

Computational Star-Tracking Compensation in Astrophotography

David Whisler
Stanford University
Dept. of Electrical Engineering
dwhisler@stanford.edu

Adam Dai
Stanford University
Dept. of Electrical Engineering
addai@stanford.edu

Abstract

Low-light and the rotation of the earth pose several challenges to astrophotographers. Modern solutions to these challenges typically involve expensive hardware. We investigate computational methods to assist in astrophotography, specifically in the deblurring of “star trail” images captured using long-exposure times during which the stars in frame rotate about the sky. This task is divided into first identifying the center of rotation in the image, then using deconvolution to remove the blur. We implement and evaluate various methods for both center of rotation detection and deconvolution, and we demonstrate the full pipeline on real star trail data.

1. Introduction

Astrophotography is a field that has been typically confined to a small subset of hobbyists and professionals with specialized equipment. Part of this is due to the low light conditions - long exposure times are needed to capture enough light to produce a reasonable image, but this often causes motion blur artifacts called “star trails” due to the rotation of the earth. Star trails cause a circular motion blur, which in the northern hemisphere is centered on the celestial north pole, roughly the location of the star Polaris (the North Star). This motion blurring is exacerbated by increased magnification levels needed to capture deep sky objects, such as planets and galaxies. Star trails are sometimes desired for artistic effect, but generally the goal is to mitigate them as much as possible. The preferred method of overcoming this limitation is through the use of mechanical tracking mounts for cameras and telescopes that physically track the location of the target object across the sky. These mounts can be fairly expensive, and require precise machining to ensure a smooth tracking motion to eliminate wobble.



Figure 1. Example of star trails in a long exposure

1.1. Deconvolution

The expense of this hardware makes computational post-processing solutions to remove this blurring more attractive, especially for hobbyists and laypeople more concerned with qualitative appearance than astronomical accuracy. One particular method to remove motion blur is through deconvolution. The motion blurred image can be represented as a convolution of the latent image with a point spread function (PSF), or blur kernel.

$$b = x * c + \eta \quad (1)$$

In this formulation, x is the latent image, c is the point spread function or kernel, and η is a noise term, which could follow a Gaussian distribution, a Poisson distribution, or a mixture of both. In non-blind deconvolution where the PSF is known, the inverse operation is used to recover the latent image subject to the amount of noise present. This method can be used to computationally recover an approximation of the latent astrophotography image after it has been subject to the motion blurring of the earth’s rotation.

2. Related Work

Prior work in deblurring long-exposure images of stars have made use of polar transform mapping[1]. This mapping gives the blur streaks a more uniform appearance, where all streaks are straight and strictly vertical. For our purposes, it also allows us to use a spatially-invariant PSF to

perform deconvolution. Deconvolution itself has also been used in deblurring star scenes, specifically Richardson-Lucy deconvolution [2]. Su et al. uses a thinning algorithm to obtain the star point trajectory and thus extract the appropriate blur kernel for deconvolution of the motion-blurred star image. In their work, this approach works well for very sparse images and is able to handle wobbliness in the motion. For our methods, we assume a linear blur (in the polar domain) for simplicity and since we want to handle both sparse images and denser ones with more background (e.g. an image of the milky way).

Traditionally, the position of the circumpolar star in the image may be determined from geographic coordinates or a technique known as plate solving in which the stars in the image are matching to known constellation maps. We did not find much on computational techniques which have been used to solve this problem. We took on the general strategy of attempting to match a circle of some radius to the image which has center at the center of rotation. Such methods of matching circles to data include least-squares fitting [5] and the Hough Transform [7].

3. Methods and Approach

3.1. Polar Remapping

With a rotational blur, the blurring convolution cannot be expressed as a linear operation, since the blur kernel is not spatially invariant. In order to apply deconvolution effectively, the image was first remapped in a polar coordinate space similar to the method in [1], which has the effect of making the blurring spatially invariant, since in the polar frame, the blur length was only a function of the angle θ which the scene had rotated around the celestial north pole. However, this transformation is dependent on knowledge of the coordinates of the rotation center, which was a problem that was addressed with several different methods. Assuming the center of rotation was known, then the image was padded such that the center of rotation was at the center of the image. Then, values for the new coordinates were chosen.

$$r = \{0, \dots, \max(H, W)/\sqrt{2}\} \quad (2)$$

$$\theta = \{0, \dots, 2\pi\} \quad (3)$$

The number of samples chosen for each was such that each pixel in the original image would have at least one corresponding sample ($\max(H, W)/\sqrt{2}$ radius samples, $2H + 2W$ angle samples). Then, these coordinates were mapped into the cartesian space of the new image.

$$x' = r \cos \theta \quad (4)$$

$$y' = r \sin \theta \quad (5)$$

Finally, the original image was interpolated at these sample points, generating the polar mapped image. This had the

effect of making the blurring throughout the image linear and spatially invariant.

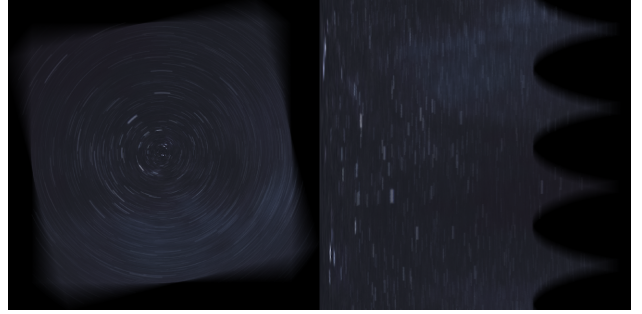


Figure 2. Polar mapping transformation

3.2. Center of Rotation Detection

Our approach for deblurring long exposure star images required knowing the position of the circumpolar star around which all the other stars in the sky rotate about. In the image, this is characterized as the center point around which the star trail arcs appear to rotate about. We experimented with and implemented various computational methods to attempt to find this center of rotation, testing on both simulated images and real images of star trails found online.

3.2.1 Least-square Circle Fit

This strategy first binarizes the input image and extracts x-y coordinates for all the hot pixels in the binary image. Then, a circle is fit to the data using a least squares objective. The intuition behind this is that the star trails form circular patterns in the image so the best fit circle may match with these circular patterns. If $u_1, \dots, u_m \in \mathbf{R}^2$ are the points extracted from the image, then we want to solve the following optimization problem:

$$\text{minimize } \sum_{i=1}^n (\|u_i - x_c\|^2 - R^2)^2$$

over optimization variables x_c , the center of the circle, and R , the radius. If we let $t = r^2 - \|x_c\|_2^2$, then we can reformulate the problem into traditional least-squares form

$$\text{minimize } \|Ax - b\|_2^2$$

where

$$A = \begin{bmatrix} 2u_1^T & 1 \\ 2u_2^T & 1 \\ \vdots & \vdots \\ 2u_m^T & 1 \end{bmatrix}, x = \begin{bmatrix} x_c \\ t \end{bmatrix}, b = \begin{bmatrix} \|u_1\|_2^2 \\ \|u_2\|_2^2 \\ \vdots \\ \|u_m\|_2^2 \end{bmatrix}$$

and we can efficiently solve the system using the optimality conditions $A^T(Ax - b) = 0$ [4]. Variations of this method were also explored in which the objective was modified to fit multiple concentric circles [6] or add regularization terms.

3.2.2 Hough Transform

The Hough transform is a feature extraction technique in which the input is mapped to a parameterized space, and a voting procedure is carried to find instances of objects with certain shape. By choosing our parameter space to match that of a circle, we can leverage this technique to find circular shapes in our image.

In practice, we implemented the Hough Transform for circle detection using the MATLAB function `imfindcircles`, which takes in an image and returns a list of circle centers, radii, and confidence levels. However, we found that this function was very slow for larger images (e.g. 4000×6000 pixels) which are typically found in astrophotography. In order to speed up the process in order for it to be feasible to run on large images, We implemented a small pipeline which first binarizes the input image and determines a scale factor based on its size. If the image is equal to or smaller than the acceptable size, the scale factor is set to 1, otherwise a scale factor is chosen that scales down the image to an appropriate size. Before the image is downsized however, it is dilated by an amount corresponding to the scale factor, essentially low-pass filtering it in order to prevent anti-aliasing. Then, the image is downscaled and the Hough transform is run on the smaller image, and the resulting center is taken and scaled back up to give the final output.

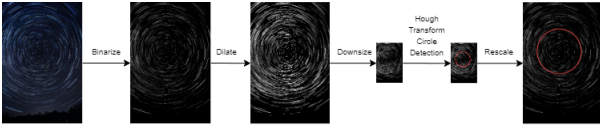


Figure 3. Hough transform-based circle detection pipeline

3.2.3 Convolutional Neural Network

Another method used to detect the center of rotation for use in the polar remapping was a convolutional neural network (CNN). Simulated star trail images were generated as grayscale 250×250 pixel images, with a randomly chosen degree of sparseness. Then, a random rotation center within the image was chosen, and a rotational blur to a randomly chosen angle θ was generated by adding incrementally rotated images to the original. The label for each image was a rotation center tuple in the form (x, y) . A neural network was then trained on 1000 simulated images for 50 epochs



Figure 4. Sample of generated training data

with a batch size of 5, using the Adam optimizer, and a mean squared error loss as defined below regressing directly on the rotation center.

$$\mathcal{L} = \frac{1}{2m} \sum_{i=1}^m (x^{(i)} - \hat{x}^{(i)})^2 + (y^{(i)} - \hat{y}^{(i)})^2 \quad (6)$$

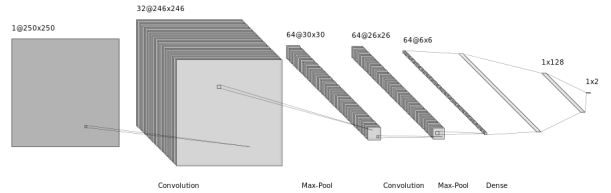


Figure 5. Neural network architecture

3.3. Deconvolution

Once the center of rotation was identified and the image was remapped into a polar space, deconvolution could be applied. To estimate the PSF for the deconvolution, all that is needed is the image exposure time, since the earth rotates about the celestial axis in approximately 24 hours.

$$n_{\text{blur}} = n_{\theta} \left(\frac{t}{24\text{hrs}} \right) \quad (7)$$

$$n_{\theta} = 2H + 2W \quad (8)$$

$$c = \begin{bmatrix} 0 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 0 & \dots & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{2n_{\theta}-1 \times 2n_{\theta}-1} \quad (9)$$

3.3.1 ADMM

One method of deconvolution used was the Alternating Direction Method of Multipliers [8]. In our formulation, this

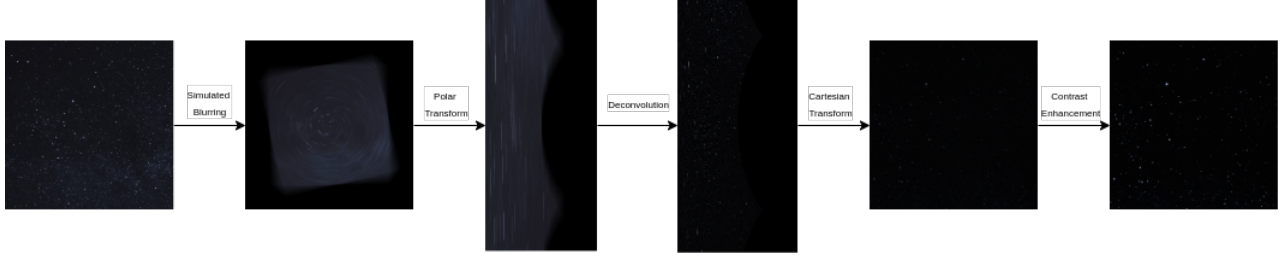


Figure 6. Image processing pipeline

optimization takes the form

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} \|Ax - b\|_2^2 + \lambda \Gamma(z) \\ & \text{subject to} \quad Kx - z = 0 \end{aligned} \quad (10)$$

Here, A is the blurring operation defined by the PSF, x is the latent image, b is the blurred image, and Γ is a prior asserted on the image, defined by K . To solve this, an augmented Lagrangian is formed. The updates to each variable (x , z , u) occur iteratively, with the following update rules after being initialized to 0:

$$x \leftarrow \text{prox}_{\|\cdot\|_2, \rho}(v) = \underset{x}{\text{argmin}} \frac{1}{2} \|Ax - b\|_2^2 + \frac{\rho}{2} \|Kx - v\|_2^2, \quad v = z - u \quad (11)$$

$$z \leftarrow \text{prox}_{\Gamma, \rho}(z) = \underset{z}{\text{argmin}} \lambda \Gamma(z) + \frac{\rho}{2} \|v - z\|_2^2, \quad v = Kx + u \quad (12)$$

$$u \leftarrow u + Kx - z \quad (13)$$

To implement these updates efficiently, the convolution operation represented by A in the x update is performed in the Fourier domain, and the z update is performed using the element-wise soft thresholding operator. As for priors, two were used - one was the sparsity prior, which promotes sparseness in pixel values (similar to a star scene against a black background), and another was the anisotropic total variation prior, which promotes sparse gradients. These took the form $K_{\text{sparse}} = I$ and $K_{\text{TV}} = D$, where D is the finite difference operator.

3.3.2 Richardson-Lucy

Another deconvolution method used was Richardson-Lucy, which is a maximum log-likelihood solution using the Poisson distribution [2], which is especially applicable in low-light conditions where photon shot noise dominates. This optimization takes the form

$$\underset{x}{\text{maximize}} \log(p(b|x)) = \log(Ax)^T b - (Ax)^T \mathbf{1} - \sum_{i=1}^M \log(b_i!) \quad (14)$$

This can be calculated with an iterative update rule of the following form:

$$x \leftarrow \left(\frac{A^T \left(\frac{b}{Ax} \right)}{A^T \mathbf{1}} \right) x \quad (15)$$

Richardson-Lucy deconvolution can also be computed with total-variation (TV) prior, where the iterative update takes the following form:

$$x \leftarrow \left(\frac{A^T \left(\frac{b}{Ax} \right)}{A^T \mathbf{1} - \lambda \left(\frac{D_x x}{|D_x x|} + \frac{D_x y}{|D_x y|} \right)} \right) x \quad (16)$$

4. Results & Analysis

4.1. Center of Rotation Detection

In order to compare the performance of our center detection approaches, we ran them on 3 sets of 3 images each. The first set consisted of fully simulated images generated by randomly distributing "stars" (hot pixels) against a black background (i.e. a sparse random binarized image) then rotating it about a given center point. The second set was created by taking a real image of unblurred stars and simulating rotational blur (in the same way that simulated images were created to test the deconvolution pipeline). The third consisted of real star trail images found online. For all the simulated test images the ground truth center of rotation was known and for the real images a ground truth center was chosen manually by visual inspection. We then computed the estimated centers for each set, averaged the mean-squared errors, and compared across the different sets and methods. We found that all the methods had different advantages and disadvantages, although the Hough transform seemed to perform the best in terms of accuracy and reliability across the board.

4.1.1 Least-square Circle Fit

The least-square circle fit had the advantage of being very fast and efficient to compute, as the computation boils down to solving a linear system. However, the solution is biased towards the image center of mass (i.e. the centroid of all the pixels in the image) since the circle is being fit not to a particular arc or set of arcs that form a circle but to the entire image. For most star images, the center of mass lies near the actual pixel center of the image since stars or star streaks are more or less evenly distributed through the image, and this

Method	Simulated Stars	Simulated Blurs	Real Star Trails	Overall	Notes
CNN	0.677e-3	53.5e-3	82.6e-3	45.5e-3	Did not generalize to real data very well
Hough Transform	0.101e-3	0.0121e-3	0.620e-3	0.198e-3	Did not perform well for sparse images
Least Squares	43.0e-3	105e-3	8.47e-3	52.2e-3	Biased toward the image center of mass

Figure 7. Comparison of mean-squared error for center detection methods.

the least-squares fit would tend to produce rotation centers biased towards the center of the image. The method does perform well for images whose rotation centers are close to the center, but does not generalize well.

4.1.2 Hough transform

The Hough transform method was slower than least-squares but was much more accurate in detecting the center of rotation, especially when the center was located near the corner or edge of the image. Across all sets of images the Hough transform consistently was able to find the center of rotation with lower MSE than the other 2 approaches. However, one drawback of the Hough transform was that for exceptionally sparse blurs in which the blur amount was not long enough to result in long arcing streaks, the Hough transform was not able to identify any circular shapes or would place circles around the individual streaks rather than the streak center.

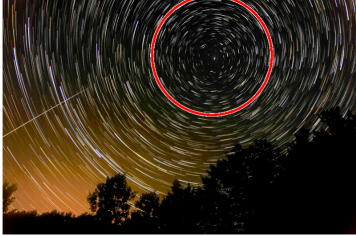


Figure 8. Detected center of rotation using Hough transform in sample star trail image. This method is able to handle centers of rotation away from the center of the image as well as foreground in the image.

4.1.3 Convolutional Neural Network

After training, the network achieved a test loss (MSE) of $0.474e-3$, where the pixel prediction error was normalized to the image dimensions. However, when tested on images that did not come from the simulated test distribution, the predictions did not generalize very well. One reason this may have happened is that the network may have learned where the brightest pixels in the image were, as this is where

the image center in the simulated data was. In the real star trail images, the trails tended to be a more constant brightness due to the common practice of generating them as a series of stacked exposures, instead of one long exposure.

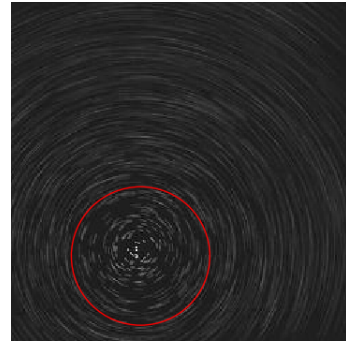


Figure 9. Sample of neural network prediction

4.2. Deconvolution Results

As shown in Figure 9, the results of the deconvolution for various priors with ADMM and Richardson-Lucy were compared qualitatively, as well as quantitatively with PSNR and SSIM. From a purely qualitative standpoint, ADMM with the sparse prior performed the best at generating a reasonable reconstruction with a minimum amount of artifacts. However, ADMM with the TV prior tended to perform better on the quantitative metrics, preserving more detail even though radial distortions are more visible. Richardson-Lucy methods had the most severe rotational blur artifacts, though they performed best from a PSNR metric standpoint.

4.3. Real Star Trail Deconvolution Results

We were able to run our full pipeline on a real star trail image and produce visually reasonable and appealing results. The full pipeline consists of first detecting the center of rotation, for which we used the Hough transform, then using that center to map that image to polar coordinates, deconvolving with a linear blur PSF with estimated length, for which we chose to use ADMM with the TV prior for best qualitative results, then mapping the deconvolved re-








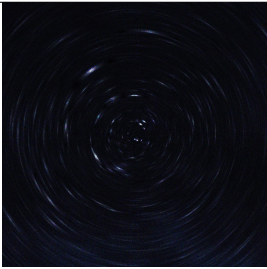

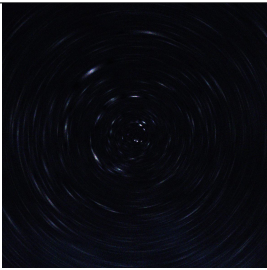
	$\theta = 5^\circ$			$\theta = 15^\circ$		
Deconvolution Method	PSNR	SSIM	Result	PSNR	SSIM	Result
Ground truth	-	-		-	-	
ADMM with sparsity prior	17.9594	0.44894		20.1671	0.19097	
ADMM with TV prior	16.4092	0.48085		20.8521	0.53328	
Richardson-Lucy	20.8218	0.32276		21.5177	0.31575	
Richardson-Lucy with TV prior	20.8293	0.3341		21.3758	0.32558	

Figure 10. Comparison of quantitative and qualitative performance of different deconvolution strategies on simulated data of circular blur amounts of $\theta = 5^\circ$ and $\theta = 15^\circ$. Hyperparameters for each method and brightness and contrast enhancement parameters were tuned for good qualitative results.

sult back to cartesian coordinates, and applying brightness and contrast enhancement.



Figure 11. Result of pipeline on example real star trail image.

5. Discussion & Conclusion

Under the right conditions, deconvolution coupled with a polar remapping performed quite well for reconstructing a qualitatively “good” looking image of an astrophotography image. All methods were prone to artifacts, but this was dependent on the inherent sparsity in the image, as well as the presence of other factors, such as light pollution and atmospheric interference other than simply stars as points of light. In general, an ADMM-based approach with a sparseness prior performed the best. However, these methods are unlikely to exceed the quality of a hardware tracking mount, and would likely be unacceptable to professional astronomers and serious hobbyists, because although the reconstruction looks “good”, then have the side effect of potentially adding false stars/features as artifacts in the image. If this was used for scientific applications, there may be doubt as to the veracity of portions of the image. On the other hand, for the consumer or photographer simply looking to produce a well-shot nighttime image, computational tools such as deconvolution can open up an entirely new area that previously had a steep learning curve. In addition to deconvolution, other methods have been developed recently to bring nighttime photography to the average consumer, such as Google’s Nightsight, which computationally stacks many shorter exposures on top of each other [9].

Our pipeline has some room for improvement, such as in the automatic detection of the rotation center. The best performing method was the Hough transform, although this has limitations for especially sparse images where the star trails do not form a complete circle, or at small blurring angles. The CNN-based model has the most potential room for improvement, since it should be able to theoretically learn for this specific task, it is simply a matter of training on the right distribution of data, and determining a more optimal set of hyperparameters. Another area for future work is in the final stages of the image processing pipeline; we implemented a simple contrast enhancement stage to boost the

brightness and contrast of the deconvolved image, although this could be better tuned. Finally, we would also like to try the pipeline on real, raw captured long exposure data on a DSLR camera, to see if real data performs differently than star trail images obtained from the internet that often generated through stacking of shorter exposures instead of one long exposure.

References

- [1] B. Sease and B. Flewelling. Polar and spherical image transformations for star localization and RSO discrimination. *25th AAS/AIAA Space Flight Mechanics Meeting*. 1993.
- [2] Su et al. Richardson-Lucy deblurring for the star scene under a thinning motion path. *Satellite Data Compression, Communications, and Processing XI*. 2015.
- [3] J. Yen and P. Bryan Low-cost Computational Astrophotography. EE 367 Project. 2019.
- [4] S. Boyd, L. Vadenberghe. Convex Optimization. Cambridge University Press. 2004.
- [5] W. Gander, G.H. Golub, and R. Strebel. Least-Squares Fitting of Circles and Ellipses. *BIT Numerical Mathematics* 34 no. 4. 1994.
- [6] Z. Drezner and J. Brimberg. Fitting concentric circles to measurements. *Mathematical Methods of Operations Research* 79 no. 1. 2014.
- [7] D. Ioannou, W. Huda, and A.F. Laine. Circle recognition through a 2D Hough Transform and radius histogramming. *Image and Vision Computing*. Vol. 17 Issue 1. 1999.
- [8] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*. Vol. 3 1-122. 2011.
- [9] O. Liba, K. Murthy, Y.T. Tsai, T. Brooks, T. Xue, N. Karnad, Q. He, J.T. Barron, D. Sharlet, R. Geiss and S.W. Hasinoff, 2019. Handheld mobile photography in very low light. *ACM Transactions on Graphics (TOG)*, 38(6), pp.1-16.