Reconstructing Images from Projections

Vishal Chandra (chandrav), Sasha Mittal (msasha), Abhinav Thakur (tabhinav)
University of Michigan – Ann Arbor
chandrav@umich.edu msasha@umich.edu tabhinav@umich.edu

1. Introduction

The question of how to capture not only what we see in everyday life, but also well beyond that, has enamored people for some time. While it is impressive that we can capture the world around us at all, we are faced with how to extend our sight using computer imaging techniques in many applications. Capturing the internal structure of objects is the motivating problem for this work, and reveals an incredibly rich domain of research in tomographic reconstruction.

We approach this problem by sending high-frequency radiation, often X-rays, through the objects we want to analyze. Because this high-energy light can pass through most materials, we can measure its intensity on the other side of the object. However, such a process will, at most, produce a projection of the object in the plane or on a line. To acquire more information about the object in question, we can rotate it and repeat this process and continue until we have rotated 180º before we start seeing flipped copies of existing data.

The problem that arises is how to compose data collected in this form to understand the internal structure of the object under experiment.This problem may seem purely academic but actually has a considerable impact on science, business, and medicine. This paper focuses on the medical side of this in applications such as Computed Tomography (CT) or Magnetic Resonance Imaging (MRI). With these systems, safely sending radiation or magnetism through the human body can generate these projections, which we can then reconstruct to see a whole world of new images.

Depending on the approach, reconstruction from projection comes down to either (1) a signal processing problem, (2) an iterative error minimization problem, or (3) a deep learning problem to translate between projection and spatial domain data. Each of these 3 approaches is closely related to methods in other areas of computer vision, and the richness of this problem yields multiple ways to approach it. It is interesting that one problem can motivate so many different methods, and that continuous advances are being made to invert the operation of projection.

In surveying three methods that correspond to the aforementioned three perspectives on this problem, we find the work of some groups to be particularly relevant. Deans (1984) approaches this problem as inverting the mathematical Radon transform and investigates the signal processing challenges that come with that perspective. Hounsfield and Gordon et al. originally approached this problem iteratively and from an error minimization standpoint. Their work was key in the first-ever CT scans and is the basis for our implementation of iterative algorithms. Finally, we acknowledge Würfl et al. for their work in Deep Learning Computed Tomography. Their perspective brings a more modern approach to a long standing problem, and inspired us to also investigate using deep learning for image reconstruction.

To understand the evolution of these algorithms, from all three perspectives presented, our approach was three-pronged.We first approached the problem geometrically, considering projection angle and ray direction. Next, we approached the problem purely numerically, beginning with a random image and minimizing discrepancies with projection data iteratively. Finally, we implemented a deep learning model to learn the correspondences between projection and image data.

2. Approach

This research paper presents a comprehensive survey of three distinct techniques for Image Reconstruction: filtered backprojection, algebraic reconstruction technique, and convolutional neural network. Below is the approach we took in understanding and implementing each of these three methods.

2.1. Filtered Backprojection

As the projections accumulate from different angles, they are organized in a figure called a sinogram.

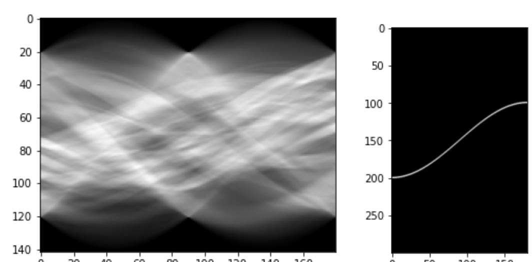

Figure 1: An example sinogram (left), sinogram for a single point (right).

Figure 1 shows the sinogram for a full image as well as the sinogram for a single white point in a fully black image. The right side of the figure gives us some intuition for why the content of a sinogram appears so sinusoidal, hence the name. Filtered backprojection works by looking at each projection (1 projection per deg, here and hereafter) and “undoing” the projection that produced it, after some sharpening operations.

First, the sinogram is filtered to remove low-frequency components. To do this, the fast Fourier transform of each sinogram row is taken, a ramp filter is applied, and the transform is reversed. From the theory of signals, we know that the ramp filter, described in equation (1) below, approximates a high-pass filter.

$$row(x) = \mathcal{F}^{-1}\{\mathcal{F}\{row(x)\}|\omega|\} \quad (1)$$

Without the filtering operation described in equation (1), the reconstruction using the following steps produces a blurry result due to a phenomenon called “1/r blurring” that occurs when combining projections.

After the filtering operation, shown in Figure 2, the sinogram is ready to be backprojected across the image plane. To do this, we compute which direction the projection came from through some conversion factor. In this case 1 sample was taken per degree, so the projection number corresponded directly to the orientation of the projection. Row by row, each projection is re-oriented to the direction it was taken and “smeared” across the reconstructed image. This process can be seen for a single row in Figure 3.

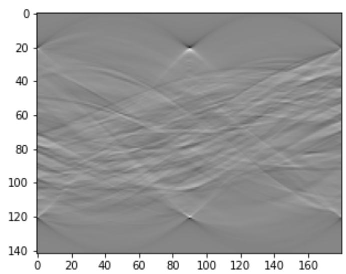

Figure 2: The sinogram from figure 1 after the filtering operation.

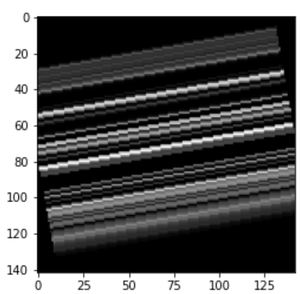

Figure 3: A single sinogram row “smeared” across the reconstructed image.

When this is done for all rows in the sinogram and the “smears” are added together, an approximation of the original image takes form.

2.2. Algebraic Reconstruction Technique (ART)

Next, we considered an older algorithm which views image reconstruction as an optimization problem.

This approach requires somewhat of an understanding of operators and the objects they act on, as it treats both projection and (unfiltered) backprojection as linear operators A and A^T, respectively, on images. It also provides an interesting perspective on the problem: we are trying to iteratively invert operator *A*, and it becomes clear that the transpose operation is significantly different. This provides some insight about Filtered Backprojection as well, as it shows that simply backprojecting is not enough, and some other steps (filtering) must be taken to better approximate *A^-1*.

We begin with an arbitrary image as our guess for the reconstruction (here, we simply begin with a black image – all zeros). From there, we perform projections of this guessed image to produce a guessed sinogram, which we can compare to the sinogram data that we have been given. By taking the difference of the two sinograms, we obtain a new ‘difference sinogram,’ which we can then backproject to the image space and add to our current estimate. This process continues until a desirable result is reached. The algorithm is described in equation (2) below.

$$x^{k+1} = x^k + \lambda_k \frac{A^T(b - Ax^k)}{A^T A(1)} \quad (2)$$

In equation 2, *x* is our estimate of the image, *b* is the given sinogram, *A* is the projection operator, *A^T* is the backprojection operator, and λ is a learning rate parameter similar to gradient descent.

To implement these abstract operators *A* and *A^T*, we adapted the algorithm to work not with vectors and matrices, but instead with images and projection/backprojection code from our work on filtered backprojection. Specifically, *A* represents an application of the projection function to create a sinogram, and *A^T* represents the backprojection function using repeated smearing. It is often the case in ART implementations that these operators are hidden away with external libraries, but it was important here and part of our value-add through this project to use transparent code from the top down. Following that ideal, we treated ART as a higher-order algorithm acting as a controller for the backprojection algorithm from 2.1.

We can see that the denominator of the fraction does not change from iteration to iteration, and represents what happens to an image of all 1s (all white) when projected to a sinogram and then backprojected.

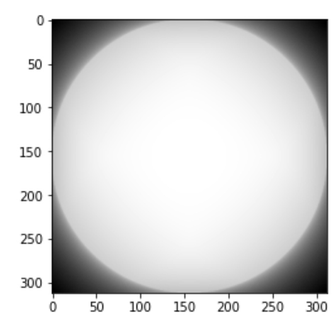

Figure 4: The denominator image in ART.

Figure 4 shows how the composed operation A^T(A) impacts a pure white image. It is clear that intensities near the corners are attenuated considerably. In some sense, this is a kind of “eigenvalue” image for the operator A^T(A), as it shows the scaling applied to each location. To correct for this, we divide by this scaling image after generating our “update image” in the numerator.

Figure 5 shows ART reconstructions of the famous Shepp-Logan phantom after one and 200 iterations.

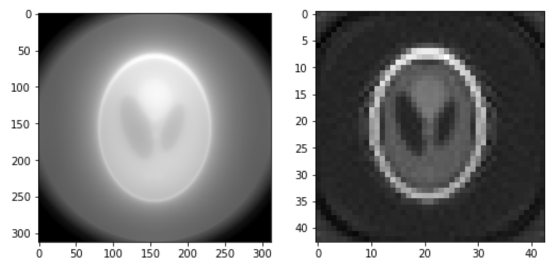

Figure 5: ART reconstruction of Shepp-Logan phantom after one (left) and 200 (right) iterations.

2.3. Convolutional Neural Network

Convolutional Neural Networks are a relatively new technique. Convolutional Neural Networks are a type of neural network designed to work with structured data such as images. Through a series of convolutional layers, the network learns to extract and identify key features from an input image. These features are then passed through a series of layers in the model, ranging from 2D convolutional layers to fully connected layers to make predictions based on the input data.

Thus far in computer vision, Convolutional Neural Networks have been used in a wide range of applications. This includes self-driving cards, medical imaging, and facial recognition.

Our goal with the Convolutional Neural Network was to learn the mapping between the sinogram of an image and the reconstructed image. Once trained, the network would be able to reconstruct high-quality images, simply from the sinogram of the image. While this technique can be challenging to implement, it has the potential to produce some of the best reconstructed images in limited conditions.

The model architecture includes a series of convolutional layers, followed by a fully connected layer, and then a rectified linear unit layer. As described in the [8], the following figure is a visual representation of a parallel beam architecture.

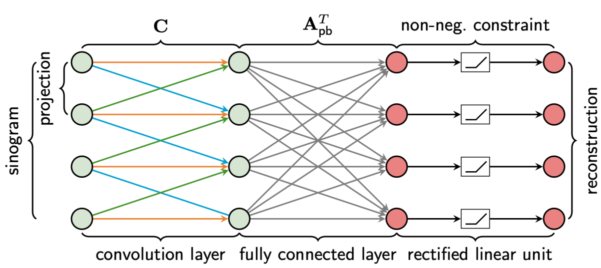

Figure 6: CNN architecture for image reconstruction.

The implementation of the model involes utilizing the Keras and TensorFlow library. The input images were first preprocessed before being split into testing and validation sets. The preprocessing includes ensuring that all the images were all the size and normalizing pixel values to between 0 and 1.

The model was compiled with a mean squared error loss function and the Adam optimizer. This was found through various experimentation, with loss functions such as categorical cross-entropy.

The mean squared error loss function calculates the average squared difference between the predicted and true outputs of the model. Larger errors are more heavily penalized. The ultimate goal of the model is to minimize loss. The model does so through iterations of minimizing the loss function through backpropogation and gradient descent.

We made the decision to utilize the Adam Optimizer as opposed to the SGD optimizer because Adam has better momentum and thus tends to converge faster. It also allows for the usage of a lower learning rate (approximately in the range of 0.001 - 0.00001).

The model was trained for 10 epochs using a batch size of 32, and the validation loss and accuracy were recorded after each epoch.

3. Experiments

3.1. Filtered Backprojection Error

To quantify the error of the FBP algorithm, we computed the mean absolute error (MAE) between the outputted image and the original projected image.

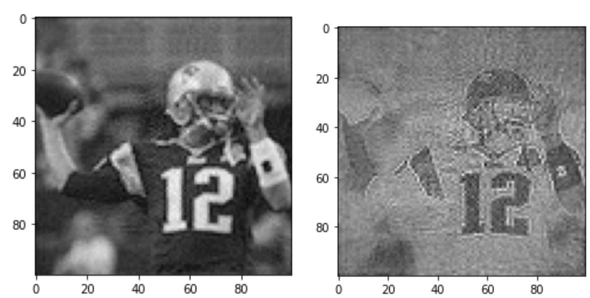

Figure 7: Reconstructed (left) and error image (right).

The right side of figure 7 shows the error image, computed as reconstructed minus input. We can see that the brightest regions are reconstructed almost perfectly (black error), while the darker regions are a little darker than the original (white error).

Overall, the MAE was 0.07 on a dynamic range of 0 to 1 (float64 image). This is more than enough for a good understanding and view into areas we cannot normally capture.

3.2. Algebraic Reconstruction Surface Parameterization

The ART algorithm is dependent on two main parameters – number of iterations and learning rate lambda – as described in the section 2.2. To understand where the algorithm performed most optimally for our inputs, we plotted the error against each of these parameters.

First, we quantified the error in largely the same manner as in section 3.1, by calculating the mean value of the update term (everything added to x in each iteration). We deliberately did not choose mean absolute error here, as we thought it may be possible that for high learning rate lambda, the algorithm actually *overshoots* the true reconstruction on the initial iteration, leading to negative errors. The results of this experiment are shown in figure 8 below.

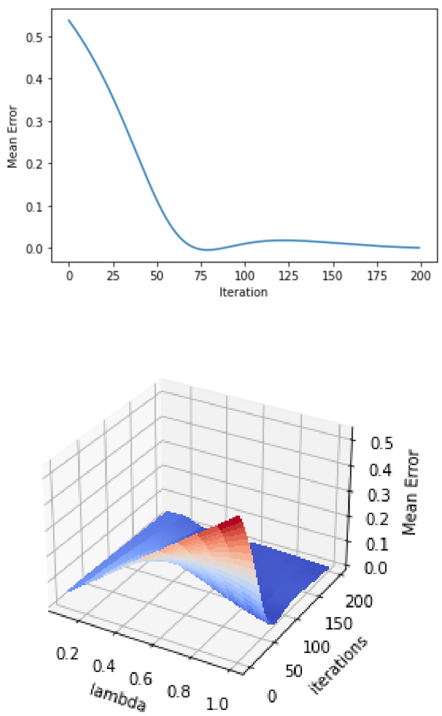

Figure 8: ART mean error vs iterations only (top); vs both iterations and learning rate lambda (bottom).

As expected, the error declines with increasing iterations, regardless of the learning rate. This is unsurprising for an error minimization algorithm.

3.3. Neural Network Validation

The network was trained on a set of dog images, as well as their corresponding sinograms. The dataset was split into training and validation sets with a ratio of 80:20 using the train test split method from the scikit-learn library. This data makes sense because the goal of our Convolutional Neural Network is to predict an original image based on the sinogram of the image, and we were able to train the model through feeding in a dog image from the dataset, calculate the loss of the original dog image with the model's predicted output, and then backpropogate to minimize that loss.

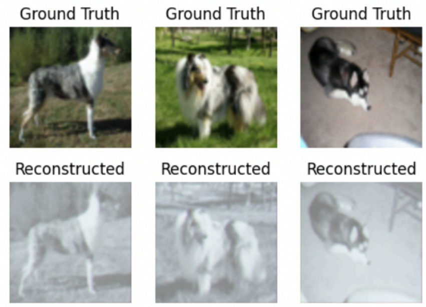

Figure 9: Results of CNN from validation set.

The above image is the result of our model's output on the validation dataset. Ground truth image are the original image from the dog dataset, while reconstructed image is what the CNN was able to create from the sinogram. This is a qualitative measure of success, as we are able to see how closely the reconstruction and original images are matched.

A potential downside of this qualitative measure is that the model is able to perform well on on validation data of dog images, but it may not be able to perform as well on other images. This is a potential next step in our project, as we look towards further testing and fine tuning of this model.

We utilize several factors to conclude our success in quantitative measures, We attempted to attain a training accuracy of 80% and a validation accuracy of atleast 75%. Furthermore, we strived to achieve a Mean Squared Loss Error of no more than 20%. Although there is no limit to how much we can vanish our MSE, 20% is a feasible task to achieve. Based on these targets, we achieved validation loss and accuracy as 35.829% and 99.998% respectively.

4. Implementation

For filtered backprojection, the code builds off of [2] to simulate tomographic projection and generate sinograms. The code borrowed from [2] also handles padding and image edge cases. The rest of the FBP and ART algorithms were implemented from the theory and converted by us to code, though the core math for ART does converge with some other implementations due to its widespread usage. The CNN portion also was built from bottom up, with the help of some course examples from EECS 442 and EECS 445.

5. Data

The data that we utilized in the Convolutional Neural Network was a dataset of twelve thousand images of dogs. This dataset was obtained through a previous project for U-M EECS 445 (Project 2, Fall 2022) by team member Abhinav Thakur.

References

[1] "Tomographic Reconstruction." Wikipedia, Wikimedia Foundation

[2] zmj."Filtered-Backprojection-Python."*GitHub*,

[3] Kesner, Adam. "Kesner Medical Physics - 3D Image Reconstruction Explained with Animated Gifs." *Kesner's Medical-Physics*,Dec.2014, available:sites.google.com/a/fulbrightmail.org/kesnersmedicalphysics/home/education/3d-image-reconstruction-explained-with-animated-gifs.

[4] Peters, Terry. (2002). CT Image Reconstruction.
Lyra, Maria, and Agapi Ploussi. "Filtering in SPECT Image Reconstruction." *International Journal of Biomedical Imaging*,23June2011,available:www.hindawi.com/journals/ijbi/2011/693795/.

[5] "Algebraic Reconstruction Technique." *Wikipedia*, 23 May 2022,available:wikipedia.org/wiki/Algebraic_reconstruction_technique.

[6] *Algebraic Reconstruction Technique (2022) Wikipedia.* available:https://en.wikipedia.org/wiki/Algebraic_reconstruction_technique

[7] Andersen, A.H, and A.C Kack. "Simultaneous Algebraic Reconstruction Technique (SART): A Superior Implementation of the ART Algorithm." *Ultrasonic Imaging*, vol. 6, no. 1, Jan. 1984, pp. 81–94,

[8] Würfl, Tobias, et al. "Deep Learning Computed Tomography: Learning Projection-Domain Weights from Image Domain in Limited Angle Problems." IEEE Transactions on Medical Imaging, 2018

[9] Cleveland Clinic, Emeritus Staff, Cleveland, OH, USA. "Image Reconstruction." *Radiology Key*, 9 Sept. 2016,