Creation of Synthetic X-Rays to Train a Neural Network to Detect Lung Cancer

Abhishek Moturu and Alex Chang

Department of Computer Science, University of Toronto, Toronto, Ontario, Canada

August 20, 2018

Abstract

The purpose of this research is to create effective training data for a neural network to detect lung cancer. Since X-rays are a relatively cheap and quick procedure that provide a preliminary look into a patient's lungs and real X-rays are often difficult to obtain due to privacy concerns, creating synthetic frontal chest X-rays using ray tracing and Beer's Law on several chest X-ray Computed Tomography (CT) scans with and without randomly inserted lung nodules can provide a large, diverse training dataset. This research project involves lung segmentation to separate lungs within CT scans and randomize nodule placement, nodule generation to grow nodules of random size and radiodensity, bone removal to obtain dual-energy X-rays, ray tracing to create X-rays from CT scans from several point sources using Beer's Law, image processing to produce realistic X-rays with uniform orientation, dimensions, and contrast, and analyzing these various methods and the results of the neural network to improve accuracy when compared to real X-rays, while reducing space complexity and time complexity. This research may be helpful in detecting lung cancer at a very early stage.

1 Introduction

1.1 Lung Cancer

Lung cancer is caused by abnormally behaving cells that grow too quickly or do not die regularly enough. Cancer cells grow into and destroy neighboring cells [1]. These cells form tumours and are commonly called nodules in Stage I when they are less than 3cm in diameter.

In Canada, as of 2018, lung cancer has the highest projected incidence and mortality rates of all cancers. It also has one of the lowest 5-year net survival rates as most diagnoses occur in later stages of the disease. There are 5 main stages of lung cancer: 0, I, II, III, and IV [2]. About 50% of lung cancer diagnoses occur during Stage IV, which has a high mortality rate [3]. However, early detection is critical for a good prognosis so this project considers the problem of detecting nodules in Stage I of growth.

1.2 Diagnosis

There are several tests that can either rule out or diagnose the existence or stage of lung cancer such as health history, blood tests, biopsies, and imaging techniques [4]. This project focuses on X-rays due to their relative affordability when compared to other imaging tests.

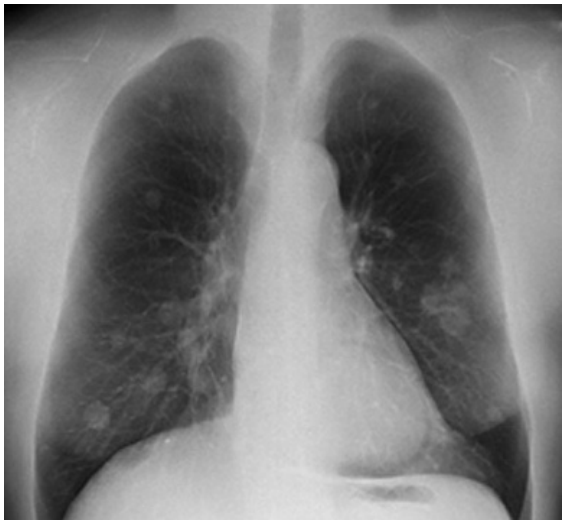

Figure 1: A real X-ray with multiple nodules. [5]

X-rays are grayscale images of a certain part of the body that capture the amount of X-ray radiation that is absorbed after various tissues within the body absorb the radiation at different levels. For example (See **Figure 1**), bones appear white since they absorb the most X-ray radiation, fat and soft tissues appear various shades of gray since they absorb lesser radiation, and air appears black since it absorbs almost no radiation. As a result, since lungs are filled with air, they appear dark in the X-rays. This makes it easier to spot nodules, which are

soft tissues, in the lungs, as opposed to those in other organs, as the air within the lungs "highlights" the nodules. However, the malignancy of a tumour cannot be determined solely from an X-ray and requires further tests [6].

Figure 2: A slice of a chest CT scan containing the lungs (dark regions) from a top view of the body.

CT scans are put together as a number of slices of X-rays which are combined to create a 3-dimensional model and provide a more comprehensive view of the inside of the body [7] (See **Figure 2**). For this project, using CT scans with or without inserted nodules as a model for a patient's body to synthesize the X-rays can provide a very large set of training data.

To get an unobstructed view of the lungs in the X-ray, bones are subtracted from the CT scan and the empty area is smoothed out (See **Section 5**). This method of assessing the soft tissue and bone separately by directing X-rays of two discrete energy levels at the body is called dual-energy X-ray absorptiometry (or DAX) [8]. In this project, the bone-subtracted soft tissue X-ray is used since the nodules may be distorted by the ribs otherwise.

1.3 Neural Networks

In the past few years, there has been vast interest and growing development in the field of Artificial Intelligence, specifically in the branch of Convolutional Neural Networks (CNNs), because of their many practical applications in various fields, especially Computer Vision. Well-trained CNNs are very useful in the field of medical imaging as they may be able to pick up details that human experts may have trouble discerning [9]. With this intention, effective training data is crucial in constructing a CNN that provides good results.

In previous research projects [10, 11], CNNs have provided promising results in detecting various abnormalities in frontal chest X-rays with significant accuracy. However, there is a lack of a large and diverse enough training database due to privacy concerns and the lack of sufficient, substantial variability within the data. So, creating synthetic X-rays allows for the control of many random variables through CT scan manipulation: location, radiodensity, size, and shape.

1.4 Project Input, Setup, Output

CT scans are the only inputs used to create the X-rays. The CT scans provided by Dr. Barfett are stored in Digital Imaging and Communications in Medicine (DICOM) files [12]. Respecting the privacy of the individuals who consented to provide their CT scans for this project, the CT scans have been anonymized. Horos, an open source medical image viewer, was used to inspect CT scans slice-by-slice [13].

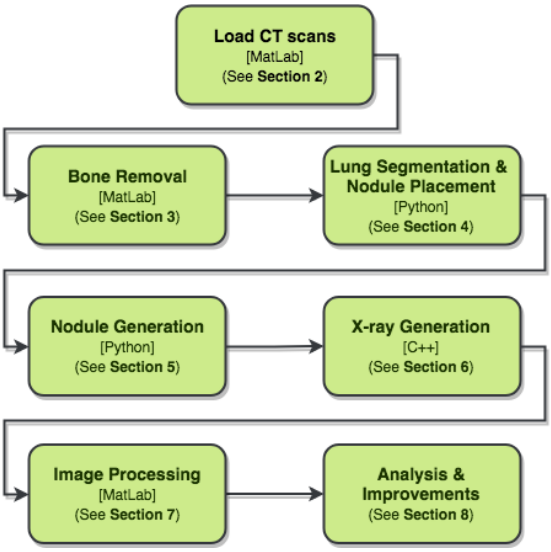

Figure 3: The project outline as of Aug. 20, 2018.

Different aspects of the project have been programmed respectively in the most suitable languages (See **Figure 3**), such as Python's NumPy [14] for handling large matrices, C++ for its flexible memory allocation, and MatLab for its Image Processing Toolbox [15].

The final X-rays for each CT scan are 256x256 pixel .png files. In the first batch, 6 CT scans were used to create 1991 X-rays (including 6 without nodules) using the parallel-ray approach with bones replaced with air, handpicked contrasts, and manual nodule placement. To improve the results of the GoogLeNet CNN for the second batch, many programs were modified, as discussed in this report, and 70 more CT scans are being exported to generate larger datasets. 30 more CT scans were obtained recently.

2 Attenuation & Radiodensity

Each pixel of an X-ray is measured using a linear attenuation coefficient (cm^{-1}), which is the nonnegative proportion of the number of photons in the X-ray that is absorbed per unit thickness of the medium [16]. Similarly, each volumetric pixel (voxel), the smallest part of a 3D object [17], of a CT scan is measured using Hounsfield Units (HU) or CT numbers, which are a dimensionless unit that linearly transform linear attenuation coefficients to the scale where water and air have radiodensity 0 and -1000 HU, respectively [18, 19].

$$HU_{medium} = 1000 \times \frac{(\mu_{medium} - \mu_{water})}{(\mu_{water} - \mu_{air})}$$

$$\mu_{medium} = \mu_{water} + \frac{HU_{medium}}{1000} \times (\mu_{water} - \mu_{air})$$

Figure 4: The linear transformation to convert from the attenuation of a medium, μ_{medium}, to radiodensity, HU_{medium}, and vice-versa.

It is necessary to be able to convert from radiodensity to attenuation since the voxels of the CT scans contain radiodensities in HU and generating X-rays requires work with attenuations (See **Figure 4**). Various tissues have varying attenuations and radiodensities (See **Table 1**).

Medium	Attenuation (cm^{-1})	Radiodensity (HU)
Bone	0.528	1000
Muscle	0.237	50
Blood	0.208	20
Water	0.206	0
Fat	0.185	-100
Lung	0.093	-200
Air	0.0004	-1000

Table 1: Some common media and their approximate attenuations and radiodensities [19].

3 Bone Removal

To mimic the effect of dual-energy X-rays (See **Figure 5**), which are created by projecting two X-rays of different energy levels to improve image contrast [20], bone voxels were modified before tracing rays through the CT. In the first batch of X-rays, voxels over an attenuation of 0.5 cm^{-1} [21, 22], which is approximately close to the attenuation of bone, were replaced with air, which has an attenuation of approximately 0 cm^{-1}). Although a large proportion of bones were removed, the substitution of bones with air was unrealistic because dark lines were noticeable in the generated X-rays. Some bones around the ribcage appeared to be remaining, giving an impression of hollow ribs.

Upon viewing attenuation plots of CT scan slices near the ribs, the attenuation threshold was decreased and the removed bone voxels were interpolated using linear interpolation [23] in 3 dimensions (trilinear interpolation) using MatLab's fillmissing function. However, this method produced more faint, but visible, bones in the X-rays. Finally, the best method was to reduce the attenuation threshold further and replace the bone voxels with water. See **Figure 6** for the three attempts to mimic soft-tissue X-rays.

4 Nodule Placement

The first set of synthetic X-rays with nodules were created with the brute-force approach. Future batches will use the lung segmentation algorithm to randomly select nodule positions.

4.1 Brute-Force Approach

To model the random positioning of nodule growth in the lungs, the first approach consisted of manually selecting points within the lungs with a modified version of a CT viewing tool developed by Dr. Barfett by scrolling through CT scan slices. Two points from each slice containing lung voxels were selected (one from each lung). Lung voxels were identified by darker shades of gray (See **Figure 2**) which had lower attenuations. Although the points appeared to be selected randomly, positions on slices with fewer lung voxels had the same probability of being chosen as those on slices with more lung voxels, which skews the data. Another downside with this approach is the time and labor required to select the positions in numerous CT scans.

4.2 Lung Segmentation Approach

The lung segmentation algorithm consists mainly of a Breadth-First Search call. A voxel belonging to each lung is manually selected similarly to the Brute-Force Approach and added to the queue as a starting point. Voxels in the queue are removed and their neighbours are added if their attenuations do not surpass a handpicked attenuation threshold associated with lung voxels. This process is repeated until no new voxels can be discovered from the lung voxels. The attenuation threshold was determined by analyzing attenuation graphs of the CT scan slices. In addition to the upper threshold, a similarly determined lower threshold was later added to prevent voxels in the trachea and the bronchi from being discovered [24]. The results of the lung segmentation algorithm (See **Algorithm 1**) can be seen in **Figure 7**. About 400 nodule centre positions are picked from anywhere within or on the borders of the segmented lungs using a uniform random distribution.

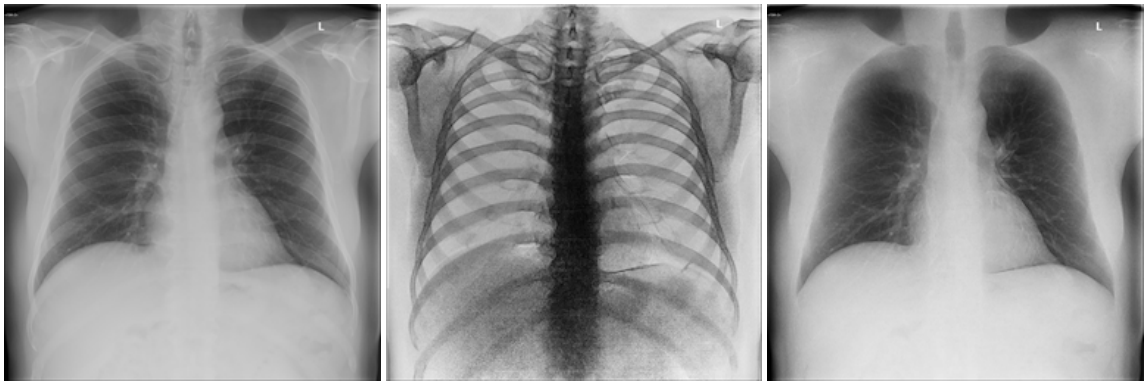

Figure 5: Real X-rays that show the effect of dual-energy X-rays. A standard X-ray (left). A negative X-ray highlighting the bones (middle). A soft-tissue X-ray with the bones subtracted (right).

Figure 6: Synthetic X-rays that mimic the effect of dual-energy X-rays. A standard X-ray (left). A soft-tissue X-ray with the bones removed and replaced with air (middle) and water (right).

```
def segment_lungs_bfs(ct, l_threshold, u_threshold, pts):
    def check_point(pt):
        if pt not in visited and \
                0 <= pt[0] <= ct.shape[0] - 1 and \
                0 <= pt[1] <= ct.shape[1] - 1 and \
                0 <= pt[2] <= ct.shape[2] - 1 and \
                l_threshold <= ct[pt + (0,)] <= u_threshold:
            q.append(pt)
            selected[pt] = 1
        visited[pt] = 1

    visited = {}
    selected = {}
    q = pts
    while len(q) > 0:
        slice_pt = q.pop()
        check_point(tuple(map(operator.add, slice_pt, (0, 0, 1))))
        check_point(tuple(map(operator.add, slice_pt, (0, 0, -1))))
        check_point(tuple(map(operator.add, slice_pt, (0, 1, 0))))
        check_point(tuple(map(operator.add, slice_pt, (0, -1, 0))))
        check_point(tuple(map(operator.add, slice_pt, (1, 0, 0))))
        check_point(tuple(map(operator.add, slice_pt, (-1, 0, 0))))
    return selected
```

Algorithm 1: Lung Segmentation algorithm (Python). The check_point function takes a point and checks if it falls in the given range of thresholds. The while loop explores all the voxels around the current point in the CT scan to check if they fall within the threshold using Breadth-First Search.

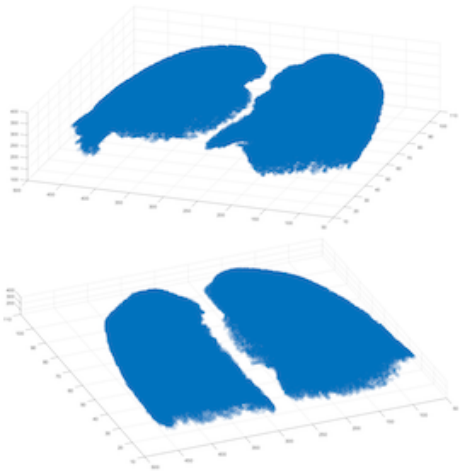

Figure 7: Points selected (in blue) using the CT scan lung segmentation algorithm from two angles.

5 Nodule Generation

Lung nodules come in various shapes, sizes, and radiodensities [25] (See **Figure 8**). The size of a lung nodule is usually between 0.5 cm to 3 cm, so it suffices to pick the size using a uniform random distribution. Similarly, the radiodensity of a nodule usually falls between 50 HU to 150 HU and is also picked using a uniform random distribution. However, generating various shapes of nodules is more challenging.

Growing the nodules to obtain realistic shapes (See **Figure 9**) took several tries. The initially explored dispersed model more closely simulated lung infections than lung cancer. Thus, this project uses the lobulated model, since it provided the most realistic shapes. Originally, the nodules were placed in the CT scans without resizing to match the dimensions of the lung voxels, making them appear stretched in the vertical direction. Resizing the nodules properly before inserting them into the CT scan fixed this issue, providing spherical nodules.

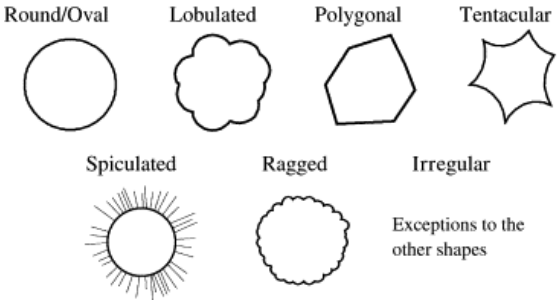

Figure 8: Various lung nodule shapes.

Since the X-rays are 256x256 pixel images and the nodules are relatively small within the X-rays, the nodules do not appear sharp enough in the X-rays for different types of nodule shapes to be distinguishable from one another. Thus, the lobulated model was used as an approximation for all of the lung nodule shapes.

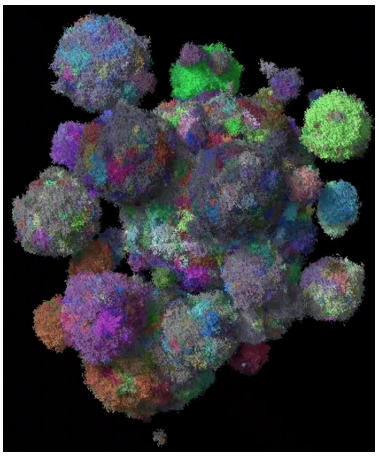

Figure 9: Accurate nodule growth model that accounts for the ability of cells to migrate locally and form microlesions, which are the smaller spheres around the central sphere of cancer cells [26].

5.1 Dispersed Model

Figure 10: X-ray with a nodule (dispersed model).

Based on [27], the first attempt was to generate nodules within a region by growing a central spherical nodule and growing smaller spherical nodules in that region until 7% of the volume in the region was filled to simulate the ability of cells to migrate locally and form "microlesions" (See **Figure 9**). According to Dr. Barfett, the resulting X-ray (See **Figure 10**) appeared like an X-ray of a lung infection rather than that of lung cancer. The generated nodules appeared "dispersed" throughout the region of interest (See **Figure 11**), rather than appear as one continuous, connected nodule.

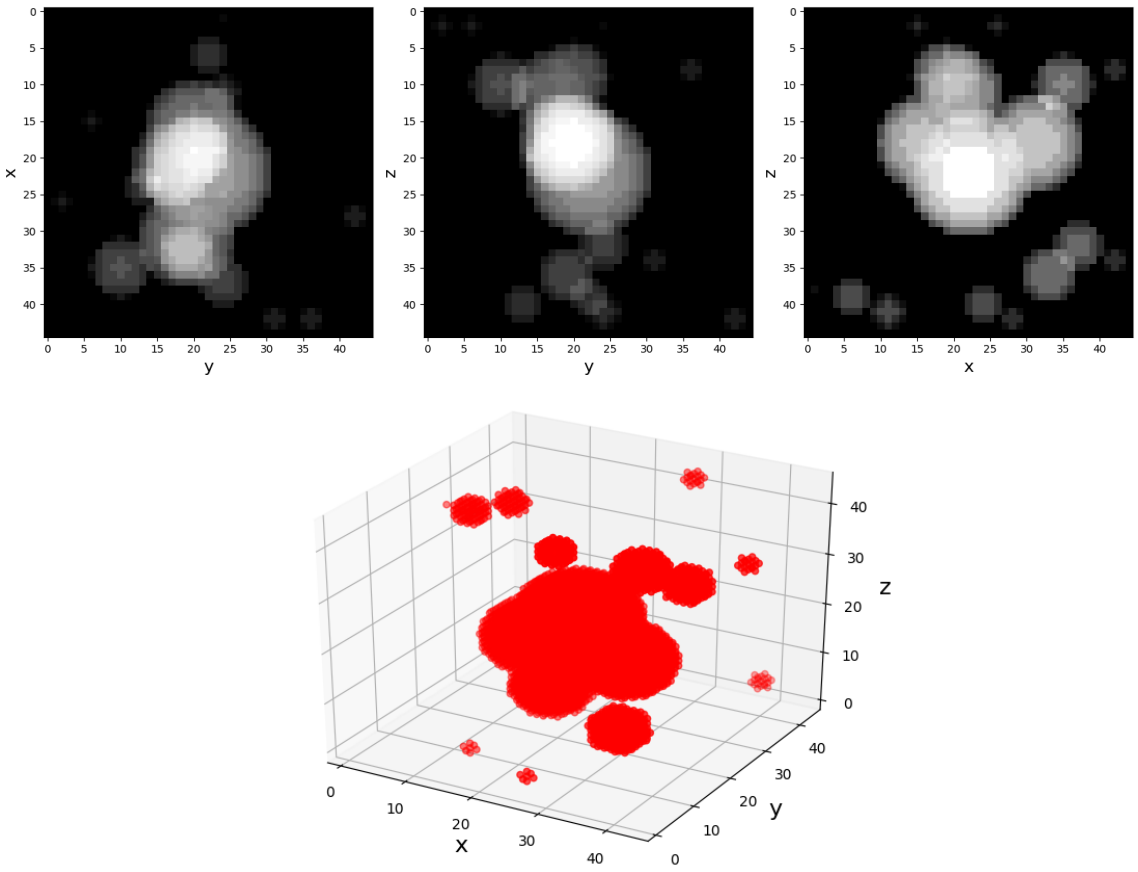

Figure 11: Growth of nodules in the dispersed model.

5.2 Lobulated Model

Figure 12: X-ray with a nodule (lobulated model).

A previous student worked on generating the lobulated model of a nodule. This model started with a central sphere of a randomly chosen diameter and iteratively added multiple hemispheres of smaller diameters on randomly chosen points on the surface of the existing shape until the required nodule diameter was reached in any one of the three dimensions. According to Dr. Barfett, the resulting X-ray (See **Figure 12**) appeared much more accurate. The generated nodules appeared as one continuous, connected nodule (See **Figure 13**).

The algorithm for the lobulated model (See **Algorithm 2**) chooses the radii of the smaller hemispheres (based on the centres) to be such that they grow no larger than the given desired size of the nodule. The nodule is grown in a cube of a slightly larger dimension for this reason, to allow enough space for growth.

6 Beer's Law

Once the generated nodules are inserted into the CT scan, the X-rays can be synthesized using Beer's Law (See **Section 6.3**). Two approaches were used to create the X-rays. The parallel-ray version is a simplification of the point-source version, which is a more accurate model of the X-ray procedure seen in practice. Both versions produce very similar results, except that rather than being 1 m away from the patient, the parallel-ray version "places" the point source infinitely far away from the patient. The point-source version also has the advantage that X-rays can be taken from different point sources for slight variations in the data.

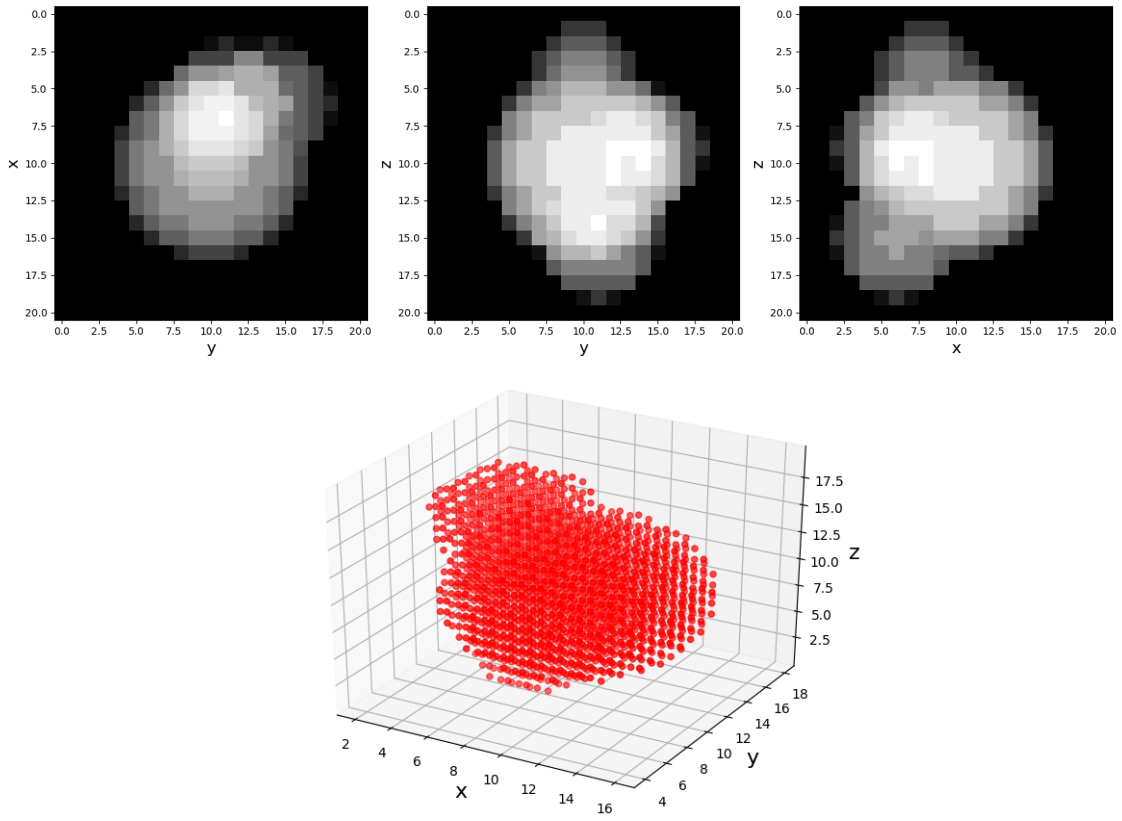

Figure 13: Growth of a nodule in the lobulated model.

```
def shape_grower(size, voxel_dimension):
    larger_dimensions = 3*[int(size/voxel_dimension)+3]
    dimensions = 3*[int(size/voxel_dimension)]
    base = np.zeros(larger_dimensions)

    centre = ()
    for i in range(len(larger_dimensions)):
        centre += (int(larger_dimensions[i]/2),) # Calculate the centre
    radius = np.random.uniform(0.5, 0.75) * \
    calculate_radius(larger_dimensions, centre) # Radius for initial centre

    # Create list of tuples of centre and radius
    cen_rads = []
    cen_rads.append((centre, radius))
    border_pts = []
    internal_points = []
    i = 0
    while not reach_desired_size(base, dimensions):
        # Poll a tuple from cen_rads list
        cen_rad_tuple = cen_rads[i]
        # Create sphere in the matrix
        border_pts = create_sphere(base, cen_rad_tuple[0], \
        cen_rad_tuple[1], 1, border_pts)
        i += 1
        if reach_desired_size(base, dimensions):
            break
        # Create a new centre and new radius
        new_cen, new_rad = find_new_centre_radius(cen_rad_tuple, \
        border_pts, larger_dimensions, size)
        cen_rads.append((new_cen, new_rad))
```

Algorithm 2: The nodule growing algorithm for the lobulated model. The calculate_radius function returns the largest possible radius for the given dimensions. The find_new_centre_radius function adds a new hemisphere of a random size and location centered on the existing shape.

6.1 Parallel-Ray Version

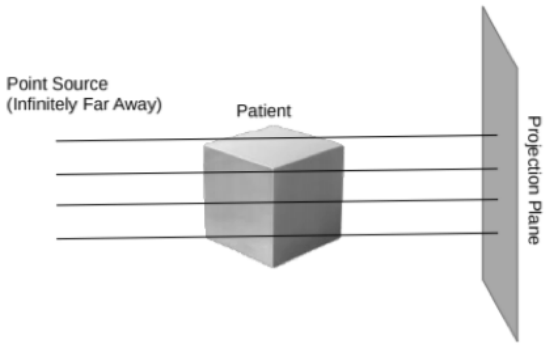

Figure 14: Parallel-ray method [28].

Parallel rays can be traced as an approximation to a real X-ray produced from an infinitely distant source point. Before the implementation of the efficient ray-tracing algorithm discussed in **Section 6.2.2**, this approximation greatly reduced the runtime as the intersecting distance of a ray through each voxel is constant (the intersecting distance is the voxel depth).

This method was used to produce the first batch of training data. However, after the first neural network training session, we hypothesized that the X-rays taken from the same position may contribute to the neural network recognizing patients and their lung shapes instead of nodule presence.

6.2 Point-Source Version

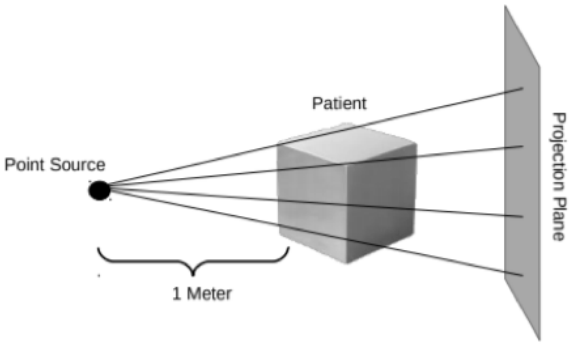

Figure 16: Point-source method [28].

To produce X-rays from a point source, the intersecting distance of each ray with each traversed voxel must be calculated. A 2D example of an efficient algorithm to accomplish this task is shown in **Figure 18**.

Starting from the entrance point, c, the points which correspond to the next voxel borders (one in each dimension) are calculated. The closest point, d, represents the next border intersection, and thus the distance from c to d is calculated. These steps are repeated with point d as the next entrance point.

The source point can also be changed to produce different perspectives of the same lungs as mentioned in **Section 6.2.1**.

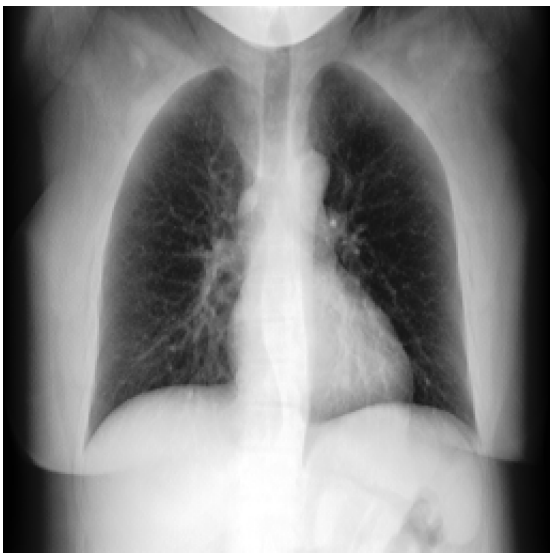

Figure 15: An X-ray generated using the parallel-ray version of X-ray generation.

Figure 17: An X-ray generated using the point-source version of X-ray generation.

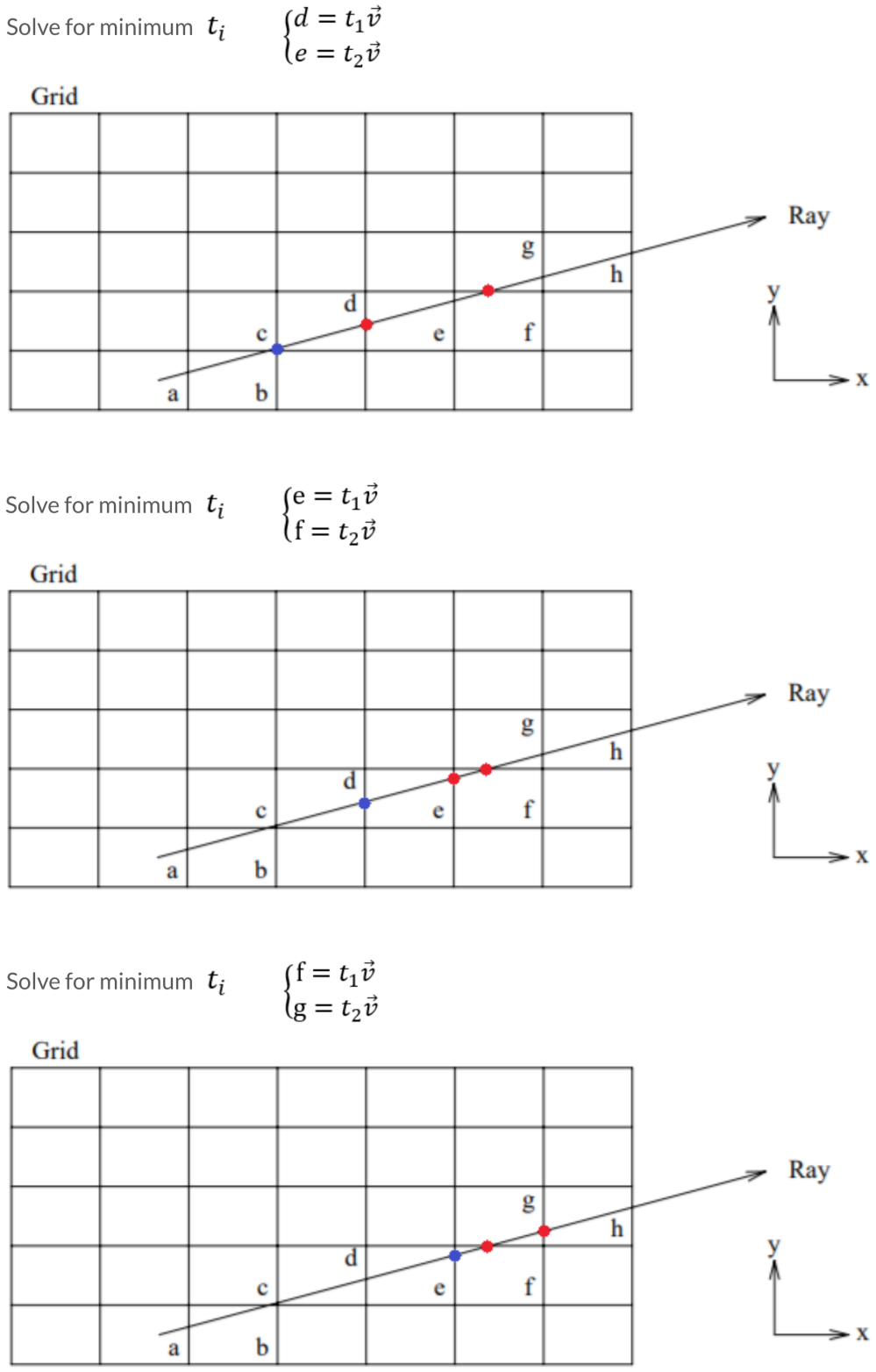

Figure 18: 2D example of the ray tracing algorithm. The points a to g represent voxel borders. The next border intersection points in each dimension (red) are calculated from the voxel entrance point (blue). The closest red point must be the voxel's exit point, and becomes the next voxel's entrance point. [29].

During the implementation of the ray traversal algorithm, the first approach was to rewrite the C++ program in Python due to its familiarity and ease of implementation. However, Python's generous memory allocation was problematic with the immense amount of CT voxel instantiation. The low-level programming provided by C++ was crucial in this step.

6.2.1 Randomized Point-Source

In order to prevent producing identical X-rays with nodule placement as the only variation in each CT scan, the next batch will also have varying source point positions. Each source point will be randomly chosen from a sphere surrounding the original source point (1 m away and centered on the front face of the CT scan), which will provide many different angles, adding variation to the data. With different source points, X-rays have slight variations (See **Figure 19**).

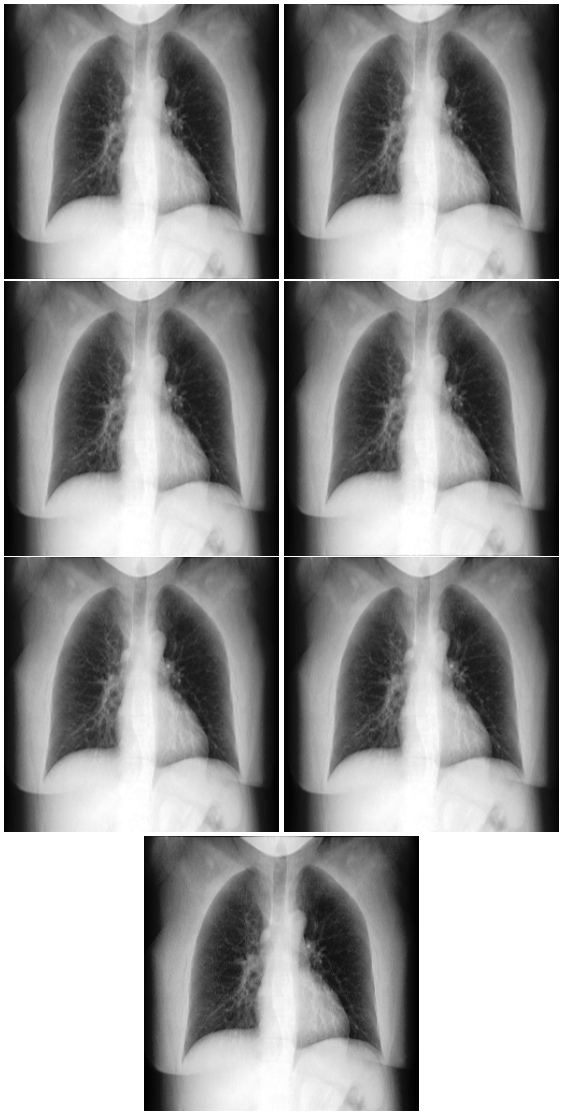

Figure 19: From top-left, X-rays generated by moving the point-source up, down, forward, backward, left, right, and leaving it centered at 1 m away.

6.2.2 Ray Tracing & Voxel Traversal

Upon implementing **Algorithm 3** in the source point X-ray generation program passed on from the previous student to obtain the voxels that were traversed by each ray and the intersecting distances, the runtime was significantly cut down from hours to under a minute, making mass production of point-source X-rays feasible.

For the upcoming batches, an X-ray with a nodule and a nodule-free X-ray will be produced from unique, randomly chosen angles. To reduce runtime, only the rays intersecting the modified nodule chunk will be traced.

6.3 Usage of Beer's Law

Once the sequence of voxels that intersect the rays are selected, the X-rays can be made using Beer's Law for both the parallel-ray version and the point-source version.

The measure of intensity loss of the X-rays as they pass through the body is modeled by Beer's Law (See **Figure 20**, line 1), which states that the rate of change of intensity per cm of an X-ray beam passing through a medium is jointly proportional to the intensity of the beam and to the attenuation coefficient of the medium [30].

Let I(x) be the intensity and A(x) be the attenuation of the x^{th} voxel in the sequence of voxels that the X-ray passes through. Let x_0 be the first voxel in the sequence and I(x_0) be the initial intensity. Let x_n be the final voxel in the sequence and I(x_n) be the final intensity. Note that Δx_i is the distance the X-ray travels through the i^{th} voxel (stays constant for parallel-ray version). Integrating is transformed into a summation since we do not have an attenuation function, but rather a list of attenuations.

$$\frac{dI}{dx} = -A(x) \cdot I(x)$$

$$\implies \frac{dI}{I} = -A(x)dx$$

$$\implies \int_{x_0}^{x_n} \frac{dI}{I} = -\int_{x_0}^{x_n} A(x)dx$$

$$\implies ln(I(x_n)) - ln(I(x_0)) = -\sum_{i=0}^{n} A(i)\Delta x_i$$

$$\implies ln(\frac{I(x_n)}{I(x_0)}) = -\sum_{i=0}^{n} A(i)\Delta x_i$$

$$\implies \frac{I(x_n)}{I(x_0)} = e^{-\sum_{i=0}^{n} A(i)\Delta x_i}$$

$$\implies I(x_n) = e^{-\sum_{i=0}^{n} A(i)\Delta x_i} \cdot I(x_0)$$

Figure 20: Starting with Beer's Law, we obtain an equation for the final intensity of a pixel [28, 30].

Note that the above method calculates the remaining intensities of the X-rays as they travel through the body as opposed to the absorbed intensities (See **Section 7**, invert colours).

```
double traverseVoxel(Coordinate *prev, double *t, int dim,
                     Coordinate *coordArray, SimulatedRay *ray,
                     vector<vector<vector<Voxel *>>> &ctVoxels,
                     double voxel_xy_dim, double voxel_z_dim){

    // Obtain the lost intensity in the traversal of this voxel.
    double localIntensityLoss = getIntensityLoss(*prev, coordArray[dim],
                                ctVoxels, voxel_xy_dim, voxel_z_dim);
    // find the next integer point for dimension dim
    double nextInt;
    if (dim == XDIM){
        nextInt = coordArray[dim].x + ray->xSign;
    } else if (dim == YDIM){
        nextInt = coordArray[dim].y + ray->ySign;
    } else {
        nextInt = coordArray[dim].z + ray->zSign;
    }
    // Update the next (integer) exit point for this dimension.
    *prev = coordArray[dim];
    *t = findT(ray, dim, nextInt);
    newCoordinate(ray, *t, coordArray[dim]);
    return localIntensityLoss;
}
```

Algorithm 3: This function shows how to traverse a single voxel. The parameter "dim" specifies the next traversed dimension. The intensity loss caused by the voxel with entrance point "prev" and exit point "coordArray[dim]" is calculated. coordArray[dim] is then updated to prepare for the next iteration.

7 Image Processing

The final part of generating accurate-looking synthetic X-rays requires image processing. Because of the way the CT scan is oriented and stored, there are several image processing techniques (most of which are found in MatLab's Image Processing Toolbox) that we have to use to make the X-ray look realistic and ensure that the dimensions, orientation, and contrast are as close to those of a real X-ray.

When making the X-ray, we measure the amount of energy left, so the colours must be inverted to get the attenuation, which is the amount of energy absorbed. Inverting the colours of the matrix simply requires assigning the matrix to one minus itself (See **Figure 21**).

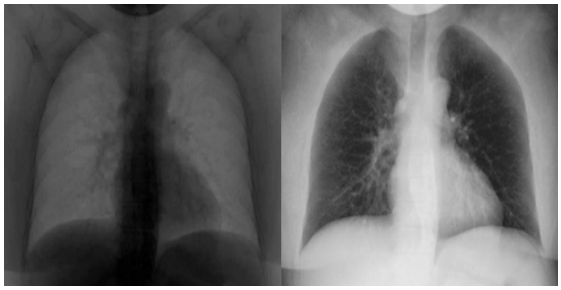

Figure 21: X-ray with inverted grayscale.

The functions rot90 (See **Figure 22**) and flip (See **Figure 23**) were used to orient the image with the lungs upright and the heart on the right side. These functions manipulate the matrix of pixels to obtain the desired orientation.

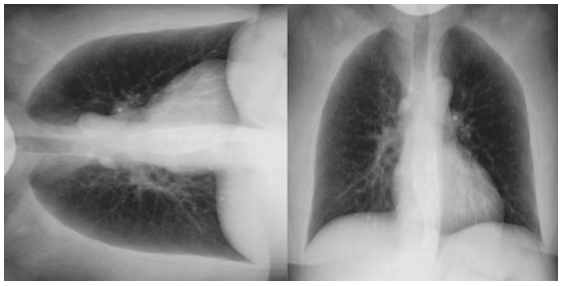

Figure 22: X-ray rotated by 90 degrees clockwise.

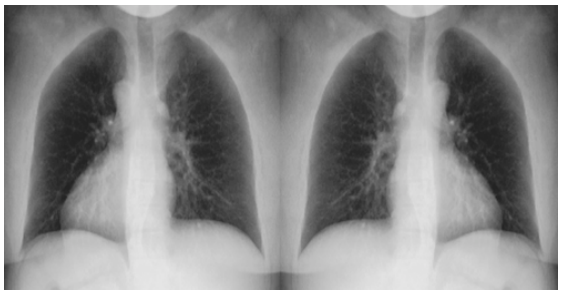

Figure 23: X-ray flipped along the vertical axis.

For the parallel-ray version of X-ray generation, resizing the images is necessary to get the desired size due to the shapes of the CT scans.

However, the most challenging part of image processing was accurately portraying the contrast within the X-rays.

7.1 Resizing with Interpolation

While generating images using the parallel-ray version, which essentially flattens the CT scans in one dimension to make the X-rays, resizing the image is necessary to create images of size 256x256 since each CT scan may have different dimensions (See **Figure 24**). CT scan slices are always 512x512 pixels, however the number of slices in each CT scan may vary from 116 to 527 based on the thickness of each slice.

The MatLab function imresize uses bilinear interpolation to fill gaps in the data or shrink the data to resize the images as desired.

Note that interpolation is not required for the point-source version as we can control the number of rays that are projected onto the 256x256 pixel image.

Figure 24: Before and after using linear interpolation in two dimensions to stretch the image vertically and shrink it horizontally. The original dimensions of the CT were 125x512x512 voxels which resulted in the dimensions of the X-ray being 125x512 pixels. After bilinear interpolation, the dimensions are corrected to be 256x256 pixels.

7.2 Contrast Enhancement

There were many attempts to get the contrast to look realistic. After using Beer's Law to create the X-rays, the X-rays had very poor contrast (See **Figure 25**). They appeared bright and faded unlike real X-rays which had lungs appearing very dark and the surrounding tissues appearing brighter, in a light gray colour.

Exploring the notion of histograms, which are representations of the distribution of a set of data [31], and in this case the distribution of the pixel intensities in an image, it was possible to fix the contrast.

First, the imadjust function was used to manually find the right contrast by eye. This method was prone to error and was very time consuming.

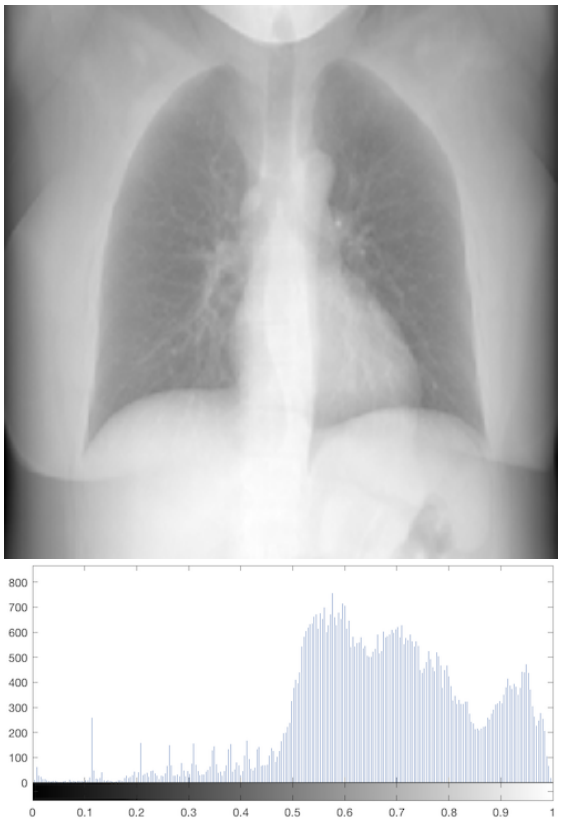

Figure 25: Original image and its corresponding histogram of intensities.

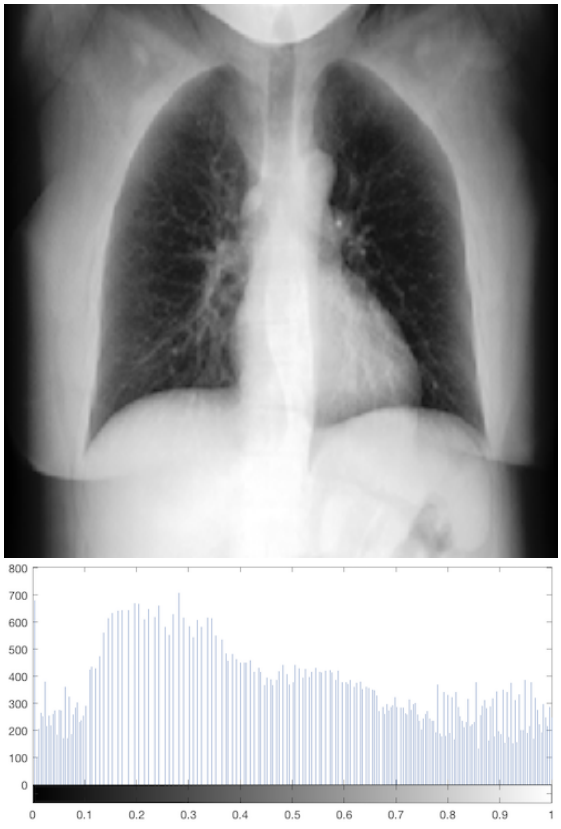

Figure 26: Gamma corrected, histogram equalized image and its corresponding histogram of intensities.

Second, the imhistmatch function was used to adjust the histogram of the synthetic X-ray to match the histogram of the real X-ray provided by Dr. Barfett. This method failed due to the differences in the size of the dark borders in the images. These borders skewed each of the CT scans differently based on the size of the patient and how they lay on the table during the CT scan procedure.

Then, we considered manipulating the borders of the X-ray by adding some dark pixels near the edges so that all X-rays have approximately the same distribution of dark pixels in the histograms on which we would then use imhistmatch. However, this procedure was prone to error and also very time consuming.

Finally, the best method to improve contrast was to use gamma correction (gamma = 2.5) [32] to weigh the mapping towards darker values and then use histogram equalization to stretch out the intensities in the image to the whole range of intensities, making the dark parts darker to produce dark lungs and bright parts brighter to produce bright surrounding tissues [33]. According to Dr. Barfett, this process made the images look much more realistic (See **Figure 26**). The MatLab functions imadjust with a gamma parameter of 2.5 and histeq with a bins parameter of 256 (for 256 intensity values) were used to improve the contrast.

8 Analysis & Improvements

To create nodules and place them in CT scans to produce X-rays, the first step was to choose positions for the nodules.

Manually scrolling through each slice of the lungs and picking a random location on the left and right lung would take many hours for several CT scans. Thus, switching to using lung segmentation to select points within the lungs and then randomly picking some of those points saves a lot of time.

Initially, there were empty lines in the X-rays. This bug occurred in the dicomHandler MatLab file and the chestCTscroller MatLab file since during initialization, more space was allocated than necessary for the slices, leaving some of them unfilled at the end. Removing the empty rows at the end fixed this issue.

Reducing space and time complexity was essential in producing a large number of X-rays efficiently. The main improvement in saving space and time was achieved by making empty X-rays from empty CT scans first and then only tracing around inserted nodules in the CT scan to update certain "chunks" of the X-ray images.

To improve accuracy, along with fixing contrast and removing bones, we have also worked on improving the implementation of the point-source ray tracing algorithm rather than use the parallel-ray algorithm since, in practice, X-rays are taken from a point-source that is usually around 1 meter away from the patient. Moving the point-source around in a ball of radius 10cm allows different perspectives of the same lungs, adding variation to the data but with the drawback that it increases the amount of space and time used to create the X-rays.

9 Future Work

There are various topics to explore for future work on this project. After obtaining 100 more CT scans, we hope to inspect and remove any corrupted or suboptimal CT scans to create approximately 50000 X-rays of training data. This can help us understand if we are making progress on the path towards making effective synthetic training data.

Another idea is to perform histogram equalization on real X-rays to be part of the training data as well. This might help make the contrast of synthetic and real X-rays match more closely when training the neural network.

On a similar note, segmenting the lungs in real X-ray libraries can aid in increasing focus analysis & detection of nodules. Since the lungs are the only organ being considered for this project, it may be sufficient to train the neural network using cropped images of the lungs rather than including parts of the abdomen on the bottom, air on the side, or collarbones on the top in the X-rays. It may be useful to consider the attenuation of lungs within the X-rays or the statistics of lung dimensions [34] to crop around the lungs accordingly.

Additionally, along with the frontal X-rays, it may be useful to train the neural network with lateral X-rays as that may reduce the possibility of nodules blending in with surrounding tissues.

Other ideas include trying different methods of nodule generation for more variation in shape and creating synthetic training data for various diseases with those shape growing algorithms. It may also be helpful to use a convex hull to smooth out any sharp edges or corners [35] in the generated shapes.

On the logistical side, considering ways of creating a balanced training dataset based on prevalence of various aspects like size, shape, location, or radiodensity may help the neural network identify nodules more naturally.

And finally, implementing a Generative Adversarial Network (GAN) may generate more realistic synthetic X-rays. Previous research with GANs on frontal chest X-rays has shown promising results for various lung abnormalities [36].

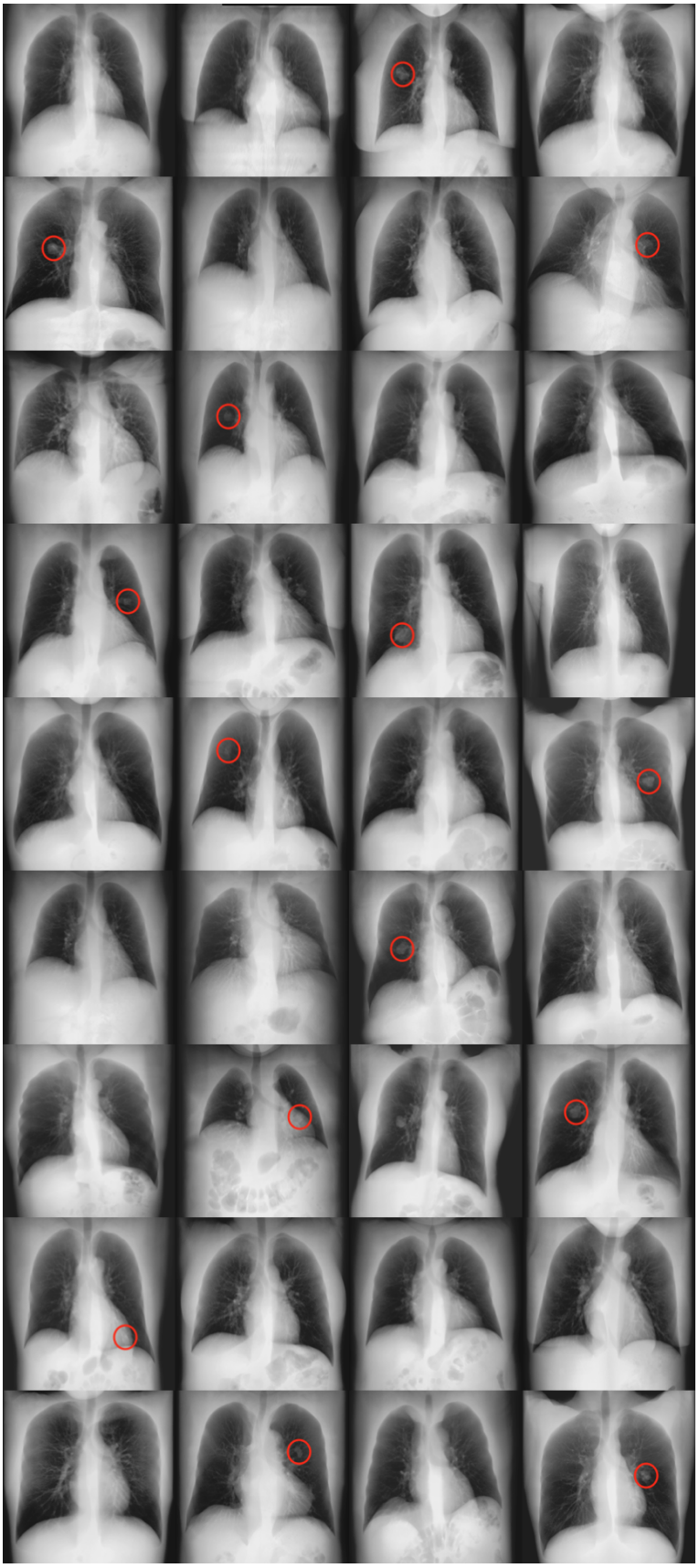

Figure 27: Examples of an assortment of synthetic X-rays, with and without nodules, generated from the 36 CT scans that we currently possess. Nodules are circled in red. Note that some of the nodules are hard to see by eye as they might be hiding behind the heart, might be nodules on the borders of the lungs (snowball lesions), might have low radiodensity, or might be blending into the surrounding tissues.

Appendix

The following link is a private repository that contains the regularly updated programs and details on running them: https://goo.gl/zLtPvd. A Bitbucket account is needed to access this private repository. Please contact abhishek.moturu or le.chang [at] mail.utoronto.ca for access.

Acknowledgments

Thanks to those who volunteered to provide their chest CT scans for this research. Thanks also to Donna Tjandra & Weicheng Cao for their initial work on the project, Hojjat Salehinejad for his work on the neural network, Dr. Errol Colak & Hui-Ming Lin for their help in exporting CT scans, and Gavin Barill. We are grateful to Prof. Kenneth Jackson & Dr. Joseph Barfett for their supervision and insights. We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) and the University of Toronto.

References

[1] "What is lung cancer? - canadian cancer society." http://www.cancer.ca/en/cancer-information/cancer-type/lung/lung-cancer/?region=on. (Accessed on 08/19/2018).

[2] "Lung cancer - non-small cell: Stages | cancer.net." https://www.cancer.net/cancer-types/lung-cancer-non-small-cell/stages. (Accessed on 08/19/2018).

[3] "Canadian-cancer-statistics-2018-en.pdf." http://www.cancer.ca/~/media/cancer.ca/CW/cancer%20information/cancer%20101/Canadian%20cancer%20statistics/Canadian-Cancer-Statistics-2018-EN.pdf. (Accessed on 08/19/2018).

[4] "Diagnosis of lung cancer - canadian cancer society." http://www.cancer.ca/en/cancer-information/cancer-type/lung/diagnosis/?region=on. (Accessed on 08/19/2018).

[5] "Dual energy subtraction - advanced applications - radiography - categories." http://www3.gehealthcare.com.au/en-au/products/categories/radiography/advanced_applications/dual_energy_subtraction#tabs/tab68F9F73A583A457D9D28B5E3306A6700. (Accessed on 08/19/2018).

[6] "X-ray - canadian cancer society." http://www.cancer.ca/en/cancer-information/diagnosis-and-treatment/tests-and-procedures/x-ray/?region=on. (Accessed on 08/19/2018).

[7] "Computed tomography (ct) scan - canadian cancer society." http://www.cancer.ca/en/cancer-information/diagnosis-and-treatment/tests-and-procedures/computed-tomography-ct-scan/?region=on. (Accessed on 08/19/2018).

[8] H. C. Lukaski, "Soft tissue composition and bone mineral status: evaluation by dual-energy x-ray absorptiometry," *The Journal of nutrition*, vol. 123, no. suppl_2, pp. 438–443, 1993.

[9] J. L. Patel and R. K. Goyal, "Applications of artificial neural networks in medical science," *Current clinical pharmacology*, vol. 2, no. 3, pp. 217–226, 2007.

[10] M. Cicero, A. Bilbily, E. Colak, T. Dowdell, B. Gray, K. Perampaladas, and J. Barfett, "Training and validating a deep convolutional neural network for computer-aided detection and classification of abnormalities on frontal chest radiographs," *Investigative radiology*, vol. 52, no. 5, pp. 281–287, 2017.

[11] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *CoRR*, vol. abs/1711.05225, 2017.

[12] "1scope and field of application." http://dicom.nema.org/medical/dicom/current/output/chtml/part01/chapter_1.html#sect_1.1. (Accessed on 08/19/2018).

[13] "About – horos project." https://horosproject.org/about/. (Accessed on 08/19/2018).

[14] "Numpy — numpy." http://www.numpy.org/. (Accessed on 08/19/2018).

[15] "Image processing toolbox documentation." https://www.mathworks.com/help/images/. (Accessed on 08/19/2018).

[16] "Attenuation coefficient." https://www.nde-ed.org/EducationResources/CommunityCollege/Radiography/Physics/attenuationCoef.htm. (Accessed on 08/19/2018).

[17] "What is a volume pixel (volume pixel or voxel)? - definition from techopedia." https://www.techopedia.com/definition/2055/volume-pixel-volume-pixel-or-voxel. (Accessed on 08/19/2018).

[18] "Hounsfield unit | radiology reference article | radiopaedia.org." https://radiopaedia.org/articles/hounsfield-unit. (Accessed on 08/19/2018).

[19] "Ct basics and pet attenuation." http://www.people.vcu.edu/~mhcrosthwait/clrs322/petctdata.html. (Accessed on 08/19/2018).

[20] "Ct physics: Dual-energy ct - xrayphysics." http://xrayphysics.com/dual_energy.html. (Accessed on 08/19/2018).

[21] *Computed Tomography - Pageburst E-book on Kno Retail Access Card Physical Principles, Clinical Applications, and Quality Control*. W B Saunders Co, 2015. pgs. 58-59.

[22] V. Rebuffel and J.-M. Dinten, "Dual-energy x-ray imaging: benefits and limits," *Insight-non-destructive testing and condition monitoring*, vol. 49, no. 10, pp. 589–594, 2007.

[23] "Linear interpolation." http://www.eng.fsu.edu/~dommelen/courses/eml3100/aids/intpol/index.html. (Accessed on 08/19/2018).

[24] W. Li, S. D. Nie, and J. J. Cheng, "A fast automatic method of lung segmentation in ct images using mathematical morphology," in *World Congress on Medical Physics and Biomedical Engineering 2006* (R. Magjarevic and J. H. Nagel, eds.), (Berlin, Heidelberg), pp. 2419–2422, Springer Berlin Heidelberg, 2007.

[25] S. Iwano, T. Nakamura, Y. Kamioka, and T. Ishigaki, "Computer-aided diagnosis: A shape classification of pulmonary nodules imaged by high-resolution ct," *Computerized Medical Imaging and Graphics*, vol. 29, no. 7, pp. 565 – 570, 2005.

[26] "New model captures shape and speed of tumor growth for the first time." https://goo.gl/suF22x. (Accessed on 08/19/2018).

[27] B. Waclaw, I. Bozic, M. E. Pittman, R. H. Hruban, B. Vogelstein, and M. A. Nowak, "A spatial model predicts that dispersal and cell turnover limit intra-tumour heterogeneity," *Nature*, vol. 525, no. 7568, p. 261, 2015.

[28] D. Tjandra, W. Cao, J. Barfett, and K. Jackson, "Synthesizing training data for a deep convolutional network to detect abnormalities in frontal chest radiographs,"

[29] J. Amanatides, A. Woo, *et al.*, "A fast voxel traversal algorithm for ray tracing," in *Eurographics*, vol. 87, pp. 3–10, 1987.

[30] T. G. Feeman, *The Mathematics of Medical Imaging*. Springer International Publishing, 2015.

[31] "Histograms - understanding the properties of histograms, what they show, and when and how to use them | laerd statistics." https://statistics.laerd.com/statistical-guides/understanding-histograms.php. (Accessed on 08/19/2018).

[32] "Gamma correction - matlab & simulink." https://www.mathworks.com/help/images/gamma-correction.html. (Accessed on 08/19/2018).

[33] "hist_eq.dvi." https://www.math.uci.edu/icamp/courses/math77c/demos/hist_eq.pdf. (Accessed on 08/19/2018).

[34] G. H. Kramer, K. Capello, B. Bearrs, A. Lauzon, and L. Normandeau, "Linear dimensions and volumes of human lungs obtained from ct images," *Health physics*, vol. 102, no. 4, pp. 378–383, 2012.

[35] "Drawing boundaries in python." http://blog.thehumangeo.com/2014/05/12/drawing-boundaries-in-python/. (Accessed on 08/19/2018).

[36] H. Salehinejad, S. Valaee, T. Dowdell, E. Colak, and J. Barfett, "Generalization of deep neural networks for chest pathology classification in x-rays using generative adversarial networks," *arXiv preprint arXiv:1712.01636*, 2017.