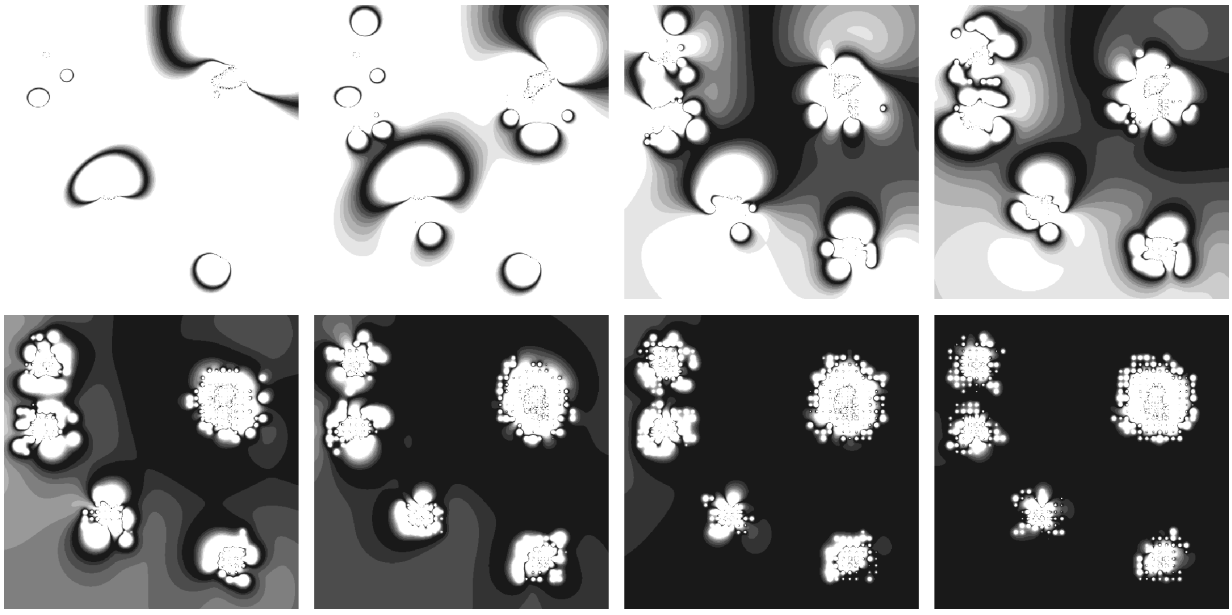
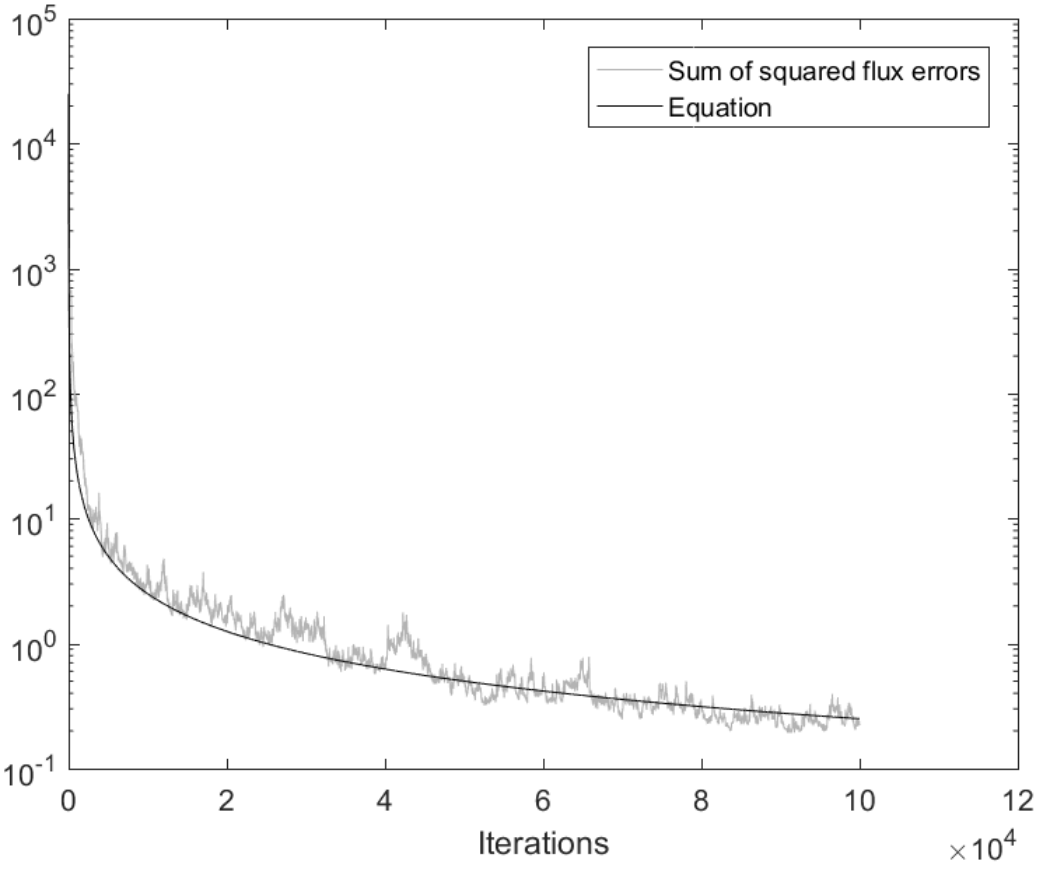
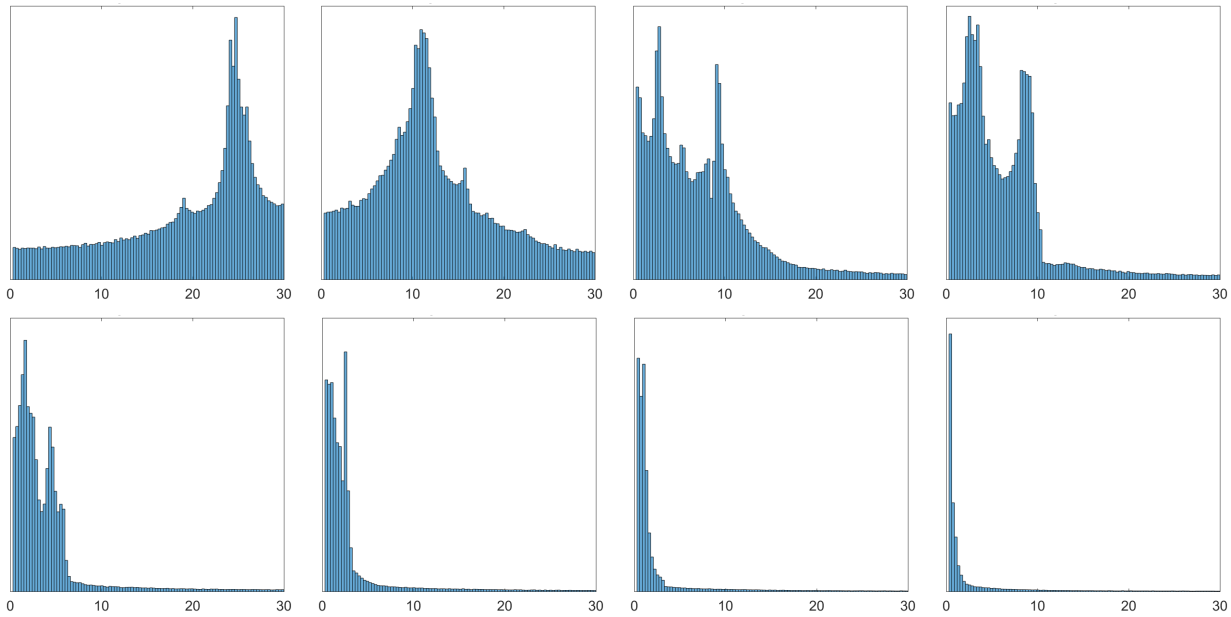


Figure 8.11: Graph comparing the sum of squared flux



Perhaps the most telling progression sequence is of the flux percent error histogram. While there is no way to predict the shape of the histogram at any given iteration, as more iterations happen, the histogram will progressively compress down to lower error amounts.

Figure 8.12: Flux error histograms after different numbers of iterations



8.3 Results with different measurement spacing

The maps in this thesis represent an area ten meters square. The default grid size for measurements is 0.25 meters. This section shows how changing the measurement grid's size affects the results of the flux match annealing algorithm. Figures 8.14 and 8.15 show maps where the measurement spacing varies from .25 meters to 1.5 meters. The actual source map is shown in Figure 8.13. Emitters in the examples are shown as light gray dots. Measurement locations are shown as black dots. The flux match annealing algorithm was run for 100,000 iterations on each set of measurements. Having a large number of measurements produces better results than a small number of measurements but the difference isn't as pronounced as might be expected. Despite only having thirty one flux measurements, the map with a grid size of 1.5 meters still manages to keep the flux percent error less than six percent in most places on the map. While it is not shown here, the flux match annealing algorithm was run for 400,000 iterations using the thirty one measurements just to see how much error improved. The result was similar to the error shown for a grid spacing of 1.0 meters.

Figure 8.13: Source map for the measurement spacing examples

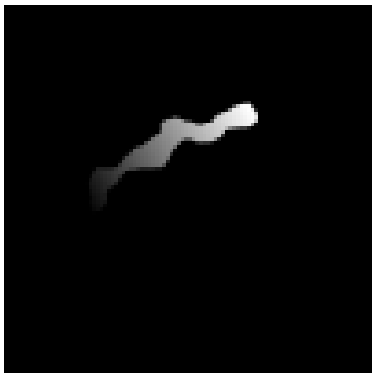


Figure 8.14: Results with measurement grid sizes from 25 to 50 centimeters

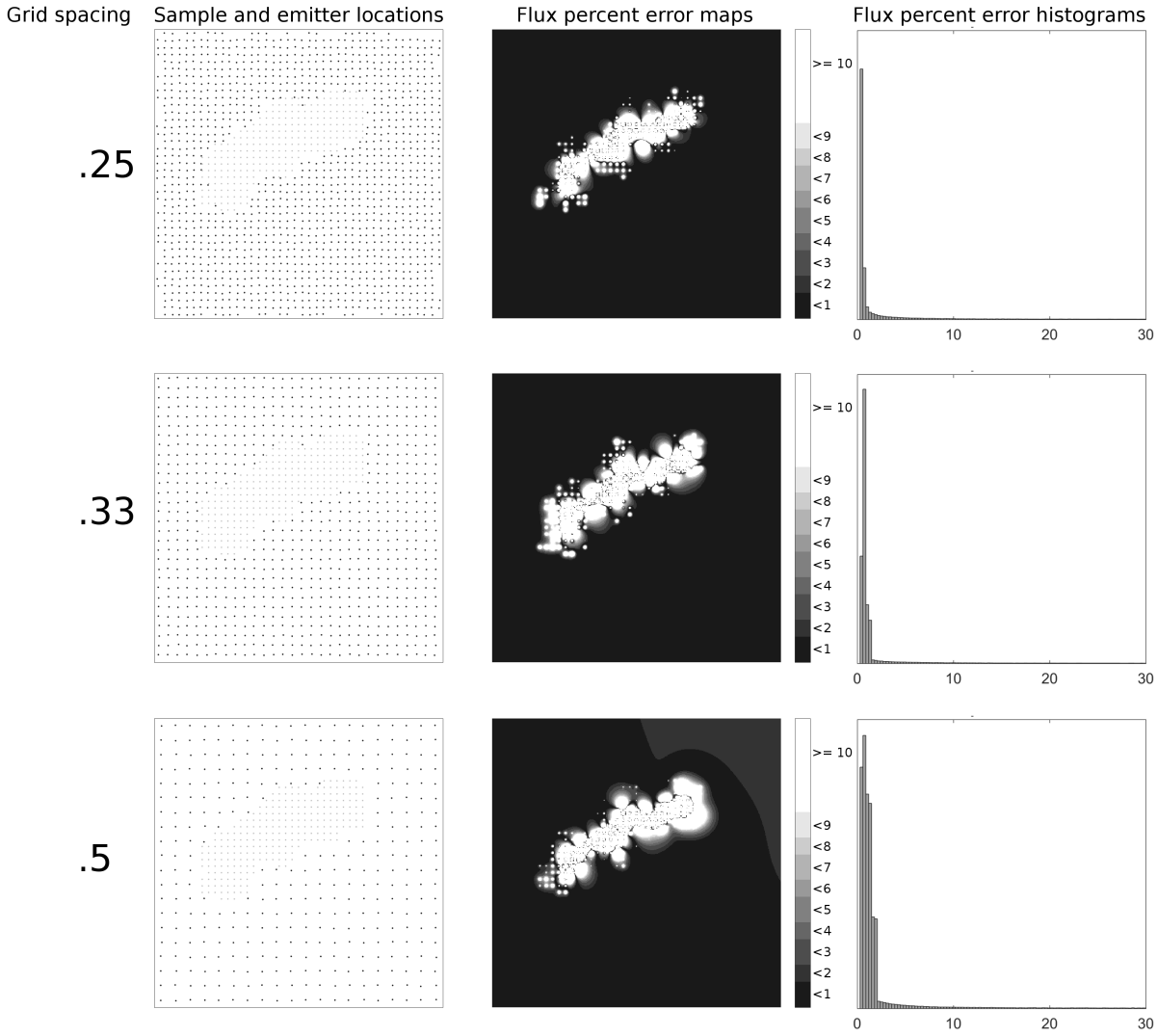
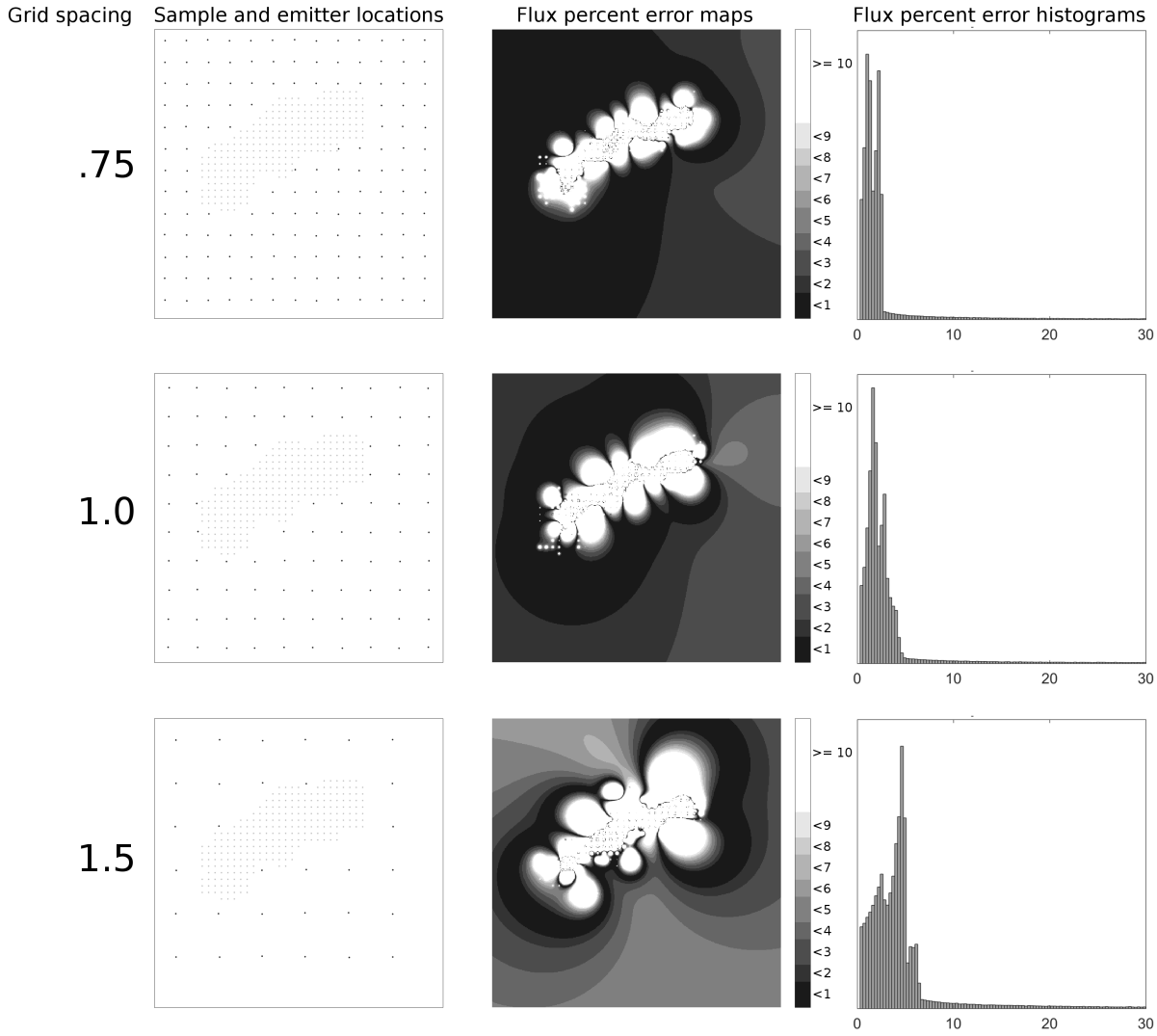


Figure 8.15: Results with measurement grid sizes from 75 to 150 centimeters



8.4 Results with shot noise

All of the previous examples have used perfect, continuous measurements that didn't contain noise. Gamma flux measurements taken with real sensors have shot noise which follows a Poisson distribution. This noise is due to the discrete nature of how gamma rays are detected. This section shows the performance of the flux match annealing algorithm using measurements that have shot noise. Gamma ray sensors detect individual gamma rays as they strike the sensor. Because of this, sensors often report in units of counts per second or counts per minute. When evaluating the performance of flux match annealing in the presence of noise, units of counts will be used because the measurement time will be varied.

Figure 8.16 shows a graph produced by running a set of measurements through this workflow three hundred times. Each trial was run for 100,000 iterations. Before each trial, the simulated measurement time was changed. After the trial, the median amount of error was found. The measurements provided to the flux match annealing algorithm were taken using a 0.75 meter grid. The same map that was used for the measurement spacing test was used for this test.

Due to the stochastic nature of this workflow, the resulting data is noisy. Despite this, a general trend can be seen where the median error decreases as the median number of counts increases. Note that for median counts above 100, there is still some reduction in error but it is minimal.

Figure 8.17 shows the sum of squared flux error over time for median counts of 8 through 233. Unlike Figure 8.11 which used perfect data and showed continual improvement over time, when noise is present, error tends to level off after a relatively low number of iterations. In the charts that used measurements with high median counts, some improvement is seen throughout but with very low median counts, no improvement is seen in later iterations.

Figure 8.16: Scatter plot showing the relationship between median counts and median percent error

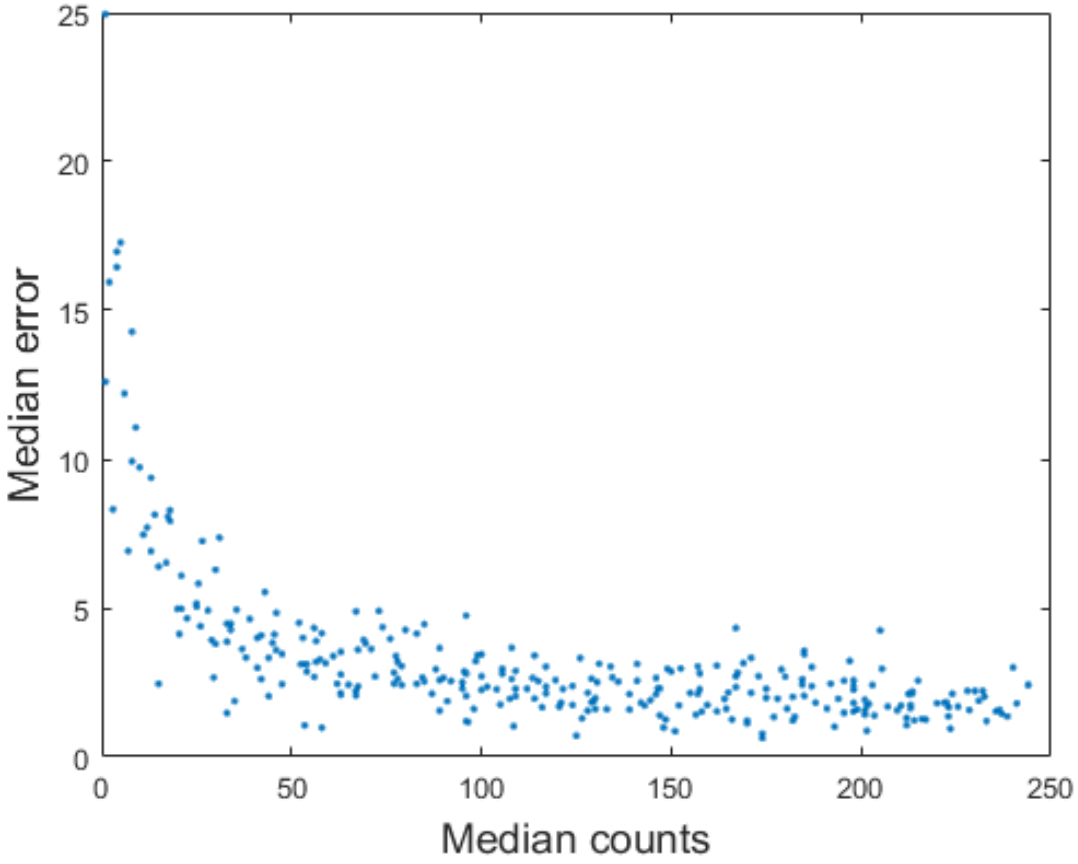
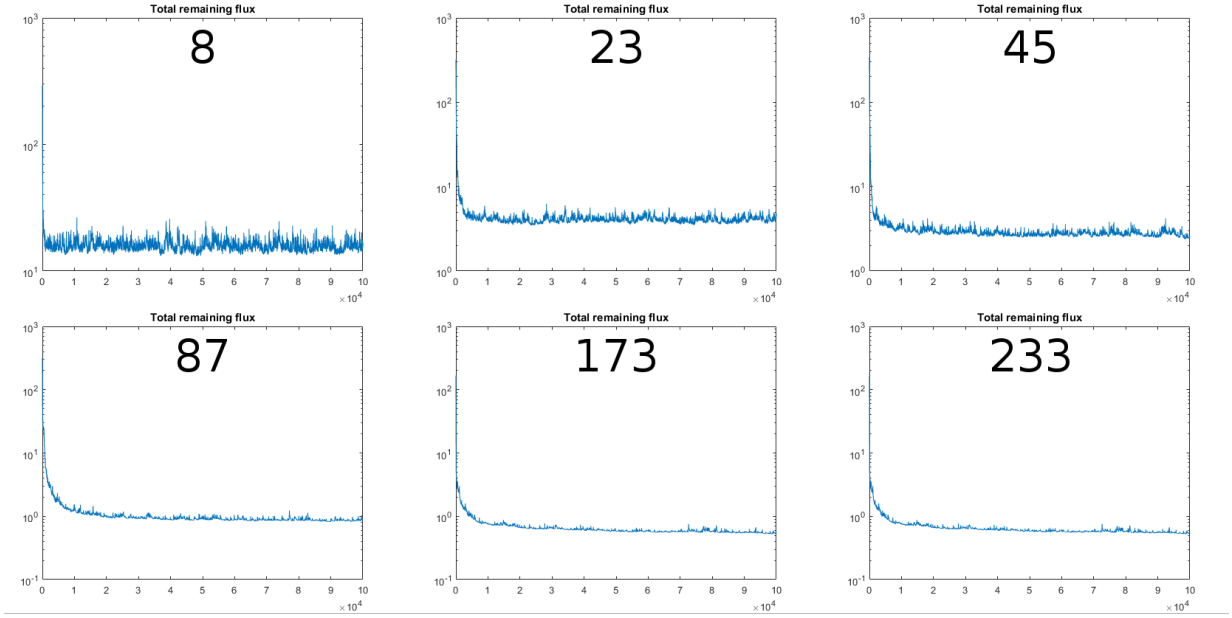


Figure 8.17: Error over time for measurements with different median number of counts



8.5 Extrapolation

Previous sections have shown only results within the sampled region. The point source renderer can generate dense flux maps of any region so extrapolation outside of the sampled region is possible. Figures 8.18 through 8.20 are formatted the same as all previous figures of this type with the exception that the flux percent error maps show a larger area around the sampled region. In each, a white square indicates the size of the sampled region.

Figure 8.18 shows results for a map containing a single area source with a gradient of activities. This is the best case scenario for extrapolation. Because error is less than one percent nearly everywhere within the sampled region, error is also less than one percent in the extrapolated region.

Figure 8.18: Extrapolated map with single source

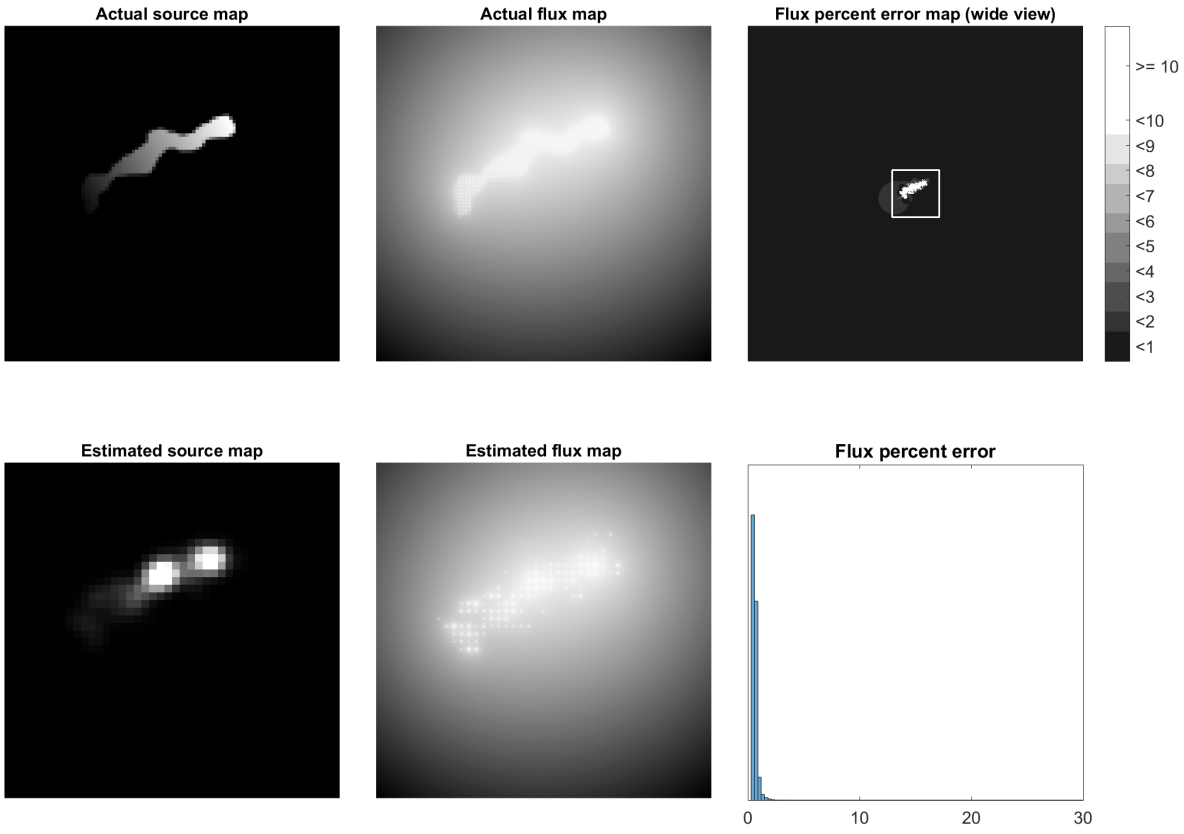


Figure8.19 shows a situation where there are areas of error greater than one percent outside of the sampled region. This map is typical of an extrapolated map where there are areas of error greater than one percent. Error outside of the sampled region is constrained to the maximum amount of error seen at the perimeter of the sampled region. This is true with two exceptions. The first is when there is a point source or thin source within the map such as seen in 8.2. In those cases, error can increase past the perimeter of the sampled region. When it does increase, the rate at which it increases slows down as you get farther away from the source. The other situation where error increases is if there is an external source. In that case, percent error can increase without limit.

Figure 8.19: Extrapolated map with multiple sources

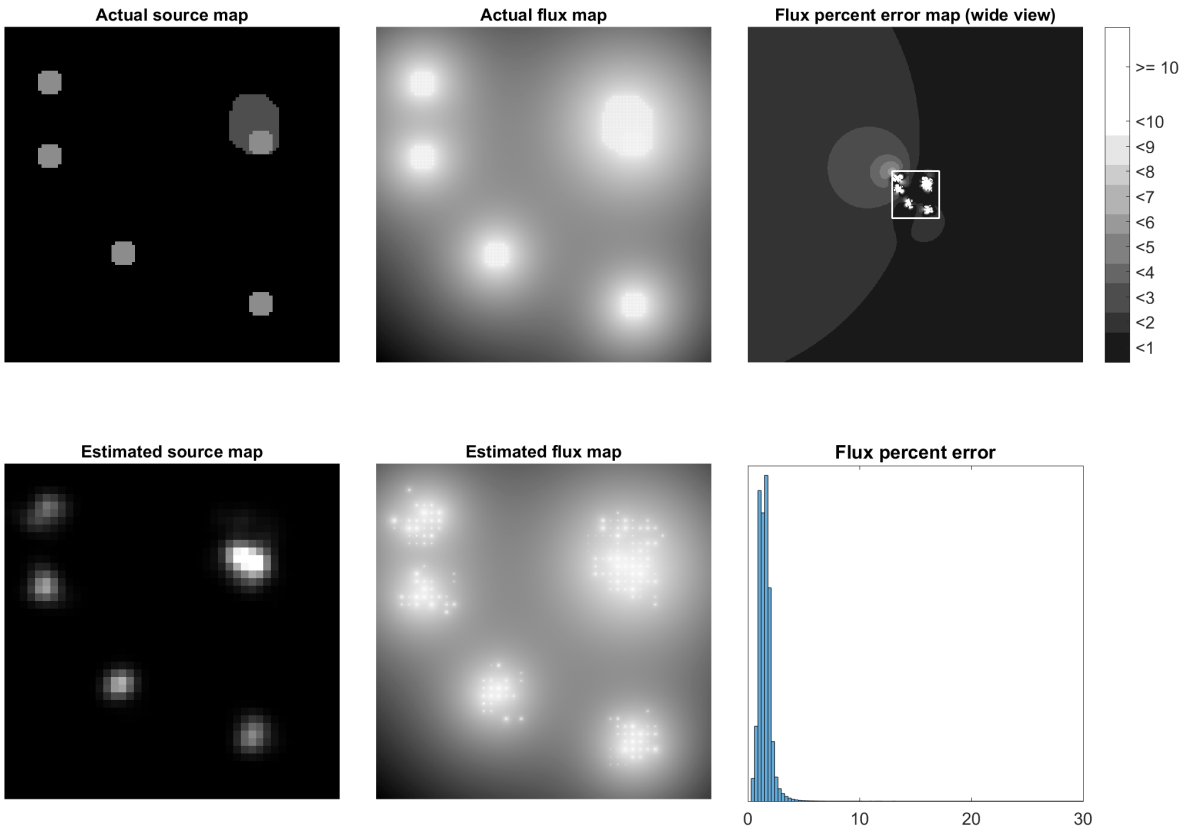
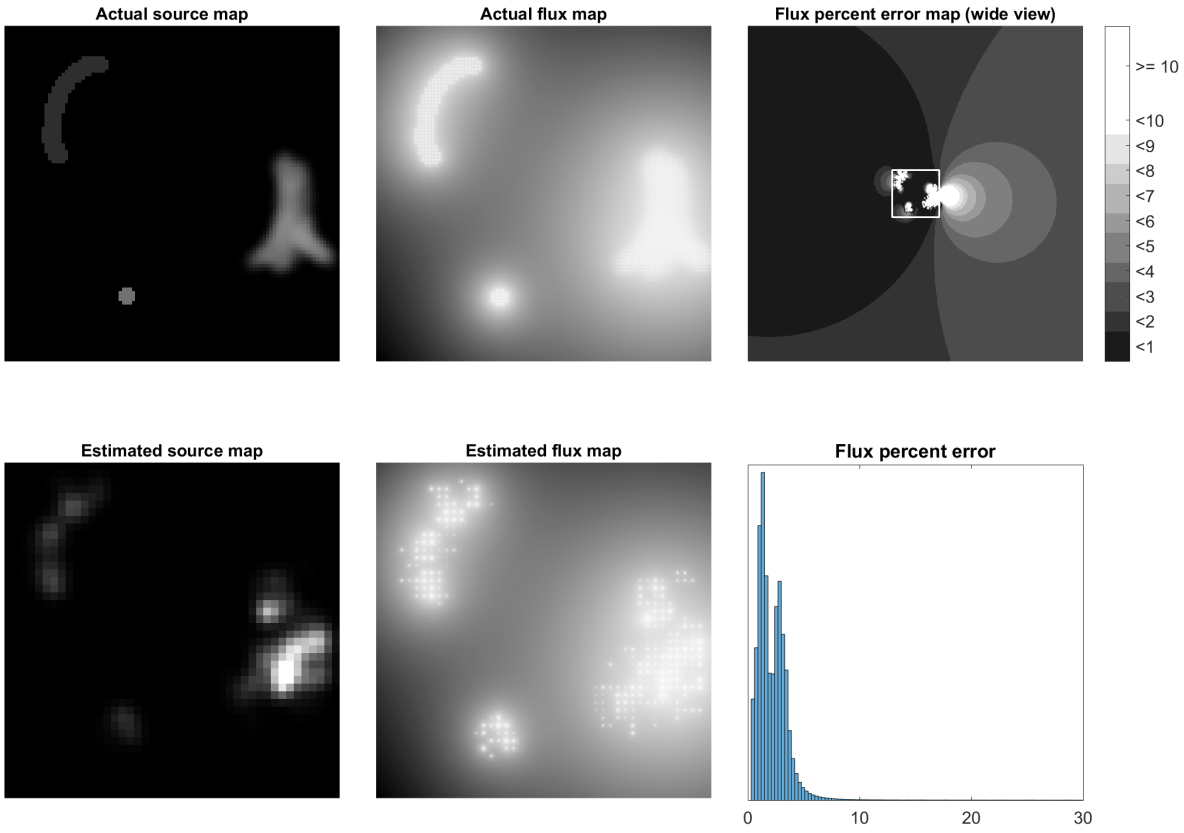


Figure 8.20 shows a situation where a source touched the edge of the sampled region. Because no measurements were taken to the right of the source, there is a large amount of error to the right of the source. This map shows the importance of completely surrounding sources with measurement locations.

Figure 8.20: Extrapolated map with multiple sources and source not completely surrounded

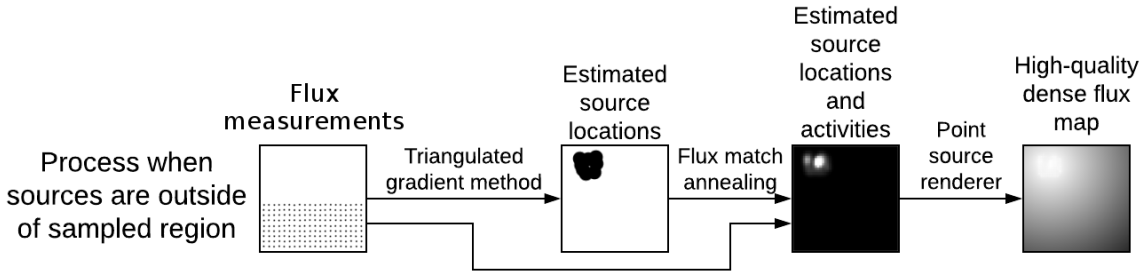


Chapter 9

Mapping external sources using the triangulated gradient method

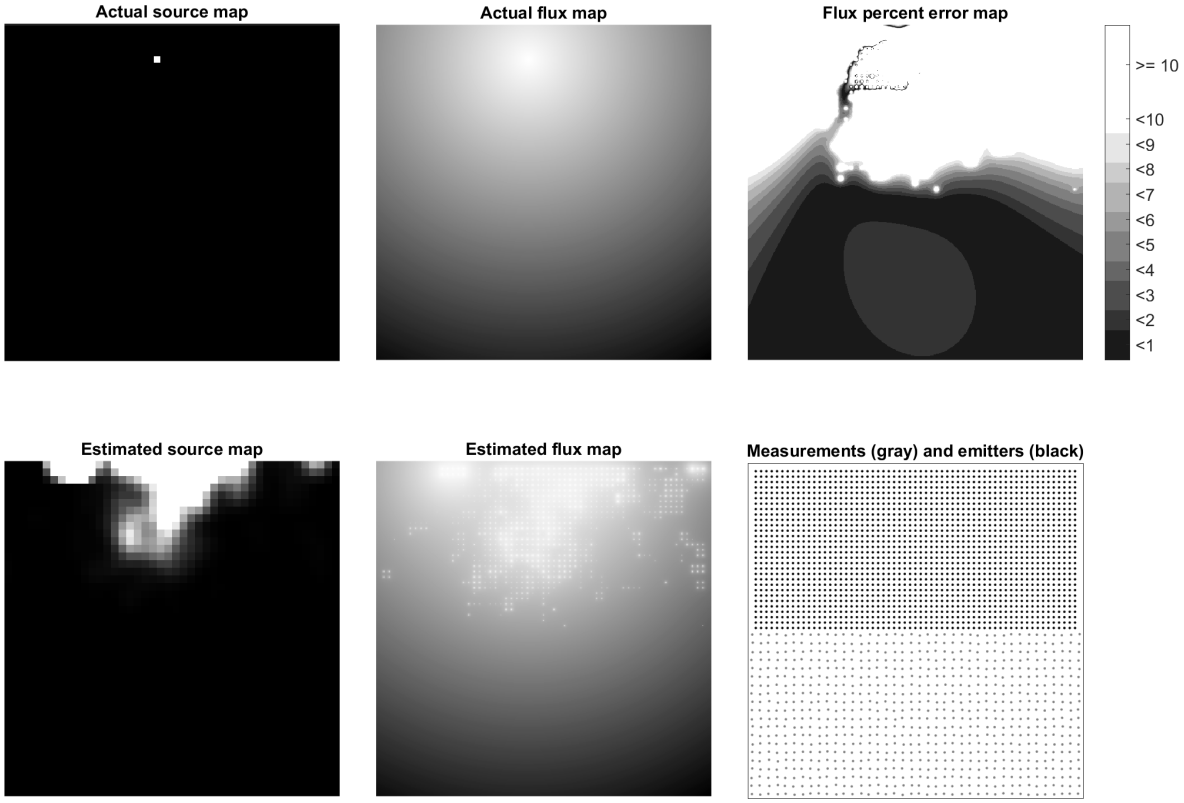
Figure 9.1 shows the third workflow proposed by this research. This workflow addresses the shortcomings of the second workflow when sources aren't located within the sampled region. In all of the examples in this section, measurements were only taken in the lower half of the map while the sources are located in the upper half of the map. The third workflow adds the triangulated gradient method to provide estimates of source locations. By placing emitters in a grid within a fixed radius of these estimates, the flux match annealing algorithm produces better estimated source activities which ultimately leads to lower error in the final flux map.

Figure 9.1: The third suggested workflow



Four maps will be used in this section. Each map will be presented twice. The first map will show results when the entire upper half of the map region is filled with a grid of emitters and the flux match annealing algorithm is free to adjust emitter activities anywhere. This is representative of the results of the second workflow. The second map will show results when the triangulated gradient method is used to help place emitters. The triangulated gradient method takes a set of gamma flux measurements and outputs estimated source locations. These locations are used to create a placement map that defines where emitters can be placed. For each estimated source location, a region within a fixed radius will be defined in the placement map as an area where emitters can be placed. Once the placement map is complete, a grid of emitters is created for the entire upper half of the map. Emitters that fall outside of the indicated areas on the placement

Figure 9.2: Map with single point source and filled region of emitters

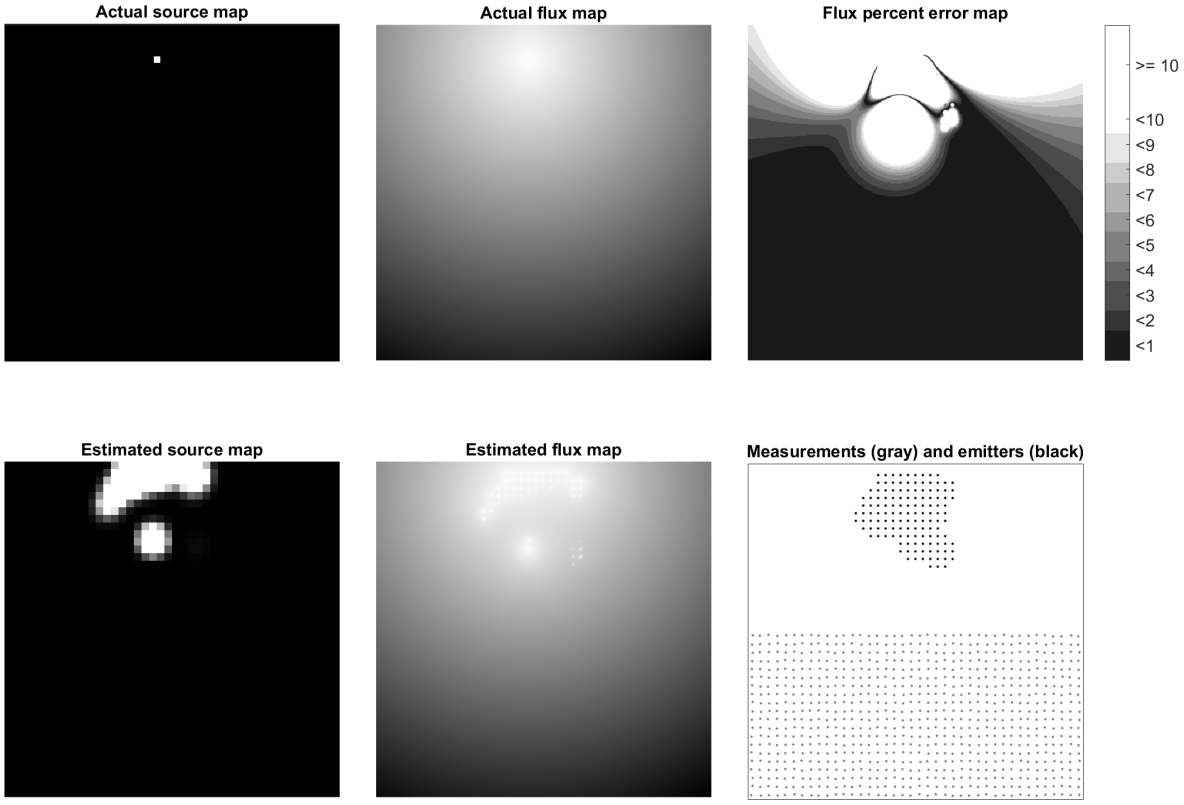


map are then removed. The remaining emitters are then passed to the flux match annealing algorithm along with the original measurements. The rest of this workflow is exactly the same as the workflow that uses flux match annealing and the point source renderer. For all examples shown here, the flux match annealing algorithm was run for 100,000 iterations. The bottom right image in each example shows where measurements and emitters were placed. Emitters are drawn as black dots and measurements are drawn as light gray dots.

Figure 9.2 shows a map that has a single point source which is located outside of the surveyed region. The entire upper area is filled with emitters. The triangulated gradient method finds a solution where the percent error in the sampled region is mostly less than two percent. The upper half of the map shows more than ten percent error in most areas. Like other examples, if the flux match annealing algorithm was run for more iterations, results would improve. This wouldn't be the best plan though. Not only would it run for more iterations, those iterations would take longer to compute than a solution where emitters were placed more strategically.

Figure 9.3 shows the same map as 9.2 but instead of blanketing the upper half with emitters, the triangulated gradient method was used to guide emitter placement. This constrained emitter placement to likely source locations and results in a more accurate result. The entire sampled region now has less than one percent error and about a quarter of the upper region does as well. The addition of the triangulated gradient method to this workflow has improved the results.

Figure 9.3: Map with single point source and emitters placed using the triangulated gradient method



The example in Figure 9.4 has a small area source in the upper left corner of the map. Here the sampled region doesn't have any areas that are less than one percent error. Almost the entire upper region has more than ten percent error.

Figure 9.4: Map with square area source and filled region of emitters

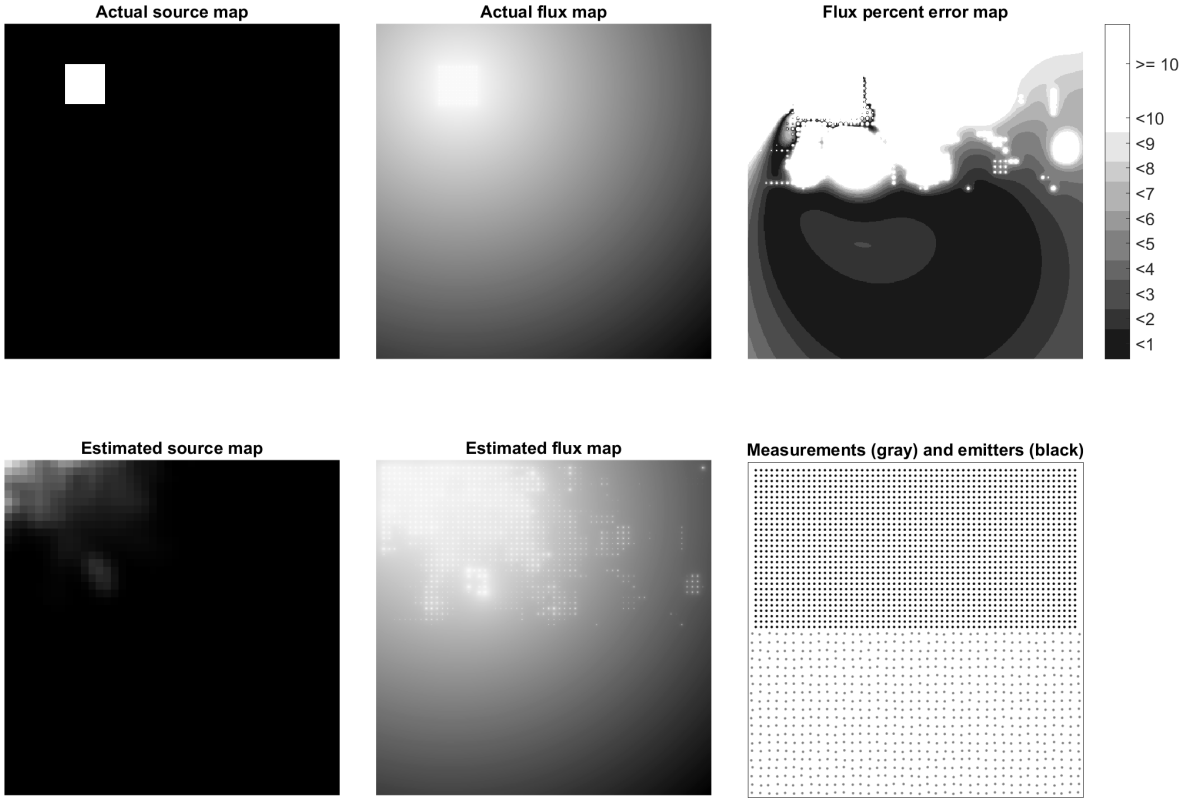


Figure 9.5 uses the triangulated gradient method to place emitters. The triangulated gradient method found the location of the actual source but also had an estimate in an incorrect area. That estimate was likely due to a triangle along the edge of the sampled region. Edge triangles were not removed for this sequence of tests. Despite the incorrectly placed emitters, adding the triangulated gradient method to the workflow increased the accuracy of the solution. Now, nearly the entire sampled region has less than one percent error and some of the upper half of the map does as well. If edge triangles had been removed then this solution likely would have been even better.

Figure 9.5: Map with square area source and emitters placed using the triangulated gradient method

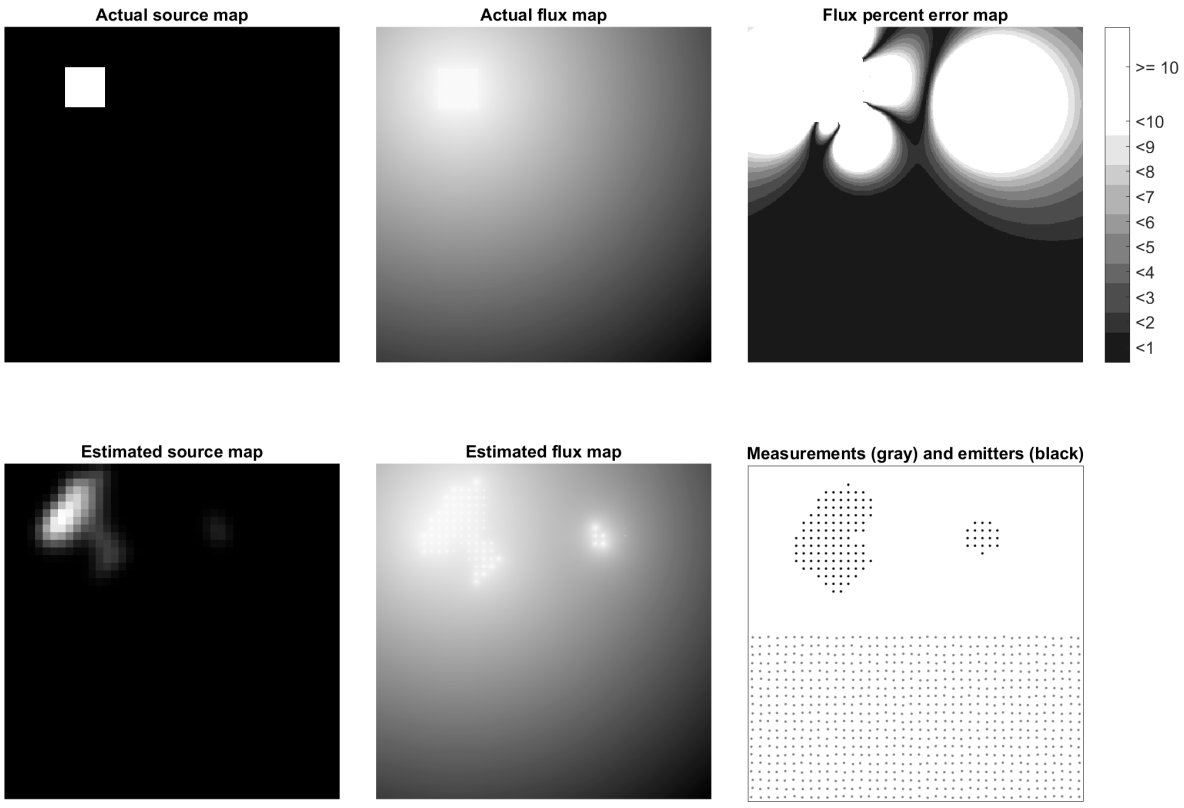


Figure 9.6 shows map with a single area source that has a gradient of activity levels. With the entire upper half of the map filled with emitters, the flux match annealing algorithm finds a solution where a large portion of the sampled region has less than one percent error.

Figure 9.6: Map with gradient area source and filled region of emitters

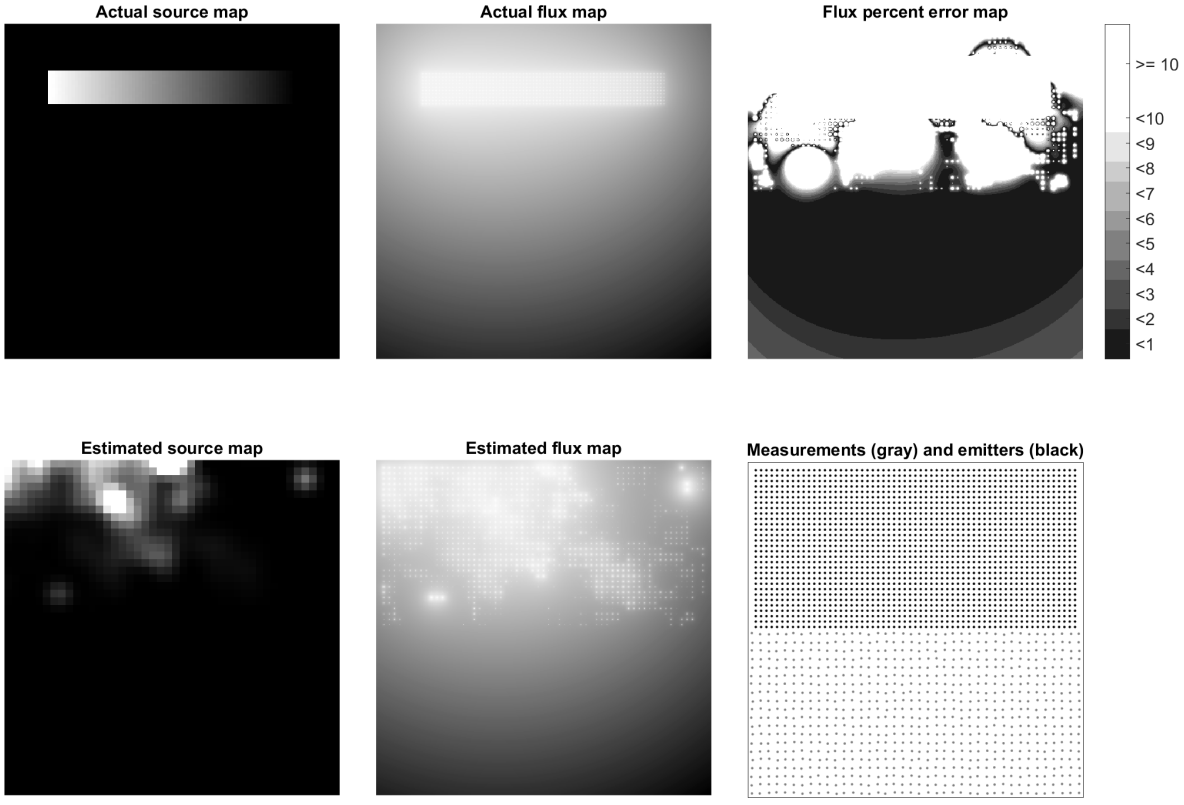
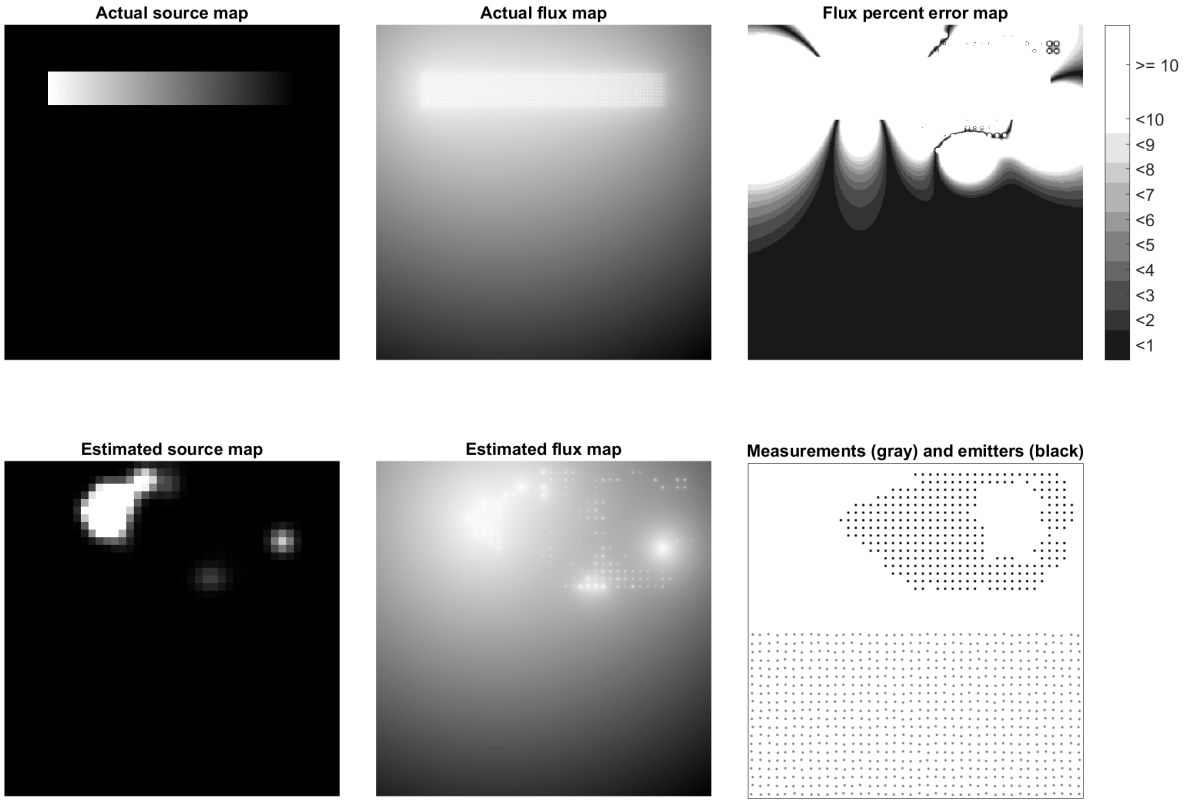


Figure 9.7 shows the results when the triangulated gradient method is used. Here, despite the triangulated gradient method producing a fairly poor set of estimates, the flux match annealing algorithm still manages to produce superior results when compared to simply filling the entire upper half of the map with emitters.

Figure 9.7: Map with gradient area source and emitters placed using the triangulated gradient method



The final example is shown in Figures 9.8 and 9.9. While using the triangulated gradient method usually produces superior results, that was not the case in this example. This map was designed to be difficult to replicate by placing one source close to the sampled region in the center of the map. A second, more active source was placed in the upper right corner and finally a third, weak source was placed in the upper left. It was expected that most solutions would gravitate towards modeling the source that is close to the sampled region and that if anything else was modeled, it would be the strong source in the upper right. The triangulated gradient method placed all of its estimates in the expected areas and completely missed the source in the upper left. This caused large errors on the left side of the map. When emitters were placed everywhere in the upper half of the map flux match annealing did a better job on the left side. This map was run twice more using both placement strategies and yielded similar results to what has been shown here.

Figure 9.8: Map with three area sources and filled region of emitters

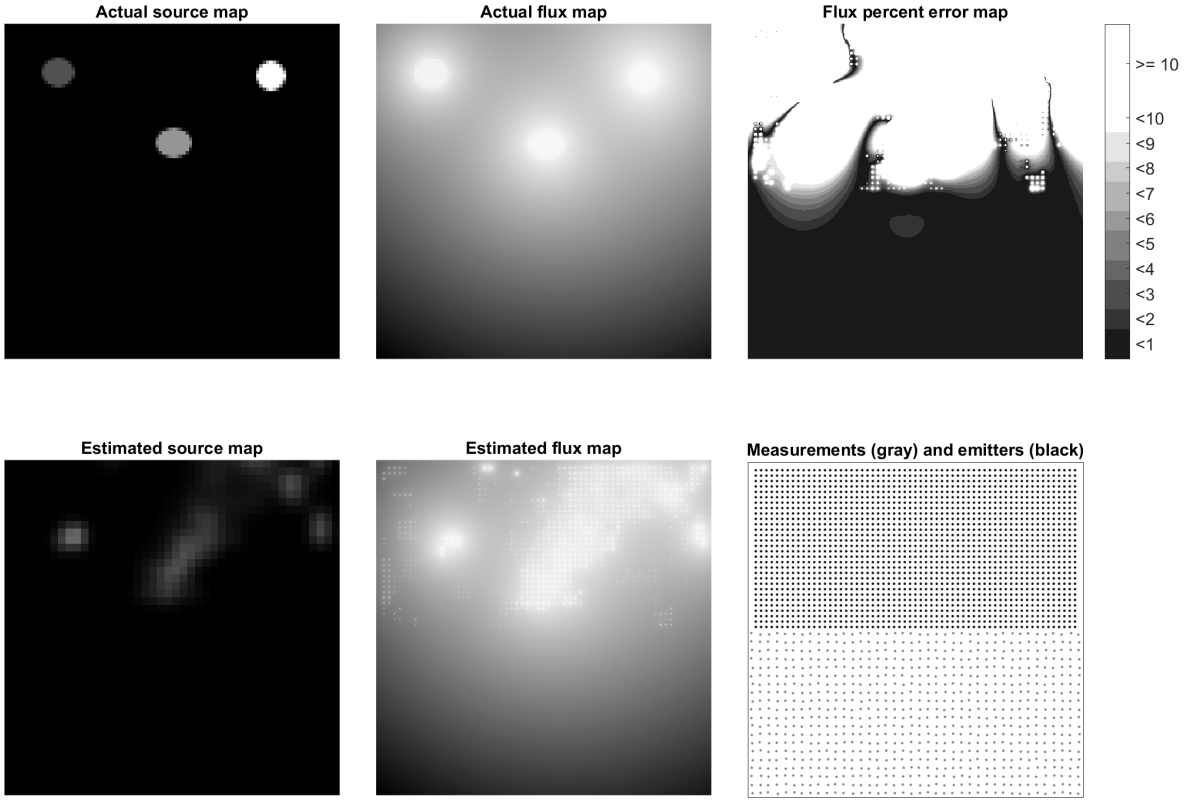
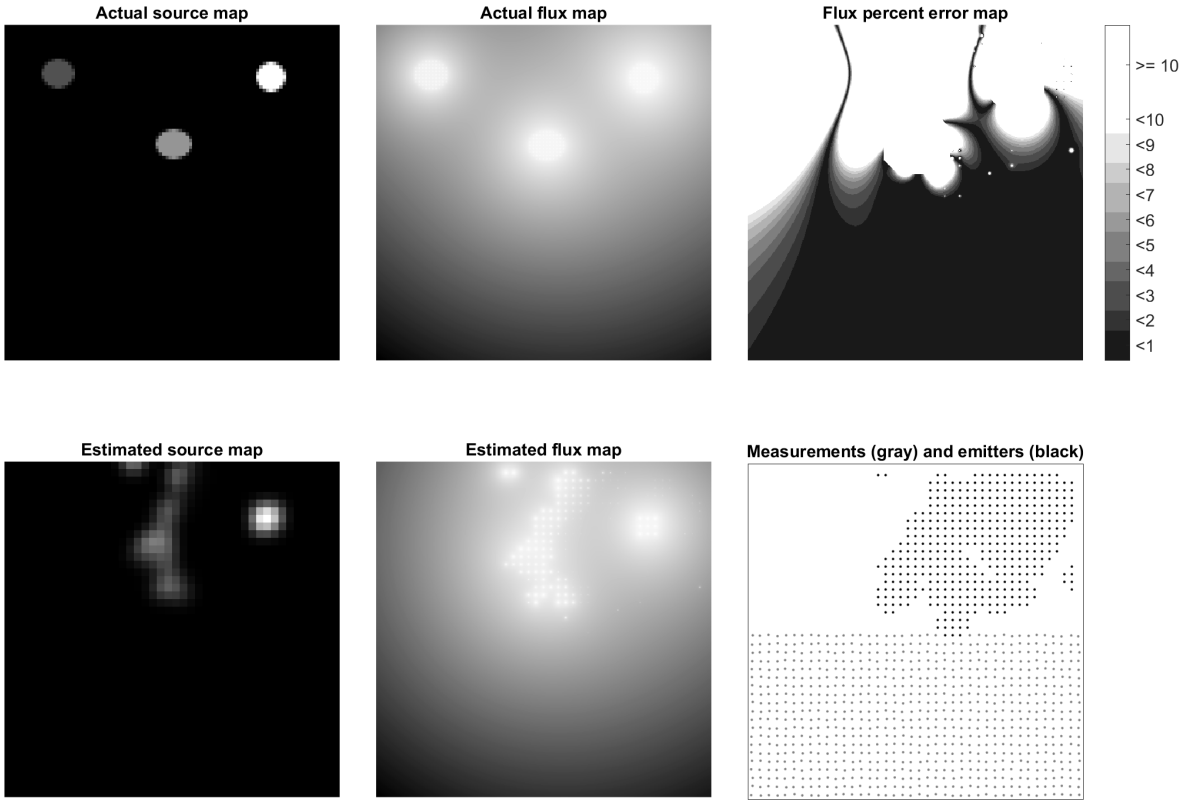


Figure 9.9: Map with three area sources and emitters placed using the triangulated gradient method



In most situations where sources are outside of the sampled region, using the triangulated gradient method to assist in placing emitters results in a solution that has less gamma flux error. Computation time is always reduced when compared to filling a large area with emitters. While using this workflow may allow for improved flux maps in some situations, the best method to reduce flux map error is to completely surround sources of gamma radiation with measurements locations.

Chapter 10

Future work

10.1 Point source renderer

Probably the largest topic that this thesis doesn't address is how to deal with other materials being between a source and a sample location. This could be air, building materials, or intentional shielding. Regardless of what the material is, it would cause some amount of attenuation to the gamma ray flux. The point source renderer currently treats all space as if it were a vacuum which doesn't attenuate gamma rays at all. When using the flux match annealing algorithm, attenuation can be addressed by modifying the point source renderer. This would require the point source renderer to accept an attenuation map. This map would contain an attenuation value for each pixel which would represent the expected attenuation per linear measurement unit. When calculating the flux contribution from an emitter, other calculations would need to be performed in addition to what is already being done. For each pixel in the attenuation map, an attenuation factor would need to be computed by considering the length of the path between the source and measurement location that passed through the pixel. This attenuation factor would represent the expected percentage of gamma rays that would exit that region of the map compared to the number which entered. The flux contribution would then be multiplied by the product of the attenuation factors for all pixels in the attenuation map between the source and measurement location. This would drastically slow computation time but if provided with an accurate attenuation map, might allow source locations to be found even in the presence of obstructions.

10.2 Triangulated gradient method

All work for the triangulated gradient method was done assuming that it would receive perfect data. In actual practice, the data has noise in it due to the shot noise inherent to radioactive decay. Because of this, it would be beneficial to adopt a probabilistic approach towards estimating source locations. Instead of representing the estimated location of the source as a single point, it would be represented by a two-dimensional probability distribution. This would take into account the number of counts in the flux measurements as well as the distance between the triangle and its estimate. Low numbers of counts would increase the size of the distribution because there

would be less certainty of the actual amount of flux at each vertex which would lead to larger variance in the source location estimates. The distance from the triangle to its estimate matters because triangles that are far from their estimated source are likely to have an incorrect gradient direction if there is more than one source. By creating a map consisting of the sum of the log probability distributions from all of the triangles, a more accurate estimate of source locations may be found.

10.3 Flux match annealing

While it may be possible to tune the acceptance rate and sigmaConstant variable to match a use case more exactly, better results might be achieved by replacing the flux match annealing algorithm with a genetic algorithm. The genetic algorithm would mutate emitters in the same manner as the flux match annealing algorithm. It would also evaluate the solutions in the same manner. Various crossover methods would need to be explored. A potential crossover strategy might be to randomly select half of the emitters from one map and to use the remaining emitters from the other. This strategy has the potential to retain what is good about both maps because it would relatively evenly select from everywhere in the maps without resorting to using a pattern which could have artifacts. A genetic algorithm may be superior to the flux match annealing algorithm. It was noted that when plotting multiple trials with the same settings that solutions which had lower sums of squared flux errors tended to stay lower. This implies that despite the local minimum avoidance shown by the flux match annealing algorithm that early decisions tended to limit the amount of error that could be achieved. A genetic algorithm would allow more successful early choices to greatly influence the end result.

Using a genetic algorithm would come at a greatly increased computational cost but on modern computers which could likely run each iteration on a graphics card, this may not be an issue because each member of the population would be generated and evaluated in parallel.

10.4 Inverse square renderer

The inverse square renderer works well everywhere except when a triangle is close to a source of radiation where there is a high degree of change in the gradient direction. This error could be reduced by changing how triangles are rendered. The current method finds a single gradient direction for each triangle and fits a flux gradient across the triangle. A possibly better method would be to calculate the location and activity of a point source for each triangle without finding the gradient direction first. The pixels within the triangle could then be rendered directly from that point source using the point source renderer. This alone wouldn't be sufficient to generate a good map because there would be continuity errors at the triangle edges.

To fix this, a variant of the Phong shading model from computer graphics could be used. The Phong shader uses a per-vertex normal to make a polygon appear to curve smoothly and to blend nicely with its neighbors. To get a normal for a vertex, the normals for all polygons that a vertex is part of are averaged. The resulting normal is applied to this vertex.

When rendering a flux map, nearly the same thing would be done. The locations and activities

of the estimated point sources for all triangles that shared a vertex would be averaged. The averaged location and activity would then be applied to the vertex. Phong shading interpolates a normal for each pixel that is rendered using the normals at the containing triangle's vertices. For the purposes of mapping gamma flux, instead of interpolating a normal, the activity and location of the estimated point source would be interpolated. The point source rendering algorithm could then use that estimated point source and the pixel's location to calculate a final flux value.

While this process would be slower than the current version of the inverse square renderer, it would have much less error in triangles that have large changes in flux direction. This modified version of the inverse square renderer would still be a good candidate for running on a mobile robot.

Besides having less error due to taking into account flux direction changes across a triangle, this change could have greater tolerance to shot noise due to the estimated point sources at the vertices being averages.

10.4.1 Three dimensions

The algorithms presented in this research functioned in two dimensions. There is no obstacle preventing any of them from being expanded into three dimensions.

- The flux match annealing algorithm can easily be expanded by using voxels instead of a two dimensional grid of emitter locations and by selecting the voxels using a cuboid instead of a rectangle. Computation time will increase but otherwise the change is trivial.
- The point source rendering algorithm that was used in this thesis currently calculates distances in two dimensions but expanding it to three dimensions is trivial.
- The triangulated gradient method of locating sources is a bit more complex to bring into 3D. Instead of performing a triangulation pass on the measured data, measurement locations would need to be turned into a set of tetrahedrons. Delauney tetrahedralization is the tetrahedral equivalent to Delauney triangulation and could be used as a starting point. Gradient directions would need to be calculated based on the tetrahedrons. An intersection between a line and a triangle must be found instead of a line and a line segment. Additionally, flux must be interpolated along a triangle instead of just a line to find the flux at the intersection points.
- The inverse square rendering method would require the same changes as the triangulated gradient method. The rest of the algorithm could proceed as it does in two dimensions with the exception that there would be a third dimension involved in the projection of a voxel onto the gradient vector.

10.4.2 Mapping the locations of differing nuclides

This thesis lumped all radioactive materials into one category. Many detectors can differentiate between different nuclides based on the energy spectrum of the gamma rays that they produce. It may be possible to expand the algorithms presented here to determine the locations of different radioactive materials existing in or around the same sampling region by taking into account the energy spectrum of received gamma rays at each measurement location.

10.4.3 Mapping using an inverse square mixture model

It may be possible to characterize sources with more fidelity by using an inverse square mixture model. This would be the same thing as a Gaussian mixture model but would use an inverse square distribution instead of a Gaussian.

Chapter 11

Conclusion

Four algorithms and three workflows that use these algorithms were presented in this thesis. Table 11.1 compares the algorithms and workflows. For the sake of brevity, the algorithm names were reduced to the following acronyms: triangulated gradient method (TGM), flux match annealing (FMA), point source renderer (PSR), and inverse square renderer (ISR).

Table 11.1: Comparison of the algorithms and workflows presented in this thesis

	TGM	**FMA**	**PSR**	**ISR workflow**	**FMA & PSR workflow**	**TGM, FMA, & PSR workflow**
Makes dense flux map	No	No	Yes	Yes	Yes	Yes
Estimates activity	No[1]	Yes	No	No[1]	Yes	Yes
Estimates source locations	Yes	Yes	No	No[2]	Yes	Yes
Computational speed	Fast	Slow	Varies[3]	Fast	Slow	Slow
Flux map quality in sampled region	—	—	—	Good	Excellent	Excellent
Flux map quality outside sampled region	—	—	—	—	Poor	Acceptable
Source activity estimate quality	—	Fair	—	—	Fair	Fair
Source location estimate quality	Varies	Fair	—	—	Fair	Varies
Tolerance to shot noise	Fair	Excellent	—	Fair	Excellent	Good

[1] The triangulated gradient method and inverse square renderer can easily be modified to provide estimates of source activity but this would likely be of little use unless there was only one point source near the sampled region.

[2] The inverse square renderer, as written, doesn't provide source location estimates but could be easily modified to do so.

[3] The point source renderer's speed varies based on the number of locations that require flux estimates as well as how many emitters are passed to it.

This thesis presented three workflows that take sparse, non-directional gamma flux measurements and output dense gamma flux maps. Measurements for all of the workflows can be made using a Geiger counter, which is readily available, inexpensive, and can be lightweight. These three features make it possible to quickly build one or more ground-based or aerial robots to survey an area. These algorithms will also work using measurements taken using other non-directional sensors.

The inverse square renderer workflow provides a flux map for areas within the convex hull of a surveyed region. Its computations are efficient which makes it ideal for use on a mobile robot that needs a dense gamma flux map. After at least three measurements have been made, each time a new measurement is taken, the inverse square renderer would only need to update the map inside of any new triangles. Additionally, for each triangle within the map, a source location estimate can be made, which could be used to inform navigational choices.

The flux match annealing and point source renderer workflow is better suited for offline mapping purposes. The flux maps that it provides have lower amounts of error than the inverse square renderer workflow's flux maps. While this workflow characterizes sources, it computes flux with higher fidelity than its determination of source location and activity. The flux match annealing algorithm's results are highly tolerant to the shot noise inherent in gamma flux measurements. In most cases, if lower amounts of error are desired, running the flux match annealing algorithm for more iterations will result in more accurate gamma flux maps. Because this workflow generates its flux maps from a set of emitter locations and activities, flux maps can extend beyond the convex hull of the sampled region. When flux maps are extended beyond the sampled region, error is bounded by the maximum error seen along the perimeter of the sampled region provided that there are no external sources. Flux match annealing will perform best when measurement locations completely surround all sources that are in or near the region in which a map is desired. While longer measurements will result in less flux map error, improvement slows greatly when the median number of counts per measurement is more than fifty. Very few measurements are needed to generate useful results with the flux match annealing algorithm. While more measurements will always improve results, a realistic target may be one or two measurements per square meter to achieve less than five percent error within the sampled region.

The triangulated gradient methodology developed in this research characterizes the location of sources that lie outside of a sampled region where it isn't possible to obtain flux measurements. When sources lie outside of the sampled region, the triangulated gradient method can be used to help create a set of emitters to be used by the flux match annealing algorithm. In most cases, placing emitters in this manner causes the flux match annealing algorithm to run more quickly and to provide better results.

The workflows created for this research provide ways to generate dense gamma flux maps within a sampled region that leave little room for improvement in flux map quality. This research didn't solve the problem of exactly characterizing area source locations and activities, however it does provide estimates of where sources are located. Research still needs to be done to find methods to more closely characterize source locations and activities both inside and outside of a sampled region using non-directional gamma flux measurements. It is likely that most future improvements in gamma source mapping will be done with Compton cameras but unless there is a large reduction in their price, those sensors won't see widespread use. Until then, relatively inexpensive non-directional sensors are likely to be the most commonly used type of radiation

sensor which will maintain the relevance of the research presented in this thesis.

Bibliography

[1] Theodore W. Manikas and James T. Cain. Genetic algorithms vs. simulated annealing: A comparison of approaches for solving the circuit partitioning problem. Technical report, Southern Methodist University, May 1996. https://scholar.smu.edu/engineering_compsci_research/1. 2.1

[2] L. Ingber. Simulated annealing: Practice versus theory. *Mathematical and Computer Modelling*, 18(11):29–57, 1993. 2.1, 2.1

[3] Peter J. M. van. author. Laarhoven and Emile H. L. Aarts. *Simulated Annealing: Theory and Applications*. Dordrecht : Springer Netherlands : Imprint: Springer, 1987. 2.1

[4] Scott Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, 34(5):975–986, 1984. 2.1

[5] Peder Kock, Christopher RÃd'Ãd'f, and Christer Samuelsson. On background radiation gradients - the use of airborne surveys when searching for orphan sources using mobile gamma-ray spectrometry. *Journal of Environmental Radioactivity*, 128:84–90, 2014. 2.2

[6] Weiyi Wang. Techniques and applications of compton imaging for position-sensitive gamma-ray detectors, 2011. 2.2.1, 2.2.1

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. 2.2.1

[8] Jonas M.C. Nilsson, Robert R. Finck, and Christopher L. Raaf. A rotating-slit-collimator-based gamma radiation mapper. *Journal of Field Robotics*, 25(8):425–466, 2008. 2.2.1, 2, 3, 2.4.3

[9] Shawn Tornga. A prototype compton imager : simulations, measurements and algorithm development, 2010. http://digitalrepository.unm.edu/ne_etds/48. 2.2.1

[10] Soo Mee Kim, Hee Seo, Jin Hyung Park, Chan Hyeong Kim, Chun Sik Lee, Soo-Jin Lee, Dong Soo Lee, and Jae Sung Lee. Resolution recovery reconstruction for a compton camera. *Physics in Medicine and Biology*, 58(9):2823–2840, 2013. 4

[11] Silvia Barros and Geehyun Kim. Monte carlo studies and response of a new rotational modulator collimator (rmc) design. *Journal of Radioanalytical and Nuclear Chemistry*, 316(3):1119–1128, 2018. 2.2.1

[12] MA Kujoory, EL Miller, HH Barrett, GR Gindi, and PN Tamura. Coded aperture

imaging of $\gamma - raysourceswithanoff - axisrotatingslit$. *Applied Optics*, $19(24)$: $4186 - -4195, 1980$. 2.2.2

[13] Lifeng Yu and Shuai Leng. Image reconstruction techniques. https://www.imagewisely.org/imaging-modalities/computed-tomography/medical-physicists/articles/image-reconstruction-techniques, 2016. 2.2.2

[14] E. E. Libin, S. V. Chakhlov, and D. Trinca. Direct method for calculating the inverse radon transform and its applications. *eprint arXiv:1512.09140*, 2015. 2.2.2

[15] Iterative reconstruction, July 2018. `https://en.wikipedia.org/w/index.php?title=Iterative_reconstruction`. 2.2.2

[16] Aya Kishimoto, Jun Kataoka, Takanori Taya, Leo Tagawa, Saku Mochizuki, Shinji Ohsuka, Yuto Nagao, Keisuke Kurita, Mitsutaka Yamaguchi, Naoki Kawachi, Keiko Matsunaga, Hayato Ikeda, Eku Shimosegawa, and Jun Hatazawa. A first demonstration of multi-color 3-d in vivo imaging using ultra-compact compton camera. Technical Report 2110, Scientific Reports 7, 2017. 2.2.2, 5

[17] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Nonparametric filters. In *Probabilistic Robotics*, pages 96–114. MIT Press, 2005. 2.2.3

[18] W Zhen, D Shah, M Lee, M Hanczor, and S Scherer. Combining 3d mapping and radiation source localization in nuclear sites., 2017. 2.2.3

[19] Dhruv Ilesh Shah and Sebastian Scherer. Robust localization of an arbitrary distribution of radioactive sources. Unpublished paper, July 2017. 2.2.3, 6

[20] J.P. Nicolet and G. Erdi-Krausz. Data presentation and integration. In *Guidelines for Radioelement Mapping Using Gamma Ray Spectrometry Data*, pages 86–87. IAEA, 2003. 2.2.4, 2.2.4

[21] N. Cressis. The origins of kriging. *Mathematical Geology*, 22(3):239–251, 1990. 2.2.4

[22] Michael F. L'Annunziata. *Handbook of radioactivity analysis*. Amsterdam : Elsevier, 1998. 2.3.1

[23] Glenn F. Knoll. Geiger müller counters. In *Radiation Detection and Measurement*, pages 214–215. New York : Wiley, 1989. 2.3.1

[24] Glenn F. Knoll. Geiger müller counters. In *Radiation Detection and Measurement*, page 221. New York : Wiley, 1989. 2.3.3

[25] Glenn F. Knoll. Geiger müller counters. In *Radiation Detection and Measurement*, pages 353–354. New York : Wiley, 1989. 2.3.4

[26] T. Q. Farfan, E. B.and Foley, T Jannik, J.R. Gordon, L.J. Harpring, R. Blessing, S.J. Stanley, C.J. Holmes, M. Oldham, and J Adamovics. Methods and results. In *Testing of the Radball technology at Savannah River National Laboratory*. SAVANNAH RIVER NUCLEAR SOLUTIONS, 2018. 2.4.1, 9

[27] S Kronenberg, G.J Brucker, E Bechtel, and F Gentner. Directional detector for arrays of gamma ray and x-ray sources. *Nuclear Inst. and Methods in Physics Research, A*, 378(3):531–540, 1996. 2.4.2

[28] Jalil Islamian, Ahmadreza Azazrm, Babak Mahmoudian, and Esmail Gharapapagh. Advances in

pinhole and multi-pinhole collimators for single photon emission computed tomography imaging. *World Journal of Nuclear Medicine*, 14(1):3, 2015. 2.4.4

[29] Erik M. Brubaker. A maximum likelihood expectation maximization iterative image reconstruction technique for mask/anti-mask coded aperture data. In *2013 IEEE Nuclear Science Symposium and Medical Imaging Conference*, pages 1–3, 2013. 2.4.5

[30] Michal J. Cieslak, Kelum A.A. Gamage, and Robert Glover. Coded-aperture imaging systems: Past, present and future development - a review. *Radiation Measurements*, 92:59–71, 2016. 2.4.5

[31] Olivier Gal, Mehdi Gmar, Oleg P. Ivanov, Frederic Laine, Fabrice Lamadie, Christophe Le Goaller, Charly MahÃl', Erwan Manach, and Vyacheslav E. Stepanov. Development of a portable gamma camera with coded aperture. *Nuclear Inst. and Methods in Physics Research, A*, 563(1):233–237, 2006. 10

[32] Glenn F. Knoll. Geiger müller counters. In *Radiation Detection and Measurement*, page 349. New York : Wiley, 1989. 2.4.6

[33] Christopher G. Wahl, Willy Kaye, Weiyi Wang, Feng Zhang, Jason Jaworski, Y. Andy Boucher, Alexis King, and Zhong He. Polaris-h measurements and performance. In *2014 IEEE Nuclear Science Symposium and Medical Imaging Conference*, pages 1–4, 2014. 2.4.7

[34] Weiyi Wang, Willy R. Kaye, Jae Cheon Kim, Feng Zhang, and Zhong He. Improvement of compton imaging efficiency by using side-neighbor events. *Nuclear Inst. and Methods in Physics Research, A*, 687:62–68, 2012. 2.4.7

[35] Christopher G. Wahl, Willy R. Kaye, Weiyi Wang, Feng Zhang, Jason M. Jaworski, Alexis King, Y. Andy Boucher, and Zhong He. The polaris-h imaging spectrometer. *Nuclear Inst. and Methods in Physics Research, A*, 784:377–384, 2015. 2.4.7

[36] C. W. Akerlof, M. F. Cawley, M. Chantell, D. J. Fegan, K. Harris, A. M. Hillas, D. G. Jennings, R. C. Lamb, M. A. Lawrence, M. J. Lang, D. A. Lewis, D. I. Meyer, G. Mohanty, K. S. O'Flaherty, M. Punch, P. T. Reynolds, M. S. Schubnell, T. C. Weekes, and T. Whitaker. Locating very high energy gamma ray sources with arc minute accuracy. In *Proceedings of the 22nd International Cosmic Ray Conference*, volume 1, pages 399–405, 1991.

[37] S J Stanley, K Lennox, E B Farfan, J R Coleman, J Adamovics, A Thomas, and M Oldham. Locating, quantifying and characterizing radiation hazards in contaminated nuclear facilities using a novel passive non-electrical polymer based radiation imaging device. *Journal of Radiological Protection*, 32(2):131–145, 2012.

[38] K. Takeuchi, J. Kataoka, T. Nishiyama, T. Fujita, A. Kishimoto, S. Ohsuka, S. Nakamura, S. Adachi, M. Hirayanagi, T. Uchiyama, Y. Ishikawa, and T. Kato. "stereo compton cameras" for the 3-d localization of radioisotopes. *Nuclear Inst. and Methods in Physics Research*, 765: 187–191, 2014.

[39] B. Budden, G. L. Case, and M. L. Cherry. Noise-compensating algebraic reconstruction for a rotating modulation gamma-ray imager, 2010. e-Print: arXiv:1003.5223 [astro-ph.IM].