Applications of Neural Radiance Fields in Astronomy

Jenna Eubank
Center for Data Science
New York University
jke261@nyu.edu

Harlan Hutton
Center for Data Science
New York University
ahh303@nyu.edu

Harshitha Palegar
Center for Data Science
New York University
hp2233@nyu.edu

Zafir Momin
Center for Data Science
New York University
zm2114@nyu.edu

Abstract

Images of distant galaxies are notoriously noisy. Telescopes are sensitive to environmental conditions such as light pollution or atmospheric turbulence and are inherently limited by the faintness of galaxies which are billions of light years away. One way to easily improve image fidelity is to "stack" multiple images of the same object. However, typical astronomical algorithms for this task are limited in scope to the point they cannot even combine images from different telescopes or at different resolutions. What if we could use scene reconstruction techniques being developed in computer vision to combine observations from different telescopes in order to create a de-noised, aggregated output of an astronomical object? We present applications of neural radiance fields (3D) and planar alignment (2D) to optimally reconstruct images of astronomical phenomena from heterogeneous data, a task not currently possible in astronomy. Such a technique has the potential to overhaul the entirety of observational astronomy.

1 Introduction

The clear pictures of stars and planets we are used to seeing in textbooks and television shows are actually artificial depictions created through a combination of image stacking and artistic renderings. Telescopes use filters that allow them to capture brightness in a specific bandpass (range of the light spectrum), for example red, green or blue visible light. Scientists must then use software to combine these single color images into one colorized photo. However, this process is further complicated by the poor overall image quality caused by environmental factors such as noise, light pollution, obstructions, and distance.

Reconstructing a 3D scene out of a collection of 2D images is a longstanding objective in computer vision. Applications can range from gaming to facial recognition to Google Earth [3]. A recent successful development, called Neural Radiance Fields (NeRF) [10], uses deep learning methods to learn the volumetric density, color, and radiance of a scene from a collection of images. Its ability to identify and remove noise is particularly relevant to challenges facing photometry in astronomy. To validate these methods, one needs a ground-truth comparison, which is inherently inaccessible for stars and other astronomical phenomena. However, by using artificially created images of star clusters, we can establish a ground-truth that does not exist in astronomy but motivates the machine learning process. We identified two NeRF techniques, NeRF in the Wild (NeRF-W) [9] and Bundle-Adjusted Radiance Fields (BARF) [8], as potential applications to astronomical imaging as they combine multiple views of the same object despite having different lighting conditions, obstructions, noise, and varied positioning of the object.

In this paper we explore the use of neural radiance fields to improve the quality of images taken by ground-based telescopes and improve image fidelity for celestial phenomena. A potential outcome is a regularized strategy to combine telescope images and create more accurate recreations of astronomical objects with modern techniques.

2 Related Work

Combining images of objects in the sky from ground-based telescopes is a common and basic task performed by astronomers. The goal is to provide a clean, static representation of the sky through modeling and "coaddition" (summing in pixel-space) of multiple images. However, the success of this process is limited by poor image quality, leading astronomers to discard more than 90 percent of input data. The process requires all photos to be taken by the same telescope, at the same resolution and with the same environmental constraints during image capture. Further limitations include cost prohibitive computations and strict requirements for input images [2].

Current coaddition techniques are used to enhance images captured in sub-optimal conditions, reduce noise, and detect faint objects through layering multiple images of the same object. Several common methods of coaddition used in astronomy include direct linear combination, point spread function (PSF) matching, and non-linear weighting. Although these are some of the most regularly used methods, they are recognized as less than optimal [2].

Other better coaddition methods have been identified but have problems or restrictions that prevent them from being broadly applicable. For example, "likelihood coaddition" calculates the joint likelihood of models of the true sky. However, it has proven to be so computationally expensive that it can not be used. Calculating the joint likelihood for a 4k $\times$ 4k coaddition would require 200 gigabytes in a single precision [2]. Decorrelated coaddition faces the same issue of cost and requires specific noise conditions. Another method, Kaiser coaddition, which is less expensive, has overly strict demands for use [6]. Implementation requires that 1) noise in the images is uncorrelated and white, 2) the PSFs are spatially constant and 3) no pixels or boundaries are missing in the images.

Masking techniques were introduced through research with the Large Synoptic Survey Telescope (LSST) to remove transient artifacts from images during coaddition [1]. The algorithm initially used PSF-matched coaddition and identified differences between a PSF-matched warp and a static sky model to remove these artifacts. However, this algorithm is in need of improvement and is used in tandem with basic coaddition methods.

Use of NeRF-W and BARF is novel to astronomy and could lead to a cost-effective way to combine images. We aim to avoid many of the image input restrictions and remove artifacts with higher accuracy than current masking methods by employing these techniques. Neural radiance fields have been proven to be effective for scene reconstruction from images with varying cameras, resolutions, brightness, noise, and obstruction. They could produce a significant development over current methods to recreate significantly more accurate images of astronomical phenomena.

3 Problem Definition and Algorithm

3.1 Task

We aim to apply two extensions of NeRF to artificially generated images of star clusters: NeRF-W and BARF. Both algorithms have two overall objectives: 1) localize 3D coordinates and camera poses (registration) and 2) learn the neural representation of a scene (reconstruction). The registration process uses a volume rendering strategy to map (x, y) pixel coordinates to (x, y, z) coordinates with z representing depth from camera to object and reproduce camera poses. The reconstruction process uses multi-layer perceptrons (MLP) to parameterize color and geometric representations of the scene. **Table 1** contains the different strategies for registration and reconstruction for the three algorithms where γ is the positional encoding.

Table 1: *Summary of Registration and Reconstruction Processes*

Registration		Reconstruction
3D Mapping	kth Positional Encoding γ_k	Algorithm
COLMAP	$[\cos(2^k\pi x), \sin(2^k\pi x)]$	NeRF
COLMAP	$[\cos(2^k\pi x), \sin(2^k\pi x)]$	NeRF-W
Neural Network	$w_k(\alpha) \cdot [\cos(2^k\pi x), \sin(2^k\pi x)]$	BARF

3.2 Registration

3D Mapping Both NeRF-W and BARF require 3D coordinate and camera pose generation from the input images before model training can begin. NeRF-W uses the same volume rendering strategy as NeRF: structure-from-motion, a computer vision imaging technique to generate 3 dimensions from 2 dimensions. The principle is to track and then match features from one image to the next, filtering out features that are incorrectly matched. Instead of using structure-from-motion techniques to preprocess images before learning, BARF flexibly learns and registers 3D coordinates and camera poses during training.

Positional Encoding In the most basic terms, a positional encoding layer transforms a finite dimensional representation of the locations of items in a sequence, in our case location in a scene, into a format that can be used as input into the model. Each row of the positional encoding matrix is a vector that represents the position of the value associated with the row index. Each row-vector is an alternating series of sines and cosines, which allows for a continuous rather than discrete encoding of a vector's position, whose weights can be parameterized and optimized. In **Table 1**, all three positional encodings take in **x**, a vector that represents the (x, y, z) coordinates of images. NeRF-W uses the same strategy as NeRF for positional encoding: a full positional encoding. In **Table 1**, the k^{th} full positional encoding has no weight and its frequencies increase by the power of k, defined as the index of the row-vector. BARF improves upon this naïve encoding by adding a parameterized weight, w.

3.3 Reconstruction

NeRF First published in August 2020, NeRF is a method used to create 3D views of objects with sparse inputs. The research was the first to use neural scene representation to create very accurate novel views of objects from images. Their results outperformed previous strategies which included models that fit neural 3D representations to scenes and deep convolutional neural networks that predicted volumetric representations. The NeRF models learn volumes, which contain object information such as density and color, and render the object by "shooting" a ray from a camera that marches around the object from each pixel towards the learned volume. Known as "ray tracing", this process replicates how light behaves in natural environments, but in the inverse direction from the camera. This enables rendering software to create more accurate and realistic views by capturing the colors and textures of objects [12].

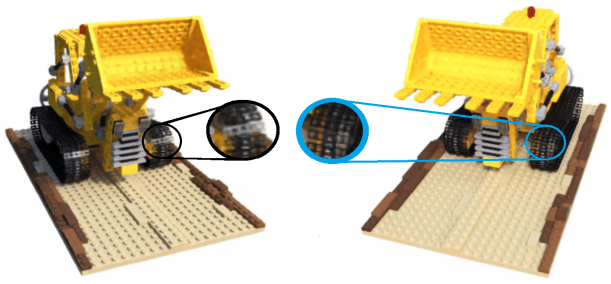

Figure 1: *The output after 10 hours of runtime on the Lego dataset. The same section of the model toy has a different color depending on the angle it is viewed from. As light is traced through the image, reflections will also change creating life-like 3D recreation.*

A "scene" or input to the model consists of the positional encoding of 5 parameters detailed by a 3D location, (x, y, z), and a 2D viewing angle given by (θ, φ). NeRF outputs a view-dependent color in RGB (red-green-blue) and a scalar volume density, . The key difference between NeRF and its predecessors in 3D view reconstruction is that NeRF uses a continuous volume encoding reparameterized with a neural network, as opposed to a discrete sampling (e.g., voxels) of the 3D space. In **Table 1**, the encoding of 3D coordinates is represented by alternating continuous sin and cosines. This encoding acts as input to a fully-connected neural network. The parameters are optimized through gradient descent, which works to minimize the error between true, observed images and the views rendered during 3D point generation. As shown in **Figure 2**, the neural radiance field is a continuous "sphere" that can accurately capture light intensity and color.

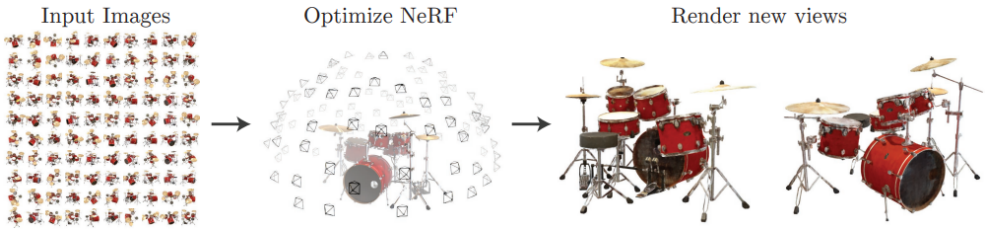

Figure 2: *A toy visualization of a neural radiance field around a drum set from a set of input images. Each square is an input image to the model and its location and angles are estimated. The rendered views are the output of the reconstruction and include novel views of the object. Figure originally published in NeRF [10].*

NeRF-W NeRF-W uses NeRF's 3D mapping and full positional encoding methods as well as its first MLP. It contains two enhancements to NeRF's reconstruction process: latent appearance modeling and infrastructure to handle transient objects in the layers that output color. **Figure 3** visualizes the differences between NeRF-W and its predecessor.

Latent appearance modeling allows NeRF-W to account for differences in lighting and image processing; an ability that the original NeRF lacks. Each image is assigned an embedding vector containing appearance information. Radiance contained in the vector becomes image dependent, causing pixel color to become image dependent as well. These image appearance vectors act as inputs to the second MLP network, which only produces color. Because the volume densities have already been outputted by the first MLP network, this process allows the model to vary the radiance between images without affecting the geometry of the scene rendering.

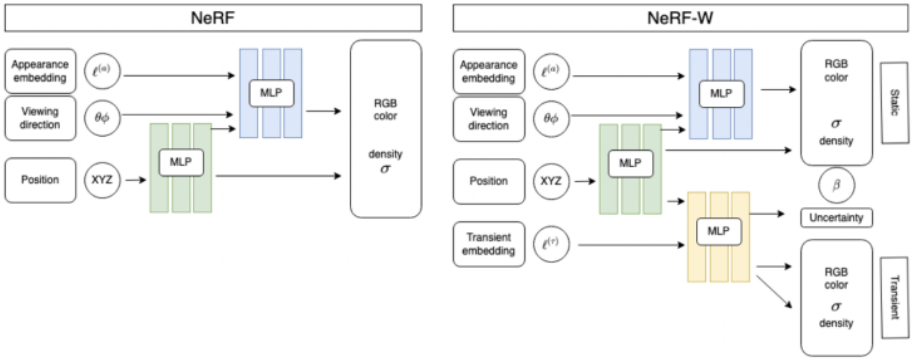

Figure 3: *Differences between NeRF and NeRF-W algorithms.*

The second enhancement NeRF-W introduces is the ability to learn both static and transient objects. For example, if the input is a collection of images of a star with, say, satellites passing in front of it, NeRF-W will learn the star itself as a static object and the satellites in the sky as transient. The

second MLP network, the one that only outputs color, is in charge of static elements. While NeRF only has two MLP networks, NeRF-W adds a third to handle transient elements with the outputs color and density. This separates the densities of transient objects without affecting the static volume density from the first MLP network. The final rendering uses the information of only static objects.

BARF First published in August 2021, BARF is a method to train NeRF from imperfect or even unknown camera poses. BARF's extension comes in the registration process during the positional encoding of the 3D coordinates of images. NeRF and NeRF-W map 3D coordinates to higher frequency dimensions using a full positional encoding, which means that every pose changes direction with the same frequency. This produces meaningless gradients for optimizing the camera poses that often cancel each other out. To improve on this, BARF changes the weights of the row-vectors to a function parametrized by $\alpha \in [0, L]$, where L represents the number of frequency bases. The frequency base controls the wave frequency of the sines and cosines in the positional embedding, so as L increases, so does the number of waves. This is the coarse-to-fine registration process that allows for meaningful gradients of the camera poses and therefore better optimization, and obtains clearer images from messier camera poses.

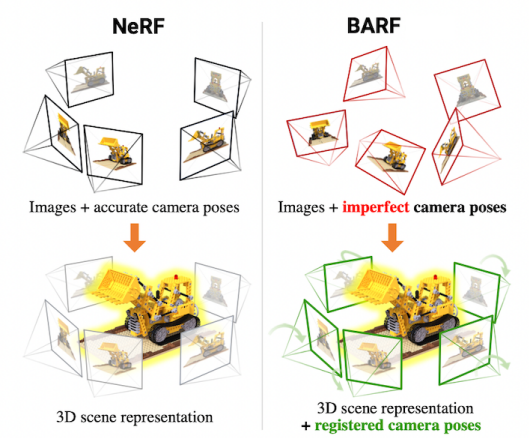

Figure 4: *Shows the differences in requirements for the input images for both NeRF and BARF. BARF calculates the camera poses through optimization thus, relaxes the need for accurate image poses. Figure originally published in BARF [8].*

Planar Alignment Planar alignment is a 2D scene reconstruction method that finds a geometric transformation (learned through a parameter that controls warp) that minimizes the photometric error between two images. It uses gradient-based optimization to learn the warp parameter, which is updated using a view synthesis-based objective function. The objective function refers to image correspondence, or the process of identifying and matching features that are present from image to image. Similar to BARF, planar alignment relies on coarse-to-fine registration for meaningful gradients and needs images with smooth signals as inputs. Because real-world images contain complex and high-frequency signals, the images are blurred at the earliest stages of registration which allows for smooth alignment. The authors of BARF introduce a method for cropping a single image into five patches which act as input to planar alignment. The purpose of using patches of a single image rather than multiple images is to provide a more comparable baseline to the 3D methods used in BARF. While neural radiance fields use multiple 2D images of a single object to reconstruct a single 3D scene, planar alignment uses multiple patches from a single input to reconstruct a single 2D scene.

4 Experimental Evaluation

4.1 Data

Existing Image Sets The NeRF authors published image collections of both simple real scenes and synthetic renderings of objects [10]. The synthetic rendering datasets contain 100 images for

training and 200 for testing with input views rendered at 800 $\times$ 800 pixels. The simple scene datasets were taken by the authors on a cellphone, vary in the number of images, and are 1008 $\times$ 768 pixels for each view. The NeRF-W authors used previously created datasets of unconstrained images of monuments scraped from the internet (similar to those found in a Google Image search) [5]. There are a different number of crowdsourced images for each landmark and the images vary in size. Of the provided datasets, we used Lego (synthetic rendering), fern (simple real scene), and Taj Mahal (unconstrained).

Figure 5: *Sample images (from left to right) synthetic rendering (noisy star), simple real scene (candle) and unconstrained complex scene (Brandenburg gate).*

Introducing New Images We created our own datasets of simple real scenes (candle) and synthetic renderings (Mickey Mouse clip art and ArtPop star clusters). The candle dataset contains 28 images of a candle taken from a front-facing cell phone camera with image size of 1024 $\times$ 768 pixels. The Mickey Mouse clip art dataset includes 28 manipulated versions (cropped, expanded, shrunk etc.) of a single original image. The python package ArtPop allowed us to create artificially rendered star clusters by manipulating object characteristics and capture quality. Image parameters are described in **Table 2**. In effect, we were able to generate multiple images of the same object from the most ideal capture conditions to extremely noisy conditions. Outputs are either black and white, replicating a single bandpass filter, or colorized, combining red, green, and blue telescope filters. Image sizes fall within the existing input requirements for the NeRF and BARF models at 508 $\times$ 507 pixels. We used two star cluster datasets, set one with 127 images and set two with 84.

Table 2: *ArtPop Rendering Parameters*

Stellar System Parameters	Image Quality Parameters
Age	Bandpass
Distance	Color composite/single bandpass
Effective Radius	Exposure Time
Ellipticity	Full Width Half Maximum
Metallicity	Noise
Number of Stars	Point Spread Function
Position Angle	Sky Brightness
Sersic Index	

Preprocessing A set of images used for successful scene reconstruction should have texture i.e. discernable differences in color intensity and spatial arrangement, similar illumination conditions, a maximum disparity between views of at most 64 pixels, and different viewpoints [11]. NeRF-W uses COLMAP, a structure-to-motion pipeline for reconstruction of image collections, to generate coordinates and poses. The method we chose to match features from image to image is an exhaustive matching strategy, where each image is compared to every other image. The poses are generated using 6 degrees of freedom (6-DoF) camera pose estimation, where 6-DoF refers to a 3D object's freedom of position and orientation [11]. Because BARF learns the 3D coordinates and camera poses using a neural network, no image preprocessing was required. As planar alignment is a 2D image to 2D scene application, it also did not require any image preprocessing.

4.2 Methodology

We first recreated outputs from the original NeRF-W and BARF models to get a better understanding of training and verify that our implementation was correct. Next, we introduced our own images using the candle dataset and compared results. Finally, we trained the planar alignment model on the Mickey Mouse and ArtPop datasets.

Parameters We trained NeRF-W with baseline parameters used by the authors: 64 additional fine samples, 0 noise standard deviation, Adam optimizer [7], 5e-4 learning rate, batch size of 1024, "steplr" scheduler type, 2, 4, and 8 decay steps, and 0.5 learning rate decay. We trained BARF with parameters used by the authors: 1e-3 learning rate, Adam optimizer, batch size of 16, and a 0.1 learning rate decay. We trained planar alignment with parameters provided by the authors: homography warp function, 1e-3 learning rate, and a batch size of 5.

Evaluation Metric NeRF-W, BARF, and planar alignment use peak signal-to-noise ratio (PSNR) to evaluate performance [10] [9] [8]. PSNR is the ratio between the maximum possible value of a signal (MAX_I^2) and the power of the corrupting noise that affects the quality of its representation (MSE). MAX_I^2 is calculated using the bits (B) of an image. MSE is calculated using a $m \times n$ image I and its approximation K. The higher the PSNR, the better the scene has been reconstructed to match the original image, as we want to minimize the MSE with respect to the MAX_I^2 of an image.

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{1}$$

$$\text{MAX}_I^2 = 2^B - 1 \tag{2}$$

$$\text{PSNR} = 10 * \log_{10} \frac{\text{MAX}_I^2}{\text{MSE}} \tag{3}$$

4.3 Results

We successfully recreated NeRF-W and BARF results using the Lego, fern and Taj Mahal datasets as well as our own candle dataset. However, when we shifted to the artificial images of stars, NeRF-W's 3D coordinate registration strategy, structure-from-motion, was unable to identify the depth dimension in order to register camera poses. BARF was also unable to learn the third coordinate during the registration process using its neural network. Both real and synthetic astronomical images lack discernible depth due to the extreme distance from the objects. This also causes an issue with camera poses as there is not enough variation in angle and thus the requirement of different viewpoints is also violated. Finally, objects may appear in different locations in the frame of the photo, which violates the maximum disparity condition.

After the algorithms failed to complete the registration process on both artificial star datasets, we shifted our efforts to 2D scene reconstruction using planar alignment. We utilized images from the same datasets as NeRF-W and BARF: Lego, fern, Taj Mahal, candle, and two artificial star datasets (high- and low-noise) as well as the cat dataset BARF used for planar alignment. We visualize select results in **Figure 5** below and report all PSNR results in **Table 3**.

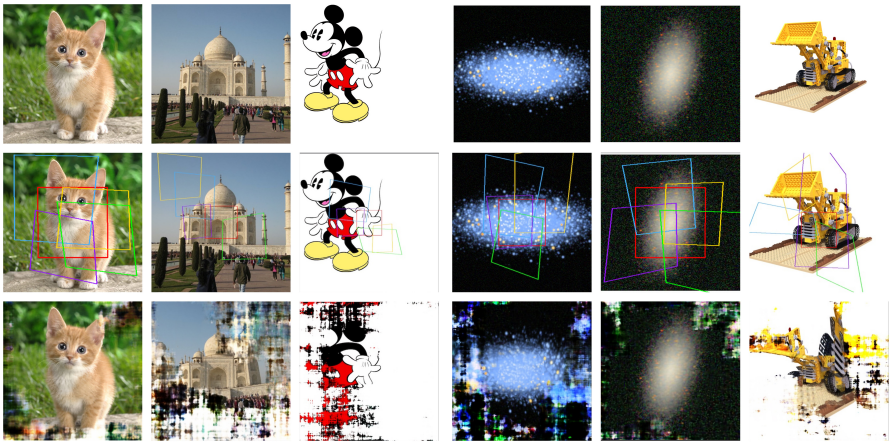

Figure 6: *From top to bottom: input image, patching, and planar alignment output for cat, Taj Mahal, Mickey Mouse, low-noise star cluster, high-noise star cluster and Lego images.*

Table 3: *PSNR of recreated images*

Image	Image category	PSNR
Cat	Simple Real Scene	35.72
Candle	Simple Real	34.72
Taj Mahal	Unconstrained Complex Scene	32.78
Mickey clip-art	Synthetic Rendering	26.56
Low noise star cluster	Synthetic Rendering	24.24
High noise star cluster	Synthetic Rendering	22.88
Lego	Synthetic Rendering	19.77

4.4 Discussion

Neither 3D registration technique, structure-from-motion (NeRF and NeRF-W) or coordinate-based neural network (BARF), was able to localize points from artificial star images. These constraints will also apply to real scenes of star clusters and limit any applications of 3D NeRF-W and BARF. While we were able to successfully reconstruct a scene from cropped patches using planar alignment, it was originally created as a proof of concept of BARF and was not the main focus of the publication.

Planar alignment was more effective at recovering real life scenes than synthetic renderings. This may indicate that hyperparameters should be tuned differently for the categories of real life scenes and artificially generated renderings in addition to adjustments for individual image content. It may also indicate that planar alignment is not invariant to rotationally symmetric inputs like galaxies.

Although the PSNR was lower for the star clusters, there are still promising computer vision applications for astronomical images like 2D neural radiance fields. At present, we believe there is room for improvement even with simple hyperparameter tuning for planar alignment around patch size, iterations, and batch size. Since we saw a meaningful difference in performance between the real life scenes and artificial renderings we may be able to tune for image type to avoid overfitting on individual image content. The current implementation has a short runtime and could afford more extensive model training.

5 Conclusions

One of the key challenges of both traditional astronomy coaddition methods and cutting edge NeRF is the lack of flexibility for image inputs. The registration of astronomical data proved to be the main hurdle in 3D scene reconstruction.

BARF's ability to reconstruct scenes from messy or unknown camera poses provides the most promising avenue for further research. We hope to continue development of a 3D coordinate and camera pose generation process for astronomical images in order to successfully apply neural radiance fields. In the meantime, planar alignment is a strong starting point in comparison to coaddition methods as well as future NeRF methods. Since planar alignment did not perform as well on the artificial images as it did on real scene images, we hope to explore improving the PSNR as well as identifying any qualities of the image that contribute to weaker outputs. Our main goal for planar alignment is to eventually incorporate an input of multiple images for scene reconstruction so that we can fine-tune its ability to remove noise and recover object features like radiance, which can vary from image to image.

6 Lessons Learned

We progressed through our research by finding and experimenting with more adaptable methods from NeRF-W to BARF to planar alignment. We were able to make the switch from one method to another by having a very focused objective and a clear process. When we faced problems, we were able to decide to move to another algorithm instead of getting stuck on a technology that would prove less than ideal.

This project also underscored the importance of having domain knowledge and applying that knowledge to data science problems. Looking back, we wish we had taken more time to learn more about the astrophysics domain on the front-end, as it may have helped us identify potential hurdles in the 3D point registration process. If we were to approach this problem again we would build a stronger foundational understanding in astronomical data before implementing code.

7 Bibliography

[1] Yusra AlSayyad. *Coaddition Artifact Rejection and CompareWarp*. 2019.

[2] Jim Bosch. *Flavors of Coadds*. 2016.

[3] Google Earth. `earth.google.com/web/`.

[4] Johnny P. Greco and Shany Danieli. *ArtPop: A Stellar Population and Image Simulation Python Package*. 2021.

[5] Yuhe Jin, Dmytro Mishkinn, Anastasiia Mishchukn, Jiri Matasn, Pascal Fuan, Kwang Moo Yi, and Eduard Trulls. *Image matching across wide baselines: From paper to practice*. 2020.

[6] Nick Kaiser. *Addition of Images with Varying Seeing*. 2001.

[7] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017.

[8] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. *BARF: Bundle-Adjusting Neural Radiance Fields*. 2021.

[9] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. *NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections*. 2021.

[10] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*. 2020.

[11] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. 2016.

[12] Turner Whitted. *An Improved Illumination Model for Shaded Display*. 1980.

8 Acknowledgements

We would like to thank Dr. Shirley Ho, Miles Cranmer PhD-C, and Dr. Peter Melchior from the Center for Computational Astrophysics at Flatiron Institute and Dr. Julia Kempe from the Center for Data Science at NYU for their guidance and help with our project.

9 Student Contributions

Jenna Eubank: ArtPop implementation, planar alignment implementation, poster, paper
Harlan Hutton: NeRF-W implementation, BARF implementation, COLMAP implementation, candle dataset creation, poster, paper
Harshitha Palegar: BARF implementation, poster, paper
Zafir Momin: COLMAP implementation, planar alignment implementation, poster, paper