Camera Trajectory Optimization for Maximizing Optical Character Recognition on Static Scenes with Text ⋆

Alexander Zabaldo * **Jun Ueda** *

* *Georgia Institute of Technology, GA 30332 USA (e-mail: [alex.zabaldo@gatech.edu] [jun.ueda@me.gatech.edu])*

Abstract: Camera systems in motion are subject to significant blurring effects that lead to a loss of information during the image capture. This is especially damaging for optical character recognition for which edge preservation is critical to achieving a high recognition rate. Using non-blind motion deblurring, a trajectory and point spread function can be designed to maximize the recognition rate while meeting endpoint constraints. Optimization through the use of radial basis function networks can therefore be used as a way to find ideal trajectories to reduce blurring effects and preserve text sharpness. This paper investigates this problem using simulation of a blurred image capture process. The simulation is automated using radial basis function network optimization and a genetic algorithm to determine trajectories with the best recognition rate. Optimized trajectories yielded recognition scores with up to 65% improvement compared to a comparable linear profile. Results are then analyzed using spectral analysis to understand why the chosen trajectories preserve text edges. These findings can be applied to a wide variety of controlled mobile camera platforms, such as autonomous automobiles or unmanned aerial vehicles, to improve their ability to gather information from their environment.

Keywords: Path Planning and Motion Control, Modeling, Identification and Signal Processing, Control Applications

1. INTRODUCTION

Cameras allow systems to extract visual information from their environment but can be subject to many types of error and degradation. One such form is motion blur, which can occur either as movement of the camera or movement of objects in the scene (Cai et al., 2016). Due to the nature of a limited exposure window, blur occurs as a result of light being captured across multiple photosensitive elements. In some cases, the trajectory of the camera may be known due to its positioning on a controlled mobile platform, such as an autonomous vehicle or unmanned aerial vehicle (Balaji et al., 2017). If the motion of the camera is known, it is possible to calculate a blur kernel that can be used with traditional deconvolution methods to restore the image (Su and Heidrich, 2015; Yadav et al., 2016). There has been significant work in this area on using dynamics-based motion deblurring for reducing image noise (Dengel et al., 2016; Kurimo et al., 2009; Tai et al., 2011), which is best suited for object recognition. Text recognition, as is relevant for this study, depends less on image noise and significantly more on edge preservation in order for text features to be extracted by text recognition engines (Katoch and Ueda, 2019). The known work completed specifically for text recognition has only used polynomial functions to generate trajectories for camera motion. This is inherently limiting as it sets a maximum complexity that the trajectory can achieve. This paper instead proposes the use of radial basis function networks for trajectory generation as a way to create highly tunable motion profiles without knowledge of a particular function form. These trajectories are optimized for text character recognition through an automated simulation.

⋆ This work was supported by the National Science Foundation under Grant 1662029. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

2. PROBLEM STATEMENT

The goal of this work is to establish an optimal, one-dimensional camera trajectory that, compared to other trajectories with the same motion and endpoint constraints, can be used to plan the motion of a mobile camera, such as in Figure 1, that will allow for the greatest rate of correct text recognition in natural images after image reconstruction. Trajectories are generated using radial basis function networks in a one-dimensional space and subject to optimization via a cost function based on the correct text recognition rate. Numerical optimization is used to attempt motion deblurring in an automated fashion against a library of text-based natural images.

3. RADIAL BASIS FUNCTION NETWORKS

Previous work in dynamics based motion deblurring utilized linear, inverse error, and fourth order polynomial functions as proposed trajectories for improving image

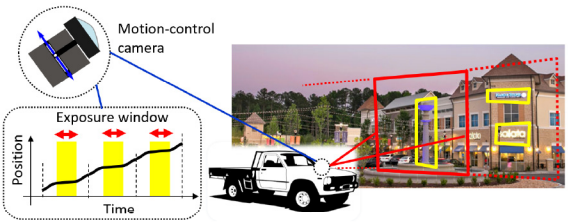

Fig. 1. Vehicle application of a mobile camera platform for street sign text recognition

sharpness (Katoch and Ueda, 2019). However, these functions are limited in shape variation given position and velocity endpoint constraints. Linear and inverse error functions with constrained endpoints only have a single possible trajectory, and polynomial functions can change coefficients, but are limited in function variation at specific points in the trajectory. This paper instead proposes the use of radial basis function (RBF) networks as a way to increase the complexity of the function form while maintaining trajectory constraints. An RBF is any real valued function in which the value is solely dependent on the difference between the input and some given center, defined as $\varphi(x) = \varphi(||x - c||)$ where x is the primary function variable and c is the RBF center (Gutmann, 2000). The most common RBF used is the Gaussian function:

$$\varphi(r) = e^{-(\epsilon r)^2}, \tag{1}$$

where r is the primary function variable and ϵ is the Gaussian shape factor. The shape factor is directly related to the standard deviation of the Gaussian and therefore controls each function's spread around a particular center. It is typically fixed for a given set of RBFs. Radial basis function networks use multiple RBFs and an equal number of weights and function offsets to generate a weighted sum of the functions:

$$s(t) = \sum_{k=0}^{n} w_k \varphi(||t - t_k||), \tag{2}$$

where $s(t)$ is the RBF network function, n is the number of RBF functions used, w_k is the function weight, t_k is the function offset, and t is the trajectory timescale. An example of the formation of an RBF network is shown in Figure 2.

One advantage of RBF networks is the ability to easily and predictably vary their shape using the weight and offset values (Mirinejad and Inanc, 2015). Given an RBF network of n RBF functions, parameters to fully describe the network can be formed as a series of n $[w_k, t_k]$ pairs. This set of parameters can be used as inputs for a number of optimization or search methods to determine optimal sets with a given cost function. It is also known that any combination of values is guaranteed to produce a smooth function.

4. DYNAMICS BASED MOTION DEBLURRING AND POINT SPREAD FUNCTION DETERMINATION

4.1 Image Formation

Images are formed based on the intensity of light exposed onto the photosensitive elements of the camera sensor. For a camera undergoing one-dimensional motion, an exposure

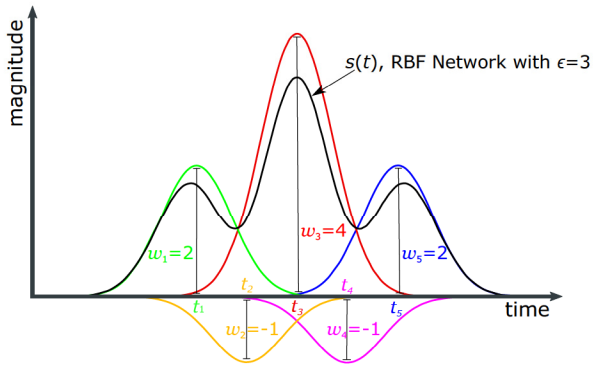

Fig. 2. An example five function symmetric RBF network with $\epsilon = 3$ and individual functions shown

position x_e is established as the average or center position during the entire exposure window T_e:

$$x_e = \frac{1}{\Delta T} \int_{T_e} x(t)dt, \tag{3}$$

where T_e is the exposure timescale, ΔT is the total exposure time and $x(t)$ is camera position.

During this image capture process, two types of noise are introduced: shot noise and thermal noise (Katoch et al., 2018). Shot noise is the result of variance in the number of photons received by each photosensitive element over time. Thermal noise $\boldsymbol{N}$ is created within the camera sensor via variation in the electrical signal due to changes in temperature and is modeled as an additive zero-mean Gaussian $\mathcal{N}$ with variance σ^2. Shot noise can be modeled as a blurry image $\boldsymbol{B}$ generated by a stationary Poisson process $\mathcal{P}$ with intensity λ:

$$\boldsymbol{B} \sim \mathcal{P}\left(\lambda \int_{T_e} \boldsymbol{L}(x(t))dt\right), \boldsymbol{N} \sim \mathcal{N}(0, \sigma^2), \tag{4}$$

where $\boldsymbol{L}$ is the latent image. The captured image $\boldsymbol{I}$ can then be defined as:

$$\boldsymbol{I} = \boldsymbol{B}(x(t), T_e, \lambda) + \boldsymbol{N}, \tag{5}$$

or as:

$$\boldsymbol{I} = \boldsymbol{K} \otimes \boldsymbol{L} + \boldsymbol{N}, \tag{6}$$

where $\boldsymbol{K}$ is the blur kernel or Point Spread Function (PSF) and $\otimes$ is the convolution operator.

4.2 Residence Time Distribution and Point Spread Function

Knowledge of the camera's motion is critical as it is needed to generate a residence time distribution (RTD). An RTD describes the amount of time the camera spends over a certain section of the image (Katoch et al., 2018). These image sections evenly divide the image based upon a set step size, image capture exposure time, and linear capture distance. The RTD is represented graphically in Figure 3 and in the equation below:

$$\tau(x) = \frac{1}{\|\dot{x}(x)\|_2}, \tag{7}$$

where $\dot{x}(x)$ is the velocity of the camera at a given point over the image and $\tau(x)$ is the RTD. For this paper, it is assumed that the trajectory is one-dimensional as the vertical motion is negligible relative to the horizontal velocity. In the case that the camera is stationary, the

result is a Dirac delta function centered on the trajectory center point with a value of the image exposure time. The sum of the RTD is equal to the exposure time ΔT for any given trajectory.

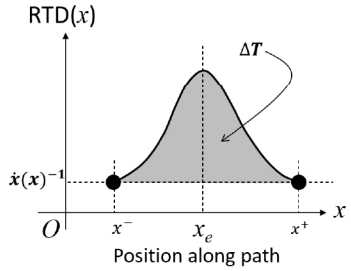

Fig. 3. Residence time distribution across a non-stationary trajectory

Using the RTD found via the known trajectory, the PSF can be determined. This PSF is used as the blur kernel for any deconvolution method as it describes the light energy distribution for a pixel in a captured image. For this one dimensional case, the PSF is represented by the RTD normalized and equally sampled to the size of the blur kernel imposed on the center row with all other elements equal to zero.

4.3 Non-Blind Motion Deblurring

The reconstruction of a blurred image via non-blind deblurring is based upon the formation of the captured image described in (6). Richardson-Lucy deconvolution is a well-known iterative approach that is computationally slower and more intensive than other methods, but produces more accurate results and is therefore used here. It can be modeled as:

$$\boldsymbol{I}^{j+1} = (\boldsymbol{B} \oslash (\boldsymbol{I}^{j} \otimes \boldsymbol{K}) \otimes \overline{\boldsymbol{K}}) \odot \boldsymbol{I}^{j}, \tag{8}$$

where $\boldsymbol{I}^{j}$ is the image estimate at the j-th iteration, $\overline{\boldsymbol{K}}$ is the blur kernel flipped along the central horizontal and central vertical axes, $\odot$ is the Hadamard product, and $\oslash$ is the Hadamard divide (Kim and Ueda, 2016). In this one-dimensional case, the $\overline{\boldsymbol{K}}$ matrix is the same as $\boldsymbol{K}$ as this matrix is a single row of symmetric values along the center row with all other values equal to zero.

5. OPTIMIZATION

Proposed RBF based trajectories must be assessed and continually modified in order to find the ideal trajectory for planned deblurring. To do this, an optimization problem can be created using weights of the RBF as an input and the assigned score as output to minimize. The RBF offsets are not modified, and instead are equally spaced across the timescale. This is because with a sufficiently large number of functions and sufficiently large shape factor, the RBF network summation can still achieve a wide range of trajectories. Constraints are also imposed on the position endpoints x_0 and x_f, velocity endpoints v_0 and v_f via a chosen velocity $\bar{v}$, and acceleration $a(t)$ through a maximum allowable acceleration a_{allow} determined by the performance limit of a mobile camera platform actuator. The optimization problem formation is shown below:

$$\underset{\boldsymbol{w} \in R^n}{\text{minimize}}\ R_{OCR}(\boldsymbol{w}) \tag{9.1}$$

$$\text{subject to } x_0 = 0 \tag{9.2}$$

$$x_f = d_{\text{exp}} \tag{9.3}$$

$$v_i = v_f = \bar{v} \tag{9.4}$$

$$\max(a(t)) \leq a_{\text{allow}}, \forall t \in [0, T_e] \tag{9.5}$$

where $\boldsymbol{w}$ is the RBF weights vector:

$$\boldsymbol{w} = (w_1, w_2, ...w_n)^T. \tag{9.6}$$

The position vector is calculated as the numerical integral of the velocity vector $x(t) = \int_0^{\Delta T} v(t)dt$ while the acceleration is the numerical derivative $a(t) = \frac{d}{dt}v(t)$. Maximum acceleration is defined as the largest acceleration value over $[0, T_e]$. a_{allow} is chosen based upon the limits of the physical system for which the trajectory optimization is to be applied and imposes a non-linear constraint in this problem.

The velocity profile is the function generated by the RBF network so that the position endpoints can be easily constrained. The total distance traveled for any RBF is the area under the given function (in this case, the Gaussian) multiplied by the sum of all weights. From (2):

$$d_{\text{exp}} = \left(\int_{T_e} e^{-(\epsilon r)^2} dr \sum_{k=1}^{n} w_k \right) + \bar{v}\Delta T. \tag{10}$$

This can be used as a linear constraint to ensure the intended distance traveled is always met. The cost function $R_{OCR}(\boldsymbol{w})$ is the percent of incorrect characters in the text recognized string compared to the correct string measured by the Levenshtein distance algorithm.

5.1 Image Evaluation

Determination of an image evaluation method is a common issue for problems involving image restoration (Narwaria et al., 2012). For this experiment, the returned character recognition text is compared to the ground truth text to form the rate of incorrect text recognition as determined by the Levenshtein distance algorithm (Priambada and Widyantoro, 2017). This algorithm determines the number of insertions, deletions, or character changes needed to be applied to a string in order to make it match the goal. The number of changes is divided by the recovered string length to create an incorrect text recognition rate:

$$R_{\text{OCR}}(\boldsymbol{w}) = \frac{n_{\text{L}}}{l_{\text{Text}}}, \tag{11}$$

where R_{OCR} is the rate, n_{L} is the number of string changes according to the Levenshtein distance algorithm, and l_{Text} is the length of the text recovered from the optical character recognition. One issue with this evaluation metric is the possibility of a significant difference in returned and actual string length, which may cause R_{OCR} to be significantly greater than 1. For this reason, a response of no text being recognized in the image is given a score of 3 to encourage the simulation to move towards any text recognition even if it contains many extra or missing characters. Any R_{OCR} over 3 are also reduced to 3 as well. Finally, any score values under 3 are scaled based on the mean R_{OCR} value of the ground truth image as a correction for some images being inherently more difficult to extract text from.

5.2 Optical Character Recognition

Optical Character Recognition (OCR) is the process of converting text within images into machine-encoded text. A well known and widely used OCR software is Google's open source Tesseract (Smith, 2007), which was used within MATLAB's optimization process. Although many OCR algorithms exist, this experiment required the use of one that could be automated to run continuously without user input, making Tesseract an obvious choice. This engine performs well and is primarily used for document text, but preprocessing and adjustments should be introduced in order to use it for natural scene images with text. This preprocessing includes first converting the image to a binary image, then subjecting it to simple morphological operations to clean up small noise. Maximally stable extremal regions (MSER) regions are then detected in an attempt to identify potential text shapes (Greenhalgh and Mirmehdi, 2015; Islam et al., 2016). These regions are subjected to property restrictions, such as area and solidity, to filter out likely non-text regions. The bounding boxes of these regions are then slightly expanded to find likely candidates for words and individual lines of text. Segmented portions of the image can then be cut from the overall image and given to Tesseract to return the identified text. This process is done automatically for every image during the optimization cycle.

5.3 Spectral Analysis

Spectral analysis is the process of deconstructing a signal in the time domain into components in the frequency domain. For the purposes of this experiment, a two-dimensional Discrete Fourier Transform is used to view and analyze frequency information within the deblurred images. Specifically, edges, such as those formed by text, tend to appear as high frequency values in the direction perpendicular to the edge in the two-dimensional power spectrum (Narwaria et al., 2012). In the case of this one-dimensional, horizontal motion, the appearance and retention of high frequency values in the center horizontal region is critical to OCR's performance.

6. SIMULATION SETUP

Determining an ideal trajectory for character recognition requires an automated optimization process that can quickly evaluate many sets of RBF weights. A simulation of the process of capturing an image, deblurring it based on a known trajectory, and evaluating it was created in MATLAB. For simplicity, all blur kernels are aligned to the horizontal axis of the blur kernel as this does not affect overall results and other blur directions can be produced through a simple transformation

For any single iteration, a ground truth image, in this case, an image from the provided data set, is supplied and subjected to horizontal linear motion blur determined by the motion of an RBF trajectory with arbitrary weights. Noise is added to the blurred image, and the image is then deblurred with Richardson-Lucy deconvolution using the same blur kernel as the original blurring. OCR is then completed on the image, and the extracted text is compared to a correct text string using Levenshtein distance. The final score is calculated using (11). The entire scoring process is completed for each image in the set using the same trajectory, and each individual image score is averaged for the set to form the epoch score. The epoch score is returned to MATLAB's genetic algorithm to determine the next suitable trajectory parameters to test. This process is repeated until a trajectory is found that satisfies the constraints in (9) as well as the optimization parameters. Figure 4 shows this optimization algorithm in flowchart form.

For this experiment, due to computing restraints, the simulation would exit upon the completion of 200 epochs. From these trajectories, the best five were pulled and run again as single tests for five trials in order to prevent improper scoring due to randomness in the OCR algorithm. Linear and inverse error trajectories were also tested as a comparison method.

Because individual natural images have only a small portion of text, a set of images is used for the optimization. For this experiment, a combination of the training and test set of images from the ICDAR 2015 OCR competition was used (Karatzas et al., 2013). This comprises a library of 462 images, however, this was reduced down to 20 by filtering for images that scored an average of no greater than 0.5 in R_{OCR} value when evaluated as static images. The final set contains 524 total characters.

Table 1 shows the simulation parameters used for the optimization. The number of RBFs and ϵ value were chosen based upon what combination could reasonably be used to fully represent almost any trajectory between endpoints while reducing computation time. $\bar{v}$ is based upon the linear velocity needed to meet position endpoint constraints and a_{max} was chosen as a high upper limit of what could be achieved with a future hardware setup.

Table 1. Simulation parameters

Simulation Parameter	Value
Number of Images	20
Maximum Epoch Iterations	200
Blur Kernel to Image Ratio	10%
Number of RBF Functions	13 (7 unique)
ϵ	4
$\bar{v}$	0.3 m/s
a_{max}	$100g$
ΔT	0.017 s (60FPS)
d_{exp}	5 mm

7. RESULTS

The five RBF-based trajectories, as well as the tested linear and inverse error trajectories, are shown in Figure 5 with a sample of a deblurred image, the velocity profile, and the residence time distribution. The performance of each of these trajectories is shown in Table 2. All of the RBF trajectories show significant improvement over the linear and inverse error trajectories with $p \ll 0.05$ statistical significance. This is apparent from the data as well as from qualitative observation of the sample images. This improvement ranges from 59-65% compared to the linear and inverse error trajectories, with the second and fourth RBF trajectories performing the best overall.

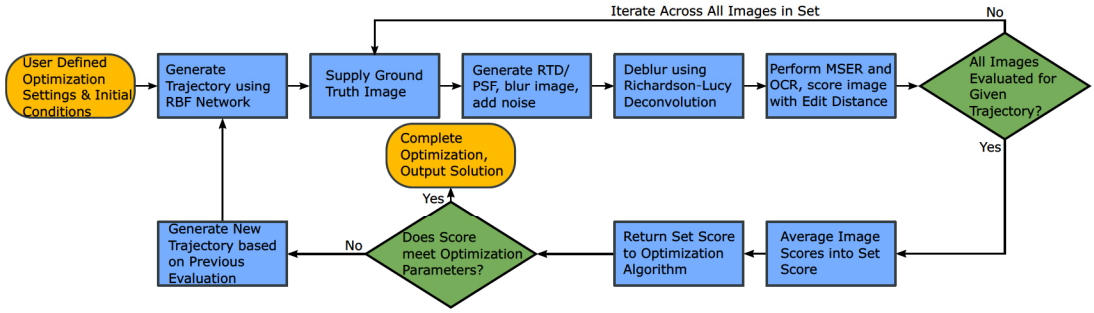

Fig. 4. Flowchart of the simulation's optimization algorithm

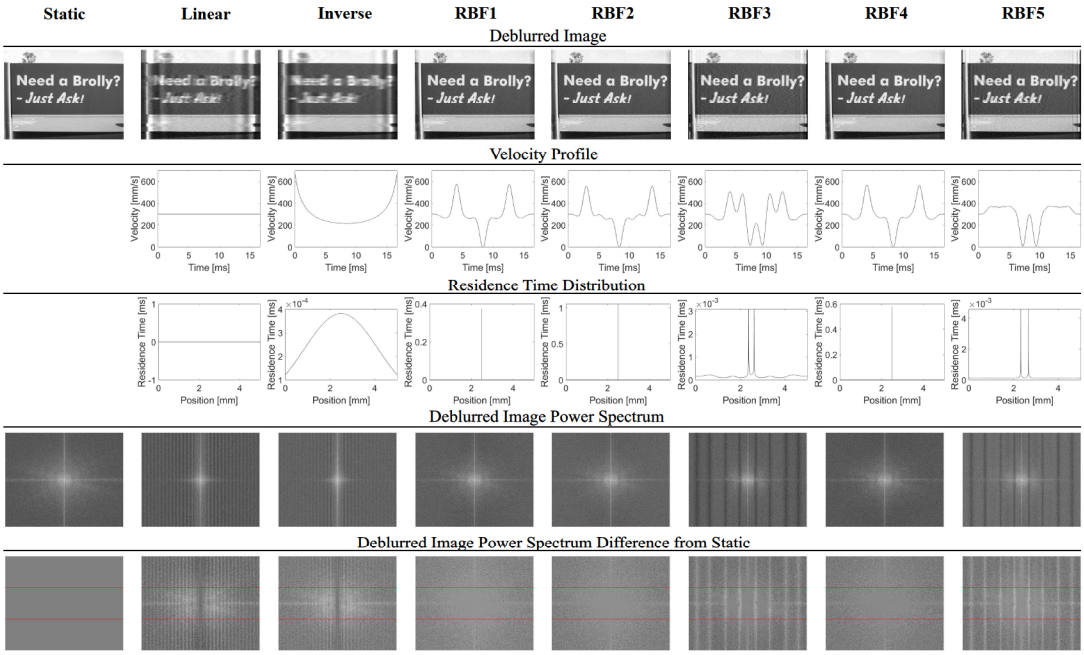

Fig. 5. Simulation results for all trajectories including deblurred images, velocity profiles, residence time distributions, and power spectrum plots

Table 2. Simulation score mean μ_{OCR} and standard deviation σ_{OCR} over five trials

Trajectory	μ_{OCR}	σ_{OCR}
Static	0.2793	0.03963
Linear	2.428	0.2034
Inverse	2.447	0.1717
RBF1	0.9517	0.07508
RBF2	0.8534	0.1131
RBF3	0.9945	0.2702
RBF4	0.8821	0.09565
RBF5	0.9824	0.08133

Spectral analysis was performed using the two-dimensional Discrete Fourier Transform on an image formed by the mean of all final images for the five trials of each trajectory. Figure 5 shows the power spectrum of the sample image for each trajectory as well as the power spectrum difference of each trajectory from the static image power spectrum. On the difference power spectrum, lines are imposed to show the central third of the image that most represents the relevant horizontal frequencies. The static image power spectrum shows high frequency components in both the purely vertical and horizontal directions representing the edges of the box in the image as well as the edges of the letters in these directions, especially for characters such as "N", "d", and "l". There is also a large amount of lower frequency information in all directions representing the curved and slanted edges of the letters. For the linear and inverse error trajectory plots, there is a significant removal of these regions as this information has been distorted by the blur. The subtraction plots show loss of both the high frequency horizontal components as well as lower frequency near-horizontal regions. This loss is much less prevalent in all of the RBF trajectory power spectra. In addition, the presence of deblurring artifacts in the third and fifth images are shown in bands throughout the power spectrum that interrupt the horizontal component. The sum of the differences offset to a minimum of zero for the central horizontal third of each image is calculated in Table 3. This shows a very close correlation between the maintenance of these image frequencies and the text recognition rate undergoing one-dimensional blur. Again, RBF trajectories two and four performed the best here as well.

Table 3. Power spectra difference plot value-sum mean μ_{PS} and standard deviation σ_{PS}

Trajectory	μ_{PS}	σ_{PS}
Static	0	0
Linear	4.181E5	5.829E3
Inverse	4.206E5	6.408E3
RBF1	3.327E5	3.472E4
RBF2	3.315E5	3.383E4
RBF3	3.822E5	1.482E4
RBF4	3.186E5	3.103E4
RBF5	3.804E5	1.833E4

8. DISCUSSION

The simulation results in Figure 5 show an interesting trend; all RTDs for the RBF-based trajectories show either a single spike or two spikes in residence time at the center of the trajectory. This makes sense, as this translates to a blur kernel with the highest values very close to the center that generates less blur to begin with. A single peak RTD, such as in RBF 1, 2, and 4, maximizes the residence time around the center of the image, while a double peak RTD, shown in RBF 3 and 5, is likely a consequence of the optimization attempting to lower the maximum acceleration while also minimizing the OCR cost function. This creates multiple phase distortions in the image that can be stitched together during the deblurring process which leads to strong artifacts, but still produces high contrast edges for text recognition. The concentration of residence time at the trajectory center is directly related to the overall OCR performance and indicates a possible attempt of saccade motion. Further investigation is needed to determine the relationship between the RTD, maximum allowable acceleration, and effect on the OCR performance.

9. CONCLUSION

This paper has proposed a novel optimization method for determining ideal, endpoint-constrained trajectories for maximizing recognition of text in natural images. RBF networks provide the ability to automatically generate smooth functions with a high degree of variability while being controlled by a small set of tunable parameters that can be easily mapped to the character recognition-based output score. It was found that trajectories with a high central residence time performed best at maintaining edge features while still meeting all physical constraints. Future work will seek to validate these results through hardware experimentation using a camera-mounted voice coil linear stage with results to be published in a journal paper.

REFERENCES

Balaji, y., Kumar, M.B., and Sujatha, y. (2017). Text information extraction and analysis for autonomous vehicle. In *2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, 1–6. IEEE.

Cai, C., Liu, A., and Zhang, B. (2016). Motion deblurring from a single image. In *2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 406–410. IEEE.

Dengel, A., Asad, F., Ul-Hasan, A., and Shafait, F. (2016). High performance ocr for camera-captured blurred documents with lstm networks. In *DAS 2016, 12th Int'l IAPR Workshop on Document Analysis Systems*. IEEE.

Greenhalgh, J. and Mirmehdi, M. (2015). Recognizing text-based traffic signs. *IEEE Transactions on Intelligent Transportation Systems*, 16(3), 1360–1369.

Gutmann, H.M. (2000). A radial basis function method for global optimization. *Journal of Global Optimization*, 19.

Islam, M.R., Mondal, C., Azam, M.K., and Islam, A.S.M.J. (2016). Text detection and recognition using enhanced mser detection and a novel ocr technique. In *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, 15–20. IEEE.

Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., Bigorda, L.G.i., Mestre, S.R., Mas, J., Mota, D.F., Almazàn, J.A., and de las Heras, L.P. (2013). Icdar 2013 robust reading competition. In *12th International Conference on Document Analysis and Recognition*, 1484–1493. IEEE Computer Society, USA.

Katoch, R., Fusaro, B., and Ueda, J. (2018). Inverse error function trajectories for image reconstruction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 7527–7532.

Katoch, R. and Ueda, J. (2019). Edge-preserving camera trajectories for improved optical character recognition on static scenes with text. *IEEE Robotics and Automation Letters*, 4(4), 4467–4474.

Kim, M.D. and Ueda, J. (2016). Real-time panoramic image generation and motion deblurring by using dynamics-based robotic vision. *IEEE/ASME Transactions on Mechatronics*, 21(3), 1376–1387.

Kurimo, E., Lepistö, L., Nikkanen, J., Grén, J., Kunttu, I., and Laaksonen, J. (2009). The effect of motion blur and signal noise on image quality in low light imaging. In A.B. Salberg, J.Y. Hardeberg, and R. Jenssen (eds.), *Image Analysis. SCIA*, 81–90. Springer Berlin Heidelberg.

Mirinejad, H. and Inanc, T. (2015). A radial basis function method for direct trajectory optimization. In *2015 American Control Conference*, 4923–4928.

Narwaria, M., Lin, W., McLoughlin, I.V., Emmanuel, S., and Chia, L.T. (2012). Fourier transform-based scalable image quality measure. *IEEE Transactions on Image Processing*, 21(8), 3364–3377.

Priambada, S. and Widyantoro, D. (2017). Levensthein distance as a post-process to improve the performance of ocr in written road signs. In *Second International Conference on Informatics and Computing*, 1–6.

Smith, R. (2007). An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition*, volume 2, 629–633.

Su, S. and Heidrich, W. (2015). Rolling shutter motion deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1529–1537. IEEE.

Tai, Y.W., Tan, P., and Brown, M.S. (2011). Richardson-lucy deblurring for scenes under a projective motion path. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8), 1603–1618.

Yadav, S., Jain, C., and Chugh, A. (2016). Evaluation of image deblurring techniques. *International Journal of Computer Applications*, 139, 32–36.