LEARNING CAMERA AUTOFOCUS BY NEURAL NETWORKS WITH REINFORCEMENT LEARNING

Po-Hsun Su and Yu-Chiang Frank Wang

Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan

ABSTRACT

We present a novel approach to camera autofocus (AF) by neural networks derived by reinforcement learning. Our method converts camera AF into an optimization task, which learns the search strategy to sample the observed focus profile, so that the focus point with the highest focus value can be automatically determine within a limited number of iterations. Different from most existing AF methods, we do not apply any predetermined rules or approximated distribution models for sampling the focus profile. Our experiments verify the effectiveness of our approach, which is shown to perform favorably against existing and popular AF methods.

Index Terms— Autofocus, reinforcement learning, neural networks

1. INTRODUCTION

To capture an image at a proper focus point, autofocus (AF) aims to adjust the camera lens so that the resulting image sharpness can be properly preserved. Figure 1 shows an example of a collection of focus values over different lens positions (i.e., focus profile), produced by sweeping the lens along the optical axis of a camera. Note that the focus value indicates the image sharpness at a particular lens position, which can be calculated by different focus measurements [1, 2].

From Figure 1, we see that the focus point is at the lens position with the maximum focus value. In practice, it is not desirable to exhaustively sample the entire profile for determining the focus point due to computation and power consumption. A search strategy that is able to identify the peak of the focus profile within a few samples would be preferable. A popular and simplest way is to apply a rule-based search strategy, which assumes that the focus profile would be bell-shaped and can be divided into three regions: flat, ramp, and peak (see Figure 1) [3]. Such methods first determine the region of the current lens position, and adjust the step size when sampling the focus profile. That is, a larger step size is applied at flat regions for saving computation time, while a smaller one would be required around peak regions for accuracy guarantees [3].

In contrast to rule-based methods for AF, modeling-based approaches choose to parameterize the focus profiles for

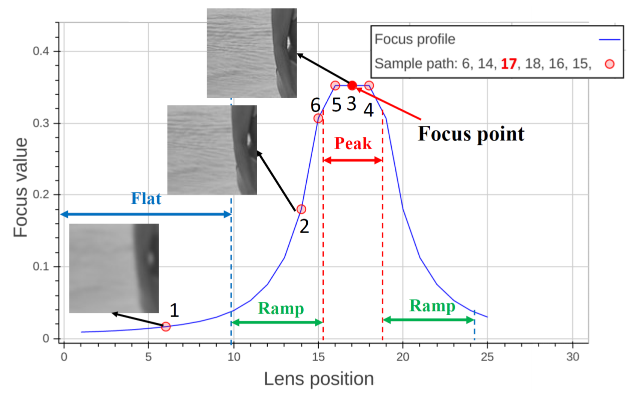


Fig. 1: An example of the focus profile, which is typically divided into regions of flat, ramp, and peak regions by rule-based AF methods. Our method learns the search strategy to identify the focus point within $T = 6$ iterations (as noted by the sample order and the associated lens positions).

searching the focus point. For example, existing methods like [4, 5, 6] propose search strategies for AF based on the assumptions that the focus profiles can be approximated by Gaussian or Cauchy distributions. With such assumptions, the effective search range over the focus profile would be limited, so that such a range would fit the assumed distribution well. And, in practical scenarios, the focus profiles might be complex or noisy, so that the use of the above distributions will not be sufficiently representative for AF purposes.

To alleviate the concerns for existing rule and model-based AF approaches, we propose a learning-based algorithm in this paper. More specifically, we present a reinforcement learning framework with neural networks to determine the search strategy for camera AF. Given training image data (or focus profiles), our method learns an effective model which is able to predict the sample points for the focus profile, with the goal to predict the focus point with the peak focus value within a limited number of samples. The details of our proposed method will be presented in Section 2.

The contributions of this paper are highlighted below:

- Based on the recent advances of (deep) neural networks, we are the first to approach the task of AF by solving a reinforcement learning problem.

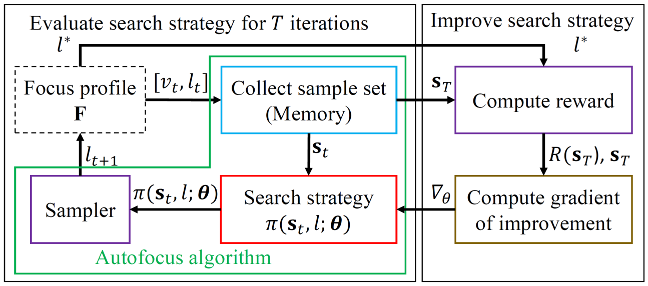


Fig. 2: The block diagram of our proposed framework.

- Our method learns the AF search strategy for sampling the focus profile, which exhibits the ability to predict the focus point with the peak focus value.
- Different from existing rule or modeling-based methods, our method does not require predetermined rules or approximated distribution models for AF.

2. OUR PROPOSED METHOD

2.1. Problem Definition

Figure 2 illustrates our proposed framework for learning the AF search strategy. Suppose that there are N lens positions over a focus profile $\mathbf{F} \in R^N$, in which each attribute denotes the corresponding focus value. For the tth iteration during the sampling process, we sample the lens position l_t and observe the focus value of $F(l_t) = v_t$. Thus, a sample set $\mathbf{s}_t = [\mathbf{v}_t, \mathbf{l}_t] \in R^{t\times 2}$ up to the tth iteration can be collected, where $\mathbf{v}_t = [v_1, \ldots, v_t]^\intercal$ and $\mathbf{l}_t = [l_1, \ldots, l_t]^\intercal$.

In this paper, we propose a learning-based approach for determining the AF search strategy. By observing a set of training focus profiles with various focus points, we learn an optimal search strategy in terms of a probability function $\pi(\mathbf{s}_t, l)$, which reveals l_{t+1} at the following iteration given the observed sample set $\mathbf{s}_t$. With the derived search strategy and the maximum number of iterations T for sampling the focus points, one can determine the focus point by the sample with the maximum focus value. In other words, our goal is to maximize the probability P that the focus point l^* will be sampled within a limited number of iterations T, i.e.,

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, P(l^* \in \mathbf{s}_T|\pi). \tag{1}$$

2.2. Reinforcement Learning for AF Search Strategy

As noted above, we view camera AF as an optimization task. Without predetermined rules or distributions to approximate the focus profile, we learn the search strategy π by solving (1), which produces the focus point from the sampled focus profile within T iterations. However, this process *cannot* be solved in a supervised manner for each sample $\mathbf{s}_t$, since no focus point will be set until having the complete sample set $\mathbf{s}_T$. Therefore, we advance *reinforcement learning* for solving this particular optimization task.

In the areas of machine learning, reinforcement learning [7] focuses on learning behaviours within a specific context, which maps the observed situations to particular actions by maximizing the numerical reward signals. In other words, the learner needs to search for various sequences of actions for determining the one with the most reward feedback [8, 9]. For the task of camera AF, we need to learn a proper search strategy for sampling the focus profile, aiming at observing the focus point with the peak focus value within T iterations.

In order to convert the above task into a reinforcement learning problem, we have the associated reward function $R(\mathbf{s}_T)$ determined as:

$$R(\mathbf{s}_T) = \begin{cases} 1 & \text{if } l^* \in \mathbf{s}_T \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

It can be seen that, $R(\mathbf{s}_T)$ returns the reward of 1 if the in-focus lens position is observed in $\mathbf{s}_T$ (and 0 otherwise).

With the above definition, the expected reward on the focus profile $\mathbf{F}$ with search strategy π would indicate the objective probability function. That is,

$$\mathbf{E}_\pi[R] = P(l^* \in \mathbf{s}_T|\pi) \tag{3}$$

In our proposed reinforcement learning framework, we choose to maximize the above expected reward by taking its gradient information during the optimization process. This is based on the stochastic policy gradient theorem [10]. More precisely, assuming that the search strategy π is differentiable and is parameterized by θ, the gradient of the expected reward with respect to θ can be computed by:

$$\nabla_\theta \mathbf{E}_\pi[R] = \mathbf{E}_\pi[R\nabla_\theta \log \pi(\mathbf{s}_t, l_{t+1}; \theta)] \tag{4}$$

The right hand side of (4) indicates that we need to increase the log probability for sampling the lens position l_{t+1} given $\mathbf{s}_t$ by the amount, which is proportional to the expected reward.

To avoid the high computation cost of the expectation calculation, the reward $R(\mathbf{s}_T)$ can serve as an estimate of the expectation. That is,

$$\nabla_\theta \mathbf{E}_\pi[R] \sim R(\mathbf{s}_T)\nabla_\theta \log \pi(s_t, l_{t+1}; \theta) \tag{5}$$

For each sequence of samples $\mathbf{s}_t$ generated by π, we apply (5) to update the search strategy for the $(t + 1)$th iteration until the maximum number of iteration T is reached.

2.3. Reinforcement Learning with Neural Networks for Camera Autofocus

In our proposed framework, we choose to utilize feedforward neural networks (NN) as the learner, i.e., the search strategy $\pi(\mathbf{s}_t, l; \theta)$ in Figures 2 and 3, since its gradient $\nabla_\theta \pi(\mathbf{s}_t, l; \theta)$ can be efficiently calculated by back propagation.

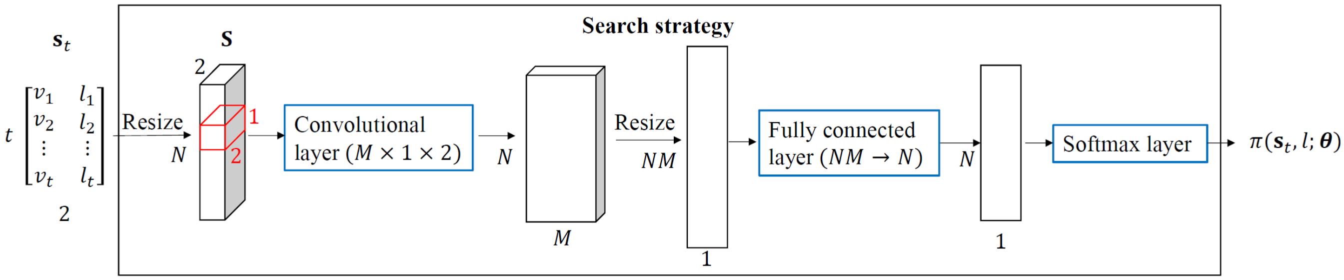


Fig. 3: Our feedforward NN for learning the AF search strategy for sampling the focus profile.

Algorithm 1: Learning Feedforward NN for AF

Input: A training set of focus profiles $\widetilde{\mathbf{F}}$, learning rate α, momentum m, and parameter M
Output: NN parameter θ for $\pi(\mathbf{s}, l; \theta)$
Initialize θ
while *keep learning* **do**
  Sample a focus profile $\mathbf{F}$ from $\widetilde{\mathbf{F}}$
  Initialize $\mathbf{s}_0 = \{\}$
  Sample initial lens position l_1 from $\pi(\mathbf{s}_0, l; \theta)$
  for $t \leftarrow 1$ **to** T **do**
    Sample focus value $v_t = \mathbf{F}(l_t)$
    Insert $\{v_t, l_t\}$ and $\mathbf{s}_{t-1}$ to $\mathbf{s}_t$
    Sample next lens position l_{t+1} from $\pi(\mathbf{s}_t, l; \theta)$
  Compute target lens position $l^* = \underset{l}{\operatorname{argmax}}\, \mathbf{F}(l)$
  Compute reward $R(\mathbf{s}_T)$
  Initialize gradient $\nabla_\theta = \mathbf{0}$
  for $t \leftarrow 0$ **to** $T-1$ **do**
    Compute and accumulate gradients
    $\nabla_\theta \leftarrow \nabla_\theta + R(s_T)\nabla_\theta \log \pi(\mathbf{s}_t, l_{t+1}; \theta)$
  update momentum gradient $\delta \leftarrow m\delta + \nabla_\theta$
  update parameter $\theta \leftarrow \theta + \alpha\delta$

Algorithm 2: Focus Point Prediction

Input: NN parameter θ and an unseen focus profile F
Output: Focus point $\hat{l}$
Initialize $\mathbf{s}_0 = \{\}$
Pick initial lens position $l_1 = \underset{l}{\operatorname{argmax}}\, \pi(\mathbf{s}_0, l; \theta)$
for $t \leftarrow 1$ **to** T **do**
  Sample focus value $v_t = F(l_t)$
  Insert $\{v_t, l_t\}$ and $\mathbf{s}_{t-1}$ to $\mathbf{s}_t$
  Pick next lens position $l_{t+1} = \underset{l}{\operatorname{argmax}}\, \pi(\mathbf{s}_t, l; \theta)$
Determine t^* by $t^* = \underset{t}{\operatorname{argmax}}\, \mathbf{v}_t$
Output the predicted focus point $\hat{l} = l_{t^*}$

We now rewrite (5) by the chain rule as follows:

$$R(\mathbf{s}_T)\nabla_\theta \log \pi(\mathbf{s}_t, l_{t+1}; \theta) = \frac{R(\mathbf{s}_T)\nabla_\theta \pi(\mathbf{s}_t, l_{t+1}; \theta)}{\pi(\mathbf{s}_t, l_{t+1}; \theta)}, \quad (6)$$

in which the model parameter θ can be updated by back propagating a loss $-R(\mathbf{s}_T)/\pi(\mathbf{s}_t, l_{t+1}; \theta)$ throughout the network. With the recent development of deep learning software like [11], the above implementation can be realized easily.

Figure 3 shows the proposed NN model for learning the optimal AF search strategy. To deal with fixed-size inputs, we convert the observed sample set $\mathbf{s}_t \in R^{t\times 2}$ at each iteration t into a fixed-size matrix $\mathbf{S} \in R^{N\times 2}$ instead, in which N denotes the total number of lens positions (see Section 2.1). The first column $\mathbf{f} = [f_1, \ \ldots, f_N]^\mathsf{T}$ in this input matrix $\mathbf{S}$ records the sampled focus value at the associated lens position (i.e., $f_{l_t} = v_t$ if l_t is sampled at iteration t). On the other hand, the second column $\mathbf{c} = [c_1, \ \ldots, c_N]^\mathsf{T}$ notes the number of sample counts at each lens position. In our work (and for practical scenarios), we have $t \le T << N$.

From Figure 3, it can be seen that our NN model consists of a convolutional layer, a fully connected layer, and a softmax layer [12]. For each possible lens position, the convolutional layer applies M kernels of size 1×2 to extract the information from each sampled focus value and its sample counts from the input. And, the output of this convolutional layer will be converted into as a vector of NM dimensions.

Next, the fully connected layer takes the output of the convolutional layer and associates the information across lens positions. It produces a N-dimensional vector with each entry indicating the probability of the corresponding lens position as the next sample point. Finally, a softmax layer normalizes the probability, which reveals the next sample point.

Standard Hyperbolic tangent non-linearity (omitted in Figure 3 for simplicity) is added after both the convolutional and fully connected layers. This allows our NN to exhibit non-linear representation capabilities. With our NN model, the training and test stages of our AF learning framework are summarized in Algorithms 1, and 2, respectively.

3. EXPERIMENTS

3.1. Dataset and Settings

To evaluate the performance of our approach, we consider 42 natural images from the image database of [13] (30 images for training, and the remaining 12 for testing). A number of

Table 1: MSE between the predicted and ground truth focus points.

Methods\iter.	5	6	7	8
Ours	**0.62**	**0.14**	**0.01**	**0.00**
Fibonacci	1.41	0.54	0.26	0.04
Rule-based	13.6	3.7	1.08	0.63

Table 2: Average sample distance. Note that the rule-based approach was not able to achieve comparable MSE as ours did.

Methods\iter.	5	6	7	8
Ours	20.0	**22.5**	**24.0**	29.7
Fibonacci	32.7	36.4	38.2	39.6
Rule-based*	**17.5**	24.0	25.8	**25.4**

10000 and 1000 patches of size 100 $\times$ 100 are randomly extracted from the training and test images, respectively. The focus point l^* ranging from $\{1, \ldots, N\}$ is randomly assigned for each patch, while its focus blurred versions at the remaining $N-1$ lens positions can be synthesized by Gaussian blur kernels with standard deviations of:

$$\sigma(l) = \sigma_{max} \frac{|l - l^*|}{N}. \tag{7}$$

Note that σ_{max} is fixed as 5 in our experiments. The focus value of each blurred patch is computed by the focus measurement of the energy of Laplacian [14].

In our experiments, we set the total number of lens position $N = 25$, and consider the maximum number of iterations (i.e., samples) T from 5 to 8. In other words, we allow at most $1/3$ of the lens positions to be sampled for predicting the focus point. We consider the performance metrics of mean square error (MSE) between the locations of the predicted and ground truth focus points, and the average sample distance measured by accumulating the travel distance across samples. The results of Fibonacci search [15] and rule-based approach [3] whose parameters are tuned with the training dataset are included for comparisons.

3.2. Evaluation

Tables 1 and 2 list the MSE and average sample distance of different approaches, respectively. Given a fixed number of iterations, we see that our approach achieved the best MSE among all methods, while the average sample distance was also the minimum compared to the others under similar MSE. These results verify that our proposed AF algorithm based on NN-based reinforcement learning is able to address the task of camera AF effectively and efficiently.

Figure 4 visualizes an example of the observed sample points $\mathbf{s}_T$, which was produced by our method with T set as 6 (i.e., only $1/4$ of the lens positions allowed to be sampled). The sample points of the base line method are also shown. It can be seen that, we successfully identified the lens position with the largest focus value at the 4th iteration.

In Figure 5, we provide two more examples with the same $T = 6$. In Figure 5(a), we see that our learned strategy was

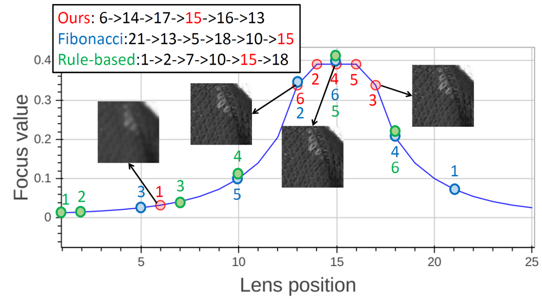
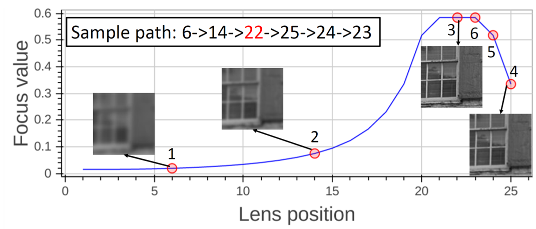
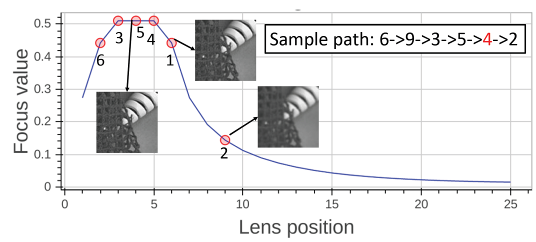


Fig. 4: Example sampling results of different methods. The numbers in red denotes the iteration with focus point identified.



(a)



(b)

Fig. 5: Example sampling results of our method. The number in red denotes the iteration with focus point identified.

able to quickly skip over the flat region for identifying the focus point at the 3rd iteratoin. As for Figure 5(b), our method sampled around ramp and peak regions for AF without sampling redundant points at the flat regions. From the above examples, it can be seen that our AF search strategy is able to amend the search direction along the focus profile, or to ignore the search regions not of interest. This further confirms the effectiveness of our proposed AF method.

4. CONCLUSION

We proposed to approach the task of camera autofocus by solving a reinforcement learning problem. With recent advances of deep learning, we particularly utilize feedforward neural networks for learning the optimal AF search strategy. With the derived search strategy, we are able to identify the focus point of an unseen focus profile within a limited sample numbers. Compared to existing AF approaches, we do not require the predetermine rules nor the use of approximated shape models to determine such strategies. From our experiments, we confirmed that our proposed NN-based method was able to achieve promising results, which supports the use of our NN model for practical camera AF problems.

5. REFERENCES

[1] S.-Y. Lee, J.-T. Yoo, Y. Kumar, and S.-W. Kim, "Reduced energy-ratio measure for robust autofocusing in digital camera," *Signal Processing Letters, IEEE*, vol. 16, no. 2, pp. 133–136, 2009.

[2] M. Kristan, J. Pers, M. Perse, and S. Kovacic, "A bayes-spectral-entropy-based measure of camera focus using a discrete cosine transform," *Pattern Recogn. Lett.*, vol. 27, no. 13, pp. 1431–1439, 2006.

[3] Y. Yao, B. R. Abidi, and M. A. Abidi, "Extreme zoom surveillance: System design and image restoration," *Journal of Multimedia*, vol. 2, no. 1, pp. 20–31, 2007.

[4] S. K. Nayar and Y. Nakagawa, "Shape from focus," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 8, pp. 824–831, 1994.

[5] M. Subbarao and J.-K. Tyan, "Selecting the optimal focus measure for autofocusing and depth-from-focus," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 8, pp. 864–870, 1998.

[6] D.-C. Tsai and H. H. Chen, "Reciprocal focus profile," *Image Processing, IEEE Transactions on*, vol. 21, no. 2, pp. 459–468, 2012.

[7] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[8] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[9] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems 27*, pp. 2204–2212. 2014.

[10] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Neural Information Processing Systems 12*, pp. 1057–1063. 1999.

[11] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, 2011.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. 2012.

[13] X. Li, B. Gunturk, and L. Zhang, "Image demosaicing: a systematic survey," *Proceedings of SPIE*, vol. 6822, pp. 68211J, 2008.

[14] M. Subbarao, T. Choi, and A. Nikzad, "Focusing techniques," *Journal of Optical Engineering*, vol. 32, pp. 2824–2836, 1993.

[15] E. P. Krotkov, *Active Computer Vision by Cooperative Focus and Stereo*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1989.