Online Lens Motion Smoothing for Video Autofocus

Abdullah Abuolaim[1]
[1]York University, Toronto
abuolaim@eecs.yorku.ca

Michael S. Brown[1,2]
[2]Samsung AI Center, Toronto
mbrown@eecs.yorku.ca

Abstract

Autofocus (AF) is the process of moving the camera's lens such that desired scene content is in focus. AF for single image capture is a well-studied research topic and most modern cameras have hardware support that allows quick lens movements to optimize image sharpness. How to best perform AF for video is less clear. Conventional wisdom would suggest that each temporal frame should be as sharp as possible. However, unlike single image capture, the effects of the lens movement is visible in the captured video. As a result, there are two parameters to consider in AF for video: sharpness and lens movement. In this paper, we show that users preferred videos with smooth lens movement, even if it results in less overall sharpness. Based on this observation, we propose two novel AF algorithms for video that strive for both smooth lens movement and sharp scene content. Specifically, we introduce (1) a bidirectional long short-term memory (BLSTM) module trained on smooth lens trajectories and (2) a simple weighted moving average (WMA) method that factors in prior lens motion. Both of these methods have demonstrated excellent results in terms of reducing lens movements (up to 64% reduction) without greatly affecting the sharpness (less than 5.2% change in sharpness). Moreover, videos produced using our methods are more preferred by users over conventional AF that aims only for maximizing sharpness.

1. Introduction

Autofocus (AF) algorithms adjust a camera's optics to focus on a region in a scene. Focusing can be performed by either adjusting the lens aperture to change the depth of field (DoF) or by moving the lens until scene content lies within the DoF. AF is generally restricted to lens motion, since changing the DoF alters the amount of out-of-focus blur. On smartphone cameras, optics have fixed apertures and moving the lens is the only means of AF.

When capturing a single image, the goal of AF is clear – move the lens to maximize image sharpness for some desired scene content and then capture the image. When capturing a video (i.e., temporal images), the goal is less clear. The challenge for video AF is often attributed to determining when and where to shift the focus while the scene content changes. The obvious solution for video AF – and one used by most digital cameras – is to apply the single image strategy to each incoming frame to maximize the overall image sharpness [1, 4, 17]. However, this approach overlooks the fact that the lens motion can be observed in the captured video. Too much motion can result in undesired 'lens wobble'.

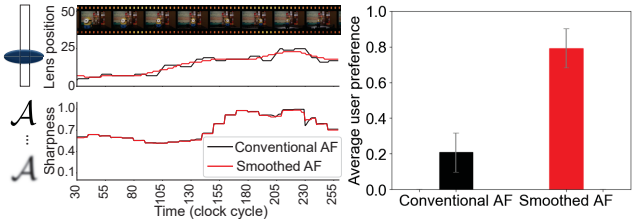


Figure 1: The left-hand plot shows lens motion and per-frame sharpness for a *conventional* AF algorithm and its corresponding smoothed lens movements. The right-hand plot shows the preferences of 32 users for six videos using *conventional* AF and those that have had their lens movement smoothed.

Recent work by Abuolaim et al. [1] offered an interesting finding regarding AF for video, lens motion, and users' preference. Their work created a new AF dataset and research platform to emulate the AF system on cameras. As part of their work, they evaluated several common AF approaches for video that strived to maximize per-frame sharpness. Abuolaim et al. [1] performed a user study to determine what part of the scene users preferred to have in focus — for example, a face region of interest (ROI) if a face was present in the video. They found that user preference *was not correlated* to a particular ROI, but instead correlated to the amount of lens motion in the overall output video. Specifically, users preferred videos that had the least amount of lens motion – irrespective of what ROI was used. The only exception was when a face was present. For such cases, users tolerated large lens movements to bring

the face into focus. Moreover, less lens motion not only is preferred by users but also reduces the power consumption required to move the lens [26]. The finding in [1] serves as our impetus to explore the idea of smooth lens motion in more detail. Fig. 1 shows an example.

Contribution We propose two new AF methods for video that incorporate *online* smoothing to reduce overall lens motion. To this end, we generate several lens trajectories for video sequences using conventional algorithms that target maximizing image sharpness. These lens trajectories are then smoothed in an *offline* manner (i.e., smoothing filter uses a window that considers prior and future lens positions) to produce videos with smoothed lens motion. We perform a user study to re-confirm that these smoothed lens trajectories are preferred over the videos targeting maximum image sharpness. These *offline* smoothed trajectories, however, cannot be performed in a real AF system because they require knowledge of where the lens will move in unseen frames. To perform *online* smoothing, we propose two methods: (1) a supervised bidirectional long short-term memory (BLSTM) module trained on smooth trajectories and (2) an unsupervised weighted moving average (WMA) method that considers scene sharpness and prior lens motion. We demonstrate good quantitative and qualitative results that show our methods can significantly reduce lens motion (up to 64%) with a very small loss in sharpness (less than 5.2%). We show that both algorithms produce outputs more preferred than videos produced with no smoothing. Users had a slightly higher preference for the WMA approach. This is an interesting finding as the WMA approach is unsupervised and can be easily incorporated into existing AF camera systems with little computational overhead.

2. Related Work

This section discusses existing work related to AF for cameras, data smoothing, and recurrent neural networks (RNN) for time series data.

AF for Cameras Contrast detection autofocus (CDAF) and phase difference autofocus (PDAF) are the most common techniques used in camera AF. CDAF uses image features, such as gradient magnitude, to estimate the sharpness in an ROI [3, 14]. CDAF approaches need to move the lens back and forth to capture different images to determine which direction to move the lens to maximize the sharpness [24].

PDAF is a newer technology that works at a hardware level. PDAF computes the sharpness of an ROI by examining the image shift (phase difference) between left and right sub-apertures of the primary lens. There are two common designs for PDAF: (1) line sensor with half sub-mirror design used in older DSLR cameras [12, 19] and (2) on-sensor dual-pixel technology used in recent DSLR and smartphone cameras [13, 22]. PDAF allows an optimal lens position to be computed with a *single* image capture. PDAF has significantly improved cameras' ability to perform autofocus quickly and accurately. In this paper, we assume our camera uses a dual-pixel sensor, and the optimal lens position to maximize focus is known at each new input frame.

Prior work using both CDAF and PDAF have primarily targeted single image capture. There has been significantly less work on AF for video. Proposed methods for video AF apply the single-image approach to each frame (e.g., see [1, 4, 17]). One of the major limitations for research targeting AF for video had been the lack of access to a dataset that provides a full focal stack at each time point. Recently, Abuolaim et al. [1] introduced a 4D temporal focal stack dataset composed of ten image sequences, each containing 50–90 focal stacks of 50 images. This work also introduced a software platform that mimics the real AF implementation with constraints such as lens motion timing with respect to scene motion and frame capture. The data and AF platform from [1] enable the work in this paper.

Data Smoothing Data smoothing is a prevalent technique used in all areas of science. Readers are referred to [2, 6] for thorough surveys. Data smoothing aims to reduce undesired signal fluctuation while preserving higher-order moments of the original signal. The moving average (MA) is the simplest digital smoothing method [7]. Despite its simplicity, the MA is optimal for certain tasks, such as reducing random fluctuation while retaining a sharp step response.

Savitzky-Golay (SG) is an established data smoothing method [21]. This approach works by applying a least squares polynomial fit to a moving window of data points over time. The SG method has an advantage over the MA as it better preserves features of the original signal, such as peak height and width, which are usually attenuated by the MA method. In our work, we use the output of SG method as a proxy to the ground truth for training our BLSTM.

RNN for Time Series Data RNNs have been highly successful in their ability to learn dependencies in time series data by considering both the current input and the previous hidden state. There are multiple types of RNNs, such as the standard RNN, long short-term memory (LSTM) [11], BLSTM [9, 23, 25], and gated recurrent unit (GRU) [5]. RNNs have been successfully applied to many time series data tasks, including action recognition [18, 27], speech recognition [8], and image captioning [15].

Standard RNNs are known to have difficulty in learning long time dependencies [10]. Unlike RNNs, LSTMs have shown capability of learning long and short time dependencies. Nevertheless, both RNNs and LSTMs process inputs in forward order and learn from previous hidden states only. BLSTMs [9, 23, 25], therefore, were introduced to process each training sequence forwards and backwards in two separate LSTM nets in which they learn from previous as well as next hidden states. In this paper, we examine using BLSTM as an *online* AF smoothing method.

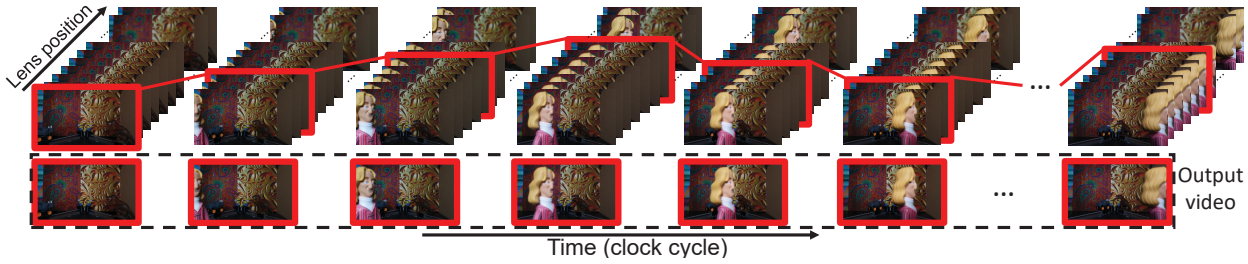


Figure 2: This figure shows an example of the 4D temporal focal stack data from [1]. The highlighted frames indicate the images that would be visible in the output video. The position of the highlighted images is achieved by moving the lens.

3. Smoothed Lens Motion for Video AF

The work in [1] found that users' preference for an AF system was correlated with the videos that had less lens motion and not what part of the scene was in focus (i.e., the ROI used). The only exception was when a face entered a scene. In this case, the ROI would be changed to the face, which could potentially incur a large change in lens position. This finding has led us to explore the idea proposed in this paper – namely, how to smooth lens motion while also attempting ROI sharpness. To this end, our aim is not to come up with a new AF objective in regards to "what to focus on", but rather an algorithm for temporal smoothing of lens motion in AF videos based on existing focusing objectives. In this section, we begin by describing the data used in our experiments. Afterwards, our *online* smoothing algorithms are described. This is followed by our quantitative and qualitative results. Lastly, we present our user study to determine which videos are preferred.

3.1. Video Focal Stack Dataset

We use the 4D temporal focal stack dataset and AF software platform provided by [1]. This dataset contains ten scenes (referred to as Scene 1–10), each consisting of 50–90 full focal stacks of 50 images each, 1 image for each possible lens positions. The AF platform is intended to emulate an AF system on a smartphone. Timing for the AF system is linked to an internal clock cycle, where movement of one lens position requires a single clock cycle. Frame capture is automatically triggered at each clock cycle. After each clock step, the optimal lens position is provided from the PDAF module depending on the AF ROI used. Motion in the scene (i.e., advancement to the next focal stack) is performed after ten clock cycles (i.e., a lens can move ten times before moving to the next focal stack). Thus, a scene with 90 focal stacks will be simulated as 30 seconds of video, at an effective frame rate of 30 frames per second, with 900 lens movements allowed. While this seems low, the motion of objects in the scenes is relatively slow (controlled by linear actuators) and the resulting videos appear visually consistent. The AF platform API provides four predefined AF objectives in terms of the ROI used – namely, global ROI targeting the whole image; 9 focus-points (9-FP) with 9 ROIs targeting the center of the frame; 51 focus-points (51-FP) with 51 ROIs; and a face region (FR).

We generate ten videos for each objective, with the exception of face region, since only six scenes contain faces. In total we have 36 videos (10 for each of the three objectives and 6 for the face region). With each output video, the API also provides meta-data, such as total number of lens movements, lens positions and the optimal (sharpest) lens positions based on PDAF at each clock cycle. Fig. 2 shows an example of the 4D temporal focal stack data from [1] with the frames used to generate the output video.

Fig. 3 shows a small sample output of the AF platform relevant to our task. At each clock cycle, the optimal lens position from the PDAF module is received, the current lens position, and the change in lens position. Lens motion at each clock cycle can take only one of three values: move forward (+1), move backward (-1), or no movement (0). Fig. 3-A shows output for the conventional method from [1]. Fig. 3-B shows *offline* smoothing applied on lens motions (Section 3.2). The smoothing is applied as a moving window (i.e., box filter) centered around the current lens position. Such *offline* filtering cannot be realized in a real camera system as it requires knowledge of future lens positions. This *offline* generated information will be used as part of our training data. The goal for an *online* AF system, given the past history of the lens positions and the current optimal lens target from the PDAF module, is to decide whether to move forward, backward, or not to move at each new clock cycle. This is illustrated in Fig. 3-C (Section 3.3 and 3.4).

3.2. Lens Smooth Training Data

Before describing our BLSTM method, we describe the *offline* data smoothing performed to compute the training data. We apply the common *offline* smoothing method – namely, Savitzky-Golay (SG). The SG method fits successive subsets of n adjacent lens positions $x_i, i = 1, ..., n$ with a low-degree polynomial. The successive subsets can be formed by sliding a time window of size l (odd number)

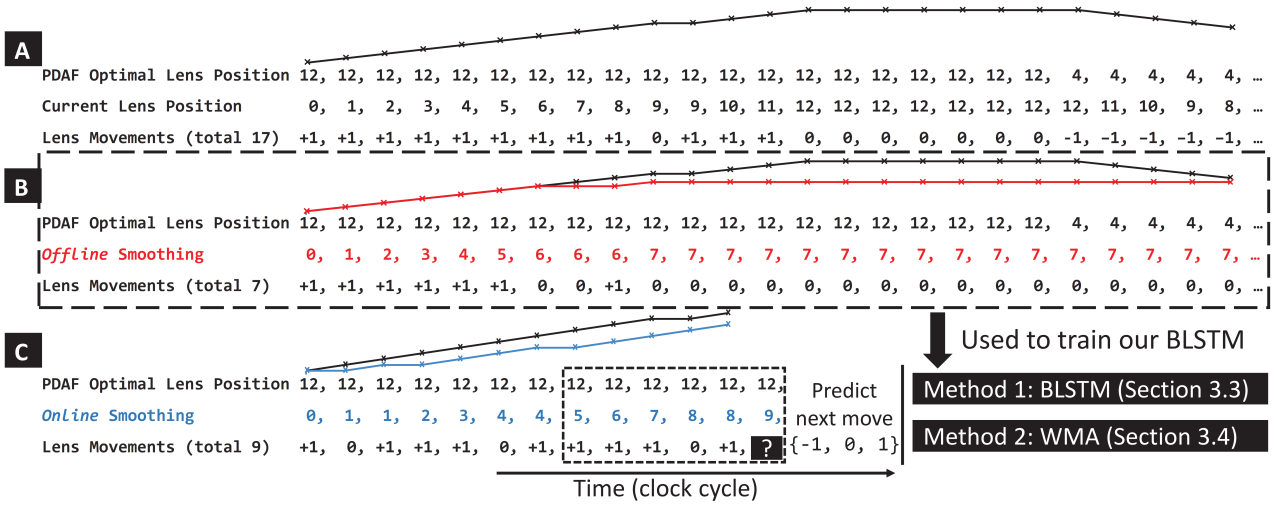


Figure 3: This figure provides an overview of the data used for our AF algorithms. (A) shows the output of the AF platform from [1]. At each clock cycle we have access to the optimal target lens position from the PDAF. We also have the current lens position as well as the next lens movement. (B) shows the results of applying *offline* smoothing to the actual lens positions. This is shown in red text. A plot of the lens position is also shown. (C) shows the problem of an *online* AF method. The *online* method must predict the next lens move based on prior observations.

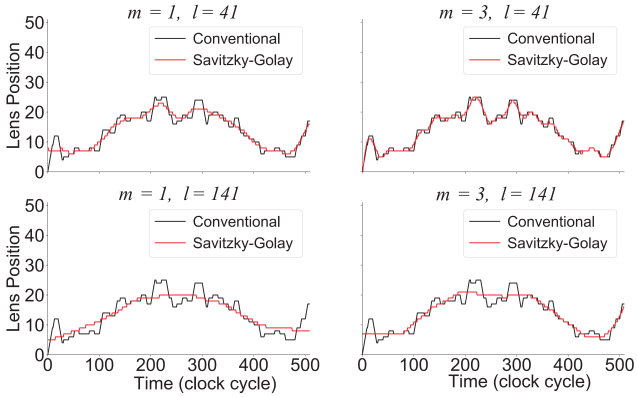


Figure 4: Savitzky-Golay method [21] applied on Scene 2 of the 9-FP objective. m is the polynomial degree and l is the time window size. By varying m and l, different smoothing trajectories can be applied.

and stride of 1. Then the smoothed lens position x_i^* can be computed as follows:

$$x_i^* = \frac{1}{l} \sum_{j \in N(i)} C_j \, x_j, \tag{1}$$

where $N(i)$ is the neighborhood lens positions centered around x_i of size l, $C_j \, x_j$ is the fitted value of x_j with a polynomial of degree m. Later, the final value of x_i^* is rounded to the nearest integer to be a valid lens position.

There are two hyper-parameters m and l used to control the strength of the smoothness applied. Fig. 4 demonstrates the direct effect of applying SG on Scene 2 of the 9-FP objective. As shown in this figure, different smoothing trajectories can be applied by varying m and l. For example, lower polynomial degree m imposes less flexible smoothing compered to the higher one. Similarly, larger window size l imposes stronger smoothing compared to the smaller one.

3.3. Online BLSTM

At each clock cycle i, the BLSTM module receives the current lens position x_i and the associated optimal lens position o_i coming from the PDAF module, as shown in Fig. 3-C. BLSTM also observes the prior lens positions for a certain time window of size l from $i - l$ to i. The goal is to predict the smoothed lens position x_{i+1}^*. Based on the AF API implementation, a camera lens is allowed to move only one step per clock cycle, $|x_{i+1} - x_i| = \{0, 1\}$, $\forall i$. Therefore, the problem can be presented as a classification problem of three classes, $Y = \{-1, 0, 1\}$: move backward, do not move, or move forward respectively. Thus, given the observed window of lens positions $\mathbf{X}_i = [(x_{i-l}, o_{i-l}), \ldots, (x_i, o_i)]$ and its associated label y_i, the task is to predict $f(\mathbf{X}_i)$ such that $x_{i+1}^* = x_i + f(\mathbf{X}_i)$.

Proposed BLSTM Architecture Our model uses two LSTM layers followed by a fully connected layer that outputs the vector $\mathbf{s} \in \mathbb{R}^3$ as shown in Fig. 5. The BLSTM architecture allows the output layer to capture information from past (backward) and future (forward) hidden states simultaneously. The vector $\mathbf{s}$ represents the scores of the three classes $\{-1, 0, 1\}$. The vector $\mathbf{s}$ and forward/backward

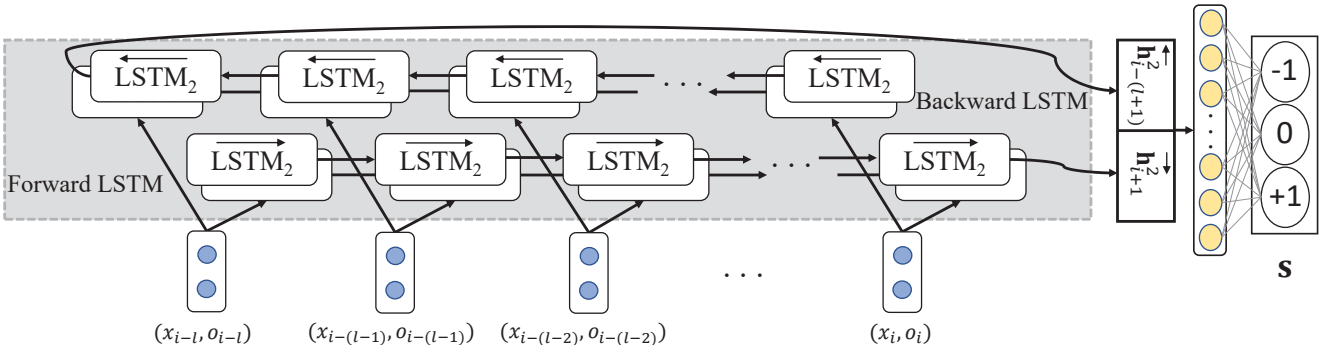


Figure 5: Our proposed BLSTM architecture of two LSTM layers followed by a fully connected layer. The term x_i is the current lens position and o_i is the optimal lens position (i.e., where the ROI would be sharpest) at clock cycle i. The output hidden states $\mathbf{h}$'s of both forward and backward LSTMs are concatenated to form our compact representation of time series data. The fully connected layer is added to output our score vector $\mathbf{s}$ that represents the scores of the three classes $\{-1, 0, 1\}$.

LSTMs are updated as follows:

$$\overrightarrow{\mathbf{h}}^1_{i+1} = \text{LSTM}_1(\overrightarrow{\mathbf{h}}^1_i, (x_i, o_i); \overrightarrow{\mathbf{W}}_1). \quad (2)$$

$$\overrightarrow{\mathbf{h}}^2_{i+1} = \text{LSTM}_2(\overrightarrow{\mathbf{h}}^2_i, \overrightarrow{\mathbf{h}}^1_{i+1}; \overrightarrow{\mathbf{W}}_2). \quad (3)$$

$$\overleftarrow{\mathbf{h}}^1_{i-1} = \text{LSTM}_1(\overleftarrow{\mathbf{h}}^1_i, (x_i, o_i); \overleftarrow{\mathbf{W}}_1). \quad (4)$$

$$\overleftarrow{\mathbf{h}}^2_{i-1} = \text{LSTM}_2(\overleftarrow{\mathbf{h}}^2_i, \overleftarrow{\mathbf{h}}^1_{i+1}; \overleftarrow{\mathbf{W}}_2). \quad (5)$$

$$\mathbf{s} = \mathbf{W}_{fc}[\overleftarrow{\mathbf{h}}^2_{i-(l+1)}, \overrightarrow{\mathbf{h}}^2_{i+1}] + \mathbf{b}_{fc}, \quad (6)$$

where LSTM_1 and LSTM_2 are the first and second LSTM layers. The $\mathbf{W}$ terms denote the different weight matrices, $\mathbf{h}$'s are the hidden states of the LSTM units, and $\mathbf{b}_{fc}$ is the bias vector of the fully connected layer. Since our proposed architecture is a bidirectional LSTM, we use right and left arrows to indicate forward and backward passes.

We set the dimension for each LSTM unit to 32, which results in a 64-dimensional fully connected layer at the top. We fix the size of the input time window l to 20. We use the Adam optimizer [16] to train our model with an initial learning rate of 0.0005, which is decreased by half every 1000 epochs. We train our model with mini-batches of size 128 using cross-entropy loss as follows:

$$\text{loss}(\mathbf{X}_i, y_i) = -\log\left(\frac{e^{\mathbf{s}[y_i]}}{\sum_j e^{\mathbf{s}[j]}}\right). \quad (7)$$

Cross-entropy loss is useful for classification problems where it increases as the predicted probability diverges from the actual label. During the training phase, we set the dropout rate to 0.1 to avoid overfitting. All the models described in the subsequent sections are implemented using Python with Pytorch framework and trained with a NVIDIA TITAN X GPU. In our experiments, it takes roughly 10 minutes to complete 1000 training epochs. We set the maximum number of training epochs to 5000. An ablation study with different LSTM architectures and settings is provided in the supplemental materials.

3.4. Online WMA

The moving average (MA) method is one of the simplest smoothing methods in the literature. The MA method requires only input lens positions over time for smoothing and does not utilize the target optimal lens position that comes for free from the PDAF module. In this section, therefore, we introduce our weighted moving average (WMA) method that uses the target optimal lens position to assign a weight w_i to each lens position x_i as follows:

$$x^*_i = \frac{\sum_{j=i-l}^{i} w_j\, x_j}{\sum_{j=i-l}^{i} w_j}, \quad (8)$$

$$w_j = e^{-\beta(|x_j - o_j|)}, \quad (9)$$

where β is our smoothing rate, and o_j is the target optimal lens position. The final value of x^*_i is rounded to the nearest integer to be a valid lens position. The reason behind introducing the weight w_i is to encourage lens movement with a large difference from o_j and suppress lens movements in case of a small difference from the current position. Parameters β and l are the hyper-parameters used to control smoothing strength.

Unlike the *offline* smoothing methods (i.e., SG and MA), our *online* WMA uses only the prior data for x^*_i calculations in which it is considered as an *online* smoothing method. The *offline* smoothing methods cannot be performed in a real AF system because they require knowledge of where the lens will move in unseen frames.

4. Experiments and Discussions

In this section we present the quantitative and qualitative results of our *online* smoothing methods: (1) a supervised BLSTM model trained on *offline* generated smooth data using SG method and (2) an unsupervised WMA method.

Evaluation Criteria For our quantitative results, we introduce four evaluation criteria:

- Accuracy: calculates the number of correctly classified samples over the total number of samples. This accuracy is measured according to how similar our BLSTM predictions are to the *offline* smoothed lens motion.
- Lens motion reduction: reports the percentage of lens movements reduced after smoothing with respect to the corresponding conventional method.
- Sharpness change: calculates the percentage of sharpness change after smoothing with respect to the corresponding conventional method. The sharpness of ROI is calculated following the procedure in [1].
- Time delay: measures the time delay between the original trajectory and our *online* smoothed trajectory by applying the cross-correlation method in [20].

Data Preparation For the BLSTM method, we first apply the *offline* SG smoothing for each of the 36 videos. We set window size $l = 141$ and polynomial degree $m = 3$ in order to impose strong smoothing with some flexibility. Next, we take the output smoothed lens positions to prepare the data for the BLSTM model as described in Section 3.3. We slide a time window of size 20 and stride of 1 to form our BLSTM input samples $\mathbf{X}_i$ and their corresponding y_i for each video data. For the WMA method, there are two hyper-parameters β and l used to control WMA's smoothing strength. We set β to 0.4 and l to 41 and empirically found these values are suitable.

BLSTM Training Procedure We partition the data based on independent scenes into training, validation, and testing data. As described in Section 3.1, for each ten scenes, there are three fixed ROI objectives (global, 9-FP, and 51-FP) where sharpness is computed. For those objectives we divide the scenes into pairs $\{\{1,8\},\{2,7\},\{3,6\},\{4,10\},\{5,9\}\}$, and then we perform cross-validation, where we take out-of-sample a pair of scenes, one for validation and another for testing. Each objective, thus, has five pairs and we need to train five models to report results for the ten scenes. For the last objective FR (face region), only six scenes contain faces and for those we perform cross-validation by taking an out-of-sample scene for validation and testing. For the FR objective, we need to train six models to report results for the six scenes. The overall number of trained models is $3 \times 5 + 1 \times 6 = 21$. We noticed that during training for some pairs there are lens positions that do not exist in the training set. For the network it is hard to predict for those values that are beyond the range of lens position values in the training set. To tackle this issue, we generate a video of reasonable lens positions that covers all the possible values 0-50 and augment this video in the training set.

Objective	**Accuracy**	**Objective**	**Accuracy**
Global	0.979	**9-FP**	0.922
FR	0.931	**51-FP**	0.837

Table 1: Average accuracy for each objective. Overall, average accuracy for all is mostly high, especially for the global one, because it always has single ROI and usually involves fewer lens movements.

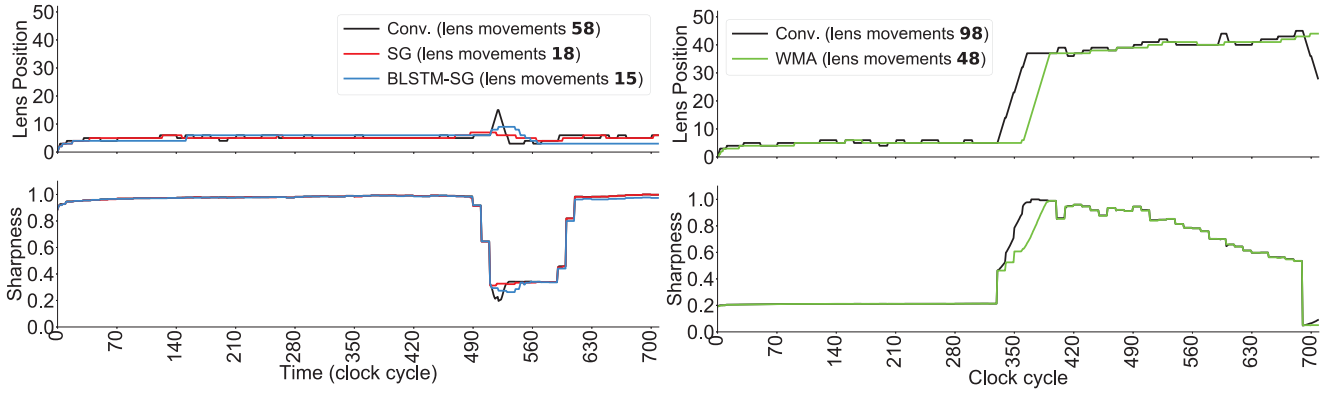
Objective	**Lens Motion Reduction**		
	Offline	*Online*	
	SG	BLSTM	WMA
Global	48.17%	33.38%	**44.49 %**
9-FP	52.32%	**63.61%**	50.17%
51-FP	49.54%	40.75%	**46.03%**
FR	25.53%	11.20%	**35.03%**

Objective	**Sharpness Change**		
	Offline	*Online*	
	SG	BLSTM	WMA
Global	-0.07%	-0.47%	**-0.13%**
9-FP	-0.46%	-2.25%	**-0.85%**
51-FP	-1.19%	-5.17%	**-0.91%**
FR	-0.42%	-1.55%	**-1.08%**

Table 2: Lens motion reduction and its effect on sharpness after applying smoothing methods for each objective. Compared with conventional methods, our *online* BLSTM and WMA have reduced lens motion significantly, with a very small loss in sharpness.

Quantitative Results Table 1 shows BLSTM average accuracy for each objective. Our BLSTM achieves an average accuracy of 91.6%, and is especially good for the global objective. This is because the global objective has a single ROI and usually involves fewer lens movements. Compared to other objectives, the 51-FP has slightly lower scene accuracy due to the fact it has 51 ROIs and usually involves more lens movements. In general, these results show that our proposed BLSTM is able to learn smoothed lens motion patterns and generalize for different scenes by testing on independent scenes that have never been seen by the network during training. Recall that the WMA method is unsupervised and we cannot evaluate its accuracy, because there is no ground truth data.

The accuracy in Table 1 evaluates the model based on the *offline* ground truth data. This accuracy metric suffers



(a) Scene 6, 9-FP objective

(b) Scene 6, FR objective

Figure 6: A comparison of lens positions between the conventional (Conv.), BLSTM, and WMA. *Offline* SG is also shown as this method is used to train the BLSTM model. The plot above presents the lens positions over time and the one below shows the effect on sharpness value on the same timeline. The total number of lens movements for each method is shown in the plot's legend. BLSTM and WMA are able to suppress small lens fluctuations without substantially affecting the sharpness.

from cumulative error for *online* evaluations. For example, constructing lens movement trajectory for the whole video using *offline* evaluated samples results in propagating the error of any misclassified sample. To that end, we measure the trained BLSTM model's ability to predict *online* stream data of consecutive lens positions. This *online* BLSTM receives the optimal value o_i coming from the PDAF module at each clock cycle i where the lens position x_i^* is predicted based on the previously observed sample $\mathbf{X}_{i-1}$. In a repeated way and by receiving o_i at a time, our *online* BLSTM is able to build the lens movements for the whole video.

In Table 2, we report the lens motion reduction and sharpness change of smoothing methods applied for different AF objectives. Compared with the conventional AF, our *online* BLSTM and WMA are able to suppress the small lens fluctuations of the conventional lens positions while maintaining reasonable sharpness values. Interestingly, by looking at the overall performance, the unsupervised WMA performs slightly better than the supervised BLTM in terms of sharpness change. However, the WMA has a higher time delay compared to the BLSTM, where the WMA method achieved an average correlation of 0.88 and the BLSTM achieved a high average correlation of 0.96.

Qualitative Results Fig. 6 illustrates the similarity between different lens motion trajectories: the conventional, the *online* BLSTM, and the *offline* SG method used for training, and the *online* WMA. This figure also plots the corresponding sharpness values over time. Fig. 6a shows a comparison between the conventional, BLSTM, and SG methods applied on the data from Scene 6 with the 9-FP objective. Fig. 6b shows a comparison between the conventional and WMA methods applied on the data from Scene 6 with the FR objective. The total number of lens movements for each method is shown in the plot's legend. Our *online* BLSTM produces lens motion trajectories quite similar to the *offline* method trajectories used for training. Besides, both BLSTM and WMA have significantly fewer lens movements compared to the conventional method, in which fewer lens movements are highly preferred, as found in [1]. See supplemental materials for additional examples, including videos and other visual comparisons.

5. User Study

A user study was conducted to investigate user preference for different smoothing methods against the conventional one. We conducted two user studies: (1) to determine if users prefer *offline* smoothed video over the conventional method, and (2) to determine if there is any particular preference for the different *online* smoothing methods.

For both studies, we defined scene number and objective as our independent variables, and user preference as our dependent variable. A force-choice paired comparison approach was adopted that requires each participant to select a single preferred video from a pair of videos. Both videos in a given pair are taken from the same scene and objective, but they have different methods.

For the first study, we picked six cases (scene with a certain objective) from the dataset (Section 3.1) to compare conventional vs. ideal. This study is provided to demonstrate the user preference for smoothed lens movements. For the second study, we compared user preference of the two *online* methods and the conventional method. We also introduced another dummy *online* method that is always out of focus with zero lens movements (maximum smoothness).

Specifically, five (5) scenes were used for the conventional, BLSTM, WMA, and out-of-focus (OOF) videos.

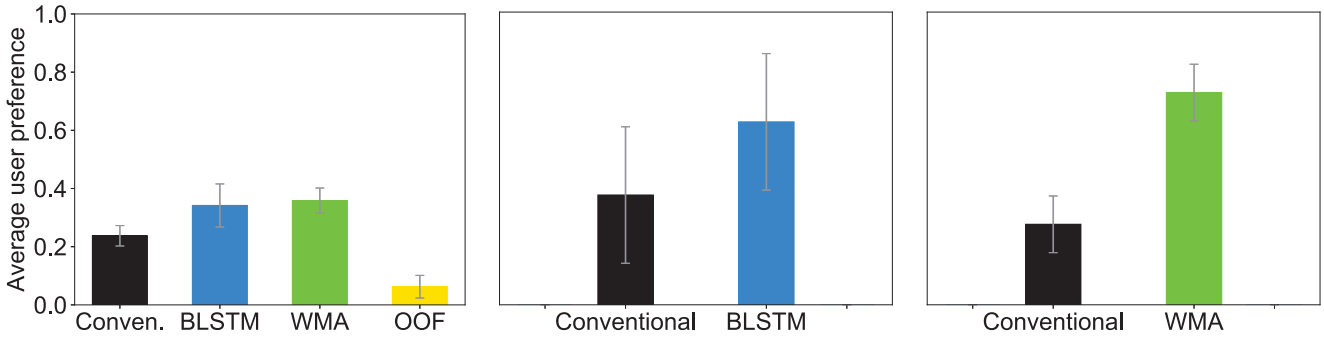


Figure 7: Results of our user study. In the first plot, all methods were preferred over the out-of-focus (OOF) method. Direct comparisons, in the second and third plots, show that BLSTM and WMA were significantly preferred over the conventional.

The total number of paired comparisons for both studies is $6 \times \binom{2}{2} + 5 \times \binom{4}{2} = 36$.

For each video pair, eight opinions were collected from 32 participants (10 females and 22 males) ranging in age from 18 to 50. We designed a graphical user interface (GUI) that allows the user to view video pairs, one pair after the other. This interface allows the participant to watch the two videos in the current pair any number of times before making a selection of the preferred one and proceeding to the next pair. Each participant was shown nine pairs selected in a random manner. The survey takes on average four minutes to complete. The experiments were performed indoors with calibrated monitors and controlled lighting.

We aggregated user votes for each method into an overall score that represents user preference by counting the number of times each method is preferred over another. This overall score is represented as a normalized average user preference. In addition, we calculated the 95% confidence intervals for these results and represented them by the error bars. For the first study, the right-hand plot in Fig. 1 clearly shows the ideal smoothed methods were significantly preferred over the conventional one. The difference in the first study was statistically significant ($F = 49$, $p < 0.0002$). For the second study, the first plot in Fig. 7 demonstrates the normalized average user preference for conventional vs. BLSTM vs. WMA vs. OOF. The difference in the second study was statistically significant ($F = 38.416$, $p < 0.000001$). The first plot in Fig. 7 shows that all methods were preferred over OOF, even though it had zero lens movements, but it was the least preferred as it offered out-of-focus video. Both BLSTM and WMA were preferred over the conventional method as they smooth and reduce lens movements without affecting the video sharpness. WMA , with smaller error bar, was slightly preferred over BLSTM. The first plot does not reflect the actual number of times that BLSTM and WMA were selected over conventional, because we take the aggregate of user votes and the conventional method was selected many times over OOF. To show comparisons of two methods independently, we consider direct comparisons of conventional vs. BLSTM and conventional vs. WMA and present them in the second and third plots respectively. From the second and third plots we can clearly see that our proposed *online* methods (i.e., BLSTM and WMA) were significantly preferred.

6. Conclusion

This paper has proposed two methods for video AF that perform *online* lens motion smoothing. Our method is motivated by a user study in prior work [1] that indicated users prefer smooth videos. While this may seem an intuitive finding, to the best of our knowledge, there is no literature explicitly examining smooth lens motion for video AF. This has motivated us to perform our own user study to show that users do prefer processed versions of conventional algorithms where the lens motion has been smoothed.

Based on this finding, we have proposed two algorithms to perform *online* lens smoothing. The first is a BLTSM learning-based method trained on smoothed lens trajectories. The second is a simple WMA. Both methods were significantly preferred over conventional AF. Interestingly, the simple WMA was slightly more preferred than the learning-based method. This is encouraging as the WMA can be easily incorporated into existing AF hardware.

Our experiments were enabled by a recent AF dataset and research platform [1]. This dataset, however, contains only ten scenes with relatively low temporal sampling (yet was still over 33,000 images). This encourages the camera manufacturers to provide low-level hardware access to AF features to enable more research on this topic.

Acknowledgments This study was funded in part by the Canada First Research Excellence Fund for the Vision: Science to Applications (VISTA) programme and an NSERC Discovery Grant. Dr. Brown contributed to this article in his personal capacity as a professor at York University. The views expressed are his own and do not necessarily represent the views of Samsung Research.

References

[1] A. Abuolaim, A. Punnappurath, and M. S. Brown. Revisiting autofocus for smartphone cameras. In *ECCV*, 2018.

[2] P. M. Atkinson, C. Jeganathan, J. Dash, and C. Atzberger. Inter-comparison of four models for smoothing satellite sensor time-series data to estimate vegetation phenology. *Remote Sensing of Environment*, 123:400–417, 2012.

[3] R. Chen and P. van Beek. Improving the accuracy and low-light performance of contrast-based autofocus using supervised machine learning. *Pattern Recognition Letters*, 56:30–37, 2015.

[4] K.-S. Choi, J.-S. Lee, and S.-J. Ko. New autofocusing technique using the frequency selective weighted median filter for video cameras. *IEEE Transactions on Consumer Electronics*, 45(3):820–827, 1999.

[5] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[6] K. M. de Beurs and G. M. Henebry. Spatio-temporal statistical methods for modelling land surface phenology. In *Phenological research*. 2010.

[7] R. De Levie. *Advanced Excel for scientific data analysis*. Book. Oxford University Press, 2004.

[8] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, speech and signal processing*, 2013.

[9] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005.

[10] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In *IEEE Press. A Field Guide to Dynamical Recurrent Neural Networks*, 2001.

[11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[12] D. Inoue and H. Takahashi. Focus detecting device and camera system using the same device, 2009. US Patent 7,577,349.

[13] J. Jang, Y. Yoo, J. Kim, and J. Paik. Sensor-based auto-focusing system using multi-scale feature extraction and phase correlation matching. *Sensors*, 15(3):5747–5762, 2015.

[14] J. Jeon, J. Lee, and J. Paik. Robust focus measure for unsupervised auto-focusing based on optimum discrete cosine transform coefficients. *IEEE Transactions on Consumer Electronics*, 57(1), 2011.

[15] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] J.-S. Lee, Y.-Y. Jung, B.-S. Kim, and S.-J. Ko. An advanced video camera system with robust af, ae, and awb control. *IEEE Transactions on Consumer Electronics*, 47(3):694–699, 2001.

[18] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot. Global context-aware attention lstm networks for 3d action recognition. In *CVPR*, 2017.

[19] K. Ohsawa. Focus detecting device and method of operation, 1996. US Patent 5,530,513.

[20] M. Rhudy, B. Bucci, J. Vipperman, J. Allanach, and B. Abraham. Microphone array analysis methods using cross-correlations. In *ASME 2009 International Mechanical Engineering Congress and Exposition*, pages 281–288. American Society of Mechanical Engineers, 2009.

[21] A. Savitzky and M. J. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

[22] P. Śliwiński and P. Wachel. A simple model for on-sensor phase-detection autofocusing algorithm. *Journal of Computer and Communications*, 1(06):11, 2013.

[23] Q. Sun, S. Lee, and D. Batra. Bidirectional beam search: Forward-backward inference in neural sequence models for fill-in-the-blank image captioning. In *CVPR*, 2017.

[24] Q. K. Vuong and J.-w. Lee. Initial direction and speed decision system for auto focus based on blur detection. In *IEEE International on Conference Consumer Electronics*, 2013.

[25] J. Wang, W. Jiang, L. Ma, W. Liu, and Y. Xu. Bidirectional attentive fusion with context gating for dense video captioning. In *CVPR*, 2018.

[26] H.-C. Yu, T.-Y. Lee, S.-K. Lin, L.-T. Kuo, S.-J. Wang, J.-J. Ju, and D.-R. Huang. Low power consumption focusing actuator for a mini video camera. *Journal of applied physics*, 99(8):08R901, 2006.

[27] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, X. Xie, et al. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *AAAI*, 2016.